STAT 453: Introduction to Deep Learning and Generative Models

Ben Lengerich

Lecture 02: A Brief History of DL

September 8, 2026

A few notes on course logistics

- Extra credit: lecture notes
  - **Sign-up sheet**
  - Check that you are correctly signed up – any issues, please reach out.
- Project
  - Honors-optional component: please reach out.
- HW1
  - Posted on website
  - Due next Friday (9/18) via Canvas
  - Assignment: Simple stats questions + implement a perceptron in Python
    - We'll discuss this in detail in lecture Tuesday, 9/15
- For your convenience, a Google calendar for the course.

Questions about Course Logistics?

Today: A Brief History of DL

1. **Artificial neurons**
2. Multilayer neural networks
3. Deep Learning
4. The DL Hardware & Software Landscape
5. Current Research Trends

A preview of our first half of our course. You don't have to get it all today.

9/8	Lecture #2 (Prof. Lengerich): **A Brief History of Deep Learning** [slides I notes]
9/10	Lecture #3 (Prof. Lengerich): **Statistics, Linear Algebra, and Calculus Review** [slides I notes]
9/15	Lecture #4 (Prof. Lengerich): **Single-layer networks** [slides I notes]
9/17	Lecture #5 (Prof. Lengerich): **Parameter Optimization and Gradient Descent** [slides I notes]
9/22	Lecture #6 (Prof. Lengerich): **Automatic differentiation with PyTorch; Project Discussion** [slides I notes]
9/24	Lecture #7 (Prof. Lengerich): **Multi-layer perceptrons and backpropagation** [slides I notes]
9/29	Lecture #8 (Prof. Lengerich): **Structure and weight sharing; CNNs** [slides I notes]
10/1	Lecture #9 (Prof. Lengerich): **Regularization** [slides I notes]
10/6	Lecture #10 (Prof. Lengerich): **Normalization and Initialization** [slides I notes]
10/8	Lecture #11 (Prof. Lengerich): **Optimization and learning rates** [slides I notes]

Recall: What is Machine Learning?

Formally, a computer program is said to **learn** from experience $\mathcal{E}$ with respect to some task $\mathcal{T}$ and performance measure $\mathcal{P}$ if its **performance at $\mathcal{T}$ as measured by $\mathcal{P}$ improves with $\mathcal{E}$.**

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

- Task $\mathcal{T}$: Learn a function $h: \mathcal{X} \rightarrow \mathcal{Y}$
- Experience $\mathcal{E}$: Labeled samples $\{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i}=1}^{n}$
- Performance $\mathcal{P}$: A measure of how good h is

Unsupervised Learning

- No labels/targets
- No feedback
- Find hidden structure in data

- Task $\mathcal{T}$: Discover structure in data
- Experience $\mathcal{E}$: Unlabeled samples $\{\mathrm{x_i}\}_{\mathrm{i}=1}^{n}$
- Performance $\mathcal{P}$: Measure of fit or utility

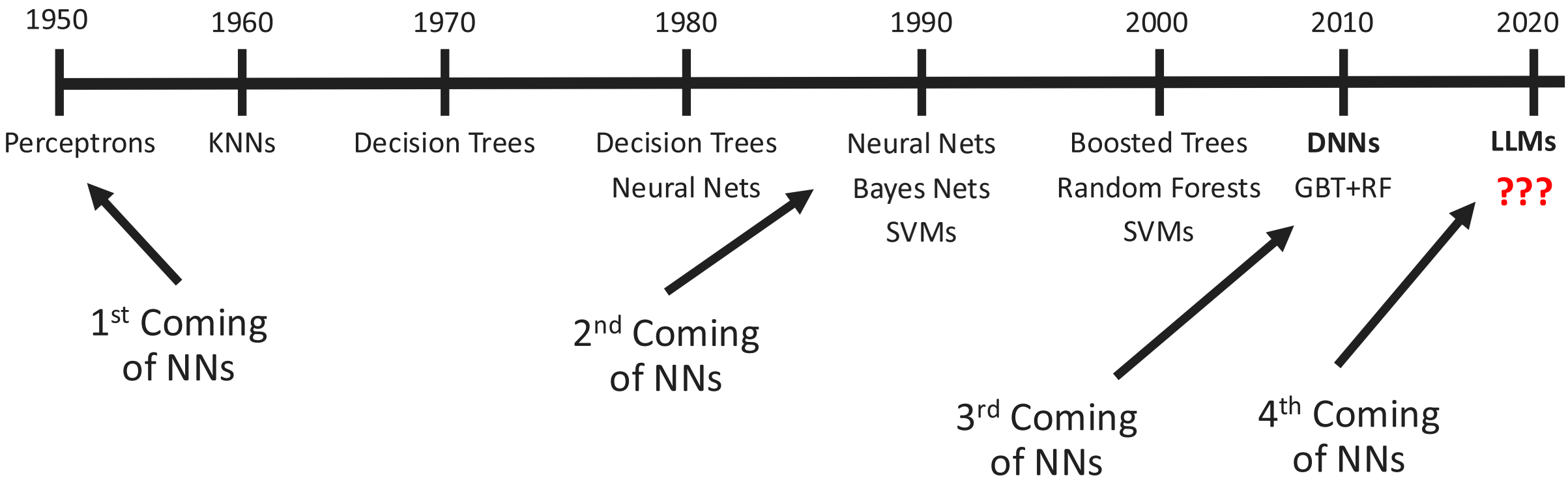
Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

- Task $\mathcal{T}$: Learn a policy $\pi: S \rightarrow A$
- Experience $\mathcal{E}$: Interaction with environment
- Performance $\mathcal{P}$: Expected reward

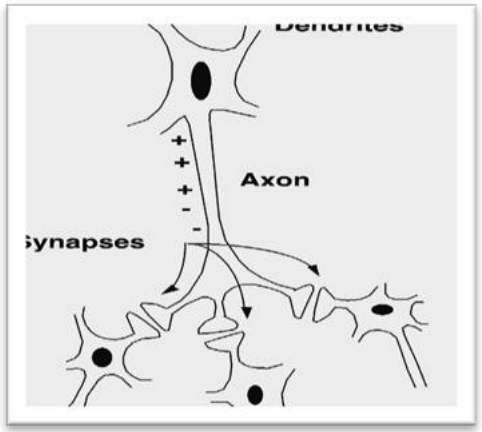
Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

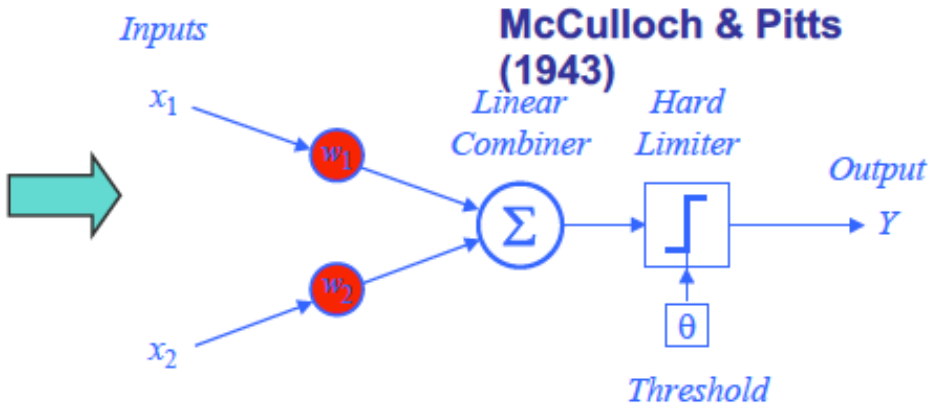
A Brief History of ML

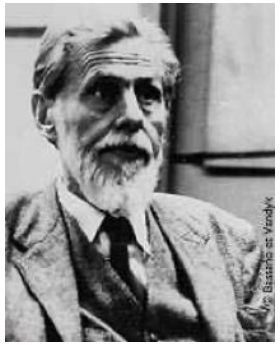


Courtesy of Rich Caruana

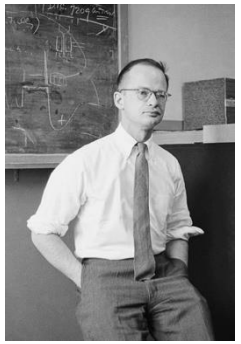
McCulloch & Pitt's neuron model (1943)







Warren McCulloch



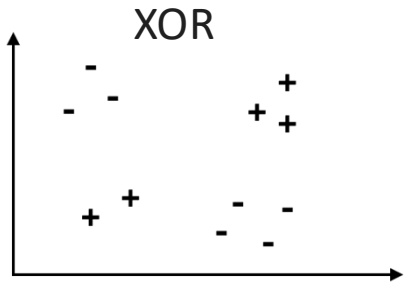
Walter Pitts

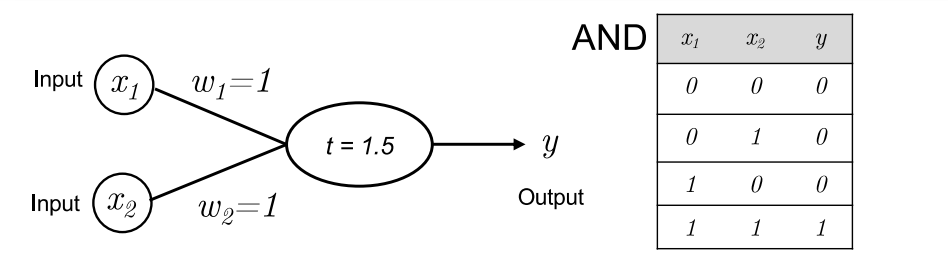
Learning algorithm: **No Learning**

McCulloch & Pitt's neuron model (1943)

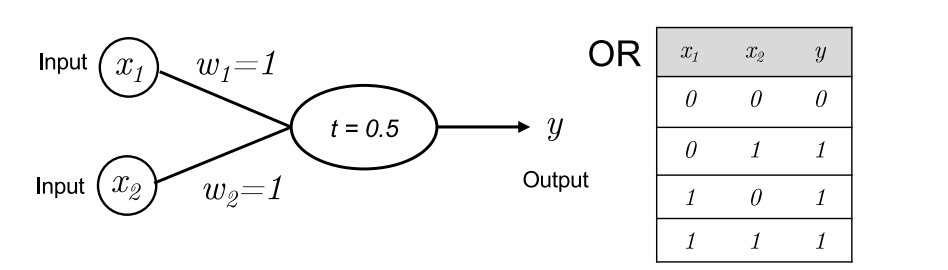
- McCulloch & Pitts neuron: Threshold and (+1, -1) weights
- Can represent "AND", "OR", "NOT":

But not "XOR"

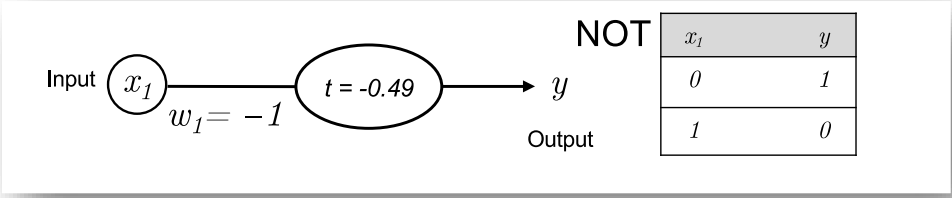




x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

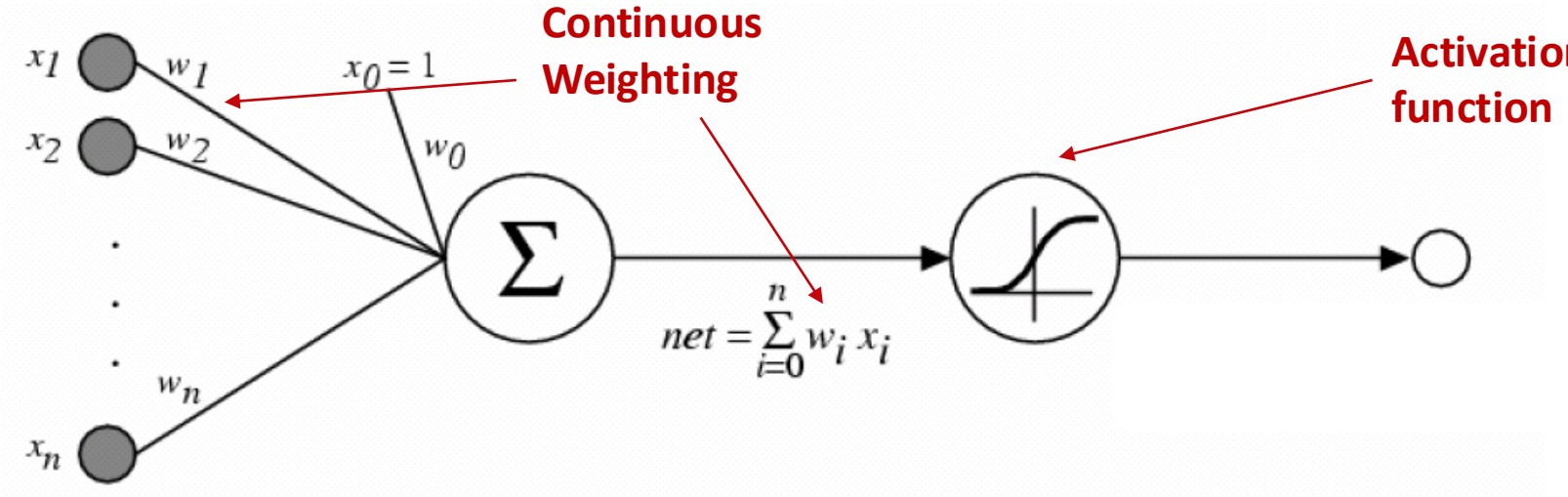


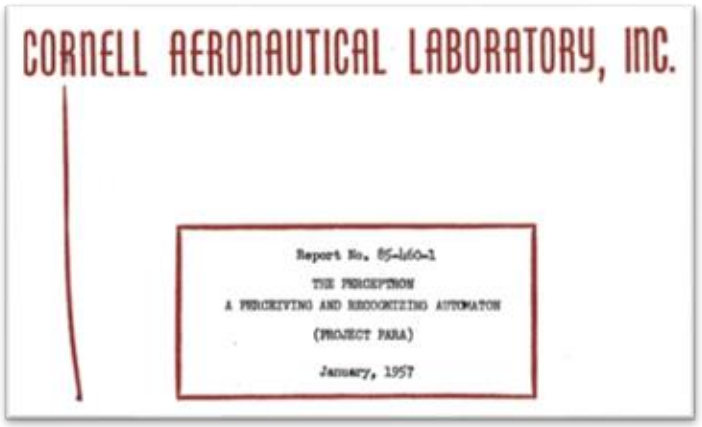
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



x_1	y
0	1
1	0

The Perceptron: Generalize MP Neurons




CORNELL AERONAUTICAL LABORATORY, INC.

Report No. 85-460-1

THE PERCEPTRON

A PERCEIVING AND RECOGNIZING AUTOMATON

(PROJECT PARA)

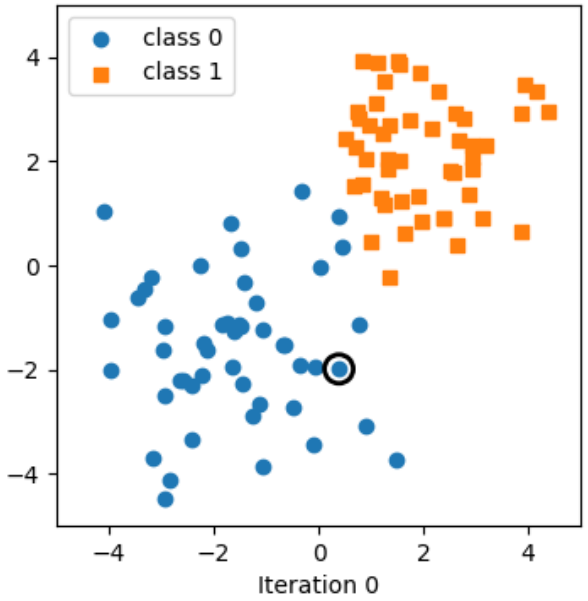
January, 1957

Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton. Project Para*. Cornell Aeronautical Laboratory.

Source: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/Members/wilex4/Rosen-2.jpg

Perceptron Learning Algorithm

- Assume binary classification task
- Perceptron finds decision boundary if classes are linearly-separable



[animated GIF]

[Code]

Learning algorithm:
Scale-invariant update based on sign-matching of predicted vs actual value.

We will discuss in detail Lec 4

Perceptron Fun Fact

Where a perceptron had been trained to distinguish between - this was for military purposes - it was looking at a scene of a forest in which there were camouflaged tanks in one picture and no camouflaged tanks in the other. And the perceptron - after a little training - made a 100% correct distinction between these two different sets of photographs. Then they were embarrassed a few hours later to discover that the two rolls of film had been developed differently. And so these pictures were just a little darker than all of these pictures and the perceptron was just measuring the total amount of light in the scene. But it was very clever of the perceptron to find some way of making the distinction.

-- Marvin Minsky, Famous AI researcher, Author of the famous "Perceptrons" book

Source: https://www.webofstories.com/play/marvin.minsky/122

https://qph.fs.quoracdn.net/main-qimg-305eb8136c4a20f348bb7ab465bc2e1b

http://theconversation.com/want-to-beat-climate-change-protect-our-natural-forests-121491

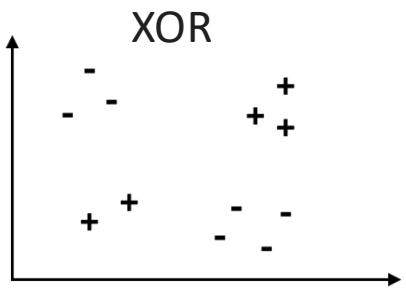
Optimism vs. pessimism

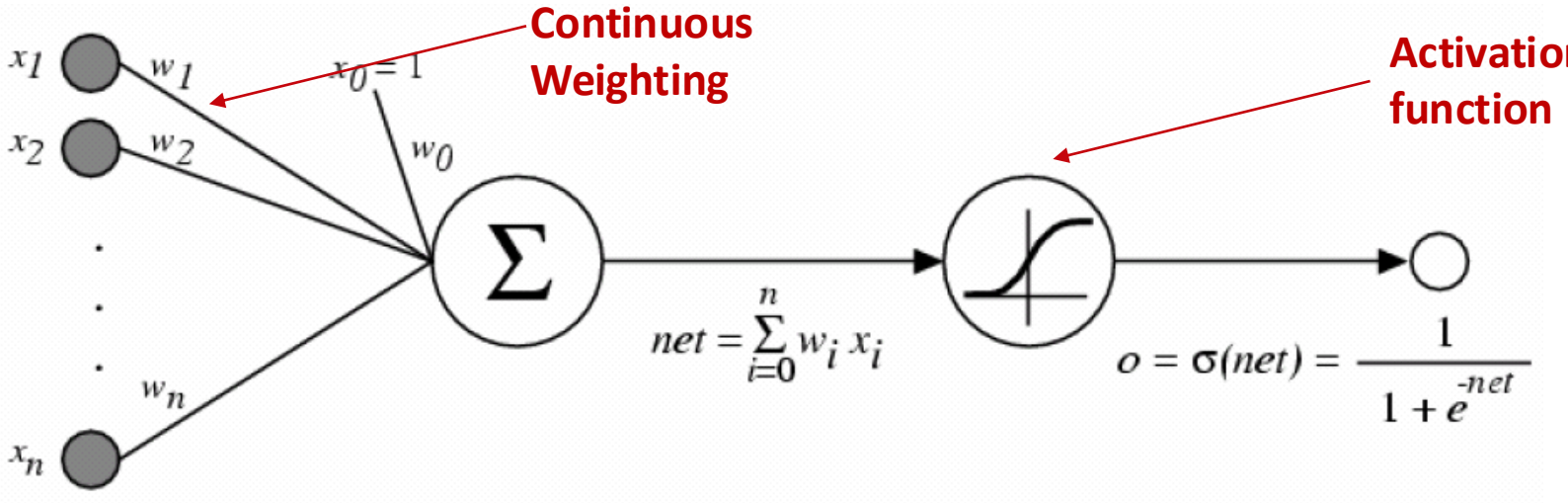
- Optimism in the early history of AI (from wiki)

> In a 1958 press conference organized by the US Navy, Rosenblatt made statements about the perceptron that caused a heated controversy among the fledgling AI community; based on Rosenblatt's statements, *The New York Times* reported the perceptron to be "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."[5]

- Discussion: what do you think about the current AI gold rush?

Can a Perceptron represent XOR?

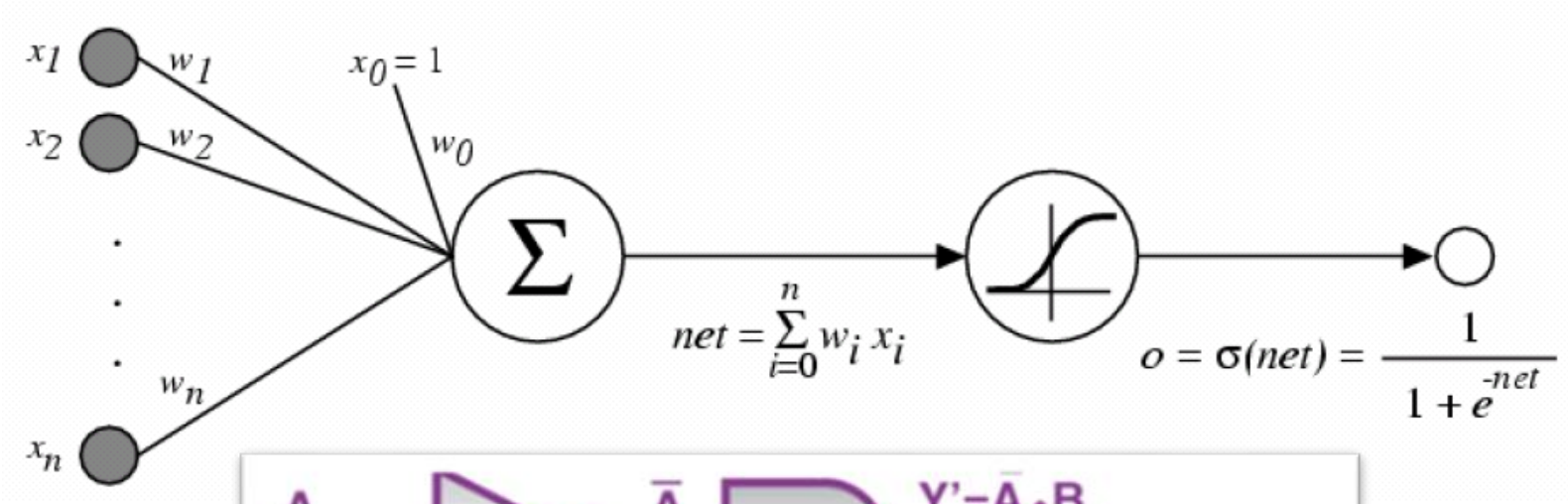




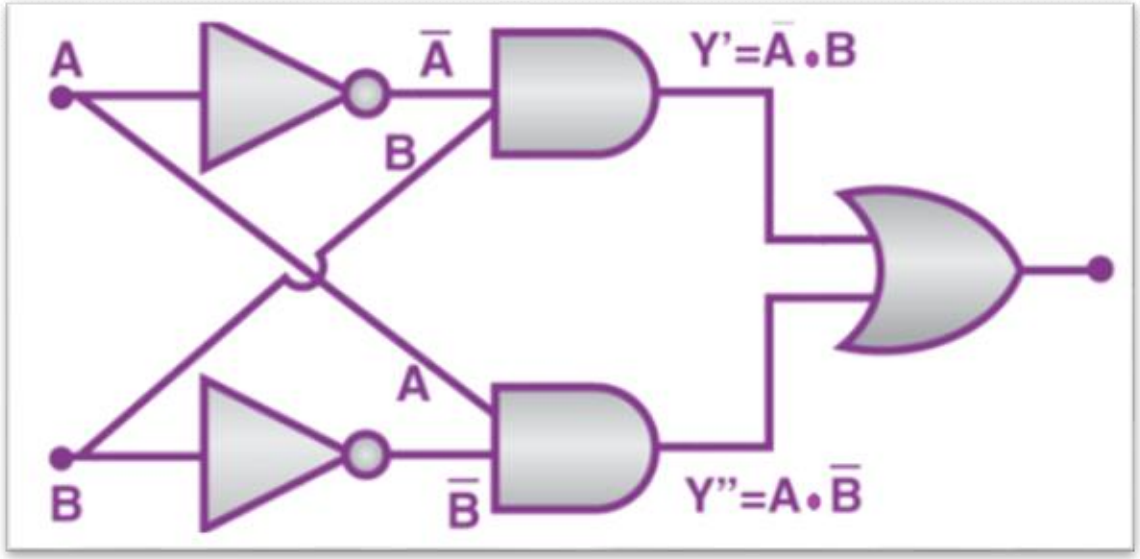
- No
- If there were, then there would be constants w_1 and w_2 such that:
  - When $x_1 = x_2$, then $\sigma(w_1 x_1 + w_2 x_2) < \theta$
  - When $x_1 \neq x_2$, then $\sigma(w_1 x_1 + w_2 x_2) \geq \theta$
  - Let $x_1 = 1, x_2 = 0$
    - **Eq. (1)**: $\sigma(w_1) \geq \theta$
  - Let $x_1 = 0, x_2 = 1$
    - **Eq. (2)**: $\sigma(w_2) \geq \theta$
  - Let $x_1 = 1, x_2 = 1$:
    - Eq. (3): $\sigma(w_1 + w_2) < \theta$

Eqs. (1) + Eqs. (2) contradict Eq. (3) for monotone σ

An XOR Logic Gate



Multi-layer Perceptron?



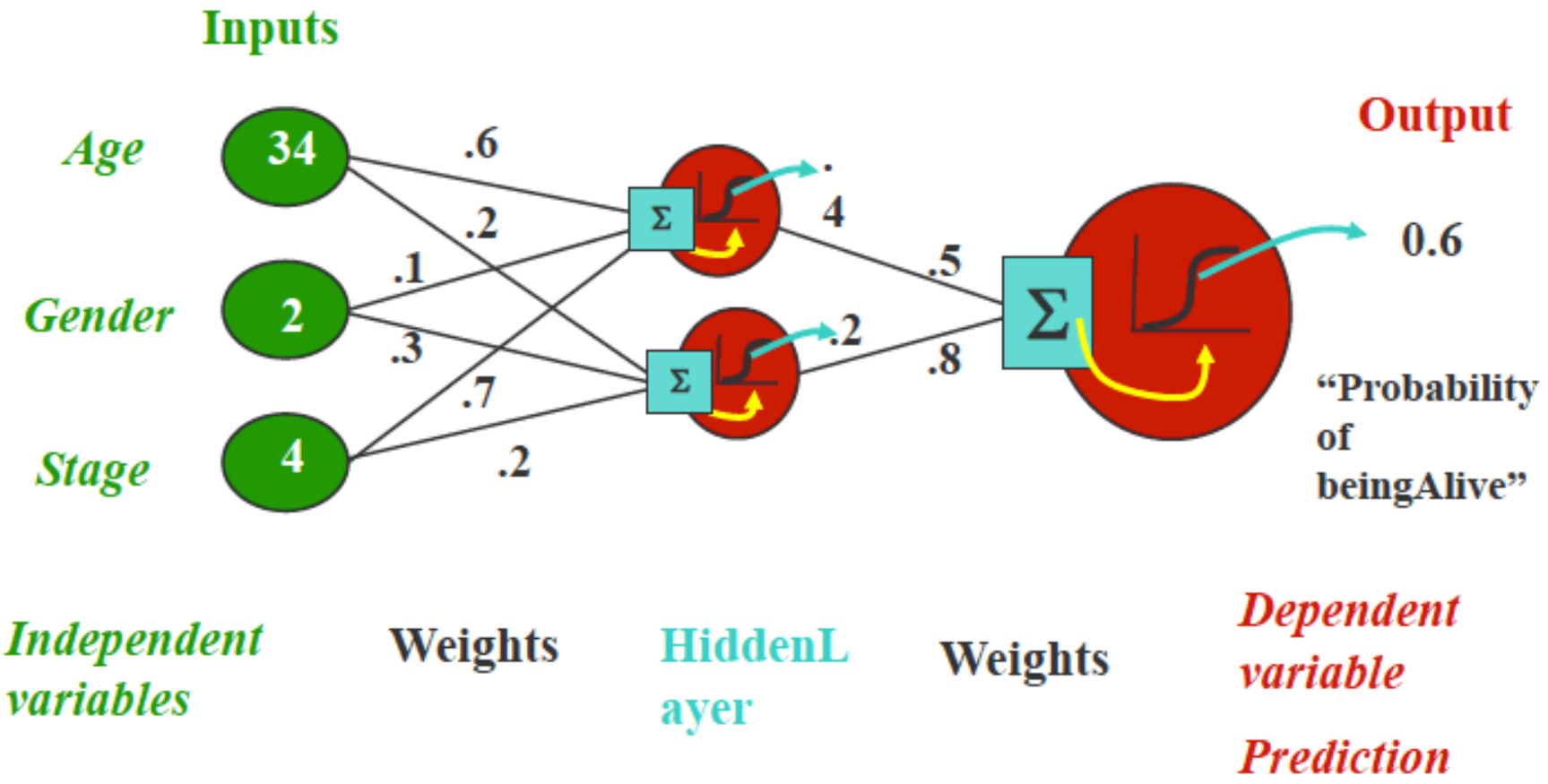
https://byjus.com/jee/basic-logic-gates/

Today: A Brief History of DL

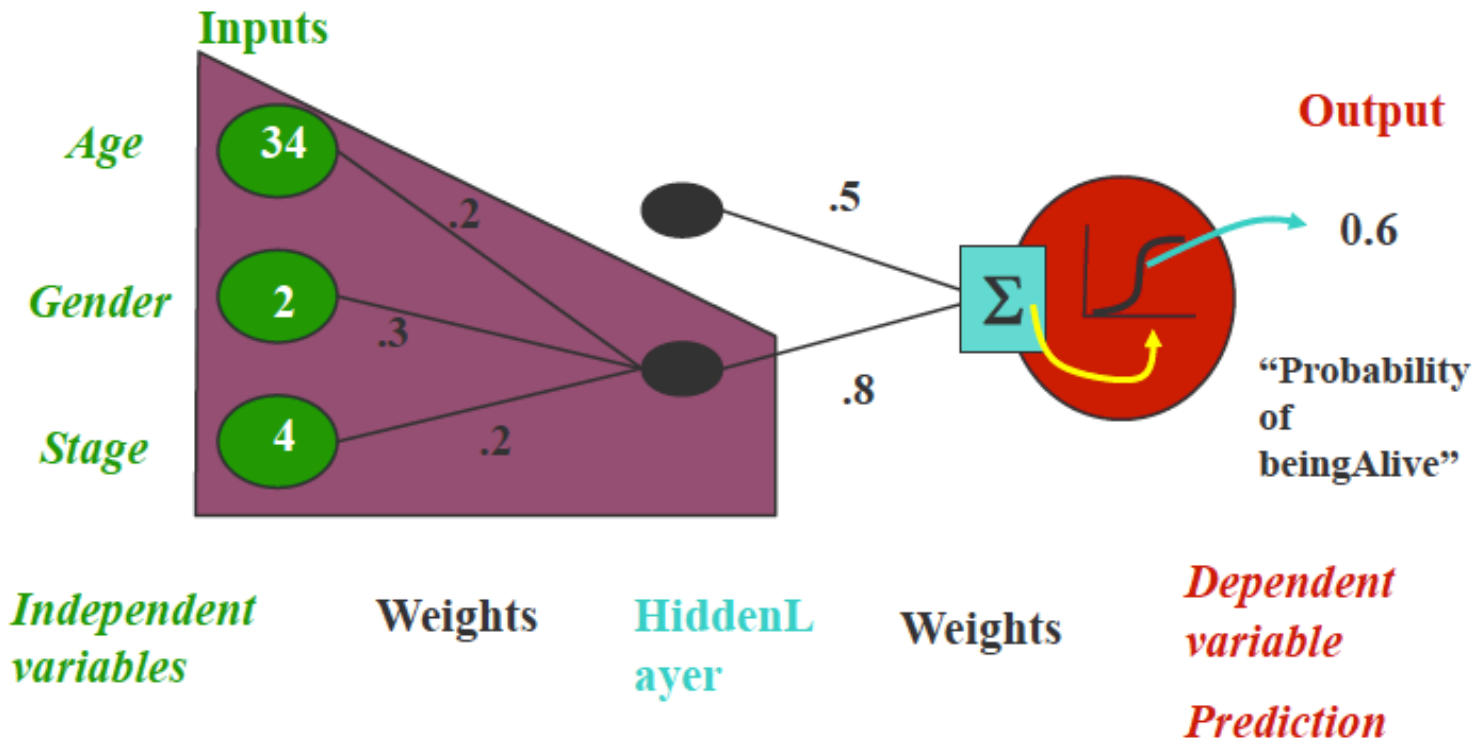
1. Artificial neurons
2. **Multilayer neural networks**
3. Deep Learning
4. The DL Hardware & Software Landscape
5. Current Research Trends

Multi-Layer Perceptrons (MLPs)

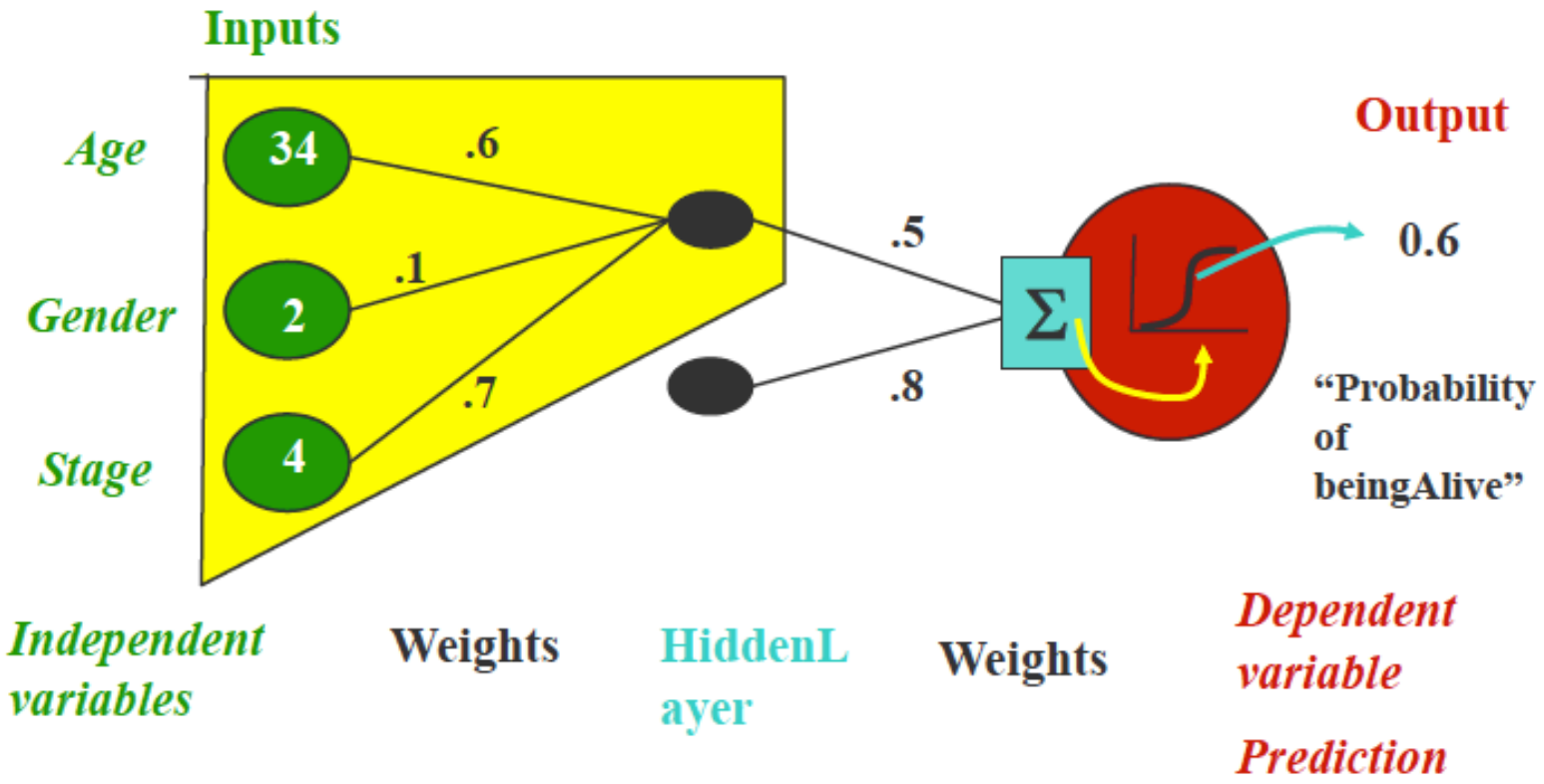
aka “Multi-layer Neural Networks”



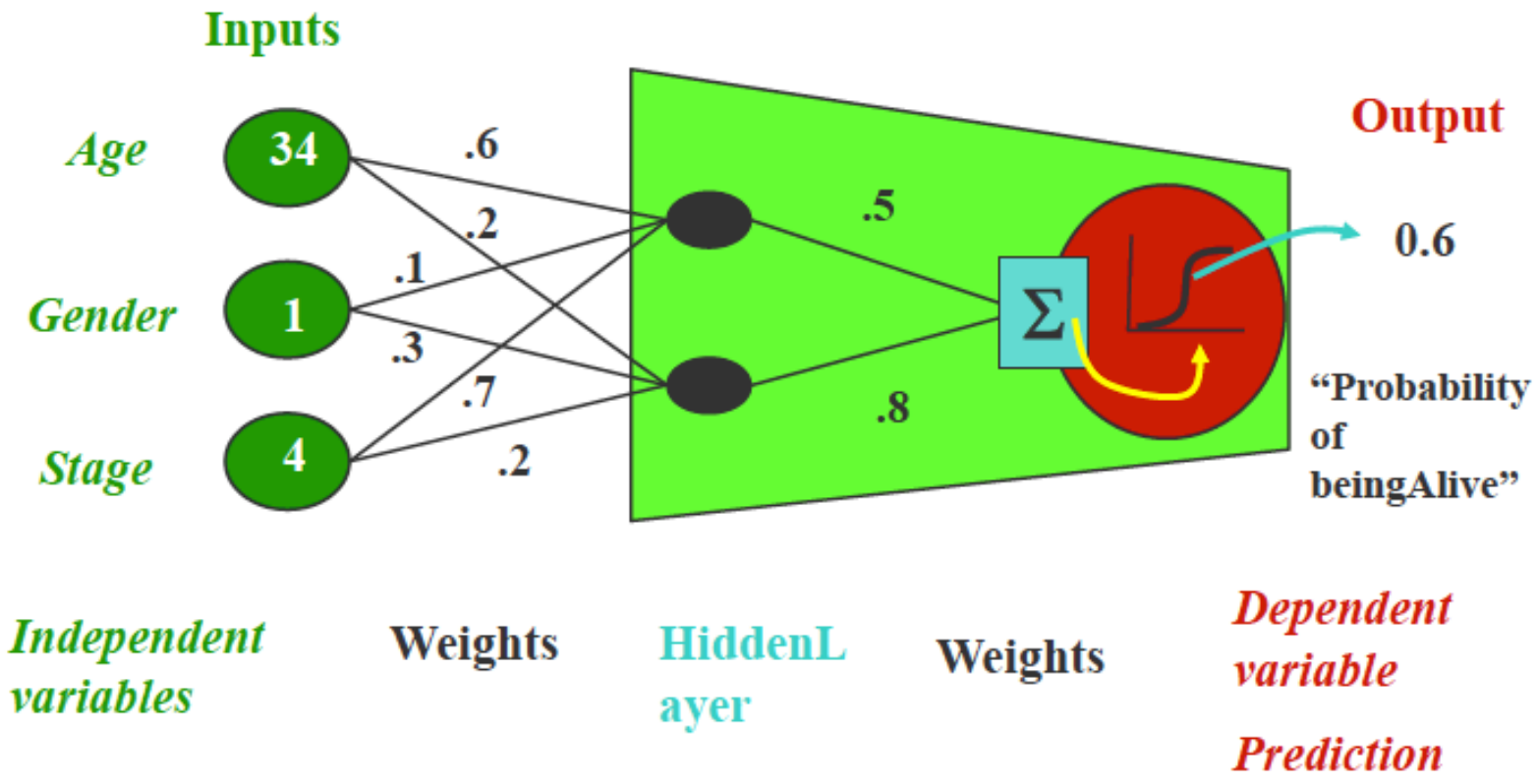
“Combined Logistic Models”



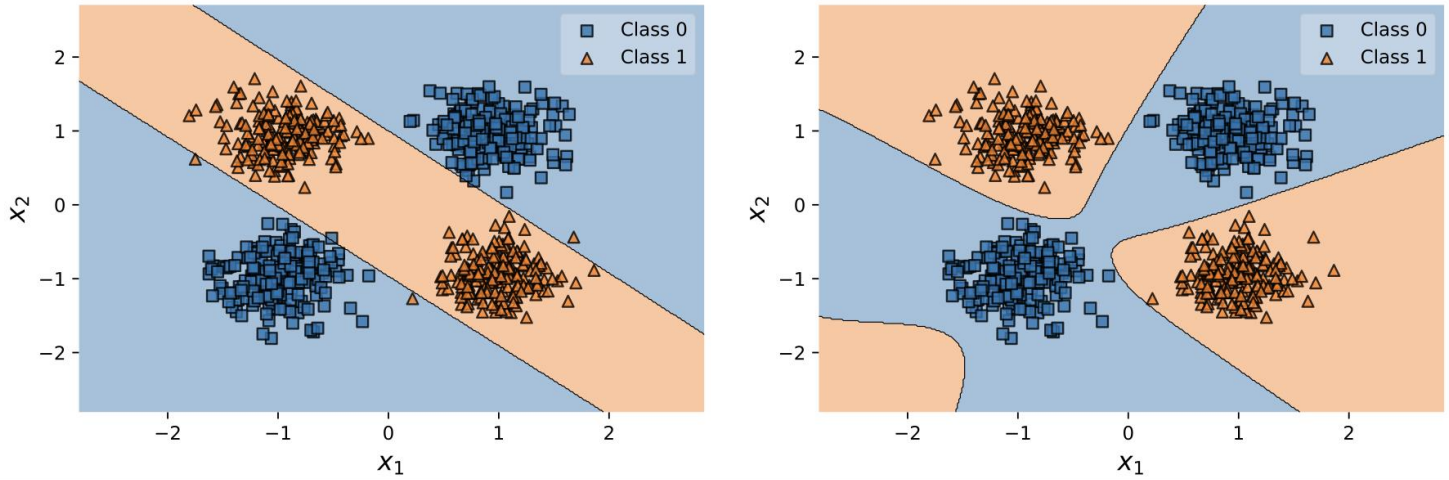
“Combined Logistic Models”



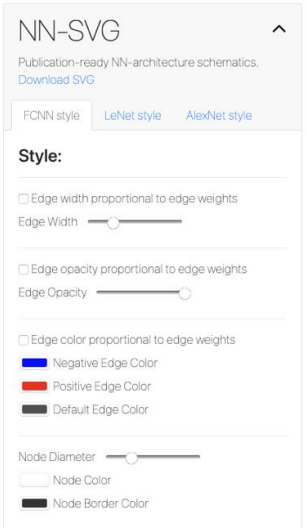
“Combined Logistic Models”

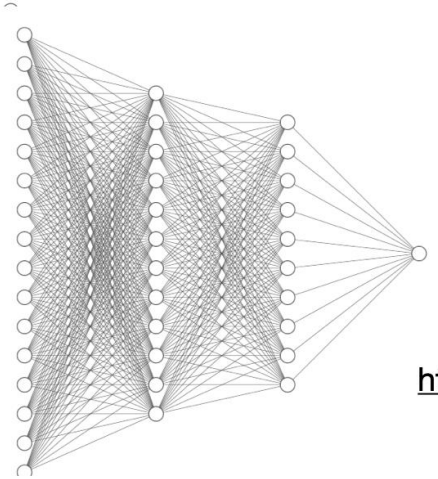


Multilayer Neural Networks Can Solve XOR



Decision boundaries of two different multilayer perceptrons on simulated data solving the XOR problem

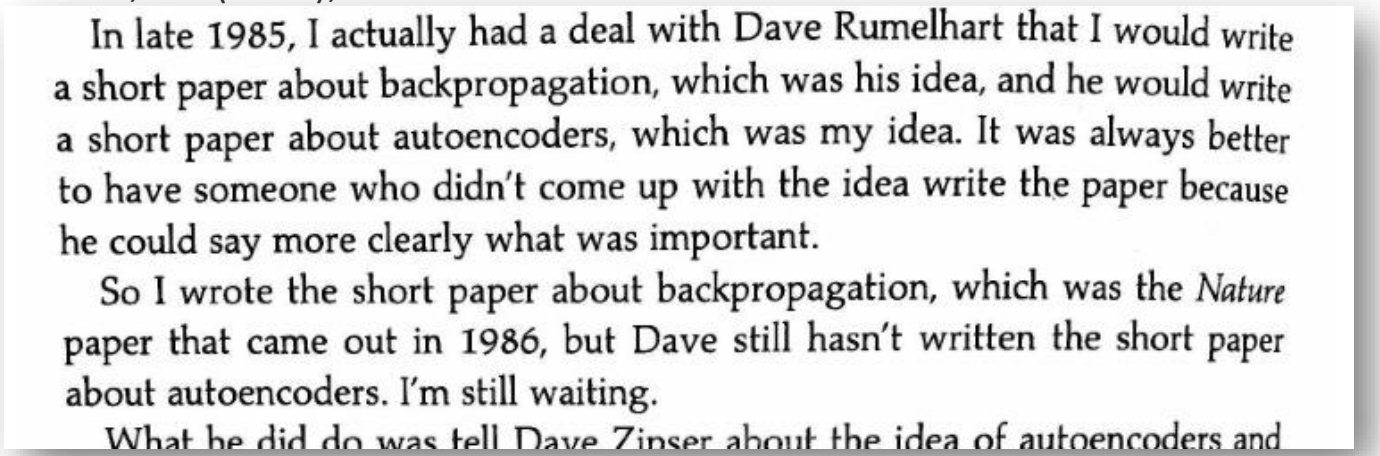




https://alexlenail.me/NN-SVG/index.html

Representation better, but what about learning?

- How can we train a multilayer model?
  - No targets / ground truth for the hidden nodes
- Solution: Backpropagation
  - Independently formulated many times
    - http://people.idsia.ch/~juergen/who-invented-backpropagation.html
  - Rumelhart and Hinton (1986) showed that it really works
    - Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning representations by back-propagating errors.* Nature, 323(6088), 533.



In late 1985, I actually had a deal with Dave Rumelhart that I would write a short paper about backpropagation, which was his idea, and he would write a short paper about autoencoders, which was my idea. It was always better to have someone who didn't come up with the idea write the paper because he could say more clearly what was important.

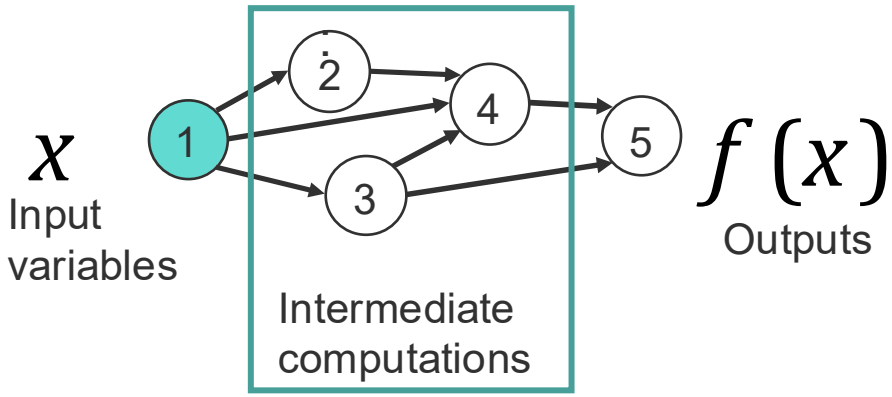
So I wrote the short paper about backpropagation, which was the *Nature* paper that came out in 1986, but Dave still hasn't written the short paper about autoencoders. I'm still waiting.

What he did do was tell Dave Zipser about the idea of autoencoders and

– Geoffrey Hinton in Talking Nets - An Oral History of Neural Networks, pg. 380

Backpropagation

- Neural networks are function compositions that can be represented as computation graphs:



$$\frac{\partial f_n}{\partial x} =$$

- By applying the chain rule, and working in reverse order, we get:

$$\frac{\partial f_n}{\partial x} = \sum_{i_1 \in \pi(n)} \frac{\partial f_n}{\partial f_{i_1}} \frac{\partial f_{i_1}}{\partial x} = \sum_{i_1 \in \pi(n)} \frac{\partial f_n}{\partial f_{i_1}} \sum_{i_2 \in \pi(i_1)} \frac{\partial f_{i_1}}{\partial f_{i_2}} \frac{\partial f_{i_1}}{\partial x} = \dots$$

Backpropagation: More than just gradient descent?

- Engineering innovations

current gradient and earlier gradients to the weight change.

To break symmetry we start with small random weights. Variants on the learning procedure have been discovered independently by David Parker (personal communication) and by Yann Le Cun[3].

Leads to Dropout?

Backpropagation: More than just gradient descent?

- Engineering innovations

The most obvious drawback of the learning procedure is that the error-surface may contain local minima so that gradient descent is not guaranteed to find a global minimum. However, experience with many tasks shows that the network very rarely gets stuck in poor local minima that are significantly worse than the global minimum. We have only encountered this undesirable behaviour in networks that have just enough connections to perform the task. Adding a few more connections creates extra dimensions in weight-space and these dimensions provide paths around the barriers that create poor local minima in the lower dimensional subspaces.

Leads to Over-parameterized models?

Today: A Brief History of DL

1. Artificial neurons
2. Multilayer neural networks
3. **Deep Learning**
4. The DL Hardware & Software Landscape
5. Current Research Trends

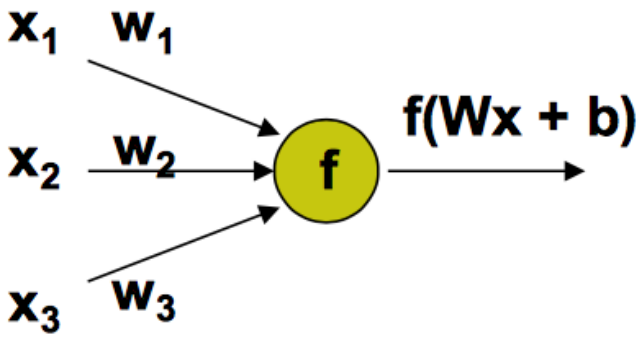
About the term “Deep Learning”

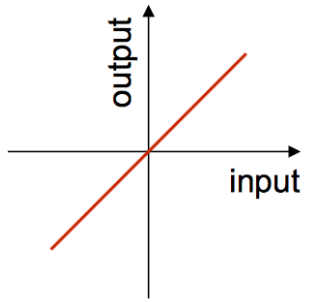
“Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. **Deep learning methods are representation-learning methods with multiple levels of representation** [...]”

-- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436.

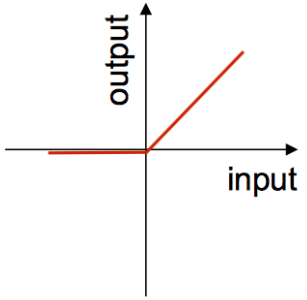
DL building blocks

- Activation Functions
  - Linear and ReLU
  - Sigmoid and tanh
  - etc.

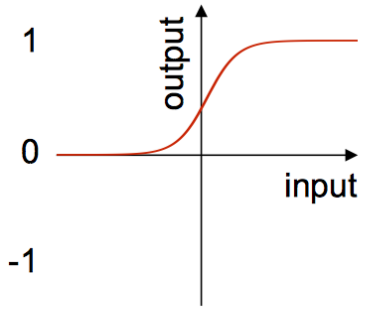




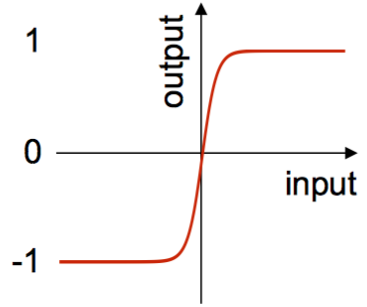
Linear



Rectified linear



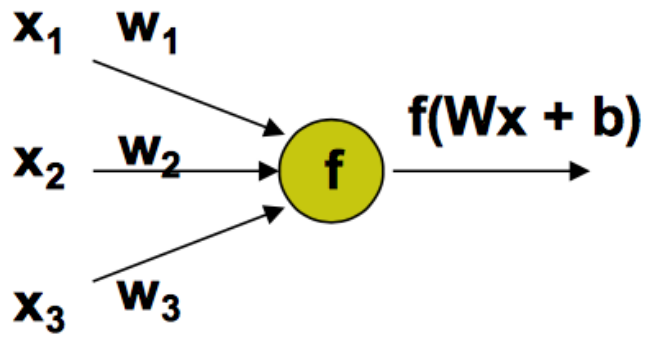
Sigmoid

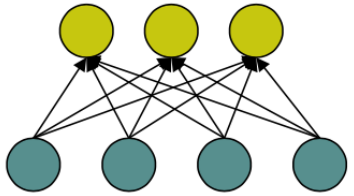


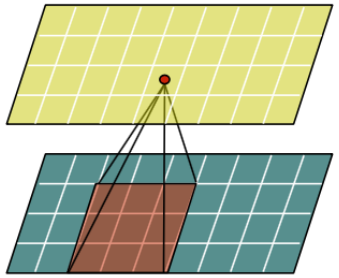
Hyperbolic tangent

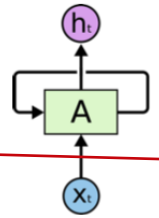
DL building blocks

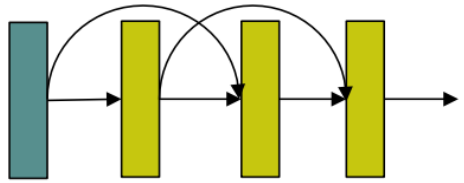
- Activation Functions
  - Linear and ReLU
  - Sigmoid and tanh
  - etc.
- Layers
  - Fully-connected
  - Convolutional & pooling
  - Recurrent
  - Residual (ResNets)
  - Transformers
  - etc.





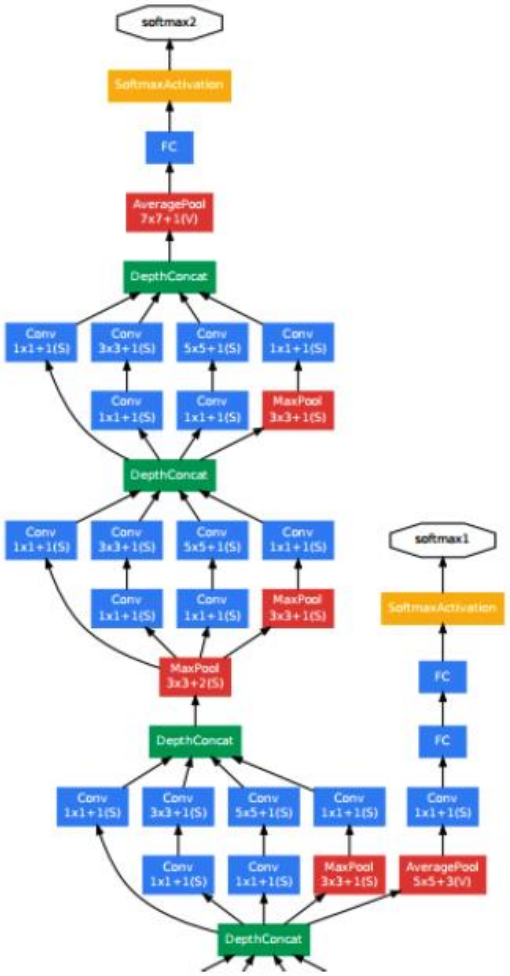




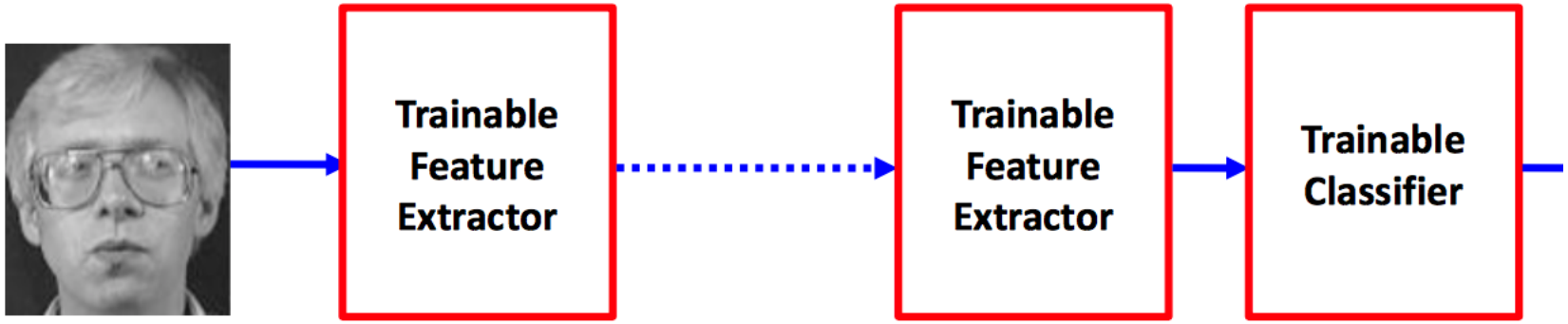


DL building blocks

- Activation Functions
  - Linear and ReLU
  - Sigmoid and tanh
  - etc.
- Layers
  - Fully-connected
  - Convolutional & pooling
  - Recurrent
  - Residual (ResNets)
  - Transformers
  - etc.
- Loss functions
  - Cross-entropy, MSE, etc.



DL learns hierarchical representations



- In Language Models: hierarchy in syntax and semantics
  - Words → Parts of Speech → Sentences → Stories
- Is Vision: hierarchy in composition
  - Pixels → Edges → Textons → Parts → Objects → Scenes

When did DL become really popular?

That was the view of people in computer vision until 2012. Most people in computer vision thought this stuff was crazy, even though Yann LeCun sometimes got systems working better than the best computer vision systems, they still thought this stuff was crazy, it wasn't the right way to do vision. They even rejected papers by Yann, even though they worked better than the best computer vision systems on particular problems, because the referees thought it was the wrong way to do things. That's a lovely example of scientists saying, "We've already decided what the answer has to look like, and anything that doesn't look like the answer we believe in is of no interest."

In the end, science won out, and two of my students won a big public competition, and they won it dramatically. They got almost half the error rate of the best computer vision systems, and they were using mainly techniques developed in Yann LeCun's lab but mixed in with a few of our own techniques as well.

MARTIN FORD: This was the ImageNet competition?

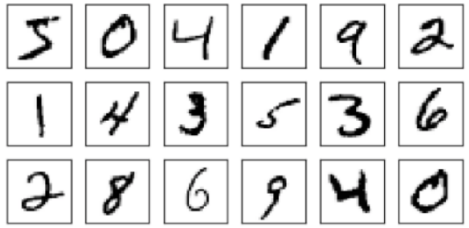
GEOFFREY HINTON: Yes, and what happened then was what should happen in science. One method that people used to think of as complete nonsense had now worked much better than the method they believed in, and within two years, they all switched. So, for things like object classification, nobody would dream of trying to do it without using a neural network now.

(Excerpt from "Architects of Intelligence")

AlexNet achieved 15.4% error on top-5 in 2012

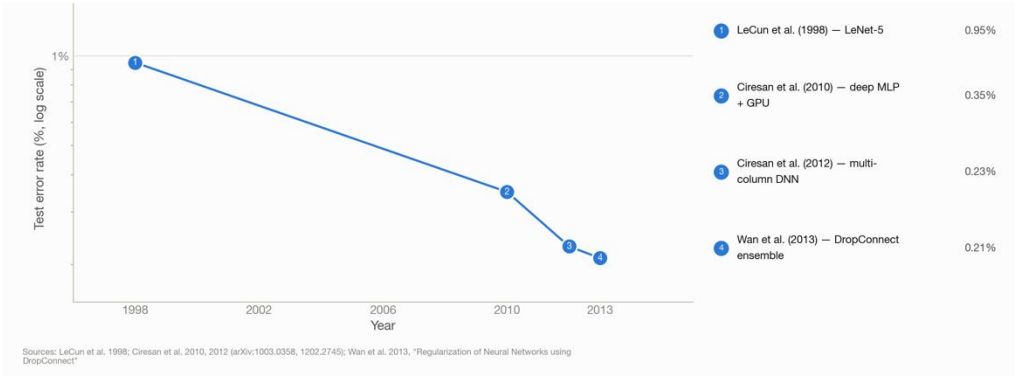
- 2nd best wasn't close: 26.2%
- (nowadays ~3% error on ImageNet)

DL: Driven by benchmark datasets



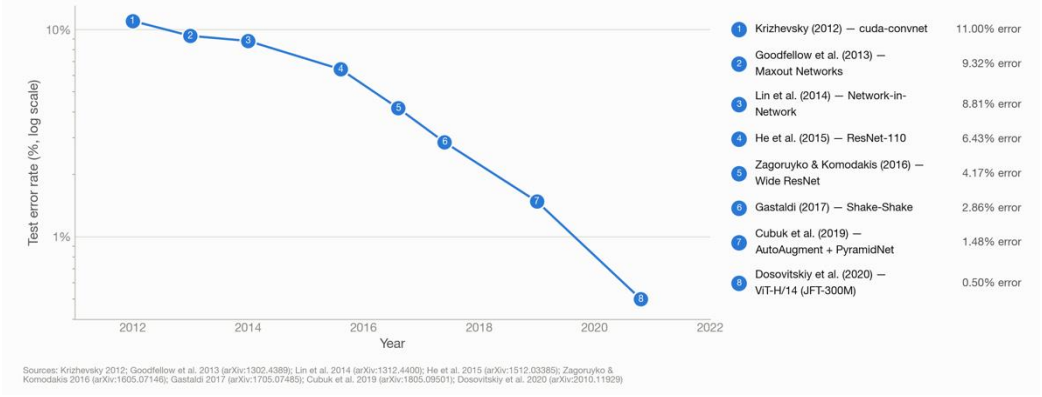
MNIST (1998)

- 60,000 examples
- 10 classes
- Features: 28x28x1
- http://yann.lecun.com/exdb/mnist



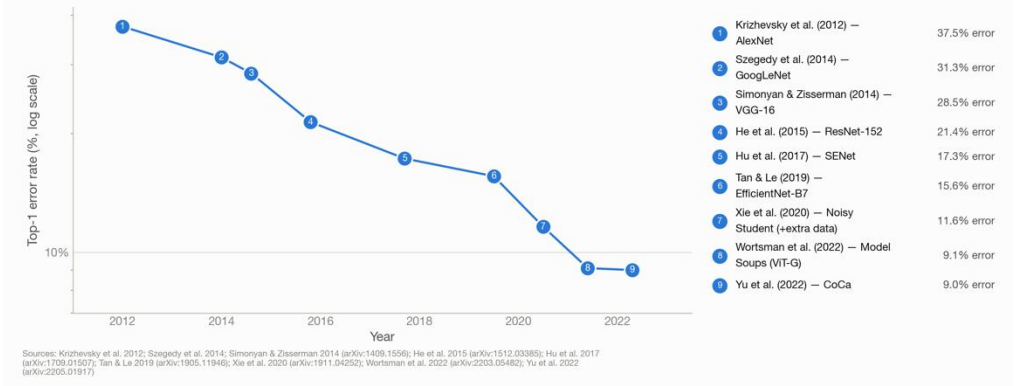
CIFAR-10/CIFAR-100 (2009)

- 60,000 examples
- 10 or 100 classes
- Features: 32x32x3
- https://www.cs.toronto.edu/~kriz/cifar.html



ImageNet (~2010)

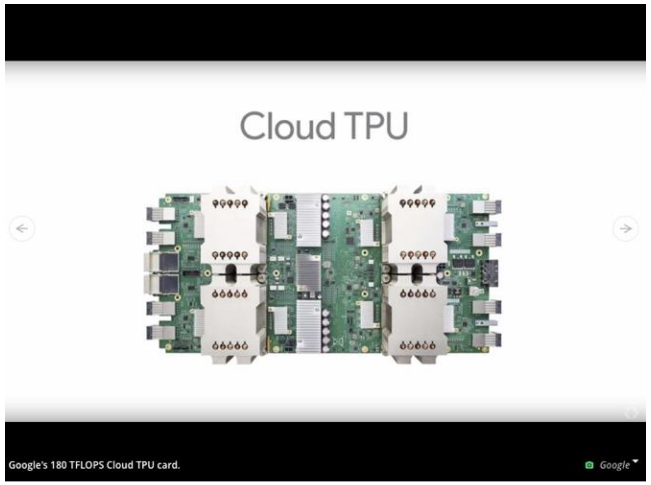
- 14 million images
- Features: full resolution
- https://www.image-net.org



Today: A Brief History of DL

1. Artificial neurons
2. Multilayer neural networks
3. Deep Learning
4. **The DL Hardware & Software Landscape**
5. Current Research Trends

Developing Specialized Hardware



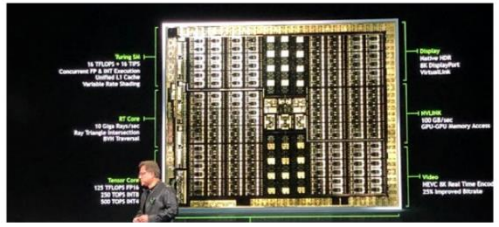
https://arstechnica.com/gadgets/2018/07/the-ai-revolution-has-spawned-a-new-chips-arms-race/

https://developer.arm.com/products/processors/machine-learning/arm-ml-processor

Opinion: New Nvidia chip extends the company's lead in graphics, artificial intelligence

By Ryan Shrout
Published: Aug 14, 2018 2:35 p.m. ET

The only question that remains: How big is Nvidia's advantage over its rivals?



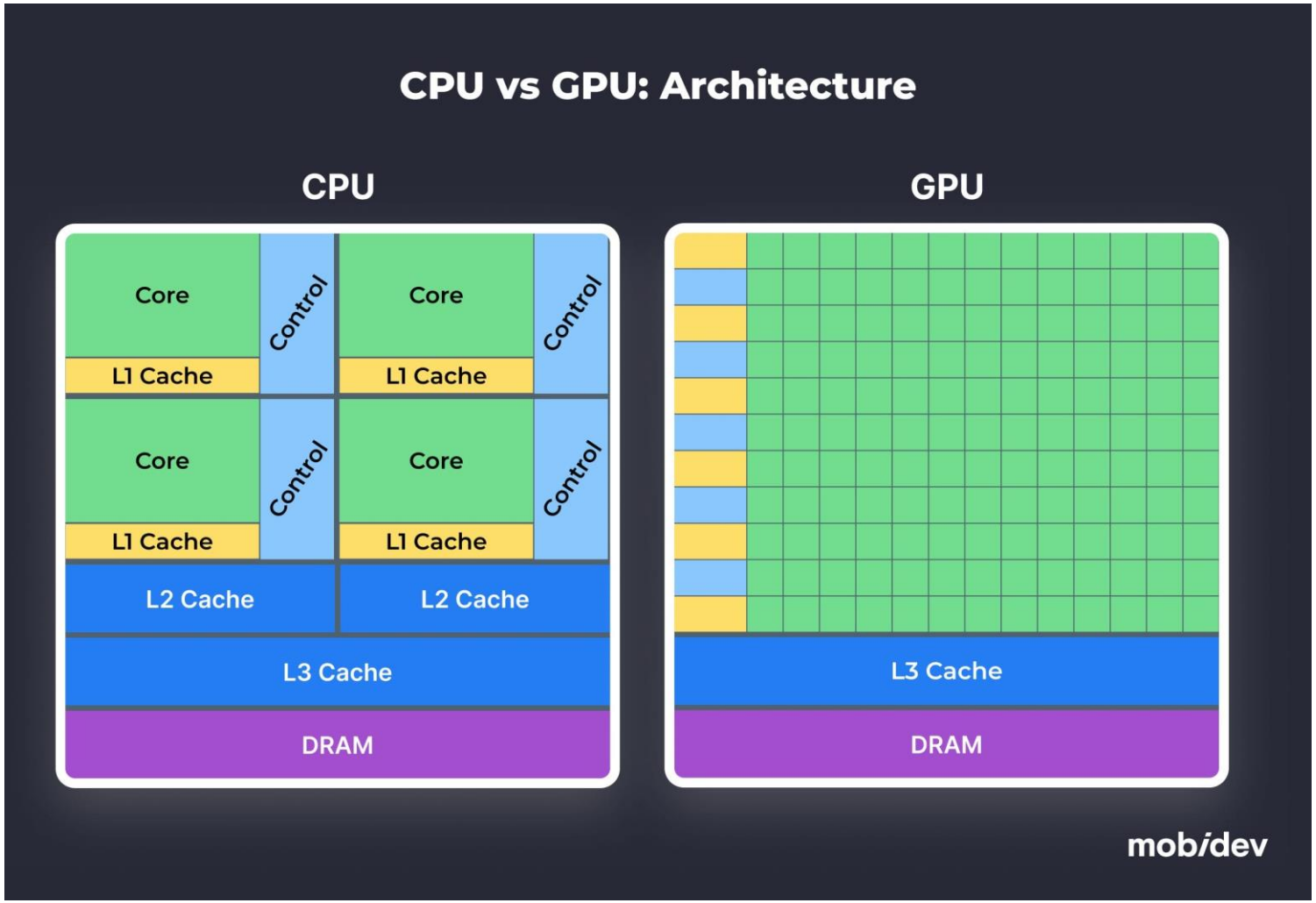
https://www.marketwatch.com/story/new-nvidia-chip-extends-the-companys-lead-in-graphics-artificial-intelligence-2018-08-14

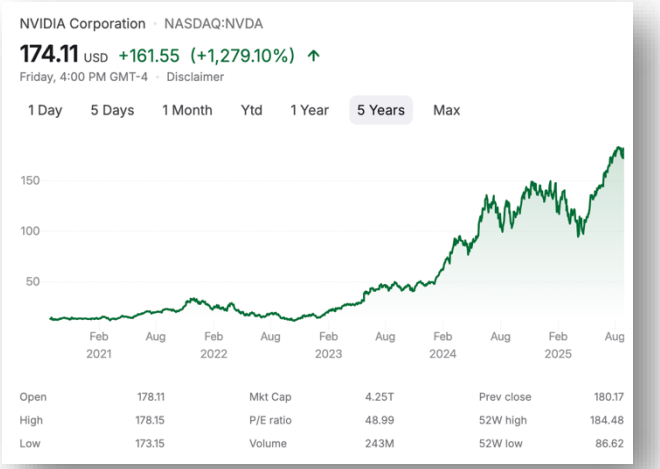
TECHNOLOGY NEWS NOVEMBER 28, 2018 / 2:59 PM / 2 MONTHS AGO

Amazon launches machine learning chip, taking on Nvidia, Intel

https://www.reuters.com/article/us-amazon-com-nvidia/amazon-launches-machine-learning-chip-taking-on-nvidia-intel-idUSKCN1NX2PY

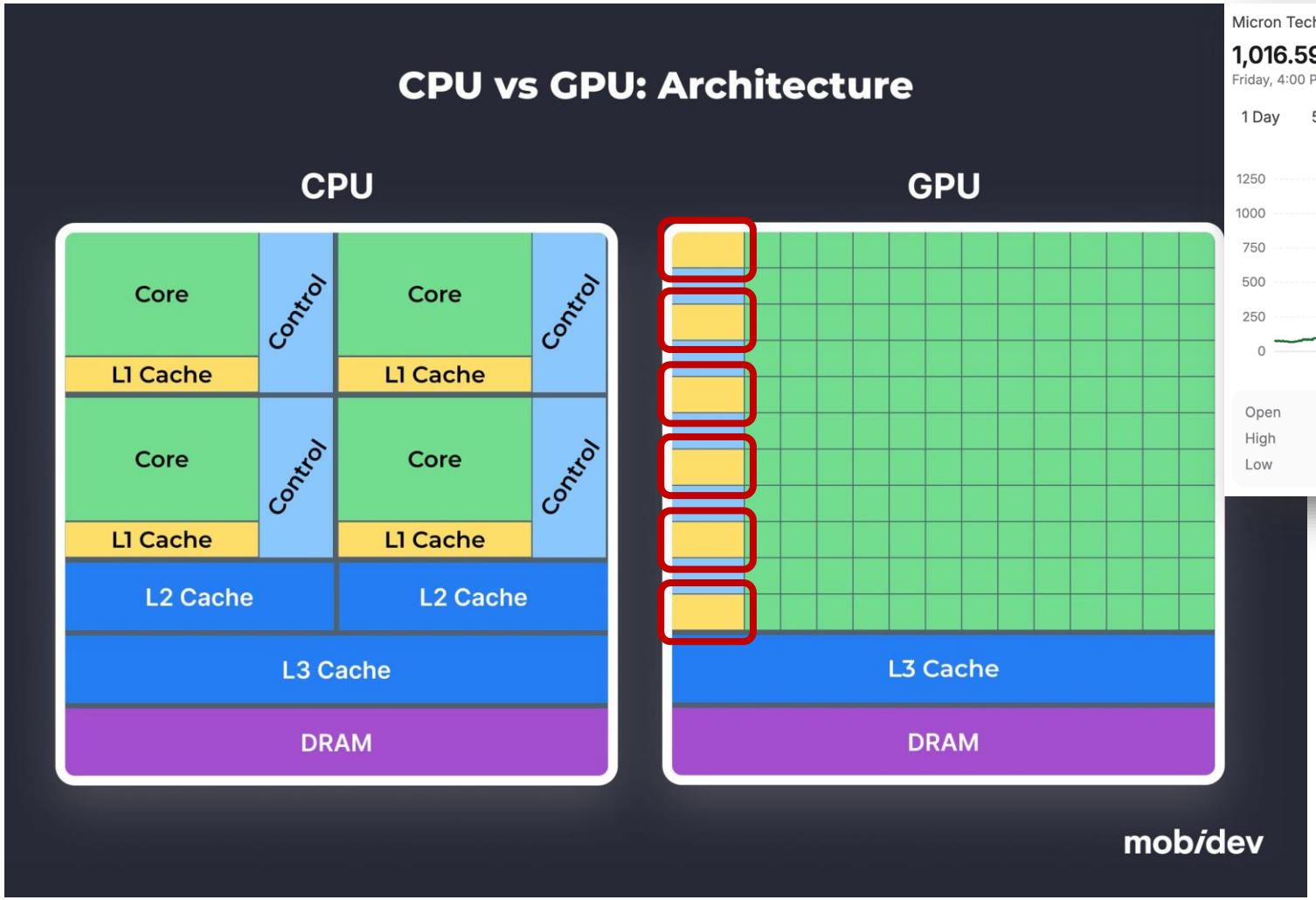
Hardware for Matrix Multiplications

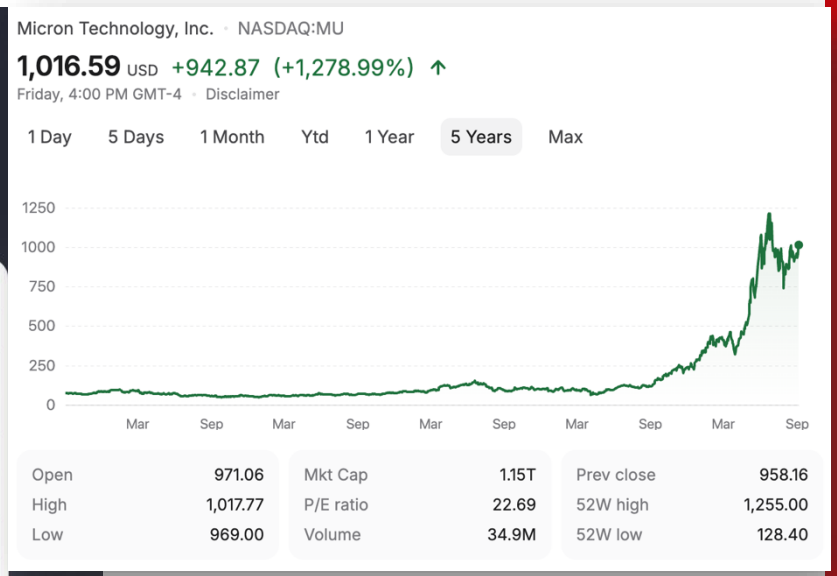




https://mobidev.biz/blog/gpu-machine-learning-on-premises-vs-cloud

GPUs make a new bottleneck: Memory

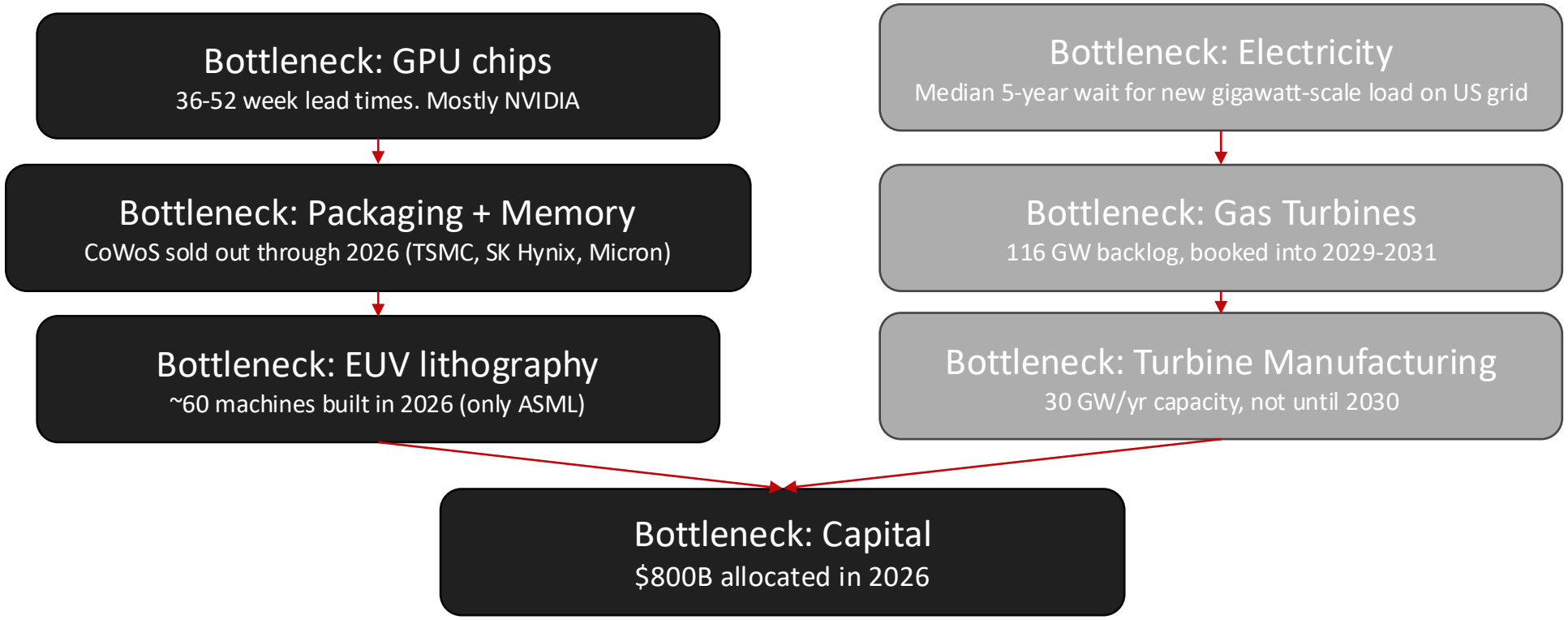




https://mobidev.biz/blog/gpu-machine-learning-on-premises-vs-cloud

An entire infrastructure stack

- GPU / AI compute demand is no longer a chip story
- Now impacting power grids, capital markets, trade policy, and national strategy



Today: A Brief History of DL

1. Artificial neurons
2. Multilayer neural networks
3. Deep Learning
4. The DL Hardware & Software Landscape
5. **Current Research Trends**

Self-supervised learning / pretext tasks

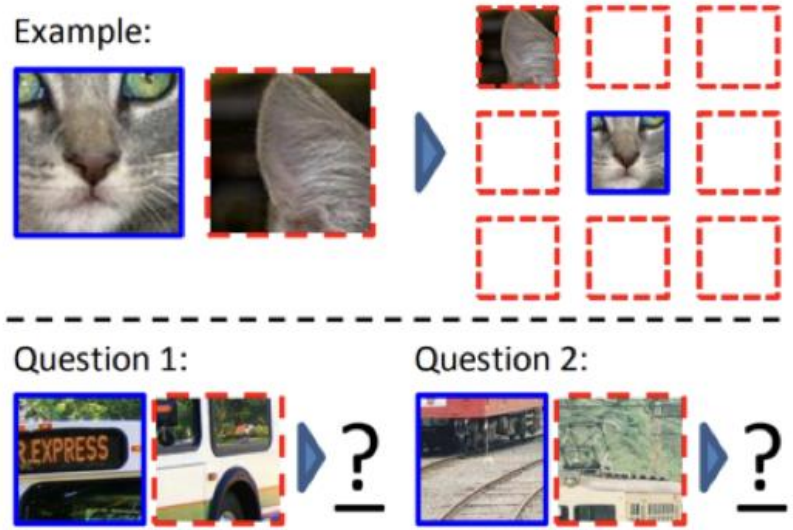
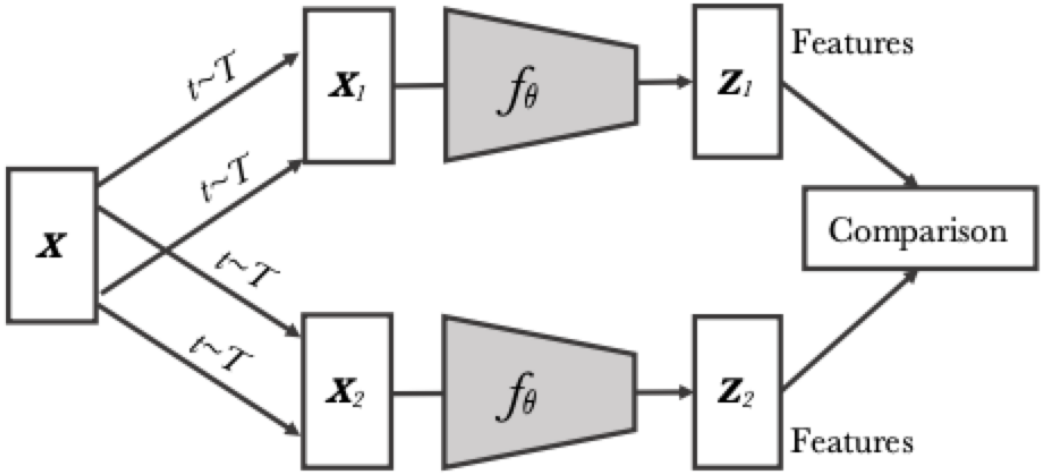


Figure 1. Our task for learning patch representations involves randomly sampling a patch (blue) and then one of eight possible neighbors (red). Can you guess the spatial configuration for the two pairs of patches? Note that the task is much easier once you have recognized the object!

Answer key: Q1: Bottom right Q2: Top center

Lee, H. Y., Huang, J. B., Singh, M., & Yang, M. H. (2017). Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 667-676).



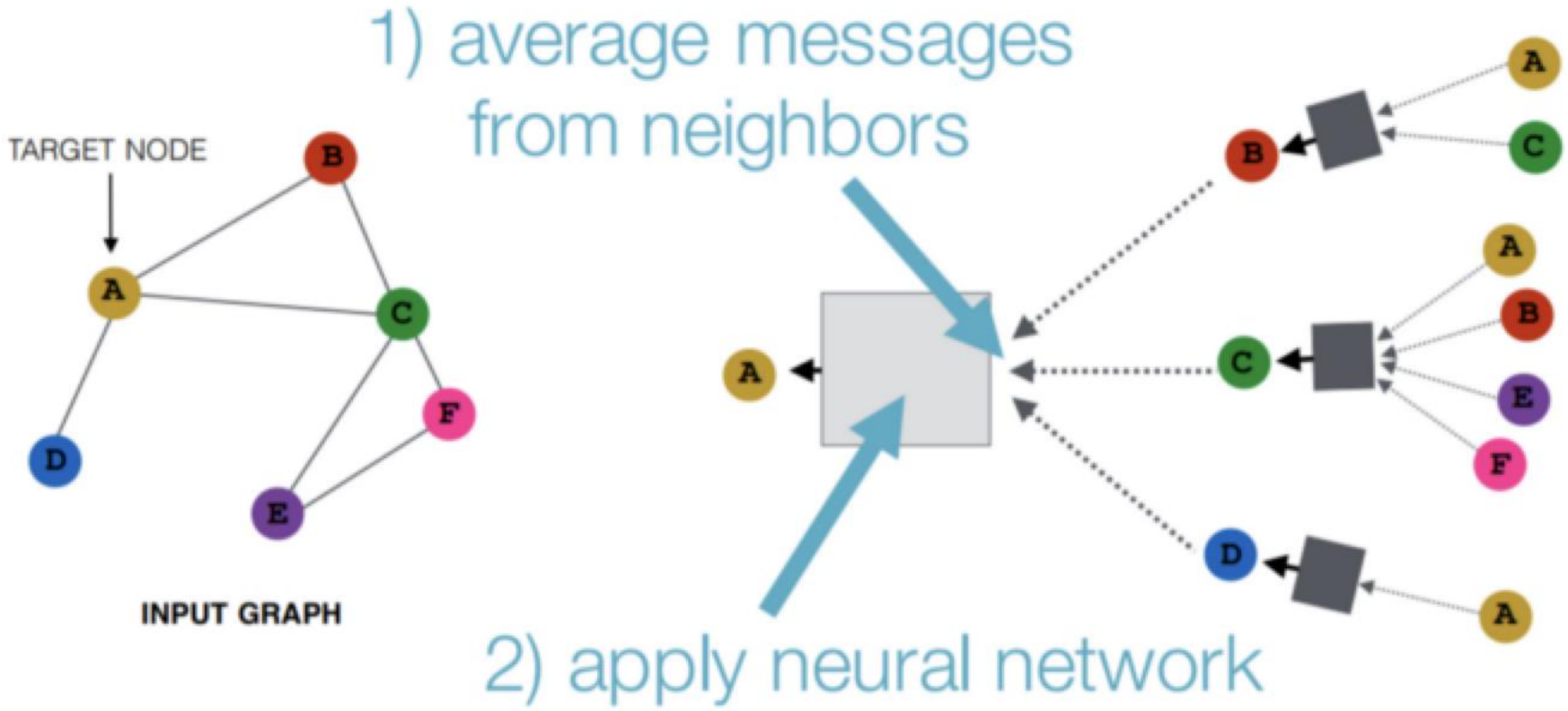
contrastive learning gained popularity in self-supervised representation learning

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

Instance discrimination compares features from different transformations of the same images to each other

DL on all kinds of structured data (graphs, etc.)

A gentle introduction to graph neural networks:
https://heartbeat.fritz.ai/introduction-to-graph-neural-networks-c5a9f4aa9e99



Massive unsupervised learning

From GPT-1 (2018) to GPT-4:

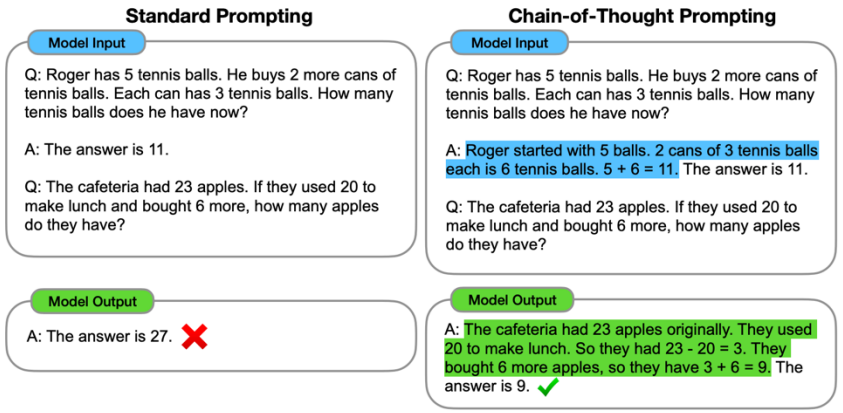
- **Architecture:**
  - **Scale:** Variety of options, with biggest (1.5B params → >1T params):
    - Block size (max context): 512 → 128k
    - Layers: 12 → >96
    - Attention Heads: 12 → >96
    - Embedding Dim: 768 → >12,288
    - Vocab: 40k → >50k tokens
  - Tokenizer: Includes image patches for multimodal
  - Mixture-of-Experts
- **Training:**
  - Dataset: BookCorpus (5GB) → Private 13T tokens (~50TB)
  - Reinforcement learning for alignment

Compute-Optimal Scaling: Data Matter as Much as Size

- *Kaplan et al.* (2020): scaling laws for parameter counts
- *Hoffmann et al.* (2022), "Chinchilla": compute-optimal training balances parameters AND training tokens
  - Training compute ≈ parameters × tokens
- Finding: **most pre-2022 LLMs were undertrained relative to their parameter count**
- Since 2022: growing focus on data curation & quality (e.g., FineWeb, Dolma, RedPajama) and synthetic data (synthetic textbooks, distillation datasets)
- Takeaway: **"bigger" alone is no longer the story — what and how much data matters as much as model size**

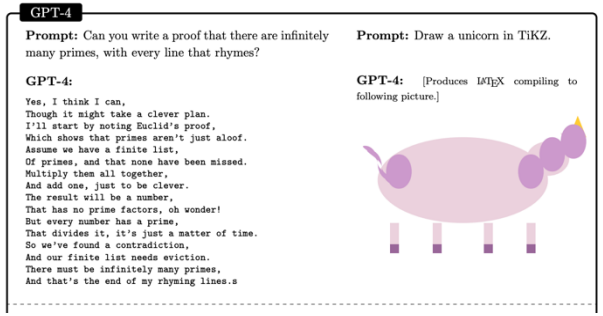
Alignment of large systems to our values

[Chain-of-Thought Prompting Elicits Reasoning in Large Language Models]

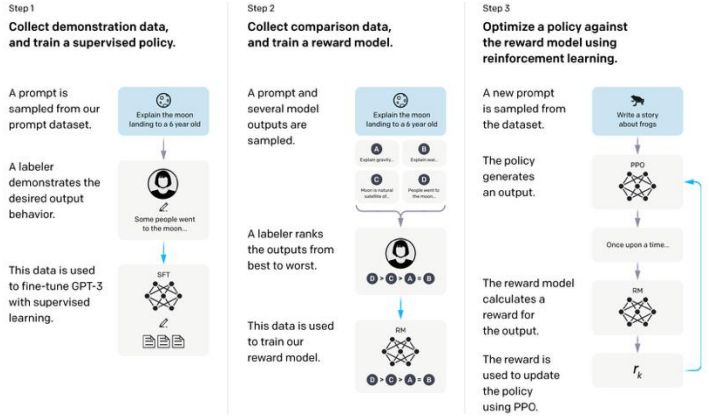


Experiments with Large Language models

[Sparks of Artificial General Intelligence: Early experiments with GPT-4]



[Training language models to follow instructions with human feedback (RLHF)]



From RLHF to DPO and RLVR

- 2022: (*Ouyang et al.*), “**R**einforcement **L**earning from **H**uman **F**eedback”
  - Learn a reward model from human preferences, optimize the policy via RL (PPO). **Effective, but complex and unstable to train**.
- 2023: (*Rafailov et al.*), "**D**irect **P**reference **O**ptimization”
  - Reformulates the RLHF objective as a direct classification loss over preference pairs. **No separate reward model, no RL loop.** [Lec 21]
- 2024–2025: (**R**einforcement **L**earning with **V**erifiable **R**ewards
  - Replace human/learned reward with programmatic verifiers. **Unit tests for code, symbolic solvers for math, proof checkers for logic**. RLVR powers today's reasoning models (e.g., OpenAI's o1, DeepSeek-R1). [Lec 22]
- **Pattern: each step removed a source of noisy supervision — human raters → preference pairs → objective verifiers**

2025's “Open Direction” Is 2026's Standard Method

- Last year (Lecture 02, Fall 2025), we listed "verifiable rewards" as an open research direction
- One year later: RLVR is how frontier reasoning models are trained — we'll cover this in Lecture 22
- The field moves quickly
  - Hottest area: scaling test-time / inference-time compute. Chain-of-thought, self-consistency sampling, tool use, tree-of-thought search
  - Key idea: capability can improve via more compute at inference time, not just bigger training runs

Many directions open…

- Verifiable Rewards (Code generation, math calculations, etc.)
- Vision-Language models and learning with multimodal data
- Large-scale reinforcement learning
- Model uncertainty, hallucinations
- Model editing, interpretability
- Model synthesis, connections, communication
- Post-training: Reward-hacking
- Agentic systems / tool use at scale
- Compute-optimal scaling
- Calibrated inference
- Efficient inference — distillation, quantization, speculative decoding
- Mechanistic interpretability — sparse autoencoders, circuit-level analysis
- World models!

Next Lecture

Stats / linear algebra / calculus review

Questions?