Math 4570 -- Matrix Methods in Data Analysis and Machine Learning

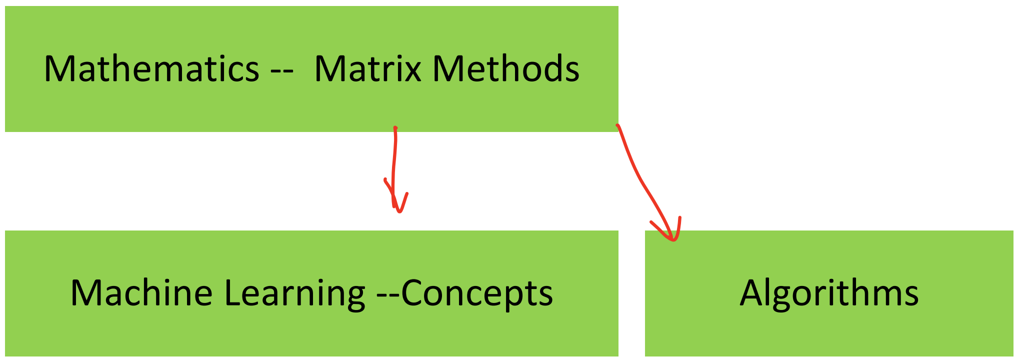
Section 0. Introduction

Instructor: He Wang
Department of Mathematics
Northeastern University

- **Prerequisite:**

1. Basic Linear Algebra
2. Multivariant calculus (partial derivatives)
3. Elementary Probability and Statistics
4. Basic programming skills

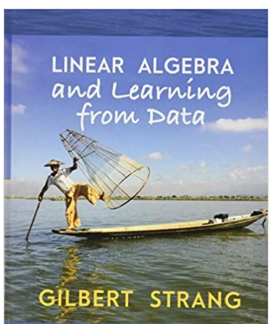
- **Goal of this class:**

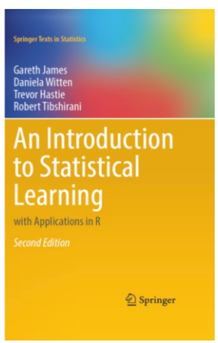


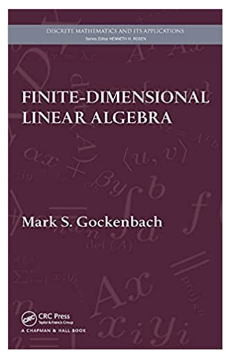
➢ A few recommended textbooks and online resources:

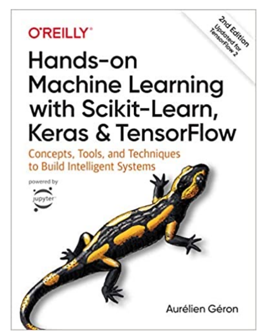
Textbook:

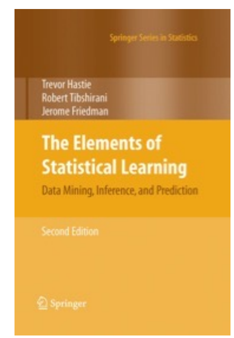
1. *Linear algebra and learning from data by Gilbert Strang*
2. *An Introduction to Statistical Learning by Gareth James Daniela Witten Trevor Hastie Rob Tibshirani*
3. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* by Trevor Hastie, Robert Tibshirani, Jerome Friedman https://web.stanford.edu/~hastie/ElemStatLearn/
4. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* - by Aurélien Géron. Code: https://github.com/ageron/handson-ml2
5. Finite-dimensional linear algebra, Mark S. Gockenbach, CRC Press.











More online sources:

Books/videos:

https://ocw.mit.edu/courses/mathematics/18-065-matrix-methods-in-data-analysis-signal-processing-and-machine-learning-spring-2018/index.htm

http://vmls-book.stanford.edu/

There are several linear algebra, machine learning open courses from Stanford/Cornell/ MIT/Carnegie Mellon, etc.

Tremendous lecture materials/code sources are on github or other online websites.

Too many sources!

I will add more extra notes and guideline on Canvas along with the class process.

- **Grades distribution:**

Homework (25%) - There will be 4 written assignments which will focus on theory.

Labs (25%) - There will be roughly 5 Labs will focus on the implementation of algorithms on real world data sets. Class time will be allotted for labs, but students may finish labs at home. In each lab, we will fit a real world data set using the algorithms of techniques introduced in that weeks' theory lecture.

Midterms (30%) Two midterms (each counts 15%).

Final Project (20%) - The final project will consist of a proposal (1 page), middle stage progress report(2-3 pages), project report (roughly 5 pages) and presentation (roughly 10-20 minutes with poster or slides). A project group should contain 4-7 students.

More on the **project**:

- This class features an XN project with an industry partner. Students are encouraged to participate in this project. (https://careers.northeastern.edu/experiential-network/)
- You can also choose any other interesting topics.
- Several data websites are available on Canvas.
- A theoretical presentation of a topic not covered in this course with a case study.
- Excellent project can consider to submit the poster to RISE (https://www.northeastern.edu/rise/about/)

I am more than happy to discuss possible projects in any of these categories with you.

Rough project timeline:

- Group Selection: **As early as possible.**
- Project Proposal Deadline: Before spring break
- Milestone progress Report (extended proposal): After we learned logistics regression.
- Draft paper and slides: One week before presentation
- Presentations: Last week's classes
- Final paper submission: May 1.

The project can be **any** topic relating to the class, however it can't be something you did it the end of the semester. It can't be a project you did in another course.

➢ Python Programing:

This course requires Python, Jupyter Notebook Server or Google Colab and github, all of which are free and open source.

After today's class:

1. Install **Python** from here: https://www.python.org/about/gettingstarted/

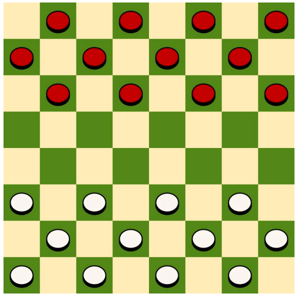
2. Install **Jupyter Notebook** from here: https://jupyter.org/

3. Try the tutorial examples using Jupyter-Notebook in the lab on Canvas.

4. Register a github account. https://github.com/

- Homework/Lab/Project Questions can be asked on Piazza.
- You should ask/answer a question with your name.
- When you sent me an email, please tell me which course are you in.
- Teaching Assistant: Hui Ying Man
- Office hours: See Canvas/Syllabus

➢ Definition of Machine Learning

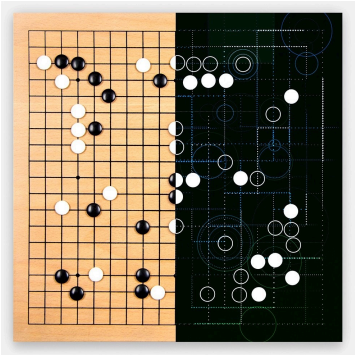


Arthur Samuel (1959): Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed.

(Samuel's Checkers Player Program.)

Tom Mitchell (1998): Machine Learning is a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

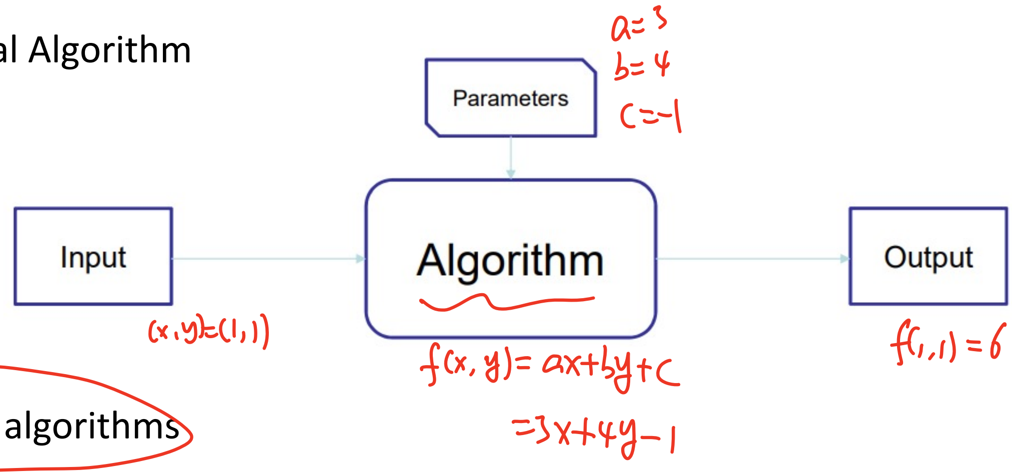
Experience E (data): games played by the program (with itself).
Task T: decisions the software need to make.
Performance measure P: winning rate.



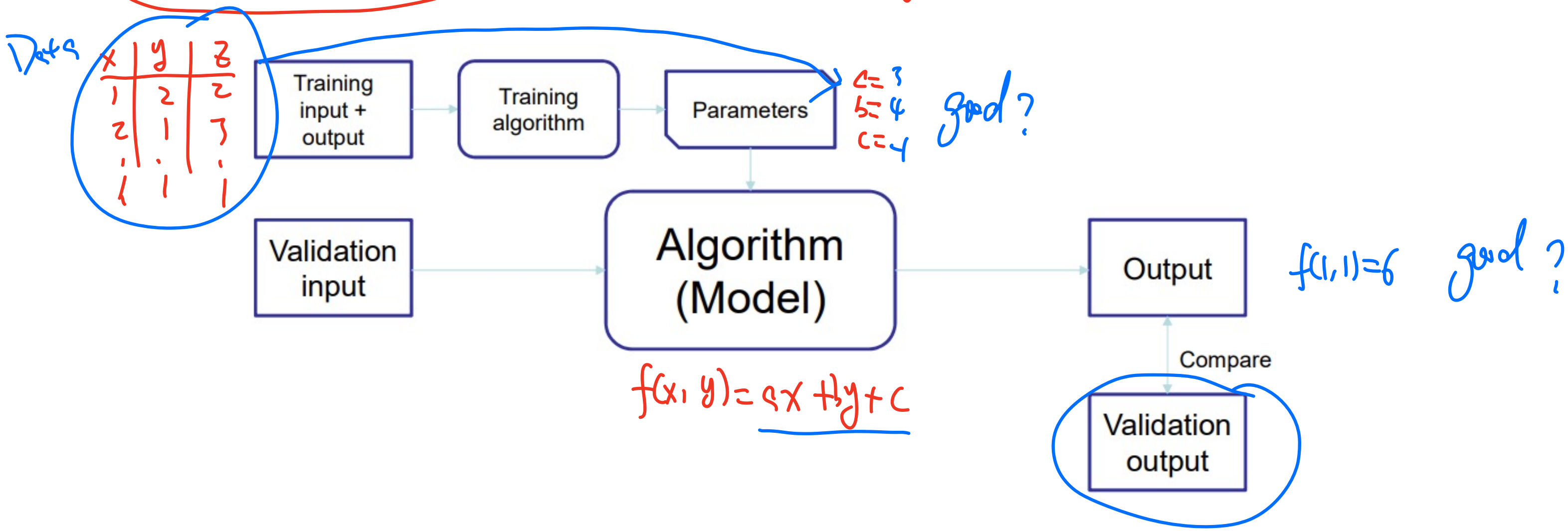
AlphaGo

Algorithms that are improved on some tasks with experiences.

Traditional Algorithm



Trainable algorithms



1. Supervised Learning:

Given a sample set of **labeled** data, can we predict the labels on new unlabeled data from the same domain?

- Regression Examples: predict house price, predict birth rate, etc.
- Classification Examples: Image classification, classification by features, Fraud detection, Email Spam Detection,etc.

2. Unsupervised Learning:

Given a set of **unlabeled** data, can we find structure within the data?

- Clustering Examples: Biology
- Dimensional reduction: face/image recognition, big data visualization.

3. Reinforcement Learning:

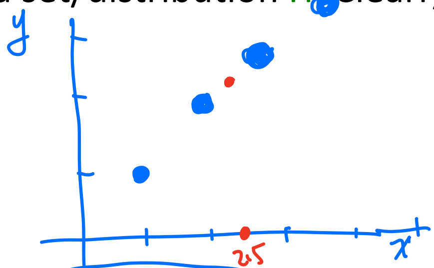
An agent that performs certain actions in an environment so as to maximize the reward.

- Examples: Gaming, Robot Navigation, etc.

No free lunch theorem.

The No Free Lunch Theorem states that every successful Machine Learning (ML) algorithm must make assumptions. This also means that there is no single ML algorithm that works for every setting.

Every ML algorithm has to make assumptions on which hypothesis class H should you choose? This choice depends on the data, and encodes your assumptions about the data set/distribution H. Clearly, there's no one perfect H for all problems.



- Example:

Assume that $(\vec{x}_1, y_1) = (1, 1)$, $(\vec{x}_2, y_2) = (2, 2)$, $(\vec{x}_3, y_3) = (3, 3)$, $(\vec{x}_4, y_4) = (4, 4)$, $(\vec{x}_5, y_5) = (5, 5)$.

Question: what is the value of y if $\vec{x} = 2.5$?

Answer: y=2.5 — Assumptions: y = x
or y=0 — y = integers
y undefined

1. Supervised Learning

Data is pre-categorized or numerical.

- Training Data Set : $\mathcal{D} = \{(\vec{x}^{(i)}, y^{(i)}) \mid i = 1 \dots n\} \subset \mathbb{R}^d \times \mathcal{C}$
- Label Space: $\mathcal{C} = \mathbb{R}$, or {0,1}, or {-1, 1}, or {1,2,3,...}

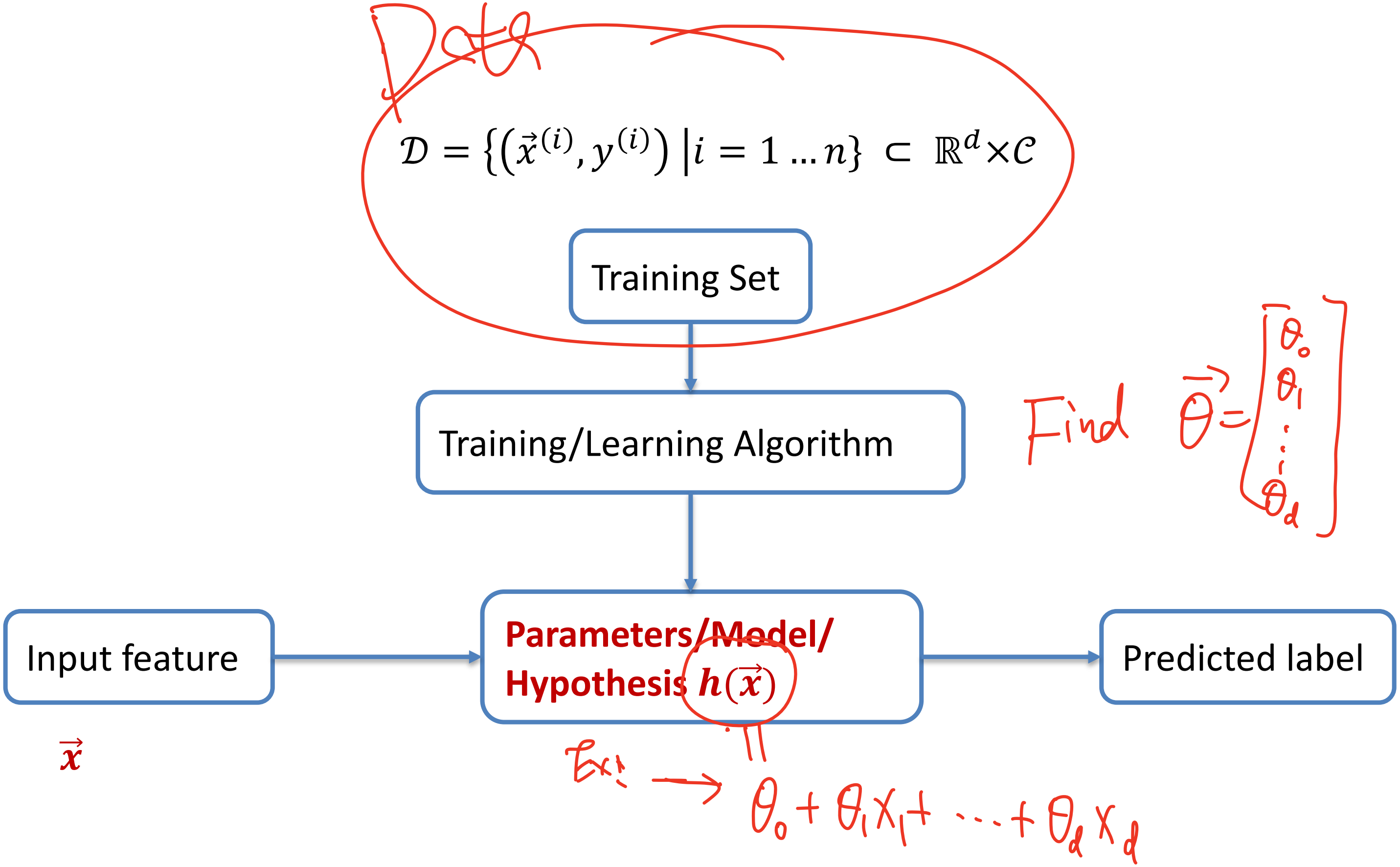
The goal in supervised learning is to **make *predictions from data*.**

- **Regression:** Predict a number. Predict house price, future temperature or the height of a person, etc.

Quantitative Variable $y \in \mathbb{R}$: Variables that take quantitative values, with some values larger than others and close values designating similar characteristics. Also called numerical.

- **Classification:** Predict a category. (Spam or not spam, dog or cat or ,)

Qualitative Variable $y \in \{0,1\}$, or $\{-1, 1\}$, or $\{1,2,3,\ldots\}$: Variables that only take discrete values, usually in a set of descriptive classes. Also called categorical or discrete variables, or factors. There could be no additional ordering or structure on the classes.

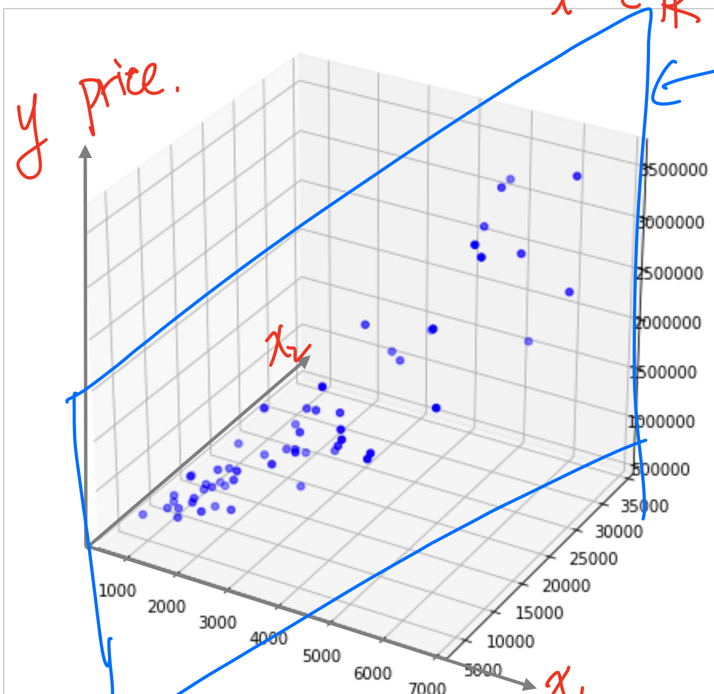


Data
$\mathcal{D} = \{(\vec{x}^{(i)}, y^{(i)}) \mid i = 1 \dots n\} \subset \mathbb{R}^d \times \mathcal{C}$
Training Set
Training/Learning Algorithm
Find $\vec{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$
Input feature
$\vec{x}$
Parameters/Model/
Hypothesis $h(\vec{x})$
Predicted label
Ex: $\rightarrow \theta_0 + \theta_1 x_1 + \dots + \theta_d x_d$

Input: a dataset that contains n samples $(\vec{x}^{(i)}, y^{(i)})$ $i=1, \dots, n$

Task: if a house has x_1 feet^2 living size and x_2 feet^2 lot size, predict its price?

$\vec{x}^{(i)} \in \mathbb{R}^2$



Assume $h_{\vec{\theta}}(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

Find $\vec{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$ from Data.

$\vec{x} = \begin{bmatrix} 2500 \\ 10000 \end{bmatrix}$

Regression:

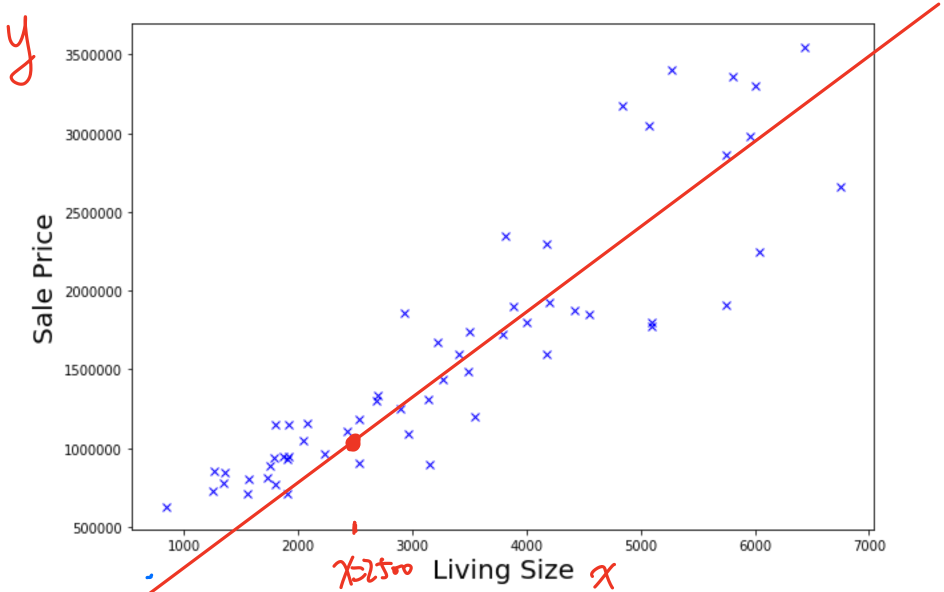
- Data Set: $\mathcal{D} = \{(\vec{x}^{(i)}, y^{(i)}) \,|\, i = 1 \dots n\} \subset \mathbb{R}^d \times \mathcal{C}$
- Label space $\mathcal{C} \subset \mathbb{R}$

Goal: Find a model $h(\vec{x})$ from the data.

Example: Predict house price.

Input: a dataset that contains n samples with d=1.

Task: if a house has x square feet, predict its price?



- Predict house price.

x_1 x_2 x_3 x_4 x_5 x_5 y

BEDS	BATHS	LOCATION	SQUARE_FEET	LOT_SIZE	YEAR_BUILT	PRICE
3	3	Newton	2969	15014	1967	1090000
3	2.5	Newton	1566	5582	1922	805000
4	2.5	Newton Corner	2532	6273	1953	905000
7	4.5	Newton Center	6748	26607	1902	2660000
4	4	West Newton	4200	20446	2007	1925000
4	2.5	Newton	2232	3966	1870	965000
2	1.5	Newton Corner	1344	5559	1851	775000
3	2.5	Newton	2898	12420	1943	1250000
2	2	West Newton	1729	4171	1953	815000
6	3	West Newton	3149	12616	1953	900000
5	3.5	West Newton	4000	12006	1912	1800000
4	3.5	West Newton	6430	30600	1920	3550000
4	1.5	Auburndale	1750	8222	1893	885000
2	2	Newton	840	5548	1955	630000
…	….	…	…	…	…	…

Input: a dataset that contains n samples $\mathcal{D} = \{(\vec{x}^{(i)}, y^{(i)}) \,|\, i = 1 \dots n\} \subset \mathbb{R}^6 \times \mathcal{C}$.

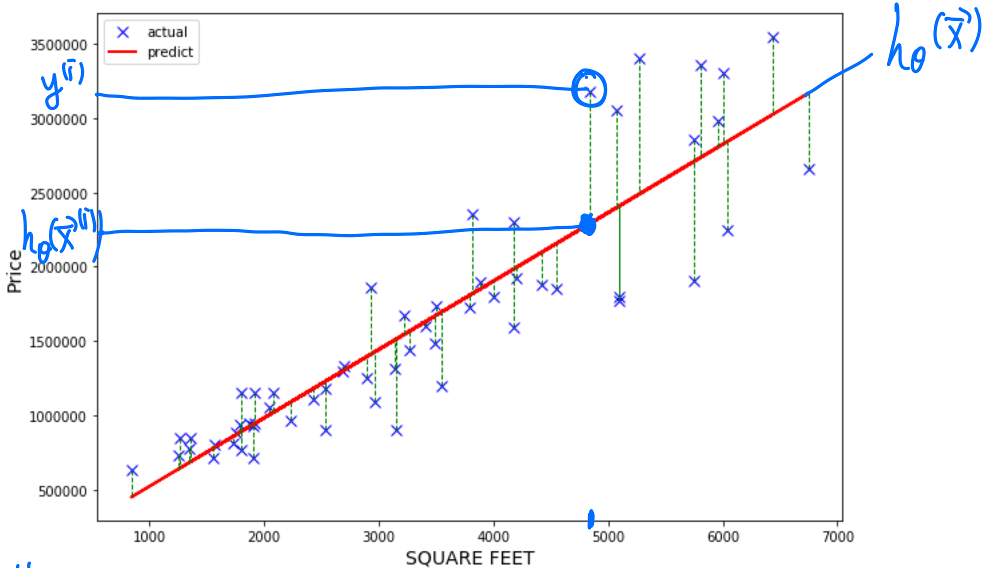
Task: predict its price, if a house has $\vec{x}$

Assume $h_\theta = \theta_0 + \theta_1 x_1 + \cdots + \theta_6 x_6$

Assume $h_\theta = \theta_0 + \theta_1 x_1^2 + \theta_2 x_1^3 + \cdots$

Assume $h_\theta = \theta_0 + \theta_1 e^{x_1} + \ldots$

➢ Evaluation: Cost/Loss Functions

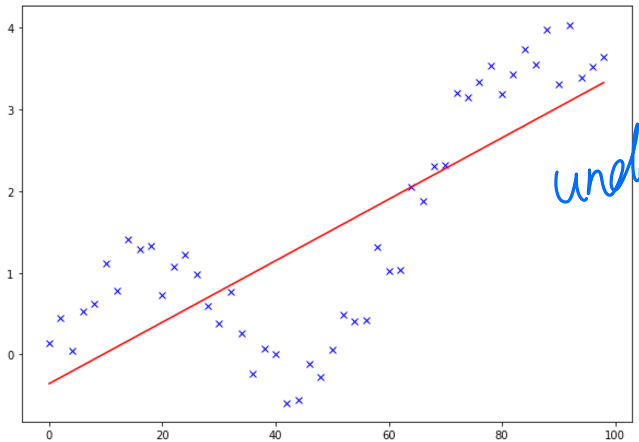


"example"

$$L(\vec{\theta}) := \sum_{i=1}^{n} \left(y^{(i)} - h_\theta(\vec{x})\right)^2$$

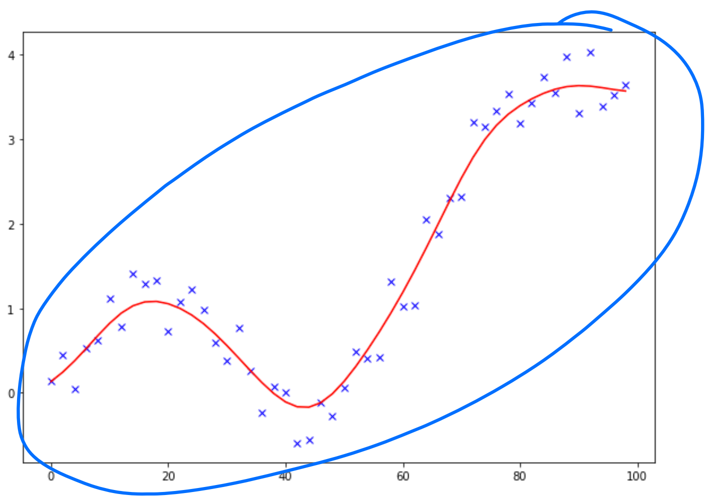
Underfitting v.s. Overfitting

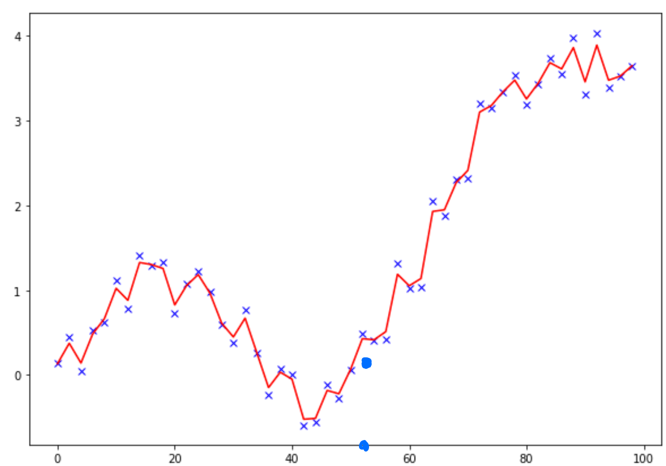
Bias-Variance Tradeoff.



underfitting

overfitting





❖ Supervised Classification.

- Training Data Set : $\mathcal{D} = \{(\vec{x}^{(i)}, y^{(i)}) \mid i = 1 \ldots n\} \subset \mathbb{R}^d \times \mathcal{C}$
- Label Space: $\mathcal{C} = \mathbb{R}$, or {0,1}, or {-1, 1}, or {1,2,3,...}

$\vec{x} = \begin{bmatrix} 8 \\ 8 \end{bmatrix}$

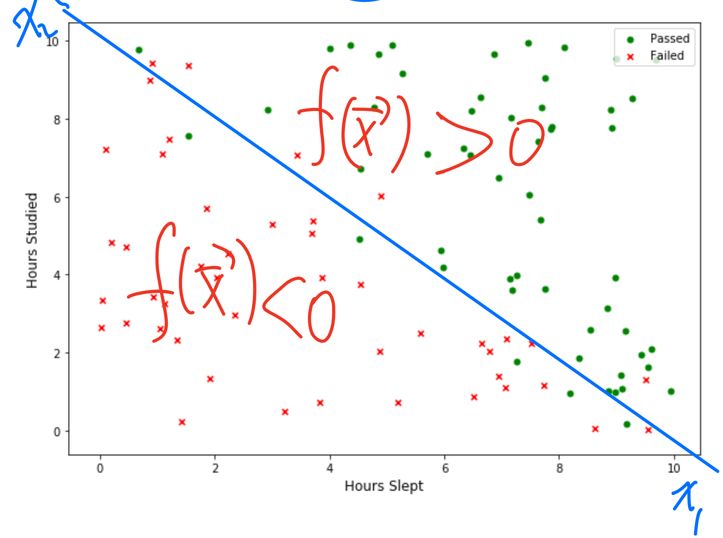
Logistic Regression Method:

$f_\theta(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

$d = 2$

Assume

$$P(y=1 \mid \vec{x}) = \frac{1}{1 + e^{-f_\theta(\vec{x})}}$$



Binary classification:

Spam filtering. Here, an email (the data instance) needs to be classified as *spam* or *not-spam*.

Dear Good Friend

I am Abdoul Issouf, I work for BOA bank Ouagadougou Burkina Faso. I have a business proposal which concerns the transfer of ($13.5 Million US Dollars) into a foreign account. Everything about this transaction shall be legally done without any problem. If you are interested to help me, Please keep this transaction as a Top Secret to your self till the Money get into your account in your Country OK. and I will give you more details as soon as I receive your positive response. You will be Entitled to 50%, 50% will be for Me If you are willing to work with me send me immediately the information listed bellow.

Your Name ....

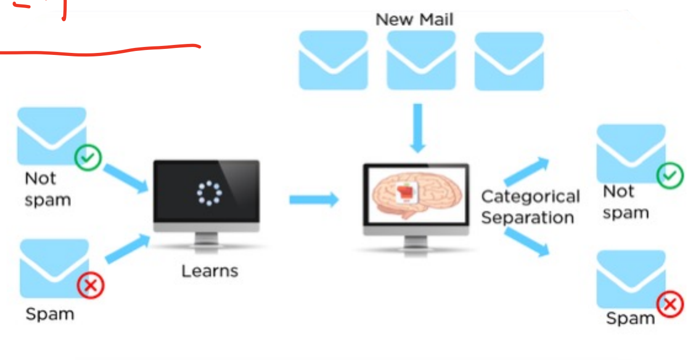
Your Nationality ....

Your Age ....

Female Or Male ....

Your Occupation ....

Your Private Telephone.............................

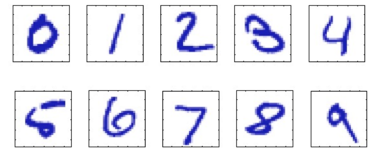


We will represent an email via a feature vector, whose length *d* is equal to the number of words in the dictionary.

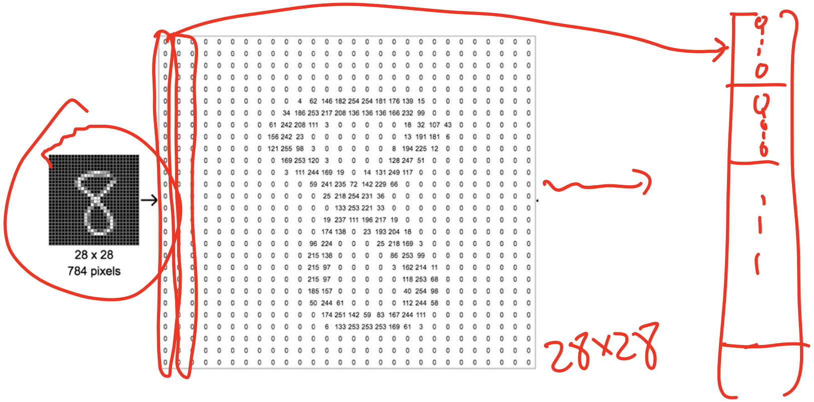
$$\vec{x} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- Multi classes classification. Examples of hand-written digits taken from US zip codes.

The MNIST (Modified National Institute of Standards and Technology) data set of handwritten numbers. It contains 60,000 training images and 10,000 testing images.

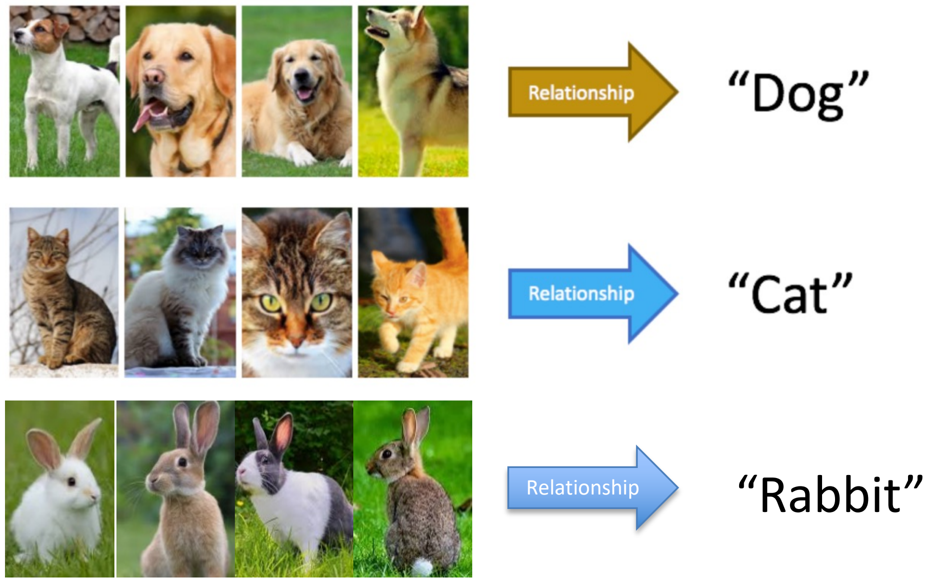


The black and white images from MNIST were normalized to fit into a 28x28 pixels.

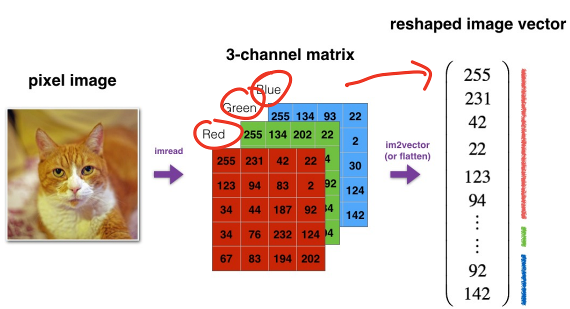


$= \vec{x} \in R^{28x28}$

- Multi-class classification: Image Classification



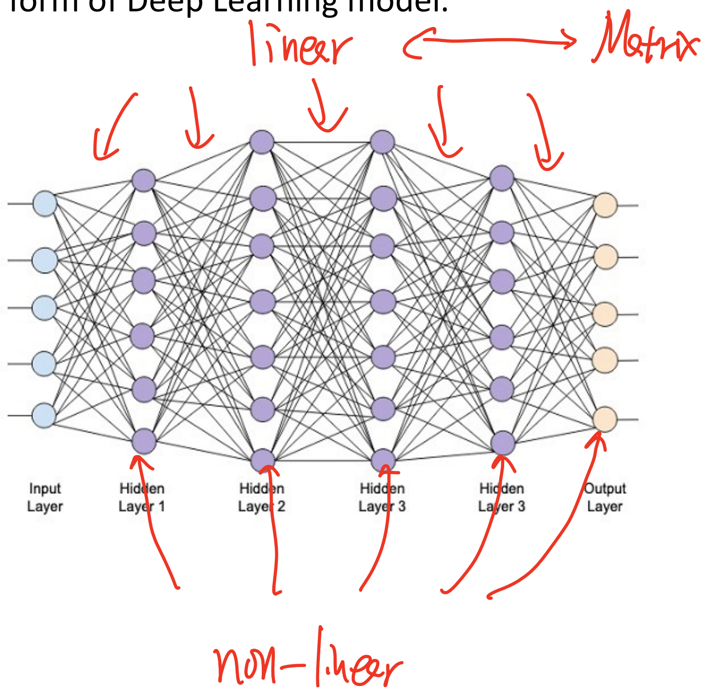
The features represent pixel values.



➢ Artificial Neural Network (ANN or NN)

Human Neural Networks was introduced in 1943 by neurophysiologist Warren McCulloch and mathematician Walter Pitts to model neurons in the brain using electrical circuits.

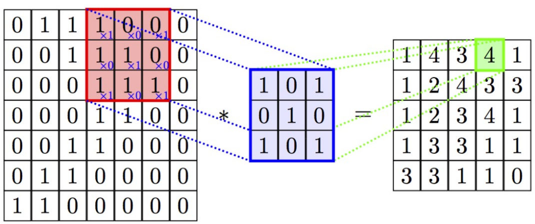
Artificial Neural networks are a series of algorithms that mimic the operations of a human brain to recognize relationships between vast amounts of data. it's a very broad term that encompasses any form of Deep Learning model.

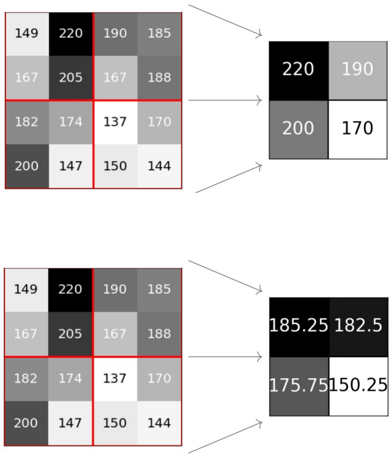


$$f(z) = \frac{1}{1+e^{-z}}$$

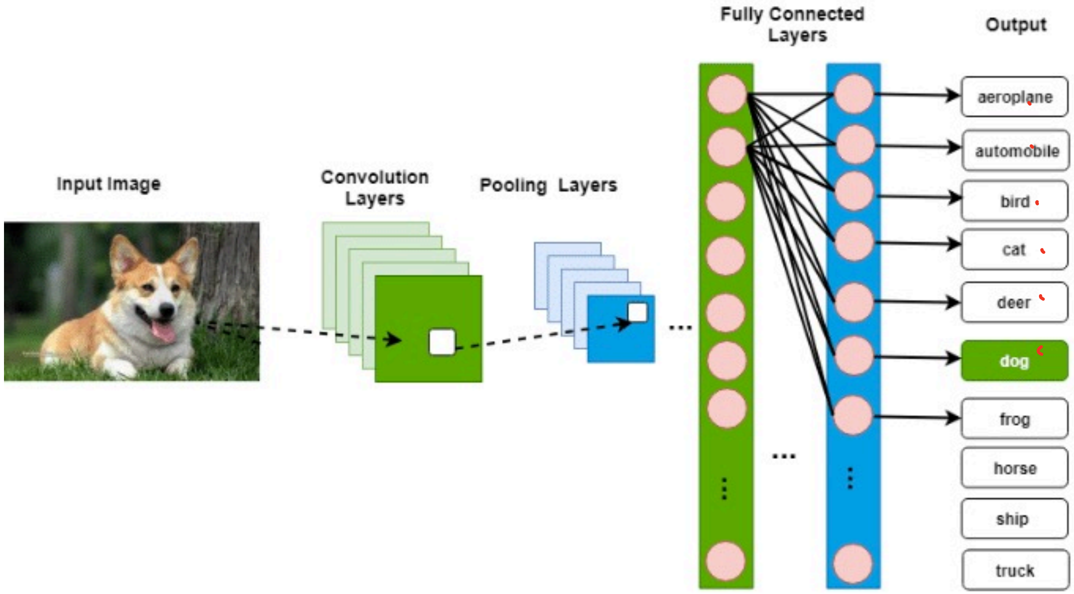
➢ Convolution Neural Networks (CNN)

CNNs are a specific type of neural networks that are generally composed of convolution layers and pooling layers.



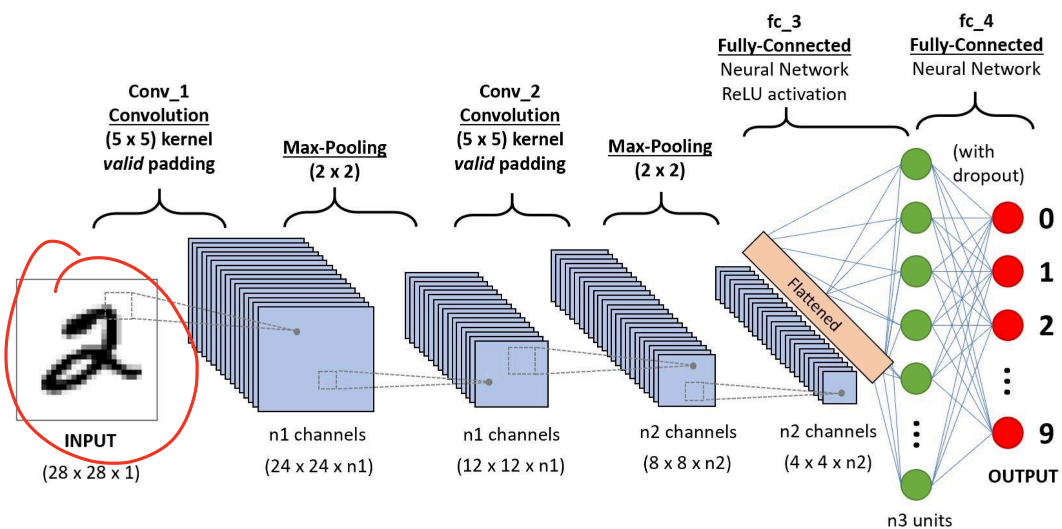


An Interactive Node-Link Visualization of Convolutional Neural Networks
https://www.cs.ryerson.ca/~aharley/vis/conv/

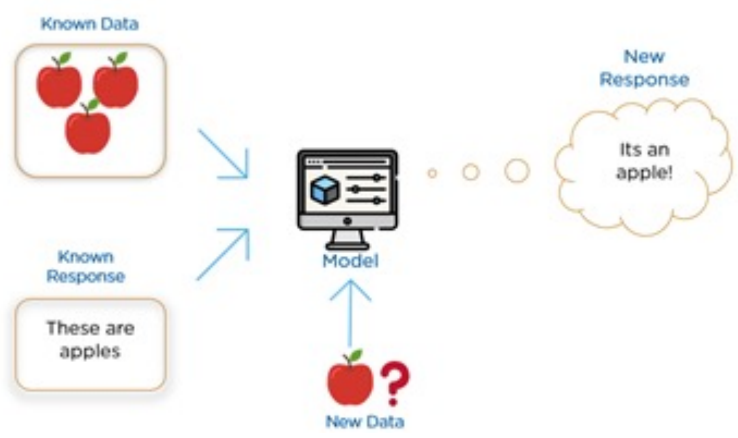


P(y=1 | pic)

P(y=6 | pic)
=
0.8

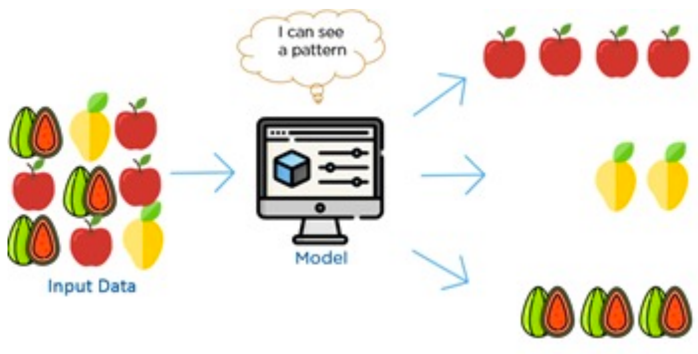


- Supervised Learning



Labeled data are expensive!

- Unsupervised Learning



Data are cheap!

Datasets collections from Tensorflow:

https://www.tensorflow.org/datasets/catalog/overview

More collections on Canvas.

Famous datasets:

MNIST: http://yann.lecun.com/exdb/mnist/

MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples.

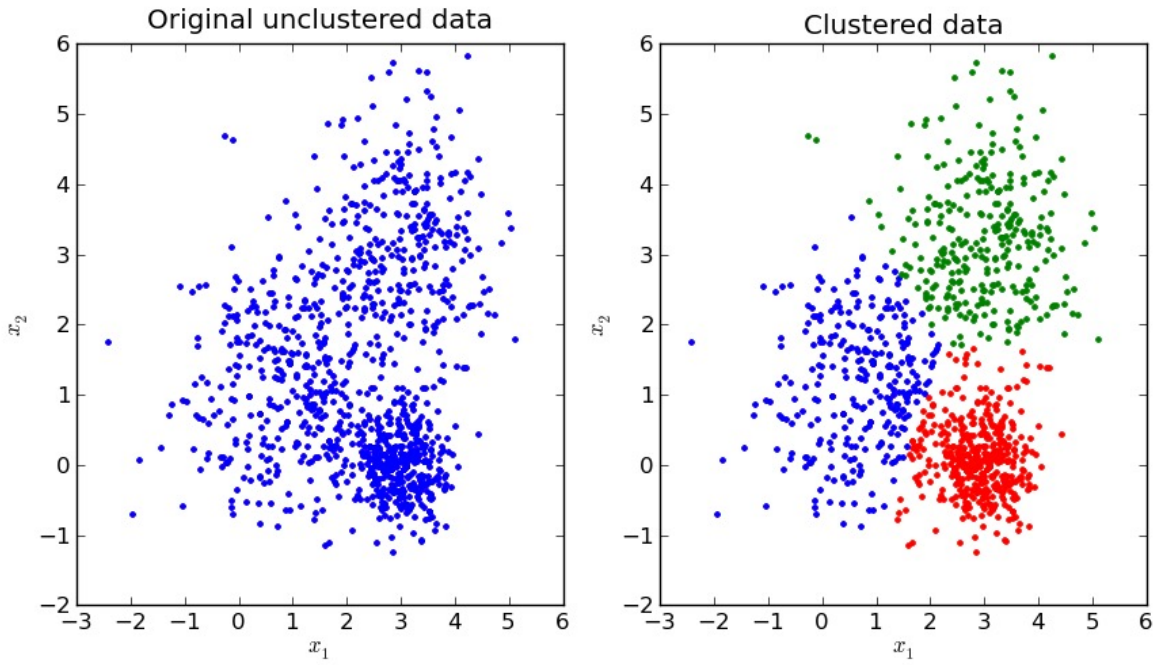
CIFAR-10: https://www.cs.toronto.edu/~kriz/cifar.html

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

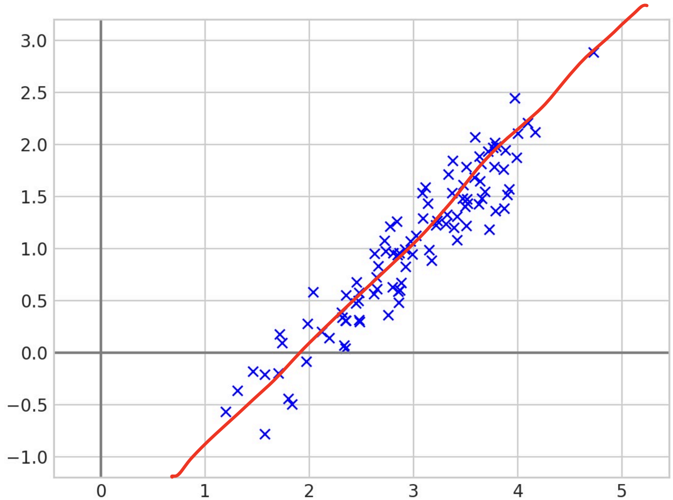
ImageNet: https://image-net.org/about.php

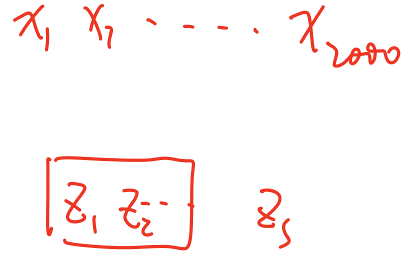
- Clustering

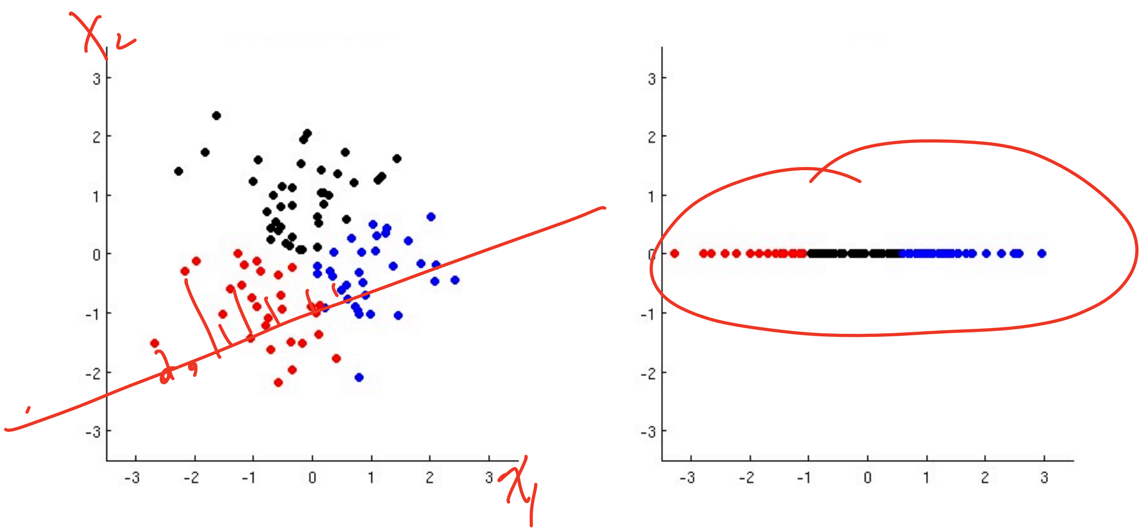
k-mean clustering



- principal component analysis

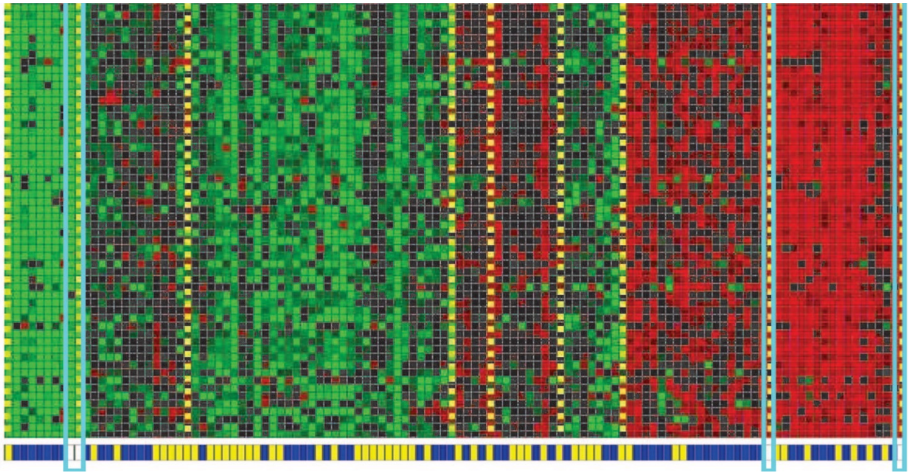


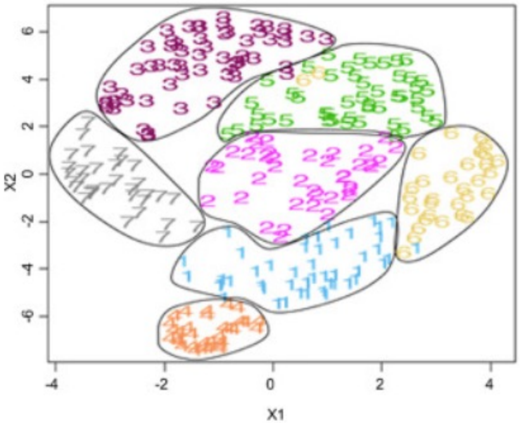


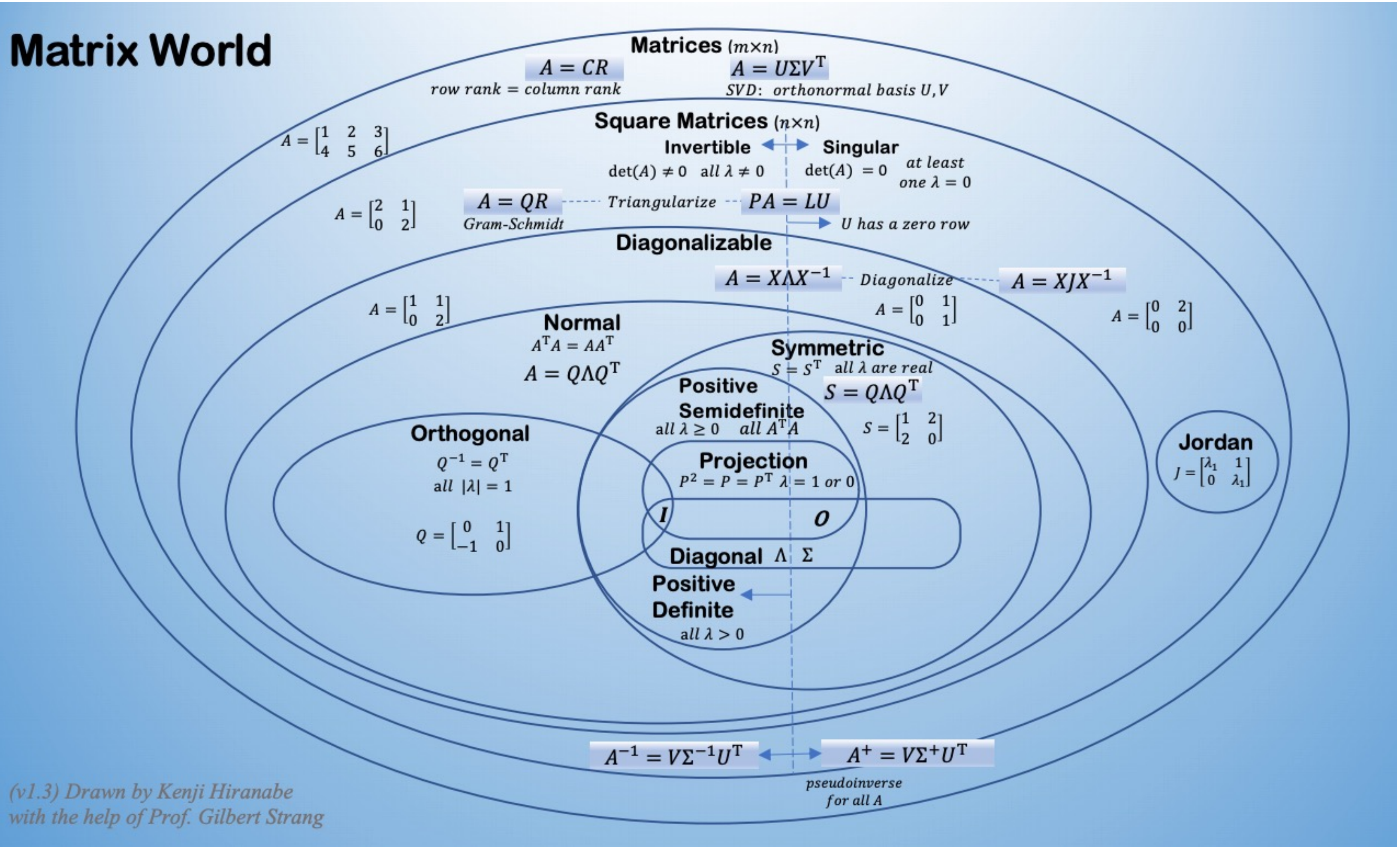


Application: Clustering Genes

23,000







Matrix World
Matrices $(m \times n)$
$A = CR$
row rank = column rank
$A = U\Sigma V^T$
SVD: orthonormal basis U,V
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
Square Matrices $(n \times n)$
Invertible
Singular
$\det(A) \neq 0$ all $\lambda \neq 0$
$\det(A) = 0$ at least one $\lambda = 0$
$A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$
$A = QR$
Gram-Schmidt
Triangularize
$PA = LU$
U has a zero row
Diagonalizable
$A = X\Lambda X^{-1}$
Diagonalize
$A = XJX^{-1}$
$A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$
$A = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$
$A = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$
Normal
$A^TA = AA^T$
$A = Q\Lambda Q^T$
Symmetric
$S = S^T$ all λ are real
$S = Q\Lambda Q^T$
$S = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}$
Positive Semidefinite
all $\lambda \geq 0$ all A^TA
Orthogonal
$Q^{-1} = Q^T$
all $|\lambda| = 1$
$Q = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
Projection
$P^2 = P = P^T$ $\lambda = 1$ or 0
I
O
Diagonal Λ Σ
Positive Definite
all $\lambda > 0$
Jordan
$J = \begin{bmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{bmatrix}$
$A^{-1} = V\Sigma^{-1}U^T$
$A^+ = V\Sigma^+U^T$
pseudoinverse for all A
(v1.3) Drawn by Kenji Hiranabe
with the help of Prof. Gilbert Strang

Disclaimer: In our Math 4570 class

- We will NOT do work like:

https://www.youtube.com/watch?v=fn3KWM1kuAw

https://www.youtube.com/watch?v=tF4DML7FIWk