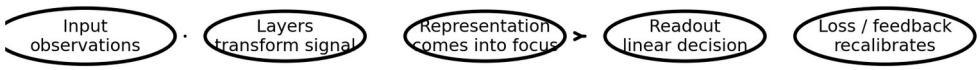
Companion Notes for “Minds Made of Math”

Memory notes for students and refresher notes for the slide designer

A history of neural networks from McCulloch & Pitts to the transformer era, with added mathematical detail, derivation sketches, and a unifying analogy.



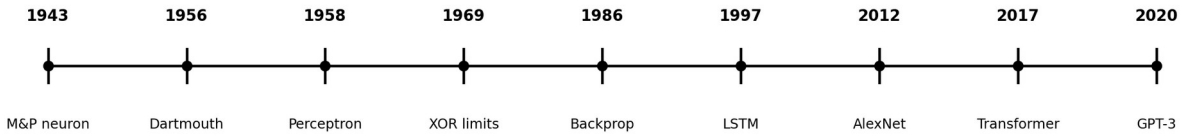
Unifying analogy: a neural network behaves like an adaptive optical system that gradually focuses task-relevant structure.

How to use these notes. Each section tracks one slide or short cluster of slides. Read the overview before class, use the “Key mathematics” block to anchor formal points, and use the “Why it matters” block to connect the math to the historical narrative.

Deck map and unifying idea

The deck can be understood as a sequence of three linked questions: what functions neural networks can represent, how those functions can be learned, and why increasing data and compute changed the practical answer so dramatically.

Unifying analogy: treat the network as an adaptive optical system. Early threshold units are crude apertures; multilayer networks create progressively more focused representations; convolution acts like a repeated lens array; attention behaves like an actively steerable spotlight; and gradient descent continually recalibrates the optics in response to error.



Era	Main technical problem	Key mathematical tool	Representative result
1943–1958	Can a neuron be formalized?	Threshold logic	Binary threshold unit
1958–1969	Can a machine learn from examples?	Linear separation and updates	Perceptron convergence

1974–1998	Can hidden layers be trained efficiently?	Chain rule and reverse-mode differentiation	Backpropagation
2006–2016	Why do deep nets work in practice?	SGD, ReLU, regularization	AlexNet / end-to-end features
2017–present	How can models use long context and scale?	Self-attention and scaling laws	Transformers and LLMs

The Biological Spark

Overview. This opening slide introduces the threshold neuron. McCulloch and Pitts showed that a neuron-like device can be modeled as a logical switch, which matters because it turns the study of mind into a study of computable structure. Hebb then supplied an early idea of learning, though not one tied to a global objective.

Key mathematics.

- Formal neuron: y = 1{w^T x >= theta}.
- Interpretation: the unit computes a half-space decision in input space.
- Important limitation: the activation is discontinuous, so gradient-based learning is not naturally available.

Why it matters. The historical insight is not that the 1943 model was realistic biology. It is that cognition could be represented as networks of simple formal operations.

Dartmouth, 1956

Overview. The Dartmouth proposal frames intelligence as something that can be described precisely enough for a machine to simulate. In classroom terms, this is the moment the field becomes explicit about computational representation and algorithm design.

Key mathematics.

- Two paths emerge: symbolic search over discrete structures and parameter optimization over continuous spaces.
- Neural network history repeatedly interacts with this divide rather than replacing it outright.

Why it matters. This slide matters because it shows that neural networks were never the whole field of AI; they were one answer to a broader question.

The Perceptron

Overview. Rosenblatt added learning to the formal neuron. The perceptron is a linear classifier trained from mistakes. Its importance comes from combining a geometric model with an explicit update rule.

Key mathematics.

- Classifier: f_w(x)=sign(w^T x).

- Update on a mistake: w_{t+1} = w_t + y_i x_i.
- Convergence theorem: if data are linearly separable with margin gamma and ||x_i|| <= R, the number of mistakes is bounded by (R/gamma)^2.

Why it matters. The theorem is strong but conditional. It proves learnability only when a separating hyperplane exists. That conditionality becomes central in the next slide.

The Minsky–Papert Winter

Overview. The famous XOR example shows that single-layer perceptrons cannot represent every Boolean function. The key lesson is that representational power and learnability are different questions.

Key mathematics.

- XOR is not linearly separable in R^2.
- A one-layer threshold model can only carve the input space with a single hyperplane.
- A multilayer model can represent XOR by mapping inputs into a new feature space where a linear separator becomes possible.

Why it matters. This slide is best remembered as a warning against overgeneralization: a limitation of one architecture was often mistaken for a limitation of all neural networks.

The Backpropagation Breakthrough

Overview. Backpropagation solves the training problem for multilayer differentiable networks by efficiently computing derivatives of the loss with respect to every parameter.

Key mathematics.

- If h_l = sigma(W_l h_{l-1}), define delta_l = dL/dz_l where z_l = W_l h_{l-1}.
- Backward recursion: delta_l = (W_{l+1}^T delta_{l+1}) ⊙ sigma'(z_l).
- Gradient: dL/dW_l = delta_l h_{l-1}^T.

Why it matters. Pedagogically, the deep idea is that composition creates power, and the chain rule makes that composition trainable.

What the 1980s Built

Overview. This cluster marks the move from theoretical possibility to early working systems such as LeNet and recurrent networks. It also introduces the first major optimization pathology of depth.

Key mathematics.

- Universal approximation: a one-hidden-layer network with enough units can approximate broad classes of continuous functions on compact sets.
- But approximation is not the same as efficient representation, successful optimization, or robust generalization.

- Vanishing gradients arise because repeated multiplication of derivatives smaller than one shrinks the learning signal exponentially.

Why it matters. Students should remember that the field stalled not because the ideas were empty, but because data, compute, and optimization were not yet aligned.

The Deep Learning Resurgence

Overview. The third wave happens when larger datasets, GPUs, and training refinements all arrive together. Deep learning becomes an engineering system rather than a fragile demonstration.

Key mathematics.

- Empirical risk minimization: minimize (1/n) sum_i l(f_theta(x_i), y_i).
- SGD update: theta_{t+1} = theta_t - eta_t ∇_theta l_i(theta_t).
- ReLU, better initialization, dropout, and batch normalization reshape optimization enough to make depth practical.

Why it matters. AlexNet is not just a benchmark victory. It is evidence that representation learning can outperform handcrafted features at scale.

What Deep Learning Actually Does

Overview. This section explains representation learning. A network seeks a map phi_theta(x) so that difficult tasks become simpler in the learned representation space.

Key mathematics.

- Readout form: y_hat = sign(w^T phi_theta(x)) for a linear probe.
- Depth can yield exponential gains in representational efficiency for some function classes compared with shallow models.
- End-to-end learning means features are optimized for the task loss instead of designed in advance.

Why it matters. This is the best slide for reminding students that modern networks are not magic feature detectors; they are learned coordinate systems.

Attention Is All You Need

Overview. The transformer replaces recurrence with attention, allowing every token to interact with every other token in one layer.

Key mathematics.

- Scaled dot-product attention: Attn(Q,K,V) = softmax(QK^T / sqrt(d_k)) V.
- Q, K, and V are learned linear projections of the same sequence representation.
- Attention produces data-dependent mixing weights rather than fixed local connectivity.

Why it matters. The move here is conceptual as much as computational: relevance is inferred on the fly from content, not from fixed position alone.

Why Attention Works

Overview. This slide makes the previous equation intuitive. A token builds its meaning by consulting context and weighting the context according to current need.

Key mathematics.

- Softmax produces a normalized distribution over the context positions.
- Multi-head attention lets different subspaces capture different relations at the same time.
- Compared with an RNN, the effective path length between distant positions is shorter, which helps gradient flow and parallelization.

Why it matters. Remember the two gains: parallel computation during training and flexible context selection during representation building.

From Transformer to LLM

Overview. This section adds scale. The transformer architecture keeps improving as parameters, data, and compute grow together.

Key mathematics.

- Autoregressive objective: maximize sum_t log p_theta(x_t | x_{<t}).
- Scaling laws fit loss curves with approximate power-law behavior over broad ranges.
- Alignment methods such as RLHF reshape a pretrained predictor into a more instruction-following assistant.

Why it matters. The main teaching point is that scale did not replace the earlier math. It amplified it.

The Through-Line

Overview. This is the synthesis slide. It should feel like the whole deck collapsing to a single template: model class + loss + optimization + data + compute.

Key mathematics.

- Master pattern: theta* = argmin_theta E_{(x,y)~D}[l(f_theta(x), y)].
- Each era changes the parameterization, the inductive bias, or the compute regime more than it changes the fundamental learning template.
- What looked like separate revolutions were often changes in one or two pieces of this template.

Why it matters. For the designer, this is the anchor slide that can organize transitions, recap, and discussion.

Open Questions

Overview. The deck closes by being intellectually honest. Capability has outpaced complete theory.

Key mathematics.

- Why exactly do scaling laws hold, and when do they break?
- How much of LLM performance is true systematic reasoning rather than sophisticated pattern completion?
- How can alignment, interpretability, and robustness keep pace with capability growth?

Why it matters. This ending works best if it leaves students with a live research frontier rather than a triumphalist narrative.

Short derivation sketches

Perceptron mistake bound

Assume a unit vector u separates the data with margin gamma, so y_i u^T x_i >= gamma. After each mistake, u^T w increases by at least gamma, while ||w||^2 can increase by at most R^2. Combining M gamma <= ||w_M|| <= R sqrt(M) gives M <= (R/gamma)^2.

Backpropagation in one line

Write the network as a composition and apply the chain rule from the loss backward through each intermediate variable. Reverse-mode automatic differentiation is efficient because each local derivative is reused across many upstream parameters.

Why gradients vanish in deep or recurrent nets

If each local Jacobian has norm below 1, then a product of many Jacobians shrinks roughly exponentially with depth or time. Early layers then receive almost no training signal, which is why activation choice, gating, normalization, and residual structure matter.

Convolutional equivariance intuition

Convolution applies the same kernel at every location. Shifting the input shifts the outputs in the same way, apart from boundary and padding effects. This is the mathematical reason CNNs embed translation structure.

Self-attention as weighted averaging

For a fixed query, dot products with keys determine relevance scores; softmax converts these scores into nonnegative weights summing to 1; the output is then a weighted sum of the value vectors. The operation is linear in V once the weights are fixed, but nonlinear overall because the weights depend on Q and K.

Presenter and designer checklist

- Keep the adaptive-optics analogy visible at transitions so the visuals feel like one story rather than separate mini-lectures.
- When explaining a theorem, state both the result and the assumptions; the assumptions are often the real historical lesson.
- Use the perceptron/XOR/backprop trio as the narrative hinge of the whole talk.
- On the transformer slides, connect the formula to a sentence-level example before discussing scale.
- On the final slide, emphasize that capability and theory have advanced at different speeds.

Selected references named in the deck

- McCulloch, W. S., and Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity.
- McCarthy, J., Minsky, M., Rochester, N., and Shannon, C. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence.
- Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.
- Minsky, M., and Papert, S. (1969). Perceptrons.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Representations by Back-Propagating Errors.
- Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning.
- Vaswani, A., et al. (2017). Attention Is All You Need.
- Kaplan, J., et al. (2020). Scaling Laws for Neural Language Models.