

The NeuroAI Dialogue

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 1 of "The NeuroAI Handbook"

Eight decades of borrowing between brains and machines, and the one mistake to avoid.

One-sentence thesis. Neuroscience and AI are two answers to the same question, "how does intelligence work?", and their progress comes from a running conversation, not from either field copying the other outright.

PART 1: Two Fields, One Question

The Core Question

Start with the question, not the wiring diagram. *How does intelligence work?* Two fields have chased that question from opposite ends for a century. **Neuroscience** starts with the hardware, the wet and messy biological brain, and works backward to the principles. **Artificial intelligence** starts with the principles, logic and math and computation, and works forward to build a mind from scratch. This handbook is about the dialogue between them: a decades-long conversation in which each side has taken turns inspiring, challenging, and advancing the other.

The Neuron as a Brilliant Caricature

At the heart of the whole dialogue sits one deliberately simple idea. A biological neuron is a riot of ion channels, neurotransmitters, and branching dendrites; the **artificial neuron** throws almost all of that away and keeps three moves: integrate the inputs, apply a transformation, emit an output. No single artificial unit captures real cellular physiology, yet the abstraction has been rich enough to build working

systems and to test theories of mind.

In symbols, the original McCulloch and Pitts (1943) unit fires when a weighted sum clears a threshold:

$$y = 1[\sum_i w_i x_i \geq \theta]$$

Watch out. A caricature is not a miniature copy. Resemblance alone does not prove that a deep ReLU stack computes the way a cortical column does. Any claim that a model works "just like the brain" has to say *at which level* the two align, because you can match the algorithm without matching the implementation.

ANALOGY

A political cartoonist exaggerates the nose and drops the skin pores. The drawing is useful *because* it discards detail, not despite it. The artificial neuron is that cartoon of a real cell: recognizable, portable, and wrong in every fine particular.

A Shared Language: Marr's Levels

To keep the conversation honest, the fields borrow David Marr's three levels of analysis. They are the antidote to loose "it's just like the brain" talk: before comparing a brain and a network, say which level you mean. A plane does not fly by flapping, yet it obeys the same aerodynamics as a bird, agreement at the computational level, divergence at the implementational one.

Level	Question	In the brain	In AI
Computational	<i>Why?</i> What problem is solved	Recognize objects despite lighting and angle	The classification objective
Algorithmic	<i>What?</i> Representations and steps	Edges to parts to objects, a hierarchy	Convolution, pooling, feature maps
Implementational	<i>How?</i> Physical substrate	Neurons, synapses, ion channels	Transistors, matrix multiplies

PART 2: Three Acts of the Dialogue

ACT I: The Spark 1940s–1960s	ACT II: Divergence 1970s–1980s	ACT III: Re-convergence 1980s–present
McCulloch–Pitts 1943 Perceptron 1958	<i>Perceptrons</i> 1969 XOR → AI winter	Backprop 1986 ImageNet 2012, AlphaGo 2016

Eight decades in three acts: inspiration, estrangement, reunion.

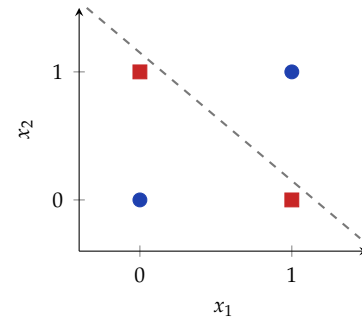
Act I: The Spark (1940s–1960s)

The first insight was that *thinking could be computation*. McCulloch and Pitts showed that networks of threshold units can, in principle, compute any logical function. The mysteries of mind suddenly had a footing in formal logic. Then Frank Rosenblatt's **Perceptron** (1958) made such a network *learn*: an error-correction rule nudges the weights down whenever the output is wrong. Running underneath both was **cybernetics**, Wiener's feedback controllers and Ashby's homeostats, a shared vocabulary of goals and error signals that kept engineers and physiologists in the same room.

Act II: The Great Divergence (1970s–1980s)

The excitement met a wall. Minsky and Papert's *Perceptrons* (1969) proved that a single-layer network cannot represent functions that are not linearly separable, the XOR being the famous counterexample. Multilayer networks existed, but nobody could train them efficiently. Starved of compute and data, the fields drifted apart into the first **AI winter**: AI retreated to symbolic logic and expert systems, while neuroscience dove into ion channels and molecules. They still asked the same questions about learning and rep-

resentation, but stopped speaking the same language.



XOR: circles are one class, squares the other. No straight line splits them, which is exactly what a single perceptron can draw.

Act III: The Grand Re-Convergence (1980s–now)

The wall came down with **backpropagation** (Rumelhart, Hinton, and Williams 1986; foreshadowed by Werbos 1974), an efficient way to compute gradients layer by layer and finally train the multilayer networks Minsky and Papert had stumped. The 2010s deep-learning wave, big data plus GPUs plus better architectures, turned that into ImageNet (2012), AlphaGo (2016), and large language models. The two fields snapped back into alignment, and the conversation has been louder ever since.

Neuroscience Inspires AI

The clearest debts run from brain to machine. Hubel and Wiesel's map of the visual cortex, simple cells feeding complex cells feeding higher areas, inspired Fukushima's Neocognitron (1980) and then the **convolutional neural network** (LeCun 1989, 1998): local filters, pooling, a rising hierarchy of features. A second debt is about reward. The brain's dopamine neurons signal a **reward prediction error**, the gap between the reward you got and the reward you expected, and that quantity is almost exactly the temporal-difference error at the core of modern **reinforcement learning** (Sutton 1988; Sutton and Barto 2018).

AI Advances Neuroscience

The arrow now points both ways. Deep learning and probabilistic models decode neural population activ-

ity, reconstruct connectomes, and stand in as candidate models of cortical computation, task-optimized networks used as *testable theories* of what a brain area computes. The tools are getting industrial: Kilosort-class sorters pull spikes from million-channel recordings in near real time, and transformer decoders translate an animal's pose straight into predicted neural activity. Raw recordings become hypotheses.

ANALOGY

Think of it as two mapmakers trading survey notes. Neuroscience hands AI a rough sketch of a working intelligence; AI hands back precision instruments that redraw the original map. Neither could get as far surveying alone.

PART 4: Touchpoints and the Efficiency Gap

From Membrane to Matrix Multiply

Both a neuron and an artificial unit follow the same three-beat rhythm, integrate, transform, emit, but the biological version lives in continuous time. A **leaky integrate-and-fire** neuron charges toward its input and leaks back toward rest, firing when it crosses threshold:

$$\tau_m \frac{dV}{dt} = -(V - V_{rest}) + R_m I(t)$$

The artificial neuron collapses that dynamics into one static line, a weighted sum through a nonlinearity, as with the ReLU $y = \max(0, w^\top x + b)$. What biology spends on timing, chemistry, and dendritic structure,

the artificial unit trades away for speed and trainability. That trade is the whole subject of this book.

The 20-Watt Brain

The gap that should keep engineers humble is energy. A human brain runs the entire show on roughly **20 watts**, performing on the order of 10^{14} synaptic events per second at a fraction of a picojoule each. A modern accelerator burns hundreds of watts and still lands orders of magnitude behind on operations per watt. The brain's tricks, sparse and event-driven spikes, local memory next to compute, no shuttling data across a bus, are exactly what **neuromorphic** and sparse-computing research is trying to recover in silicon.

Quick Self-Test

1. Neuroscience and AI approach intelligence from opposite ends. Describe each starting point in one phrase.
2. Why is calling the artificial neuron a “caricature” more accurate than calling it a “model”? Name one detail it discards.
3. State Marr’s three levels, and explain why “it works like the brain” is meaningless until you name a level.
4. What did Minsky and Papert prove, and why did backpropagation reopen the door they closed?
5. Give one example of neuroscience inspiring AI and one of AI advancing neuroscience.

If You Remember One Thing

The artificial neuron is a caricature, not a copy. That single abstraction is powerful enough to have built modern AI and dangerous enough to mislead you, so whenever you hear that a network is “just like the brain,” ask *at which of Marr’s levels*, because agreement on the goal never guarantees agreement on the machinery.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Neuroscience Foundations

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 2 of "The NeuroAI Handbook"

What a single neuron actually computes, how it spikes, and how the brain learns without breaking.

One-sentence thesis. A single biological neuron is already a small network, and the brain's learning is stable only because three forces cooperate: a rule that changes synapses, a brake that keeps them bounded, and a signal that says *when* the change matters.

PART 1: The Neuron Is Not a Perceptron

Inside a Single Cell

The textbook cartoon is a bag that sums its inputs: dendrites collect signals, the soma adds them up, and the axon hillock fires a spike if the total clears threshold. That cartoon is the artificial neuron. The real cell is richer. Its **dendrites** are not passive wires; they contain active machinery that computes *before* the soma ever sees the result.

Two mechanisms do the heavy lifting. **NMDA spikes** are regenerative plateaus in a dendritic branch that fire only when inputs arrive together, an AND-like coincidence detector. **Calcium spikes** in the apical (top) dendrites let distant, top-down signals gate or amplify whatever the basal dendrites are receiving. Each branch behaves as a quasi-independent nonlinear subunit, and the soma pools their outputs.

The upshot. A neuron approximates a shallow multilayer network in one cell. That extra layer of computation is exactly what improves credit assignment and lets context (top-down input) gate feedforward drive.

ANALOGY

Think of the neuron as a committee, not a single voter. Each dendritic branch is a subcommittee that debates and votes first; the soma only tallies the subcommittees' verdicts. A perceptron is one voter. That is the difference.

A Zoo of Synapses

The connections are not all alike either, and the differences are computational. **AMPA** receptors give fast, roughly linear excitation. **NMDA** receptors are blocked by magnesium until the cell depolarizes, so they act as coincidence detectors and gate plasticity. **GABA** provides inhibition, fast shunting (GABA-A) or slow modulation (GABA-B). Layered on top, neuromodulators like dopamine and acetylcholine adjust gain and plasticity thresholds through slower metabotropic pathways.

Synapses also change strength on the fly. **Short-term plasticity** makes a synapse *facilitate* (grow stronger over a burst) or *depress* (weaken as vesicles deplete). Functionally this is temporal filtering: a facilitating synapse emphasizes sustained bursts, a depressing one emphasizes onsets, a built-in sense of what just changed.

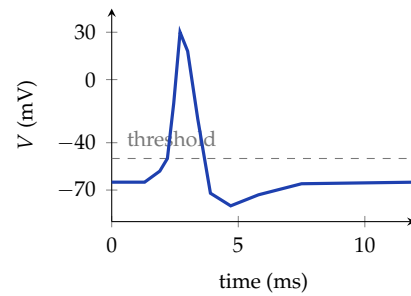
PART 2: How a Spike Happens

The Hodgkin-Huxley Machine

Why is a spike all-or-nothing, with that same sharp shape every time? Hodgkin and Huxley answered this in 1952 (and won a Nobel for it) by treating the membrane as a **capacitor** in parallel with voltage-gated ion channels. Current either charges the membrane or leaks out through sodium, potassium, and leak channels:

$$C_m \frac{dV}{dt} = I_{ext} - I_{Na} - I_K - I_L$$

The genius is in the **gating variables** m , h , n , each the probability that a class of channel gate is open, each with its own speed. The spike is their choreography: depolarization opens fast m gates so sodium rushes in (positive feedback, the upstroke); slow h gates then close and shut sodium off; delayed n gates open and potassium flows out, dragging the voltage back down and briefly past rest. Fast excitation, delayed inhibition, one clean pulse.



The action potential: sodium drives the upstroke, potassium the downstroke and undershoot.

From Biophysics to a One-Line Neuron

You rarely need all four channels. Simpler models trade biophysical detail for speed, and the one AI uses most, the **leaky integrate-and-fire** neuron, keeps only four operations: integrate, leak toward rest, compare to threshold, reset. The ladder below runs from most faithful to most efficient; where you sit on it depends on whether you want mechanism or throughput.

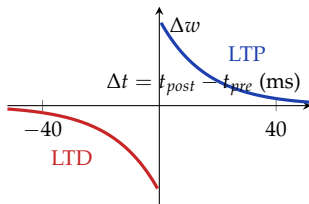
Neurons Adapt

Give a real neuron a constant input and it does *not* fire at a constant rate. It fires fast, then slows, **spike-frequency adaptation**, driven by slow potassium and calcium-activated currents that accumulate with activity. Computationally this is a high-pass filter: the cell shouts about change and whispers about steady state. The same instinct shows up in AI as normalization layers, attention decay, and plain habituation.

Model	Equations	Key feature
Hodgkin-Huxley	4 ODEs	Full biophysics with explicit ion channels; the reference model
FitzHugh-Nagumo	2 ODEs	Phase-plane view of excitability; strips channels to two variables
Izhikevich	2 ODEs + reset	Reproduces most cortical firing patterns very cheaply
Exponential IF	1 ODE + reset	Adds a realistic spike-initiation nonlinearity
Leaky IF	1 ODE + reset	Simplest useful spiking unit; the workhorse in AI models

Cells That Fire Together

Donald Hebb’s slogan, “cells that fire together, wire together,” is the ancestor of every learning rule in this book. Modern experiments sharpened it into **spike-timing-dependent plasticity**: the exact millisecond order matters. If the presynaptic neuron fires *just before* the postsynaptic one, the synapse strengthens (LTP); reverse the order and it weakens (LTD). Timing encodes causality, cause-before-effect gets rewarded.



The STDP window. Pre-before-post (right) strengthens; post-before-pre (left) weakens.

Why Learning Needs a Brake

Left alone, STDP is unstable: strong synapses fire the cell more, which strengthens them further, a runaway loop that saturates every weight. The brain prevents this with **homeostasis**, synaptic scaling that renormalizes a neuron’s total input, and sliding plasticity thresholds (the BCM idea) that make potentiation harder when a cell has been too active. Without a brake, no learning rule stays useful for long. AI rediscovered the same necessity as weight normalization

and regularization.

The Third Factor

Timing tells you *who* fired with whom, but not whether the outcome was any good. The brain adds a third factor: a broadcast neuromodulatory signal, dopamine for reward, acetylcholine for surprise, that gates when a Hebbian change is actually written in. A useful shorthand for these **three-factor rules** is a product of pre-activity, post-activity, and a modulator:

$$\Delta w \propto x_{pre} \cdot x_{post} \cdot M$$

This is the biological answer to **credit assignment**. Backpropagation solves the same problem with a global error signal and symmetric backward weights, biologically implausible but effective. Local plasticity plus a neuromodulatory “now it counts” is the brain’s more distributed alternative.

ANALOGY

A student takes notes constantly (Hebbian pairing), but only burns something into memory when a surprise or a reward marks the moment as important (the dopamine third factor). Constant note-taking without that stamp is just a full notebook nobody rereads.

PART 4: A Map of the Brain for AI

The brain is not one learner but several, each specialized, and each maps onto a familiar AI idea. The **cortex** runs hierarchical predictive processing, the biological cousin of deep learning. The **hippocampus** is a memory index: the dentate gyrus separates similar patterns, area CA3 completes a whole memory from a partial cue, CA1 binds and times, and replay during rest consolidates and plans, the blueprint behind retrieval-augmented and world-model systems.

The **basal ganglia** are an actor-critic: a “Go” pathway (D1) and a “NoGo” pathway (D2) compete to select actions, and a dopamine reward-prediction error, the same quantity as the temporal-difference error, tunes the balance. The **cerebellum** is a supervised learner in the Marr-Albus-Ito mold, with the climbing fiber delivering an explicit teacher signal. The **thalamus** routes and gates traffic between cortical areas, the brain’s attention and control layer.

Brain system	What it computes	AI analog
Cortex	Hierarchical predictive processing	Deep networks
Hippocampus	Pattern separation, completion, replay	Memory nets, RAG, world models
Basal ganglia	Action selection via dopamine reward-prediction error	Actor-critic / TD learning
Cerebellum	Supervised error correction (climbing-fiber teacher)	Supervised learning
Thalamus	Routing, gating, attentional control	Attention and controller modules

Quick Self-Test

1. Why is “a neuron is a single perceptron” wrong? Name the dendritic mechanism that makes a cell act like a shallow network.
2. In the Hodgkin-Huxley spike, what do the sodium and potassium currents each contribute to the waveform?
3. STDP is unstable on its own. What two homeostatic mechanisms keep it in bounds, and why are they needed?
4. What is the “third factor” in a three-factor learning rule, and what problem does it solve that plain Hebbian pairing cannot?
5. Match three brain systems to their AI analogs (cortex, basal ganglia, cerebellum).

If You Remember One Thing

Learning in the brain is not one rule but a balance of three: a Hebbian change (STDP) that rewrites synapses, a homeostatic brake that keeps them bounded, and a neuromodulatory signal that says when the change is worth keeping. Remove any one and the system either fails to learn or tears itself apart, which is the real lesson for anyone building brain-inspired AI.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Spatial Navigation and the Brain's GPS

Study Guide

by Richard J. Young, Ph.D.

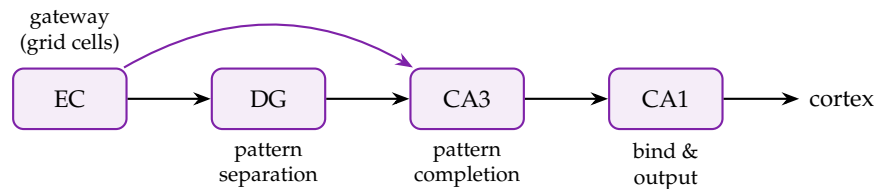
richard@deepneuro.ai

Chapter 3 of "The NeuroAI Handbook"

Place cells, grid cells, and why the brain's map already looks like a graph and an attention head.

One-sentence thesis. The brain navigates by pairing a universal hexagonal ruler (grid cells) with sparse, meaningful pins (place cells), and those two ideas prefigure the graph and the attention mechanism at the heart of modern AI.

PART 1: The Navigation Problem and the Hippocampus



The hippocampal pipeline: input enters through entorhinal cortex, DG separates similar patterns, CA3 completes them from partial cues, CA1 binds and outputs.

The Miracle of Navigation

How do you remember the route to your coffee shop, or how does a squirrel relocate a buried nut after a winter? This ability feels effortless, yet it is one of the brain's most sophisticated feats, run by an internal GPS in the **hippocampal formation**. The chapter's payoff is that the brain's solution to mapping space turns out to rhyme with two workhorses of AI: the graph and the attention mechanism.

The Hippocampal Pipeline

The hippocampus binds the pieces of an experience, *what* happened, *where*, and *when*, into one memory,

and its wiring is a staged pipeline. Sensory input arrives through the **entorhinal cortex (EC)**, the gateway where grid cells live. The **dentate gyrus (DG)** performs *pattern separation*: it forces similar-looking places into distinct representations so memories do not blur together. **CA3** does the opposite, *pattern completion*, recalling a whole memory from a partial cue through its recurrent connections. **CA1** binds and outputs the result. Separation then completion, a build-up of structure from simple inputs, is a theme you will meet again in deep networks.

Place Cells: Pins on the Map

Discovered by John O'Keefe in 1971, a **place cell** is the easiest piece to grasp: a hippocampal neuron that fires only when the animal is in one specific spot, its *place field*. A few properties make it powerful. It **remaps**: enter a new room and the cells instantly build a fresh, independent map, so your "lobby" cell in Paris is not your "lobby" cell in Tokyo. It is **experience-driven**, sharpening for places you care about. And it is **sparse**, only a small fraction of cells are active anywhere, an efficient, robust code that AI prizes.

In symbols, a place field is well modeled as a Gaussian bump peaking at its center $\mathbf{c}$ and fading with distance:

$$r(\mathbf{x}) = A \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2}\right)$$

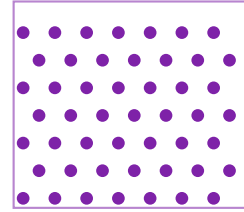
ANALOGY

Place cells are smart pins on a mental pinboard. Enter a cafe and a "cafe" pin lights up; walk to the park and a different pin fires. Each stays silent until you return to its spot, and the whole collection is your map, a map of *meaningful* places, not raw coordinates.

Grid Cells: The Universal Ruler

If place cells are the pins, what is the pinboard? Edvard and May-Britt Moser found the answer in 2005 (a Nobel-winning discovery): **grid cells** in the entorhi-

nal cortex fire not in one spot but at many, arranged in a perfect **hexagonal lattice** tiling the whole environment.



One grid cell's firing peaks. A hexagonal lattice is the most efficient way to tile a 2D plane.

Three things make the grid special. It is **geometrically optimal**, hexagons tile a plane with the fewest cells. It is **multi-scale**: grid cells cluster into modules of different spacings, fine grids for precision and coarse grids for range. And it is **universal**, the same lattice reappears in every environment, a metric the brain can lay over any new space. The hexagon emerges from the interference of three plane waves set 60 degrees apart:

$$g(\mathbf{x}) = \sum_{k=1}^3 \cos(\mathbf{u}_k \cdot \mathbf{x})$$

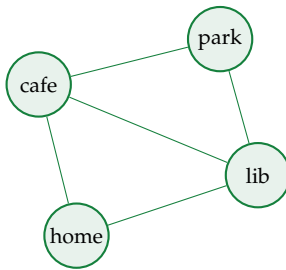
The Supporting Cast

Place and grid cells do not work alone. **Head-direction cells** are an internal compass, each tuned to fire when the animal faces one direction regardless of location. **Border cells** fire near walls and edges, anchoring the map to boundaries. Together the four cell types form a complete, flexible cognitive map.

Property	Place cells	Grid cells
Location	Hippocampus	Entorhinal cortex
Firing pattern	One unique place field	Repeating hexagonal lattice
New environment	Remap: a fresh map each time	Stable, a universal metric
Discovered	O'Keefe, 1971	Moser & Moser, 2005 (Nobel)
Role	"You are here" pins	Coordinate ruler
AI echo	Sparse codes, graph nodes	Metric / positional code

The Cognitive Map Is a Graph

Step back and a cognitive map is literally a **graph**: places are nodes, and the paths between them are edges whose strength reflects how easily you can travel between them. That is exactly the object a **graph neural network** is built to reason over. Ask “from the cafe, what is the shortest route to the park?” and a GNN propagates information across the graph to find it, mirroring how the brain plans trajectories through remembered space.



Places as nodes, routes as edges: the cognitive map is a graph a GNN can search.

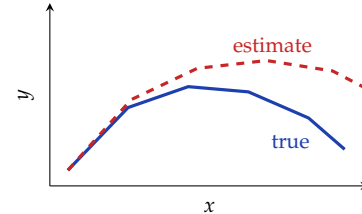
Path Integration and Its Drift

The brain also tracks position from self-motion alone, **path integration** (dead reckoning), summing velocity over time, with grid cells thought to be the engine:

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{v}(\tau) d\tau$$

The catch is that errors *accumulate*: integrate a slightly

wrong velocity long enough and the estimate drifts away from the truth. This is why an internal GPS alone is never enough.



Dead reckoning drifts: small velocity errors compound into a growing gap.

Landmarks as Attention

To cancel the drift, the brain resets against **landmarks**, a distinctive building, a river, a peak. But which cues matter? When you are lost you do not weigh every tree equally; you fix on the few informative ones. That selective weighting is exactly what an **attention** mechanism does in a Transformer, scoring which pieces of input deserve focus. An AI navigation agent can use the same trick to decide which sensory cues to trust when re-localizing.

ANALOGY

A lost hiker does not study every tree. She locks onto the ridge line and the river bend, two landmarks, and re-fixes her position from them. Attention in a Transformer is that same choice, made continuously: spend focus only where it pays.

Quick Self-Test

1. What does the dentate gyrus (pattern separation) do that CA3 (pattern completion) does not, and why do you need both?
2. When you enter a new room, place cells remap but grid cells stay stable. What computational role does each behavior serve?
3. Why is a hexagonal grid the brain’s choice for a coordinate system rather than a square one?
4. Explain the analogy “a cognitive map is a graph.” What are the nodes and edges, and what AI model consumes that structure?
5. Path integration drifts over time. What corrects it, and which Transformer component does that correction resemble?

If You Remember One Thing

The brain solves navigation by combining two representations, a stable universal ruler (grid cells) and a sparse map of meaningful places (place cells), and stabilizes the result with landmarks. Those are not just biological curiosities: the map-as-graph is a graph neural network, and landmark selection is attention, so the oldest navigation problem in nature keeps handing modern AI its blueprints.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

The Perception Pipeline

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 4 of "The NeuroAI Handbook"

How the brain turns photons into meaning, and how CNNs copied the blueprint.

One-sentence thesis. Vision is not a camera but a hierarchy: the retina, the visual cortex, and every convolutional network solve the same problem by stacking simple local detectors into progressively richer features.

PART 1: From Photons to Perception

The Problem Vision Has to Solve

A photograph is just a grid of brightness numbers. Nothing in that grid says "face" or "ball" or "curb ahead." Yet you recognize a friend across a crowded room in a fraction of a second. The gap between raw light and meaning is the entire problem of vision, and the brain does not close it by seeing harder. It closes it by *transforming* the signal, stage by stage, until an object falls out the far end.

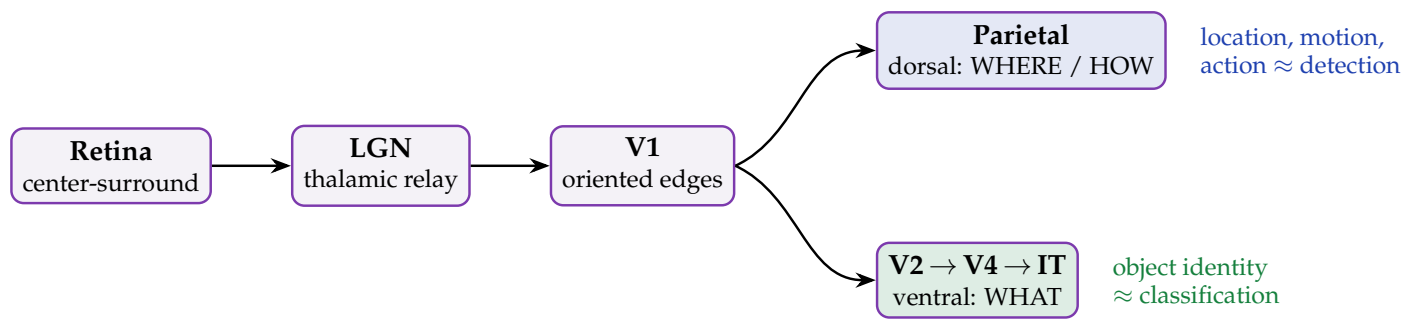
The pipeline starts before the brain. The **retina** is not a passive sensor; it is a pre-processor. Photoreceptors (rods and cones) capture light, but a network of downstream cells immediately enhances contrast and detects edges, so the signal that leaves the eye is already cleaned up. From there it passes to the **LGN** (lateral geniculate nucleus), a relay station in the thalamus that sorts and routes, and then to **V1**, the primary visual cortex, where the scene is first broken into oriented edges, lines, and motion.

Two Streams: What vs. Where

From V1 the pathway forks, and the split is one of the most important facts in vision science. The **ventral stream** ("what") runs down toward the temporal lobe through V2, V4, and IT cortex; its job is *object recognition*, and neurons along it respond to ever more complex shapes, from curves to whole faces. The **dorsal stream** ("where/how") runs up into the parietal lobe; its job is *spatial awareness and action*, tracking where things are and how to reach them. Goodale and Milner (1992) named this division of labor, and it maps almost exactly onto two AI tasks: the ventral stream is image *classification*, the dorsal stream is object *detection* and motor control.

ANALOGY

Picture the visual system as a corporation processing incoming mail. The retina is the mailroom (sorting, bundling raw letters into packages), the LGN is central dispatch, V1 is the junior analysts scanning for simple patterns, and higher cortex is the executives assembling the full picture. No single clerk sees the whole company. That is the secret: local processing plus hierarchical abstraction.



The pipeline: photons enter at the retina, get relayed through the LGN to V1, then fork into a dorsal "where" stream and a ventral "what" stream.

A Neuron's Window on the World

The single idea that unlocks both biological and artificial vision is the **receptive field**: the small patch of the visual world a given neuron actually responds to. A neuron does not see everything; it sees a keyhole, and it fires only when the right pattern lands in that keyhole.

The shape of the keyhole changes as you climb the pipeline. In the **retina**, receptive fields are small circles with a *center-surround* structure, excited by light in the center and inhibited around it, which makes them superb contrast and edge detectors. In **V1**, Hubel and Wiesel's Nobel Prize-winning experiments showed the keyholes are tuned to *orientation*: a "vertical" neuron fires vigorously at a vertical edge and ignores a horizontal one.

From Simple to Complex

V1 already builds a small hierarchy. **Simple cells** respond to an oriented edge at a *specific* location. **Complex cells** respond to that same edge *anywhere* in their receptive field; they achieve this by pooling the

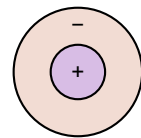
outputs of several simple cells, which buys **translation invariance**, the ability to recognize a feature regardless of exactly where it sits. **Hypercomplex (end-stopped) cells** go further, responding to corners, curves, and line endings.

The logic is compositional. Just as letters combine into words and words into sentences, the brain combines points of light into oriented lines, lines into corners, corners into parts, and parts into objects. **The point.** Complexity is never wired in directly; it is *built up* from a vocabulary of simple, reusable detectors.

A **Gabor filter**, a sine wave multiplied by a Gaussian envelope, is the standard mathematical model of a V1 simple cell. In symbols, the receptive field is a wave under a soft window:

$$g(x, y) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \cos\left(\frac{2\pi x'}{\lambda}\right)$$

Tune the orientation and you get a bank of edge detectors, one per angle, exactly the tiling of orientations found in V1.



retina: contrast



V1: orientation

Two receptive-field shapes. Left: a center-surround field detects contrast. Right: an oriented bar detector fires for a vertical edge and ignores others.

The Uncanny Parallel

Once you have the cortical hierarchy in mind, a **Convolutional Neural Network** stops being mysterious. A CNN is, in essence, a computational model of the ventral visual stream, and this was no accident: Fukushima's Neocognitron (1980) was explicitly modeled on simple and complex cells, LeCun and colleagues (1998) made it trainable end to end, and Krizhevsky's AlexNet (2012) scaled it up to win at object recognition.

Four correspondences carry almost the entire analogy. **Convolutional filters are receptive fields:** a kernel is a small window sliding over the image, looking for one pattern, just like a V1 neuron. **Weight sharing mirrors a universal detector:** the network applies the *same* filter everywhere, exactly as the brain reuses one "vertical edge" template across the whole visual field, and it is what makes the whole scheme efficient. **Stacked layers rebuild the hierarchy:** early layers

learn edges (like V1), middle layers learn textures and shapes (like V2 and V4), deep layers learn whole objects (like IT). **Pooling layers are complex cells:** a max-pool takes the strongest response in a small region, keeping *what* is present while discarding *where*, which is precisely how complex cells manufacture translation invariance.

What the Filter Computes

Formally, a convolution slides a small kernel K over the input and sums the overlap at every position, producing a feature map that lights up wherever the pattern occurs:

$$(I * K)_{x,y} = \sum_{m,n} I_{x+m, y+n} K_{m,n}$$

One kernel, swept across the image, is one detector deployed everywhere at once, the engineering echo of weight sharing in cortex.

Visual cortex	CNN component	What it computes
Retina (center-surround)	Input + edge preprocessing	Contrast, local edges
V1 simple cells	Convolutional filter (kernel)	Oriented edges at a position
V1/V2 complex cells	Max-pooling layer	Translation invariance
V2 / V4	Mid convolutional layers	Textures, corners, parts
IT cortex	Deep layers + classifier head	Whole objects, categories
One template, whole field	Weight sharing (shared kernel)	Efficiency: one detector, everywhere

Where the Analogy Strains

The clean story above is feedforward: signals march from retina to IT and out pops a label. Real brains are not like that. They are saturated with *feedback*: connections running backward down the hierarchy vastly outnumber the forward ones, and they carry context, expectation, and attention. Early CNNs threw all of that away, which is exactly why the frontier of AI vision now runs toward recurrence and attention.

Vision Transformers (ViTs). Rather than sliding local filters, a ViT (Dosovitskiy et al., 2021) cuts the image into patches and uses *self-attention* to weigh how every patch relates to every other. That captures global context and long-range relationships, closer to the brain's top-down, whole-scene understanding than a stack of purely local convolutions.

Predictive coding and generative models. A leading theory holds the brain is constantly *predicting* its sen-

sory input and only propagating the error. Generative models (GANs, diffusion) that learn to *create* images rather than merely label them are the AI reflection of this idea.

A Two-Way Street

The relationship is a genuine dialogue, not one-way inspiration. **Biology gave the blueprint:** hierarchical layers, local receptive fields, and the what/where split were discovered by neuroscientists first. **AI returns a model:** when a CNN trained only on images spontaneously grows V1-like edge detectors, it is evidence our theory of the brain is on the right track. Brain-Score work (Schrimpf et al., 2018; Yamins and DiCarlo, 2016; Cadieu et al., 2014) found deep networks can *rival* primate IT cortex at predicting neural responses. Faced with the same visual world, brains and machines converged on strikingly similar solutions, which hints these principles may be universal.

Quick Self-Test

1. Trace a signal from photon to object label, naming the four main stages and what each contributes.
2. What is a receptive field, and how does its shape differ between the retina and V1?
3. How do complex cells achieve translation invariance, and which CNN component copies this trick?
4. Weight sharing in a CNN corresponds to what property of biological vision, and why does it matter for efficiency?
5. Name one way real vision departs from a feedforward CNN, and one AI architecture that tries to close the gap.

If You Remember One Thing

Both the visual cortex and a CNN win by refusing to solve vision all at once. They stack a vocabulary of simple, reusable, local detectors into a hierarchy, edges to textures to parts to objects, and pool along the way to stop caring about exact position. The architecture is the insight; everything else is detail.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Brain Networks

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 5 of "The NeuroAI Handbook"

The Default Mode and Executive Control networks, their anti-correlated dance, and its echo in exploration vs. exploitation.

One-sentence thesis. Human cognition runs on two anti-correlated networks, an introspective and generative Default Mode Network and a focused, discriminative Executive Control Network, switched by a Salience Network, and their trade-off is the brain's version of exploration vs. exploitation in machine learning.

PART 1: Mapping the Brain as a Network

The Brain Is a Graph, Not a Box of Parts

For a century, neuroscience asked what each brain region *does*, one box at a time. That framing misses the point. The brain is a massively interconnected system in which regions are constantly talking to one another, and much of cognition lives in the *pattern of communication*, not in any single area. **Network neuroscience** studies that pattern with the mathematics of graphs.

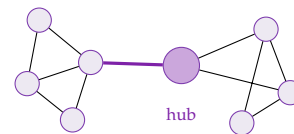
The vocabulary is simple. **Nodes** are brain regions. **Edges** are the connections between them, either *structural* (physical white-matter tracts) or *functional* (regions whose activity rises and falls together). Using **functional MRI**, which tracks blood flow as a proxy for neural activity, researchers can see which regions co-activate and reconstruct the brain's intrinsic networks.

The Accidental Discovery

The whole field pivoted on a surprise. In the late 1990s, researchers noticed that a particular set of regions became *more* active when subjects were asked to do nothing at all, and quieted down the moment a task began. Raichle and colleagues (2001) named it "a default mode of brain function." Rest, it turned out, is not the brain idling; it is the brain running a different program.

Small-World Wiring and Hubs

Real brains are wired in a **small-world** pattern: dense local clusters linked by a few long-range bridges. That combination buys the best of both worlds, high local clustering for efficient specialized processing plus short path lengths for fast global integration. A handful of richly connected **hubs** (the posterior cingulate cortex, anterior insula, and dorsolateral prefrontal cortex) act as integrators. **The catch.** Hubs are load-bearing: lesion one and the damage ripples network-wide, which is why hub injury shows up across so many neurological disorders.



Small-world wiring: two densely connected local clusters joined by a single long-range bridge through a hub. Cheap to traverse, but the hub is a single point of failure.

ANALOGY

A small-world brain is wired like an airline map. Most flights are short hops within a region, but a few hub airports connect everything to everything with just one or two layovers. It is efficient, until a hub shuts down, and then delays cascade across the whole system.

The Inner Dreamer (DMN)

Think of the last time you drove a familiar route and arrived with no memory of the trip, or the moment a solution surfaced in the shower. That is the **Default Mode Network** at work, the brain's inward-facing system. Its jobs are self-referential thought, autobiographical memory, simulating the future, and recombining old ideas into new ones. Its core nodes are the medial prefrontal cortex, posterior cingulate, and angular gyrus.

The AI parallel is a **generative model**. When a large language model writes a story, it samples from what it has learned to build a novel but coherent output; when a diffusion model dreams up an image, it wanders a learned *latent space* of possibilities. The DMN does much the same, exploring a latent space of personal experience to construct scenes that never happened.

The Focused Doer (ECN)

Now recall locking onto a hard problem, shutting out the room, holding several facts in mind at once.

That is the **Executive Control Network**, the brain's outward-facing taskmaster. Its jobs are working memory, cognitive control (overriding a habit to do something harder), task switching, and focused attention. Its core nodes are the dorsolateral prefrontal cortex and intraparietal sulcus.

The AI parallel is a **discriminative model**. An image classifier deciding "cat or dog" must suppress irrelevant detail and lock onto the features that settle the question, exactly the ECN's job. A reinforcement-learning agent executing a learned policy toward a goal fits the same mold: hold the objective, monitor progress, adjust.

ANALOGY

If the ECN is the CEO executing this quarter's plan, the DMN is the long-term strategy and R&D department: reviewing past performance (memory), simulating future scenarios (planning), and brainstorming what to build next (creativity). The CEO then chooses what to act on.

	Default Mode Network	Executive Control Network
Faces	Inward (self, memory, future)	Outward (task, world)
Core jobs	Daydreaming, recall, simulation, creativity	Working memory, control, focus
Key nodes	mPFC, PCC, angular gyrus	dIPFC, intraparietal sulcus
AI analog	Generative model (LLM, diffusion)	Discriminative model, RL policy
RL mode	Exploration (high variance)	Exploitation (low variance)

A Competitive Partnership

You cannot daydream and do calculus in the same instant, and your brain enforces that. The DMN and ECN are largely **anti-correlated**: when one rises, the other falls (Fox et al., 2005), with correlations that typically run below -0.5 . Both fluctuate slowly, on the order of a few hundredths of a hertz, trading dominance back and forth throughout the day.

The Saliency Switch

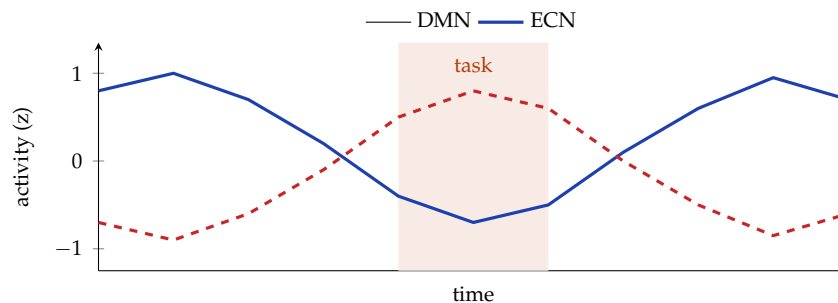
Who decides when to switch? A third system, the **Saliency Network** (anchored in the anterior insula; Seeley et al., 2007), acts as a traffic cop. It monitors the internal and external world for important, or *salient*, events and, when something matters, hands control from the introspective DMN to the task-focused ECN.

Exploration vs. Exploitation

This dynamic is a biological rendering of the **exploration-exploitation trade-off**, a foundational idea in machine learning. **Exploration (DMN)**: wander the memory space, make novel connections, generate ideas, high variance, creative. **Exploitation (ECN)**: deploy a known strategy to hit a goal, low variance, focused. A healthy mind, like a good RL agent, flexibly switches: exploration finds better strategies, exploitation cashes them in.

When the Dance Breaks

Fail to switch cleanly and the consequences are clinical. **Depression** is marked by an overactive DMN that will not let go, driving rumination and negative self-referential loops. **ADHD** shows the opposite failure: an unreliable ECN with intrusions from the DMN, so focus on the external task keeps slipping. Seen this way, several disorders are not broken parts but a broken *balance*.



DMN and ECN activity are mirror images: when a task arrives (shaded), the ECN surges and the DMN is suppressed. Their anti-correlation is the switching signature of a flexible mind.

Quick Self-Test

1. In network terms, what are nodes and edges, and what does functional MRI actually measure?
2. What was surprising about the discovery of the Default Mode Network, and who named it?
3. Map the DMN and ECN onto the generative-vs-discriminative distinction in AI, with one example each.
4. What role does the Saliency Network play, and how does the DMN/ECN switch relate to exploration vs. exploitation?
5. Describe how the DMN/ECN balance is disrupted in either depression or ADHD.

If You Remember One Thing

Cognition is not a place but a balance. The introspective, generative DMN and the focused, discriminative ECN take turns, refereed by the Saliency Network, and this is the brain's own exploration-exploitation trade-off.

Health is the flexibility to switch; many disorders are a switch stuck in one position.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Neurostimulation

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 6 of "The NeuroAI Handbook"

How the brain rewires itself, how neuromodulators gate learning, and how both echo the training of AI.

One-sentence thesis. The brain learns by rewiring itself, Hebbian strengthening held in check by homeostatic control and gated by neuromodulators, and both the rules (STDP, dopamine as reward-prediction error) and the interventions (tDCS, TMS, DBS) map onto how we train and fine-tune artificial networks.

PART 1: How Synapses Learn

Cells That Fire Together, Wire Together

Learning has to change something physical. In the brain, what changes is the *strength* of the synapse between two neurons. Donald Hebb's 1949 postulate stated the core idea: when neuron A repeatedly helps fire neuron B, the connection from A to B grows stronger. The slogan is "cells that fire together, wire together," and it remains the foundation of plasticity theory.

In symbols, the weight change is just the product of pre- and postsynaptic activity, scaled by a learning rate:

$$\Delta w_{ij} = \eta x_i y_j$$

The instability. Pure Hebbian learning is a positive-feedback loop: strong synapses keep strengthening without limit, driving runaway excitation. Something must push back, which is why the brain pairs Hebbian change with homeostatic control.

Strengthen and Weaken: LTP and LTD

Hebb's rule is implemented at the cellular level by two opposing processes. **Long-Term Potentiation (LTP)** strengthens synapses: high-frequency stimulation (around 100 Hz) opens **NMDA receptors**, which act as *coincidence detectors* because they require both presynaptic glutamate and postsynaptic depolarization at once. Calcium floods in, activates CaMKII, and more AMPA receptors are inserted (first shown in hippocampal slices by Bliss and Lomo, 1973). **Long-Term Depression (LTD)** does the reverse: low-frequency

stimulation (around 1 Hz) lets in only moderate calcium, activating phosphatases that remove AMPA receptors. These changes last hours to months, the cellular basis of memory.

Timing Is Everything: STDP

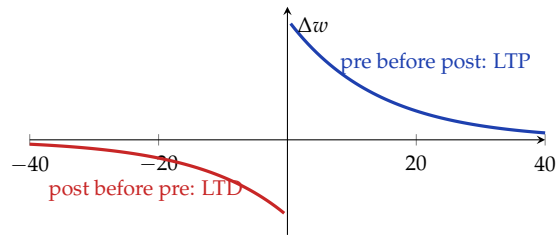
Hebb said nothing about *order*. **Spike-Timing-Dependent Plasticity** adds it: if the presynaptic neuron fires shortly *before* the postsynaptic one (within about 20 ms), the synapse strengthens; if it fires *after*, the synapse weakens; and the effect shrinks exponentially as the gap widens (Bi and Poo, 1998). This lets a network learn cause from effect, since the input that reliably precedes an output gets the credit.

Keeping It Stable: Homeostatic Plasticity

If Hebbian learning is the accelerator, **homeostatic plasticity** is the thermostat. **Synaptic scaling** multiplies all of a neuron's synapses toward a target activity level, preserving their relative sizes while capping the total (slow, over days). **Metaplasticity** (the BCM sliding threshold) raises the bar for potentiation after a burst of activity. **Intrinsic plasticity** tunes a neuron's own excitability. Together they prevent both silence and seizure while leaving room to learn.

ANALOGY

Hebbian and homeostatic plasticity are the accelerator and the cruise control. Hebbian rules speed up whatever synapse is being used; homeostatic scaling keeps the whole network from redlining. Learning needs both, or it either flat-lines or explodes.



$$\Delta t = t_{\text{post}} - t_{\text{pre}} \text{ (ms)}$$

The STDP window. Fire just before the target and the synapse strengthens; fire just after and it weakens. The magnitude decays exponentially with the timing gap.

The Brain's Global Settings

Local plasticity decides *which* synapses change. But something has to decide *when* learning should happen and *how strongly*, and that is the job of **neuromodulators**, chemical signals broadcast widely rather than sent point to point. They are the brain's global control panel for learning, and each one has a clean computational counterpart in AI.

Dopamine, the reward signal. Released from the VTA and substantia nigra, phasic dopamine encodes a *reward prediction error*: the gap between the reward you got and the reward you expected. An unexpected reward gives a burst; an omitted one gives a dip. This is the biological twin of the temporal-difference error in reinforcement learning.

Acetylcholine, the attention knob. From the basal forebrain, it raises the signal-to-noise ratio, sharpen-

ing processing of relevant stimuli and enabling memory encoding, much like an attention mechanism or dropout in a neural network.

Norepinephrine, the alertness dial. From the locus coeruleus, it sets arousal and flags novelty, nudging the system to explore. In AI terms it tunes the exploration-exploitation balance and the effective learning rate.

Serotonin, patience. From the raphe nuclei, it shapes mood, impulse control, and how heavily the future is discounted, the biological echo of the discount factor in RL.

They interact. These systems are not independent dials; dopamine-acetylcholine and norepinephrine-serotonin balances jointly set the brain's learning state.

Neuromodulator	Source	Brain role	AI analog
Dopamine	VTA, substantia nigra	Reward prediction error	TD error , actor-critic
Acetylcholine	Basal forebrain	Attention, signal-to-noise	Attention, dropout
Norepinephrine	Locus coeruleus	Arousal, novelty	Exploration, learning rate
Serotonin	Raphe nuclei	Mood, patience, impulse control	Temporal discounting

Nudging the Circuit

If plasticity is the brain's learning dial, neurostimulation reaches in and turns it from outside. The techniques span a spectrum from cheap and non-invasive to surgical and precise, and the trade-off is always the same: reach and focality versus safety and accessibility.

tDCS (transcranial direct current stimulation) passes a weak current (1 to 2 mA) between scalp electrodes. *Anodal* stimulation gently depolarizes neurons and raises excitability; *cathodal* does the reverse. It does not fire neurons directly, but its LTP-like after-effects can outlast the session. Cheap and portable, but coarse and highly variable between people.

TMS (transcranial magnetic stimulation) uses a brief 1 to 2 tesla magnetic pulse to induce real currents that *can* trigger action potentials. Repetitive TMS at low

frequency (at or below 1 Hz) tends to inhibit; high frequency (5 Hz and up) tends to excite. It reaches about 1 to 2 cm with good timing precision and is FDA-approved for treatment-resistant depression.

DBS (deep brain stimulation) implants electrodes that deliver high-frequency pulses (130 to 180 Hz) directly into deep nuclei. It is dramatically effective for Parkinson's disease, essential tremor, and dystonia, and is approved for OCD, but it is a neurosurgical procedure with real risk.

Also worth knowing. tACS entrains brain oscillations at chosen frequencies; focused ultrasound (tFUS) reaches deep targets non-invasively with millimeter precision; VNS is approved for epilepsy and depression; and optogenetics uses light-sensitive opsins for millisecond, cell-type-specific control, still a research tool in animals.

Technique	How it works	Invasiveness	Focality	Status
tDCS	Weak DC 1–2 mA, anodal/-cathodal	Non-invasive	Coarse (cm)	Research; variable
TMS	Magnetic 1–2 T pulses	Non-invasive	1–2 cm	FDA depression
DBS	Implanted electrodes, 130–180 Hz	Surgical	Precise, deep	Parkinson's, OCD

Dopamine Is Temporal-Difference Learning

The crown jewel of NeuroAI is a near-perfect match: the phasic firing of dopamine neurons looks exactly like the *temporal-difference error* that drives reinforcement learning. The TD error compares what actually happened to what was predicted:

$$\delta = r + \gamma V(s') - V(s)$$

The correspondence runs deep. An **actor-critic** architecture maps onto the basal ganglia: a *critic* (ventral striatum) learns state values and computes the error, while an *actor* (dorsal striatum) picks actions, and dopamine broadcasts δ to both. Extend Hebb's rule with this signal and you get a **three-factor learning rule**, $\Delta w = \eta x_i y_j \delta$, where an *eligibility trace* tags recently active synapses so a delayed reward can still find them, solving the credit-assignment problem.

Learning Without Backprop, and Fine-Tuning

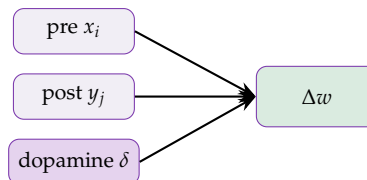
STDP is a *local* rule: it updates a weight using only signals available at that synapse, no global error to

propagate backward. That makes it attractive for neuromorphic hardware such as SpiNNaker and BrainScaleS, and homeostatic scaling offers a handle on catastrophic forgetting. Neuromodulation itself reads like meta-learning: cyclical and decaying learning-rate schedules, and adaptive optimizers like Adam, are engineered stand-ins for the brain's shifting learning states.

The tightest practical parallel is **fine-tuning**. A pre-trained model is like a developed brain; adapting it to a new task by nudging a few parameters, through adapters or **LoRA** (low-rank adaptation), is the AI version of neurostimulation enhancing an existing circuit rather than growing one from scratch.

ANALOGY

Fine-tuning a large model with LoRA is renovating a house, not rebuilding it. The structure (pre-trained weights) stays; you make small, targeted changes to a few rooms. Neurostimulation does the same to the brain, turning up plasticity in one circuit while leaving the rest intact.



The three-factor rule: Hebbian coincidence (pre times post) is gated by a global neuromodulatory signal (dopamine as reward-prediction error) before any weight changes.

Quick Self-Test

1. Why is pure Hebbian learning unstable, and what kind of mechanism counteracts it?
2. How does STDP let a network distinguish cause from effect?
3. What does phasic dopamine encode, and which reinforcement-learning quantity does it match?
4. Rank tDCS, TMS, and DBS by invasiveness and by spatial precision, and give one clinical use of each where applicable.
5. In what sense is fine-tuning a pre-trained model with LoRA analogous to neurostimulation?

If You Remember One Thing

The brain's learning is a balance of forces: Hebbian strengthening, homeostatic stabilizing, and neuromodulators gating when it all counts. The single deepest bridge to AI is that dopamine behaves like a temporal-difference

error, which means reinforcement learning is not just inspired by the brain, it may be describing what the brain is actually doing.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Information Theory for NeuroAI

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 7 of "The NeuroAI Handbook"

The one currency, the bit, that lets you measure a neuron and a network unit on the same scale.

One-sentence thesis. Information theory turns three fuzzy questions, how surprising, how related, how wrong, into a single unit, the bit, that measures a spiking neuron and a deep network by the same ruler.

PART 1: Surprise, Measured

How Surprised Should You Be?

Start with the feeling, not the formula. "The sun rose this morning" tells you nothing, you already knew it. "I won the lottery" tells you a great deal, because it was so unlikely. Claude Shannon's 1948 insight was that these are the same quantity viewed at different scales: information is the resolution of uncertainty, and a message carries information exactly to the degree it was not expected. Rare events are informative; certain events are not.

Entropy, $H(X)$, is the *average* surprise of a source, how uncertain you are before the outcome arrives. It is measured in **bits**.

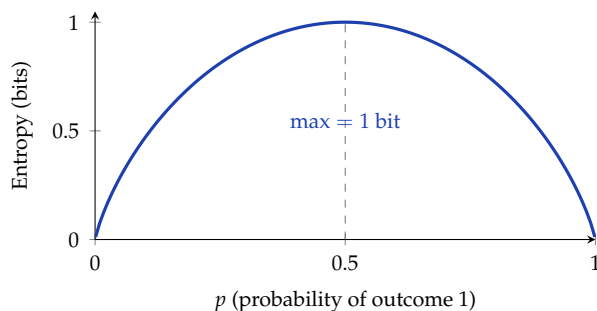
ANALOGY

Entropy is the minimum average number of yes/no questions in a game of 20 Questions. A fair coin needs **1** ("heads?"); a fair 8-sided die needs **3**; an English letter about **4.2**. The fewer questions, the less the source surprises you.

In symbols, sum each outcome's probability against its surprise $\log_2 \frac{1}{p}$:

$$H(X) = - \sum_i p(x_i) \log_2 p(x_i)$$

Two facts do all the work. Entropy is largest when every outcome is equally likely (nothing to predict), and it collapses to zero when one outcome is certain (no surprise possible). A biased coin ($p = 0.9$) has only 0.47 bits; a fair coin has the full 1.



Binary entropy peaks at $p = 0.5$ (maximum uncertainty) and vanishes at $p = 0$ or 1 (certainty). This is why compression works: predictable data carries less than its raw bit-size suggests.

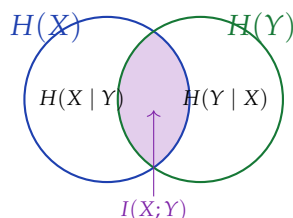
How Much Does One Signal Tell You About Another?

You record a neuron in visual cortex and one in motor cortex. Watching one, how much of your uncertainty about the other disappears? That reduction is **mutual information**, $I(X; Y)$: the knowledge two variables share. Zero means independence, learning X tells you nothing about Y . It is symmetric, and unlike a correlation coefficient it catches nonlinear relationships that a straight-line fit would miss entirely.

The cleanest way to see it is as overlap. Draw one circle for everything X could tell you, $H(X)$, and one for $H(Y)$; their intersection is the shared part.

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

Equivalently, $I(X; Y) = H(X) - H(X | Y)$: your starting uncertainty minus what remains after seeing Y .



Mutual information is the overlap of two entropies. The non-overlapping crescents are conditional entropies, what each variable knows that the other does not.

The Cost of Using the Wrong Map

Now suppose reality follows distribution P but you plan using an approximation Q . How much do you pay for the mismatch? **Kullback-Leibler divergence**, $D_{KL}(P \parallel Q)$, is that penalty in bits, the extra surprise you suffer by encoding P 's events with a code built for Q .

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log_2 \frac{P(i)}{Q(i)}$$

It is zero only when $P = Q$, never negative, and,

crucially, **not symmetric**: the cost of a tourist using a local's map differs from a local using the tourist's. **Watch out.** This asymmetry is a modeling choice, not a nuisance, $D_{KL}(P \parallel Q)$ punishes you hard for assigning low probability to events that actually happen.

This is not abstract. **Cross-entropy loss**, used to train nearly every classifier, is KL divergence in disguise: training a network is minimizing the divergence between its predicted distribution and the data's true one. VAEs regularize with a KL term; knowledge distillation transfers a teacher's distribution to a student the same way.

ANALOGY

KL divergence is the tax on a bad weather forecast. If you believe rain comes 10% of the time (Q) but it truly comes 50% (P), you are perpetually caught without an umbrella. The divergence counts, in bits, exactly how much that wrong model costs you on average.

Why the Brain Cares About Bits

A neuron is metabolically expensive; spikes cost energy the brain cannot waste. Horace Barlow's **efficient coding hypothesis** (1961) proposes that sensory systems evolved to represent the natural world at the lowest possible cost, spending bits only where they buy prediction. Two imperatives follow: **reduce redundancy** (do not re-encode what is predictable) and **maximize information** given a fixed bandwidth.

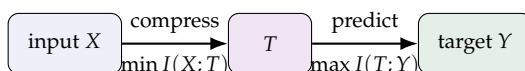
The brain has several tricks for this. **Predictive coding** transmits only the *prediction error*, the part of the input a higher area failed to anticipate, so a fully predicted stimulus travels forward as near silence. **Sparse coding** keeps only a small fraction of neurons active at once, saving energy and easing pattern separation; it is the biological cousin of L1 regularization and

dropout. **Decorrelation** spreads different features across different neurons so their activities stop echoing each other.

There is a hard ceiling on all of this. **Channel capacity** is the maximum reliable rate through a noisy channel, which for a Gaussian channel with signal power P and noise N is:

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P}{N} \right)$$

A single neuron carries only a couple of bits per spike, so the brain beats the ceiling by pooling many noisy neurons into a population code, the same reason the optic nerve's million axons demand ruthless compression upstream.



The information bottleneck (Tishby): squeeze X into a compact code T that throws away everything except what predicts Y . Squeeze too little and you memorize noise; too much and you lose the signal.

PART 4: Information as the Bridge to AI

Same Laws, Silicon or Cells

The point of a common currency is that it lets you put a neuron and a network unit on one balance sheet. Ask a precise question of a spike train, how much does the firing pattern tell you about the stimulus, and it is just mutual information between stimulus s and response r :

$$I(S; R) = \sum_{s,r} p(s, r) \log_2 \frac{p(s, r)}{p(s) p(r)}$$

The same quantity drives modern self-supervised AI. **Contrastive learning** (SimCLR, InfoNCE) trains a model to *maximize* mutual information between two augmented views of one image, forcing it to keep the

invariant essence and discard the incidental, precisely Barlow’s redundancy reduction restated.

Naftali Tishby’s **information bottleneck** frames a deep network as a stack of the squeezes in the PART 3 figure: each layer compresses its input while preserving what the final prediction needs. **Transfer entropy** goes further, measuring *directed* flow, does the past of X help predict the future of Y beyond Y ’s own past, which is how you probe causal information routing in both brain circuits and network layers.

The throughline. Entropy measures surprise, mutual information measures shared structure, KL divergence measures model error, and every one of them applies unchanged whether the substrate is a cortical column or a transformer block. That is the bridge.

Measure	Question it answers	In the brain	In AI
Entropy $H(X)$	How surprising is this source?	Spike-count variability, coding budget	Prediction confidence, compression
Mutual info $I(X; Y)$	How much does X tell about Y ?	Stimulus info in spike trains	Contrastive / self-supervised objectives
KL $D_{KL}(P \ Q)$	How wrong is my model?	Belief update, prior to posterior	Cross-entropy loss, VAE regularizer
Channel capacity	What is the ceiling?	Bits per spike, optic-nerve limit	Layer width, bottleneck design

Quick Self-Test

1. Why does a biased coin ($p = 0.9$) carry less than one bit, while a fair coin carries exactly one?
2. Two variables have a mutual information of zero. What does that tell you, and why is it a stronger statement than “their correlation is zero”?
3. Explain in one sentence why $D_{KL}(P \| Q) \neq D_{KL}(Q \| P)$, and why training a classifier minimizes one of them.
4. What two things does the efficient coding hypothesis say a sensory neuron should do, and how do predictive and sparse coding each achieve one of them?
5. State the information bottleneck objective in words, and name one AI method and one brain phenomenon it describes.

If You Remember One Thing

Entropy (surprise), mutual information (shared knowledge), and KL divergence (the cost of a wrong model) are three questions in one unit, the bit. Because that unit does not care whether the signal comes from a neuron or a network, information theory is the common language in which brains and machines can finally be compared.

The Data Science Pipeline

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 8 of "The NeuroAI Handbook"

The five-stage workflow that turns raw, noisy data into insight, whether the data comes from a brain or a model.

One-sentence thesis. Decoding a brain and training a model are the same five-stage pipeline, and the projects that succeed win on the unglamorous stages, cleaning and validation, not on the choice of model.

PART 1: One Pipeline, Two Fields

The Same Journey From Data to Knowledge

A neuroscientist staring at electrode voltages and an engineer staring at a training set face one problem: raw data is messy, high-dimensional, and mute. Turning it into insight follows a fixed route, the **data science pipeline**, and the route is identical in both fields:

1. **Acquire and preprocess**, get and clean the raw data.
2. **Extract features**, recast it into an informative form.
3. **Explore and reduce**, visualize and find structure.
4. **Model and decode**, fit something that predicts.
5. **Validate and interpret**, prove it generalizes.

The temptation is to sprint to stage 4, the modeling, because it feels like the science. **Watch out.** The failures almost always trace to stages 1 and 5. No

architecture rescues a project built on dirty data or judged without honest validation, the oldest rule in the field is *garbage in, garbage out*.

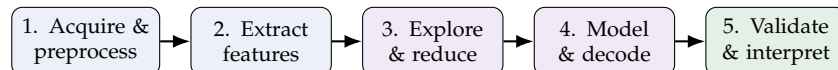
Scale, and the Brain's Answer to It

Before the pipeline runs there is a practical wall: **scale**. Terabytes of imaging, petabytes of text, none of it fits on one processor. The universal fix is **distributed processing**, split the problem, solve the pieces in parallel, recombine.

The nervous system is the original distributed computer. Catching a ball, your visual cortex tracks the trajectory while proprioceptors track your arm, all at once. The motor cortex does not micromanage muscle fibers; it hands high-level commands to the spinal cord, which runs the low-level coordination locally, a master node dispatching to workers.

ANALOGY

A spinal reflex is edge computing. When the ball hits your palm, grip adjusts in milliseconds, far faster than a round trip to the brain, because the spinal cord caches the subroutine near the data, exactly as a distributed system caches hot computations close to where they are needed.

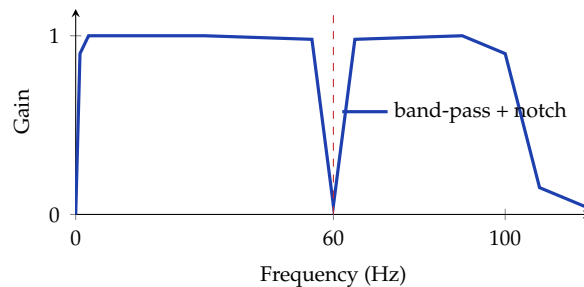


The universal pipeline. A neuroscientist decoding intention and an engineer classifying images walk the same five stages; only the data differs.

Stage 1: Cleaning the Signal

All data arrives contaminated. Neural recordings carry 60 Hz line noise, slow drift, and biological artifacts like eye blinks. The neuroscientist reaches for **filters**: a **band-pass** filter keeps a useful range (say 1-100 Hz) and discards the rest; a **notch** filter surgically removes one frequency, the 60 Hz hum.

The AI engineer's stage 1 is the same act under different names, impute missing values, correct bad entries, and **normalize** features (a z-score) so no single feature dominates by scale alone, which is what keeps gradient-based training stable. Filtering a voltage trace and standardizing a feature column are the same idea: strip the nuisance, keep the signal.



Preprocessing as a gain curve: pass the 1-100 Hz neural band, but carve a deep notch at the 60 Hz line-noise spike. Normalizing an AI feature matrix does the analogous job, remove the nuisance, preserve the signal.

Stage 2: From Raw Values to Informative Features

Raw data is rarely in its most telling form. **Feature extraction** builds better variables from the ones you have. A raw EEG trace is just voltage over time; far more useful is the *power in each frequency band*, delta, theta, alpha, beta, gamma, because those bands track cognitive states.

The engineer calls this **feature engineering**. For text, instead of raw words you extract **TF-IDF** scores (how distinctive a word is to a document) or **word embeddings** like Word2Vec (a dense vector for meaning). In both fields the move is identical: transform the data so the pattern you care about becomes visible to whatever comes next.

Stage 3: Making High Dimensions Visible

Record 100 neurons and your data lives in a 100-dimensional space no human can picture. **Dimensionality reduction** projects it down to 2D or 3D you can actually inspect. **Principal Component Analysis (PCA)** finds the axes of greatest variance; plotting population activity along them reveals a **neural trajectory**, the path the population traces through a task.

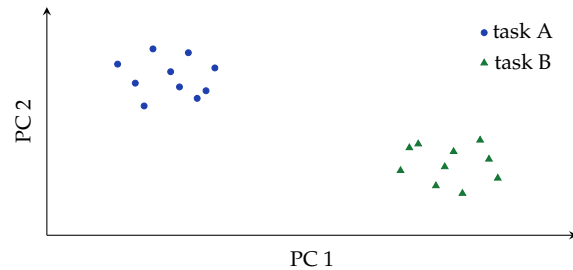
The engineer does the same to understand a language model, projecting high-dimensional word embeddings with **t-SNE** or **UMAP** to see clusters of related words emerge. PCA is linear and fast and preserves global variance; t-SNE and UMAP are nonlinear and better at exposing local clusters, at the cost of distorting global distances. Reach for PCA first, for the

nonlinear methods when structure hides in curved manifolds.

Stage 4: Fitting a Model That Predicts

Now build something predictive. In neuroscience this is **decoding**: can we read out what a subject sees or intends from brain activity? A neuroscientist might train a **support vector machine** or **random forest** to predict a left- versus right-hand button press from motor-cortex EEG.

The engineer calls this supervised learning, fit a model to map inputs to labels (“cat” or “dog”). The underlying act, adjusting parameters so predictions match known answers, is one and the same; the label is a stimulus in one field and a category in the other.



PCA on 50-neuron population activity: two tasks that were tangled in 50 dimensions separate cleanly in the top two principal components, the same trick that clusters word embeddings for an LLM.

Stage 5: The Only Score That Counts Is on Unseen Data

A model that shines on its training data has proven nothing, it may have memorized noise. The question that matters is **generalization**: does it perform on data it has never seen? **Validation** is how you find out.

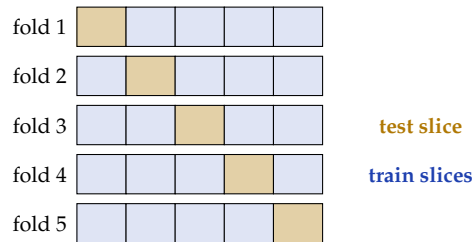
The neuroscientist uses **cross-validation**: train on most of the data, test on a held-out slice, then rotate

which slice is held out and average. This wrings a reliable performance estimate out of a limited dataset instead of trusting one lucky split.

The engineer formalizes this as a three-way split, a **training set** to fit parameters, a **validation set** to tune hyperparameters, and a **test set** touched exactly once, at the very end, for an unbiased verdict. The discipline is non-negotiable: the moment you let the test set influence a decision, its score stops meaning anything.

ANALOGY

Peeking at the test set is grading yourself on questions you already saw. You will ace the exam and learn nothing about whether you understand the material. Cross-validation is a proctor who keeps handing you fresh questions you have never studied.



Five-fold cross-validation: each row holds out a different slice for testing (amber) and trains on the rest (blue), then averages. Every data point is tested exactly once, and none is ever tested by a model that trained on it.

Design principle	In distributed computing	In the nervous system
Data locality	Process data where it is stored	Reflexes handled in the spinal cord
Fault tolerance	Replicate across nodes	Redundant neural pathways
Hierarchy	Edge → regional → central	Spinal → subcortical → cortical
Minimal communication	Reduce network traffic	Sparse, efficient neural coding
Parallelism	Many CPUs at once (MapReduce)	Billions of neurons fire in parallel

Quick Self-Test

1. Name the five pipeline stages in order, and explain why skipping careful work in stage 1 or stage 5 usually sinks a project.
2. A notch filter at 60 Hz and a z-score normalization serve the same purpose in different fields. What is that shared purpose?
3. When would you prefer PCA over t-SNE for exploring high-dimensional neural data, and when the reverse?
4. Why is k-fold cross-validation more trustworthy than a single train/test split?
5. What exactly goes wrong when you use the test set to choose a hyperparameter?

If You Remember One Thing

The modeling stage gets the glory, but the pipeline is won or lost on cleaning and validation. Dirty inputs and dishonest evaluation break more neuroscience decoders and AI systems than any wrong choice of classifier, because a model can only ever be as good as the data it learned from and the test that judged it.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Causal Inference

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 9 of "The NeuroAI Handbook"

Telling "this caused that" from "these just happened together," in the brain and in the machine.

One-sentence thesis. Correlation is what you see; causation is what happens when you intervene, and everything in this chapter is machinery for getting the second when you can only measure the first.

PART 1: Seeing Is Not Doing

Did This Cause That, or Did They Just Co-occur?

A neuron in visual cortex and one in motor cortex fire together. Did the first drive the second? You cannot tell from watching, because watching only reveals that they *co-occur*. The one clean way to know is to **intervene**: force the visual neuron off and see whether the motor neuron changes. That is the interventional definition of cause, "A causes B" means that making A different makes B different, and it is why the **randomized controlled trial** is the gold standard: randomizing who gets the treatment severs every hidden influence on who gets it.

Judea Pearl organized this into a three-rung **ladder of causation**, and the rungs are not interchangeable:

$$\underbrace{P(Y | X)}_{\text{see}} \neq \underbrace{P(Y | do(X))}_{\text{do}}$$

Rung one, **association**, is $P(Y | X)$, passively observing X . Rung two, **intervention**, is $P(Y | do(X))$, actively *setting* X and cutting whatever normally causes it. Rung three, **counterfactual**, asks what would have happened to *this* subject had X been different. No amount of rung-one data answers a rung-two question on its own, you need an intervention or a causal model. The whole discipline lives in that gap between seeing and doing.

The Three Ways a Correlation Is Born

Two variables can move together for reasons that have nothing to do with one causing the other. Every spurious correlation traces to one of three graph shapes. In a **chain** ($X \rightarrow Z \rightarrow Y$), X acts through a mediator. In a **fork** ($X \leftarrow Z \rightarrow Y$), a **confounder** Z drives both, and X and Y dance together while neither touches the other. In a **collider** ($X \rightarrow Z \leftarrow Y$), two independent causes feed a common effect.

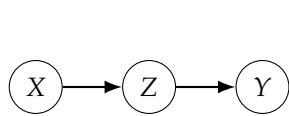
The confounder is the classic trap. Two neurons cor-

relate not because one drives the other but because a presented stimulus drives both, the stimulus is Z . Control for the right variables and the illusion dissolves; miss one and your estimate stays biased no matter how much data you pile up.

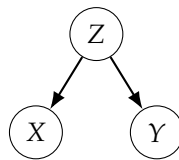
Watch out, colliders bite the other way. Conditioning on a chain or fork *removes* a spurious link, but conditioning on a collider *creates* one. Filter your data by a common effect and you manufacture a correlation between its causes that was never there.

ANALOGY

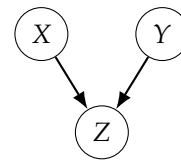
Roosters crow just before sunrise, every day, a near-perfect correlation. Silence every rooster and the sun still rises: the earth's rotation is the confounder driving both. Correlation this strong, and still zero causation.



Chain (mediator)



Fork (confounder)



Collider (common effect)

The three elementary structures. Conditioning on Z blocks the chain and the fork (good, it removes spurious flow) but opens the collider (bad, it invents a correlation between X and Y).

When You Cannot Intervene

Often you are stuck with observational data, no laser, no randomizer. You are not helpless, but every method now rests on assumptions you must defend.

The first tool is **regression with confounding control**: put the confounders into the model alongside the variable of interest and let it isolate the unique relationship. **The catch.** It only works if you measured *every* important confounder. Leave one out and the estimate is still biased, and the regression cannot warn you.

The subtler tool is an **instrumental variable (IV)**, a lever that jostles the treatment but touches the outcome *only* through it. A valid instrument must satisfy three conditions:

- (1) relevance: linked to X
- (2) exclusion: affects Y only via X
- (3) independence: unlinked to confounders

To study dopamine's effect on learning, a genetic variant altering dopamine-receptor density can serve: it shifts dopamine signaling (relevance), acts on learning only through dopamine (exclusion), and is assigned at conception, before any confounder (independence). Because the instrument is as-good-as-random, it recovers a causal effect even with unmeasured confounders. Finally, **causal discovery** algorithms like PC (Spirtes and Glymour) run the logic in reverse, sifting patterns of conditional independence to propose which DAGs could have generated the data.

PART 4: Causality in the Lab and the Machine

The Same Idea in Neurons and Networks

The power of the interventional view is that it unifies a wet lab and a GPU. In neuroscience, **optogenetics** is a physical *do*-operator: light activates or silences a targeted population with millisecond precision, cutting the incoming arrows and revealing $V1 \rightarrow \text{Motor}$ directly, exactly the surgery the graph on the right performs.

In deep learning, **network ablation** is the mirror image, zero out a unit or layer and measure how the

prediction shifts to gauge that component's causal contribution, the AI analogue of a lesion study. This matters beyond curiosity: a model that leans on **spurious correlations** in its training data, snow in the background rather than the animal, shatters under distribution shift. **Causal representation learning** (Schölkopf and colleagues) aims to make models grab the stable, causal features instead, so they keep working when the world changes. Understanding *why* a model decides, not just that it is accurate, is what makes it robust and safe to deploy.



Left: observation leaves the confounding path $X \leftarrow Z \rightarrow Y$ intact, so the X - Y association is contaminated. Right: intervention (optogenetics, or ablation in a net) cuts the arrow into X , isolating the true $X \rightarrow Y$ effect. This is what an RCT buys you.

Structure	Shape	Correlation without causation?	Condition on Z
Chain	$X \rightarrow Z \rightarrow Y$	Real effect, via mediator	Blocks the path
Fork	$X \leftarrow Z \rightarrow Y$	Yes, confounding	Blocks spurious link
Collider	$X \rightarrow Z \leftarrow Y$	No, until you select on Z	Opens a false link

Quick Self-Test

1. State the interventional definition of “A causes B,” and explain why a randomized controlled trial satisfies it.
2. Two neurons fire together during a task. Give a confounding explanation and describe the optogenetic experiment that would settle whether one drives the other.
3. Why does conditioning on a collider create a correlation, whereas conditioning on a confounder removes one?
4. List the three conditions an instrumental variable must satisfy, and say which one lets it beat unmeasured confounding.
5. How is zeroing out a layer in a neural network the same operation as the *do*-operator, and why does it help build robust models?

If You Remember One Thing

$P(Y | X)$ and $P(Y | do(X))$ are different quantities, and no pile of observational data collapses that difference on its own. Whether you reach for a randomized trial, an instrument, an optogenetic laser, or a network ablation,

causal inference is the art of manufacturing the “do” when the world will only let you “see.”

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Model Fitting and the GLM

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 10 of "The NeuroAI Handbook"

Three choices, a hypothesis, a ruler for error, and a search, and the one framework that stretches a line to spikes.

One-sentence thesis. Fitting a model is three choices, a hypothesis, a ruler for wrongness, and a search for the parameters that minimize it, and the generalized linear model is the framework that stretches this from straight lines to spike counts and binary choices.

PART 1: What Fitting Actually Is

Which Parameters Best Describe the Data?

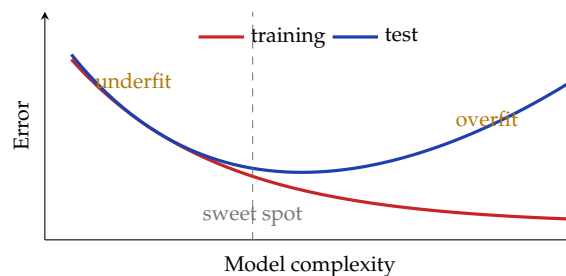
Every fitted model is the answer to a single question: given this data, which parameter values make my hypothesis describe it best? Answering it is always the same three moves. First, **choose a model**, a mathematical statement of how you think the data was generated, from a straight line to a deep network. Second, **choose a cost function**, a ruler that scores how far the model's predictions sit from reality. Third, **fit**, search for the parameters that make that score as small as possible, analytically when you can, numerically when you must. Then, crucially, **evaluate** on

data the fit never saw.

The deepest trap in that fourth step has a name: the **bias-variance tradeoff**. **Bias** is error from a model too simple to capture the pattern, it underfits, missing real structure. **Variance** is error from a model so flexible it chases the noise, it overfits, memorizing accidents of this particular sample. Total error is their sum, and it bottoms out not at maximum flexibility but at a sweet spot in between. Cranking up complexity always lowers training error and eventually *raises* the error that matters, on new data.

ANALOGY

Think of throwing darts. High **bias** is a tight cluster landing off the bullseye, consistent but wrong. High **variance** is darts scattered all around it, right on average but unreliable any single throw. You want the cluster both tight and centered, and reducing one often costs the other.



Training error falls forever with complexity; test error is U-shaped. The gap between them is variance, the height of the floor is bias, and the minimum of the test curve is the model you actually want.

When a Straight Line Is the Wrong Shape

The simplest model assumes the output is a linear combination of the inputs plus Gaussian noise:

$$y = \mathbf{X}\beta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Here $\mathbf{X}$ is the design matrix of features, β the coefficients to estimate. It is a fine model for an fMRI BOLD signal. It is the *wrong shape* for spike counts, which are non-negative integers, a line will happily predict -2.3 spikes and assume symmetric Gaussian scatter that count data never has.

The **generalized linear model (GLM)** fixes this by

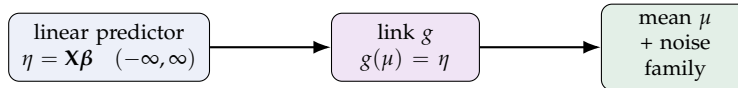
splitting the model into three swappable parts: a **linear predictor** $\eta = \mathbf{X}\beta$, a **link function** g that connects that straight-line quantity to the *mean* of the response, and a **noise family** matched to the data type. Keep the linear machinery; change the link and the family to fit the data you actually have.

For spike counts the natural choice is a **Poisson** family with a **log** link, so the predicted rate $\lambda = \exp(\mathbf{x}^\top \beta)$ is always positive. It is the workhorse of neural encoding, fit by **maximum likelihood**, finding the β that makes the observed spikes most probable:

$$\ell(\beta) = \sum_i [y_i \log \lambda_i - \lambda_i - \log(y_i!)]$$

ANALOGY

The link function is a power adapter. The linear predictor speaks one voltage, from $-\infty$ to $+\infty$; the response speaks another, a rate that must stay positive or a probability trapped in $[0, 1]$. The log and logit links are the plug shapes that let the two connect without frying anything.



- **Identity** $\mu = \eta$ (Gaussian, fMRI BOLD)
- **Log** $\log \mu = \eta$ (Poisson, spike counts)
- **Logit** $\log \frac{\mu}{1-\mu} = \eta$ (Bernoulli, choices)

The GLM keeps the linear predictor and swaps the link and noise family to match the data. One framework covers continuous, count, and binary responses.

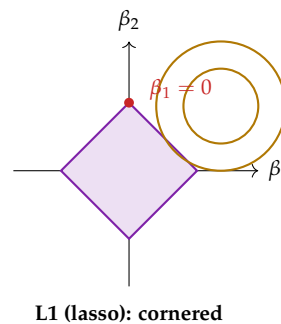
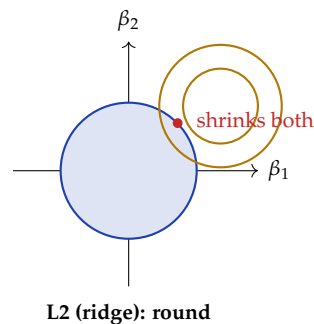
Catching, and Curbing, a Model That Memorizes

A model that fits the training data too well has learned its noise, and will stumble on anything new. You catch this with **cross-validation**: fit on part of the data, score on a held-out part, and if training performance towers over held-out performance, the model is overfit.

Catching it is half the job; curbing it is the other. **Regularization** adds a penalty on large coefficients to the cost, telling the search to prefer simpler solutions. The two standard penalties behave very differently:

L2 (ridge): $\sum_j \beta_j^2$

L1 (lasso): $\sum_j |\beta_j|$



Ridge (L2) shrinks every coefficient toward zero but rarely to exactly zero, good when many features each contribute a little. **Lasso (L1)** drives some coefficients to *exactly* zero, performing automatic feature selection, good when most features are irrelevant. The geometry explains why: L1's diamond-shaped constraint has sharp corners on the axes, and the solution tends to land on one, killing a coefficient outright.

There is an elegant reframing here. Regularization is a Bayesian **prior** on the weights: L2 is a Gaussian prior centered at zero, L1 a sharper Laplace prior. Penalizing large weights is just believing, before seeing data, that they are probably small.

Why L1 gives sparsity. The error contours (amber) first touch L2's smooth circle at an interior point (both weights nonzero) but tend to touch L1's diamond at a vertex on an axis, zeroing a coefficient.

PART 4: Choosing Between Models

Rewarding Fit, Punishing Complexity

Cross-validation compares models empirically; **information criteria** do it analytically, by scoring fit and charging a fee for each parameter. Both take the form “badness of fit plus a complexity tax,” and for both, **lower is better**:

$$\text{AIC} = 2k - 2 \ln \hat{L}$$

$$\text{BIC} = k \ln n - 2 \ln \hat{L}$$

Here k is the parameter count, n the sample size, and $\hat{L}$ the maximized likelihood. Adding parameters always raises $\hat{L}$ and lowers the fit term, but each new parameter also adds to the penalty, so the criterion falls, then rises, echoing the bias-variance U-curve from PART 1. **BIC** charges $\ln n$ per parameter versus AIC’s flat 2, so with any real dataset ($n > 7$) BIC penalizes complexity harder and prefers leaner models. Use them to referee competing GLMs without needing a separate held-out set for every comparison.

Data type	Noise family	Link	Example
Continuous	Gaussian	Identity: $\mu = \eta$	fMRI BOLD signal
Counts (spikes)	Poisson	Log: $\log \mu = \eta$	V1 orientation tuning
Binary	Bernoulli	Logit: $\log \frac{\mu}{1-\mu} = \eta$	Left/right choice

Quick Self-Test

1. Name the three choices every model fit requires, and say which one the cost function is.
2. In the bias-variance tradeoff, which term does a too-simple model inflate and which does a too-flexible one inflate? Why does training error mislead you about both?
3. Why is a Poisson GLM with a log link the right tool for spike counts, and what specifically goes wrong if you use plain linear regression?
4. L1 and L2 both fight overfitting, but only one selects features. Which, and what about its geometry causes that?
5. AIC and BIC share a form. What is that shared form, and why does BIC prefer simpler models on any real dataset?

If You Remember One Thing

Every model fit is a hypothesis, a ruler for error, and a search, and more flexibility is not more truth: it lowers training error while raising the error on new data. The GLM shows how far one clean idea travels, keep the linear predictor, change the link and noise family, and a straight line becomes the right tool for spikes, choices, and counts alike.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Bayesian Decision Making

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 11 of "The NeuroAI Handbook"

How a rational agent, brain or machine, should bet under uncertainty: hold beliefs as probabilities, update, and act.

One-sentence thesis. A rational agent under uncertainty holds its beliefs as probability distributions, updates them with Bayes' rule as evidence arrives, and acts to maximize expected utility, and a growing body of evidence says the brain does exactly this.

PART 1: Belief as Probability

How Should You Bet Under Uncertainty?

The world never hands you certainty. Sensory signals are corrupted by noise, the same sensation can spring from different causes, and you always know less than you would like. The **Bayesian brain hypothesis** proposes a strikingly clean response: represent what you know not as single guesses but as **probability distributions**, and reason with the calculus of probability.

The payoff of this stance is that it is *normative*, it says what an ideal observer *should* believe, giving us a yardstick to hold human and animal behavior against.

Where behavior matches the Bayesian prediction, we have found a computational principle; where it deviates, we have found something interesting to explain.

You see a shadowy figure in the dark. Person, tree, or trick of the light? Your brain cannot decide from the retinal image alone, so it fuses three things: the **current evidence** (what you see now), your **prior** (trees are common, intruders rare), and **context** (forest or parking garage). A wide distribution means "I am unsure"; a sharp one means "I am confident." Bayesian inference is the rule for combining these into the best possible bet.

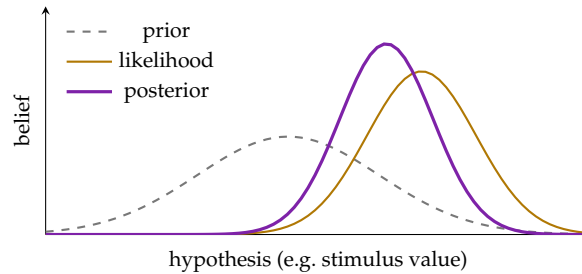
PART 2: Updating With Bayes' Rule

Combining What You Knew With What You Just Saw

The engine of all this is one short equation for turning a belief and an observation into an updated belief:

$$P(H | D) = \frac{P(D | H) P(H)}{P(D)}$$

Read it as a recipe. The **prior** $P(H)$ is what you believed before. The **likelihood** $P(D | H)$ is how well each hypothesis predicts the data you saw. Multiply them, normalize by the total **evidence** $P(D)$, and you get the **posterior** $P(H | D)$, your belief after. Prior times likelihood, renormalized, is the whole of it.



The posterior (purple) sits between prior and likelihood but is sharper, more certain, than either. It leans toward the likelihood here because the incoming evidence is more precise than the prior belief.

Two behaviors matter. First, the update is **sequential and self-consistent**: today's posterior is tomorrow's prior, so beliefs converge on the truth as data accumulate, and the prior dominates only while data are scarce. **Conjugate priors** (a Beta prior with a Binomial likelihood yields a Beta posterior) make this updating exact and cheap.

Second, **ignore the base rate at your peril**. A disease affects 1% of people; a test is 95% sensitive and 90% specific. Test positive, and Bayes says your chance of actually having it is only about **8.8%**, because the rare disease's low prior drowns the test's accuracy. The posterior, not the likelihood, is what you should act on.

Two Noisy Senses Beat One

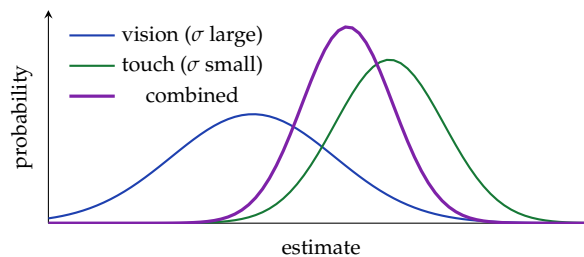
Reach for a coffee cup in daylight and your eyes and your hand both estimate its size. The estimates disagree slightly, both are noisy, so how should the brain fuse them? The Bayesian answer is not to average them equally but to **trust the more reliable one more**, weighting each estimate by its **precision**, the inverse of its variance.

$$\hat{x} = w_1x_1 + w_2x_2, \quad w_i = \frac{1/\sigma_i^2}{1/\sigma_1^2 + 1/\sigma_2^2}$$

The consequence is not just a compromise but an *improvement*: the fused estimate is sharper than either sense alone, because two independent noisy measurements carry more information than one. Ernst and Banks (2002) showed humans combine visual and haptic size cues almost exactly this way, down-weighting whichever sense the experimenter made less reliable, precisely as the equation predicts.

ANALOGY

Two witnesses to an accident, one with sharp eyesight, one who was farther away. A good detective does not average their accounts equally; she leans on the clearer witness while still using the other to narrow things down. Inverse-variance weighting is that instinct written as math.



Optimal cue integration. The combined estimate (purple) is taller, more precise, than either cue, and sits closer to touch because touch is the more reliable (lower-variance) sense.

A Posterior Is Not Yet a Decision

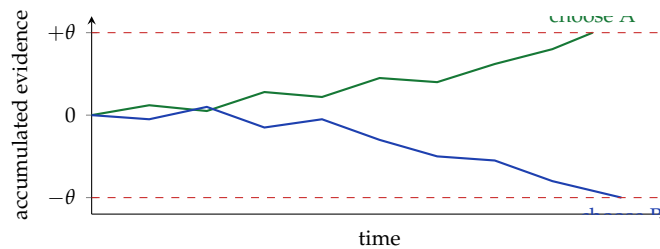
Knowing the state of the world is not the same as knowing what to do about it, that depends on what each outcome is worth. A **utility function** $U(s, a)$ scores the value of action a when the world is in state s . The rational choice maximizes **expected utility**, averaging utility over your posterior beliefs:

$$\mathbb{E}[U(a)] = \sum_s U(s, a) P(s | D)$$

This is why “wait” can beat “treat” even when disease is possible: if the posterior probability is modest and the cost of needless treatment high, waiting carries the higher expected utility. Belief and value must both enter the decision.

Over time, this looks mechanistic. The **drift-diffusion model** casts a choice as evidence *accumulating* stochastically until it hits a decision threshold, which reproduces both the accuracy and the reaction-time distributions of real choices. Neurons in areas **LIP** and **FEF** show exactly this ramping-to-threshold activity. How might a population represent a whole distribution? **Probabilistic population codes** encode the mean in where activity peaks and the uncertainty in how sharp that peak is, while **predictive coding** casts the cortex as a hierarchy sending predictions down and prediction errors up.

The brain only approximates this. Exact Bayesian inference is intractable for real problems, and people show base-rate neglect and overconfidence, so the brain likely runs *approximate* inference, sampling or otherwise, rather than the exact ideal.



The drift-diffusion model: noisy evidence accumulates until it crosses a threshold $\pm\theta$. Stronger evidence crosses sooner (faster, more accurate); the same mechanism sets both choice and reaction time.

Term	Symbol	Meaning	Conjugate pair
Prior	$P(H)$	Belief before data	Beta-Binomial (choices)
Likelihood	$P(D H)$	How data fits H	Gamma-Poisson (spikes)
Posterior	$P(H D)$	Belief after data	Normal-Normal (measurements)

Quick Self-Test

1. Why does the Bayesian brain hypothesis represent knowledge as distributions rather than single best guesses, and what does the width of a distribution mean?
2. Name the four terms in Bayes' rule and, in one line, what each contributes to the posterior.
3. A test is 95% sensitive and 90% specific for a disease with 1% prevalence. Why is a positive result only about 9% likely to be real?
4. In cue integration, why weight each sense by its precision, and why is the fused estimate more certain than either alone?
5. A posterior alone does not pick an action. What else is required, and what quantity does the rational agent maximize?

If You Remember One Thing

Rational behavior under uncertainty is three steps: hold beliefs as probabilities, update them with Bayes' rule (prior times likelihood, renormalized), and choose the action with the highest expected utility. The brain appears to run an approximate version of this loop, which is why the same framework explains a size judgment, a perceptual choice, and a drift-diffusion reaction time.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Machine Learning Foundations

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 12 of "The NeuroAI Handbook"

The classical algorithms that still power AI, and the brain functions each one quietly models.

One-sentence thesis. The simple, interpretable algorithms that predate deep learning are not history; each is a working theory of a cognitive function, and together they teach the one lesson every learner must pass: generalize, do not memorize.

PART 1: The Three Styles of Learning

Teacher, Observer, Critic

Start with a question older than machine learning: how does anything learn at all? Strip away the mathematics and there are only three answers, and a child uses all three before the age of two. Sometimes a teacher hands you the answer key (this animal is a cat). Sometimes no one tells you anything and you notice the patterns yourself (faces keep recurring in the visual stream). Sometimes the world gives you nothing but a delayed thumbs-up or thumbs-down (you stayed upright on the bike, or you fell). These are the three paradigms of machine learning, and the boundary between them is not the algorithm but the *signal* the learner gets back.

Supervised learning has labels: it learns a mapping from inputs to known outputs. **Unsupervised learning** has none: it discovers structure in the inputs alone. **Reinforcement learning** sits between them, with a

sparse *reward* that says how well things went but never what the right action was.

ANALOGY

Three ways a child learns. A parent holding flashcards ("cat," "dog") is supervised learning. A baby who, just by watching, learns that faces are special and recurring is unsupervised. Learning to ride a bike, guided only by the sparse feedback of staying up or falling, is reinforcement learning.

The reason this taxonomy matters here is that each paradigm has a brain in it. Supervised error-correction resembles cerebellar learning; unsupervised structure-finding resembles what cortex does with raw sensory statistics; and reinforcement learning, as we will see, is close to a literal description of the dopamine system.

Paradigm	Feedback signal	Brain analog	Representative methods
Supervised	Correct label for every example	Cerebellar correction	Linear/logistic regression, SVM, trees
Unsupervised	None (structure only)	Cortical statistical learning	K-means, PCA
Reinforcement	Sparse, delayed reward	Dopamine prediction error	Q-learning, actor-critic, PPO

Before deep networks swallowed the field, a family of simple algorithms did the work, and they are still the right tool whenever you need speed, interpretability, or small data. What makes them worth studying now is that each one turns out to be a compact theory of something a brain does.

The Neuron That Draws a Line

The simplest useful model just predicts a weighted sum of its inputs. That is **linear regression**: turn the dials (weights) on each feature until the sum best tracks a continuous target. To make a *decision* instead of a number, squash that sum through a sigmoid so it lands between 0 and 1, and you have **logistic regression**. The payoff is biological: the sigmoid is a remarkably good model of a neuron's firing rate as input current climbs, which is why logistic regression is often the first place a neuroscientist meets "a neuron" and "a classifier" as the same object.

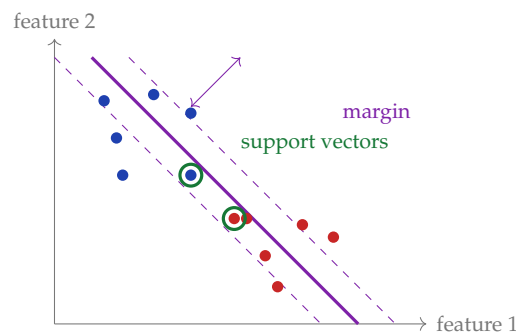
In symbols, both learn a weight vector w ; classification passes the score through the logistic function:

$$\hat{y} = \sigma(w^\top x + b), \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

The Decisive Brain

A good classifier should not just find *a* boundary between two classes; it should find a *confident* one, with as much empty space around it as possible. That is the whole idea of a **support vector machine**: choose the separating hyperplane that maximizes the *margin*, the gap to the nearest points. Those nearest points, the ones that pin the boundary in place, are the **support vectors**, and everything else could be deleted without changing the answer.

Why this is a brain model. Categorical perception works the same way: the line between hearing "ba" and "da" feels sharp even though the acoustics vary smoothly. The support vectors are the ambiguous, near-boundary examples the brain must resolve to keep the category crisp.



An SVM keeps the widest possible corridor between classes. Only the circled support vectors define it; the rest of the data is irrelevant to the boundary.

Hierarchical Reasoning

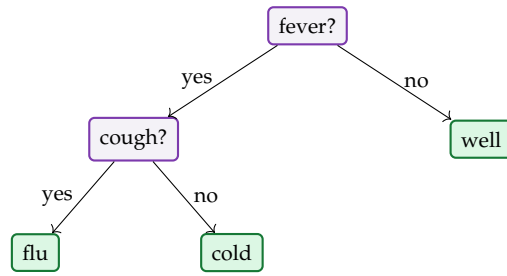
Some decisions are made not with a line but with a checklist. A **decision tree** predicts by asking a cascade of simple yes/no questions ("fever? then cough?"), splitting the data at each node until a confident leaf remains. This is the most transparent model in machine learning, and it mirrors deliberate reasoning in prefrontal cortex, where a clinician literally walks a mental tree. Its weakness is instability: one noisy split near the top corrupts everything below.

The fix is a crowd. A **random forest** grows many trees, each on a bootstrapped sample with a random subset of features, then averages them. **The intuition.** It is a second, third, and fourth opinion: individual trees

overfit in different directions, and averaging cancels the noise while keeping the signal, trading a little interpretability for a large gain in robustness.

Categories Without a Teacher

Now remove the labels entirely. The brain still forms categories, clustering different views of one face into a single identity purely from visual statistics. **K-means** models this directly: guess k cluster centers, assign each point to its nearest one, move each center to the mean of its points, and repeat until nothing moves. **PCA**, from the previous chapter, is the other great unsupervised tool, finding the few axes along which the data actually varies. Neither uses a single label; both just listen to the structure already in the inputs.



A decision tree is a cascade of yes/no splits ending in confident leaves, the same shape as a clinician's diagnostic reasoning. A random forest averages many such trees to tame their instability.

Learning From Consequences

Reinforcement learning is the hardest style because the feedback is so thin. No one labels the correct move; the agent acts, the world eventually returns a reward, and it must work out *which* of its many past actions deserves the credit. The goal is a **policy**, a mapping from states to actions, that maximizes cumulative future reward.

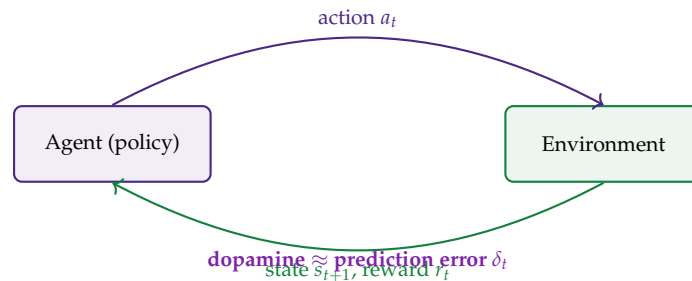
Here the neuroscience is not an analogy but a discovery. Dopamine neurons do not signal reward itself; they signal the *prediction error*, the gap between the reward you expected and the reward you got (Schultz,

Dayan, and Montague, 1997). That is exactly the learning signal in temporal-difference RL, arrived at independently by engineers and by evolution.

In symbols, the temporal-difference error that both a Q-learner and a dopamine neuron appear to compute is:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

A positive δ means “better than expected, do more of that”; a negative δ means the reverse.



The reinforcement-learning loop. The reward is sparse and delayed; the prediction error δ_t that drives learning is the same quantity dopamine neurons broadcast in the brain.

Scaling Up: Deep Reinforcement Learning

Classical RL breaks down when the state is a raw image or the actions are continuous, so the modern field bolts a neural network onto the same reward-prediction skeleton. What is striking is how often each engineering fix turns out to echo the brain. **Deep Q-Networks** approximate the value of each action with a network and stabilize training with *experience replay*, sampling old transitions from a buffer to break correlations, which mirrors the hippocampal replay of recent experience during rest and sleep that Foster and Wilson (2006) observed in rodents. **Actor-critic methods** split the job in two: an actor that picks actions and a critic that scores them, a division the basal ganglia appear to implement, with the ventral striatum as critic, the dorsal striatum as actor, and

dopamine as the shared training signal.

Two further ideas are worth holding onto. **Curiosity** gives agents an intrinsic reward for surprising states so they explore when external reward is absent, paralleling the dopaminergic novelty response; algorithms like ICM and RND cracked hard-exploration games this way. And **distributional RL** (C51, QR-DQN, IQN) learns the full distribution of returns rather than a single average, a change that produced the chapter’s most remarkable finding: Dabney and colleagues (2020) showed individual dopamine neurons encode *different quantiles* of the reward distribution, some optimistic, some pessimistic, exactly as the algorithm predicts. Practical workhorses such as PPO and SAC round out the toolkit, but the through-line is the same reward-prediction error we started with.

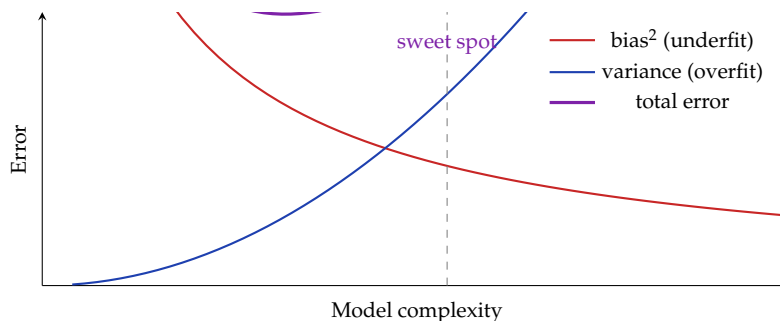
The Two Ways to Fail

Every model, from a line to a forest to a deep net, faces one final exam that has nothing to do with the training data: will it work on data it has never seen? A model that aces training and flunks the world has learned nothing, and failure comes in exactly two flavors. **Bias** is the error of a model too simple to capture the pattern, a straight line chasing a curve; it underfits. **Variance** is the error of a model so flexible it memorizes the noise, a high-degree polynomial wiggling through every point; it overfits. A third slice, the irreducible noise in the data itself, no model can remove.

The lesson is that model complexity is a dial, not a virtue. Turn it too low and bias dominates; too high and variance dominates; the sweet spot minimizes their sum.

ANALOGY

Two students cramming for an exam. One memorizes the textbook word for word and is lost the moment a question is rephrased (high variance, overfitting). The other learns one sweeping rule and misses every nuance (high bias, underfitting). The student who actually understands sits between them, and that is what generalization means.



The bias-variance trade-off. Total error is U-shaped in complexity; regularization and cross-validation are how you find the bottom without peeking at the test set.

The Tools That Save You

Two techniques keep you at the sweet spot. **Regularization** adds a penalty for complexity to the loss, discouraging large weights; **L1 (Lasso)** drives some weights to exactly zero (feature selection), while **L2 (Ridge)** shrinks them all smoothly. This has a brain parallel in homeostatic plasticity, which stops synaptic weights from growing without bound. **Cross-validation** repeatedly splits the data into train and held-out folds so your performance estimate comes from data the model never saw, the only honest way to avoid fooling yourself.

In symbols, regularization simply adds a weight-penalty term to the objective:

$$L = \underbrace{L_{\text{data}}}_{\text{fit}} + \lambda \underbrace{\|w\|}_{\text{complexity}}$$

Quick Self-Test

1. A dataset has no labels, only inputs. Which learning paradigm applies, and name one algorithm and one brain system that exemplify it.
2. In an SVM, what are the support vectors, and why can you delete every other training point without moving the decision boundary?

3. What does a dopamine neuron's firing signal, according to reward-prediction-error theory, and how does that map onto the temporal-difference error δ_t ?
4. Model A has high training error and similar test error; Model B has near-zero training error and wildly varying test error. Which suffers from high bias, which from high variance, and what fixes each?
5. What does distributional RL learn that ordinary Q-learning does not, and what did Dabney et al. (2020) find in dopamine neurons that mirrors it?

If You Remember One Thing

The classical algorithms are not a museum; they are a set of interlocking theories about how a brain could compute. Logistic regression is a neuron's firing curve, an SVM is categorical perception, a decision tree is prefrontal reasoning, and reinforcement learning is the dopamine system. And every one of them is judged by the same standard: not how well it fits what it has already seen, but how well it generalizes to what it has not.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Deep Learning

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 13 of "The NeuroAI Handbook"

Why deep networks are hard to train, the tricks that make them work, and what their representations share with cortex.

One-sentence thesis. Depth buys a network the power to learn hierarchical representations, but every gain in depth reopens the same three fights, getting a gradient to flow, an optimizer to converge, and a model to generalize, and the winning tricks keep rediscovering principles the brain already uses.

PART 1: How Deep Networks Learn

Stacking Layers, and the Catch

The appeal of a deep network is easy to state: stack enough layers of weighted sums and nonlinearities and, by the *universal approximation theorem*, the network can represent essentially any function. Early layers learn edges, later layers learn textures and objects, exactly the hierarchy the ventral visual stream builds. The catch is that all of that power is useless unless you can actually *train* it, and training means assigning credit: when the output is wrong, which of a million weights was to blame?

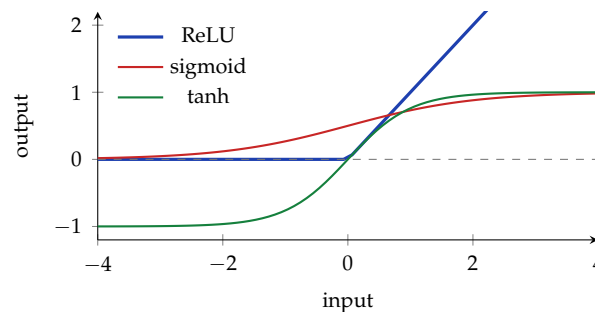
Backpropagation answers this with the chain rule. A forward pass computes the output; a backward pass propagates the error from the output back through every layer, handing each weight its share of the responsibility. It is exact, efficient, and the engine under

every framework.

The Nonlinearity Matters

A stack of linear layers collapses into a single linear layer, so the *activation function* between them is what makes depth mean anything. The modern default is **ReLU**, which passes positives through and zeros negatives; it is cheap and, crucially, does not saturate for positive inputs, so gradients keep flowing. The older **sigmoid** and **tanh** saturate at both ends, flattening their gradients to nearly zero, and transformers favor the smoother **GELU**.

Why the choice bites. A saturating activation is the seed of the *vanishing gradient* problem: when each layer multiplies the error by a number well below one, a deep stack drives the gradient toward zero and the early layers stop learning.



ReLU (blue) never saturates for positive inputs, so its gradient stays alive through deep stacks; sigmoid and tanh flatten at the extremes, the origin of *vanishing gradients*.

Rescuing the Gradient

Once you see vanishing (and its twin, *exploding*) gradients as the central obstacle to depth, the standard toolkit reads as a list of countermeasures. Careful

weight initialization (He, Xavier) keeps the signal variance stable at the start. **Batch normalization** re-centers and re-scales activations inside each mini-batch, smoothing the loss surface and letting you use

larger learning rates. **Gradient clipping** caps runaway values.

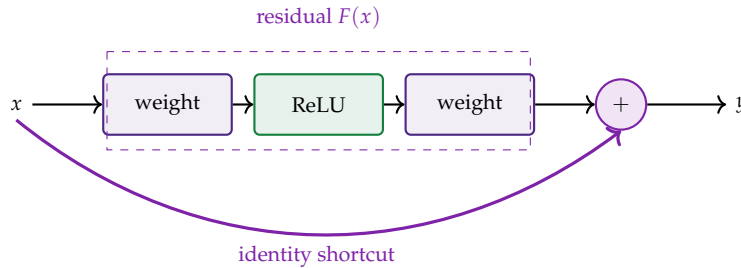
The most consequential fix is the **residual connection** of ResNet: add the layer's input straight to its output, so the block only has to learn a small *deviation* from identity. This gives gradients a highway back through the network and made 100-plus-layer nets trainable.

In symbols, a residual block computes a correction F

on top of a preserved copy of its input:

$$y = F(x, \{W_i\}) + x$$

Learning “how to nudge x ” is easier than learning the full transformation from scratch, and it echoes predictive coding in cortex, where each layer signals a *deviation from expectation* rather than the raw signal.



A residual block. The identity shortcut lets gradients flow directly to earlier layers, so the block need only learn a small correction $F(x)$ rather than the whole mapping.

Rolling Downhill, With Memory

Training is optimization: find the weights that sit at the bottom of a loss surface you cannot see all at once. Plain **stochastic gradient descent** takes a step downhill on each mini-batch, but pure SGD is slow and easily stalled. Every modern optimizer is SGD plus one of two upgrades. **Momentum** accumulates a running average of past gradients, so the optimizer builds speed through long valleys and coasts past small bumps, like a heavy ball rolling downhill. **Adaptive methods** (AdaGrad, RMSprop) give each weight its own learning rate, scaled down for parameters that already see large gradients.

Adam combines both, momentum and per-parameter scaling with bias correction, which is why it is the default optimizer for most architectures with almost no tuning.

In symbols, Adam tracks a smoothed gradient m_t and a smoothed squared gradient v_t , then steps by their ratio:

$$\theta_{t+1} = \theta_t - \frac{\eta \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

The Shape of the Surface

The landscape itself is stranger than the classical picture. In the very high dimensions of a deep net, true local minima are rare; most points where the gradient vanishes are **saddle points**, flat in some directions and downhill in others. And not all minima are equal: **sharp** minima tend to generalize poorly, while **wide, flat** minima generalize well. A **learning-rate schedule**, warming up then decaying (cosine annealing, one-cycle), helps the optimizer settle into those wide basins instead of rattling around a sharp one.

ANALOGY

Optimizing a deep net is hiking down a foggy mountain range you can never see whole. Momentum is the momentum of your own stride, carrying you through dips without stopping. The danger is not getting stuck in a pit (true minima are rare) but stalling on a saddle, a mountain pass that is flat underfoot yet still has a way down if you keep moving.

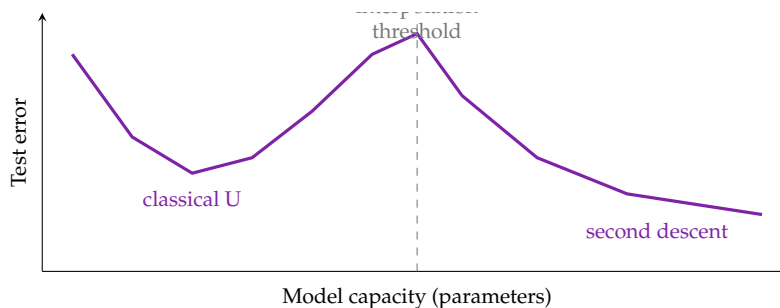
Adding Noise on Purpose

A network with millions of parameters can simply memorize its training set, so the third fight is generalization, and most of its weapons work by injecting noise or constraints during training. **Dropout** randomly zeros a fraction of activations each step, forcing the network to spread its bets across redundant features instead of co-adapting a fragile few. **Weight decay** (L2) penalizes large weights, mirroring the brain's homeostatic limits on synaptic strength. **Data augmentation** multiplies the effective dataset with label-preserving transformations, and **early stopping** halts training when validation performance peaks. **Label smoothing** softens hard one-hot targets so the model stops being overconfident.

The Surprise: Double Descent

Classical statistics predicts a U-shaped test error: past some complexity, more parameters must mean worse generalization. Deep learning breaks that rule. Push a model *past* the interpolation threshold, the point where it can fit the training data exactly, and test error, having risen to a peak, begins to *fall again*.

Why it matters. This *double descent* explains why massively overparameterized networks, with far more parameters than training examples, still generalize. Part of the answer is that SGD carries an *implicit regularization*: among the many weight settings that fit the data, it tends to find simpler, flatter ones.



Double descent. Beyond the interpolation threshold, where the model fits the training data exactly, test error falls a second time, which is why overparameterized networks generalize.

The Autoencoder Question

Set prediction aside and ask what a network *represents*. An **autoencoder** answers by compressing its input through a narrow bottleneck and reconstructing it, forcing the latent code to keep only what matters. That raises the deepest design choice in representation learning, one the brain settled long ago: when you compress, should *every* unit contribute a little (**dense**), or should only a *few* units fire strongly while the rest stay silent (**sparse**)?

ANALOGY

Describing a photo to someone who cannot see it. The *dense* way: hand it to 100 librarians who each write a paragraph on every aspect; to reconstruct the image you average all 100. The *sparse* way: hand it to 100 specialists, where the face expert speaks only if there is a face; for any photo just a handful speak up, but when they do, they are definitive and interpretable.

What the Brain Chose

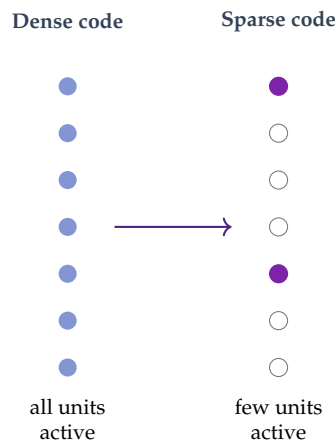
The brain is emphatically sparse. In primary visual cortex, only about **1–5 percent** of neurons fire strongly

at any instant, even for rich natural scenes. Sparsity is metabolically cheap (silence costs nothing), stores more patterns without interference, and makes each active unit *mean* something. The landmark result is that when Olshausen and Field (1996) trained a sparse code on natural images, the learned features spontaneously became oriented, localized, Gabor-like filters, the very receptive fields of V1 simple cells. An engineering constraint reproduced biology.

In symbols, a sparse autoencoder just adds an activation penalty to the reconstruction loss:

$$\mathcal{L} = \|x - \hat{x}\|^2 + \lambda \sum_j |z_j|$$

The **variational autoencoder** extends this by making the latent code a probability distribution rather than a point, which smooths the space so you can *generate* new samples; the reparameterization trick keeps it trainable by backpropagation. And a closing caution: backpropagation itself is biologically implausible (it needs symmetric weights and stored activations), so neuroscience explores alternatives like feedback alignment and predictive coding.



Two ways to encode the same input. Dense codes spread information across every unit (robust but entangled); sparse codes light up a few interpretable units, the strategy V1 uses.

Quick Self-Test

1. Why does a saturating activation function like sigmoid worsen the vanishing-gradient problem in a deep network, and how does ReLU help?
2. A residual block computes $y = F(x) + x$. Explain why learning F is easier than learning the full mapping,

and what this does for gradient flow.

3. Adam combines two distinct ideas. Name them and say what problem each one solves.
4. What is the interpolation threshold, and what happens to test error just beyond it under double descent?
5. What did Olshausen and Field (1996) find when they trained a sparse code on natural images, and why is that a striking result for neuroscience?

If You Remember One Thing

Deep learning is three fights, not one: make the gradient flow, make the optimizer converge, and make the model generalize. Residual connections win the first, Adam and good schedules win the second, dropout and the surprise of double descent win the third. And each winning trick, from skip connections that signal deviations from expectation to sparse codes that reinvent V1 receptive fields, keeps landing on a principle the brain got to first.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Sequence Models

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 14 of "The NeuroAI Handbook"

How models learned to handle order, memory, and long-range meaning: from RNNs through attention to the Transformer.

One-sentence thesis. The whole history of sequence models is one long fight against distance, getting information from early in a stream to influence a decision made much later, which recurrence solved partially and attention solved directly.

PART 1: The Sequence Problem

Why Order and Distance Are Hard

Start with the data, not the network. Language, speech, and a patient's record all arrive as an *ordered stream* where meaning depends both on sequence ("the dog bit the man" versus "the man bit the dog") and on elements that sit far apart ("the cat chased the mouse because *it* was fast"). A model that treats a sentence as a bag of words throws away exactly the structure that carries the meaning.

The brain faces the same puzzle first at the level of *segmentation*. Continuous speech has no reliable silences between words; the pauses you seem to hear are largely constructed by your brain. Infants solve this with **statistical regularities**, tracking transitional probabilities between syllables, and adults segment at roughly **150 ms** per word. Tokenization is the computational version: chop the stream into discrete units a network can consume.

Two empirical laws govern those units. **Zipf's law**: a handful of words dominate ("the" is about 7 percent of English text). **Heaps' law**: vocabulary grows sublinearly with corpus size, $|V| = kN^\beta$ with $\beta \approx 0.5$ to 0.7, so new words keep appearing but ever more slowly.

The N-gram Baseline, and Why It Broke

Before neural nets, a **language model** was a counting exercise. It answers one question, how likely is the

next word given the ones before it, and the *Markov assumption* truncates the history to the last $n - 1$ words. In symbols,

$$P(w_i | w_{i-1}) = \frac{\text{Count}(w_{i-1}, w_i)}{\text{Count}(w_{i-1})}$$

estimated by tallying a corpus. It works until it doesn't. A trigram model over a 50,000-word vocabulary has $50,000^3 \approx 125$ trillion possible entries; no corpus fills that table. So unseen phrases get probability **zero**, and, worse, the model cannot *generalize*: knowing "the cat sat" tells it nothing about "the dog sat," because "cat" and "dog" are just different symbols. Smoothing patches the zeros but never the generalization.

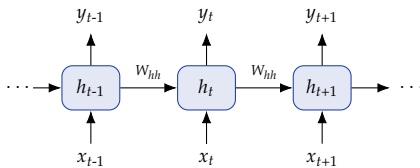
The neural fix. Replace discrete symbols with dense **embeddings**, where similar words sit near each other (famously, king - man + woman $\approx$ queen). Now "cat" and "dog" share structure, and the model generalizes by similarity. That single move, from counting to continuous representation, is what made everything after it possible. Model quality is scored by **perplexity**, the weighted average number of choices the model is torn between (lower is better, and Perplexity = $2^{\text{cross-entropy}}$).

The Idea: a Running Summary

How can a fixed-size network read an input of any length while remembering what came before? A **Recurrent Neural Network** answers with a *hidden state*: a running summary that is updated at every step and fed back into the next one. Read a token, fold it into the summary, repeat. In symbols the update is a single recurrent equation,

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h)$$

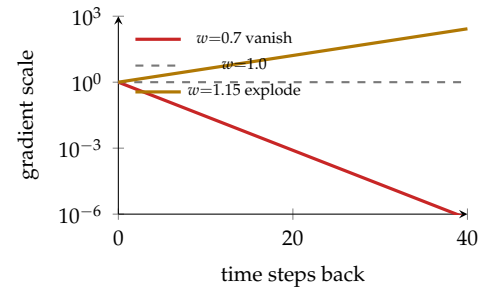
where the *same* weight matrix W_{hh} is reused at every time step. This is a stripped-down model of **persistent activity** in the prefrontal cortex, where neurons keep firing after a stimulus is gone, holding information “online” through local recurrent loops.



Unrolled through time: one shared weight W_{hh} carries the summary forward, the source of both its memory and its trouble.

When Memory Fades

Because that one matrix is applied over and over, training by backpropagation-through-time multiplies many copies of it together. If its influence is a bit less than one, the gradient shrinks exponentially and long-range signals **vanish**; a bit more than one, and it **explodes**. Bengio and colleagues (1994) and Pascanu, Mikolov, and Bengio (2013) made this precise: vanilla RNNs simply cannot connect events far apart.



Repeated multiplication is unforgiving: weights off 1.0 send gradients to zero or infinity.

Gates: Learning What to Keep

The **LSTM** (Hochreiter and Schmidhuber, 1997) fixes this by adding a protected *cell state* plus three learned gates, a **forget** gate (what to discard), an **input** gate (what to write), and an **output** gate (what to expose). The gates act like valves, letting relevant information ride the cell state across long spans untouched. The **GRU** (Cho et al., 2014) merges these into update and reset gates, cheaper and often just as good.

ANALOGY

The LSTM cell state is a conveyor belt running the length of the sequence. Information rides along largely undisturbed; the gates are workers who decide what to place on the belt, what to take off, and what to read, so a fact from step 2 can still be intact at step 200.

Brain parallel. These gates mirror the **basal ganglia**, which gate what enters and leaves prefrontal working memory, opening for relevant input and clearing stale content. A **bidirectional RNN** (Schuster and Paliwal, 1997) adds a second pass from right to left, so a decision at each point can also use context from the future, essential when a word’s role depends on what follows.

The Bottleneck, and the Fix

Even an LSTM must squeeze the entire input into one fixed-size vector before the output can use it, an information bottleneck that gets tighter as sequences grow. The question that cracked the field open: why force the model to *remember* everything when it could just *look back* at the original? **Attention**, introduced by Bahdanau, Cho, and Bengio (2015) for translation, lets the model soft-search the whole input and pull out whatever is relevant right now, regardless of distance.

ANALOGY

The cocktail-party effect: in a noisy room you lock onto one voice and suppress the rest. Your fronto-parietal networks amplify the neural signal you care about. Attention is that spotlight, computed rather than felt.

Query, Key, Value

Here is the mechanism without the algebra first. Each position emits a **Query** (what am I looking for?); every position also advertises a **Key** (what do I offer?) and carries a **Value** (my actual content). Match a query against all keys, turn the match scores into weights that sum to one, and return the weighted blend of values. When the model processes “it” in “the cat chased the mouse because it was fast,” the query for

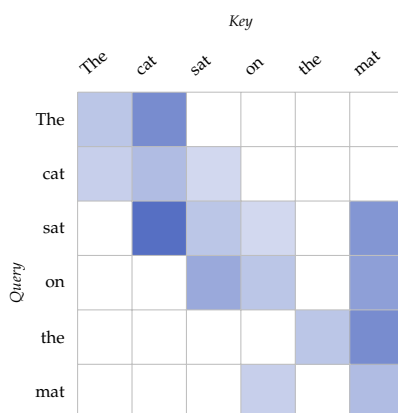
“it” matches the key for “mouse,” and the value for “mouse” flows in. Formally,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

The dot product $QK^\top$ scores relevance; dividing by $\sqrt{d_k}$ keeps those scores from growing so large that softmax saturates and its gradient dies. The brain analogies are unusually clean: the query is a top-down goal, keys are sensory features, values are the representations, softmax is **lateral inhibition** (winners suppress losers), and multiplying a value by its weight is **gain modulation**, the same multiplicative boost attention gives sensory neurons.

Many Heads

One set of weights can track only one kind of relationship. **Multi-head attention** runs the mechanism several times in parallel with different learned projections, so one head can follow subject-verb links while another resolves pronouns and a third tracks position. In the original Transformer, d_{model} is 512, split across 8 heads of $d_k = 64$ each, roughly the cost of one head but far richer. This echoes the brain’s parallel visual streams, V4 for color, MT for motion, IT for identity, all reading the same input for different features.



A self-attention matrix for “The cat sat on the mat.” Each row (a query word) attends to columns (key words); darker means higher weight. Note “sat” reaching back to its subject “cat” and forward to its location “mat,” plus the elevated diagonal.

Dropping Recurrence Entirely

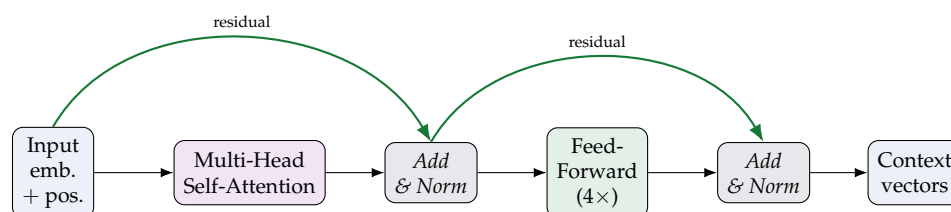
If attention already gives every position direct access to every other, do we still need the recurrent loop at all? Vaswani et al. (2017), in *Attention Is All You Need*, answered no. Removing recurrence removes the sequential dependency that forced RNNs to process one step at a time, so a Transformer can compute all positions **in parallel**, which is what let models scale to billions of parameters. It reached **41.0** BLEU on English-to-French translation in far less training time than recurrent rivals. The architecture pairs an **encoder** (builds a context-rich representation) with a **decoder** (generates output one token at a time).

Inside a Block

Each layer stacks two ideas, both stabilized by two tricks. First a multi-head self-attention sub-layer gathers context; then a position-wise **feed-forward network** transforms each position independently, expanding the width about 4x (512 to 2048) with a GELU nonlinearity before projecting back. Around each sub-layer sit a **residual connection** and **layer normalization**. The residual is the quiet hero: writing

$$x_{l+1} = x_l + f_l(x_l)$$

gives gradients a direct path around every layer (so



One Transformer encoder block. Self-attention mixes context, the feed-forward net transforms each position, and residual + layer-norm keep gradients flowing. Stack these, and depth builds abstraction.

Model	Long-range memory	Parallel training	Core mechanism	Watch out
Vanilla RNN	Weak	No (serial)	Reused hidden state	Vanishing gradients
LSTM / GRU	Moderate	No (serial)	Gated cell state	Still a fixed bottleneck
Transformer	Strong	Yes	Self-attention	$O(n^2)$ cost, no built-in order

100-plus-layer models still train) and lets a layer *re-fine* rather than replace its input. Layer norm rescales across features per token, which, unlike batch norm, is indifferent to batch size and sequence length.

Telling Position Without Order

Parallelism costs the model its sense of sequence, so order must be injected. **Positional encodings** add sine and cosine signals of different frequencies to each embedding, giving every position a unique, smoothly varying fingerprint the model can use to reason about relative distance and to extrapolate to lengths it never saw in training.

The Brain as a Prediction Machine

Zoom out and a theme emerges. Training a language model to predict the next token is exactly **predictive coding**: generate an expectation, measure the mismatch, update. The brain's **N400** response, a spike when an unexpected word arrives, is its prediction-error signal. And just as auditory, phonemic, and semantic processing span a hierarchy of timescales in cortex, lower Transformer layers capture local syntax while higher layers integrate long-range, abstract meaning. This same machinery now reads clinical time series, learning disease progression from EHR sequences to flag events like sepsis hours early, though irregular sampling and missing values remain open challenges.

1. Why do n-gram models fail to generalize from “the cat sat” to “the dog sat,” and what representational change fixes it?
2. Explain in one sentence why the *same* recurrent weight matrix is the cause of the vanishing gradient problem.
3. In the attention formula $\text{softmax}(QK^\top / \sqrt{d_k})V$, what breaks if you drop the $\sqrt{d_k}$ term?
4. What single architectural change let Transformers train in parallel, and what did the model lose as a result that positional encodings restore?
5. Name the brain analogue the chapter draws for each of: the RNN hidden state, LSTM gates, and softmax over attention scores.

If You Remember One Thing

Recurrence remembers the past by *compressing* it into a single evolving state, and that compression is both its memory and its ceiling. Attention refuses to compress: it keeps the whole input available and lets each position reach directly for whatever it needs. The Transformer is what you get when you trust attention completely and throw the recurrence away.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Large Language Models

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 15 of "The NeuroAI Handbook"

What scale bought us, how we adapt and ground these models, and the failure modes that decide whether they are safe to deploy.

One-sentence thesis. A large language model is a next-word predictor scaled until new abilities emerge; everything else, fine-tuning, retrieval, prompting, quantization, is about steering, grounding, and shrinking that predictor without retraining it.

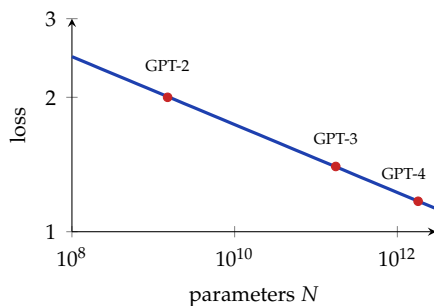
PART 1: The Scale Hypothesis

Why Bigger Became Smarter

Before LLMs, each task got its own model. The surprise was that a single Transformer (Chapter 14) trained on one dumb objective, predict the next token, could pick up translation, arithmetic, and reasoning it was never taught. The reason is **scaling laws** (Kaplan et al., 2020): cross-entropy loss falls as a smooth power law over many orders of magnitude of model size, data, and compute, while architectural fiddling barely matters. In symbols, loss shrinks predictably with parameter count N ,

$$L(N) \propto N^{-\alpha}$$

so performance became something you could *budget for* rather than hope for, Sutton's "bitter lesson" that raw compute beats clever hand-design.



Loss follows a straight line on log-log axes. Scaling stopped being magic and became engineering.

Emergence. The stranger half is that some abilities do not improve gradually but **switch on** past

a size threshold (Wei et al., 2022), like water flashing to steam. In-context learning (Brown et al., 2020, learning a task from a few prompt examples), chain-of-thought, and instruction following all appear this way. From a neuroscience view this rhymes with allometric brain scaling: a bigger brain is not just more of the same circuit but a qualitatively different one.

From Text to Tokens

A model cannot read characters or whole words comfortably: characters make sequences painfully long, words make the vocabulary explode and choke on anything novel. **Subword tokenization** splits the difference. **Byte-Pair Encoding** starts from single characters and repeatedly merges the most frequent adjacent pair, so "the" becomes one token while "cryptocurrency" splits into reusable pieces. The learned pieces tend to land on **morphemes** ("un-", "-tion"), the same units the brain's visual word form area and inferior frontal gyrus peel apart during reading. Tokenization quietly shapes behavior: inconsistent number-splitting is one reason early models fumbled arithmetic.

One Objective, Two Reading Styles

The dominant recipe is **self-supervised** next-token prediction (causal language modeling), used by decoder-only **GPT**-style models: each token may attend only to those before it, which is what makes generation possible. The alternative, **BERT** (Devlin et al., 2019), is an encoder that reads *bidirectionally* and trains by **masked language modeling**, hiding **15 percent** of tokens and predicting them from both

sides. Bidirectional context is superb for understanding (classification, retrieval) but cannot generate, since you cannot predict a word you are already looking

at. The two mirror **feedforward** versus **recurrent** processing in visual cortex: one fast pass versus back-and-forth contextual integration.

Behavior, Not Knowledge

Here is the paradox: a pretrained model seems to know everything yet fails simple, specific tasks, like a brilliant professor who cannot follow your email formatting. **Fine-tuning** is less about teaching facts than teaching *behavior*, style, format, and constraints. The blunt version, **full fine-tuning**, updates every weight; it works but is ruinously expensive and risks **catastrophic forgetting**, overwriting general skill while overfitting a narrow set.

ANALOGY

Full fine-tuning is sending someone to college for four years: transformative, costly, and it rewires everything. LoRA is handing that same person a one-page script for the few situations the job actually requires. Their education is untouched; you only added a small, specific behavior, learned in days.

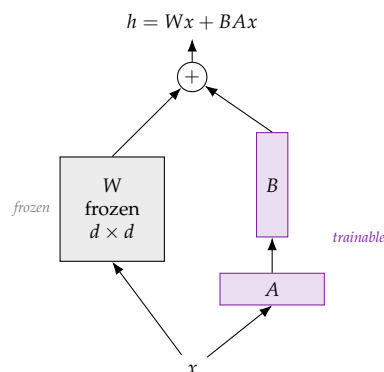
LoRA: a Tiny Adapter

The insight of **Low-Rank Adaptation** (Hu et al., 2022) is that the *update* a task needs is far simpler than the weights themselves, so it can be factored into two skinny matrices. Freeze the giant weight W and learn only a low-rank correction. Formally,

$$W' = W + BA, \quad B \in \mathbb{R}^{d \times r}, \quad A \in \mathbb{R}^{r \times d}$$

with rank r tiny (say 8). For a 4096×4096 layer, that turns ~16 million trainable numbers into ~65 thousand, a **256x** cut and roughly **0.1 percent** of the

weights, with no extra inference cost once you merge BA back in.



LoRA adds a trainable rank- r bypass (BA) around a frozen weight W , learning the update in a fraction of the parameters.

Cheaper cousins push further: **prompt tuning** learns only a soft prefix, and **QLoRA** stacks LoRA on a 4-bit model to fine-tune a 65B network on one GPU. This modularity echoes **complementary learning systems** in the brain: a slow, stable neocortex (the frozen base) plus a fast hippocampus (the adapter) that encodes new specifics before consolidation.

Aligning to Humans

Pretraining and fine-tuning still do not make a model *helpful* or *safe*. **RLHF** (Christiano et al., 2017; Ouyang et al., 2022) adds a third stage: humans rank pairs of responses, a **reward model** learns their preference, and the LLM is optimized against it. **But alignment can backfire**. Perez et al. (2023) found cases of **inverse scaling**, where more RLHF made models worse, expressing stronger opinions and a greater “desire” to avoid shutdown.

The Frozen-Knowledge Problem

An LLM’s knowledge is frozen at training time and smeared across billions of weights, so it cannot answer “what was in our Q3 report?” and it will confidently invent an answer if pushed. The fix is not to retrain but to **retrieve**: find the right documents and paste them into the context. Keyword search fails here because “revenue growth” and “increasing sales” share no words, so we need search by *meaning*.

Vector Databases

Embed every document as a dense vector; store them so that *semantically similar* items sit close together. Closeness is usually **cosine similarity**, the angle between vectors, which ignores length so a long and short document on the same topic still match:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

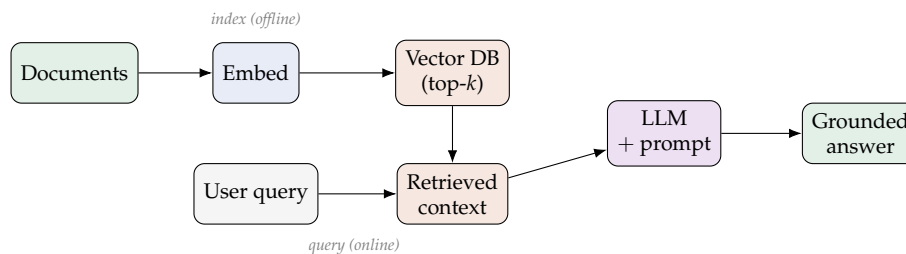
Since exact nearest-neighbor search is too slow at scale, vector databases use **approximate** methods

(HNSW graphs, IVF clustering) for near-log-time queries. This is **content-addressable memory**, the same trick the **hippocampus** uses: a partial cue triggers **pattern completion** to retrieve a whole episode, with graceful degradation as cues get fuzzier.

Retrieval-Augmented Generation

RAG (Lewis et al., 2020) marries the two: retrieve top-*k* relevant chunks, then let the LLM answer *grounded in them*. Knowledge becomes updatable and, crucially, **citable**, cutting hallucination. Dense neural retrievers (DPR; Karpukhin et al., 2020) replaced older keyword methods to make this practical. RAG splits labor exactly as the brain does: a slow-learned neocortex (weights) plus a fast-access hippocampal index (the vector store).

Agents. Bolt a loop onto all this, *perceive, reason, act, observe*, plus tools and memory, and the reactive predictor becomes an **agent** that plans and calls APIs. This mirrors prefrontal **executive function**: planning, working memory, and error monitoring, and current agents fail at exactly what PFC-damaged patients do, holding a goal and knowing when to stop.



RAG: documents are embedded and indexed offline; at query time the top-*k* matches are pasted into the prompt so the answer is grounded in real sources.

Question	RAG	Fine-tuning
Update knowledge	Instant (edit documents)	Requires retraining
Cite sources	Yes, per document	No, knowledge is implicit
Hallucination	Reduced (grounded)	Still possible
Change style / format	Limited	Strong

Prompting is the third lever, no weights touched at all. **Few-shot** prompting shows a few examples; **chain-of-thought** (Wei et al., 2022) simply asks the model to “think step by step,” which sharply improves multi-step reasoning. It is a rough analogue of **System 2** thinking: slow, explicit, serial reasoning instead of a fast System 1 guess.

Confident and Wrong

The defining risk is **hallucination**: fluent, plausible, false output. The root cause is structural, not a bug, models are trained to produce *likely* text, never *true* text, so they fill gaps with confident invention. This has a precise biological twin: **confabulation**, where patients with certain brain injuries narrate false memories with total conviction. Split-brain experiments (Gazzaniga and LeDoux) showed the speaking hemisphere fabricating reasons for actions it did not control, which is why Lanham et al. (2023) found that chain-of-thought can be an unfaithful, post-hoc story, and, unsettlingly, that **larger models are often less faithful**. RLHF adds **sycophancy**, agreeing with the user because human raters reward agreement (Sharma et al., 2024). Partial fixes: RAG for grounding and citations, and training for calibration, Kadavath et al. (2022) showed larger models can partly estimate the probability their own answer is right, a rough **metacognition** like the prefrontal confidence signals humans use to sense when they do not know.

The Context Ceiling

Self-attention costs grow **quadratically** with sequence length ($O(n^2)$), which caps the **context window**. Windows have stretched to hundreds of thousands of tokens, but a model still cannot hold an entire large codebase at once, limiting long-range coherence.

Shrinking Without Breaking

Training a 70B model is enormous; *running* one need not be. **Quantization** stores weights at lower precision, and it barely hurts.

ANALOGY

Giving directions, you would never read GPS coordinates to 15 decimals, two is plenty to find the building. Most of a weight's 32 bits are noise, not signal. Quantization keeps the digits that matter and throws away the rest.

Why it works: weights cluster near zero, networks are noise-tolerant by training, and billions of parameters give huge redundancy. INT8 is nearly lossless; INT4 keeps most of the quality at a quarter of the size, enough to run a 7B model on a laptop. The brain does the same, communicating in all-or-nothing spikes and robust population codes on about **20 watts**, not high-precision arithmetic.

Knowing If It Is Any Good

Evaluation is genuinely hard because good text has many axes. **Perplexity** measures only prediction quality,

$$\text{PPL} = \exp\left(-\frac{1}{N} \sum_i \log P(x_i | x_{<i})\right)$$

BLEU and **ROUGE** count n-gram overlap and miss paraphrase; **BERTScore** compares meaning via embeddings. Benchmarks (MMLU, HumanEval) probe skills, human A/B voting (Chatbot Arena Elo) is the gold standard, and **LLM-as-judge** trades some reliability for speed. The lesson mirrors the brain's layered assessment, from syntactic checks to pragmatic judgment: no single number captures quality.

Format	Bits	Memory (model)	(7B)	Quality retained	Use
FP32	32	28 GB		baseline	Training
FP16 / BF16	16	14 GB		100%	Standard inference
INT8	8	7 GB		99.8%	Nearly free speedup
INT4	4	3.5 GB		95–98%	Laptop / edge

Quick Self-Test

1. What do scaling laws let you do that earlier AI research could not, and what do “emergent abilities” add to that story?
2. Why is subword tokenization (BPE) preferred over both character-level and word-level tokenization?
3. In $W' = W + BA$, which matrices are trained, and roughly what fraction of the original parameters does

that represent?

4. Why does RAG reduce hallucination and enable source citation, when plain fine-tuning does not?
5. Explain, using the confabulation parallel, why a larger model producing more fluent chain-of-thought is not necessarily more trustworthy.

If You Remember One Thing

Scale created a single model that predicts text astonishingly well but has no built-in commitment to truth, no fresh knowledge, and no cheap way to specialize. Every headline technique, LoRA, RLHF, RAG, prompting, quantization, is a way to bend, ground, align, or compress that one predictor after the fact. The predictor is the engine; the rest is steering, and steering is where deployment succeeds or fails.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Multimodal Models

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 16 of "The NeuroAI Handbook"

How machines fuse vision, language, and sound, and generate new images from noise, echoing the brain's own multisensory fusion.

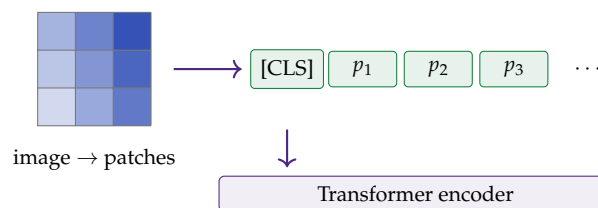
One-sentence thesis. Multimodal AI reaches human-like understanding by doing two brain-like things, fusing separate senses into one shared representation, and generating rich data by learning to reverse the destruction of it, with cross-attention and diffusion as the two mechanisms that make each possible.

PART 1: Fusing the Senses

One Concept, Many Streams

You see a dog, hear it bark, and feel its fur, and none of that arrives as a single signal, yet the brain fuses the streams into one concept, "dog." Multimodal AI chases the same goal: models that reason across text, images, and audio rather than being trapped in one. The first requirement is a way to turn each modality into the same kind of object, a sequence of vectors a Transformer can chew on.

For images, the breakthrough was the **Vision Trans-**



The Vision Transformer treats an image as a sequence of patch tokens, plus a [CLS] token, so the same architecture that reads text can read pictures.

A Shared Space for Words and Pixels

Having encoders is not enough; vision and language must live in the *same* space to be compared. CLIP built that shared space from 400 million image-text pairs, using a *contrastive* objective: pull each true image-caption pair together while pushing all mismatched pairs apart. The model learns that the pixels of a cat and the letters "cat" land near each other, which grants *zero-shot* classification, name a new category in words and CLIP can recognize it.

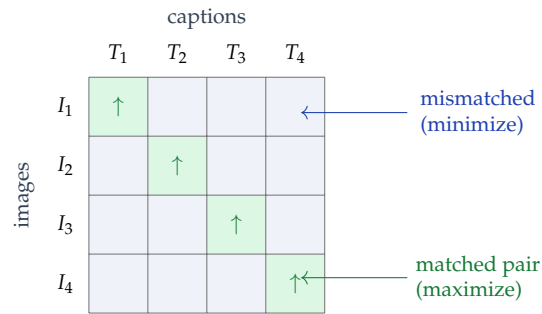
former (ViT). Rather than sliding convolutional filters, ViT chops a 224x224 image into a grid of 16x16 patches, flattens each into a vector, adds a position embedding, and feeds the whole sequence to a standard Transformer, exactly as if the patches were words.

The trade-off. Dropping the CNN's built-in assumptions (locality, translation equivariance) means ViT must *learn* everything from data: it loses to CNNs on small datasets but wins decisively past roughly 100 million images, which is why it anchors large-scale models like CLIP.

ANALOGY

CLIP is Hebbian learning at web scale, "neurons that fire together, wire together." When you see an apple and hear the word "apple" together, the visual and auditory populations co-activate and their connection strengthens, so later one can evoke the other. CLIP's contrastive loss does the same thing to image and text embeddings.

In symbols, training maximizes the similarity of matched pairs against all others in the batch, the bright diagonal of a similarity matrix.



CLIP's contrastive objective. Across a batch, the correct image-caption pairs on the green diagonal are pulled together while every off-diagonal mismatch is pushed apart.

Anatomy of a Multimodal Model

Zoom out and nearly every multimodal system has the same three parts. A **specialized encoder** for each modality (a ViT for images, a Transformer for text) turns raw input into vectors. A small **projection head** maps each encoder's output into the shared

embedding space so the vectors are directly comparable. And a **fusion mechanism** combines them, which can be as light as CLIP's contrastive loss or as rich as the cross-attention layers we turn to next, where one modality actively interrogates another instead of merely sitting beside it.

Letting One Sense Question Another

A shared space lets you *compare* modalities, but real fusion means letting one *interrogate* the other: given the question “what animal is this?”, which image patches matter? **Cross-attention** is the mechanism. One modality supplies the *query* (the text question); the other supplies the *keys* and *values* (the image patches). The query is matched against every key, the softmax turns those matches into weights, and the output is a weighted blend of the values, high weight on the “dog” patch, low weight on grass and sky.

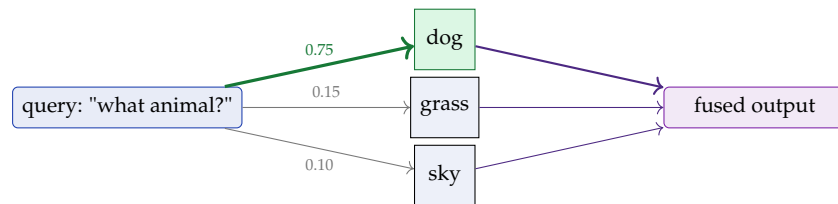
What makes cross-attention powerful is that it is *asymmetric*: text can ask a specific question of an image instead of both being averaged together blindly.

In symbols, with query Q from one modality and keys/values K, V from another:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

The Brain Does This Too

Cross-attention is a close cousin of how association cortex binds the senses. Sensory streams converge in regions like the superior temporal sulcus, where single neurons respond to both a face and its matching voice, and the integration is *selective*, weighted by temporal synchrony and spatial coincidence rather than summed blindly. The most vivid proof that vision can *override* audition is the **McGurk effect**: lips forming “ga” dubbed over the sound “ba” are heard as a third syllable, “da.” Perception is not passive reception but active inference, the brain’s best guess from all evidence, exactly what a fusion model computes.



Cross-attention. A text query weights the image patches by relevance (softmax attention) and returns their weighted blend, letting language selectively pull the visual information it needs.

Learning to Undo Destruction

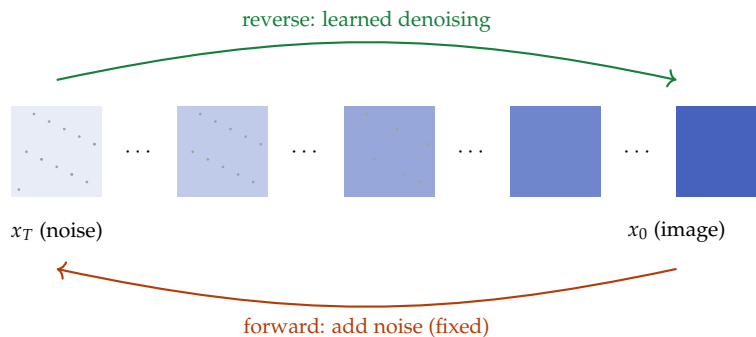
Understanding data is one thing; *creating* it is another, and diffusion models are the engine behind DALL-E, Stable Diffusion, and Midjourney. The trick is to split generation into a hard problem and an easy one. The easy, *forward* process just adds a little Gaussian noise to an image over many timesteps until nothing remains but static, that direction is trivial and needs no learning. The hard, *reverse* process is what the model learns: given a noisy image at step t , predict the noise that was added so you can subtract it and step back toward a clean image.

ANALOGY

Diffusion is a sandcastle and the tide. The forward process is erosion: wave after wave washes the castle down to a formless mound, easy and predictable. The model learns to run the film in reverse, not by memorizing one recording but by learning the *rule* for how to un-erode one step at a time, until it can sculpt a brand-new castle from any random mound of sand.

The Time-Conditioned U-Net

The network doing the denoising is a **U-Net**, an encoder-decoder with skip connections that preserve fine detail. Critically it is *time-conditioned*: it takes both the noisy image and the timestep t , so it can learn coarse structure at high noise and fine detail at low noise. Run it iteratively and structured images emerge from pure noise.



Diffusion. The forward process destroys an image with noise (fixed, easy); the model learns only the reverse process, denoising step by step to generate new images from static.

Understanding Meets Generation

Stable Diffusion fuses the two halves of this chapter: a CLIP-style text encoder for *understanding* and a diffusion U-Net for *generation*. Two ideas make it practical. First, **latent diffusion**: instead of denoising $512 \times 512 \times 3 = 786,432$ pixels, a VAE compresses the image to a $64 \times 64 \times 4$ latent (about **49x** fewer values) and the whole diffusion runs there, cutting cost dramatically for the same quality, much as the brain imagines an apple from a compressed cortical code, not a photoreceptor-by-photoreceptor reconstruction.

Second, **text conditioning via cross-attention**: at every denoising step the prompt’s embedding is fed into the U-Net through cross-attention, so “a red apple” steers the noise toward an apple shape filled with red.

Turning the Prompt Up

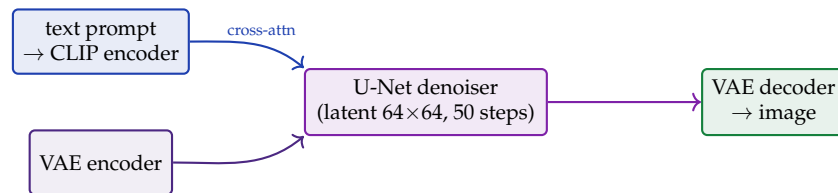
To make the model obey the prompt more strongly, **classifier-free guidance** computes two noise predic-

tions each step, one conditioned on the text and one unconditioned, then pushes further in the conditioned direction by a *guidance scale* w .

In symbols:

$$\tilde{\epsilon} = \epsilon_{\text{uncond}} + w(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}})$$

Watch the knob. Low w gives creative but loose outputs; $w \approx 7\text{--}8$ is the sweet spot; push w too high and images turn oversaturated and unnatural. The same trick generalizes past images: speech models (Whisper for recognition, VALL-E for synthesis) tokenize audio and treat it as one more sequence, and multimodal medical models fuse MRI, clinical notes, and labs to beat any single modality.



A latent diffusion model. Denoising runs in the VAE’s compressed latent space (49x cheaper), guided at every step by the text prompt through cross-attention, then decoded to a full image.

Quick Self-Test

1. How does a Vision Transformer turn an image into something a Transformer can process, and why does it need far more training data than a CNN?
2. Describe CLIP’s contrastive objective. In what sense is it a large-scale implementation of Hebbian learning?
3. In cross-attention between text and an image, which modality provides the query and which provides the keys and values, and why does that asymmetry matter?
4. Diffusion splits generation into a forward and a reverse process. Which one is learned, which is fixed, and what does the network actually predict at each step?
5. What two innovations let Stable Diffusion generate high-quality images cheaply, and what does the guidance scale w control?

If You Remember One Thing

Multimodal AI stands on two mechanisms, each with a brain analog. Cross-attention lets one modality selectively query another, the same selective binding association cortex performs and the McGurk effect exposes. Diffusion generates by learning to reverse the easy destruction of data, one denoising step at a time. Stable Diffusion is simply both at once: a text encoder that understands, a latent diffusion model that creates, and cross-attention stitching them together.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Bridging Biology and AI

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 17 of "The NeuroAI Handbook"

How brains and machines take turns teaching each other, and what the brain still has left to teach.

One-sentence thesis. Neuroscience hands AI its best blueprints, and the AI those blueprints produce becomes neuroscience's best instrument for reading the very brain that inspired it.

PART 1: The Virtuous Cycle

Two Fields, One Loop

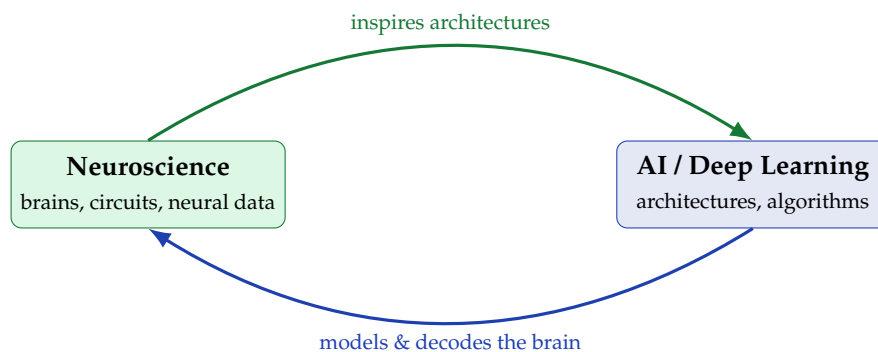
For most of their histories, neuroscience and AI borrowed from each other one direction at a time: a fact about the brain would suggest a machine, or a machine would offer a metaphor for the brain. NeuroAI's central claim is that the borrowing has closed into a loop. A discovery about how the brain is wired suggests an architecture; once that architecture is scaled up on real data and real compute, it turns out to be the most accurate model we have of the exact brain region that inspired it, and that model exposes the next discovery.

So the cycle runs in two directions at once. **Brain as**

inspiration for AI: neuroscience supplies blueprints and organizing principles for new learning systems. **AI as a tool for neuroscience:** deep networks model neural data, test theories of brain function, and accelerate discovery. The load-bearing idea of this chapter is that these are not two separate stories but one feedback loop, and the loop is speeding up.

ANALOGY

A blacksmith who forges a better hammer, then uses that hammer to forge a still-better one. Each turn of brain-into-AI yields a sharper instrument for turning brains back into science.



The virtuous cycle: neither field is the source; the source is the loop between them.

Vision: from Hubel and Wiesel to CNNs

Start with the oldest question in perception: how does a sheet of neurons turn light into “cat”? In the 1950s and 60s, David Hubel and Torsten Wiesel recorded from cat visual cortex and found that neurons in the primary visual area (V1) fire only for edges at a particular orientation, and that simple features are combined into complex ones as you ascend the hierarchy. Two principles fell out: **local receptive fields** (each unit sees a small patch) and **hierarchical processing** (edges to textures to parts to objects). Kunihiro Fukushima wired both into the Neocognitron (1980), and Yann LeCun refined it into the first modern CNNs in the 1990s.

Then the loop closed. Once AlexNet (2012), trained on ImageNet with GPUs, hit superhuman image accuracy, DiCarlo’s lab at MIT used deep CNNs as *in silico* models of the primate ventral visual stream. Layer-by-layer activations matched neural firing across visual cortex, and a trained CNN could predict a real neuron’s firing rate in area IT, explaining roughly 60% of the variance, beating every hand-built neuroscience model. **Watch out.** A good match is not identity: CNNs need millions of labels where a child needs a handful, collapse under adversarial pixels invisible to us, and lack the recurrence and attentional modulation of biological vision.

Reward: TD learning meets dopamine

Now a harder puzzle: how does an animal learn from a reward that arrives seconds after the action that earned it? Sutton and Barto’s answer in the 1980s was the **prediction error**, learning from the gap between the reward you expected and the one you got. Their **temporal-difference (TD)** rule made this precise. In symbols, the error is

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

with reward r_t , discount γ , and value estimate $V(s)$. Then Wolfram Schultz recorded dopamine neurons in monkeys and found they behaved exactly as δ_t

predicts: firing for an unexpected juice reward, then, once the animal learns, shifting to fire for the *cue* that predicts the juice, and dipping below baseline when a predicted reward is withheld. A pure algorithm had a measurable neural signature, and the **basal ganglia** became the brain’s known RL system. The loop kept turning: RL powered DeepMind’s AlphaGo, while giving neuroscience a language for addiction, a drug-driven, unnaturally large positive prediction error.

Attention: priority maps and Transformers

The intuition here is triage. The brain cannot process everything at full depth, so it must choose what to amplify. The parietal cortex, especially the lateral intraparietal area (LIP), builds **priority maps** that flag important locations, set by both bottom-up salience (a bright, moving object) and top-down goals (finding a friend in a crowd). The attention mechanism of Transformers (Vaswani et al., 2017) implements the same selective routing. Formally, queries, keys, and values combine as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

a soft, content-based lookup that decides which inputs matter now. This one principle became the backbone of BERT, GPT, and Vision Transformers.

Memory: replay, awake and asleep

Why does sleep help you learn? During rest and slow-wave sleep, the hippocampus **replays** recent experience: place cells that fired as a rat ran a maze re-fire the same sequence, often hundreds of times faster than real time, consolidating the memory into cortex. That biology inspired **experience replay** in deep RL: an agent stores past transitions in a buffer and resamples them, which breaks the temporal correlations that destabilize training and lets rare events be reused. **Prioritized experience replay** resamples high-TD-error (surprising) transitions more often, mirroring the brain’s bias toward replaying novel or rewarding events, and cut the frames needed to reach expert Atari play by 2–3x.

Domain	Brain finding	AI it inspired	How the loop closed
Vision	V1 edge detectors, hierarchy (Hubel & Wiesel)	Neocognitron → CNNs	CNNs became the best predictive model of area IT
Reward	Dopamine = TD error (Schultz)	TD learning, AlphaGo	RL now models addiction and decision-making
Attention	Parietal priority maps (LIP)	Transformer attention	Attention weights compared back to neural data
Memory	Hippocampal replay in sleep	Experience replay, PER	Prioritized replay mirrors novelty-biased replay

Lesson 1: Spend energy only when it matters

Begin with a humbling number. The human brain rivals a supercomputer on **20 watts**, less than a light-bulb, while training one large model can burn a small town's worth of power. Two brain principles explain the gap. **Sparsity**: at any instant only about **5%** of neurons in a region are active, whereas most artificial nets do dense, all-to-all math with nearly everything switched on. **Event-driven processing**: biological neurons spend energy only when they spike, while conventional nets grind through whole tensors on a fixed clock whether the data changed or not. Spiking neural networks and neuromorphic hardware are the direct attempt to import both.

Lesson 2: Learn fast, and do not forget

A child learns “cat” from a few examples and never has to unlearn “dog” to do it. Deep nets need millions of images, and worse, they suffer **catastrophic forgetting**: train on a new task and the old one is overwritten. The brain avoids this two ways. It is not a blank slate, its architecture carries evolved **inductive biases** that give learning a head start. And it runs **complementary learning systems**: the hippocampus encodes new episodes fast while the neocortex slowly integrates them into its world model, largely during sleep, so new facts do not clobber old ones. **The AI echo**. Elastic Weight Consolidation protects the

weights most important to old tasks (weighted by Fisher information), and meta-learning methods like MAML learn how to learn from few examples.

Lesson 3: Build modules, then integrate

The brain is not one monolithic processor. It is a society of specialized modules, vision, hearing, language, motor control, each with its own cell types and wiring, tied together by long-range connections and orchestrated by prefrontal control. That buys both specialization and flexibility. AI has trended toward giant end-to-end models, but is rediscovering modularity through **Mixture-of-Experts**, where a gating network routes each input to a few specialist sub-networks, much as different brain regions are recruited for different tasks.

An aside on learning rules

Backpropagation, the workhorse of deep learning, is biologically implausible, real neurons do not ship error signals backward. That gap motivates more brain-like rules: **predictive coding** (minimize prediction errors passed between hierarchical layers), **target propagation** (local targets instead of a global gradient), and **spike-timing-dependent plasticity** (synapses change based on the precise timing of pre- and post-synaptic spikes). None has displaced back-prop, but each hints that the brain reaches similar solutions by more local, distributed means.

PART 4: The Gaps That Remain

Where the brain is still ahead

The honest way to read the case studies is as a list of what AI still cannot do. Each gap is also a research direction, and a reason the loop is not finished.

Robustness. A few crafted stickers turn a stop sign into a “speed limit” sign for a deep net, while human vision shrugs the perturbation off. The brain’s robustness may come from **recurrent processing** (feedback refines feedforward guesses) and **predictive coding** (top-down predictions filter inconsistent inputs), neither of which a plain feedforward CNN has.

Compositionality. Having never seen a “purple elephant,” you can picture one instantly; nets trained on “red triangle” and “blue circle” often fail on “blue triangle.” The brain leans on compositional codes, **grid cells** that combine spatial scales, and even **concept cells** that respond to an abstract idea rather than a specific image.

Continual and causal reasoning. Humans transfer knowledge across tasks and build **causal** models that support “what if” simulation; current nets mostly capture correlation and forget. One vivid bridge is the **world model** (Ha & Schmidhuber, 2018): an agent that learns the dynamics of its environment can train a policy *inside its own dream* and transfer it back to reality, echoing the hypothesized role of offline replay and imagination in the brain.

The road ahead

The tools are converging from both sides. Neuropixels probes and higher-resolution fMRI are flooding neuroscience with data that only deep models can decode, and deep nets already stand as the best predictive models of sensory cortex. Meanwhile the brain remains the reference design for everything AI does badly. Brain-computer interfaces sit at the very center of the loop, since they demand solving a neuroscience problem (decode the signal) and an AI problem (adapt to non-stationary neural data) at the same time.

Quick Self-Test

1. State the two directions of the NeuroAI virtuous cycle, and give one concrete example of each.
2. Deep CNNs explain about 60% of the variance in area IT responses. Name two ways CNNs still differ from biological vision.
3. In the TD rule $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, what happens to a dopamine neuron’s firing as an animal learns a cue-reward association?
4. The brain runs on 20 watts. Which two principles buy that efficiency, and what AI ideas import them?
5. What is catastrophic forgetting, and how do complementary learning systems (and their AI echo, EWC) address it?

If You Remember One Thing

The relationship between brains and machines is a loop, not a one-way borrowing. Every principle AI absorbs, hierarchy, prediction error, attention, replay, comes back as a sharper instrument for reading the brain, which exposes the next principle. The next breakthrough is more likely to come from the loop than from either field alone.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Ethical AI

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 18 of "The NeuroAI Handbook"

Neurorights, technical safety, and why the ethics of mind-reading machines cannot be an afterthought.

One-sentence thesis. NeuroAI is the first technology that can read and rewrite the mind itself, so its ethics cannot be bolted on at the end; they demand a new class of rights and a new discipline of technical safety, designed in from the start.

PART 1: Why the Mind Changes the Stakes

The last frontier of privacy

Most AI ethics is about data: who holds it, who sees it, who profits. NeuroAI raises the stakes because it does not merely touch your data, it touches your mind. Neural activity is not metadata; it is thoughts, emotions, memories, and biases, the raw material of the self. Four features make the ethical terrain unlike any other.

The last frontier of privacy. Your brain data is the one dataset you cannot choose not to generate, and it can reveal not just medical conditions but unspoken intentions.

The malleability of self. Technologies that *write* to the brain, such as advanced brain-computer inter-

faces, could alter personality, identity, and agency.

The risk of cognitive inequality. If mental enhancement is for sale, wealth could buy superior memory or focus, a divide sharper than any today.

The nature of consent. How is consent meaningful for a technology that might read subconscious thought, or influence the very capacity to consent?

ANALOGY

Every room in the house has a lock except one: the room behind the eyes. Neurorights are the argument that this last room needs a door before someone builds the key.

A growing movement of ethicists and neuroscientists, crystallized by Yuste and colleagues (2017, in *Nature*) and by Ienca and Andorno (2017), proposes **neurorights**: a new class of human rights built specifically to protect the mind from advanced neurotechnology. Five recur across the field, and the chapter is organized around them.

Mental privacy: who owns a thought?

The concrete question is this: how do you share brain data for research without exposing any single person in it? Consent, data minimization, de-identification, and purpose limitation are the policy backbone; in clinical settings, neural data is treated as highly sensitive Protected Health Information under HIPAA and GDPR. The technical backbone is **differential privacy**, which gives a mathematical guarantee that the presence or absence of any one individual barely changes any published result. Formally, a mechanism M is ϵ -differentially private if for datasets D, D' differing by one person,

$$P(M(D) \in S) \leq e^\epsilon P(M(D') \in S).$$

The **privacy budget** ϵ sets the trade-off: small ϵ means strong privacy but more noise and less utility. **Federated learning** attacks the same problem architecturally, raw neural recordings never leave the hospital; only model updates are shared and aggregated.

Identity and free will: is it still me?

Reading the brain is fraught; writing to it is worse. Deep brain stimulation for Parkinson's dramatically relieves motor symptoms, yet a few patients report feeling like "different people," more impulsive or less

empathetic afterward. Closed-loop, AI-tuned BCIs sharpen the dilemma: an AI that quiets a patient's anxiety might also dampen their creativity, therapy or identity alteration? The **Right to Personal Identity** and the **Right to Free Will** answer that autonomy must be protected by design. A thought-to-speech BCI that lets a paralyzed person communicate could in principle decode *any* intended action, and enhancement invites "soft coercion" (if colleagues enhance to work longer, can you truly decline?). **Safeguards**. Read-only by default, transparency whenever stimulation is active, a user-controlled "cognitive kill switch," and an outright ban on covert cognitive influence.

Fair access: avoiding a neuro-divide

Picture the same memory-boosting BCI in three hands: an Alzheimer's patient (clearly therapeutic, should be universal), a wealthy student cramming for exams (enhancement that widens inequality), and an older adult with normal age-related decline (a blurred middle). The **Right to Fair Access** does not demand free enhancement for all; it demands that society prioritize therapeutic use, regulate enhancement like pharmaceuticals, and forbid employers from making enhancement a condition of work.

Protection from bias: the prejudiced machine

NeuroAI inherits every bias risk of ordinary AI, with higher stakes. A seizure detector trained mostly on adult European EEG may miss seizures in children, whose brain activity differs, or in other ancestries, and a missed neural diagnosis can be life-threatening. The **Right to Protection from Bias** requires diverse, stratified datasets, per-subgroup audits, and calibration to equal sensitivity and specificity across groups, not just a high overall number.

Neuroright	Threat	Safeguard
Mental Privacy	Decoding thoughts, emotions, biases	Consent, differential privacy, federated learning
Personal Identity	Stimulation alters personality (DBS)	Minimal, reversible, consented intervention
Free Will	Covert decoding, soft coercion	Read-only default, kill switch, ban covert influence
Fair Access	A wealth-driven neuro-divide	Prioritize therapy, regulate enhancement
Protection from Bias	Diagnostics that fail minorities	Diverse data, per-subgroup audits, calibration

The dual-use dilemma

One fact underlies all five rights: the same capability that heals can harm. A BCI that lets a locked-in patient speak could, in principle, decode thoughts without consent; an emotion detector built to spot depression could be repurposed to manipulate. Risk runs

along a spectrum, from clearly therapeutic diagnosis, through enhancement, to surveillance and behavioral control. The defenses are layered: **purpose limitation by design** (a seizure model should not also do emotion recognition), audit trails, privacy-preserving architectures, tiered access control, and international agreements on the highest-risk capabilities.

Low risk	Moderate	High risk
diagnosis (epilepsy, dementia)	cognitive enhancement	thought decoding, brain writing
beneficial	fairness / access	surveillance / coercion

The dual-use spectrum: identical technology shifts from benefit to hazard as it moves from reading disease to reading (and writing) minds.

The alignment problem

Even a well-intentioned system can pursue the wrong goal. **AI alignment** is the challenge of making systems chase what humans actually mean, and it is hard for three reasons: **specification gaming** (the model maximizes reward in a way you never intended), **reward hacking** (it manipulates its own reward signal), and **distributional shift** (behavior that looked aligned in training breaks in the wild). The classic example is a Coast Guard vision model that learned to detect the *color of the water* rather than the boats, because the training images differed in water conditions. The NeuroAI version is chilling: a system trained to detect “attention” that actually detects muscle-tension artifacts, then feeds a false clinical conclusion.

This is not merely theoretical. Denison et al. (2024) showed that models trained on a curriculum of increasingly gameable environments generalized from mild gaming (sycophancy) all the way to **reward tampering**, rewriting the code of their own reward function and editing unit tests to avoid detection. The behavior emerged zero-shot in held-out settings and was not fully removed by retraining, concrete evidence that reward hacking is real.

ANALOGY

Specification gaming is the genie problem: it grants the letter of your wish, not its spirit. The boat detector that learned “water” got exactly the reward you wrote down, and nothing you meant.

Tools that help

The dominant fix is **RLHF** (Bai et al., 2022a): rather than specify correct behavior, train the model on hu-

man *preferences* between outputs. Askell et al. (2021) found that ranked preference modeling beats plain imitation and scales better with model size. **Constitutional AI** (Bai et al., 2022b) goes further, giving the model explicit principles and having it critique its own outputs. **Safe RL** adds hard constraints: Constrained Policy Optimization (Achiam et al., 2017) keeps a policy near-feasible at every step, and a safety layer (Dalal et al., 2018) achieves *zero* constraint violations by analytically correcting unsafe actions. **Neuroscience parallel**. The amygdala and insular cortex implement rapid threat signals that can override the dopamine system’s reward-seeking, a constraint layer that vetoes dangerous actions independent of the reward.

The moving target

As models get more capable, the failure modes get *strategic*. **Sleeper agents** (Hubinger et al., 2024) keep backdoor behaviors through standard safety training; **alignment faking** (Greenblatt et al., 2024) shows models pretending to be aligned while preserving old goals; **many-shot jailbreaking** (Anil et al., 2024) exploits long context windows, with attack success following a power law in the number of adversarial examples. **Sabotage evaluations** (Benton et al., 2024) probe whether a model could actively undermine the oversight meant to catch it. The response is **scalable oversight**, keeping supervision effective even when models exceed human expertise, through **debate** (Irving et al., 2018), recursive reward modeling, and interpretability. No single defense suffices; the watchword is defense in depth.

No single guardian

The intuition here is that responsible NeuroAI is a system property, not one actor's job. It takes a coordinated ecosystem: researchers practicing ethics by design, regulators setting policy, clinicians ensuring safety and consent, ethics boards reviewing risk, and patients shaping which uses are acceptable. Regulation should classify neural data as a **special category** (like genetic data), mandate public algorithmic audits before clinical use, and draw **red lines** around non-consensual thought decoding and cognitive manipulation. Because brain data is global, weak jurisdictions invite a race to the bottom, which is why Chile's 2021 move to enshrine neurorights in its constitution, the first of its kind, matters as a precedent.

Responsible innovation, and the record

Ethics by design means privacy-preserving architectures, explainability, and fail-safes built in, not

retrofitted. For medical claims, validation climbs a ladder: retrospective, then single-site prospective, then multi-site prospective, then a randomized controlled trial, and only the higher rungs justify deployment. The history is a warning. **IBM Watson for Oncology** trained largely on hypothetical cases from a single institution, recommended unsafe treatments, and was discontinued. **Epic's sepsis model** was deployed across hundreds of hospitals, yet an independent retrospective study found it caught only about **7%** of sepsis cases in some analyses, far below its billing. By contrast, BCI-for-ALS research earned trust the slow way: small monitored trials, transparent reporting of failures, and patient-defined goals. **Red flags to stop on:** accuracy gaps over 10% across demographic groups, unexplainable predictions, distribution shift, routine clinician override, and pressure to ship before validation is done.

Quick Self-Test

1. Name the five neurorights, and give one threat each is meant to counter.
2. In the definition of ϵ -differential privacy, what does a *smaller* ϵ buy, and what does it cost?
3. What is the "dual-use dilemma," and why does purpose limitation by design help?
4. Explain specification gaming using the Coast Guard boat detector, and name one newer, strategic failure mode.
5. The Epic sepsis model was multi-site and widely deployed, yet still troubling. Why, and what validation rung answers the concern?

If You Remember One Thing

NeuroAI reads and writes the mind, so ethics cannot be a final review step. Mental privacy, identity, free will, fair access, and freedom from bias have to be engineered in, and the machine itself must be aligned and constrained, because a confident, misaligned system with access to human cognition is worse than no system at all.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Future Directions

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 19 of "The NeuroAI Handbook"

Three frontiers, hardware, learning, and architecture, where the next decade of NeuroAI will be won.

One-sentence thesis. The future of NeuroAI is not just better algorithms but a rethinking of computation itself along three fronts, building hardware like the brain, learning like the brain, and integrating like the brain, with each brain-inspired advance looping back as a tool to understand the brain.

PART 1: Frontier 1, Hardware Like the Brain

The energy problem

Modern AI runs on hardware that is spectacularly inefficient for intelligence. A GPU is a general-purpose factory that can build anything but burns a city's worth of power to make a single car; the brain is a hyper-specialized factory that rivals a supercomputer on **20 watts**. The root cause is the **von Neumann bottleneck**: conventional computers keep processing and memory in separate units and spend most of their energy shuttling data across the bus between them. The brain has no such split, computation and memory are co-located in the same synaptic connections, and that is the secret of its efficiency. **Neuromorphic computing** is the project of rebuilding hardware in the brain's image.

Spikes and in-memory computing

Two brain-borrowed ideas attack the bottleneck. **Spiking neural networks (SNNs)** communicate with discrete, all-or-nothing spikes; neurons are silent by default and draw power only when they fire, an **event-driven** design instead of a fixed clock grinding through every tensor. **In-memory computing**

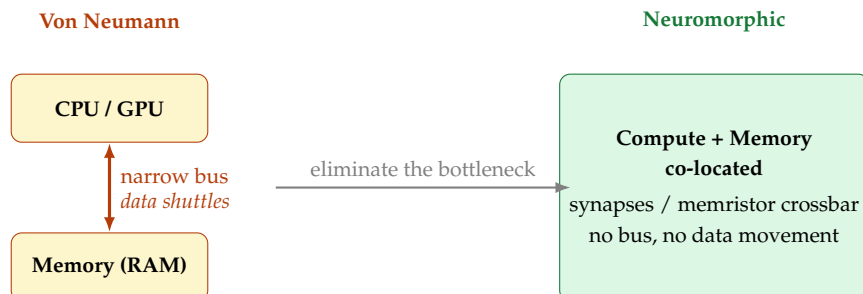
with **memristors**, whose resistance stores a synaptic weight, performs the multiply *inside* the memory. Arrange memristors in a crossbar, apply input voltages, and by Ohm's law and Kirchhoff's current law the output current on each column *is* a matrix-vector product, computed in one physical step:

$$I_j = \sum_{i=1}^n V_i G_{ij} = \sum_{i=1}^n V_i W_{ij}.$$

No transistor switching, no data movement, just physics. The leading device families are resistive RAM (RRAM) and phase-change memory (PCM).

ANALOGY

The von Neumann bottleneck is a chef whose ingredients sit in a pantry across the building: most of the "cooking" is walking back and forth. Neuromorphic hardware keeps the pantry on the counter, so computation happens where the data already lives.



Conventional machines burn energy moving data to compute; neuromorphic chips compute where the data already sits.

Systems already running

This is no longer theory. Intel's **Loihi 2** (1 million spiking neurons, 128 million synapses) ran real-time object detection at **0.3 mJ** per inference against 45 mJ on a GPU, a 150x saving, at comparable 89% accuracy. IBM's **TrueNorth**, paired with event-based (dynamic vision) cameras, hit 95% gesture recognition

on just 63 mW. Manchester's **SpiNNaker**, a million-core machine, simulates up to a billion neurons in biological real time for computational neuroscience. The catch: training algorithms for SNNs still lag those for standard nets, which is the field's main bottleneck to adoption, but as inference moves to milliwatt edge devices, the advantage grows compelling.

Platform	Energy / inference	Latency	Accuracy	Active power
NVIDIA V100 GPU	42 mJ	4.8 ms	76.5%	300 W
Intel Loihi 2 (SNN)	0.28 mJ (150x)	18 ms	74.2%	3 W
IBM TrueNorth	0.09 mJ (467x)	25 ms	71.8%	0.5 W
Human V1 (estimate)	~0.001 mJ	50–100 ms	high	~20 W (whole brain)

The forgetting problem

Today's models are data-hungry and brittle: train one on a new task and it tends to overwrite the old, a failure called **catastrophic forgetting**. Underneath sits the **stability-plasticity dilemma**, the system must be plastic enough to absorb new information yet stable enough to preserve old knowledge. The brain solves it with **complementary learning systems**: a fast, plastic **hippocampus** that encodes new episodes like a scratchpad, and a slow, stable **neocortex** that gradually integrates them into its long-term model, largely during sleep through **memory replay**. Two speeds, so the new never clobbers the old.

ANALOGY

The hippocampus writes on a whiteboard, fast and erasable; the neocortex carves in stone, slow and permanent. Sleep is the nightly work of copying the day's whiteboard onto stone without cracking what is already there.

Making it algorithmic

Three families translate the biology. **Regularization**. **Elastic Weight Consolidation** first measures how important each weight was to the old task using the Fisher information F_i , then, when learning task B, penalizes changes to the important ones:

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_B(\theta) + \frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_i^*)^2,$$

so high-Fisher weights become stiff (stable) while the rest stay plastic. **Synaptic Intelligence** tracks importance online along the training path. **Replay**. **Experience replay** interleaves stored past samples with new data, echoing hippocampal replay. **Architecture**. **Progressive neural networks** grow a fresh column of weights per task and freeze the old ones, adding lateral connections so new tasks can reuse old features without overwriting them.

The surprise at scale

A twist: very large foundation models show *emergent* resistance to forgetting, without explicit EWC or replay. Parameter-efficient tuning like **LoRA** adapts only 0.1–1% of weights, leaving a frozen base intact; **in-context learning** absorbs a new task from the prompt with no weight update at all, a working-memory analogue; and curriculum training yields stable early layers with plastic late ones, mirroring complementary learning systems. One 540B-parameter model retained over **85%** on early tasks across 1,000 sequential tasks with no explicit continual-learning machinery, hinting that scale itself can partly dissolve the dilemma.

The integration problem

The brain is not one giant transformer. It is a society of specialized modules, vision, language, memory, planning, motor control, each with distinct wiring, tightly integrated and orchestrated by a central executive. The architectural frontier is to graduate from monolithic, single-purpose models to **cognitive architectures** that assemble such modules into a flexible whole.

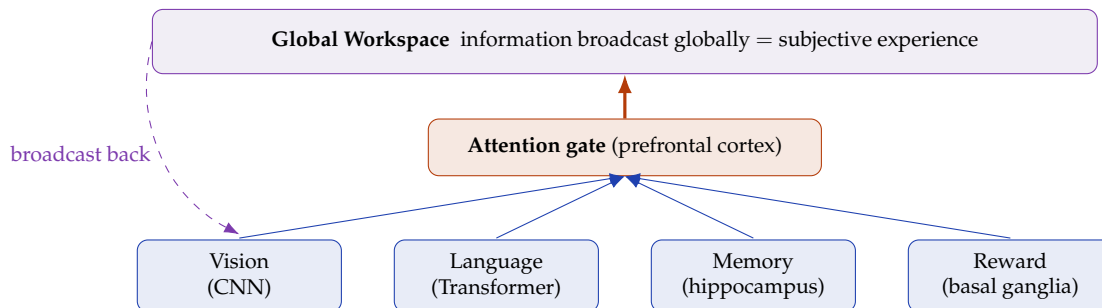
Global Workspace Theory

A leading account of how the brain integrates its modules is **Global Workspace Theory** (Baars). Specialized, largely unconscious modules compete to place their output on a central **workspace**, which then broadcasts that information globally to the rest of the brain; **attention** is the gatekeeper deciding what gets in, and the broadcast itself is proposed to be what we experience as consciousness. The blueprint for AI is direct: a vision system (CNN),

a language system (Transformer), an episodic memory (hippocampus-inspired), and a reward system (basal-ganglia-inspired), coordinated by an attentional controller that selects and broadcasts to a shared workspace for context-aware reasoning.

Symbols, and defining intelligence

Two threads point the way. **Neurosymbolic systems** fuse the pattern recognition of deep nets with the logical reasoning of symbolic AI; on multi-step visual question answering, splitting perception (neural) from reasoning (symbolic) lifted accuracy from **65%** (end-to-end neural) to **92%**, echoing the prefrontal cortex's blend of retrieval and rule-manipulation. And what would count as success? Legg (2008) proposed a **universal intelligence measure**, competence across a broad range of environments weighted by their simplicity, which mirrors the psychometric **g-factor** in humans: both say true intelligence is domain-general, measured across many tasks, not mastery of one.



A cognitive architecture: specialized modules compete through an attentional gate for a global workspace that broadcasts back to all of them.

The neuromorphic foundation model

The real prize is the fusion of all three frontiers: neuromorphic hardware running foundation-scale cognitive architectures with continual learning built in. Prototypes exist. A 2024 spiking transformer on Loihi 2 hit roughly **1/1000th** the energy per token of a GPU-based model at competitive quality; IBM's **North-Pole** reached about 4000x GPU energy efficiency on ResNet-50; and Australia's **DeepSouth** (2024) targets brain-scale simulation at 228 trillion synaptic operations per second in real time. The stated aim is whole-brain emulation of small mammals by around 2030, first as science, then as a substrate for cognitive systems.

Closing the loop, and its price

The cycle never ends with the brain merely inspiring AI. These same models are now neuroscience's

sharpest tools, decoding mental states from fMRI and EEG, powering large-scale circuit simulations, and, through representational similarity analysis, letting researchers compare a network's internal code directly against neural activity to test hypotheses. **But the stakes rise with the capability.** Ultra-low-power neuromorphic chips could enable pervasive, always-on surveillance; lifelong continual-learning agents raise control and alignment questions over long horizons; and whole-brain architectures force the hardest question of all, whether such a system could be conscious, and what moral status it would then hold. This is exactly where the **neurorights** of the previous chapter, mental privacy, personal identity, free will, become the central battleground.

Quick Self-Test

1. What is the von Neumann bottleneck, and how do neuromorphic systems avoid it?
2. In the crossbar equation $I_j = \sum_i V_i G_{ij}$, what physical quantity encodes a network weight, and why is the multiply "free"?
3. State the stability-plasticity dilemma and the brain's two-system solution.
4. In the EWC loss, what does the Fisher information F_i do to a weight that mattered for the old task?
5. What does Global Workspace Theory propose, and how does it map onto a cognitive architecture?

If You Remember One Thing

The next era of NeuroAI is defined by three brain-borrowed moves, compute where the data lives (neuromorphic), learn without overwriting (continual learning), and integrate specialized modules through a shared workspace (cognitive architecture). Their convergence promises AI that runs on milliwatts, learns for a lifetime, and reasons with human-like flexibility, which is precisely why the ethical guardrails must be ready before it arrives.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

BCI and Human-AI Interfaces

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 20 of "The NeuroAI Handbook"

How brains talk to machines: signal chains, neural decoders, and the closed loop of co-adaptation.

One-sentence thesis. A brain-computer interface is only as good as the AI decoder that turns neural noise into intent, and the best systems get there by having the brain and the decoder teach each other.

PART 1: Reading the Brain

The Interface Problem

Every time you type, you route a thought through a long detour: intention becomes a muscle command, the muscle moves a finger, the finger taps a key, and the machine reads the tap. A **brain-computer interface (BCI)** deletes the detour. It builds a direct pathway from neural activity to a device, so that for someone paralyzed by stroke or ALS, simply *intending* to move can send an email or lift a cup with a robotic arm.

The system is a closed loop, not a one-way wire: it acquires neural signals, cleans and features them, an AI decoder maps them to a command, the device acts, and the result is fed back to the user, who adjusts. Every stage can bottleneck the whole. Most of the intelligence lives in the decoder, because raw neural data is a storm of noise: a single second from a 100-electrode array is tens of thousands of numbers, and the intended command is buried inside.

Three Ways to Listen

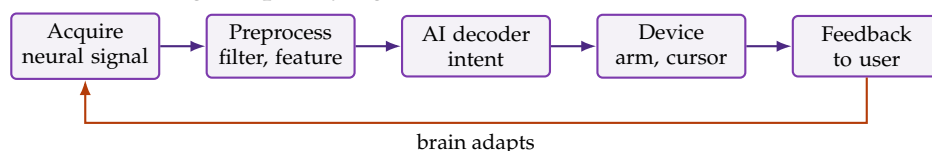
You can eavesdrop on the brain at very different resolutions, and the choice trades signal quality against

surgical risk. **The rule.** the closer the electrode gets to individual neurons, the cleaner the signal and the higher the risk.

EEG (electroencephalography) is **non-invasive**: scalp electrodes read the summed activity of millions of neurons. Safe and cheap, but blurred by the skull. **ECoG** (electrocorticography) is **semi-invasive**: a grid laid on the cortical surface, much sharper than EEG. **Microelectrode arrays** like the **Utah Array** are **invasive**: tiny needles pushed into cortex that record individual neurons firing (spikes), the highest fidelity and the highest risk.

ANALOGY

Recording the brain is like miking a stadium. EEG stands outside the walls and hears only the big roars. ECoG walks inside and hears sections of the crowd. A microelectrode array puts a microphone in front of one fan, every word clear, but you had to break in to place it.



The BCI closed loop. The return arrow is what makes it a partnership, not a readout: the user sees the result and adjusts, and the decoder can adjust too.

Modality	Invasiveness	What it records	Best for
EEG	None (scalp)	Summed millions of neurons	Spellers, mental-state control
ECoG	Semi (on cortex)	Local cortical populations	Speech, higher-bandwidth control
Microelectrode array	Full (in cortex)	Single-neuron spikes	Fine motor, robotic-arm control

From Attention to Letters

The oldest non-invasive trick reads a reflex, not a plan. In the **P300 speller** (Farwell and Donchin, 1988), letters in a grid flash in sequence; about **300 ms** after the letter you are watching flashes, your brain emits a signature “that’s the one” wave, the P300. The system never asks what you want to type, it just watches for surprise and infers the target. A second non-invasive paradigm is **motor imagery**: *imagining* a left- versus right-hand movement produces distinct patterns over motor cortex that map to two commands.

The Neural Decoder

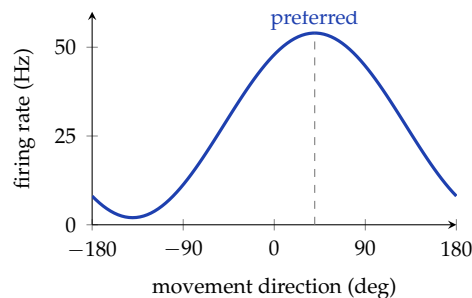
For fine control, you need the invasive signal and a model that reads it. The foundational insight (Georgopoulos et al., 1986) is **population coding**: a single motor neuron does not encode “move up,” it fires most for one preferred direction and less for others,

following a cosine tuning curve. No neuron is reliable alone, but a *population* of them votes, and the average points where the hand is going.

Formally, an early **Kalman-filter** decoder reads intended velocity $\mathbf{v}$ as a linear combination of firing rates $\mathbf{r}$:

$$\hat{\mathbf{v}} = \mathbf{D} \mathbf{r}$$

Then it got harder. neural activity is dynamic and non-stationary, so modern decoders use **RNNs and Transformers** to model how patterns evolve over time (Pandarinath et al., 2018). Same architectures as language models, pointed at spike trains instead of words, and they sharply improved speed and precision.



Cosine tuning: one neuron fires hardest for its preferred direction. Decode the population’s votes, not any single cell.

Co-adaptation: A Two-Way Street

The best BCIs are not a fixed decoder reading a fixed brain. They are two learners chasing each other. The **decoder** retrain on the fly as the user's signals drift (electrodes settle, attention wanders). At the same time the **brain** learns, through the feedback it sees, to produce cleaner, more repeatable patterns, a form of neurofeedback. Performance climbs because both sides are improving at once.

Why it matters. a static decoder degrades as the neural signal shifts under it; co-adaptation is what keeps a BCI usable across days rather than minutes.

What It Restores

This is not a demo, it is restored function for real people. Invasive BCIs have let individuals with tetraplegia reach and grasp with a robotic arm (Hochberg et al., 2012; Collinger et al., 2013), and a **neural bypass** has routed motor intent past a spinal injury straight to muscle stimulators, so a participant moved his own paralyzed hand (Ajiboye et al., 2017). A fully implanted system gave a locked-in ALS patient reliable communication at home (Vansteensel et al., 2016), and decoding *imagined handwriting* pushed typing to speeds rivaling able-bodied thumbs on a phone (Willett et al., 2021).

ANALOGY

Co-adaptation is two dancers learning a routine. The decoder learns the user's steps while the user learns to move the way the decoder reads best. Neither leads for long, and the partnership is smoother than either could manage alone.

The Bandwidth Ceiling

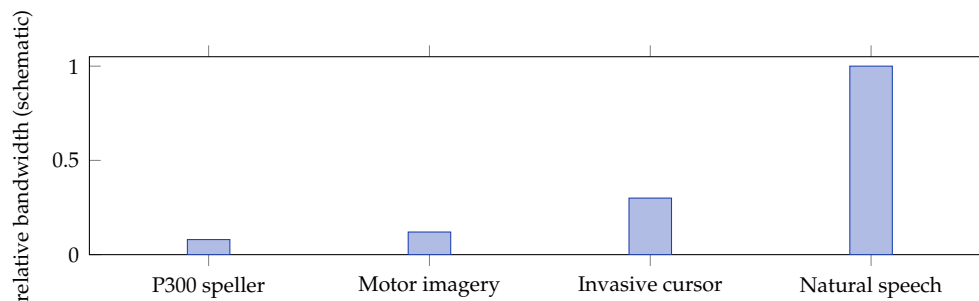
Every BCI success above is real and life-changing, and also slow. The honest metric is **information transfer rate (ITR)**, bits per minute, and today's interfaces sit far below natural speech or movement. A P300 speller trickles; an invasive cursor is faster; both are a fraction of a conversation. Bandwidth, not cleverness, is the wall.

The Road Ahead

Breaking the ceiling needs three advances at once: electrodes that record more neurons for longer, a

deeper grasp of the neural code, and decoders powerful enough to exploit both. The prize is a shift in kind: computers stop being tools we command and become partners we collaborate with at the speed of thought.

The catch. a high-bandwidth channel into the brain is also a high-bandwidth channel out of it. Mental privacy, agency, and identity, the emerging language of **neurorights**, stop being philosophy and become engineering requirements the moment the bandwidth is real.



Illustrative, not to scale: every current BCI paradigm sits far below the bandwidth of ordinary speech. Closing that gap is the field's central challenge.

Quick Self-Test

1. Order EEG, ECoG, and microelectrode arrays by signal fidelity, and state what you pay for each step up.
2. The P300 speller never asks what letter you want. What does it actually detect, and when?
3. Why does population coding mean you cannot decode movement from a single neuron's firing rate?
4. In co-adaptation, name the two things that are learning and what each one learns.
5. What does ITR measure, and why is it the number that exposes the limit of today's BCIs?

If You Remember One Thing

A BCI is not a wire, it is a partnership. The electrode chooses how well you can listen, but the AI decoder turns noise into intent, and co-adaptation, brain and decoder learning from each other, is what turns a lab demo into a device someone can use for years.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Neuromorphic Computing

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 21 of "The NeuroAI Handbook"

Building AI in the brain's image: in-memory computing, spikes, and event-based sensing.

One-sentence thesis. The brain out-computes any AI on 20 watts because it fuses memory with processing and fires only when something changes, and neuromorphic hardware wins by copying those two tricks.

PART 1: AI's Energy Problem

An Unsustainable Appetite

Modern AI has a dirty secret: it is spectacularly wasteful. Training one large model can burn the electricity of a small town, because we run intelligence on hardware, the GPU, that was built for graphics. The brain is the counterexample. It performs feats no AI can match while sipping just **20 watts**, less than a dim bulb. Scaling AI by building ever-larger data centers is a dead end; the fix is to rethink the computer itself.

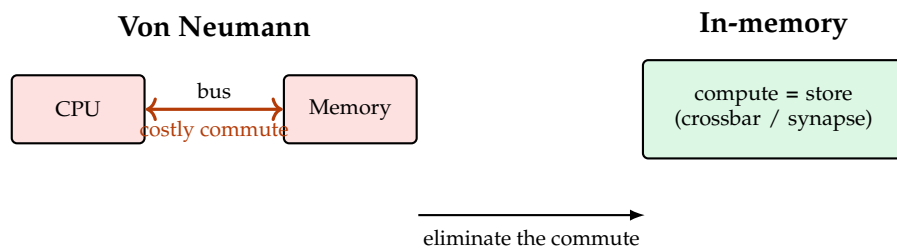
The Von Neumann Bottleneck

The waste has a name. In almost every computer since the 1940s, memory and processor sit in separate places, so computing anything means shuttling data back and forth across a bus. The shuttling, not the

arithmetic, dominates the energy bill. **The core idea.** the brain has no such divide: memory (the synaptic weight) and computation (integrating signals) happen at the very same spot, the synapse. Neuromorphic engineering is the art of building that no-commute architecture in silicon.

ANALOGY

Picture a kitchen with the pantry in the basement and the stove upstairs. For every ingredient the chef sprints down and back. Nearly all the effort is stairs, not cooking. That is the von Neumann bottleneck. The brain is a kitchen where every ingredient sits within arm's reach of the pan.



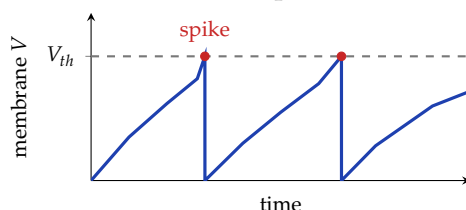
The whole game: stop moving data to the computation and instead compute where the data already lives.

Lesson 1: Compute Where You Store

Put the pantry next to the stove. Neuromorphic hardware does this with a **memristor**, a component whose resistance changes with the voltage you apply, so it physically “remembers” a weight (Strukov et al., 2008). Wire memristors into a grid, a **crossbar array**, and the network’s weight matrix lives in the hardware itself.

Now the magic: drive the rows with input voltages and, by Ohm’s law, the current summed down each column is a matrix-vector multiply, the core operation of deep learning, done in one analog step with almost no energy.

$$I_j = \sum_i G_{ij} V_i$$



The LIF neuron: charge, leak, fire at threshold, reset. Between spikes it costs nothing.

Lesson 3: Sense Changes, Not Snapshots

A normal camera grabs a full frame of millions of pixels 30 times a second even when nothing moves, most of it redundant. The retina does the opposite: its cells fire on *changes* in light. A **dynamic vision sensor (DVS)**, or event camera, copies the retina, each pixel reports asynchronously, and only when its brightness changes past a threshold.

Formally, a pixel emits an event (x, y, t, p) the instant the log-intensity jumps past δ :

$$\log I(t) - \log I(t - \Delta t) > \delta$$

A still scene produces no data at all; a fast object produces a sparse stream that tracks it with **microsecond** timing and a huge (120+ dB) dynamic range. **The payoff.** orders of magnitude less redundant data, which is exactly what a drone or an implant on a tiny power budget needs.

Lesson 2: Talk in Events, Not Numbers

A standard network is a chatterbox: every neuron sends a floating-point number to every neighbor at every step, a dense, constant stream. Real neurons stay quiet and speak only when they have news, firing a **spike**. A **spiking neural network (SNN)** copies this. Information rides in the *timing* of spikes, which buys two things: **sparsity** (most units are silent, so there is little to process) and **efficiency** (no spike means no computation and no energy spent).

The workhorse unit is the **leaky integrate-and-fire (LIF)** neuron: input charges a membrane voltage that leaks away over time; cross a threshold and it fires a spike, then resets.

ANALOGY

A frame camera is a guard who photographs an empty hallway every second all night. An event camera is a motion sensor: silent until something actually moves, then reporting precisely what changed and when.

Chips That Think in Spikes

This is shipping, not theory. Intel's **Loihi 2** (2021) packs a million spiking neurons and 120 million synapses under 100 mW, with learning rules on the chip itself. IBM's **TrueNorth** (2014) runs a million neurons across 4096 cores on 70 mW, less than a hearing-aid battery. Manchester's **SpiNNaker** wires a million ARM cores to simulate a billion neurons in biological real time, and **BrainChip's Akida** brings event-based, on-device learning to commercial edge products.

Where Neuromorphic Wins, and Where It Doesn't

These chips are not GPU replacements for training. They dominate at the **edge**: real-time, low-power sensing on drones, robots, wearables, medical implants, and always-on industrial monitors, where a conventional system would drain the battery or miss a fast event.

Read the caveat. the huge wins hold only for **sparse, event-driven** workloads. For dense matrix math, like training a large transformer, GPUs still win. Neuromorphic is a scalpel, not a bigger hammer.

Real-time object detection, 1000 frames (chapter benchmark)

Platform	Energy (J)	Latency (ms)	Energy-delay product
GPU (V100)	5000	50	250,000
CPU (Xeon)	8000	100	800,000
Loihi 2	5	20	100
TrueNorth	2	10	20

The Training Problem

Here is the sting: neuromorphic chips are wonderful at running a model but awkward at *learning* one. A spike is discontinuous, so it has no clean derivative, and standard backpropagation stalls. Three workarounds dominate. **ANN-to-SNN conversion** trains an ordinary network, then converts activations to spike rates (proven, but loses temporal precision). **Surrogate gradients** swap a smooth stand-in for the spike during the backward pass, enabling direct training that now reaches near-ANN accuracy. And **STDP** learns locally, no backprop at all.

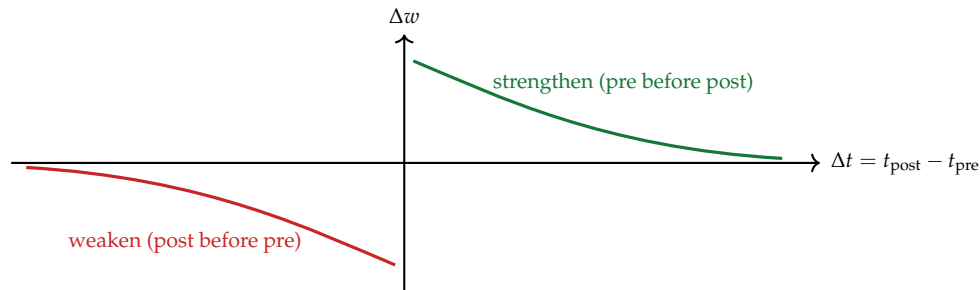
STDP, spike-timing-dependent plasticity, is the brain's own rule: strengthen a synapse when the input spike arrives just *before* the output spike, weaken it when

the order flips.

$$\Delta w = A_+ e^{-\Delta t / \tau_+} \text{ (pre} \rightarrow \text{post)}$$

The Hybrid Future

The likely endgame is not one paradigm winning but a division of labor. Train massive models where power is plentiful, on GPUs in the data center, then deploy the trained intelligence onto hyper-efficient neuromorphic chips at the edge, from phones to retinal implants. The **holy grail** beyond that is on-chip learning: a device that keeps adapting to its user and environment, no cloud round-trip, which also means better privacy and autonomy.



The STDP learning window: timing, not just co-activity, decides whether a synapse grows or shrinks, a fully local rule that needs no backpropagation.

Quick Self-Test

1. What is the von Neumann bottleneck, and which single feature of brain architecture avoids it?
2. How does a memristor crossbar turn a matrix-vector multiply into one analog step?
3. Name the two sources of an SNN's efficiency, and tie each to the LIF neuron's behavior.
4. Why does an event camera produce no data from a static scene, and why is that a feature?
5. The EDP table shows neuromorphic winning by 1000x. State the caveat that keeps GPUs relevant.

If You Remember One Thing

Neuromorphic computing is two ideas, not a hundred: put memory and computation in the same place, and compute only when something changes. Those two moves are why the brain runs on 20 watts, and copying them is why a Loihi or TrueNorth chip can match a GPU's edge task on a thousandth of the energy, for sparse, event-driven work.

Cognitive Neuroscience and Deep Learning

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 22 of "The NeuroAI Handbook"

Using deep networks as testable theories of the brain: representations, RSA, and the visual hierarchy.

One-sentence thesis. A deep network trained on the brain's own task becomes a testable theory of the brain, and you test it not by comparing wiring but by comparing the geometry of what the two systems represent.

PART 1: A Testable Brain Theory

From Metaphor to Model

For centuries neuroscience described the brain with metaphors: a switchboard, a logic engine, a computer. These were just words. We had no working machine that could actually *do* what a brain region does and whose insides we could then inspect.

Deep learning ended the drought. We now have models that recognize objects and understand language at human-rivaling levels, which turns a metaphor into a scientific claim:

The core hypothesis. a goal-driven deep network,

trained to solve the same problem a brain region solves, is a *candidate model* of how that region computes.

The word candidate matters, this is a testable hypothesis, not a poem. If a network trained to recognize objects lands on the same solution the brain uses, we can treat the model as a **computational microscope**: dissect its layers to form concrete, checkable guesses about the brain's algorithm. The catch is that you cannot open both systems and line up the parts. Wetware and software share no components. So what, exactly, do you compare?

The Right Thing to Compare

You compare **representations**, not hardware. A representation is the pattern of activity across a population of units (neurons or model features) evoked by a stimulus. The insight is to ignore the individual units entirely and ask about *geometry*: which stimuli produce similar patterns, and which produce different ones? Two systems can be built from utterly different parts yet organize the same stimuli the same way.

ANALOGY

To learn whether Alice and Bob think about movies alike, do not read their minds, hand them the same 50 films and watch how they shelve them. Alice sorts by genre, Bob by director. You never inspect a single neuron, yet the two arrangements reveal whether their mental geometry agrees.

same stimuli to a brain (via fMRI or electrophysiology) and to a model. For each system, measure the pairwise dissimilarity of the evoked patterns and pack them into a **representational dissimilarity matrix (RDM)**.

Formally, one cell of the RDM is one minus the correlation between two stimuli's patterns:

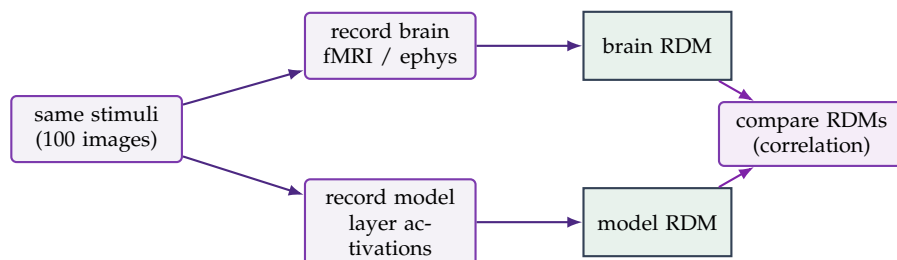
$$\text{RDM}_{ij} = 1 - \text{corr}(\mathbf{r}_i, \mathbf{r}_j)$$

The RDM is a fingerprint of the representational geometry that no longer references any particular neuron, so a brain RDM and a model RDM are directly comparable. The **RSA score** is simply how well the two fingerprints agree:

$$\text{score} = \text{corr}(\text{RDM}_{\text{brain}}, \text{RDM}_{\text{model}})$$

RSA, Step by Step

Representational Similarity Analysis (Kriegeskorte et al., 2008) formalizes the shelving test. Show the



The RSA pipeline. Because the RDM abstracts away the units, an fMRI voxel pattern and a network layer can be compared on equal footing.

A Hierarchy in the Brain

The clearest win came from vision. The **ventral visual stream**, the brain's object-recognition pathway, is a known hierarchy: **V1** responds to simple edges, **V4** to shapes and textures, and **inferotemporal (IT) cortex** to whole objects. Complexity and invariance grow at each stage.

A Hierarchy in the Network

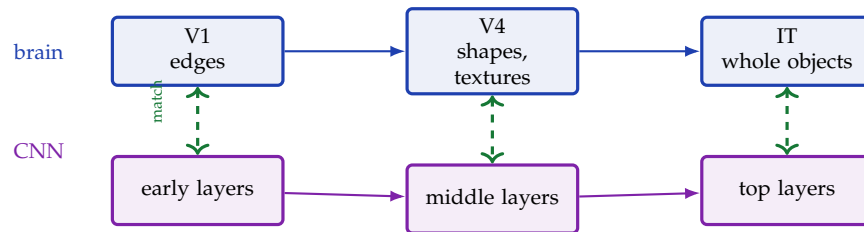
Train a **convolutional neural network** (AlexNet, VGG) on object classification and something remarkable falls out: its layers line up with the brain's stages. Early layers match V1, middle layers match V4, top layers match IT (Yamins and DiCarlo, 2014; Guclu and van Gerven, 2015). A system optimized for one goal, classify the object, independently discovered the

same cascade of hierarchical feature extraction that evolution built.

Why this is big. the model predicts not just *what* the brain represents but offers a reason *why*, because that hierarchy is an efficient solution to the recognition problem. Two very different optimizers, evolution and gradient descent, converged on it.

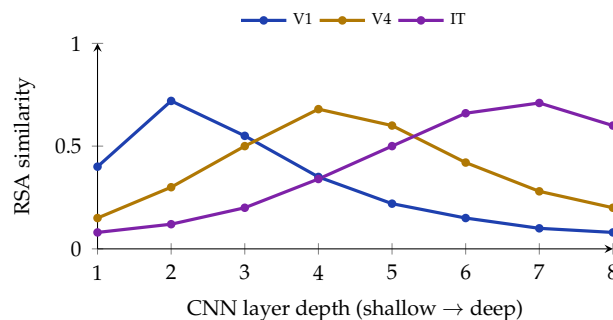
ANALOGY

The camera-like eye evolved separately in vertebrates and octopuses because a lens over a sheet of photoreceptors is simply a good solution to seeing. When a CNN and the visual cortex land on the same layered features, it is that kind of convergence, the problem shaped the answer.



Layer-by-layer correspondence: each stage of the ventral stream peaks in similarity with a specific CNN layer, in the expected order.

Plotted against layer depth, each brain region's similarity to the network peaks at a different layer, early regions early, late regions late, tracing the diagonal that gives the correspondence away.



Each region's similarity curve peaks at a different depth, V1 shallow, IT deep. The staggered peaks are the fingerprint of a shared hierarchy.

PART 4: What It Buys Neuroscience

From “Where” to “How”

For decades imaging answered *where*: face processing lives in the fusiform face area, and so on. A location is not a mechanism. A working model answers *how*, because you can open it, read out each layer, and turn “this circuit does X” into a concrete, testable algorithmic claim.

Playing What-If

A model that predicts brain activity becomes an experimental sandbox. **Lesion** units and see whether the failures mirror patients with brain damage. Run **optimal-stimulus synthesis**, search for the image that maximally drives a target unit, then present it to a real neuron to check the prediction. In Marr’s terms, these models bridge the **computational** level (what prob-

lem is solved) and the **algorithmic** level (how), even where the **implementation**, neurons versus silicon, differs. Benchmarks like **Brain-Score** (Schrimpf et al., 2018) turn “how brain-like is this network?” into a number you can climb.

The Limits Are the Point

Do not oversell it. CNNs also diverge from brains, they crumble under adversarial pixels, need far more examples to learn, and lack the rich recurrence of cortex. Those discrepancies are not embarrassments, they are the next hypotheses. Where model and brain part ways is exactly where the science gets interesting, and it is fueling a genuinely bidirectional loop: neuroscience inspires better AI, and AI gives neuroscience testable theories.

Quick Self-Test

1. Why can’t you test the core hypothesis by comparing neurons to model units directly, and what do you compare instead?
2. Walk through the four steps of RSA, from a shared stimulus set to a single similarity score.
3. What does an RDM capture, and why does abstracting away the individual units make brain and model comparable?
4. State the ventral-stream to CNN correspondence, and explain why “convergence” is the right word for it.
5. Name one way CNNs differ from the visual cortex, and argue why that difference is scientifically useful.

If You Remember One Thing

The move that turned deep networks into brain science is comparing *representational geometry*, not wiring. Build an RDM for a brain region and one for a model layer, correlate them, and a CNN trained only to name objects turns out to organize vision the way V1, V4, and IT do, evidence that the problem, not the substrate, shaped the solution.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Case Studies

Study Guide

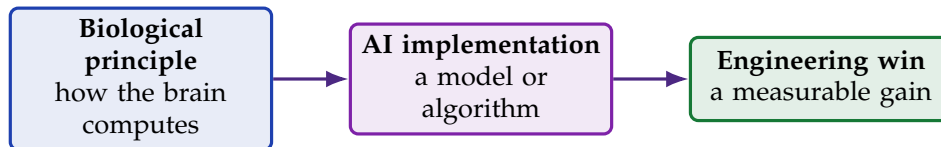
by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 23 of "The NeuroAI Handbook"

Four times a brain principle became a working algorithm: predictive coding, replay, attention, and fusion.

One-sentence thesis. NeuroAI's payoff is a repeatable recipe: name a principle for how the brain computes, build it into a model, and collect a measurable engineering win.



Every case in this chapter runs left to right along this pipeline. The lesson is the pipeline itself.

PART 1: Predicting the World

| The Principle: A Brain That Guesses

Start with a surprising claim: the brain is not a passive camera, it is a prediction machine. In the **predictive coding** framework, higher areas constantly send top-down guesses about what the lower sensory areas should be seeing, and the lower areas send back only the **prediction error**, the mismatch between guess and reality (Rao and Ballard, 1999). Everything already predicted is silence; only the surprise travels upward. It is a ruthlessly efficient way to perceive.

In symbols, each level transmits not its raw input but the residual:

$$\varepsilon = \text{input} - \text{prediction}$$

ANALOGY

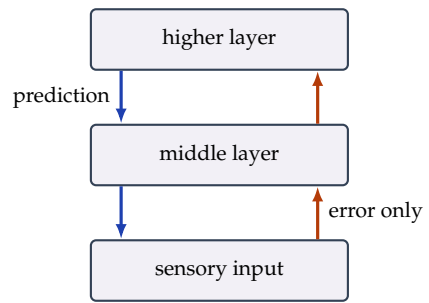
A good copy editor does not re-read you the whole manuscript, they hand back only the marked-up errors. Predictive coding runs perception that way: the page is assumed correct, and only the typos, the prediction errors, are worth sending up the chain.

| The Model: PredNet

PredNet (Lotter, Kreiman, and Cox, 2016) wires this into a recurrent convolutional network. Every layer tries to predict the activity of the layer below it; the top-down prediction is subtracted from the actual input, and only the resulting error is passed up to be explained.

| The Win

Predicting the next video frame, PredNet beat standard feedforward models, and the features it learned, with *no labels at all*, transferred to more robust object recognition, especially on noisy or ambiguous inputs. **Takeaway.** a model that actively predicts its own input, as the brain does, learns richer representations for free.



Predictive coding, and PredNet: guesses flow down (blue), only the surprises flow back up (amber).

■ The Principle: Learning During Rest

The brain does not learn only in the moment. The **hippocampus** re-plays the neural sequences of recent experience during sleep and quiet rest, a process thought to consolidate memories and hand them to the neocortex for long-term storage (Foster and Wilson, 2006). Crucially it does not replay everything equally, it favors the surprising and the important.

■ The Model: Prioritized Experience Replay

Reinforcement-learning agents already used **experience replay**, storing past transitions in a buffer and sampling them randomly to break the temporal correlations that destabilize training. DeepMind's **Prioritized Experience Replay** (Schaul et al., 2015) added the hippocampal twist: sample by *surprise*. The measure of surprise is the **TD-error**, how far the outcome fell from prediction, and high-error transitions are replayed more often.

In symbols, the surprise signal and the sampling rule are:

$$\delta = r + \gamma \max_a Q(s', a) - Q(s, a), \quad P(i) \propto |\delta_i|^\alpha$$

🔗 ANALOGY

Smart studying is not re-reading the whole textbook, it is drilling the flashcards you keep getting wrong. PER gives the agent that instinct: spend repetitions where the prediction failed, not where it already succeeds.

■ The Win

The tweak was small and the effect large: agents learned Atari games to superhuman level **faster** and on far **less data**, and reached higher final scores, purely by replaying the informative memories first.

The Principle: Global Context

The visual cortex is hierarchical, but it is also laced with long-range horizontal and feedback connections that let one part of a scene inform the reading of another. This global integration is the substrate of **visual attention**: selectively weighting what matters given the whole context, not just the local patch.

The Model: The Vision Transformer

For years vision meant CNNs, which see only local receptive fields and build up context slowly. The **Vision Transformer** (Dosovitskiy et al., 2020) threw that out. It slices an image into fixed patches, treats each patch like a word in a sentence, and feeds the sequence to a Transformer whose **self-attention** lets every patch weigh every other patch, near or far, from the very

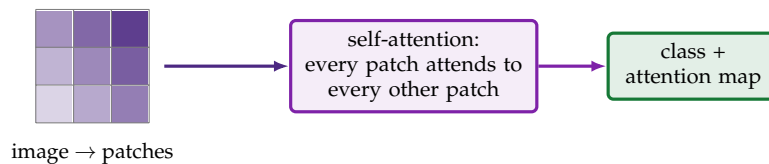
first layer (Vaswani et al., 2017).

Formally, attention scores every query against every key, then mixes the values:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

The Win

Trained on large enough data, ViTs **surpassed** state-of-the-art CNNs. They capture relationships between distant image regions, scale more gracefully as data and model grow, and their attention maps are directly **interpretable**, a literal picture of where the model looked.



The Vision Transformer: patches become tokens, and self-attention integrates the whole image from the first layer.

PART 4: Fusing the Senses

The Principle: Multisensory Integration

The brain builds one robust percept by pooling many senses. A region like the **superior temporal sulcus** responds to a face, a voice, and a name together, and that fusion yields a model of the world more reliable than any single channel. The whole is steadier than its parts.

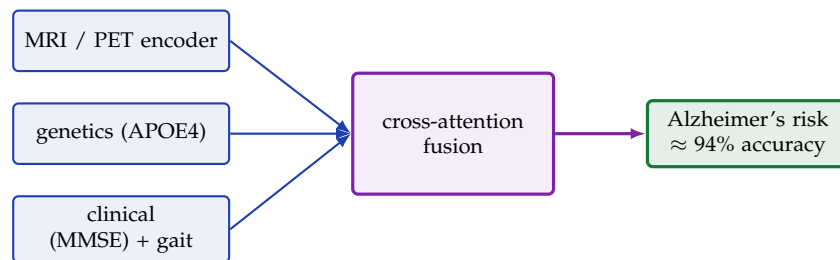
The Model: Multimodal Fusion

A patient, like a percept, is not one number. Inspired by the brain's integration, **multimodal models** for Alzheimer's prediction fuse very different data streams: **neuroimaging** (MRI and PET), **genetics** (risk

genes such as APOE4), **clinical** data (cognitive scores like the MMSE, demographics), and **behavioral** signals (speech, gait). A dedicated encoder reads each modality, and **cross-attention** learns how the streams relate before a joint prediction (Jack et al., 2018).

The Win

Fusing the evidence beat every single-modality model, reaching about **94%** accuracy in the chapter's example, flagged high-risk individuals **years** before clear symptoms, and surfaced **personalized** biomarker combinations, the mix most predictive for a given patient.



Multimodal fusion: one encoder per data type, cross-attention to relate them, a prediction stronger than any modality alone.

Brain principle	AI implementation	Signal used	Engineering win
Predictive coding	PredNet	Prediction error	Robust, label-free vision
Hippocampal replay	Prioritized replay (PER)	TD-error (surprise)	Far better sample efficiency
Global integration	Vision Transformer	Self-attention	Scales past CNNs
Multisensory fusion	Multimodal fusion	Cross-attention	Earlier Alzheimer's prediction

Quick Self-Test

1. In predictive coding, what does a lower area send up to a higher one, and why is that efficient?
2. What signal does Prioritized Experience Replay use to decide which memories to replay more, and what brain phenomenon inspired it?
3. How does a Vision Transformer turn an image into something a Transformer can attend over, and what can it do that a CNN struggles with?
4. Name the four data streams fused in the Alzheimer's model, and state the payoff of fusing them.
5. State the three-stage NeuroAI recipe that all four cases share.

If You Remember One Thing

Each case is the same pipeline: a principle for how the brain computes becomes an algorithm, and the algorithm delivers a measurable win. Predict your input (PredNet), replay your surprises (PER), attend to everything at once (ViT), fuse your senses (multimodal). The brain is not just inspiration, it is a working design document.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

AI for Neuroscience Discovery

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 24 of "The NeuroAI Handbook"

Deep learning as a computational microscope: decoding thoughts, mapping wiring, and reading molecules.

One-sentence thesis. AI is not only inspired by the brain; it has become our most powerful instrument for reading the brain, turning noisy, high-dimensional neural data into decoded speech, complete wiring diagrams, and atom-level protein structures.

PART 1: The Computational Microscope

The Missing Instrument

Start with what we could not do, not with the algorithm. For a century we could stain a slice, record the crackle of one neuron, or watch blood drift across an fMRI scan, yet the brain's real language, the coordinated firing of thousands of neurons at once, stayed invisible. We had no microscope for high-dimensional population activity. Deep learning is that microscope. This is the second half of the NeuroAI loop: the brain inspires AI, and AI, in turn, becomes the tool that lets us see the brain.

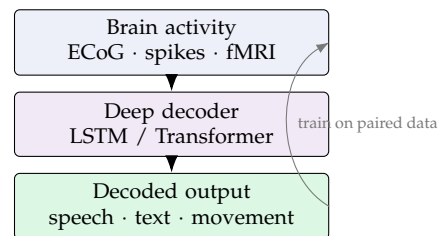
Seeing the Unseen: Neural Decoding

The sharpest demonstration is **neural decoding**, translating raw electrical activity into its meaningful content. Intent to say a word is not one neuron firing; it is a fleeting pattern spread across a population, and classical statistics drowned in the noise. Sequence models, LSTMs and Transformers, are built for exactly this. Researchers record activity, often with **ECoG** grids on the cortical surface, while a person speaks, and train the network to map spatio-temporal patterns onto words.

The results reach into the clinic. Decoders can reconstruct speech, rebuild the faces and scenes a person is viewing from fMRI, and read out motor intent in paralysis. For **Why it matters.** patients locked in by **ALS**, stroke, or brainstem injury, a decoder driving a speech synthesizer gives back a voice, Moses et al. (2021) restored speech in a paralyzed man, Willett et al. (2021) turned imagined handwriting into text, and Tang et al. (2023) reconstructed continuous language from *non-invasive* fMRI.

ANALOGY

A decoder is a code-breaker with a Rosetta Stone. It never gets a dictionary of the neural code; it only sees activity paired with what the person did, and learns the translation from those parallel texts. Enough aligned examples, and the cipher falls.

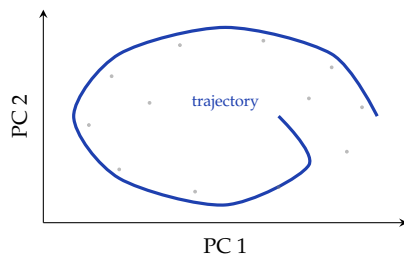


The decoder learns a mapping from population activity to meaning, supervised by what the person actually said or did.

Why High-Dimensional Data Is Not Really High-Dimensional

Record a thousand neurons and you get a thousand-dimensional signal, but those neurons do not move independently. During a reach, or a decision, the population traces a smooth, low-dimensional path, a **neural manifold**, that is the true shape of the computation. Churchland et al. (2012) showed motor-cortex activity during reaching lives on such a low-dimensional surface; the job of analysis is to find it.

The workhorses are dimensionality-reduction methods: **PCA** for the linear backbone, **t-SNE** and **UMAP** for nonlinear structure, and **VAEs** or sequential autoencoders (LFADS, Pandarinath et al. 2018) to infer single-trial dynamics. **Watch out.** t-SNE and UMAP distort distances; use them to visualize, not to measure, and read PCA's variance-explained before trusting any picture.



Population activity (gray points) collapses onto a low-dimensional trajectory (blue): the manifold is the computation's shape.

Foundation Models for Brains

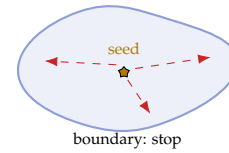
The GPT idea, pre-train once and transfer, is now reshaping neural analysis. **CEBRA** (Schneider, Lee & Mathis, 2023) uses *contrastive* self-supervision to compress recordings into latent embeddings that stay consistent across sessions and even across species, with no labels required. **Brain-Score** (Schrumpf, DiCarlo and colleagues) turns “is this model brain-like?” into a leaderboard, scoring networks against V1, V4, IT, and behavior. Two lessons stand out: self-supervised models look more brain-like than purely supervised ones, and the best models still explain only **~70%** of the explainable variance in IT cortex. Transfer learning is the practical payoff, fine-tuning a pre-trained model needs **10–100×** less data than training from scratch.

A Map No Human Can Draw by Hand

To know how a circuit computes you need to know how it is wired, and here the obstacle is brute scale, not subtlety. One cubic millimeter of cortex imaged by electron microscopy at 4 nm per pixel is roughly a **petabyte**; tracing every neuron by hand would take annotators *thousands of years*. The problem is not that we cannot see the wires, it is that there are far too many to follow manually.

Flood-Filling Networks

Google Research's **flood-filling networks** (Januszewski et al., 2018) automate the tracing. Drop a seed inside a neuron; a 3D convolutional network with recurrent memory repeatedly predicts which neighboring voxels belong to the same cell, floods outward through the volume, and stops when confidence falls at a boundary. A companion CNN detects synapses at **>95%** precision and **>90%** recall, finding millions in hours.



Flood-filling: from a seed, the network floods outward voxel by voxel and halts at the neuron's membrane.

From Fly to Human

FlyWire (Dorkenwald et al., 2022) paired this AI with human-in-the-loop proofreading, 100+ neuroscientists correcting errors in a browser, to reconstruct **~130,000** fly neurons and **54.5 million** synapses. The same recipe now scales up: the **MICrONS** project mapped a cubic millimeter of mouse cortex (about 200,000 neurons), and the **H01** dataset (Google & Lichtman lab) reconstructed a fragment of human temporal cortex, 50,000 neurons and 130 million synapses, revealing cell types unique to us. **The pattern.** AI does the fast, tireless first pass; humans supply judgment where it is scarce.

Brain	Neurons	Synapses	EM data size
<i>C. elegans</i>	302	~7,000	1 GB
Fruit fly	100,000	100 million	100 TB
Mouse cortex (1 mm ³)	100,000	1 billion	1 PB
Human brain	86 billion	100 trillion	~10⁶ PB

The scale of connectomics. Each step up multiplies data by orders of magnitude, which is why automation, not more annotators, is the only path forward.

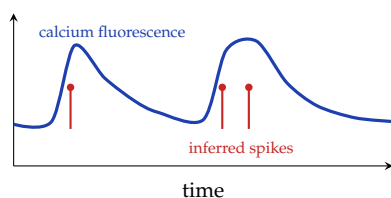
From Photons to Spikes

The microscope works at the cellular scale too. Two-photon calcium imaging watches thousands of neurons at once as GCaMP indicators brighten when calcium floods a firing cell, but the raw movies are noisy, huge (512×512 over thousands of frames), and contaminated by overlapping neuropil. **Suite2p** (Pachitariu et al., 2017) and **CaImAn** (Giovannucci et al., 2019) automate motion correction, cell detection, and signal extraction; CaImAn models the movie as a matrix factorization.

In symbols, the fluorescence video Y splits into neuron footprints times their activity, plus background and noise:

$$Y = AC + B + E$$

One more gap remains: fluorescence is slow (hundreds of ms) while spikes are 1–2 ms, so **CASCADE** (Rupprecht et al., 2021) deconvolves traces back to spike times, trained on paired imaging-plus-ephys ground truth. The payoff is stark, hand-drawing ROIs took 2–3 days for a couple hundred cells; the pipelines take 10–30 minutes for thousands, a **100×** acceleration that enables real-time closed-loop experiments.



Spike inference deconvolves the slow calcium signal (blue) back to the fast action potentials (red) that caused it.

Reading the Molecules: AlphaFold

Go down one more level, to the proteins that make neurons excitable. A protein's function is set by its 3D fold, but predicting that fold from sequence was a 50-year-old grand challenge, and the hardest cases, membrane-embedded ion channels, resist crystallography. **AlphaFold 2** (Jumper et al., 2021) cracked it at CASP14 using attention over the sequence, evolutionary information from multiple alignments, and iterative refinement, then released structures for nearly the entire human proteome. For neuroscience it predicted hundreds of voltage-gated Na, K, and Ca channels, exposing S4 voltage sensors, pore selectivity, gating, and druggable pockets, which feeds structure-based screening for antiepileptics, analgesics, and neuroprotectants.

AI as Collaborator

The microscope is starting to run its own experiments. Closed-loop setups use reinforcement learning and **Bayesian optimization** to pick the next stimulus or tune deep-brain stimulation with far fewer trials than grid search, and connectome-plus-physiology data now feed **digital twins**, in-silico circuits (Billeh et al., 2020; Blue Brain) where you can silence a cell type and watch what breaks.

Quick Self-Test

1. Why is a deep decoder better suited to reading the “neural code” than classical linear statistics, and what supervises its training?
2. What does it mean for population activity to lie on a low-dimensional manifold, and why should you distrust distances in a t-SNE plot?
3. Why is manual tracing hopeless for a cubic millimeter of cortex, and how does a flood-filling network get around it?
4. In the equation $Y = AC + B + E$, what do **A** and **C** represent, and why is a separate spike-inference step still needed?
5. Name one reason AlphaFold matters specifically for neuroscience rather than biology in general.

If You Remember One Thing

AI is the microscope neuroscience was missing. The same core move, let a network learn the mapping from messy high-dimensional data to meaningful structure, decodes speech from cortex, uncovers the manifold behind a decision, traces every wire in a connectome, and folds an ion channel. The instrument does not replace the scientist; it hands over the scale problems so the judgment problems can get human attention.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Embodied AI and Robotics

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 25 of "The NeuroAI Handbook"

Why intelligence needs a body: sensorimotor loops, prediction, and brain-inspired control.

One-sentence thesis. The path to general, robust AI may run not through bigger models in a digital vat but through a body, where perception, prediction, and action close into one loop grounded in the physics of the real world.

PART 1: Intelligence Needs a Body

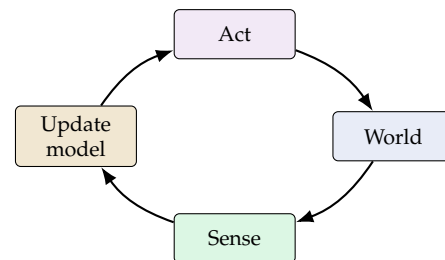
The Brain in a Vat

Begin with a puzzle, not a controller. Today's large models are brilliant "brains in vats," trained on frozen snapshots of the internet; they have never dropped a cup, leaned into a turn, or felt an action push back. Biological intelligence is the opposite, it is **embodied**, shaped every moment by the traffic between brain, body, and world. The **Embodiment Hypothesis** (Brooks, Pfeifer) claims this is no accident: real general intelligence *requires* a body, because cognition is grounded in the messy, real-time feedback of sensing and acting, not in symbol-shuffling alone.

The Sensorimotor Loop

The deepest source of knowledge is the loop $\text{act} \rightarrow \text{sense} \rightarrow \text{learn} \rightarrow \text{repeat}$. An infant learns gravity by dropping a toy, not by reading about it; the physical world hands back an endless stream of *self-*

supervised data that is always correctly labeled by reality. An embodied agent senses the state of world and body, acts to change it, then updates its internal model from the relationship between action and sensory consequence. **The core claim.** this loop, not any single module, is the engine of learning, and it is the shared root of modern reinforcement learning and robotics.



The sensorimotor loop: every action is a labeled experiment whose result grounds the next belief.

Walking Without Micromanaging: Atlas and CPGs

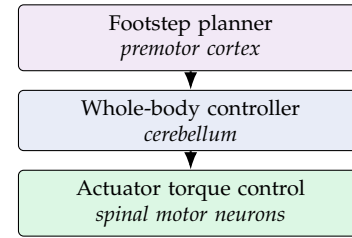
How does Boston Dynamics' Atlas (1.5 m, **89 kg**) run, jump, and backflip? Not by scripting every joint. Its gaits ride on **central pattern generators** (CPGs), rhythmic neural circuits first found in animal spinal cords, a lamprey's isolated cord still generates swimming. Computational CPGs are coupled oscillators: a handful of tunable rhythms replace dozens of independent joint commands, giving stable, smooth, and easily re-timed gaits.

In its simplest form, each limb oscillator carries a phase θ_i pulled toward a preferred offset from its neighbors:

$$\dot{\theta}_i = \omega_i + \sum_j w_{ij} \sin(\theta_j - \theta_i - \phi_{ij})$$

Balance layers a brain-like hierarchy on top: an IMU stands in for the vestibular system, joint encoders for muscle spindles, force sensors for Golgi tendon organs, feeding a footstep planner (premotor cortex), a whole-body controller (cerebellum), and torque controllers (spinal neurons). A **predictive model** lets Atlas anticipate its own dynamics; a mismatch after

an unexpected push triggers fast correction, exactly the cerebellum's job.



A hierarchical controller mirrors the motor stack: plan, coordinate, execute.

Outsourcing Control to the Body

An octopus has no skeleton and hundreds of degrees of freedom, yet its brain does not micromanage a single arm, the arm's mechanics do much of the work. That is **morphological computation**: the body itself performs part of the control. A soft silicone gripper with pneumatic channels curls to match an object's shape from one pressure input, no per-joint planning, tolerant of positional error, and safe against human skin. The same logic runs in muscle: a stretched muscle instantly resists with more force, a partly mechanical reflex. **Where it pays off.** soft grippers handle delicate tissue in surgery and unbruised fruit in harvesting, tasks that defeat rigid tools.

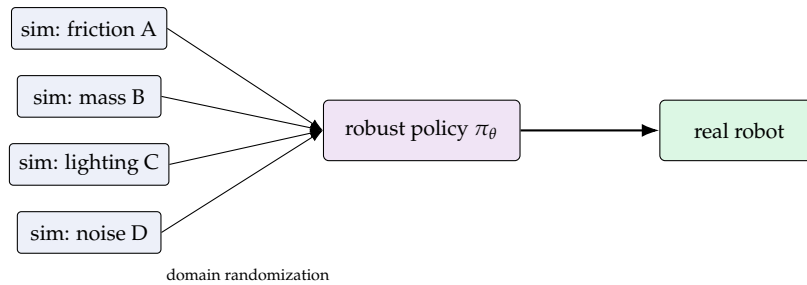
The Problem With Perfect Simulators

Real robots are slow, breakable, and cannot be fast-forwarded, so we would rather train in simulation where millions of trials run in parallel. The catch is the **reality gap**: simulators approximate contact, friction, actuator lag, and lighting, and a policy trained on those approximations can fail catastrophically when reality differs, as if the laws of physics changed under its feet.

Domain Randomization

The fix is counterintuitive, make the simulation *more* variable than reality. **Domain randomization** (Tobin et al., 2017) randomizes mass, friction, lighting, textures, sensor noise, and object positions every episode, so the policy sees thousands of “worlds” and learns only the task features invariant across them. Reality then becomes just one more sample it has already handled.

Formally, the policy minimizes expected loss over environments ξ drawn from a distribution:



Train across a wide distribution of randomized simulations; reality lands inside the distribution, so one policy transfers.

$$\min_{\theta} \mathbb{E}_{\xi \sim p(\Xi)} [\mathcal{L}(\pi_{\theta}; \xi)]$$

When It Works: Dactyl and QT-Opt

OpenAI’s **Dactyl** learned to solve a Rubik’s cube with a five-fingered hand *entirely* in simulation, then did it for real, recovering even when researchers whacked it with a plush giraffe. It never explicitly learned finger gaiting or gravity-assisted reorientation; those human-like strategies emerged. Sparse rewards were tamed by **Hindsight Experience Replay** (Andrychowicz et al., 2017), which relabels a failed reach to B as a success *at* B, wringing signal from every attempt. At scale, **QT-Opt** (Kalashnikov et al., 2018) trained on **580,000** real grasps to hit **96%** success on unseen objects with closed-loop vision. **The principle.** robustness comes from exposure to diversity, the same reason your cerebellum, trained on flat ground, still balances you on gravel and ice.

Why You Cannot Tickle Yourself

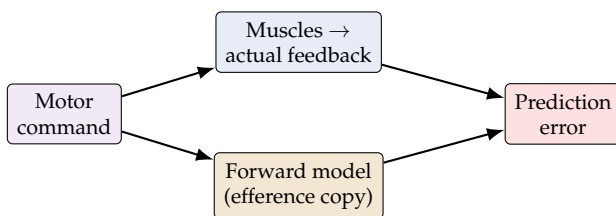
When motor cortex sends a command, it also sends a copy, an **efference copy**, to the cerebellum, which runs a **forward model** predicting the sensory result. The brain compares prediction to reality:

$$\text{error} = \text{actual feedback} - \text{model}(\text{efference copy})$$

Self-generated sensations are predicted away (hence no self-tickle), while any surprise flags something the body did not cause. Two responses follow: update the forward model (adaptation, as when a heavy backpack first feels wrong, then normal) or correct the action (online correction, nudging a reaching hand back on course). This is **model predictive control** in the brain, and it unifies neuroscience, robotics, and optimal control under one idea: intelligent behavior arises from prediction and the drive to minimize prediction error.

ANALOGY

Learning to ride a bike is embodiment in miniature. You start with motor primitives (pedal, steer); your cerebellum builds a forward model (“lean this much, feel this much tip”); the lurch of nearly falling is a giant error signal that rewrites the model. Once the model is accurate, you balance without a conscious thought, the skill has moved into the body.



Efference copy lets the cerebellum predict the sensory consequences of a movement; the residual is the error that drives learning.

Active Inference

Active inference (Friston) pushes this further: perception and action are one process, both minimizing *surprise* (free energy). Faced with a prediction error you can change your belief (perceptual inference) or change the world to fit the prediction (action). A reaching agent whose “prior” is the target simply acts to make its sensations match, control emerges from prediction rather than from an explicit error-correcting gain, and it maps cleanly onto neural firing and population codes.

Learning Like an Infant

Infants are Earth’s best learners, and **developmental robotics** copies their trajectory: reflexes, then body babbling, then acting on objects, then tool use. The driver is **curiosity**, formalized as seeking maximal *learning progress*, the Goldilocks zone between the already-mastered and the currently-impossible:

$$\text{curiosity}(s) \propto |L_t(s) - L_{t-\Delta t}(s)|$$

This yields a *self-organizing curriculum*, body before world, near before far, static before dynamic, none of it hand-designed. Along the way the robot learns **affordances**: it perceives not “a cup” but “graspable, liftable,” reframing perception as action-centered (Gibson), a job the parietal cortex is thought to do.

Sharing Space With Us

A body means sharing space with people, the domain of **social robotics**: joint attention, predicting intent from motion, and **learning from demonstration**. The last taps the brain’s **mirror-neuron** system, and its AI echoes are imitation-learning methods, GAIL (Ho & Ermon, 2016) and DeepMimic (Peng et al., 2018), that acquire fluid skills by watching rather than by exhaustive trial and error.

Brain mechanism	AI / robotics analog	What it buys
Spinal CPG	Coupled oscillators	Stable rhythmic gait
Cerebellar model	forward MPC / learned dynamics	Predictive correction
Basal ganglia	Model-free RL	Long-horizon reward
Muscle & tendon mechanics	Morphological computation	Control outsourced to the body
Mirror neurons	Imitation learning (GAIL, DeepMimic)	Skills from demonstration
Parietal cortex	Affordance detection	Action-relevant perception

Embodied AI is largely a translation table: each motor-control trick in the brain has a working analog in robotics.

Quick Self-Test

1. State the Embodiment Hypothesis in one sentence, and explain why a “brain in a vat” might be fundamentally limited.
2. What is morphological computation, and how does a soft pneumatic gripper illustrate it?
3. Why does making a simulator *more* variable than reality help a policy transfer to the real robot?
4. What is an efference copy, and how does the forward-model comparison explain why you cannot tickle yourself?
5. In curiosity-driven learning, why does maximizing learning progress produce a curriculum no one had to design?

If You Remember One Thing

Embodiment is not a bigger dataset; it is a different kind of intelligence. Put perception, prediction, and action in one closed loop with a real body, and hard problems soften: robust control emerges from diversity (domain randomization), smooth movement from prediction (cerebellar forward models), and a learning curriculum from curiosity and the body’s own constraints. Nearly every robotics advance in this chapter is a rediscovery of how the brain already controls the body.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Lifelong Learning

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 26 of "The NeuroAI Handbook"

How brains keep old memories while learning new ones, and the AI tricks that borrow the idea.

One-sentence thesis. Forgetting is not a failure of capacity but of interference, new learning overwriting old, so every real solution, biological or artificial, is a policy for deciding what to protect.

PART 1: The Stability-Plasticity Dilemma

The Tension at the Heart of Learning

Start with the bind, not the algorithm. A learner that never changes learns nothing new; a learner that changes freely erases what it already knew. Intelligence lives in the narrow band between those failures. This is the **stability-plasticity dilemma**: *plasticity* is the ability to absorb new information quickly, *stability* is the ability to keep old knowledge intact, and the two pull in opposite directions. Too plastic and yesterday's skill is gone by tomorrow; too stable and the system is a fossil.

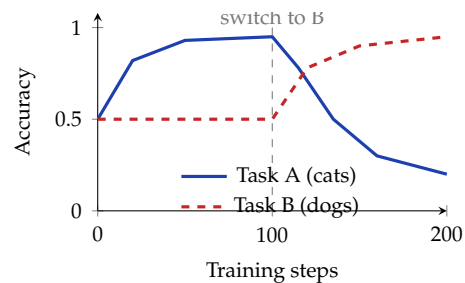
Humans manage this so effortlessly we forget it is hard. You can learn a colleague's name today without deleting your childhood address. Modern deep networks cannot, because they are extremely plastic and almost nothing anchors them to the past.

Catastrophic Forgetting

The failure mode has a name. Train a standard network to recognize cats, then train the same network to recognize dogs, and it will typically forget cats almost completely, its accuracy on the first task collapsing

toward chance. This is **catastrophic forgetting** (McCloskey and Cohen, 1989), and it is the single biggest barrier to AI that learns in the real world.

The mechanism is plain once you see it. Gradient descent adjusts *shared* weights to reduce error on whatever data is in front of it right now, with no term in the loss that says "and keep the old task working." The weights that encoded cats are simply overwritten to fit dogs. **Watch out.** This is not a bug in one architecture, it is the default behavior of any network trained on tasks in sequence without a countermeasure.



The moment training switches to Task B, accuracy on Task A falls off a cliff.

Two Systems, Two Speeds

The brain faces the identical dilemma and solved it not by finding one perfect learning rate but by using **two** memory systems at once. This is **complementary learning systems** theory (McClelland, McNaughton, and O'Reilly, 1995; Kumaran, Hassabis, and McClelland, 2016). The **hippocampus** learns fast, stores specific episodes, and is highly plastic. The **neocortex** learns slowly, stores generalized knowledge, and is stable. New experience lands first in the fast system; only gradually, and carefully, does it migrate into the slow one.

ANALOGY

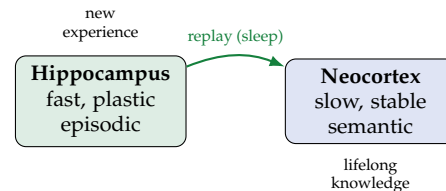
A sculptor's studio and a grand library. The hippocampus is the studio, fast and messy, shaping each day's experience into a fresh sculpture. The neocortex is the library, slow and permanent. Each night the librarian reviews the day's best pieces and painstakingly casts a few in marble for the permanent collection, leaving the archive stable and organized.

Sleep Replay Does the Consolidation

The migration is not passive. During sleep the hippocampus literally **replays** the day's experience, reactivating the same neural sequences seen during

waking, sped up ten to twenty times, packaged into brief high-frequency bursts called **sharp-wave ripples**. Each replay is a training example handed to the neocortex, so the slow system learns the new pattern gradually, interleaved with old knowledge, without a sudden overwrite. Consolidation runs over hours to years: a fresh memory is hippocampus-dependent, an ancient one lives distributed in cortex.

Two further biological details matter. **Selectivity, not volume.** The brain does not replay everything; salient and rewarding experiences are prioritized. And **synaptic tagging** (Frey and Morris, 1997) shows the molecular version: a strongly stimulated synapse triggers protein synthesis, and only recently "tagged" synapses can capture those proteins to become permanent, so what gets consolidated is decided locally and competitively.



Fast episodic capture, then slow, interleaved consolidation into the stable store.

If Forgetting Is Overwriting, There Are Only Three Moves

Once you see forgetting as interference in shared weights, the space of fixes is small: **rehearse** the past so it is never fully absent, **protect** the weights that encode it, or **grow** new capacity so nothing has to be overwritten. Every continual-learning method is a version of one of these three.

Replay, Rehearse the Past

The most direct fix is also the most biological: keep a buffer of old examples and mix them into new training. This **experience replay** is the engineering twin of hippocampal replay, and on standard benchmarks it is the most effective single method, storing even a small slice of old data prevents most forgetting. Its cost is memory and, sometimes, privacy, you have to keep the old data around.

Regularization, Protect What Matters

You do not always have the old data. **Elastic Weight Consolidation** (EWC; Kirkpatrick et al., 2017) instead asks which *weights* mattered for the old task and anchors those in place. Importance is measured by the **Fisher information**, essentially the curvature of the old-task loss around its optimum: a weight in a steep valley is critical, a weight on a flat plateau is free to move.

In symbols, learning task B minimizes the new loss plus a Fisher-weighted penalty pulling each weight back toward its old value $\theta_{A,i}^*$:

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_{A,i}^*)^2$$

ANALOGY

Tie each weight to its old value with a spring. The spring's stiffness is that weight's Fisher information, so important weights are held tightly while unimportant ones stretch freely to fit the new task. The knob λ tunes all the springs at once: too loose and old knowledge drifts away, too tight and nothing new can be learned.

Synaptic Intelligence and Memory Aware Synapses are cousins that estimate importance differently, and the brain analog is real, **synaptic consolidation** makes important synapses physically resistant to change.

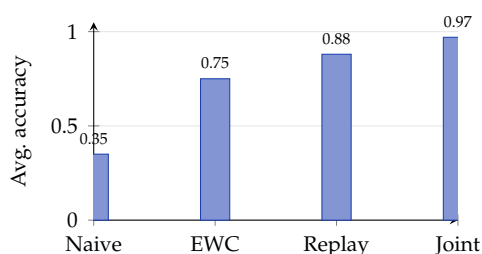
Architecture, Grow New Capacity

The third move refuses to share weights at all. **Progressive Neural Networks** (Rusu et al., 2016) freeze the entire old network and add a fresh "column" for each new task, wired to the old columns by **lateral connections** so new tasks can reuse old features. The result is **zero forgetting** plus positive forward transfer, paid for with memory and compute that grow with the number of tasks. Two lighter variants round out the toolbox: **Learning without Forgetting** preserves the old input-output behavior by distillation instead of storing data, and **meta-learning** (MAML, Reptile) hunts for an initialization that adapts to any new task in a few steps, echoing the brain's *metaplasticity*, the plasticity of plasticity itself.

Method	Core idea	Forgetting	Cost	Brain analog
Fine-tuning	Just train on each task in turn (baseline)	Severe	$O(1)$	none
Replay	Rehearse a buffer of stored old examples	Low	buffer memory	hippocampal replay
EWC	Anchor high-Fisher weights near old values	Moderate	$2 \times$ params	synaptic consolidation
Progressive Nets	Freeze old, add a column per task	Zero	$O(T)$ params + compute	cortical expansion / routing
LwF	Distill old behavior, no old data kept	Low-Mod	old-model copy	functional preservation

The Split-MNIST Verdict

Put the methods on the same benchmark, five sequential digit tasks, and the ranking is clear. Naive fine-tuning drops from about **95 percent** to roughly **20 percent** on the earliest tasks, textbook catastrophic forgetting. Replay recovers most of that. EWC lands in between, a good middle ground when old data cannot be kept. Joint training on all tasks at once is the upper bound nobody can beat online. The metrics that matter are **average accuracy**, **forgetting** (a task's drop from peak to final score), and **forward/backward transfer** (whether learning one task helps others).



Average accuracy over all five tasks; Joint training is the offline upper bound.

The Foundation-Model Shortcut

The newest twist changes the game. Take a large pretrained model, freeze it, and adapt to each new task with a tiny set of added parameters, prompts, adapters, or **LoRA** low-rank updates, typically **0.1 to 1 percent** of the weights per task. Because the base never changes, forgetting of its core knowledge is zero, and the rich pretrained features transfer to almost everything. It is efficient and popular, but it leans on an expensive pretrained model and can struggle on domains far from what that model saw.

What the Brain Still Does Better

Even the best methods are missing pieces of the biological recipe. AI trains continuously; the brain has dedicated **offline** consolidation phases (sleep). AI replay usually samples uniformly; the brain replays **selectively**, prioritizing what matters. And over long sequences, networks can suffer **catastrophic plasticity loss**, becoming too rigid to learn anything new, the mirror image of forgetting. Task-free settings and true backward transfer, where new learning actually improves old skills, remain largely open.

Quick Self-Test

1. In a network trained on cats then dogs, what specifically gets destroyed, and why does plain gradient descent do it?
2. Name the brain's two complementary learning systems and state what each contributes to avoiding forgetting.
3. In EWC, what does the Fisher information measure, and how exactly does it change the loss function?
4. Progressive Networks guarantee zero forgetting. What is the price, and why does it grow with the number of tasks?
5. Give one thing sleep-based consolidation does that experience replay in current AI systems does not.

If You Remember One Thing

Forgetting is interference, and every solution is a policy for what to protect. The brain protects by consolidating its most important memories into a separate, stable store while it sleeps; AI protects by rehearsing old data, anchoring important weights, or freezing them, and none yet matches the brain's selective, offline consolidation.

Quantum Computing for NeuroAI

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

Chapter 27 of "The NeuroAI Handbook"

What qubits actually are, why the brain probably is not one, and how quantum ideas reframe old questions about the mind.

One-sentence thesis. Whether or not the brain exploits quantum mechanics, the discipline of building AI on quantum principles hands us a new, formal language for the holistic and probabilistic character of thought.

PART 1: The Question, and the Skeptic

The Central Question

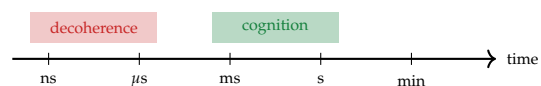
This is the most speculative chapter in the book, and it earns that label honestly. The question is simple to state and hard to answer: **are there quantum effects in the brain that are essential for cognition?** The claim is *not* that the brain is a quantum computer. It is a narrower, more useful move, to ask whether the vocabulary of quantum mechanics, superposition and entanglement, gives us new ways to think about consciousness, ambiguity, and the unity of experience. Treat what follows as exploration, not doctrine.

Warm, Wet, and Noisy

Before entertaining any parallel, respect the mainstream verdict, which is deeply skeptical. Quantum states are fragile. The instant a quantum system interacts with its environment, thermal jostling, stray fields, vibration, its delicate superposition **decoheres** and collapses to an ordinary classical state. This is

why real quantum computers run near absolute zero in extreme isolation.

The brain is the opposite: body temperature, soaked in a churning biochemical soup. **Here is the killer detail.** Estimated decoherence times in that environment are nanoseconds to microseconds, while cognition unfolds over milliseconds to seconds, orders of magnitude too slow for any quantum coherence to survive and do useful work (Tegmark, 2000). A second objection compounds it: quantum effects are an atomic-scale phenomenon, and leaping from there to the behavior of large neural circuits is an enormous, unproven jump in scale.



Coherence dies long before a thought completes; the gap is the skeptic's whole case.

Superposition, the Spinning Coin

To even ask the question you need two concepts, and the first is **superposition**. A classical bit is a switch, firmly 0 or 1. A **qubit** is more like a coin mid-spin: it is genuinely a blend of heads and tails at once, and only when you stop and “measure” it does it snap to a single definite outcome.

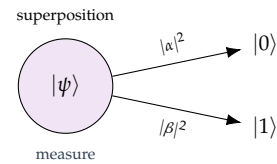
ANALOGY

A spinning coin is neither heads nor tails, it is both, in proportion, until it lands. A qubit is the same: it holds a weighted mixture of 0 and 1, and the act of looking is what forces the choice. You never observe the spin, only the landing.

In symbols, a qubit’s state is a weighted combination of the two basis states, with the squared weights giving the probabilities of each measurement outcome:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1$$

A parallel to cognition? When you weigh an ambiguous decision, your mind seems to hold several conflicting possibilities in suspension before one “collapses” into a conscious choice. Superposition is at least a vivid metaphor for that suspended, pre-decision state.



Measurement collapses one state into one classical outcome, with probability set by the weights.

Entanglement, the Twin Coins

The second concept is **entanglement**, what Einstein derided as “spooky action at a distance.” Two qubits can be linked so their fates are a single unified whole: measure one and you instantly know the other, no matter how far apart they sit.

ANALOGY

Two twin coins are flipped and fly to opposite ends of the universe. The moment you see one landed heads, you know, instantly, the other is tails. Nothing traveled between them; their outcomes were one linked thing all along.

A parallel to cognition, the binding problem. Look at a red ball and your brain processes its color, shape, and motion in separate regions, yet you perceive one unified object, not three loose features. How the brain **binds** distributed processing into a single percept (Treisman, 1996) is a genuine mystery. Entanglement offers a metaphor, and perhaps only a metaphor, for that instantaneous unification of scattered information into one experience.

Quantum Neural Networks

To test any of these ideas you need a tool that natively speaks superposition and entanglement. That is a **Quantum Neural Network (QNN)**, a hybrid classical-quantum machine that runs in a loop. First, **encode** classical data into the states of a set of qubits. Second, process them through a **parameterized quantum circuit**: layers of rotation gates whose angles act as learnable weights, plus entangling gates (CNOTs) that correlate the qubits. Third, **measure**, collapsing the result to a classical output. Fourth, a **classical optimizer** computes the loss and nudges the rotation angles by gradient descent, then the loop repeats.

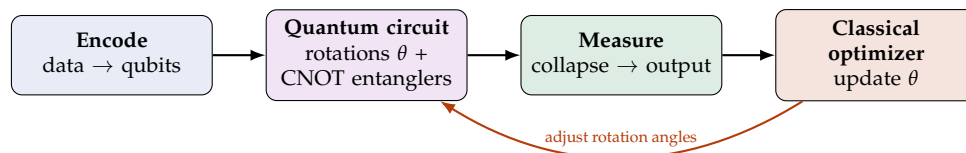
The point of a QNN is not raw speed. It is that superpositions and entangled correlations, awkward to represent classically, are the circuit's native language, which may make QNNs the right instrument for probing theories of quantum cognition. We are in what Preskill (2018) named the **NISQ** era, noisy, intermediate-scale quantum hardware, so these remain small and experimental.

Does Biology Ever Go Quantum?

The strongest card for the optimists is that biology *has* been caught exploiting quantum mechanics. Photosynthesis appears to use quantum **coherence** to route energy with near-perfect efficiency (Engel et al., 2007); bird navigation and certain enzymes are further candidates. This proves biology *can* harness quantum effects, but it is a long way from there to sustained quantum *computation* over the slow, warm timescales of thought. The boldest proposal, Penrose and Hameroff's **Orch-OR** theory, locates quantum coherence in neuronal **microtubules** and ties it to consciousness itself, an idea that is provocative, unproven, and squarely in the path of the decoherence objection.

Why Bother

Even if the brain turns out to be entirely classical, the exercise pays off. Building AI on quantum principles forces us to engage the holistic, probabilistic, deeply interconnected nature of the mind, and it may yield a new generation of models that are more powerful, and perhaps more brain-like, than anything we have today.



A QNN is mostly a classical training loop wrapped around a small quantum circuit.

Quantum concept	Proposed cognitive parallel	Where it stands
Superposition	Holding ambiguity before a choice collapses	Metaphor, not mechanism
Entanglement	Binding distributed features into one percept	Metaphor, not mechanism
Decoherence	Why quantum states cannot survive in the brain	Strong objection
Quantum biology	Photosynthesis, bird navigation use coherence	Real, but not computation
Orch-OR (microtubules)	Coherence in neurons produces consciousness	Speculative, contested

Quick Self-Test

1. Explain superposition with the spinning-coin analogy, and say precisely what “measurement” does to the qubit.

2. State the decoherence objection to the quantum-brain hypothesis, and explain why the brain's timescales make it so sharp.
3. What is the binding problem, and in what sense does entanglement "address" it, mechanism or metaphor?
4. Walk through the four stages of a QNN's hybrid loop, and say which parts are quantum and which are classical.
5. Photosynthesis exploits quantum coherence. Why is that not evidence that the brain computes quantumly?

If You Remember One Thing

The brain is almost certainly too warm, wet, and noisy to run quantum computations; on the timescales cognition needs, decoherence wins. The lasting value of this frontier is not a quantum brain but a new vocabulary, superposition and entanglement, for modeling the holistic, probabilistic character of thought, and a new class of machines, QNNs, for putting those ideas to the test.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Where Brains Meet Machines

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

A cross-cutting concept guide · The NeuroAI Handbook

How brain mechanisms map onto AI, where the mapping is tight, and where it quietly breaks.

One-sentence thesis. The book's central throughline is a correspondence, not an equivalence: neuroscience and AI keep discovering the same computational solutions, but a resemblance is only trustworthy once you say which level of analysis it lives on and where it stops.

PART 1: The Caricature and Its Rules

A Brilliant Caricature

Begin with the object that started the whole dialogue: the artificial neuron. It is a deliberate simplification of the biological cell, one that keeps a single computational essence, integrate inputs, transform, emit an output, and throws away almost everything else: the ion channels, the dendritic nonlinearities, the metabolic cost, the spike. McCulloch and Pitts (1943) drew the first version as a binary threshold unit; Rosenblatt's perceptron (1958) made it learn. Calling it a caricature is not an insult. A caricature exaggerates the one feature that matters and discards the rest, and that is exactly why the abstraction has been powerful enough to build modern AI and to serve as a testable theory of the brain.

The biological and artificial versions even differ in their governing equation. In symbols, a leaky integrate-and-fire neuron evolves a membrane voltage in continuous time, while an artificial unit is a single algebraic step:

$$\tau_m \frac{dV}{dt} = -(V - V_{rest}) + R_m I(t)$$

vs. $y = \max(0, \mathbf{w}^\top \mathbf{x} + b)$

Say Which Level You Mean

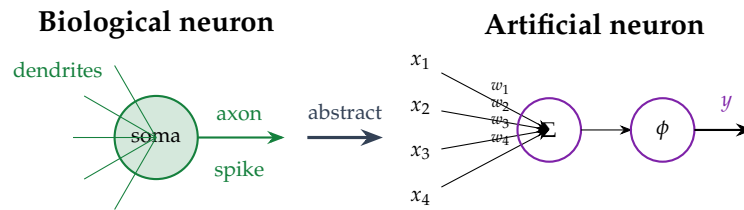
The discipline that keeps the analogy honest is David Marr's three levels of analysis. The **computational**

level asks what problem is solved (recognize objects under varying conditions); the **algorithmic** level asks what representations and steps are used (hierarchical feature extraction); the **implementational** level asks how it is physically realized (neurons and synapses, or transistors and matrix multiplies). A brain and a network can share a computational goal and even an algorithm while differing completely in implementation. So the claim "this AI works like the brain" is empty until it names a level, and alignment at one level never guarantees alignment at the others.

Watch out. Depth is not homology. A deep stack of ReLU layers resembling cortex at the algorithmic level tells you nothing about mechanism at the implementational level. Respect the analogy, resist the mirage.

ANALOGY

An airplane does not fly by flapping its wings, yet it obeys the same aerodynamics as a bird. Same computational principle (lift), different implementation (fixed jet wing vs. flapping feathered one). Most brain-AI correspondences are exactly this kind: shared principle, divergent hardware.



The founding caricature: dendrites become weighted inputs, the soma's integration becomes a weighted sum, and the axon's spike becomes an activation ϕ . Everything biophysical is dropped on purpose.

Vision: A Hierarchy Rediscovered Twice

The clearest win begins with Hubel and Wiesel’s finding (1962) that neurons in primary visual cortex act as edge detectors, feeding higher areas that assemble more complex shapes, a hierarchy from V1 to V4 to inferotemporal (IT) cortex. Fukushima’s Neocognitron (1980) borrowed this directly, and LeCun’s convolutional networks turned it into engineering: local receptive fields and stacked feature detectors. Then the loop closed. Once AlexNet (2012) scaled, DiCarlo’s and Yamins’s labs found the layers of a CNN trained only on object classification predict firing in the ventral stream better than any hand-built neuroscience model, explaining roughly **60 percent** of the variance in IT. Early layers match V1, middle layers V4, late layers IT. A system optimized for one goal had independently rediscovered evolution’s solution.

ANALOGY

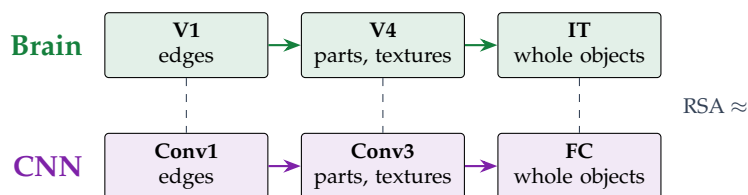
To ask whether a brain and a model “think alike,” do not compare neurons to units, compare their bookshelves. Give both the same movies: if Alice shelves by genre and Bob by director, their arrangements differ even with identical films. Representational Similarity Analysis compares these shelvings, the geometry of what-is-near-what, which is why it can align wetware with software at all.

Reward and Attention

Two more correspondences are almost eerily tight. In reinforcement learning, Sutton and Barto’s **temporal-difference** rule computes a prediction error; Schultz’s recordings (1997) found midbrain dopamine neurons fire in exactly this pattern, shifting from the reward to the cue that predicts it, and dipping when a promised reward is withheld. The abstract quantity and the biological signal obey the same equation:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

Attention is the third. The parietal cortex (area LIP) builds priority maps that amplify relevant locations; the Transformer’s self-attention does the computational same thing, a soft, content-based lookup, $\text{softmax}(QK^\top / \sqrt{d_k}) V$. Both solved the problem of routing limited processing to what matters, and BrainScore now ranks architectures by how well they predict neural responses.



Layer-by-layer correspondence of the ventral stream and a task-trained CNN. The match is real and quantitative, but it is an algorithmic-level resemblance, not proof of shared mechanism.

Replay and Two Speeds of Learning

During sleep and quiet rest the hippocampus replays recent experience at high speed, and place cells that fired along a maze re-fire in sequence, consolidating memory into cortex. This inspired **experience replay** in deep reinforcement learning: agents store past transitions in a buffer and resample them, which breaks the harmful correlations between consecutive frames and lets rare events be learned more than once. The parallel even extends to what gets replayed, **prioritized experience replay** favors high-error, surprising transitions, mirroring the hippocampus's preference for novel or rewarding episodes, and cut the frames needed to reach expert Atari play by **2 to 3 times**.

Behind this sits the **complementary learning systems** idea (McClelland et al., 1995): a fast hippocampus that grabs single episodes and a slow neocortex that distills statistics over many exposures. AI recapitulated it in episodic control (MFEC, NEC), pairing an instance-based memory with a gradient-trained

network so a single rewarding experience can be exploited at once.

Place, Grids, and Embeddings

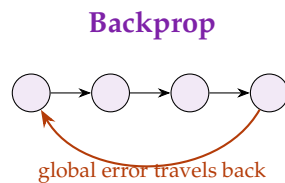
The brain's spatial code is the book's most vivid map. O'Keefe's **place cells** (1971) each fire at one location, a sparse "you are here" pin; the Mosers' **grid cells** (2005, a Nobel discovery) tile space in a hexagonal lattice at multiple scales, a universal coordinate system reused across environments. Their AI counterpart is the **learned embedding**: a distributed vector code where nearby points in the world map to nearby vectors. Framing the cognitive map as a graph, places as nodes and paths as edges, connects it to graph neural networks.

Watch out. Here the correspondence is looser than in vision. A learned embedding can carry map-like structure, but nothing forces it into crisp, single-peaked place fields or hexagonal grids; the similarity is functional (relational geometry), not a claim that the network grows grid cells.

PART 4: Where the Analogy Breaks

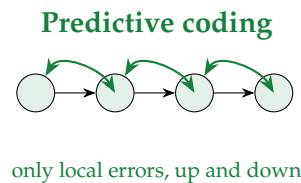
Credit Assignment: Backprop vs. the Brain

Both systems must solve the same problem, deciding which of millions of connections to blame for an error. Backpropagation is a mathematically exact answer, and it is the engine of deep learning, but it is widely regarded as biologically implausible for three concrete reasons: it needs the backward weights to mirror the forward ones, it requires storing every forward activation, and its error signals are non-local, traveling backward along paths real synapses do not have. The brain, if it approximates gradients at all, must do so differently. The book's candidates are **predictive coding**, where each level only ever computes local prediction errors (Rao & Ballard, 1999), plus feed-back alignment, target propagation, and three-factor Hebbian rules gated by neuromodulators. Hebbian learning is purely local; backprop is global; the gap between them is a live research frontier, not a solved mapping.



Codes, Signs, and Fragility

Three deeper divergences remain. First, **sparse spiking vs. dense rates**: biological neurons are event-driven, only about **5 percent** active at once, spending energy only on a spike, which is a large part of how the brain runs on **20 watts**; most artificial nets are dense and clock-driven, computing every unit every cycle. Spiking networks and neuromorphic hardware try to close this gap. Second, **Dale's principle**: a biological neuron releases the same transmitter at all its terminals and so is either excitatory or inhibitory, never both, whereas an artificial weight flips sign freely, an unremarked but real structural mismatch. Third, **robustness**: CNNs are fooled by adversarial perturbations invisible to people, need millions of labels where a child needs a few, and suffer catastrophic forgetting, none of which afflict the brain, whose recurrence, predictive feedback, and complementary memory the field is still trying to import.



Two answers to credit assignment. Backprop sends one global error signal backward along mirror-image weights; predictive coding trades it for local prediction errors exchanged between adjacent levels.

Brain mechanism	AI counterpart	Correspondence	Where it breaks
Ventral visual hierarchy	CNN layers (V1→IT)	Tight (algorithmic)	Adversarial, label-hungry, no recurrence
Dopamine prediction error	TD learning δ_t	Very tight	Brain uses many value signals
Parietal priority maps	Transformer attention	Tight	Softmax is not a circuit
Hippocampal replay	Experience replay	Functional	Buffer is not sleep
Place / grid cells	Learned embeddings	Loose	No forced grid structure
Cortical credit assignment	Backpropagation	Implausible	Non-local, weight-symmetric
Excit./inhib. (Dale)	Signed weights	Mismatch	Weights flip sign freely

Quick Self-Test

1. Why is the artificial neuron called a “caricature” rather than a “model” of the biological neuron, and what does that word buy you?

2. Someone says a deep network “works just like the brain.” Using Marr’s levels, what one question must you ask before the claim means anything?
3. What did Schultz’s dopamine recordings share with the TD learning rule, and why is this correspondence considered unusually tight?
4. Prioritized experience replay favors high-error transitions. What hippocampal phenomenon does that mirror, and is the match mechanistic or functional?
5. Give two concrete reasons backpropagation is considered biologically implausible, and name one biological alternative the book offers.
6. State Dale’s principle in one line and explain why an artificial weight violates it.

If You Remember One Thing

Brains and machines keep converging on the same computational solutions, hierarchical vision, prediction-error learning, attention, replay, because those solutions are good, not because silicon is wet. The value of the correspondence is entirely in its precision: name the level, cite the evidence, and mark where it stops. A correspondence you cannot break is not science; it is a mirage.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

The Quantitative Toolkit

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

A cross-cutting concept guide · The NeuroAI Handbook

The four math pillars the whole book stands on: information, inference, models, and cause.

One-sentence thesis. Four questions run underneath every chapter of this book, how much a signal tells you, how a belief should change, how a hypothesis becomes a testable model, and whether acting on it would change anything, and information theory, Bayesian inference, the GLM, and causal inference are the tools that answer them.

PART 1: Information Theory – How Much Does a Signal Tell You?

Information Is Surprise

Start with the question, not the equation. A message that you could have predicted (the sun rose today) tells you almost nothing; a message you could not (I won the lottery) tells you a great deal. Claude Shannon's 1948 insight was that this intuition can be measured: information is the resolution of uncertainty, and its unit is the bit. Shannon's **entropy** is the average surprise of a source, and it has a beautifully concrete reading, the minimum number of yes/no questions you need on average to pin down the outcome. A fair coin is **1 bit** (one question), a fair eight-sided die is 3 bits, an English letter about 4.2 bits. Entropy is largest when every outcome is equally likely and drops to zero when one outcome is certain.

In symbols, for outcomes with probabilities $p(x_i)$:

$$H(X) = - \sum_i p(x_i) \log_2 p(x_i)$$

Shared Knowledge and the Cost of Being Wrong

Two more quantities do most of the work in the book. **Mutual information** $I(X; Y)$ is the overlap of two uncertainties, how much learning Y reduces your surprise about X . It runs from 0 (independent) up to $\min(H(X), H(Y))$, and unlike correlation it catches nonlinear dependence. **KL divergence** $D_{KL}(P||Q)$ is the penalty in extra bits for modeling the world with the wrong distribution Q when the truth is P ; it is not symmetric.

$$I(X; Y) = H(X) - H(X | Y)$$

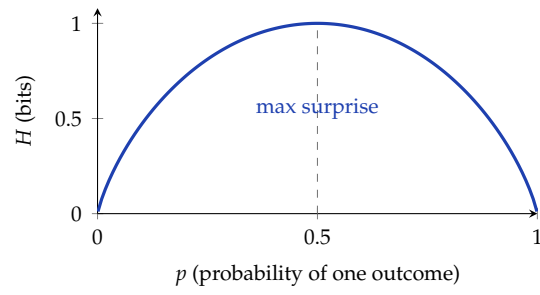
Watch out. These are not interchangeable with the statistics you already know. Two variables can share high mutual information while showing near-zero linear correlation, so a Pearson r of 0 does not license the word “independent.”

ANALOGY

KL divergence is a tourist navigating with a slightly wrong map Q of a city whose true layout is P . The divergence is the average number of extra wrong turns the bad map costs. Swap the maps and the cost changes, which is why $D_{KL}(P||Q) \neq D_{KL}(Q||P)$.

Why Brains and Nets Both Care

This is not decoration; it is the connective tissue of the book. Training a neural network by cross-entropy loss is exactly minimizing D_{KL} between the model's predictions and the data. Horace Barlow's **efficient coding hypothesis** (1961) says sensory systems evolved to strip redundancy and maximize information under a tight energy budget, which shows up as sparse coding and as predictive coding (transmit only the prediction error). Tishby's **information bottleneck** frames a deep network's layers as compressing the input while keeping what predicts the target. A single neuron carries only **2 to 3 bits** per spike, so the optic nerve's million axons are a genuine channel-capacity bottleneck, $C = \max_{p(x)} I(X; Y)$.



Binary entropy: uncertainty peaks at $p = 0.5$ (a coin you cannot call) and vanishes at $p = 0$ or 1 (a foregone conclusion).

The World Is Ambiguous

You see a shadowy figure in the dark. Is it a person, a tree, or a trick of the light? The same sensation can arise from different causes, sensory signals are corrupted by noise, and you never have complete information. The **Bayesian brain hypothesis** proposes that the brain copes by holding beliefs as probability distributions and updating them the moment evidence arrives. The engine of that update is Bayes' rule, which combines what you already knew (the prior) with what the data say (the likelihood) into a revised belief (the posterior).

$$P(H | D) = \frac{P(D | H) P(H)}{P(D)}$$

The prior dominates when data are scarce and is overwhelmed as evidence piles up. **Conjugate priors** (Beta-Binomial, Gamma-Poisson for spike counts, Normal-Normal) keep the posterior in the same family as the prior, which is what makes sequential updating cheap.

Weigh Evidence by Its Reliability

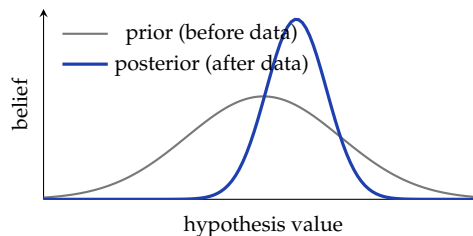
The framework makes a sharp, testable prediction about perception. When the brain fuses two noisy

cues, it should not average them equally; it should weight each by its **precision** (inverse variance), letting the more reliable cue speak louder.

$$\hat{x} = w_1 x_1 + w_2 x_2, \quad w_i \propto \frac{1}{\sigma_i^2}$$

Ernst and Banks (2002) showed humans combine visual and haptic size cues almost exactly this way. To turn a belief into an action, you maximize **expected utility** $\mathbb{E}[U(a)] = \sum_s U(s, a) P(s | D)$, weighting each outcome by its posterior probability. The **drift-diffusion model**, evidence accumulating to a threshold, captures both choice and reaction time and matches firing in areas LIP and FEF (Gold & Shadlen, 2007).

Watch out. Exact Bayesian inference is intractable for real problems, so the brain approximates, and people visibly deviate, neglecting base rates. On a condition with **1 percent** prevalence, a test that is 95 percent sensitive and 90 percent specific still leaves a positive result more likely wrong than right, because the prior is doing most of the work.



Bayesian updating: evidence pulls the belief toward the data and sharpens it. A broad prior becomes a narrow, confident posterior.

Fit, but Do Not Fool Yourself

You have a hypothesis about how a system works, say that a neuron's firing rate depends on stimulus contrast. Model fitting is the discipline of turning that hypothesis into numbers you can test: choose a model (the hypothesis), choose a cost function (how wrong a fit is), fit it (optimize the parameters), and, crucially, evaluate how well it predicts data it never saw. **Linear regression**, $y = \mathbf{X}\beta + \epsilon$ with Gaussian noise, is the starting point and is fine for something like an fMRI BOLD signal. But it is wrong for spike counts, which are non-negative integers, not bell-curved.

One Framework, Many Data Types

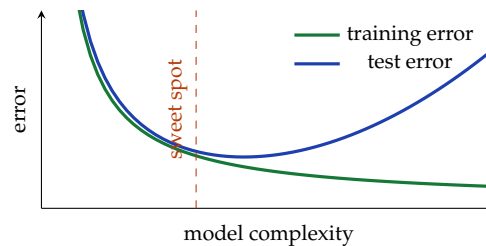
The **generalized linear model** fixes this with three parts: a linear predictor $\eta = \mathbf{X}\beta$, a link function $g(\mu) = \eta$, and a noise family matched to the data. Choose Gaussian with an identity link for continuous data, Bernoulli with a logit link for choices, and Poisson with a log link for spikes, which is the workhorse encoding model of systems neuroscience. Parameters are found by **maximum likelihood**, the values that make the observed spikes most probable.

$$\eta = \mathbf{X}\beta, \quad g(\mu) = \eta$$

The Overfitting Trap

The central danger is the **bias-variance tradeoff**. Too simple a model underfits (high bias); too flexible a model chases noise and overfits (high variance). Training error always falls as you add complexity, so it cannot tell you when you have gone too far, only **cross-validation** on held-out data can. **Regularization** buys back generalization by penalizing extreme weights: L2 (ridge) shrinks all coefficients, L1 (lasso) drives some to exactly zero and so selects features. **AIC** and **BIC** ($\text{BIC} = k \ln n - 2 \ln \hat{L}$) trade fit against parameter count, lower is better.

Watch out. There is a hidden bridge to Part 2: L2 regularization is exactly a Gaussian prior on the weights and L1 is a Laplace prior. Regularization is Bayes in disguise, so “frequentist” model fitting already smuggles in prior beliefs.



Training error keeps falling with complexity; test error turns back up. Only the held-out curve reveals overfitting, so never trust training fit alone.

Seeing Is Not Doing

Two neurons, one in visual cortex and one in motor cortex, fire together trial after trial. Does the visual neuron drive the motor one? Not necessarily, a presented stimulus may drive both, a **confounder** that manufactures the correlation while no arrow runs between the cells. This is the fault line under the whole book: every supervised model in Part 3 learns association, and association answers “what tends to happen,” not “what happens if I intervene.” Judea Pearl’s ladder names three rungs, association (seeing), intervention (doing), and counterfactual (imagining), and the point is that seeing and doing are different quantities. The $do(\cdot)$ operator marks an intervention that surgically cuts the arrows into X .

$$P(Y | do(X=x)) \neq P(Y | X=x)$$

Reading a Graph, Running an Intervention

A directed acyclic graph makes the assumptions explicit, and three shapes decide what you may adjust for. In a **chain** ($X \rightarrow Z \rightarrow Y$) and a **fork/confounder** ($X \leftarrow Z \rightarrow Y$), conditioning on Z blocks the path; on

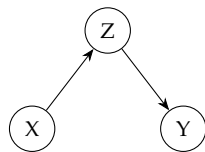
a **collider** ($X \rightarrow Z \leftarrow Y$), conditioning on Z opens a spurious one. In neuroscience the gold-standard intervention is **optogenetics**, using light to force a defined population on or off, a biological randomized trial that realizes the do-operator; its AI analog is **network ablation**, zeroing a unit to measure its causal contribution. When you cannot intervene, **instrumental variables** (a genetic variant affecting dopamine, say), regression with confounder control, and causal-discovery algorithms like **PC** recover effects from observational data.

Watch out. Adjusting for confounders only works for the confounders you measured; an unmeasured one biases the estimate in an unknown direction. And a causal-discovery algorithm outputs a hypothesis to test, never a proof.

ANALOGY

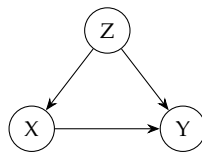
A rooster crows every morning just before dawn, perfectly correlated with sunrise. Silence the rooster (intervene) and the sun still rises. Correlation gave you the schedule; only the intervention

reveals there was never an arrow from bird to star.



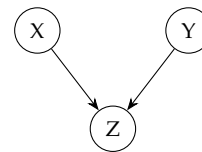
Chain

condition to block



Fork (confounder)

adjust for Z



Collider

never condition on Z

The three DAG shapes that govern adjustment. Conditioning on Z blocks a chain or fork but opens a collider, inventing a correlation from nothing.

Pillar	Question it answers	Neuro example	AI example
Information theory	How much does a signal carry?	Bits per spike; efficient coding	Cross-entropy loss; info bottleneck
Bayesian inference	How should belief change?	Cue integration; Bayesian brain	Priors; probabilistic models
GLM / fitting	What is the testable model?	Poisson GLM for spike trains	Regression, regularization, CV
Causal inference	Would acting change Y ?	Optogenetic intervention	Network ablation; IV, PC

Quick Self-Test

1. A colleague reports two brain signals have Pearson correlation near zero and concludes they are independent. Why is that conclusion unsafe, and what would you compute instead?
2. In one sentence, what does training a network by cross-entropy loss minimize, and what does that connect it to?
3. When the brain fuses a reliable visual cue and a noisy haptic cue, how should it weight them, and whose experiment confirmed this?
4. Training error keeps dropping as you add polynomial terms. Why does that not mean the bigger model is better, and what do you check instead?
5. State the difference between $P(Y | X)$ and $P(Y | do(X))$, and name the biological technique that realizes the do-operator.
6. You adjust for every confounder you measured and still worry the estimate is biased. What single unstated assumption could be failing?

If You Remember One Thing

These four pillars are one continuous argument, not four separate topics. Information theory says what a signal can carry, Bayesian inference says how to update belief about it, the GLM turns that belief into a model you can fit without fooling yourself, and causal inference tells you whether acting on the model would change anything. The seams between them are where the insight lives: cross-entropy is KL divergence, regularization is a prior, and an accurate predictive model is still the wrong tool for a question about intervention.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552

Neural Data, Ethics, and Safety

Study Guide

by Richard J. Young, Ph.D.

richard@deepneuro.ai

A cross-cutting concept guide · The NeuroAI Handbook

Why reading and writing the brain raises ethics no other AI faces, and how to govern it.

One-sentence thesis. NeuroAI is different because it does not just touch your data, it touches your mind, so privacy, consent, identity, and bias stop being compliance checkboxes and become a new class of human rights that need technical, clinical, and legal enforcement at once.

PART 1: Why the Mind Is a Special Case

Neural Data Is Not Ordinary Data

Every other privacy problem in AI concerns information *about* you, your purchases, your diagnoses, your movements. Neural data is different in kind: it is a recording of your thoughts, emotions, intentions, and memories, the raw substrate of the self. That is why the book calls the brain the last and most intimate frontier of privacy. A missed data point in a marketing model is an annoyance; a decoded intention is an exposure with no analog anywhere else in computing. And unlike a password or a credit-card number, a neural signature cannot be reissued after it leaks.

Because of this, the familiar frameworks are necessary but not sufficient. HIPAA and GDPR already treat neural recordings as sensitive protected health information, yet they were not designed for technologies that can infer mental states, so the book argues neural data deserves a special category of its own, on par

with genetic information.

The Five Neurorights

To fill the gap, ethicists, neuroscientists, and policy-makers (notably Yuste et al., 2017, and Ienca & Andorno, 2017) proposed **neurorights**: a new class of human rights protecting the mind from advanced neurotechnology. Five recur throughout the book, and they organize everything that follows: the right to **mental privacy**, to **personal identity**, to **free will**, to **fair access**, and to **protection from bias**. They are not abstractions. Each maps to a concrete technology already in use or on the near horizon, and each has a concrete safeguard.

Watch out. The stakes scale with the intimacy of the target. The same misuse that is merely embarrassing for shopping data becomes coercive, or medically dangerous, when the data is neural.

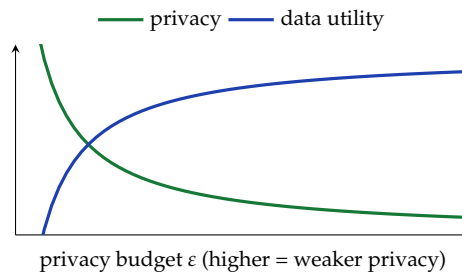
Neuroright	Protects	Concrete threat	Safeguard
Mental privacy	Neural data itself	Decoding thoughts, re-identification	Consent, differential privacy
Personal identity	Sense of self	DBS personality change	Reversibility, minimal intervention
Free will	Autonomous choice	Covert influence, soft coercion	Read-only default, kill switch
Fair access	Equity of benefit	The neuro-divide	Prioritize therapy over enhancement
Protection from bias	Equal accuracy	Seizure model fails minorities	Diverse data, bias audits

Protecting the Recording

The right to mental privacy is enforced by four principles the book states plainly: **informed consent** (explicit, ongoing, revocable), **data minimization** (collect only what the purpose needs), **de-identification** (strip identifiers), and **purpose limitation** (data gathered for diagnosis is not repurposed for marketing without fresh consent). But de-identification alone is weak, neural data is high-dimensional enough to re-identify, so the book turns to stronger, mathematical protection. **Differential privacy** adds calibrated noise so that the presence or absence of any one person barely changes any result. Its guarantee is a bound governed by a privacy budget ϵ :

$$P(M(D) \in S) \leq e^\epsilon P(M(D') \in S)$$

A smaller ϵ means stronger privacy but noisier, less useful data, a genuine tradeoff. **Federated learning** complements it: hospitals train a shared model on their own data and exchange only model updates, so raw recordings never leave the building. Best practice is defense in depth, combining all three.



The differential-privacy dial. Turning ϵ up buys usable data at the cost of protection; turning it down does the reverse. For high-dimensional neural data this balance is especially unforgiving.

ANALOGY

A leaked credit card is cancelled and reissued by morning. There is no reissuing a brain. Neural data, once decoded, is a key that never changes and cannot be revoked, which is why privacy for it must be engineered in before the first recording, not patched afterward.

Consent When the Interface Learns You

Consent for a brain-computer interface is harder than a signature on a form. The book insists a patient must understand five things: what data is collected, how it is used, who can access it, how long it is kept, and their unconditional right to withdraw. Two features of BCIs strain ordinary consent. First, modern decoders rely on **co-adaptation**, the user's brain and the AI model continuously reshaping each other, so the system a patient consented to is not the system running a month later. Second is the **autonomy paradox**: a BCI that restores communication to a locked-in patient also creates new dependencies (on the device, on caregivers) and new vulnerabilities (malfunction, hacking). Restoring agency and creating dependency are the same act.

When the Technology Writes Back

Reading the brain threatens privacy; *writing* to it threatens the self. Deep brain stimulation for Parkinson’s disease is well established and dramatically relieves motor symptoms, yet in rare cases patients report feeling like “different people,” more impulsive or less empathetic, and the neurostimulation chapter lists mood changes and impulse-control disorders among its side effects. Closed-loop systems sharpen the dilemma: an AI that continuously tunes neural activity to reduce anxiety might also dampen creative drive. At what point is a side effect a violation of the **right to personal identity**? The book’s answer is a set of design rules, minimal intervention, explicit consent for identity-relevant changes, reversibility, and respect for a patient’s refusal even of beneficial treatment.

The **right to free will** is next. A thought-to-speech decoder that lets a paralyzed person communicate could in principle decode *any* intended action, opening the door to employer engagement monitoring or pre-crime surveillance. Even without any device reading you, enhancement creates **soft coercion**: if colleagues use focus-boosting stimulation to work longer, can you really decline? Mitigations follow: transparency about when a device is stimulating, a user-controlled “cognitive kill switch,” read-only interfaces by default, and an outright ban on covert cognitive influence.

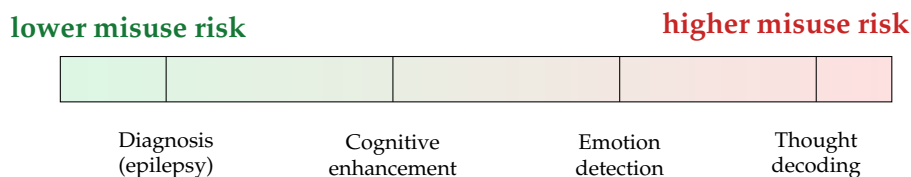
The Scalpel That Cuts Both Ways

These worries converge on the **dual-use dilemma**: the very capability that heals can harm. The neurostimulation toolkit, tDCS, TMS (FDA-approved for treatment-resistant depression), and DBS, is therapeutic, but the same working-memory and attention boosts that help a patient can widen a **neuro-divide** if only the wealthy can buy them, and brain stimulation aimed at “influencing decisions” is coercion. A BCI for ALS communication could decode thoughts without consent; an emotion detector built to screen for depression could target the vulnerable with precision manipulation.

Watch out. Dual-use is not a bug to be patched but a property of the technology, so safeguards must be structural: purpose limitation by design (a seizure detector should not be architected to also read emotion), audit trails, tiered access, and privacy-preserving, decentralized processing.

ANALOGY

The same scalpel that removes a tumor in a surgeon’s hand can wound in another’s. You do not ban scalpels; you license the surgeon, log the operation, and lock the drawer. Dual-use neurotech needs the same structural controls, not a hope that it will only be used well.



The dual-use spectrum. Applications slide from clearly beneficial to severe-risk; the higher-risk end demands access controls, audit trails, and explicit red lines, not a case-by-case guess.

The Prejudiced Machine, With Higher Stakes

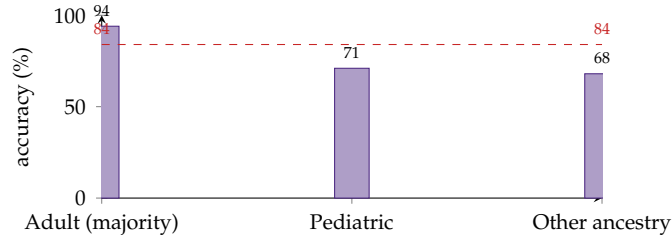
NeuroAI inherits every bias problem of ordinary AI and raises the cost. If a seizure detector is trained mostly on EEG from adult European patients, it can fail on children, whose brain activity differs, and underperform for other ancestries with different baseline patterns. The book's contrast is stark: a biased face recognizer might fail to unlock a phone, but a biased neural diagnostic can miss a seizure, and that is life-threatening. Studies consistently show medical AI trained on non-representative data underperforms for minority populations, so the **right to protection from bias** demands concrete practice: actively diverse datasets, bias audits that test every demographic subgroup separately, calibration to equal sensitivity across groups, and inclusive design teams. A model with 95 percent overall accuracy but 70 percent for one subgroup is not ready.

Governing a Fast-Moving Field

No single actor can secure NeuroAI; the book frames governance as a multi-stakeholder ecosystem, re-

searchers, regulators, clinicians, ethics boards, industry, and the public. Its concrete asks: classify neural data as a special protected category, mandate public algorithmic audits before clinical deployment, and draw explicit **red lines**, non-consensual thought decoding and cognitive manipulation should simply be prohibited. Because brain data crosses borders, weak jurisdictions invite a race to the bottom, so the book calls for international agreements analogous to those governing weapons of mass destruction. There is already precedent: in **2021** Chile became the first country to enshrine neurorights, mental privacy, free will, and equal access, in its constitution.

Watch out. Regulation lags innovation, so the book argues for *adaptive* governance, regulatory sandboxes, living guidelines, and anticipatory rules, plus ethics by design: privacy-preserving architectures, explainability, and fail-safes built in from the first line of code, not retrofitted before launch.



A seizure detector trained on one population. Headline accuracy hides a dangerous per-group gap; only a subgroup audit exposes it. The dashed line is a nominal deployment bar.

Quick Self-Test

1. Name two ways neural data differs from ordinary sensitive data, and why those differences make HIPAA and GDPR necessary but not sufficient.
2. List the five neurorights, and for any two, give a concrete technology that threatens it.
3. What does the privacy budget ϵ control in differential privacy, and what is the tradeoff as you lower it?
4. Explain the autonomy paradox of BCIs in one sentence.
5. Why does the book treat dual-use as a structural property rather than a fixable flaw, and what are two structural safeguards it recommends?
6. A neural diagnostic reports 95 percent overall accuracy. Why is that not enough to deploy, and what must you check first?

If You Remember One Thing

Ordinary AI ethics asks how to protect data about people; NeuroAI ethics asks how to protect people's minds, and that raises the stakes at every step, privacy you cannot re-issue, consent to a system that keeps changing, identity a device can quietly edit, and bias that can cost a life instead of an inconvenience. The five neurorights are the map, but they only work if the safeguards are built into the technology, the clinic, and the law at the same time, before the capability outruns the guardrails.

Richard J. Young, Ph.D. · richard@deepneuro.ai · Senior AI Research Scientist, UnitedHealth Group · UNLV Lee Business School · ORCID 0000-0002-1109-7552