CIS 6930 Topics in Computing for Data Science
Week 1a: Deep Learning Basics

9/9/2021

Yoshihiko (Yoshi) Suhara

2pm-3:20pm & 3:30pm-4:50pm (5-6pm office hour)

Course Instructor Team

Yoshi Suhara
(Course Instructor)

Andrew Reilly
(Teaching Assistant)

Course Logistics

Class Time

- **Class time: Tue/Thu 2pm-3:20pm (*)**
  - (*) **2pm-4:50pm** on Thu 9/9, Tue 9/14, Thu 9/16, Tue 9/21, Thu 9/23, Tue 11/16, Thu 11/18
- **Office hours:** Thu after the class
  - *another office hour TBD

Course Plan (1/2) (Bold-faced = doubled pace!)

- **Week 1: Deep Learning Basics (Thu 9/9)**
- **Week 2: AutoEncoder (Tue 9/14)**
- **Week 3: Convolutional Neural Networks (Thu 9/16)**
- **Week 4: GAN (Tue 9/21)**
- **Week 5: Word embeddings: Word2vec, GloVe (Thu 9/23)**
- Week 6: Recurrent Neural Networks (Tue 9/28, Thu 9/30)
- Week 7: Review/Project pitch & Mid-term (Tue 10/5, Thu 10/7)
- Fall Break

Course Plan (2/2) (Bold-faced = doubled pace!)

- Week 8: Transformers (Tue 10/19, Thu 10/21)
- Week 9: Pre-trained Language Models (Tue 10/26, Thu 10/28)
- Week 10: More Machine Learning Techniques (Tue 11/2, Thu 11/4)
- Week 11: More Deep Learning Techniques for NLP (Tue 11/9, Thu 11/11)
- **Week 12: Advanced Techniques and Challenges (Tue 11/16)**
- **Week 13: Final project presentations (Thu 11/18)**

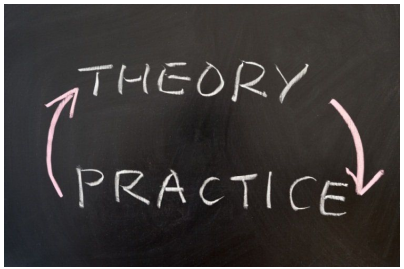
Grading

- In-class exercises & class participation (15%)
- Weekly or biweekly assignments (20%)
- Midterm exam (30%)
- Project report (25%)
- Project presentation (10%)

Teaching Style

- Fully online via Zoom (* recording permission)
  - I'd like to make it interactive as much as possible
  - Please turn on cameras. This helps me feel the in-class atmosphere :)

- Theory & practice
  - i.e., lectures + hands-on session/assignments/term projects
  - My focus is to teach **Why? (Theory)** and **How? (Practice)**



python + colab PyTorch

In-class Policy & Communication

- Please feel free to interrupt me and ask questions anytime
  - Can you click “Raise Hand” button?

- Let’s use Slack for questions. Please ask questions anytime.
  - Is everybody in the Slack channel?

Any questions?

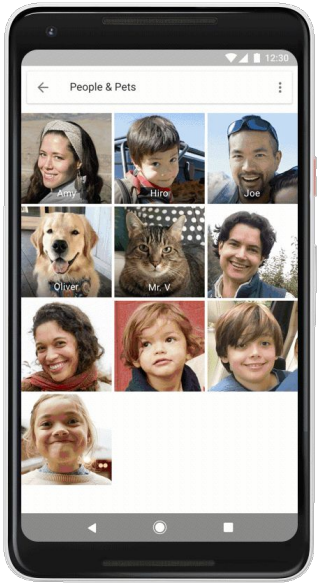
Week 1a: Deep Learning Basics (-3:20pm)

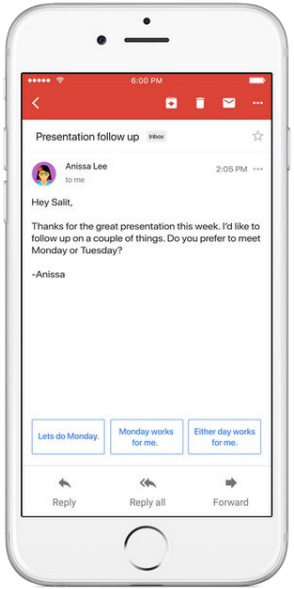
- Recap: Machine Learning Basics
- History of Neural Networks
- Neural Networks 101

Recap: Machine Learning Basics

What is Machine Learning?

- A magic that makes your laptop/smartphone really smart

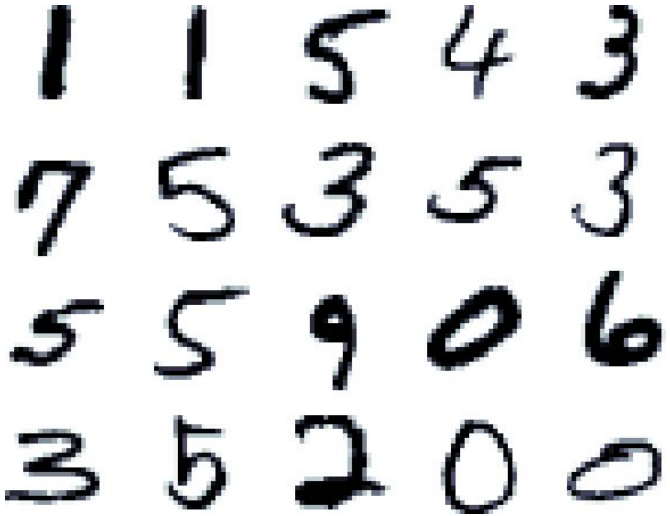




Machine Learning Categories

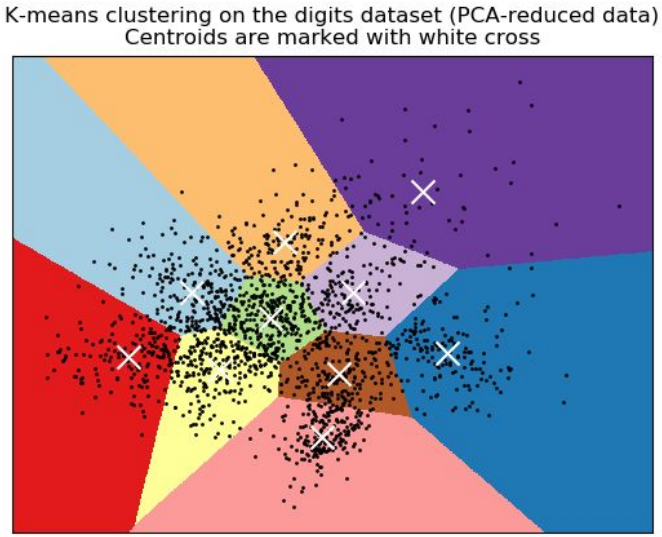
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

(1) Supervised Learning



(2) Unsupervised Learning

- Clustering or Representation Learning for Visualization or better supervised learning models



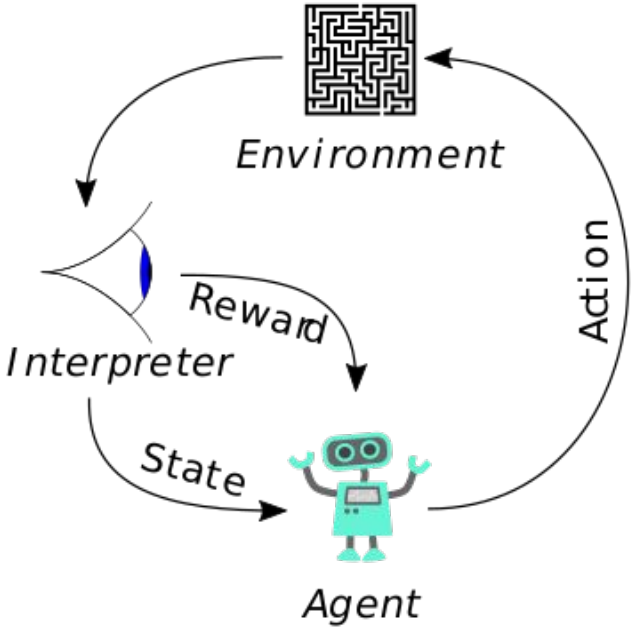
Clustering + PCA

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Topic Models

(3) Reinforcement Learning

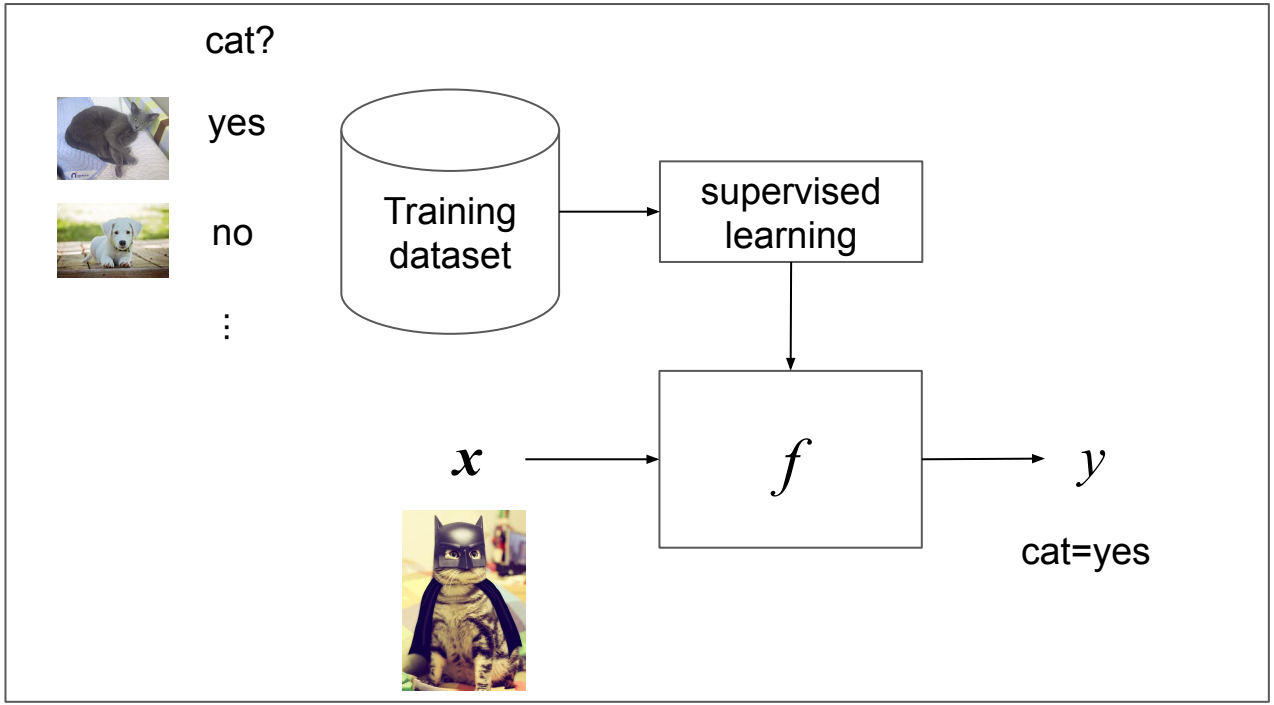


RL is not be covered by this course

Supervised Learning

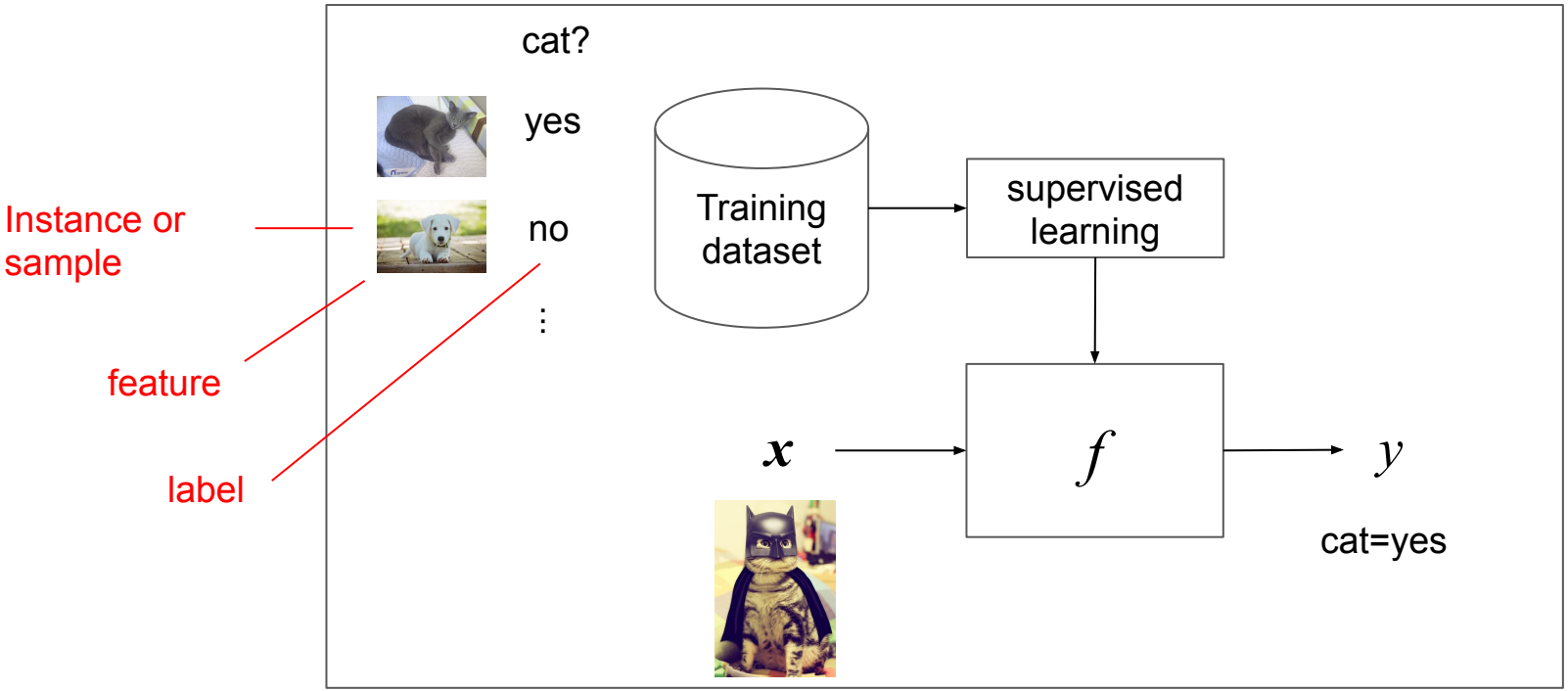
Supervised Learning

Supervised learning is a framework that builds a predictive model based on "labeled" training data



Supervised Learning

Supervised learning is a framework that builds a predictive model based on "labeled" training data



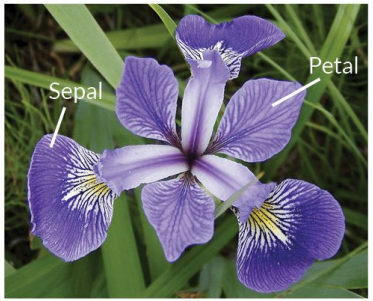
Supervised Learning = Learning a function

Supervised learning algorithm learns a function that maps a **feature vector** into a **target value**

$$f : \boldsymbol{x} \rightarrow y$$

$\{0, 1\}$	Binary classification
$\{0, 1, 2, \ldots, N\}$	Multi-class classification
$\mathbf{R}$	Regression

Example: Fisher's Iris Datasets



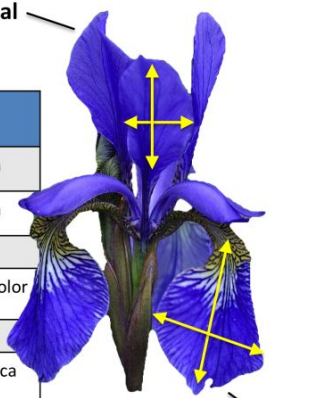
Iris Versicolor

Iris Setosa

Iris Virginica

Samples
(instances, observations)

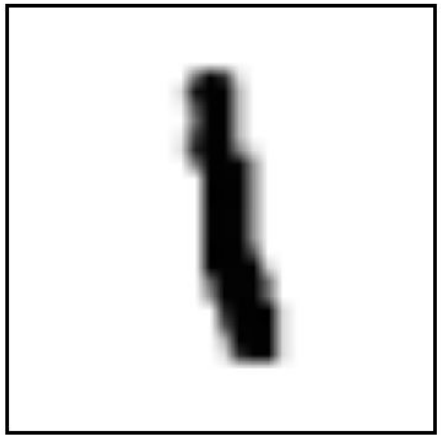
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Class labels
(targets)

Features
(attributes, measurements, dimensions)

Example: Handwritten Digit Recognition



$\approx$

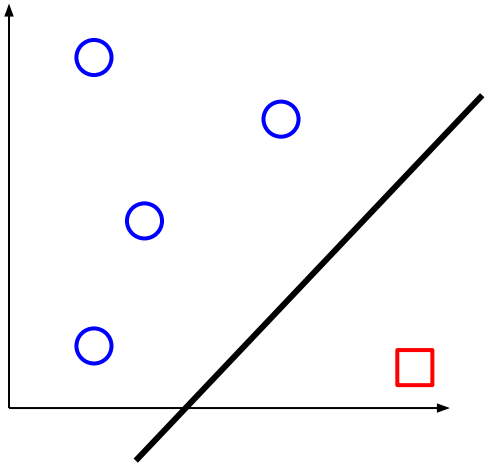
$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

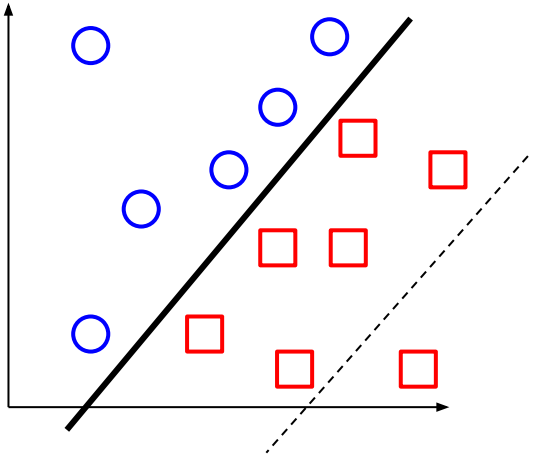
$\rightarrow$

x	y
	1

Positive/negative examples

- ML algorithms need both **positive/negative** examples to have a good model (especially, **borderline examples**)





Imbalanced dataset may need a special care (sampling or class-weights)

ML Problem Formulation

Different ML Problems

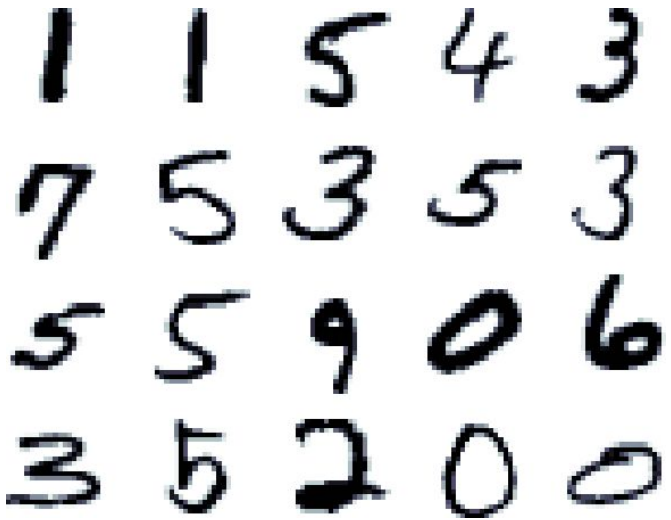
- Different target sets (often) need different classes of ML algorithms

$$f : \boldsymbol{x} \rightarrow y$$

$\{0, 1\}$	Binary classification
$\{0, 1, 2, \ldots, N\}$	Multi-class classification
$\mathbf{R}$	Regression

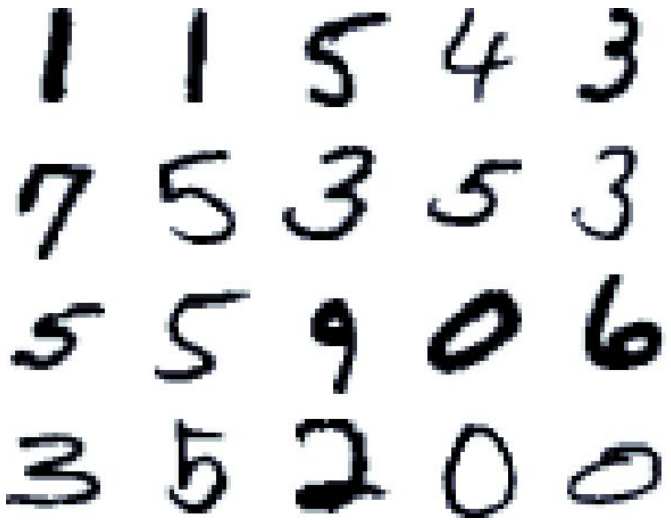
Quiz 1. Handwritten Digit Classification

- Input:
- Target:
- Task:



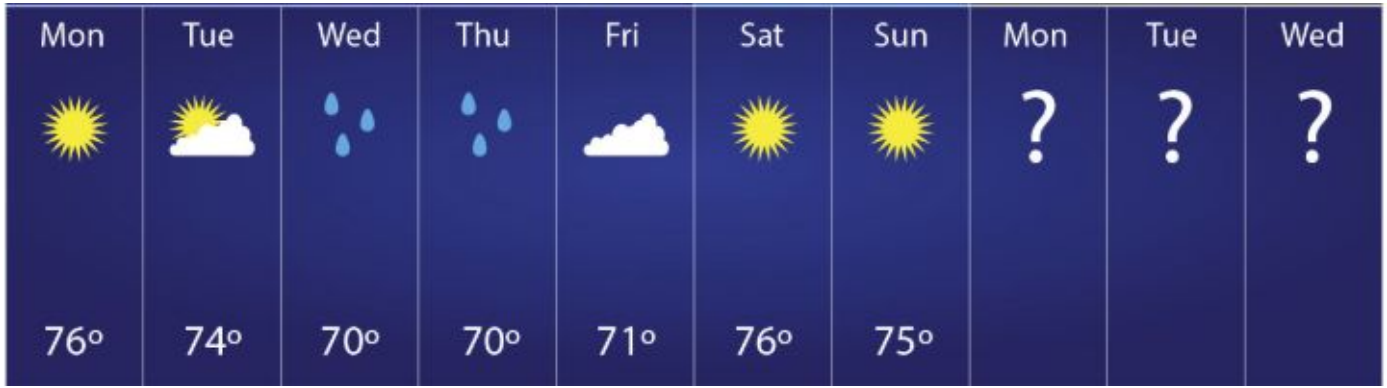
Quiz 1. Handwritten Digit Recognition

- Input: Grayscale pixel data
- Target: 10 digits (0-9)
- Task: Multi-class classification



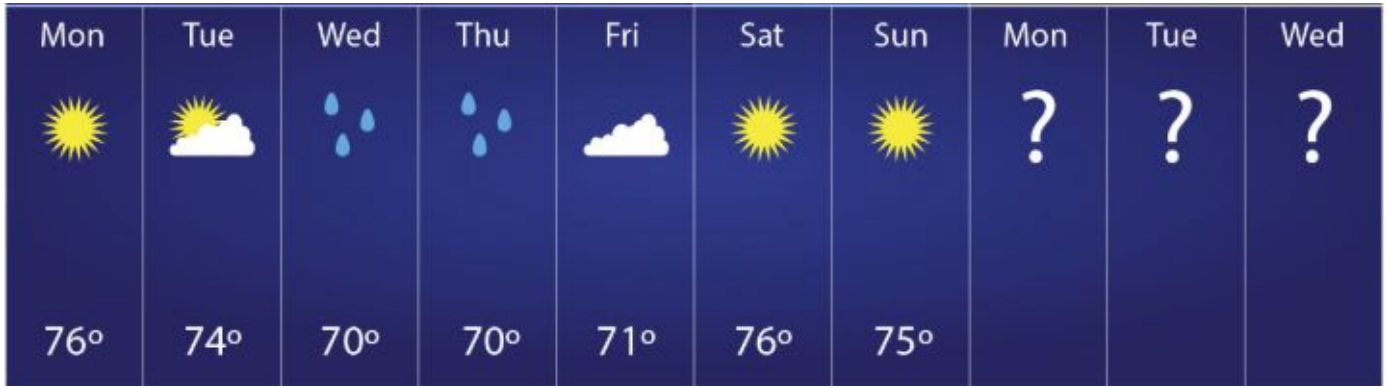
Quiz 2. Temperature Prediction

- Input:
- Target:
- Task:



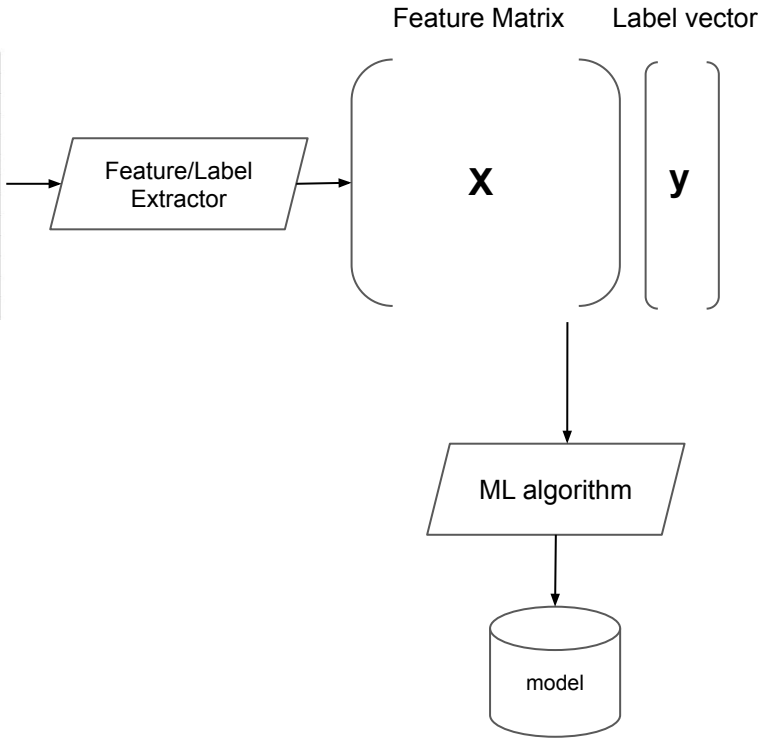
Quiz 2. Temperature Prediction

- Input: Temperature/Humidity/Weather in the past days
- Target: Temperature (of the next day)
- Task: Regression



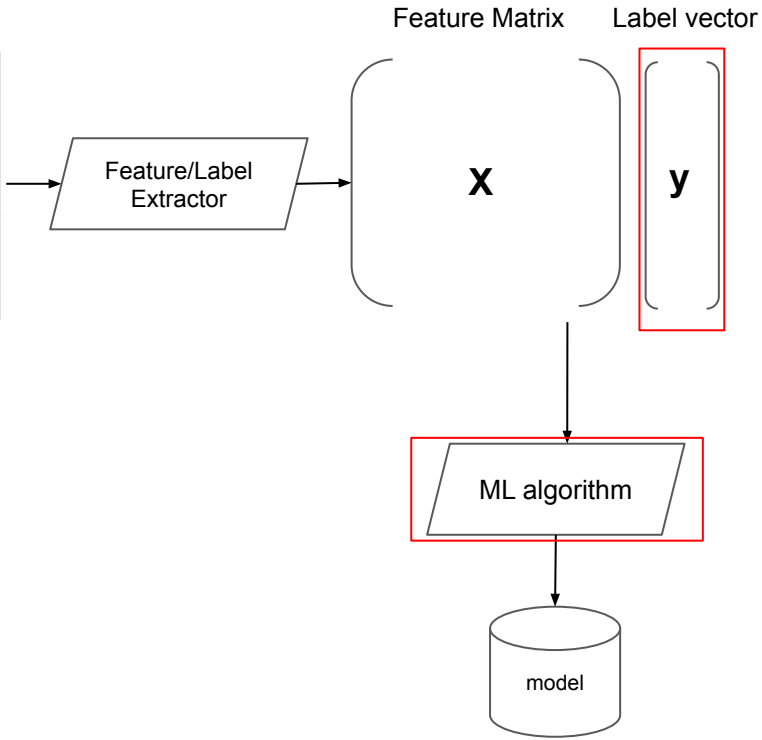
ML Workflow

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S



ML Workflow

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S

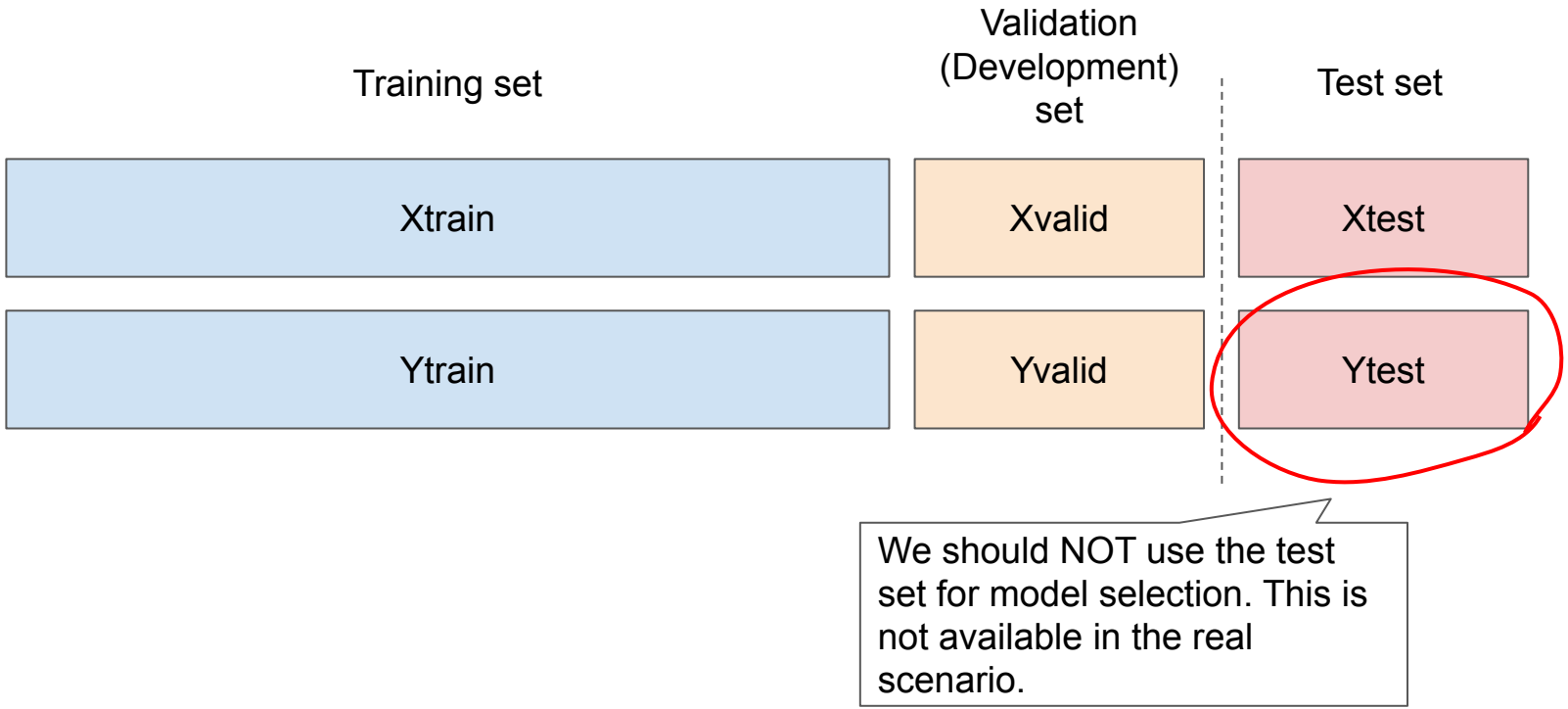


Key points

- 1) Feature matrix: **X**
- 2) Label vector: **y**
- 3) Task (e.g., classification, regression, etc.)

Training/Validation/Test Splits

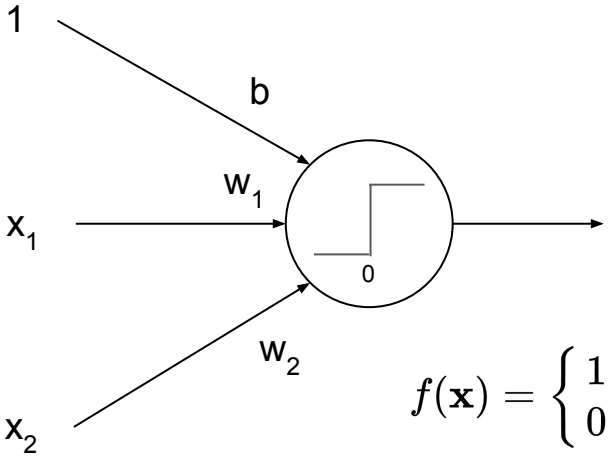
- We often split data into train/test to evaluate the generalizability performance



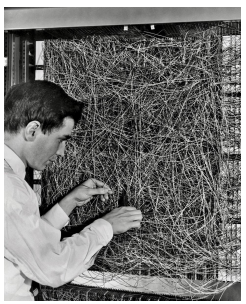
History of Neural Networks

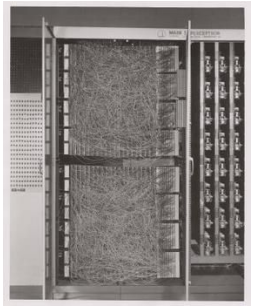
Perceptron (Rosenblatt 1958)

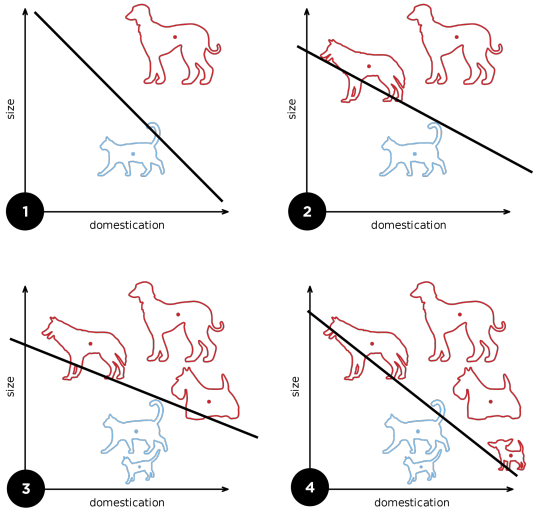
- A first solution to **“linearly separable”** problems



$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0, \\ 0 & \text{otherwise} \end{cases}$$



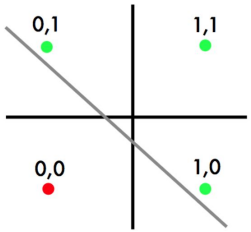




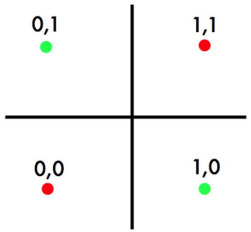
Cool!

“XOR Affair” (Minsky & Papert 1972)

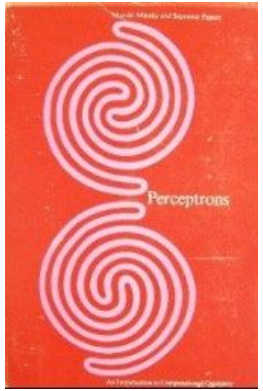
- Perceptrons cannot solve the problems that are not linearly separable



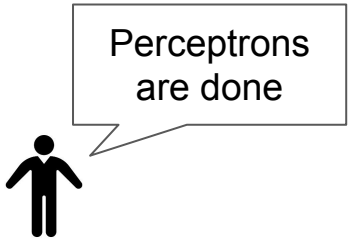
OR



XOR



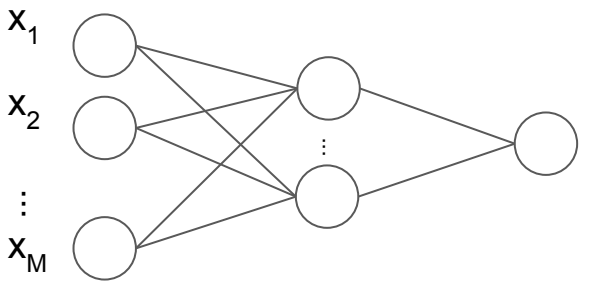
“Perceptrons” by Minsky & Papert (1972)



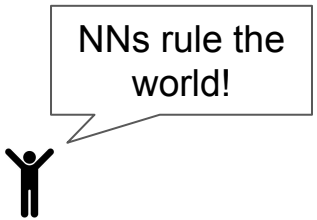
Backpropagation (1980s)
Rise of Multi-layer Perceptrons

- A new paradigm of Neural Networks
  - It is considered being invented in 1969 but recognized in 1980s.
- Universal approximation theorem (1989-)
  - Multi-layer Perceptrons (Neural networks) can approximate **any functions (theoretically)**

This is not enough. Why?



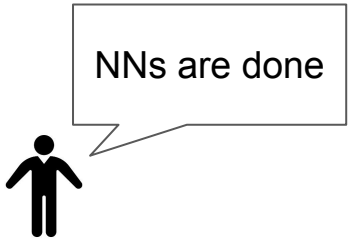
3-layer Perceptron/NN

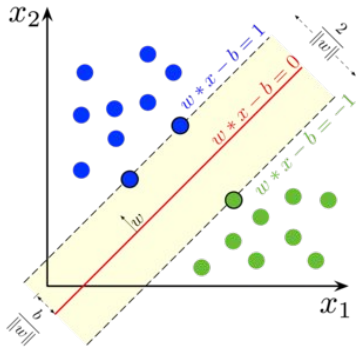


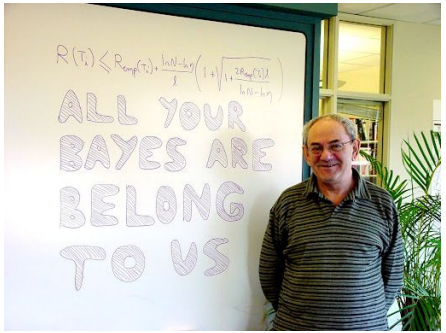
Another Winter Age for Neural Networks (late 90s-00s)

- Neural Networks were not that successful
  - N-layer NN (N > 3) was not better than N=3 (**Not Deep!**)
  - NNs did not perform robustly and training time was time-consuming
  - Optimization issues (e.g., local minima)

- + Attack of Support Vector Machines! (late 1990s)
  - The objective function is convex (i.e., global optima)



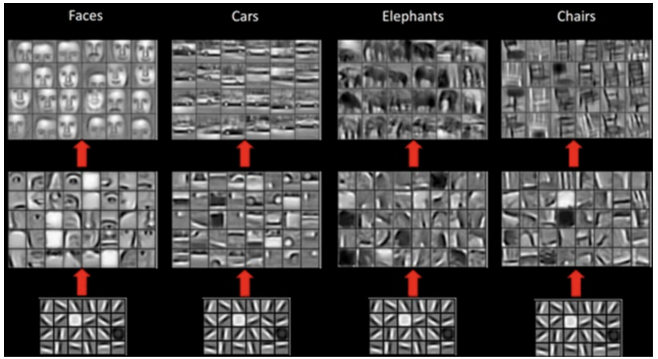


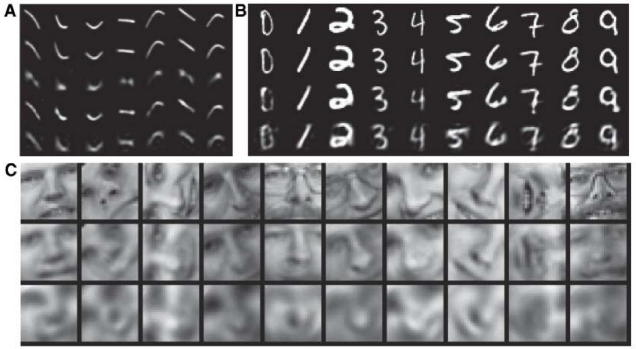


Return of Neural Networks: Deep Learning (2010s-)

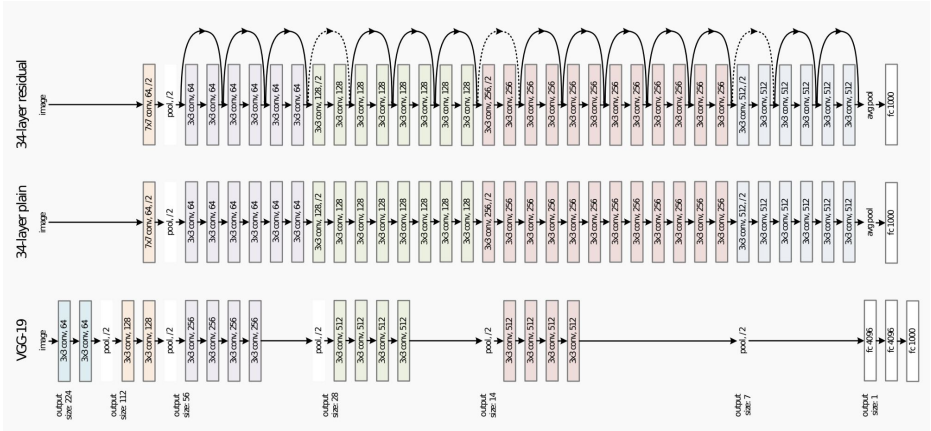
- Maturity of Neural Networks techniques → Representation Learning
  - Auto-Encoders (2006)
  - Advances in Convolutional Neural Networks (originally 1998)

- The age of Big Data
  - Computational resource ↑
  - Data size ↑



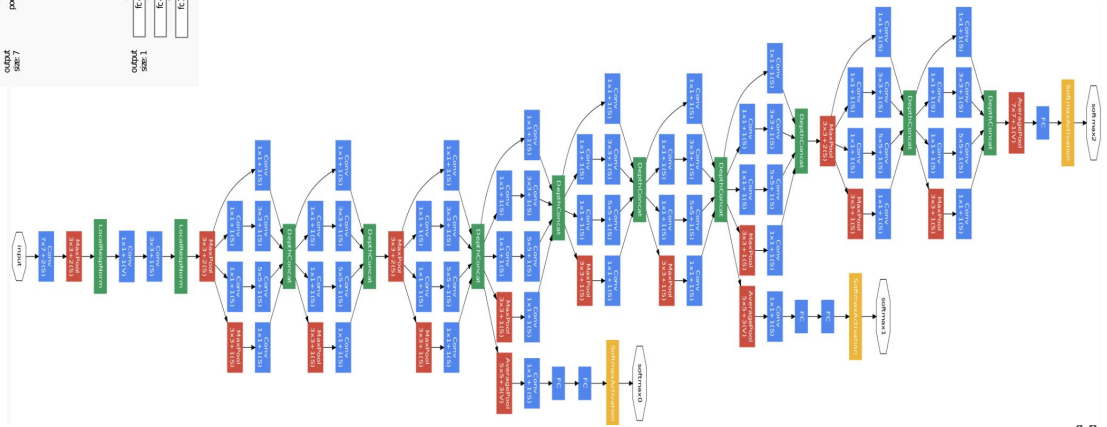


Today (1/2)



VGG-19 (2014)

Oh, that's cool. But, only for Computer Vision.



GoogLeNet (2014)

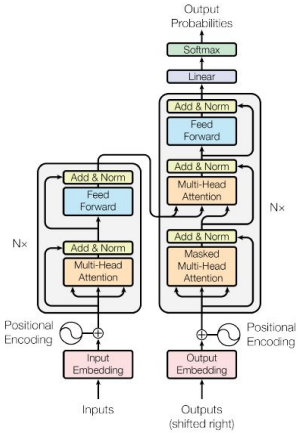
ImageNet competitions

Feature Engineering

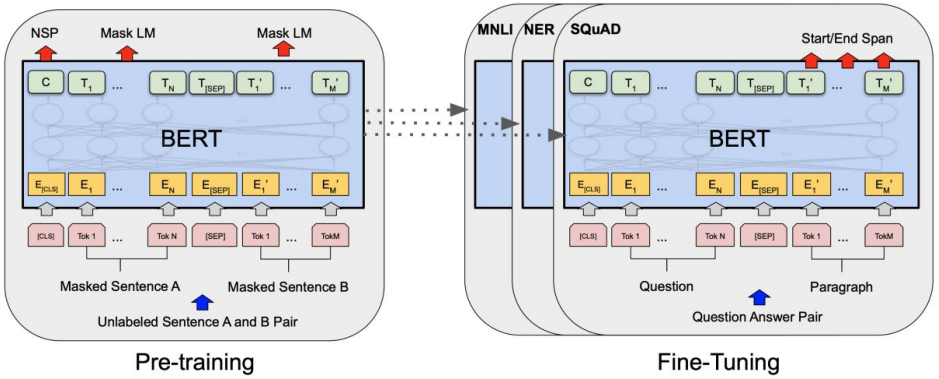
Convolutional Neural Networks

Today (2/2)

- Machine Learning =~Deep Learning/Neural Networks
- … for any domains



Transformers (2017)



BERT (2018)

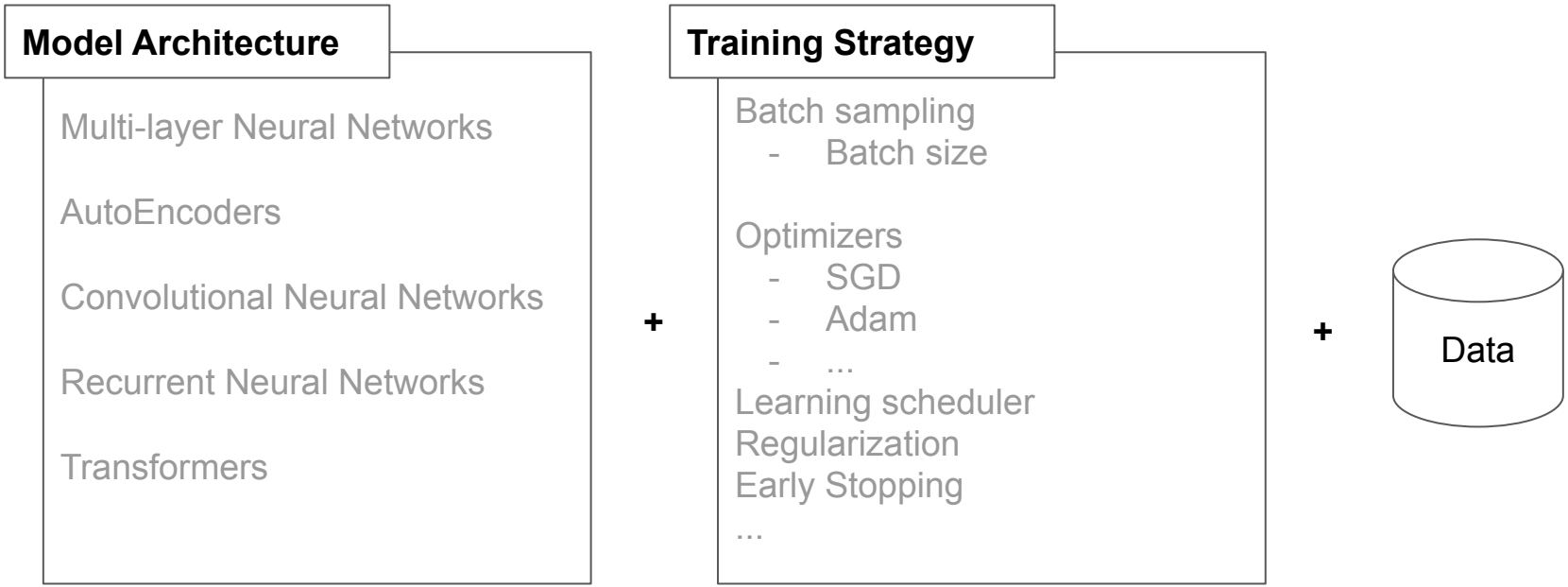
Neural Networks are all we need!

Neural Networks 101

Big Picture: Deep Learning as Building Blocks

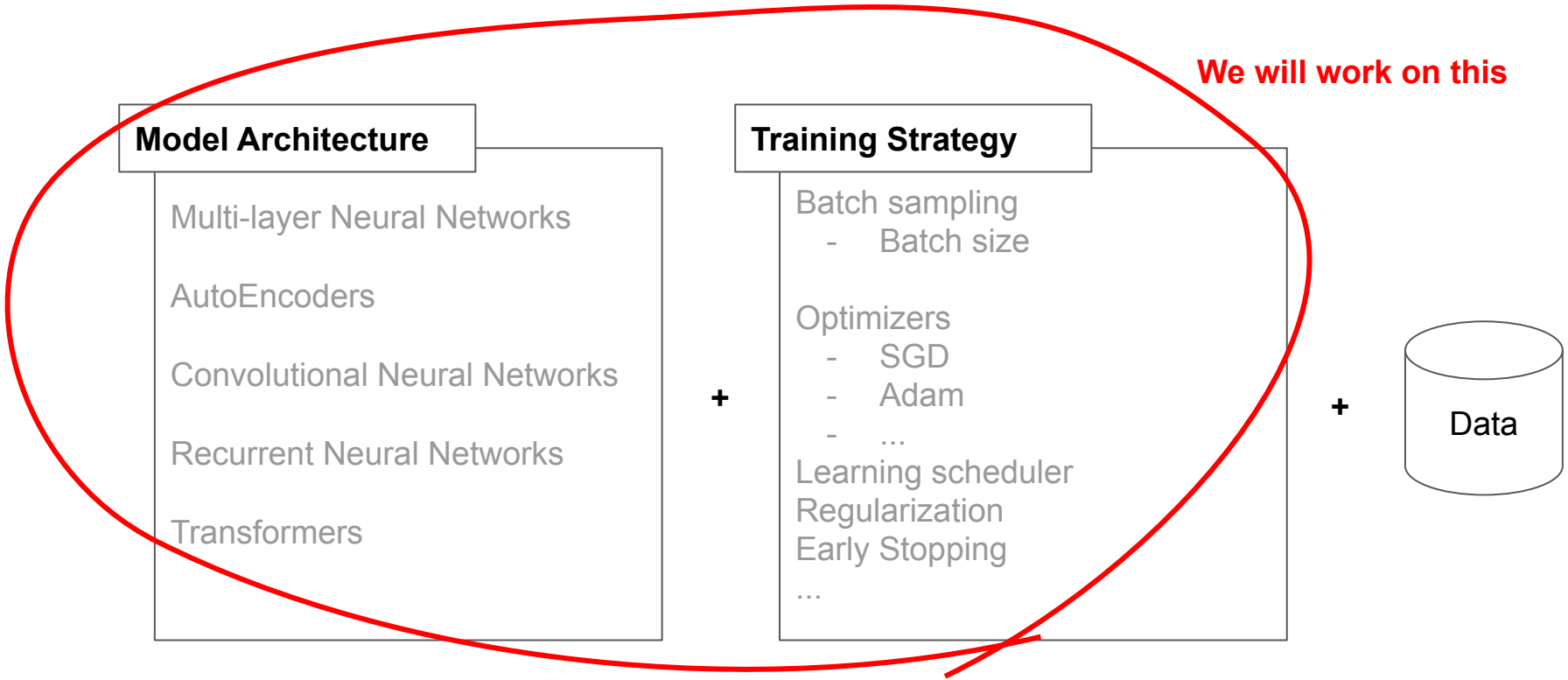
- **(A) Model Architecture + (B) Training Strategy + (c) Data**

What to optimize & How to optimize



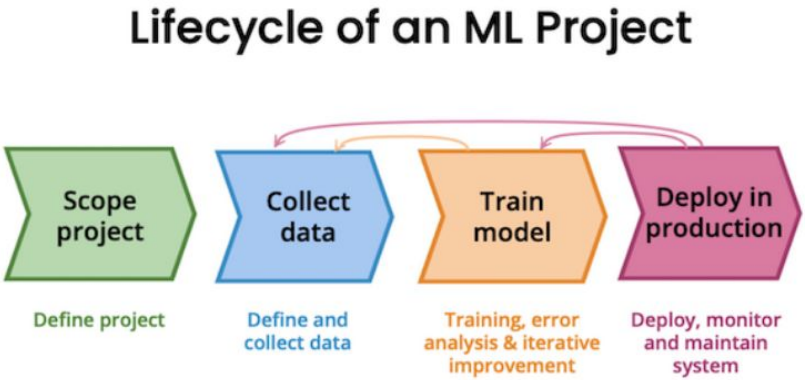
Big Picture: Deep Learning as Building Blocks

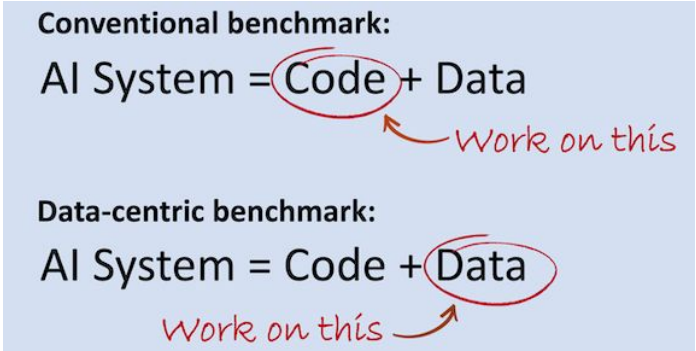
- **(A) Model Architecture + (B) Training Strategy + (c) Data**



Note: Model-centric AI to Data-centric AI

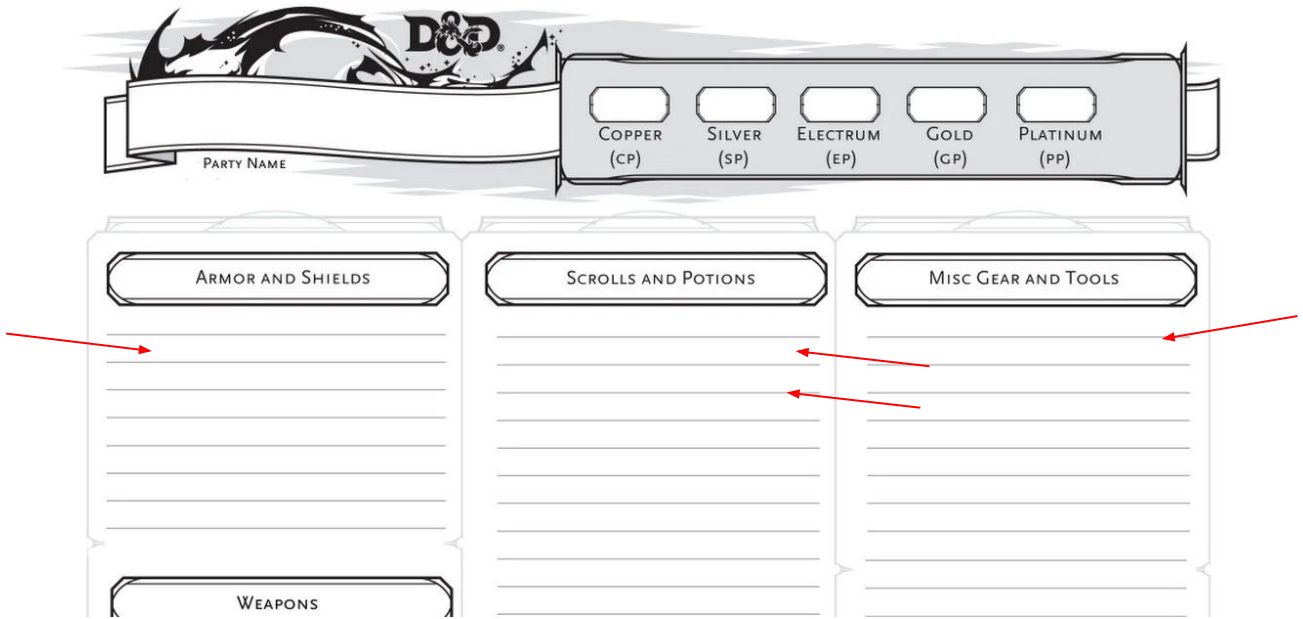
- The course focuses on the Model side, but it's important to keep in mind that **Data matter**



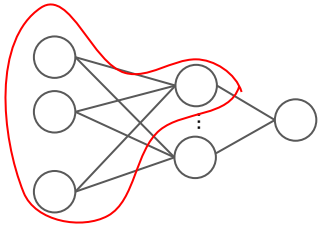


Get More Armors/Weapons/Magic Spells!!

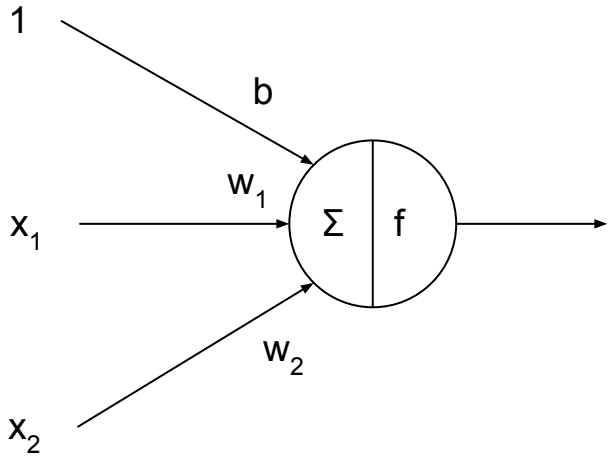
- But, what is more important is to learn how to beat monsters (i.e., problems) with them



Neuron: The minimal unit of Neural Networks



- Weighted sum + Activation function



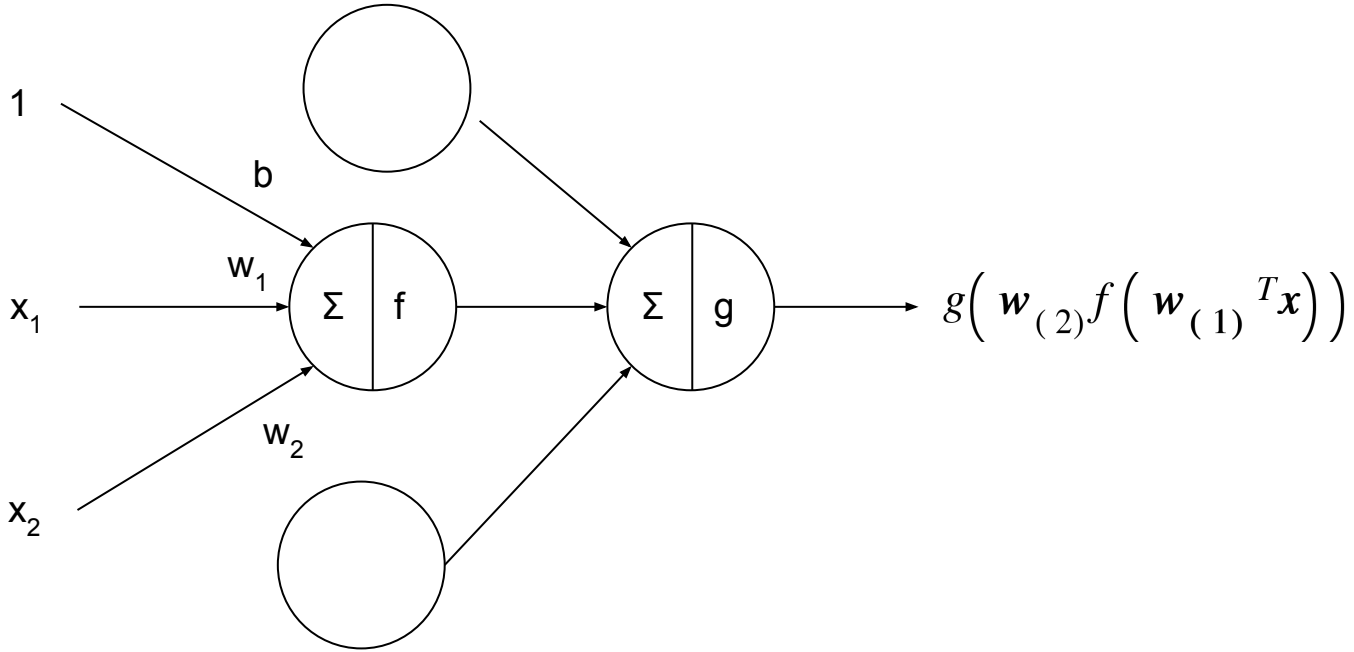
$$f\left(w_1 x_1 + w_2 x_2 + b \right)$$

general form: $f\left(\boldsymbol{w}^T \boldsymbol{x} \right)$

[x1, x2, 1]
(w1, w2, b)

Why Activation Function?

- Activation functions make **non-linear models**

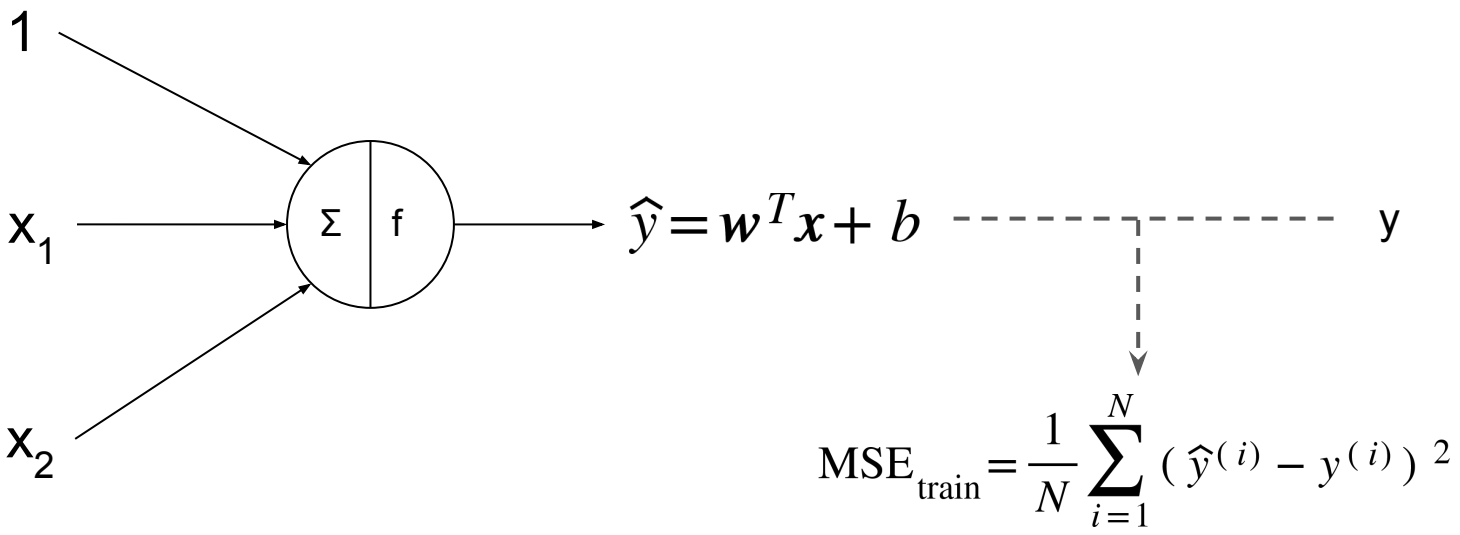


The simplest “Neural” models

- 1) Linear Regression
- 2) Logistic Regression

1) Linear regression (aka Ordinary Least Squares)

- The simplest neural network trained by Mean Squared Error (MSE)



Optimization for Linear Regression

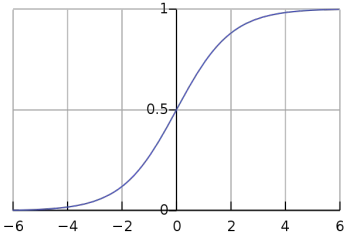
- Calculate gradient wrt **weight parameters**
- → Closed-form solution (known as normal equations) :)

$$\nabla_w \text{MSE}_{\text{train}} = 0$$

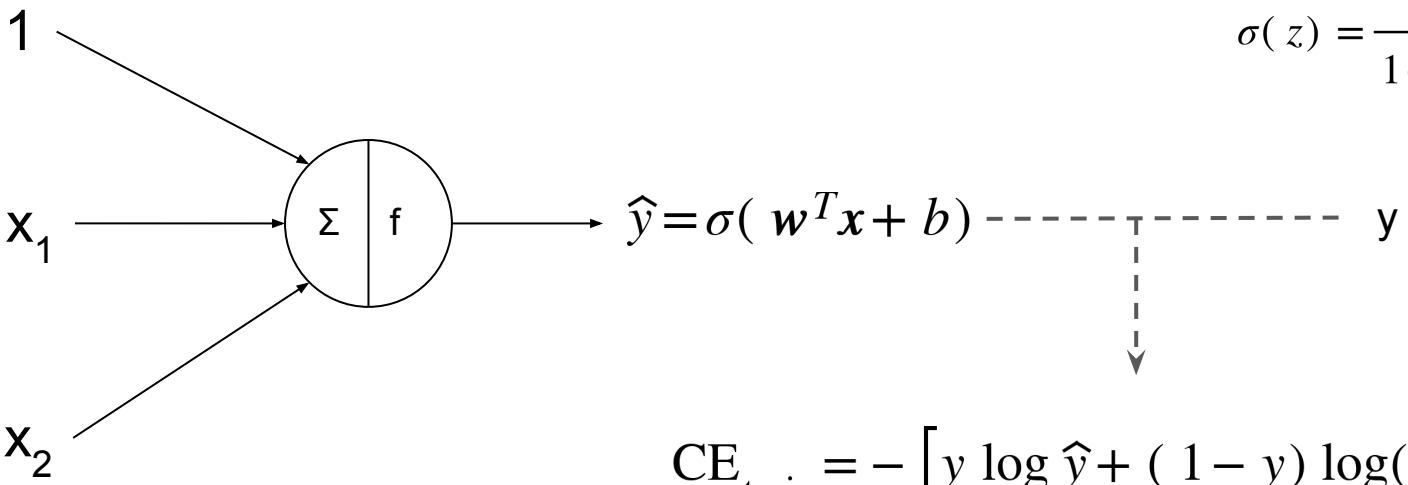
$$\boldsymbol{w} = (X^T X)^{-1} X^T Y$$

2) Logistic regression

- Logistic sigmoid to convert [-inf, inf] to [0, 1]



$$\sigma(z) = \frac{1}{1+e^{-z}}$$



$$CE_{train} = -\left[y \log \hat{y} + (1-y) \log(1-\hat{y}) \right]$$

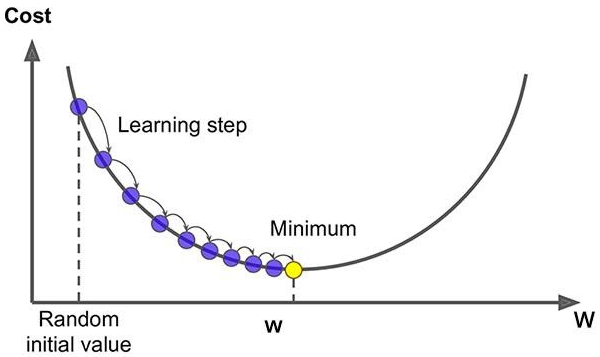
Optimization for Logistic Regression

- Calculate gradient wrt **weight parameters**
- No closed-form solution → Numerical optimization!

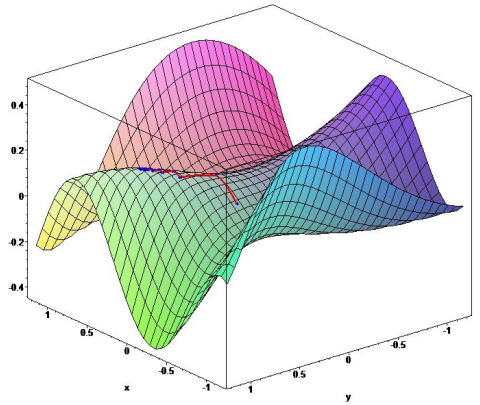
$$\nabla_{\boldsymbol{w}} \mathrm{CE}_{\text{train}} = (\hat{y} - y)\, x^{(i)}$$

Stochastic Gradient Descent

- Optimization method based on gradient that is calculated by (randomly chosen) samples



1d parameter space (Convex)



2d parameter space (Non-convex)

$$w^t \leftarrow w^{t-1} - \eta \nabla w$$

Optimization method = Gradient (direction) + Learning rate (Step size)

Note: Softmax Function & Multi-class Cross Entropy

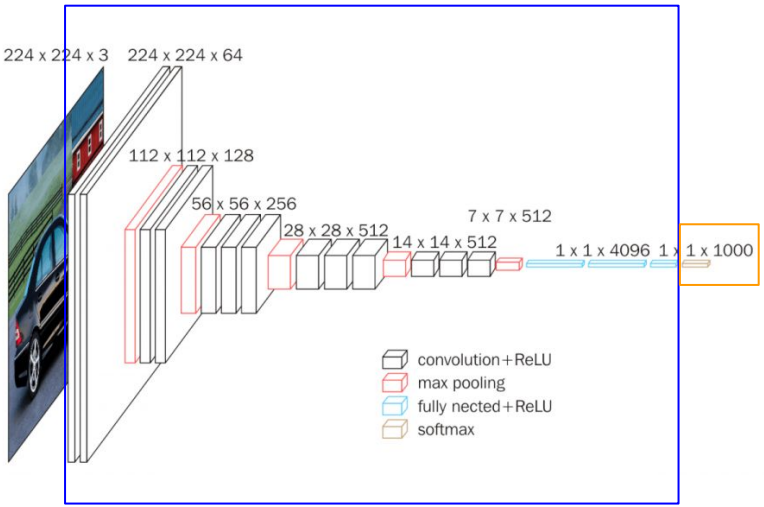
- General form of the logistic sigmoid function

$$\sigma(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

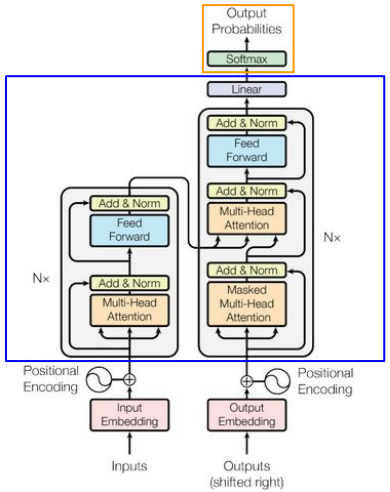
$$\text{MCE}_{\text{train}} = -\sum_k^K y_k \log \hat{y}_k$$

Neural Networks as Representation Learning

- Deep NN = **Representation extraction** + Linear/Logistic Regression



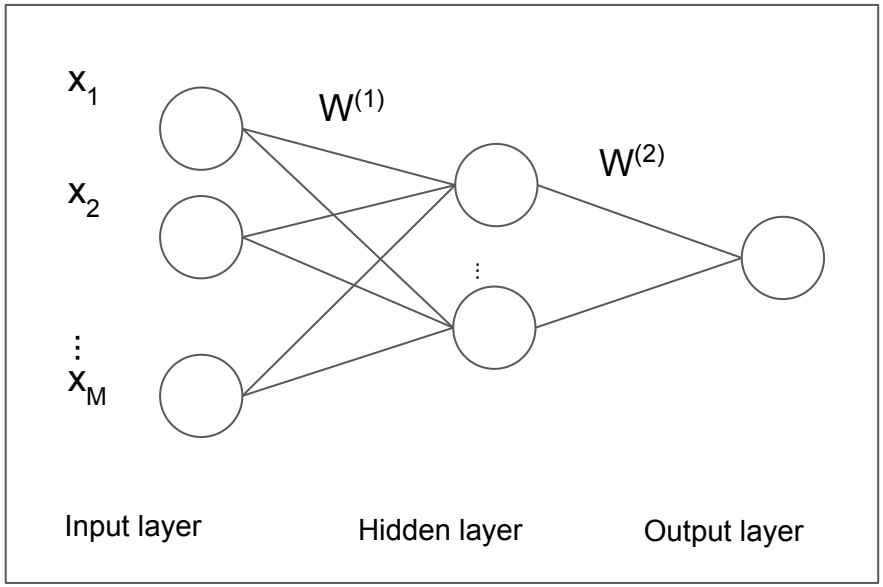
VGG 16



Transformer

Multi-layer Perceptron (MLP)

- Input layer + Hidden layer + Output layer



3-layer Perceptron/NN

$$h^{(1)} = g^{(1)}\left(W^{(1)T}\boldsymbol{x} + \boldsymbol{b}^{(1)} \right)$$

$$\hat{y} = g^{(2)}\left(W^{(2)T}h^{(1)} + \boldsymbol{b}^{(2)} \right)$$

$$\mathrm{MSE}_{\mathrm{train}} = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}^{(i)} - y^{(i)})^2$$

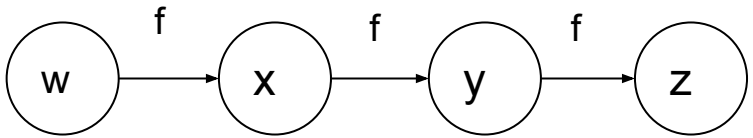
$$\nabla_{w^{(1)}} \mathrm{MSE}_{\mathrm{train}}$$

??

$$\nabla_{w^{(2)}} \mathrm{MSE}_{\mathrm{train}}$$

Optimization for MLP? (Credit Assignment Problem)

- How can we derive the gradient of each parameter?
- Simplified example: x = f(w), y = f(x), z = f(y)



- To compute $\frac{\partial z}{\partial w}$

$$\frac{\partial z}{\partial w} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w} \quad \textbf{(chain rule)}$$

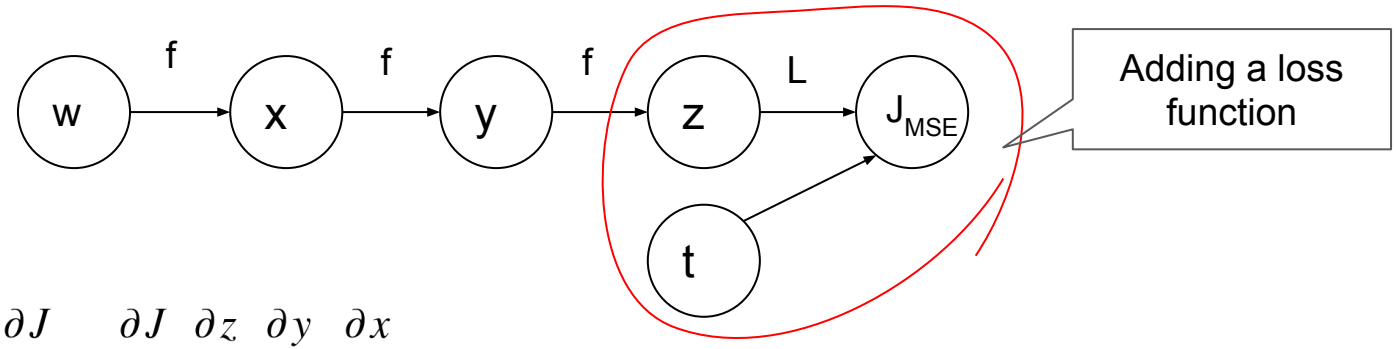
$$= f'(y) f'(x) f'(w)$$

$$= f'(f(f(w))) f'(f(w)) f'(w)$$

Those values can be stored in the forward propagation

Backpropagation from 3000 ft

- Multiplications of **the derivative** of each step + **Outputs** of the previous step



$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w}$$

$$= L'(z, t) f'(y) f'(x) f'(w)$$

$$= L'(f(f(f(w))), t) f'(f(f(w))) f'(f(w)) f'(w)$$

Differentiability is Key!

- The gradients of any parameters can be calculated **as long as the functions are differentiable!**

- Backpropagation = A gradient calculation method
  - i.e., can be coupled with any optimization method (e.g., SGD)

- We don’t have to implement the backpropagation step on our own (Yay!)

Summary

- Recap: Machine Learning Basics
- History of Neural Networks
- Neural Networks 101
  - Linear Regression & Logistic Regression
  - Multi-layer Perceptron (MLP)
  - How backpropagation works

10-minute break