Lecture 1: Introduction

CS/DATA 182: Designing, Visualizing, and Understanding **Deep Learning** Networks

08/29/2024

Course staff

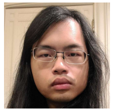
Enrollment questions: cs-enrollments@berkeley.edu

Instructors

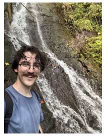
Eric Kim
ekim555@berkeley.edu

Naveen Ashish
nashish@berkeley.edu

TAs

	Rami Ratl Mrad	ramiratlmrad@berkeley.edu
	Vivek Verma	vivekverma@berkeley.edu
	Yuxi Liu	yuxi_liu@berkeley.edu
	William Chen	verityw@berkeley.edu
	Jerry He	hzyjerry@berkeley.edu

Tutors

	Amogh Tantradi	amogh_t2003@berkeley.edu
	Richard Chao	rqchao@berkeley.edu
	Ryan Campbell	ryancampbell@berkeley.edu
	Jinjian Liu	leifmax@berkeley.edu

General course information

Course website: https://datac182fa24.github.io/

- This course will taught in HYBRID mode
  - Ashish, in-person; Kim, on Zoom
- Relevant prerequisites:
  - Strong background in probability (CS 70, Stat 134, or similar)
  - Strong background in vector calculus (e.g., can you take the gradient of a matrix vector product)
  - Strong background in machine learning is preferred (CS 189, or similar)
  - Strong programming skills in Python (e.g., can you learn new libraries quickly)

Lectures and Discussions

Tuesday & Thursday : 6:30-8PM

- Lecture recordings will be available some time after the live lecture
- There will be various guest lectures which may not be recorded
- Discussion sections (schedule and locations will be available soon)

Discussion sections and office hours

https://datac182fa24.github.io/schedule/

- You are encouraged to attend any discussion section that you like that has room
- It is **very important** that you read the office hours policy on Edstem
  - You should come to OH **prepared** and **with reasonable expectations**
  - You should actively look for **other students** working on the same problems
  - You will be limited to a **10 minute window** when there is a queue

Homework assignments

DSP students: `Please contact the instructors at nashish@berkeley.edu | ekim555@berkeley.edu`

- There are **four** homework assignments total, released every three weeks or so
- You will have ~2.5 weeks to complete each homework
- Each homework assignment is worth **15%** of your overall grade
- There are no homework drops, but there are five total slip days for the semester to be **reserved for emergencies** — no other late homework will be accepted
- You are encouraged to discuss problems, but the code/writeup must be your own — infractions will result in (at least) an immediate zero on the assignment
- There is NO final exam BUT there is a FINAL PROJECT in lieu of the exam

Exams

DSP students: Please contact the instructors at nashish@berkeley.edu | ekim555@berkeley.edu

- There is ONE midterm exam for 182 students, **both in person if permitted**
- Exam infractions are serious are will result in (at least) significant points deducted

Final project

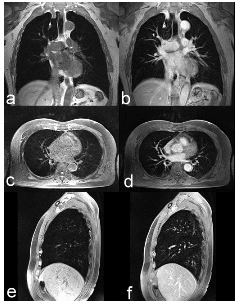
- In lieu of MT2 students will complete an open-ended final project
- The expected novelty and quality of this project is such that it could reasonably be submitted to a research conference or journal, possibly with additional work
- The final project will be worth **25%** of the overall grade for students
- More details about the final project, including timeline and milestones, will be announced as the semester progresses

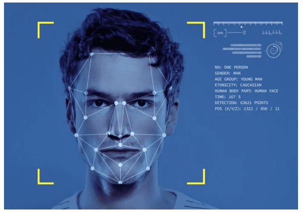
Grading

- This course grading will follow a “mix” of curved and straight-scale

Why are we here ?

- *The engine **driving breakthroughs** across nearly every field.*
- *In **healthcare**, deep learning is revolutionizing **diagnostics** by analyzing medical images to detect diseases like cancer at earlier stages than ever before.*
- *In **finance**, it powers algorithms that predict market trends and detect fraud with remarkable accuracy.*
- *Deep learning fuels **full self driving***
- *The **recommendation systems** on streaming platforms that suggest your next favorite show*
- *The **facial recognition** technologies used in security systems.*
- *It's transforming industries like **manufacturing** with predictive maintenance and robotics,*
- *Enhancing natural language processing for **virtual assistants** and translation services,*
- *Enabling groundbreaking research in **climate science** by modeling complex environmental patterns*
- *Basis of **transformers** (and "attention") : the basis of **Generative AI and LLMs***

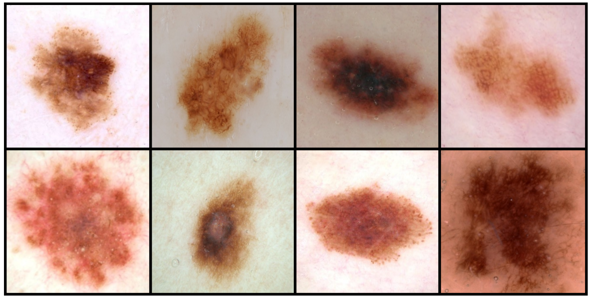




*this course is to empower you to become a good potential **applicator**, as well as **creator** of deep learning*

deep learning: IMPACT

Success stories



Melanoma detection

Protection structure prediction
(green is ground truth)

Image synthesis

Large Language Models !

- PROMPT: *"write a proof of the fact that there are infinitely many primes; do it in the style of a Shakespeare play with two parties arguing over the proof"*

GPT-4: Proof of infinitely many primes

```
FIRST PARTY:
Thou dost not understand, my friend,
There simply is no upper end
To primes, those numbers indivisible,
No matter if it seems impossible.

SECOND PARTY:
But how dost thou so surely know?
To prove this truth, what canst thou show?

FIRST PARTY:
Consider this, my doubtful peer,
A clever proof that will make clear:
Assume that there's a final prime,
The largest one we'll see in time.
Let's call this number "P" for short,
And then, my friend, we will purport
To build a number "Q" that's new,
One larger than "P" and prime too.

SECOND PARTY:
But how, my friend, will this be done?
The proof is not yet fairly won.

FIRST PARTY:
It's simple, friend, just wait and see,
We'll multiply the primes, that's key.
All primes from two to "P" combined,
Will make this new "Q" we will find.
But one thing more we must include,
To make the proof more watertight,
We'll add a one onto the end,
And now the proof is in our sight.

SECOND PARTY:
I see it now, it's crystal clear,
For "Q" must be a prime, my dear.
It can't divide by any prime,
All smaller than itself in time.
And since we chose "P" as the last,
This proves that our assumption's past.
There is no final prime, we see,
The numbers go on endlessly.
```

this is …..TRANSFORMATIONAL

ARCHITECTURAL

+

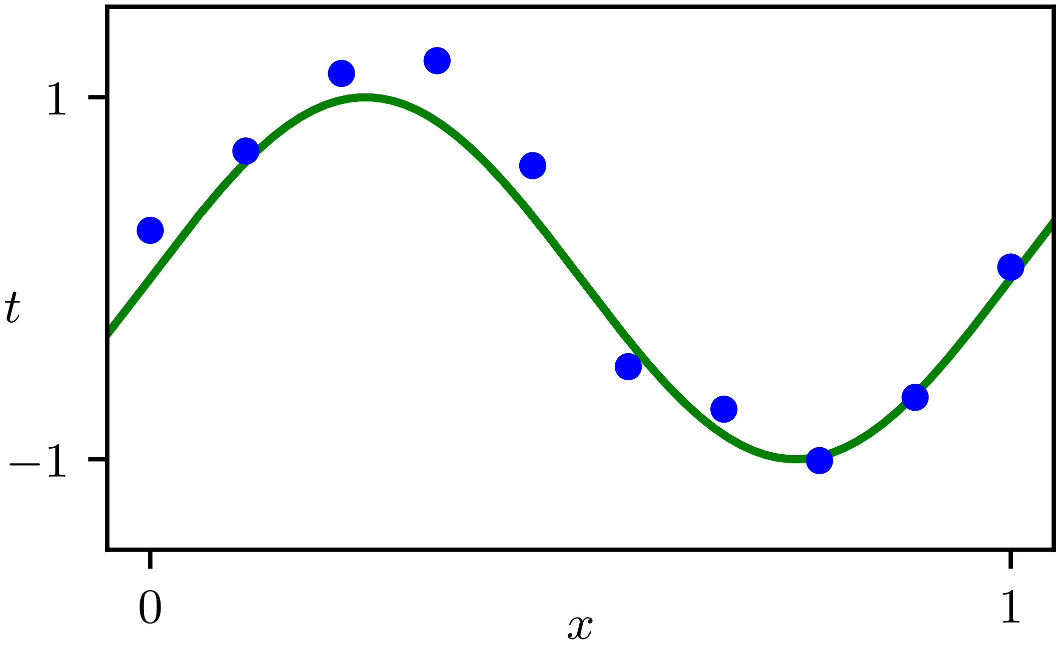
COMPUTATIONAL POWER

evolution

→

Deep Learning Transformation

Generalization



Linear models, Error functions

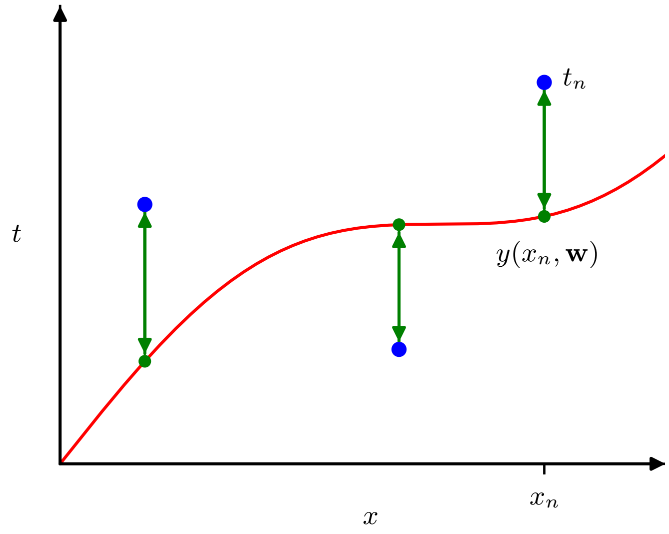
- *Synthetic* data

- *Linear* models

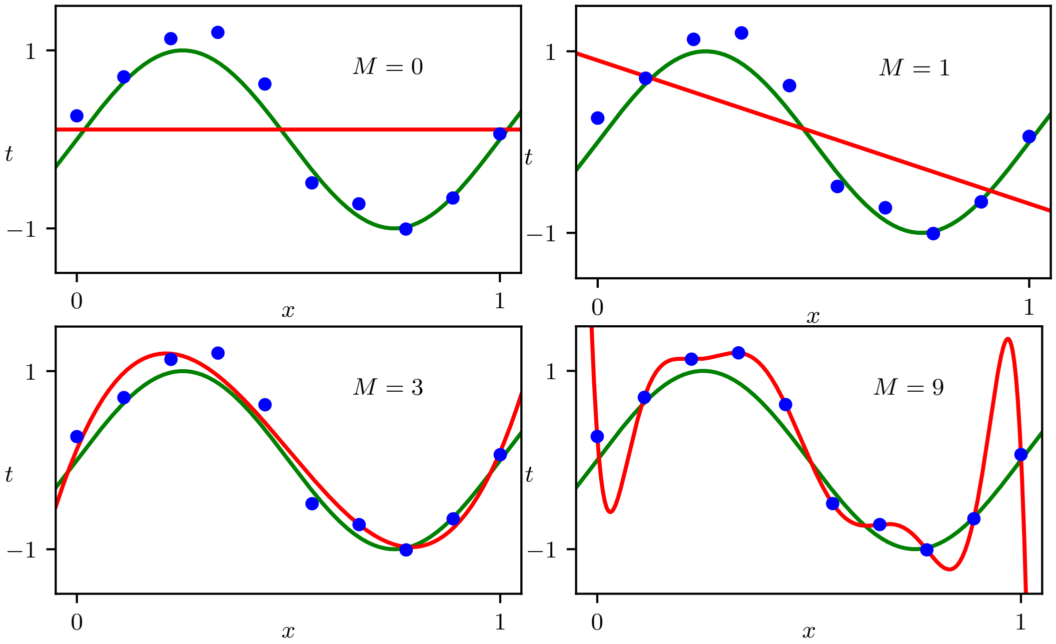
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

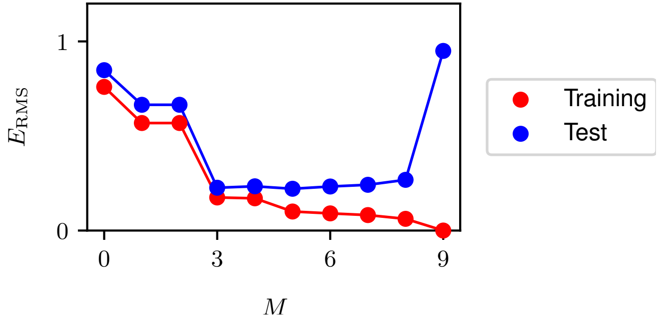
- The *error function*

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$



Model *complexity*

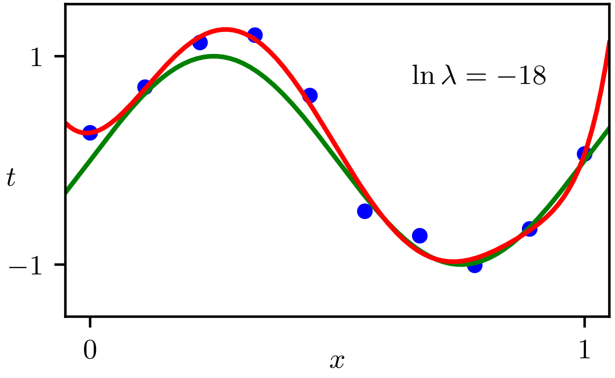


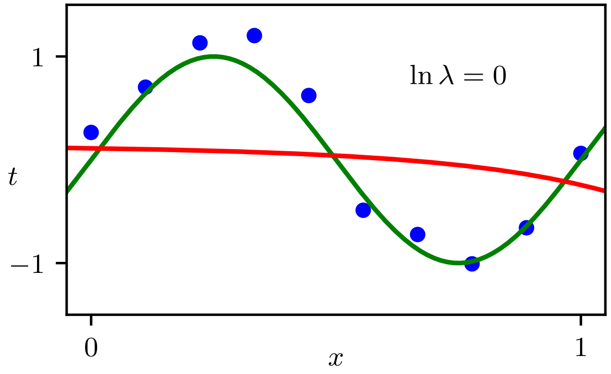


- *Overfitting*
- The *root-mean-square* ***error*** RMSE

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2}$$

Regularization





$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- λ .... is a *penalty*

Model selection

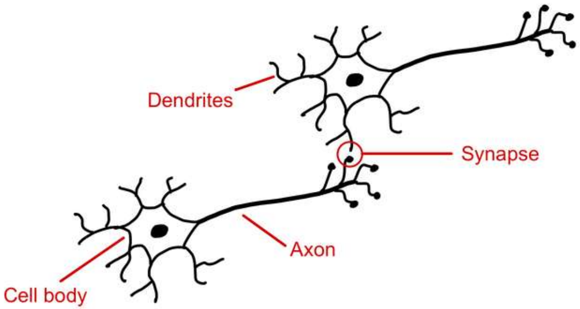
Table of the coefficients $\mathbf{w}^{\star}$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

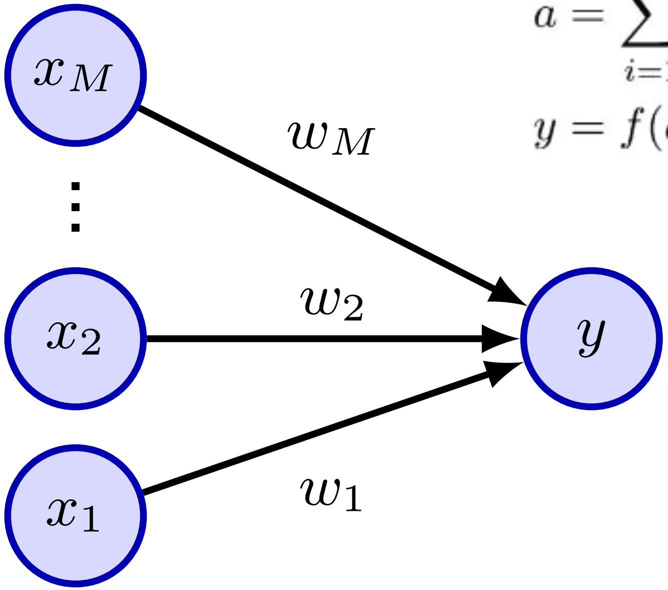
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^{\star}$	0.11	0.90	0.12	0.26
$w_1^{\star}$		-1.58	11.20	-66.13
$w_2^{\star}$			-33.67	$1,665.69$
$w_3^{\star}$			22.43	$-15,566.61$
$w_4^{\star}$				$76,321.23$
$w_5^{\star}$				$-217,389.15$
$w_6^{\star}$				$370,626.48$
$w_7^{\star}$				$-372,051.47$
$w_8^{\star}$				$202,540.70$
$w_9^{\star}$				$-46,080.94$

- *Hyperparameters*

DEEP Learning

Perceptron, Single LAYER





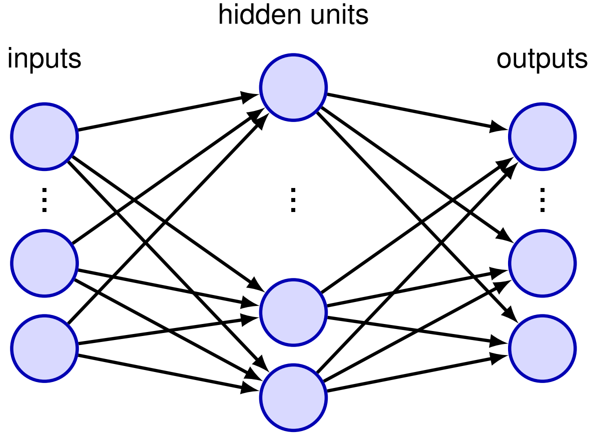
$$a = \sum_{i=1}^{M} w_i x_i$$

$$y = f(a)$$

- SINGLE layer
- *Activation function*

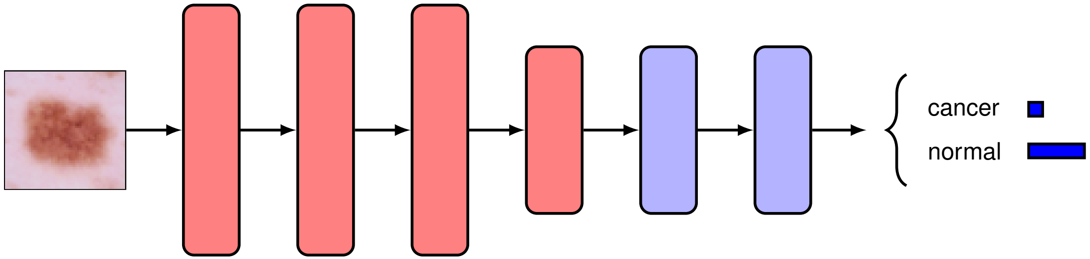
$$f(a) = \begin{cases} 0, & \text{if } a \leqslant 0, \\ 1, & \text{if } a > 0\,. \end{cases}$$

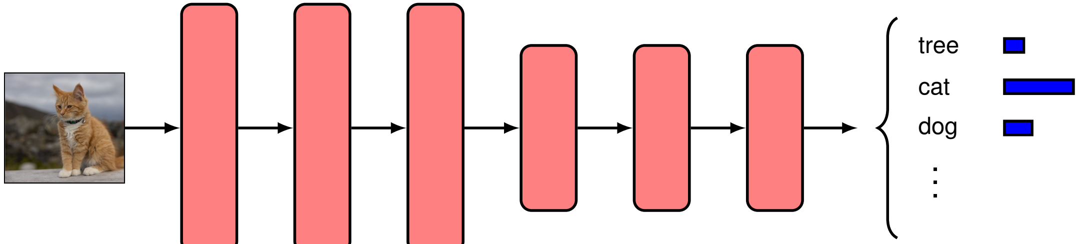
Backpropagation



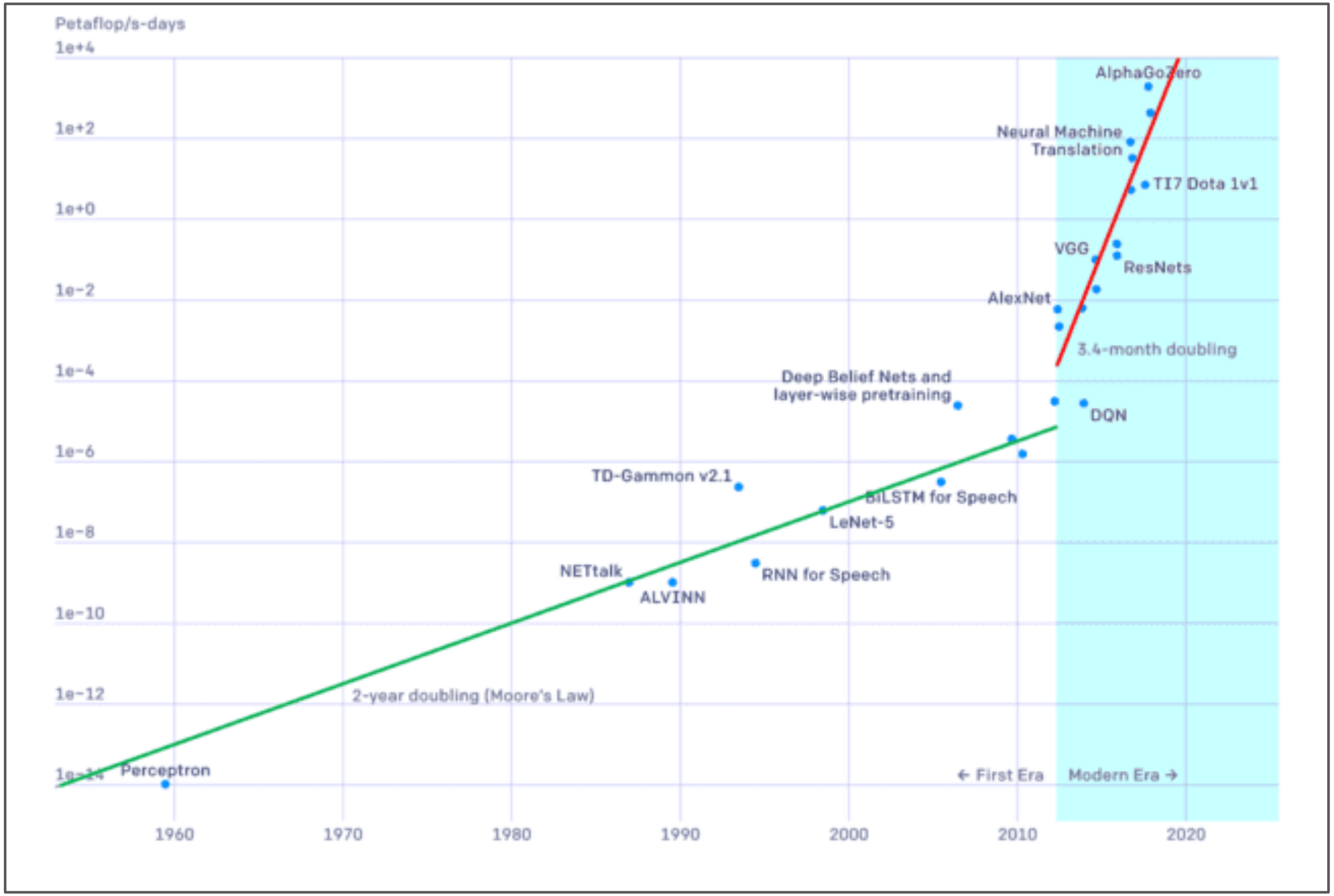
- *Error* backpropagation
- *Feed-forward*
- *Stochastic gradient descent*
- *Inductive bias*
- *Hidden units*

Deep Learning: *Multiple* layers

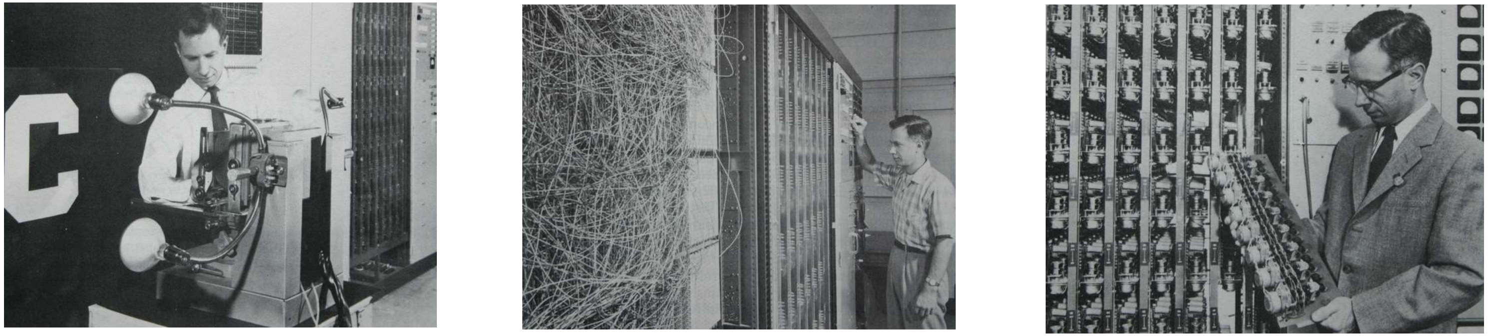




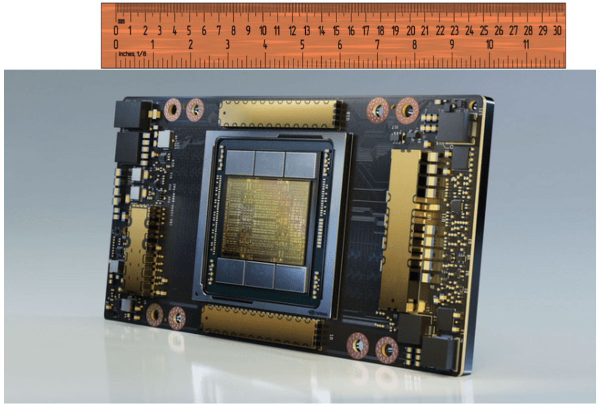
Deep Learning: Compute Needs



Compute



Mark 1 Perceptron: 1969



Today: The GPU

Deep NNs

- *Hidden* layers
- *Representation* learning & *Unsupervised* feature learning
- *Transfer learning*
- *Fine tuning*
- *Foundational* models

Summary

Supervised / Unsupervised

Generative models/AI

Auto-regressive

Linear models

Error-function

Overfitting

RMSE

Regularization

Hyperparameters

Unsupervised feature learning

Perceptron/Single-layer

Activation function

Layers

Backpropagation

Feed-forward

Hidden layers

Self-supervised feature learning

Representation learning

Fine tuning

Foundational model