CPEN 455: Deep Learning

Lecture 1: Introduction to Deep Learning

Renjie Liao

University of British Columbia

Winter, Term 1, 2026

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics
- Course Roadmap
- Reading Deep Learning Research

Outline

- **Course Information**
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics
- Course Roadmap
- Reading Deep Learning Research

Course Information

- [Course website: https://lrjconan.github.io/UBC-CPEN455-DL/](https://lrjconan.github.io/UBC-CPEN455-DL/)
- Deep learning foundations, generative models, and LLM post-training
- Assumes basic knowledge of calculus, linear algebra, and probability
- The first homework will help you review the mathematical background
- Assumes familiarity with Python programming for implementing models
- Assumes familiarity with LaTeX for writing homework and project reports

Course Information

- Lectures: Wed. & Fri., 1:00–2:30 p.m., September 9–December 4, 2026
- Tutorials: Mon., 9:00–10:00 a.m., September 14–December 7, 2026
- Both meet at UBCV, Food, Nutrition and Health Building (FNH), Floor -1, Room 60
- Office hour: Tuesday, 2:00–3:00 p.m., KAIS 3047 (Ohm)
- TA: Felix Fu
- Lectures are in person without recording. See the syllabus for holidays and the midterm break.

Course Information

- Tutorials cover PyTorch, model implementation, reproducible experiments, and discussion of homework and assignments.
- Use Piazza for course discussion and questions:

  https://piazza.com/ubc.ca/winterterm12026/cpen455

- Use Canvas for submissions and course announcements:

  CPEN_V 455 101 2026W1 Deep Learning

- Participation includes useful questions, thoughtful answers, and constructive discussion in class or on Piazza.

Course Information

- Expectation & Grading:
  - [20%] 2 × Homework
  - [30%] 3 × Programming Assignments
  - [20%] 2 × In-class Quizzes
  - [30%] Individual Course Project
  - Participation: up to 3 extra percentage points, with the final grade capped at 100%

  All work is individual. Project details will be released near the middle of the term.

Course Policies

- Work individually. Discuss concepts, but develop your own solutions and code and acknowledge sources.
- Submit through Canvas by 11:59 p.m., Vancouver time, unless assessment instructions specify otherwise.
- Late work: up to 72 hours, 10% off the assessment's earned mark. Over 72 through 120 hours, 20% off. Over 120 hours, zero. Penalties are not cumulative.
- Two quizzes, near midterm and later, both before November 27. Dates, topics, and permitted materials will be announced one class ahead.
- Contact the instructor privately about concessions. Approved concessions take precedence. A missed quiz without an approved concession receives zero.

Full course policies

Generative AI: Optional Use

- Homework: general concept explanations and editing your own writing. No generated solutions or derivations for assigned questions.
- Programming assignments: syntax explanations and debugging hints for your code. Implement assessed algorithms yourself unless instructions explicitly permit generated implementations.
- Project: brainstorming, editing your own writing, and debugging. Broader permissions will appear in the brief. Conduct your own experiments and interpret the results.
- Declare the tool, purpose, and affected parts in each homework, programming assignment, and project submission, or state no use. No GenAI during quizzes.
- Keep relevant prompts and outputs and be able to explain your work. Grading concerns may lead to follow-up questions and requests for relevant records.

[Full course policies]

Course Information

- Compute for course work
  - [Google Colab: https://colab.research.google.com/](https://colab.research.google.com/)
  - Colab provides hosted notebooks. Free GPU availability, hardware, and usage limits vary. Save checkpoints regularly.
  - Cloud GPU virtual machines may require paid billing and quota approval. Google Cloud Free Trial accounts cannot add GPUs.
  - Debug small models on subsets of the data using a CPU or an available GPU.
  - Use focused experiments, mixed precision when appropriate, and checkpoints to stay within your compute budget.

Outline

- Course Information
- **Introduction to Deep Learning**
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics
- Course Roadmap
- Reading Deep Learning Research

What is Deep Learning?

- Deep learning uses multi-layer neural networks to learn representations from data.
- Data: labeled and unlabeled examples, including text, images, audio, and video
- Models: neural networks such as CNNs and transformers
- Learning: back-propagation computes gradients. Optimizers such as SGD and AdamW use these gradients to update parameters.

Outline

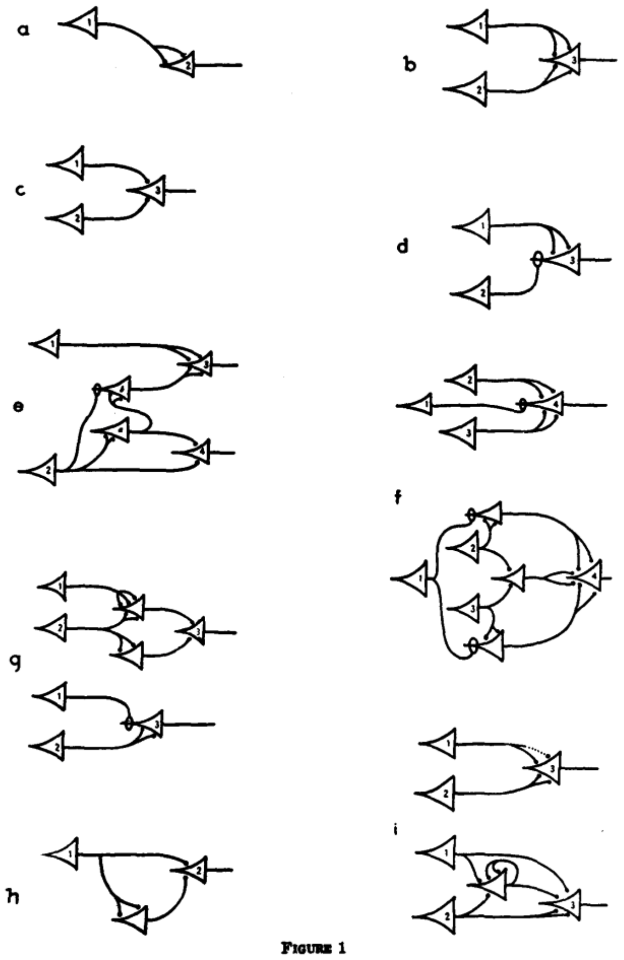
- Course Information
- Introduction to Deep Learning
  - **History**
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics
- Course Roadmap
- Reading Deep Learning Research

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)



Threshold Logic Unit (TLU):

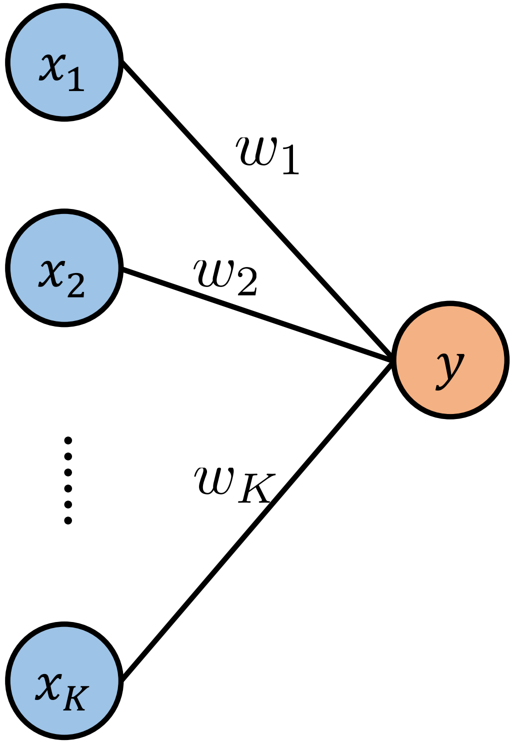
- Binary input and output
- Heaviside step function

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)

History of Deep Learning (Connectionism)

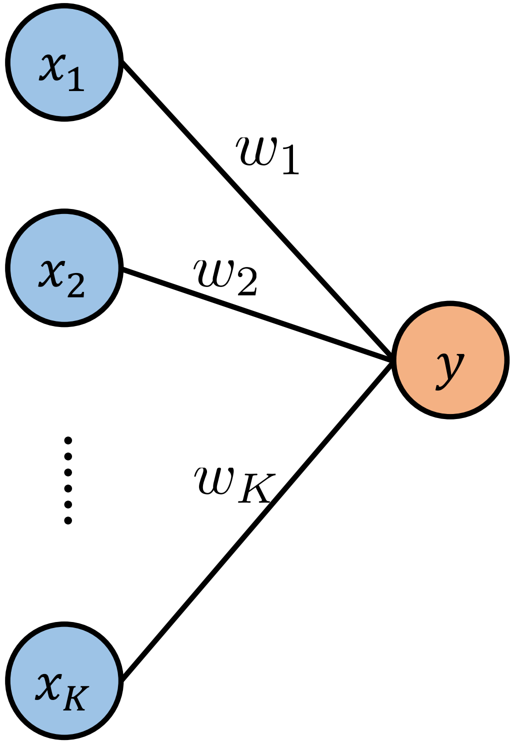
- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)



Linear Unit: $$y = \sum_{k=1}^{K} w_k x_k$$

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)

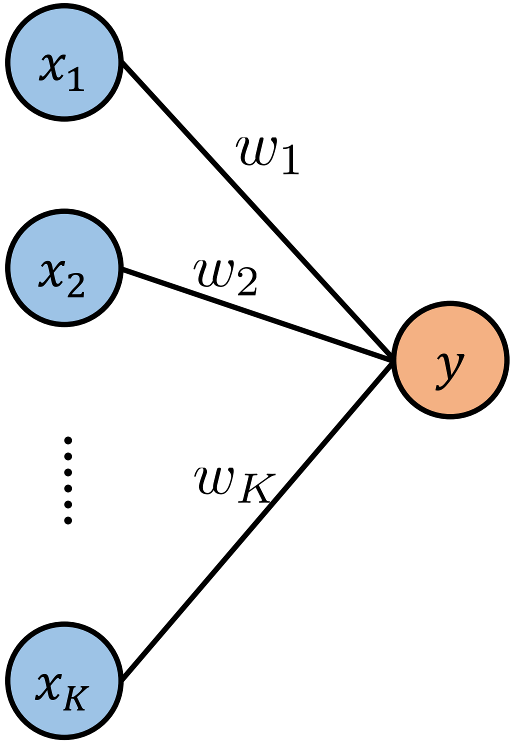


Linear Unit: $y = \sum_{k=1}^{K} w_k x_k$

Learning Rule: $w_k = w_k + \boxed{\eta \mathbb{E}\left[y x_k\right]}$

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)



Linear Unit: $$y = \sum_{k=1}^{K} w_k x_k$$

Learning Rule: $$w_k = w_k + \boxed{\eta \mathbb{E}\left[y x_k\right]}$$

Cells that fire together wire together!

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)

In 1951, Marvin Minsky and Dean Edmunds build SNARC (Stochastic Neural Analog Reinforcement Calculator), the first artificial neural network, using 3000 vacuum tubes to simulate a network of 40 neurons.

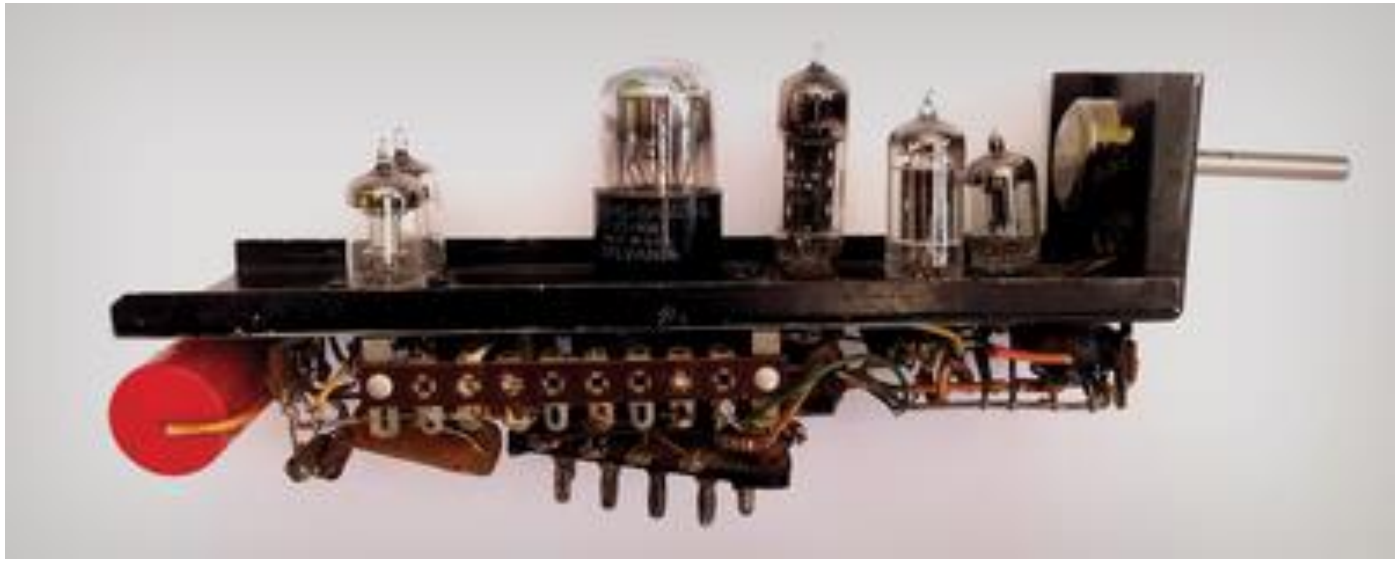


Image Credit: https://www.the-scientist.com/foundations/machine--learning--1951-65792

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)

History of Deep Learning (Connectionism)

- Artificial Neurons ([McCulloch and Pitts 1943]())
- Hebbian Learning Rule ([Donald Hebb 1949]())
- SNARC ([Minsky and Edmunds 1951]())
- Perceptron ([Frank Rosenblatt 1958]())

Frank Rosenblatt working on the Mark I Perceptron (1956).

Mark I Perceptron can classify 20x20 images.

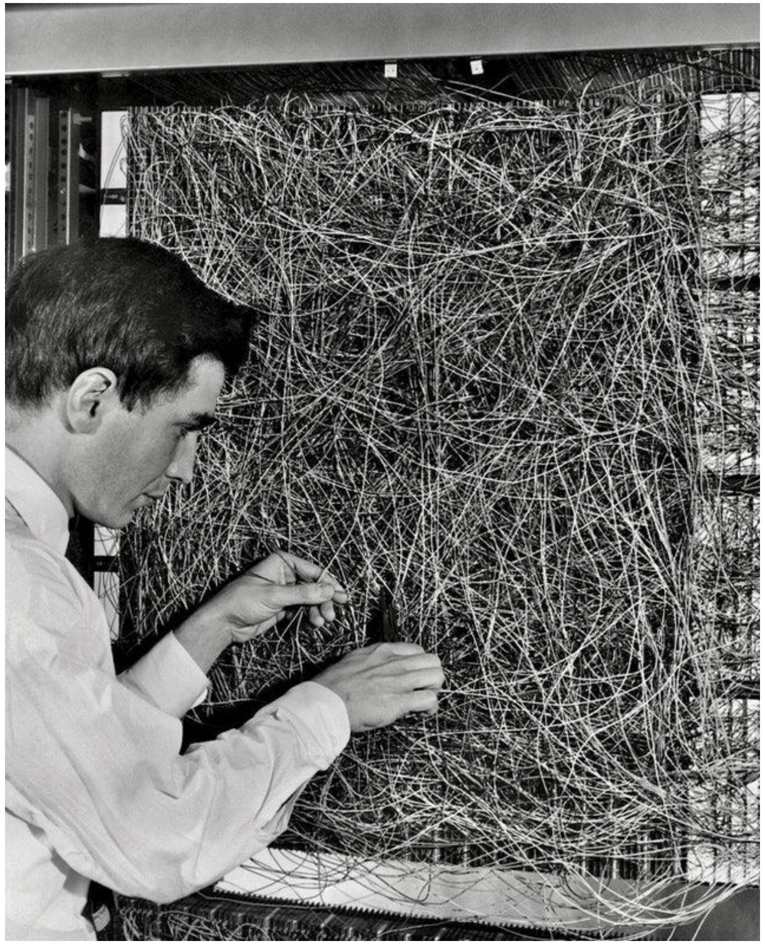


Image Credit: https://www.reddit.com/r/interestingasfuck/comments/e8a8oy/frank_rosenblatt_with_a_mark_i_perceptron/

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)
- Discovery of orientation selectivity and columnar organization in the cat’s visual cortex (Hubel and Wiesel 1959)

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)
- Discovery of orientation selectivity and columnar organization in the cat's visual cortex (Hubel and Wiesel 1959)

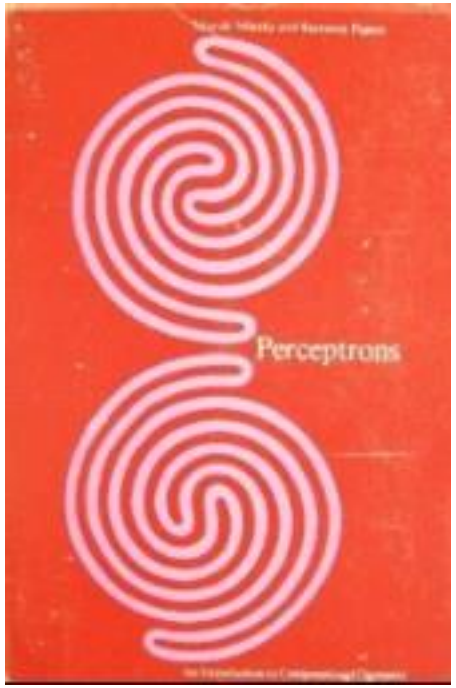
Their breakthrough discoveries about the visual system and visual processing earned them the Nobel Prize for Physiology or Medicine in 1981.

Image Credit: https://braintour.harvard.edu/archives/portfolio-items/hubel-and-wiesel

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)

In 1969, Marvin Minsky and Seymour Papert publish a book, “*Perceptrons: An Introduction to Computational Geometry*”, highlighting the limitations of simple neural networks, e.g., Perceptrons can not solve XOR problem.



This book contributed to the so-called AI winter of the 1980s.

Image Credit: http://harveycohen.net/image/perceptron.html

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)

In 1969, Marvin Minsky and Seymour Papert publish a book, “*Perceptrons: An Introduction to Computational Geometry*”, highlighting the limitations of simple neural networks, e.g., Perceptrons can not solve XOR problem.

This book contributed to the so-called AI winter of the 1980s.

Can you tell the difference between two shapes in the cover?

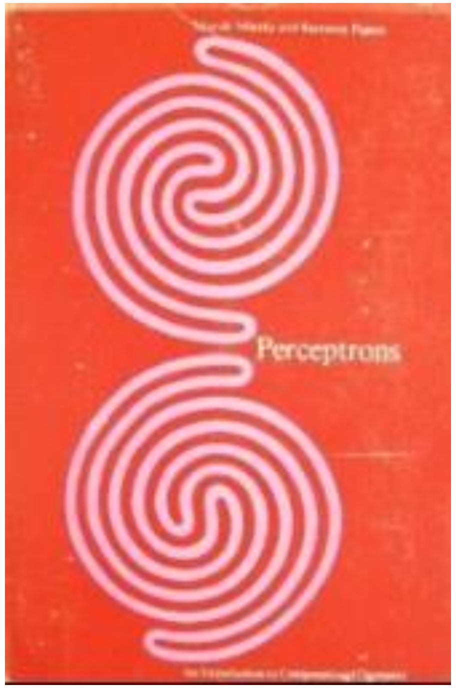


Image Credit: http://harveycohen.net/image/perceptron.html

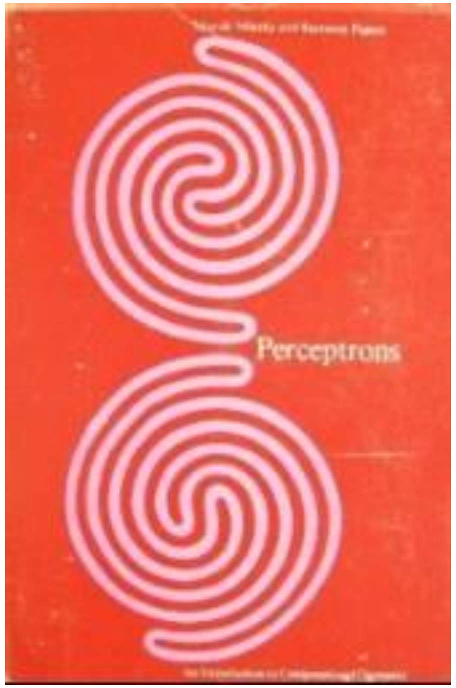
History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)

In 1969, Marvin Minsky and Seymour Papert publish a book, "*Perceptrons: An Introduction to Computational Geometry*", highlighting the limitations of simple neural networks, e.g., Perceptrons can not solve XOR problem.

This book contributed to the so-called AI winter of the 1980s.

Can you tell the difference between two shapes in the cover?



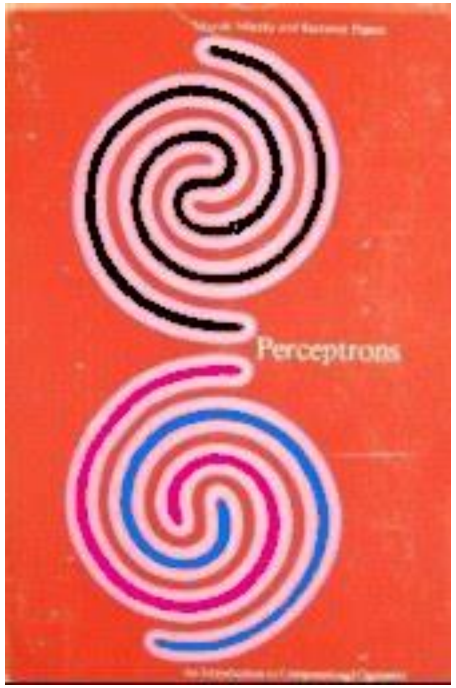


Image Credit: http://harveycohen.net/image/perceptron.html

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)

Image Credit: https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)

Inspired by the model proposed by Hubel & Wiesel in 1959, Fukushima proposed the first convolutional neural network architecture for Japanese handwritten character recognition.

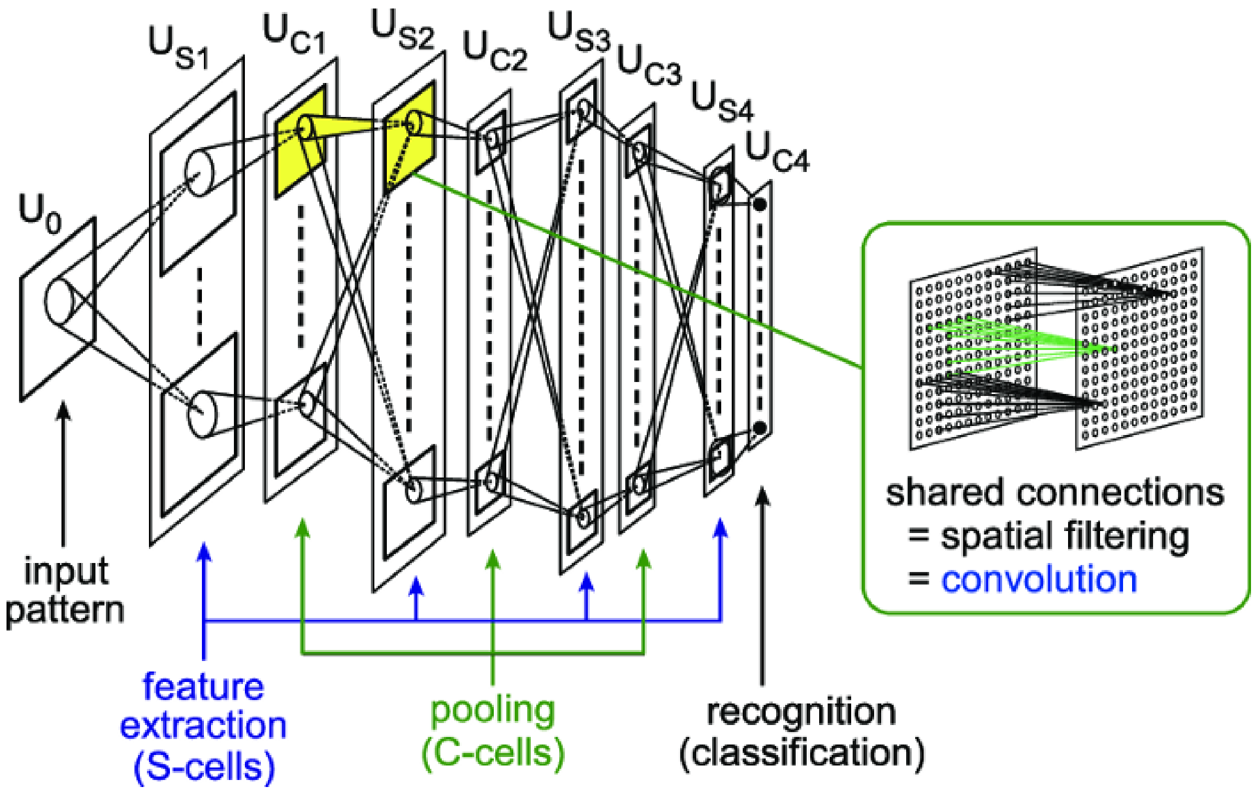


Image Credit: https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)

Image Credit: https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)

Inspired by Ising model in statistical physics, it consists of fully-connected variables with deterministic binary states and an energy function.

It is a recurrent neural network (RNN).

It learns to memorize data via energy minimization, thus being able to simulate associative memory.

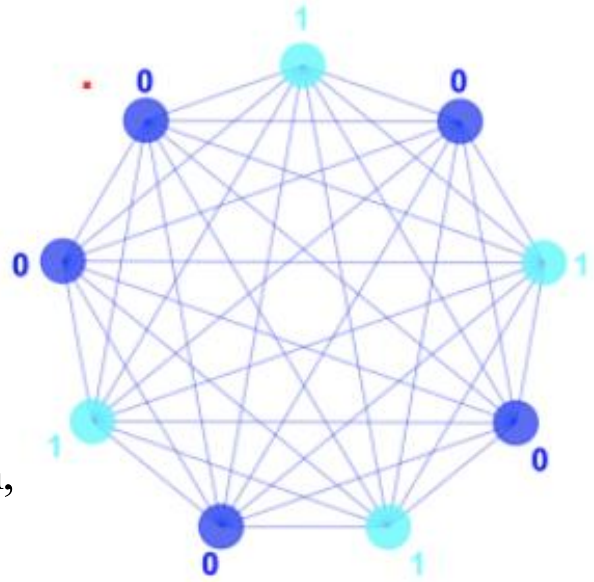


Image Credit: https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073

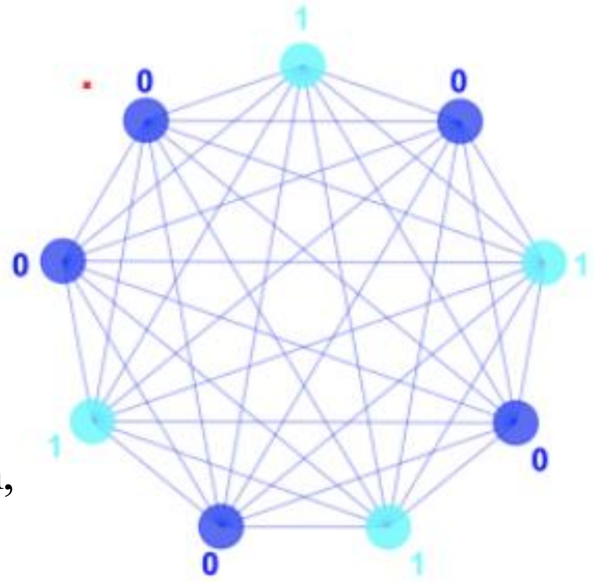
History of Deep Learning (Connectionism)

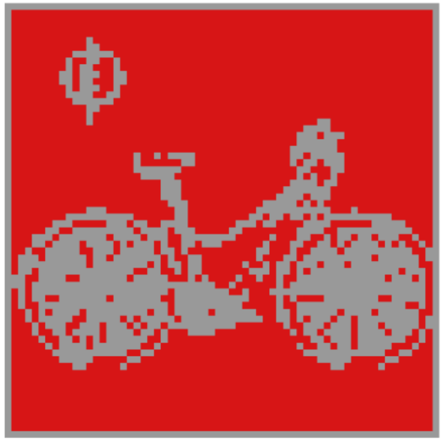
- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)

Inspired by Ising model in statistical physics, it consists of fully-connected variables with deterministic binary states and an energy function.

It is a recurrent neural network (RNN).

It learns to memorize data via energy minimization, thus being able to simulate associative memory.





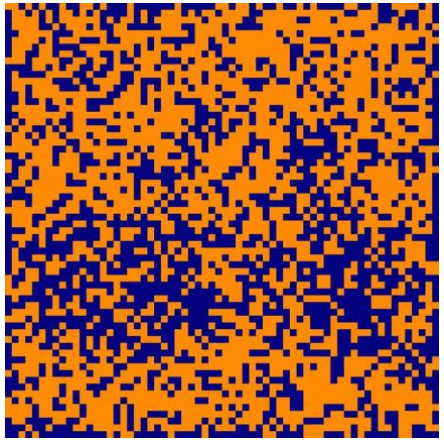


Image Credit: https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)

It generalizes Hopfield Networks by introducing 1) stochastic binary states 2) hidden/latent variables

The study of deep (layer-structured) Boltzmann machines led to the inception of deep learning in 2006.

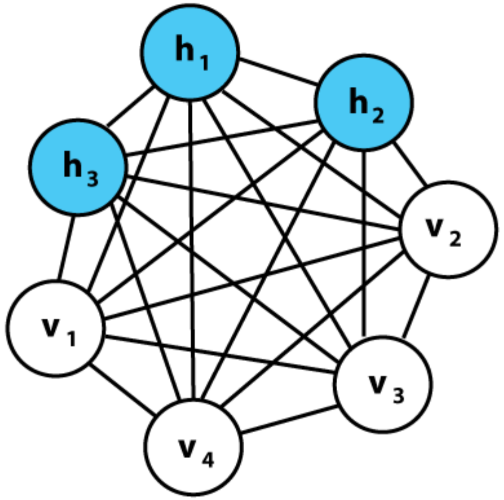


Image Credit: https://en.wikipedia.org/wiki/Boltzmann_machine

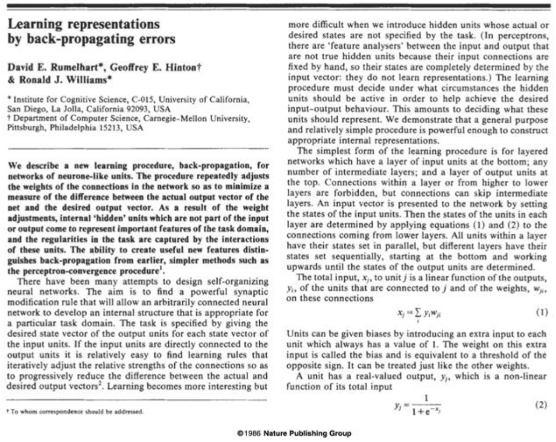
History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)

  Back-propagation efficiently computes gradients for training neural networks.



Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the desired state vector of the output units for each state vector of the input units. If the input units are directly connected to the output units it is relatively easy to find learning rules that iteratively adjust the relative strengths of the connections so as to progressively reduce the difference between the actual and desired output vectors[2]. Learning becomes more interesting but more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input-output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j, to unit j is a linear function of the outputs, y_i, of the units that are connected to j and of the weights, w_{ji}, on these connections

$$x_j = \sum_i y_i w_{ji} \qquad (1)$$

Units can be given biases by introducing an extra input to each unit which always has a value of 1. The weight on this extra input is called the bias and is equivalent to a threshold of the opposite sign. It can be treated just like the other weights.

A unit has a real-valued output, y_j, which is a non-linear function of its total input

$$y_j = \frac{1}{1+e^{-x_j}} \qquad (2)$$

† To whom correspondence should be addressed.

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)

LeCun's CNN demo from 1993

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)
- Long short-term memory (Hochreiter and Schmidhuber 1997)

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)
- Long short-term memory (Hochreiter and Schmidhuber 1997)

It partially resolves the vanishing gradient problem in training RNNs!

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Mnih et al. 2015), AlphaGo (Silver et al. 2016)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Mnih et al. 2015), AlphaGo (Silver et al. 2016)
- Breakthrough in natural language processing: Seq2seq (Sutskever et al. 2014), Transformers (Vaswani et al. 2017), GPT-3 (Brown et al. 2020), ChatGPT (OpenAI 2022)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Mnih et al. 2015), AlphaGo (Silver et al. 2016)
- Breakthrough in natural language processing: Seq2seq (Sutskever et al. 2014), Transformers (Vaswani et al. 2017), GPT-3 (Brown et al. 2020), ChatGPT (OpenAI 2022)
- Breakthrough in protein structure prediction: AlphaFold (Jumper et al. 2021)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Mnih et al. 2015), AlphaGo (Silver et al. 2016)
- Breakthrough in natural language processing: Seq2seq (Sutskever et al. 2014), Transformers (Vaswani et al. 2017), GPT-3 (Brown et al. 2020), ChatGPT (OpenAI 2022)
- Breakthrough in protein structure prediction: AlphaFold (Jumper et al. 2021)
- Breakthrough in LLM reasoning: DeepSeek-R1 (DeepSeek-AI et al. 2025); Gemini Deep Think reaches IMO gold-medal (Luong and Lockhart 2025)

The future depends on some graduate student who is deeply suspicious of everything I have said.

— Geoffrey Hinton

History of Deep Learning (Connectionism)

Yann LeCun, Geoffrey Hinton, and Yoshua Bengio received the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.

Image Credit: https://www.nytimes.com/2019/03/27/technology/turing-award-ai.html

History of Deep Learning (Connectionism)

The Nobel Prize in Physics 2024 was awarded jointly to John J. Hopfield and Geoffrey E. Hinton "for foundational discoveries and inventions that enable machine learning with artificial neural networks"

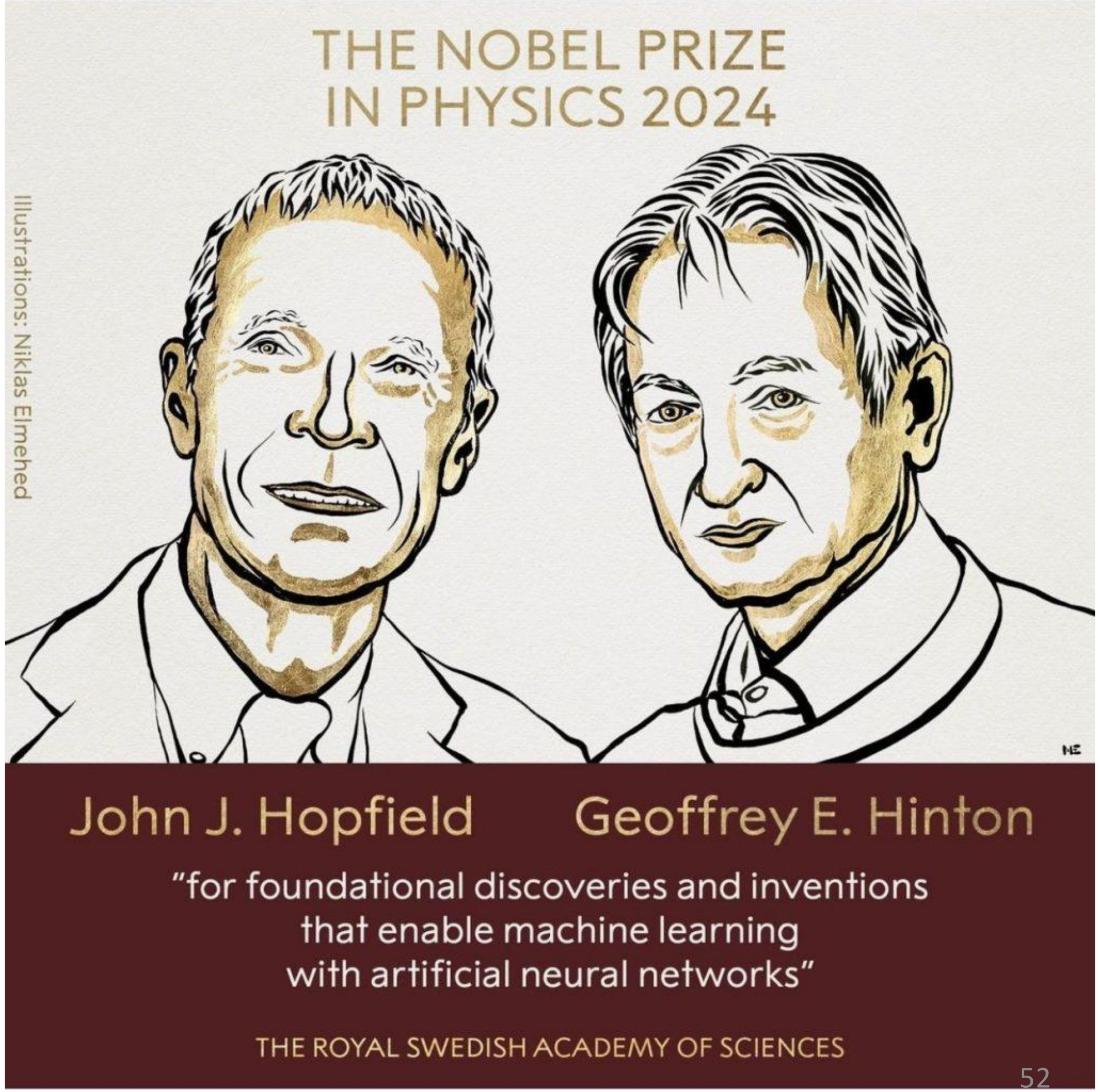


Image Credit: The Royal Swedish Academy of Sciences

History of Deep Learning (Connectionism)

The Nobel Prize in Chemistry 2024 was divided, one half awarded to David Baker "for computational protein design", the other half jointly to Demis Hassabis and John Jumper "for protein structure prediction"

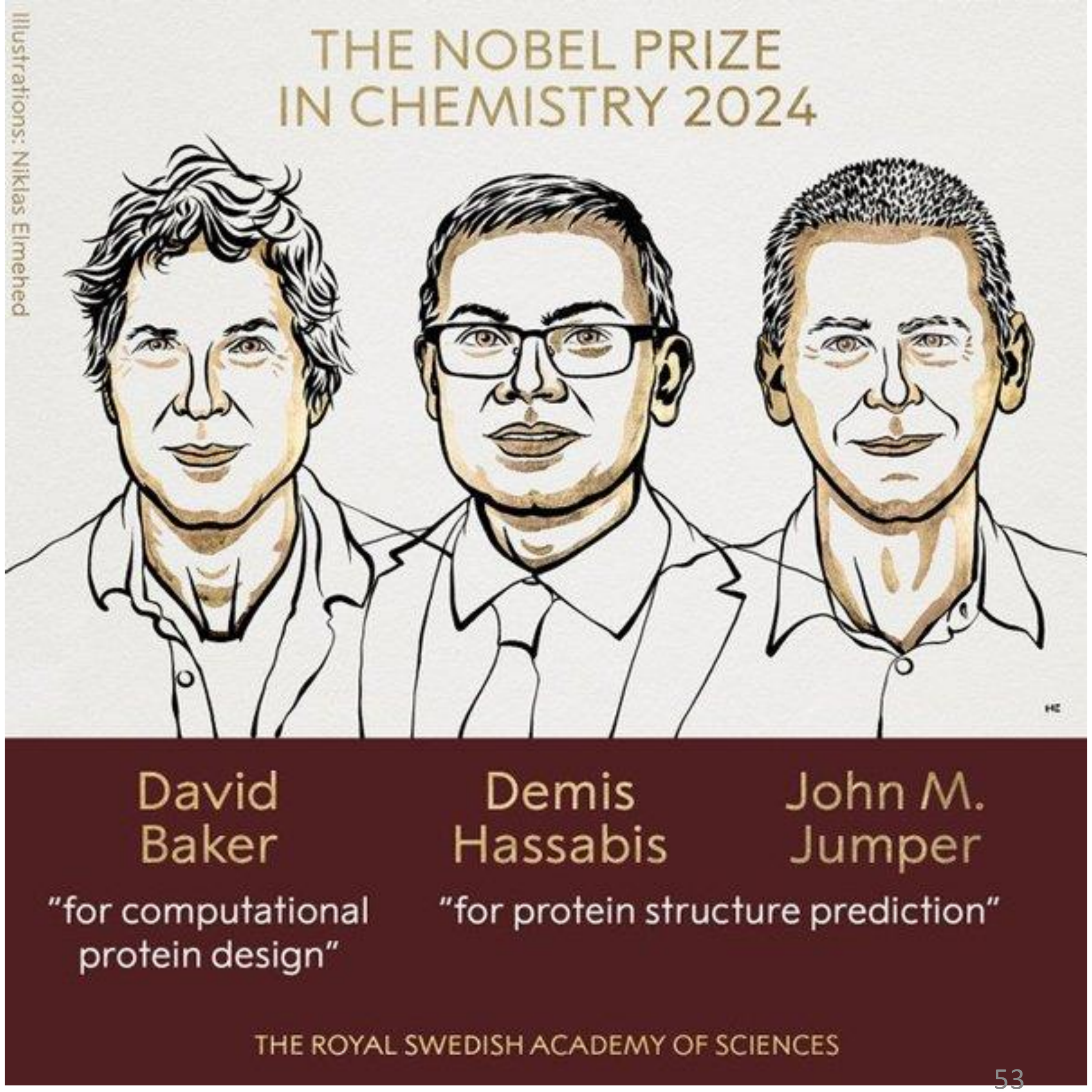


Image Credit: The Royal Swedish Academy of Sciences

Outline

- Course Information
- Introduction to Deep Learning
    - History
    - **Modern Applications**
    - Taxonomy & Connections to ML/AI/Statistics
- Course Roadmap
- Reading Deep Learning Research

Applications of Deep Learning

Self-Driving Cars (Waymo, 2026)

Perception and planning in London traffic.

Testing with trained specialists behind the wheel.

[Play full video]

[Source and video: Waymo, London autonomous testing (April 2026).]

Applications of Deep Learning

Interactive World Models (Project Genie, 2026)

Generate and explore worlds from text and images.

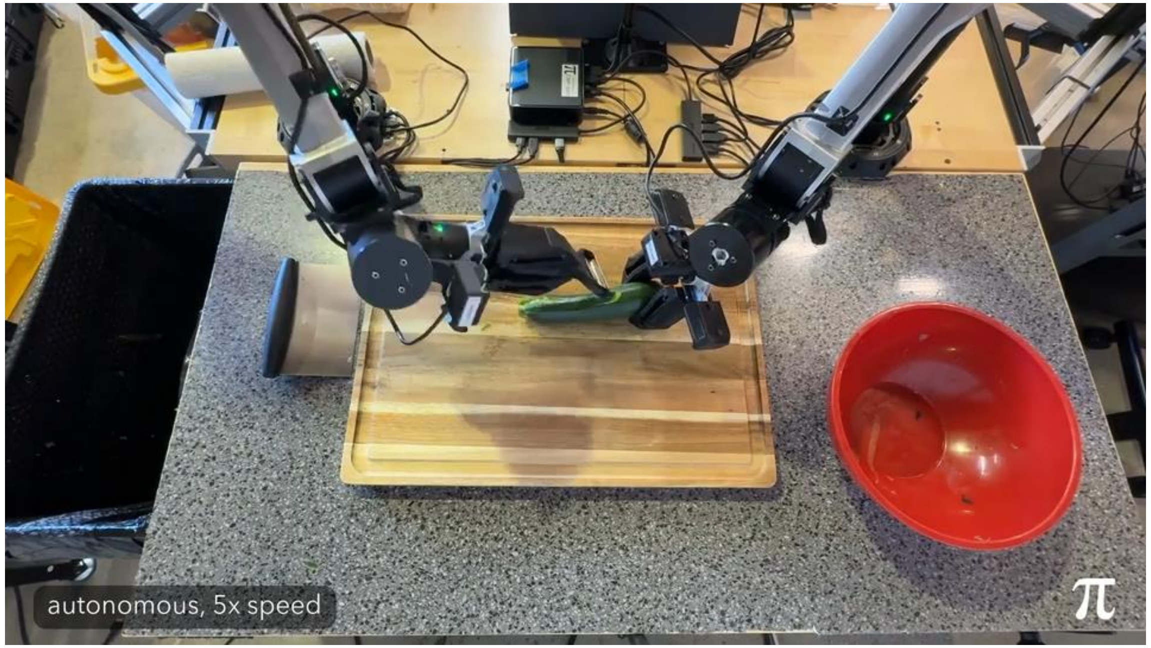
The model predicts what happens as the user moves.

Experimental worlds can still be physically inconsistent.

[Demo: Google DeepMind / Google Labs, Project Genie (2026), powered by Genie 3 (2025)]

Applications of Deep Learning

Robotics (Physical Intelligence $\pi_{0.7}$, 2026)



One vision-language-action model controls diverse manipulation tasks.

Click either image to play its video.

Video: Physical Intelligence, $\pi_{0.7}$ (April 2026), folding jeans
Video: Physical Intelligence, $\pi_{0.7}$ (April 2026), peeling zucchini

Applications of Deep Learning

Humanoid Robots and Embodied / Physical AI

Figure Helix 02 (2026)

Spot (2026 report)

Humanoid describes a human-like body.

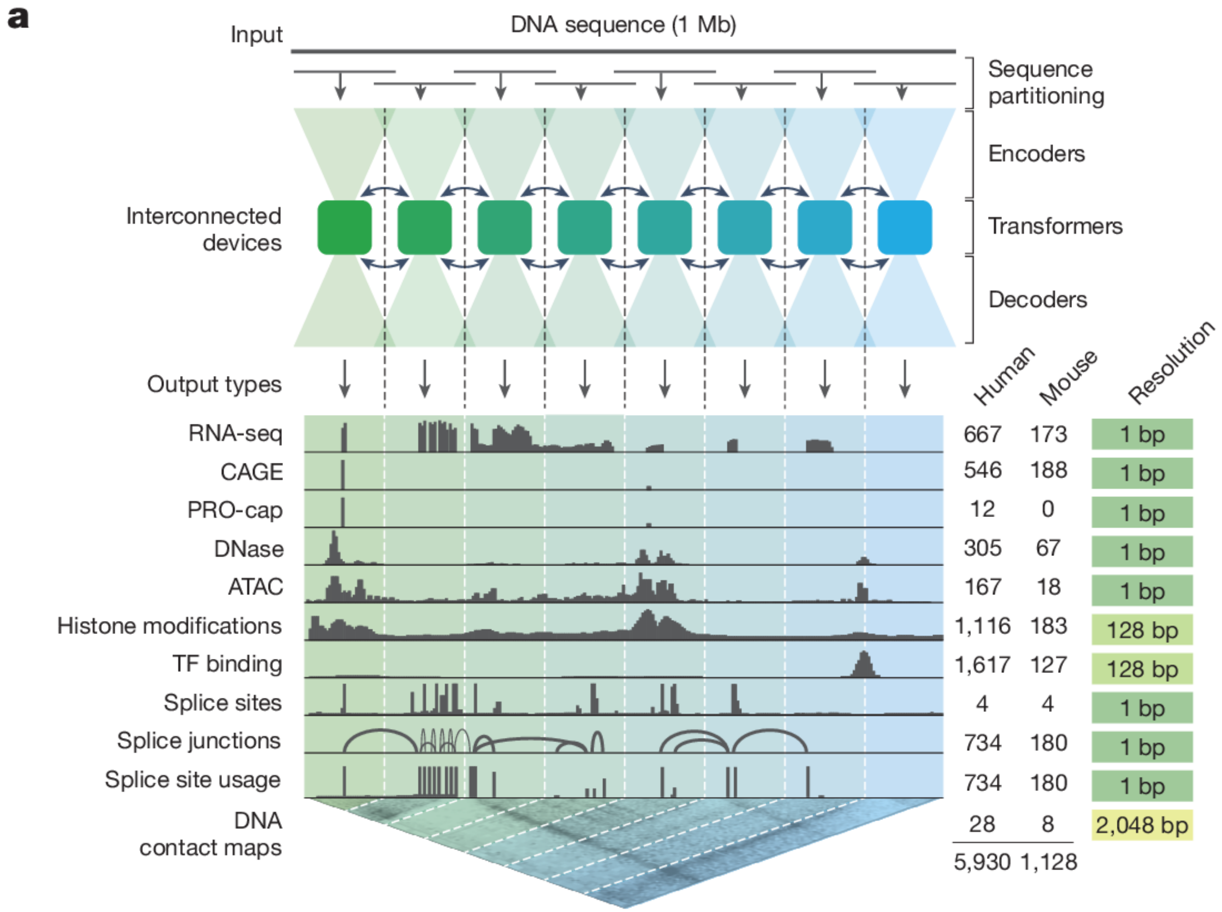
Embodied / physical AI connects perception and action through a body, including arms, quadrupeds, and vehicles.

[Video: Figure, Helix 02 Bedroom Tidy (May 2026)]

[Image: Boston Dynamics, Spot and Gemini Robotics-ER 1.5 (April 2026 report)]

Applications of Deep Learning

Genome Interpretation (AlphaGenome, 2026)



Predict gene activity and other molecular signals from one million DNA letters.

AlphaGenome Atlas (Sept. 2026) precomputes predicted effects for 9 billion single-letter variants.

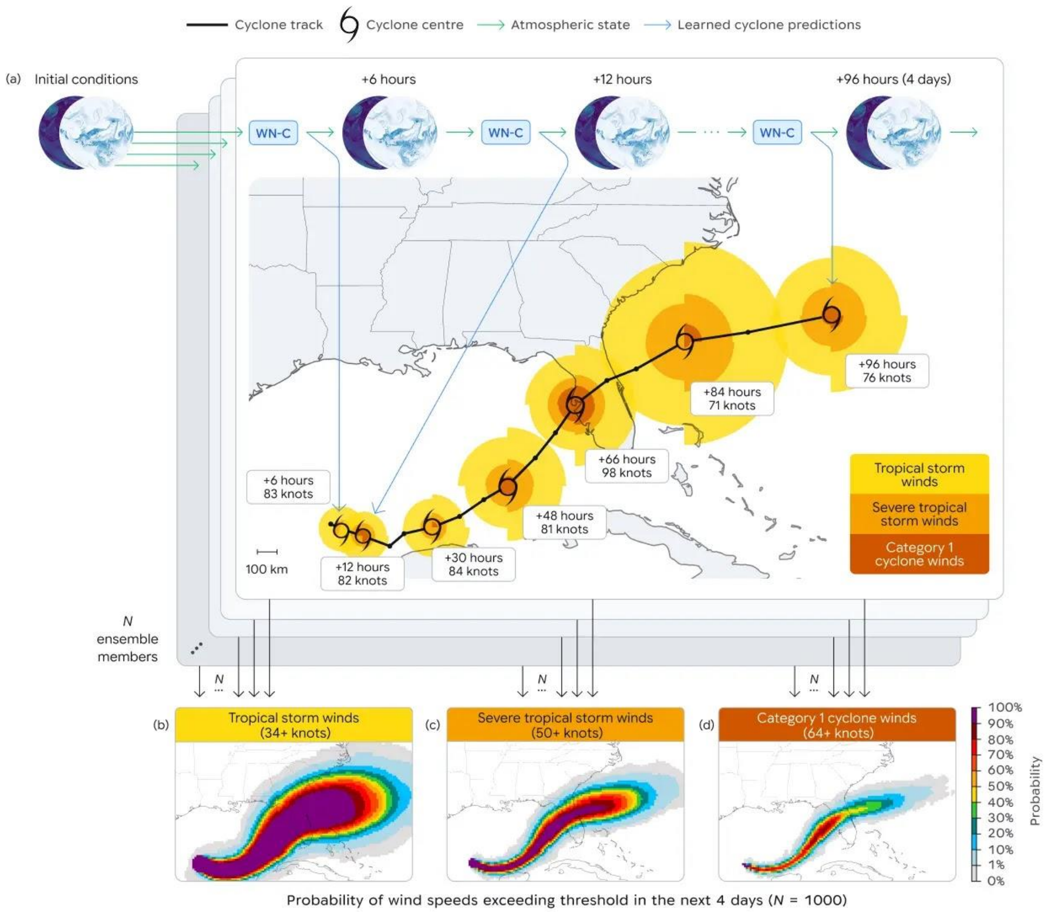
Research predictions help prioritize biological experiments.

[Figure: Avsec et al., Nature (2026), Fig. 1a (cropped), CC BY 4.0]

[Atlas update: Google DeepMind (8 September 2026)]

Applications of Deep Learning

Weather Forecasting (WeatherNext Cyclones, 2026)



Predict cyclone paths, intensity and wind structure together.

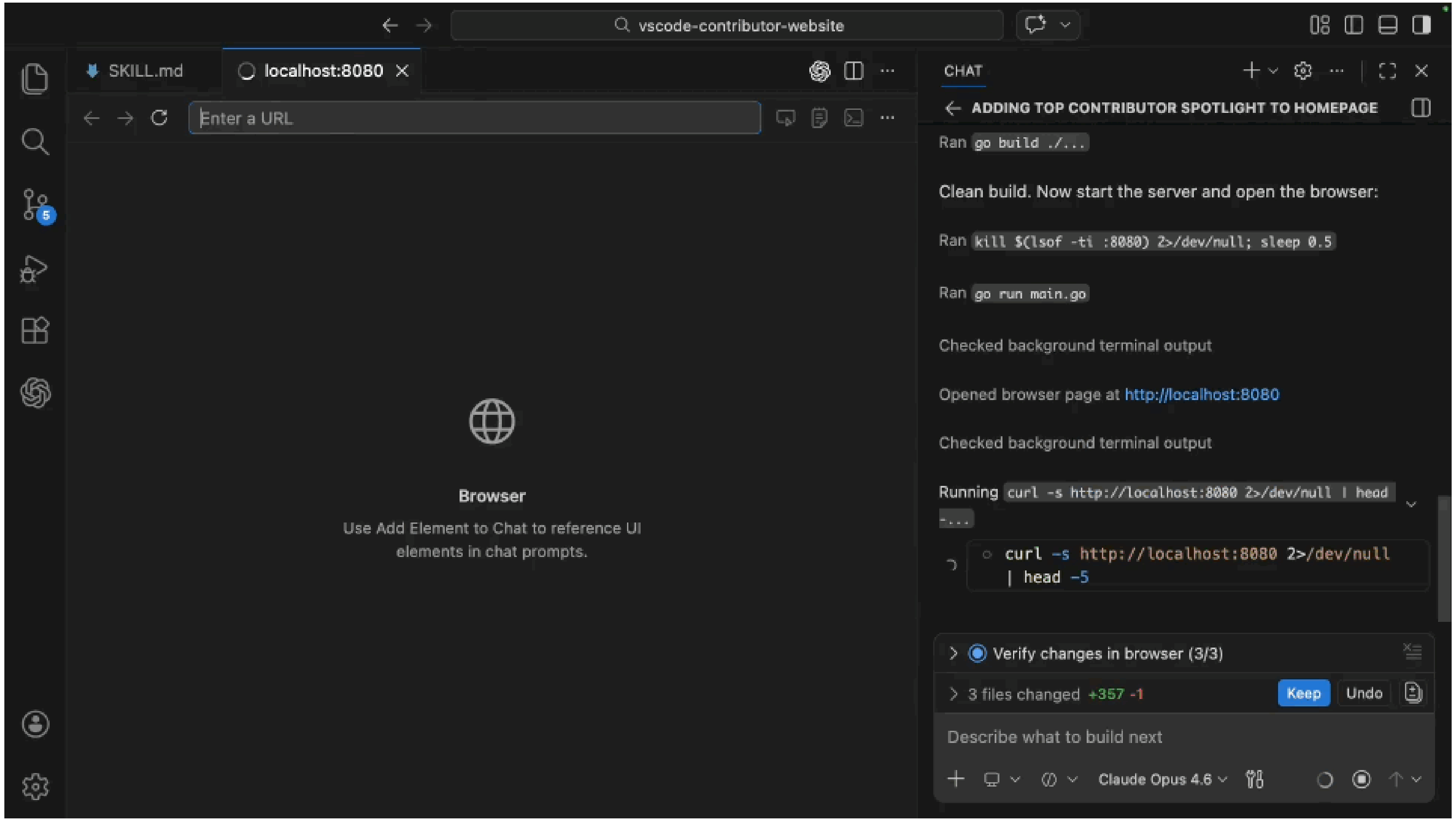
Generate many possible forecasts to estimate uncertainty.

These forecasts provide guidance for human forecasters.

Figure: Google DeepMind / WeatherNext team (August 2026). Alet et al., Nature

Applications of Deep Learning

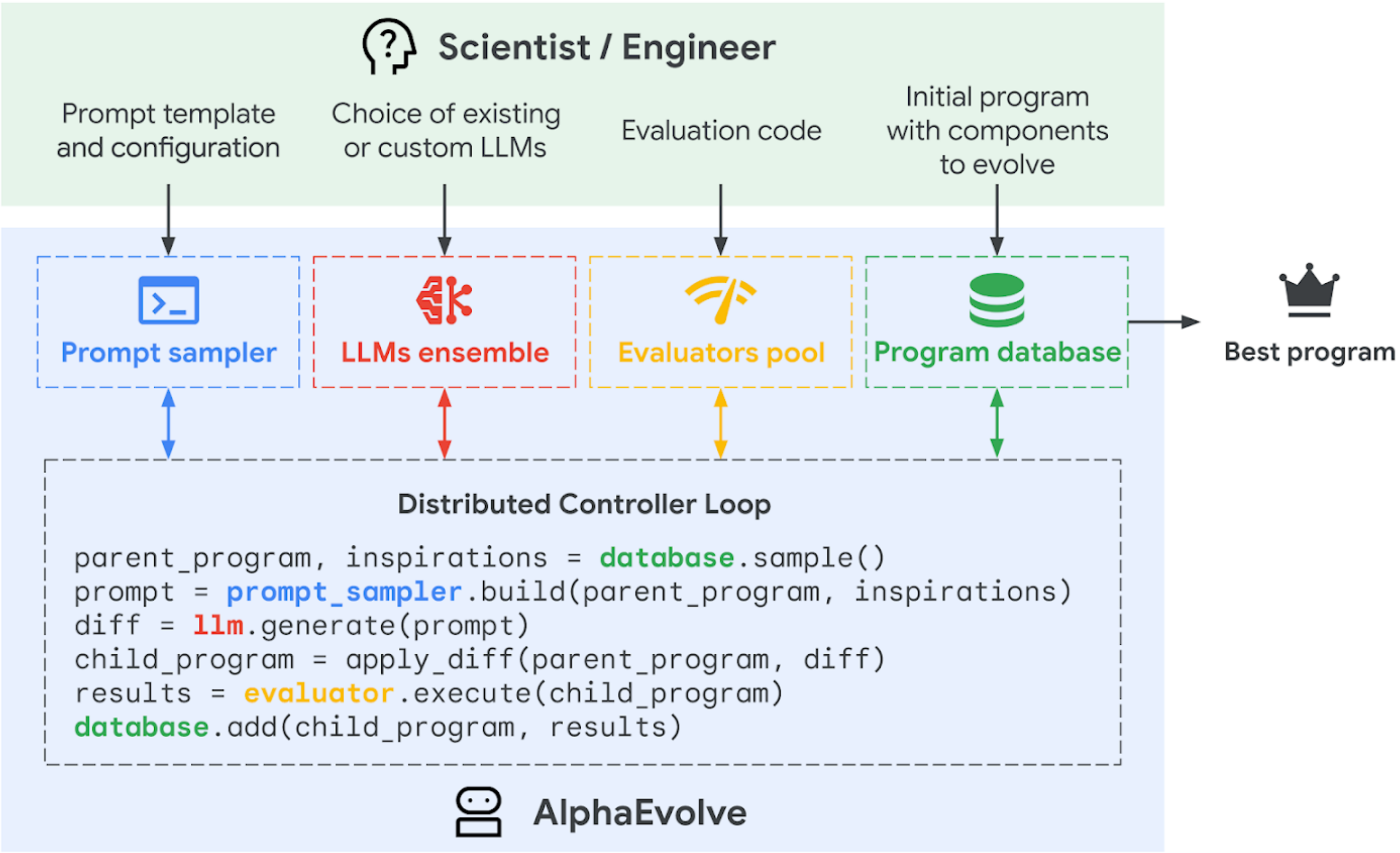
Coding Agents (VS Code, 2026)



[Demo: Microsoft / VS Code Team (Mar. 2026), Claude Opus 4.6 in VS Code. Click the demo to play the full video.]

Applications of Deep Learning

Algorithm Discovery (AlphaEvolve, 2026)



Language models propose programs. Automated tests guide the search.

2026 applications include DNA sequencing and power-grid optimization.

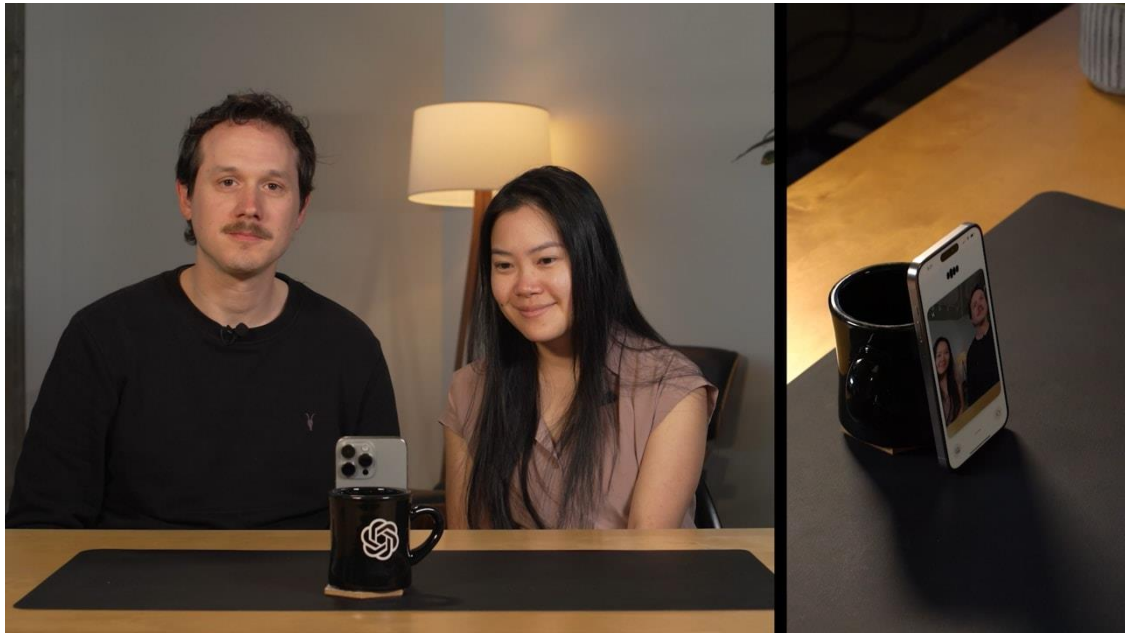
[Diagram: Google DeepMind, AlphaEvolve (May 2025)]
[Applications update: Google DeepMind (May 2026)]

Applications of Deep Learning

Multimodal Assistants: Gemini and OpenAI

Gemini Live: visual guidance (2026)

OpenAI: camera and voice (2024)



[Play OpenAI demo: Rock Paper Scissors]

LLM Post-training and Reasoning

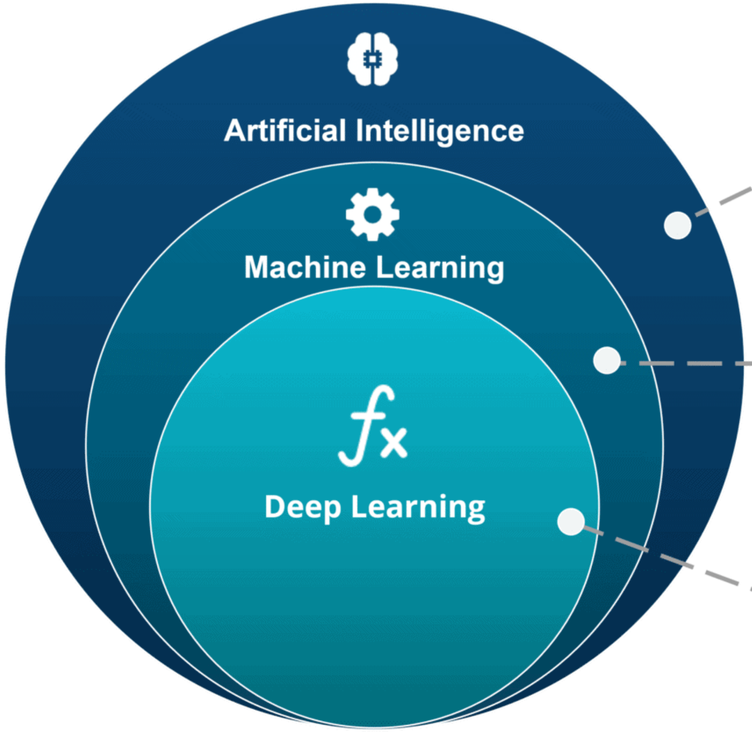
- Pre-training learns broad patterns from data. Post-training adapts a model to instructions and desired behavior.
- Supervised fine-tuning (SFT) learns from demonstrations of useful responses.
- Preference learning uses comparisons between responses. We will cover RLHF and direct preference optimization (DPO).
- Reinforcement learning with verifiable rewards can improve math and coding performance, as illustrated by DeepSeek-R1 (2025).
- More computation at inference can help reasoning. Reliability, evaluation, and verification remain essential.

Outline

- Course Information
- Introduction to Deep Learning
    - History
    - Modern Applications
    - **Taxonomy & Connections to ML/AI/Statistics**
- Course Roadmap
- Reading Deep Learning Research

Relationships w. ML & AI

Deep learning underlies modern language, vision, and multimodal systems.



ARTIFICIAL INTELLIGENCE

Studies systems that perceive, reason, and act

MACHINE LEARNING

Builds models that learn from data and experience

DEEP LEARNING

Learns representations using multi-layer neural networks

Image Credit: https://k21academy.com/datascience/deep-learning/dl-vs-ml/

DL/ML and Statistics

- ML and statistics share probability, estimation, optimization, and learning from data.
- Prediction concerns generalization to new data. Inference concerns quantities, uncertainty, and assumptions.
- These goals overlap in both fields.
- A historical perspective: The Two Cultures (Breiman 2001).

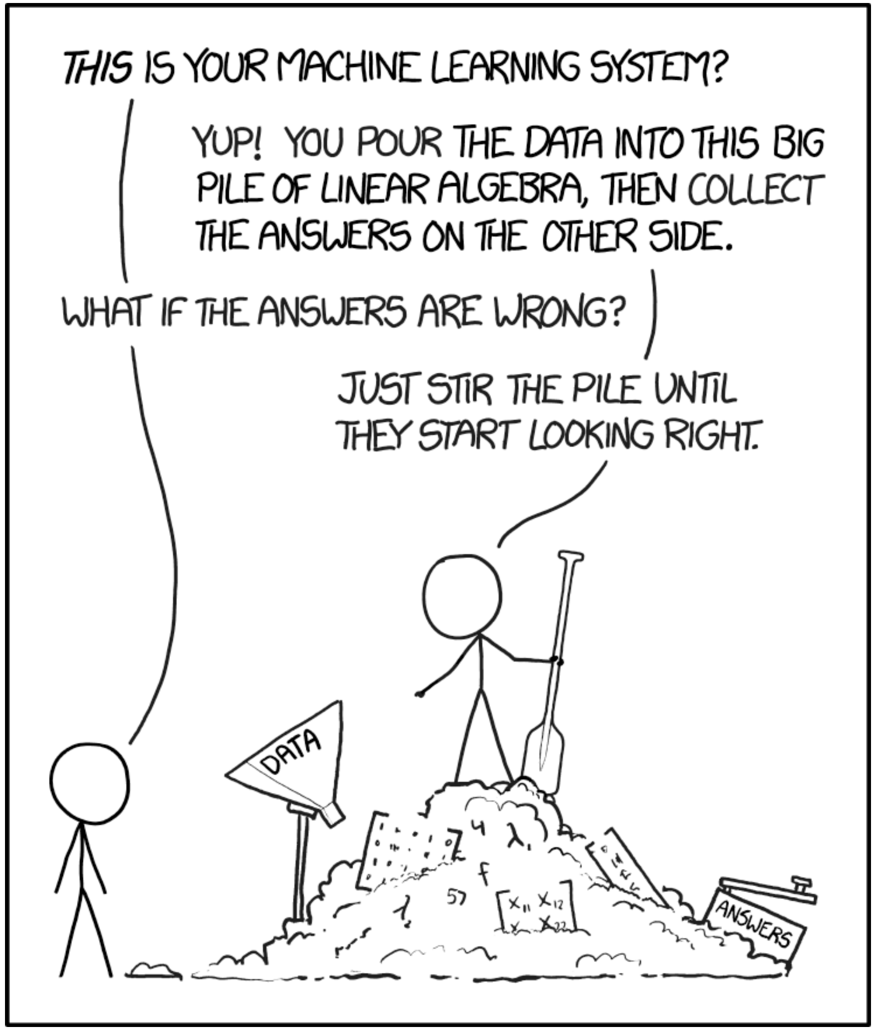


Image Credit: Randall Munroe, xkcd #1838, "Machine Learning" (CC BY-NC 2.5)

Symbolism and Connectionism

- Symbolism: represent knowledge with explicit symbols, rules, and logic. Reason through search and planning.
- Connectionism: learn representations and behavior by adjusting neural-network weights from data.
- Hybrid systems combine learned models with rules, programs, or search.
  - Example: AlphaEvolve generates code, evaluates it, and searches for improvements (Novikov et al. 2025).

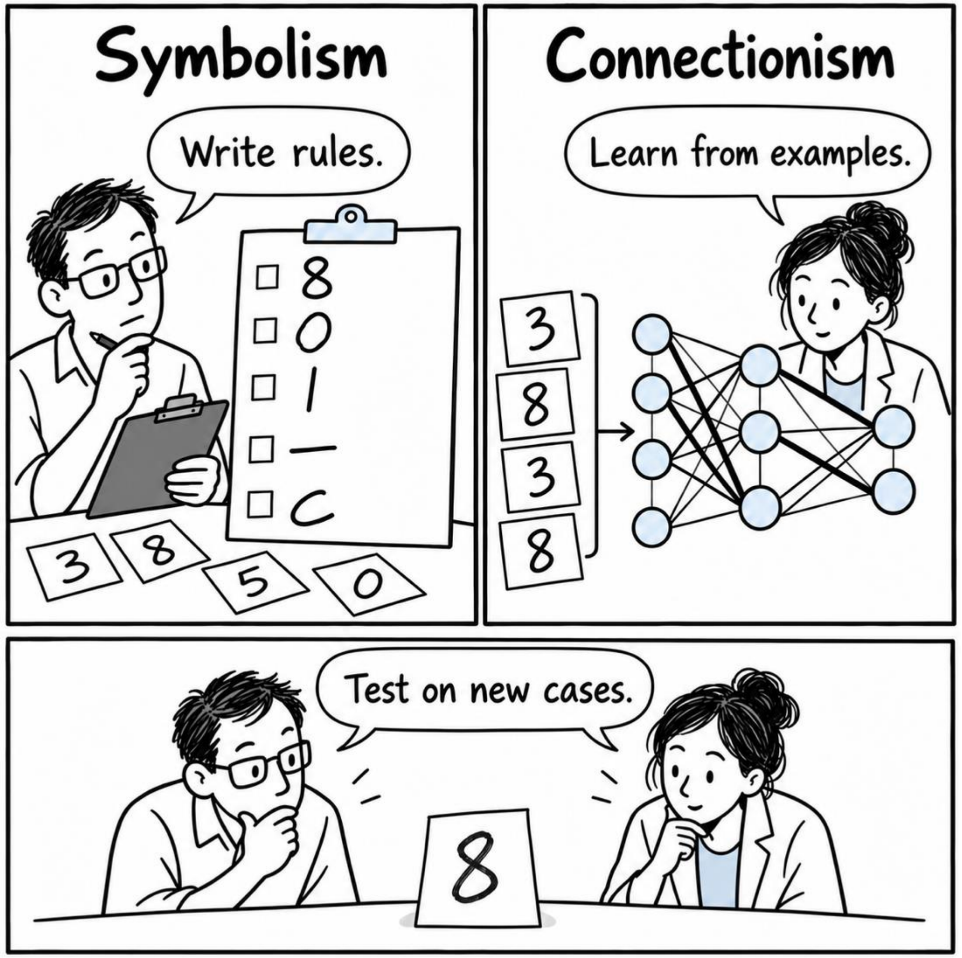
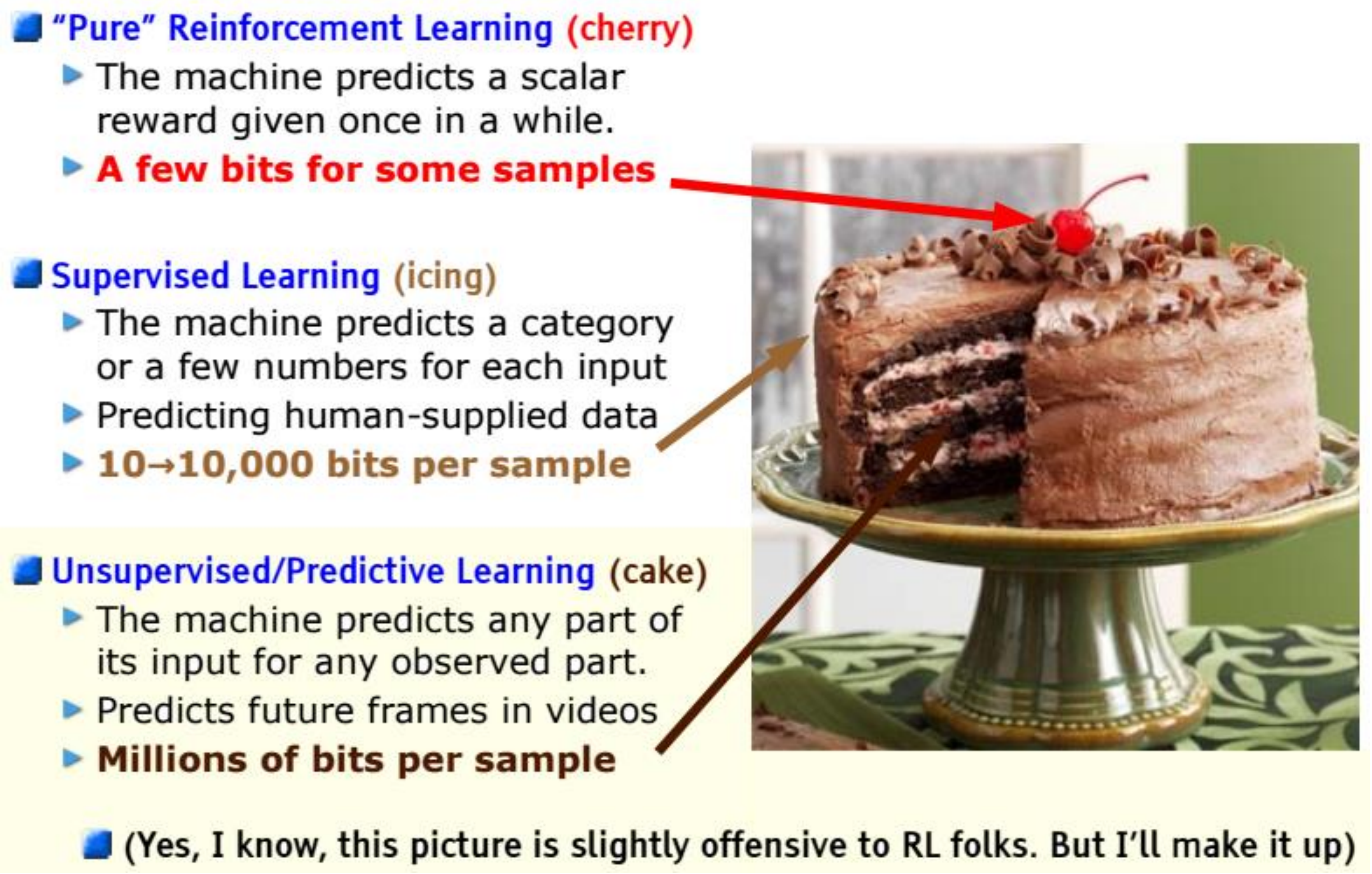


Illustration: generated with OpenAI image generation

Learning Paradigms

- Supervised: learn from input–target pairs.
- Unsupervised / self-supervised: learn from unlabeled data, including targets created from it.
- Reinforcement: learn policies that maximize cumulative reward.



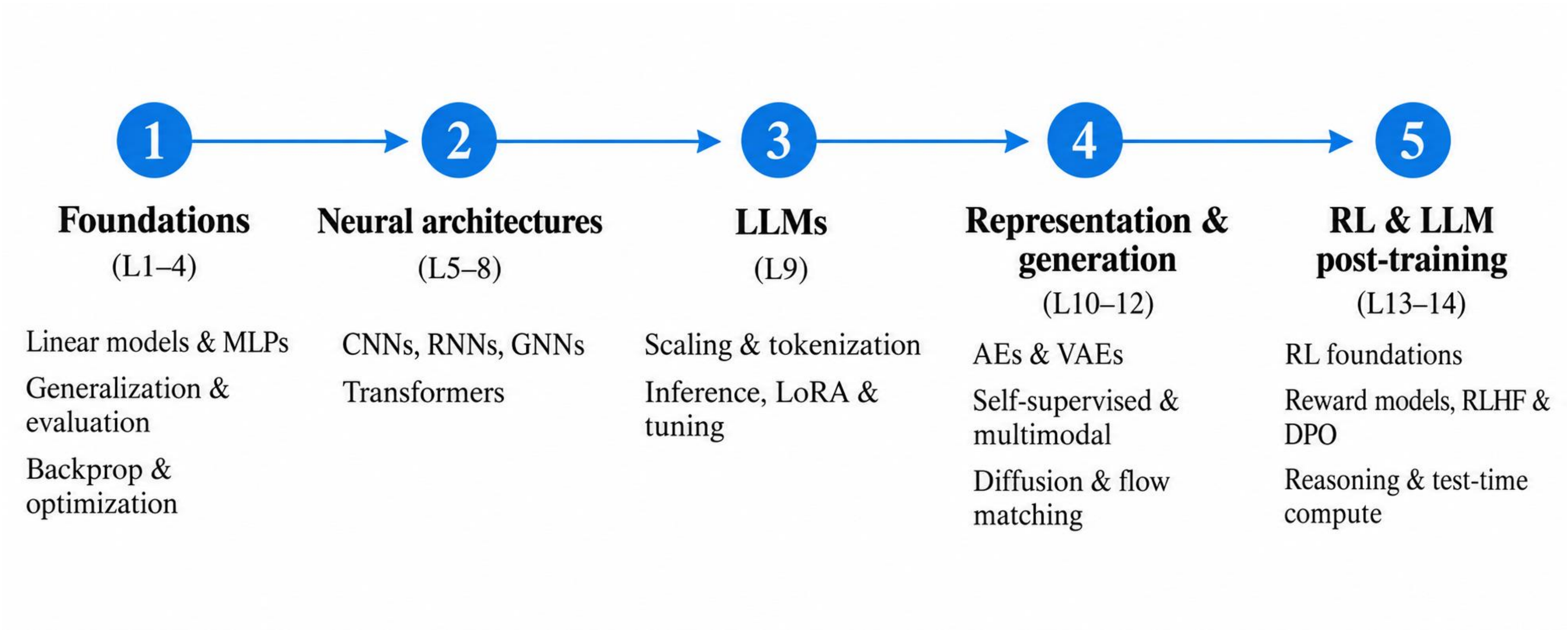
One system can combine these training signals, e.g., LLM pre-training and post-training.

Image Credit: Yann LeCun, NIPS 2016

Outline

- Course Information
- Introduction to Deep Learning
    - History
    - Modern Applications
    - Taxonomy & Connections to ML/AI/Statistics
- **Course Roadmap**
- Reading Deep Learning Research

Course Roadmap



Assignments: implementation, model adaptation, and reproducible evaluation.

Illustration: generated with OpenAI image generation

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics
- Course Roadmap
- **Reading Deep Learning Research**

Reading Deep Learning Research

Research venues

- General ML: ICLR, NeurIPS, and ICML.
- Vision and language: CVPR, ICCV, ACL, and EMNLP.
- Robotics, embodied AI, and physical AI: ICRA, IROS, RSS, and CoRL.

Tips for reading papers

- Identify the problem, assumptions, and main idea. Inspect data, baselines, and evaluation.
- Check ablations, compute requirements, failure cases, and available code or data.
- Benchmark results may not transfer to your task. Preprints may change before peer review.

References

[1] Mcculloch, W. S. & Pitts, W. H. (1943), 'A Logical Calculus of the Ideas Immanent in Nervous Activity', Bulletin of Mathematical Biophysics 5 , 115--133.

[2] Hebb, D. O. (1949), The Organization of Behavior: A Neuropsychological Theory , Wiley , New York .

[3] “Stochastic neural analog reinforcement calculator.” Wikipedia, Wikimedia Foundation, 24 December 2022, en.wikipedia.org/wiki/Stochastic_neural_analog_reinforcement_calculator.

[4] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408.

[5] Minsky, M. & Papert, S. (1969), Perceptrons: An Introduction to Computational Geometry , MIT Press , Cambridge, MA, USA .

[6] Fukushima, Kunihiko (1979). “位置ずれに影響されないパターン認識機構の神経回路のモデル --- ネオコグニトロン ---” [Neural network model for a mechanism of pattern recognition unaffected by shift in position — Neocognitron —]. Trans. IECE (in Japanese). J62-A (10): 658–665.

[7] Hopfield, J. J. (1982), 'Neural Networks and Physical Systems with Emergent Collective Computational Abilities', Proceedings of the National Academy of Sciences, USA 79 , 2554--2558.

[8] Hinton, G. E. & Sejnowski, T. J. (1983), Optimal Perceptual Inference, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[9] Kelley, H. J. (1960). Gradient theory of optimal flight paths. Ars Journal, 30(10), 947-954.

[10] A. E. Bryson. (1961). A gradient method for optimizing multi-stage allocation processes. Proc. Harvard Univ. Symposium on digital computers and their applications.

References

[11] Dreyfus, S. (1962). The numerical solution of variational problems. Journal of Mathematical Analysis and Applications, 5(1), 30-45.

[12] Bryson, A. E., & Ho, Y.-C. (1969). Applied optimal control: optimization, estimation, and control. Blaisdell.

[13] Seppo Linnainmaa (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki, 6–7.

[14] Dreyfus, S. (1973). The computational solution of optimal control problems with time lag. IEEE Transactions on Automatic Control, 18(4), 383-385.

[15] Werbos, Paul. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University.

[16] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[17] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

[18] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[19] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[20] Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on audio, speech, and language processing, 20(1), 30-42.

References

[21] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.
[22] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[23] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.
[24] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489.
[25] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.
[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[27] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.
[28] https://openai.com/blog/chatgpt/
[29] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., ... & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589.
[30] https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/

References

[31] DeepSeek-AI, Guo, D., Yang, D., et al. (2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948

[32] Novikov, A., Vũ, N., Eisenberger, M., et al. (2025). AlphaEvolve: A coding agent for scientific and algorithmic discovery. arXiv:2506.13131

[33] Avsec, Ž., Latysheva, N., Cheng, J., et al. (2026). Advancing regulatory variant effect prediction with AlphaGenome. Nature, 649, 1206–1218. doi:10.1038/s41586-025-10014-0

[34] Breiman, L. (2001). Statistical Modeling: The Two Cultures. Statistical Science, 16(3), 199–231. doi:10.1214/ss/1009213726

[35] Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. The Journal of Physiology, 148(3), 574–591. doi:10.1113/jphysiol.1959.sp006308

Questions?