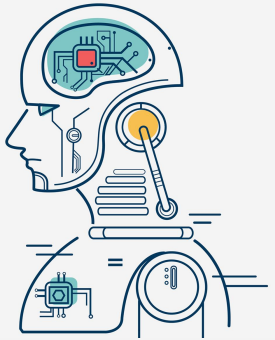
IEEE DTU Student Branch

TechWeek 2021

Machine Learning



Mentors

Priyansh Tyagi

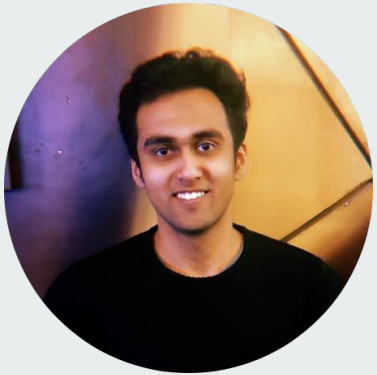
priyanshty19

Sakshi Arora

Sakshi-8

Pranay Khosla

PranayK-666



Jatin Pandey

jatinpandey02

Varang Rai

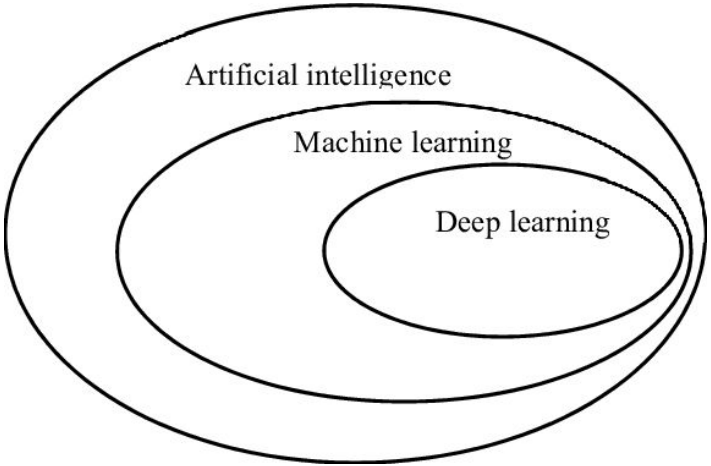
varangrai

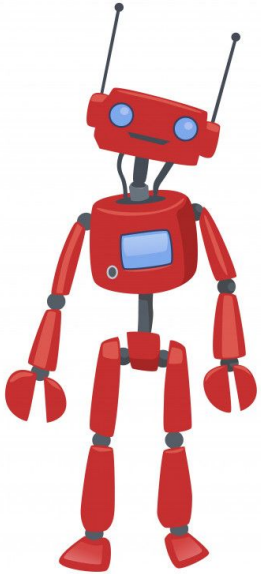
What are we going to do today?

- Streamlit Predictor App
- Artificial Intelligence and ML's use in today's world
- Basics of Python
- Regression and Classification
- Neural Networks
- Creating the web app

Artificial Intelligence

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving





Technologies Used

Web Application for Live Code

Development Platform for Version Control

Google's Cloud based Python Notebook

Open Source Programming Language

ANACONDA®

World's most famous distribution of Python for Scientific Computing

TensorFlow

Open Source library for Deep Learning

Libraries for Data Science

NumPy

Numerical Computing

Visualization Patterns

Machine Learning

Basic Plotting

Data Manipulation

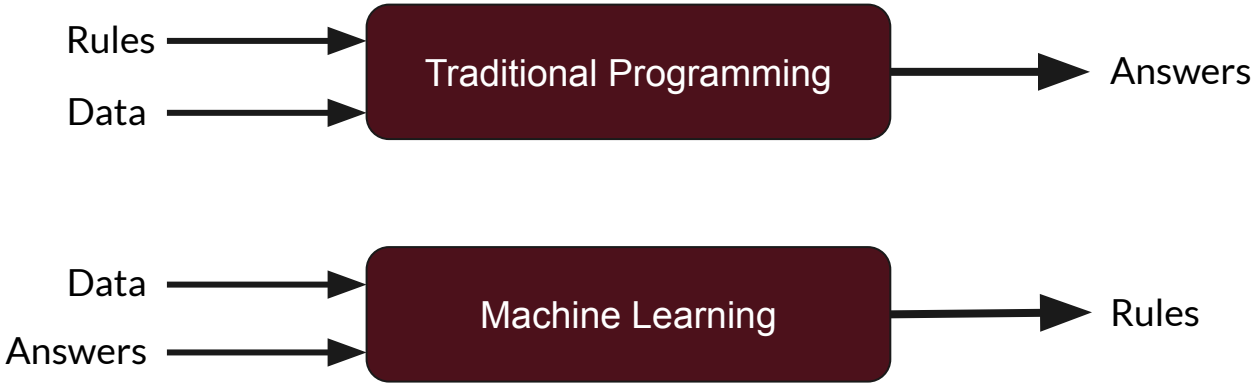
Formal Definition of Machine Learning

“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed"

~ Arthur L. Samuel, AI pioneer, 1959

"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

~ Tom Mitchell, Professor at Carnegie Mellon University

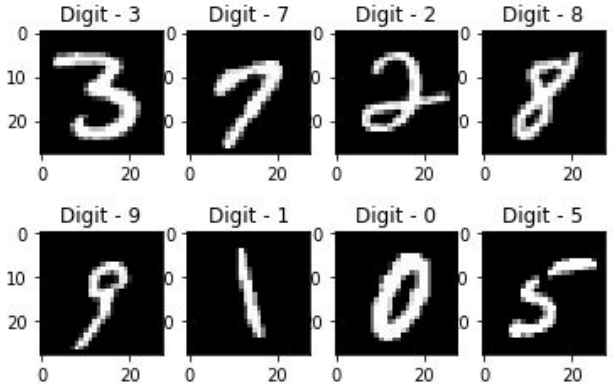

Rules
Data
Traditional Programming
Answers
Data
Answers
Machine Learning
Rules

Applications of Machine Learning

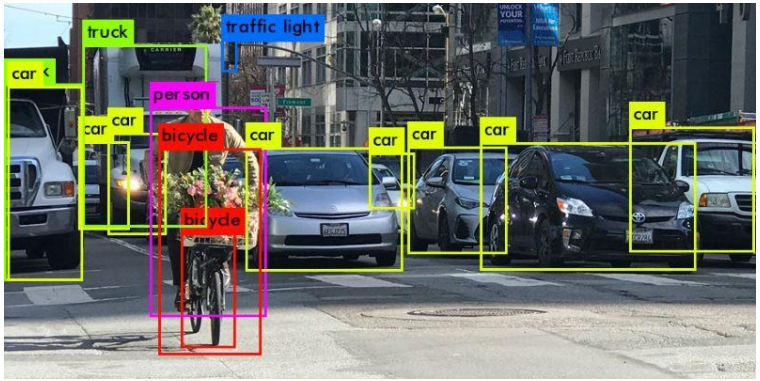
Image Recognition

Google Lens

Face Id

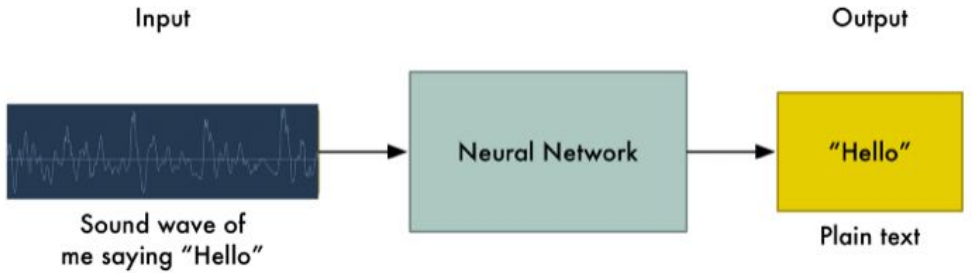


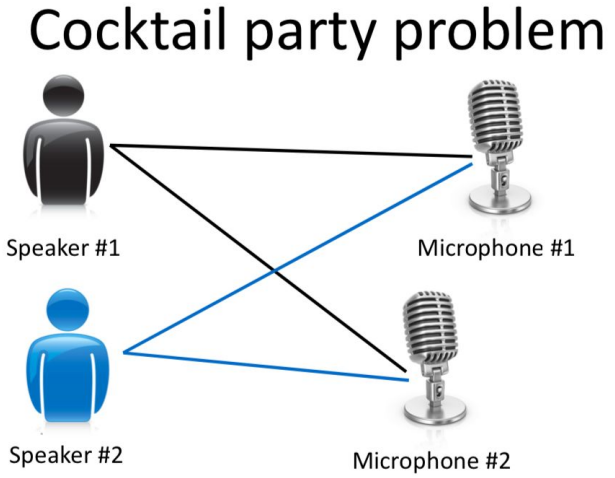
Digit Recognition



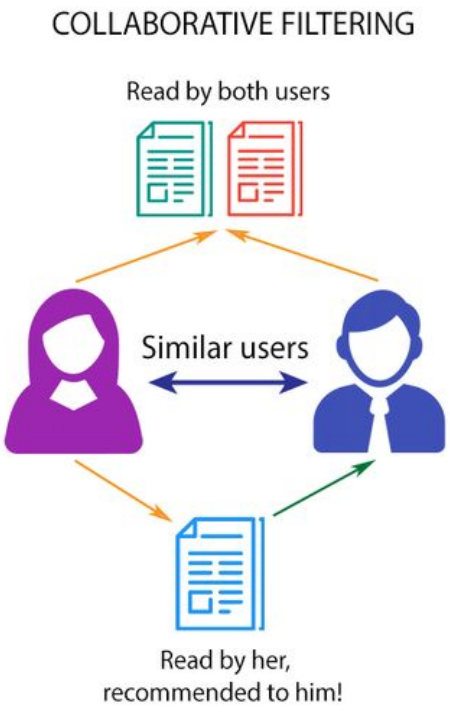
Object Detection

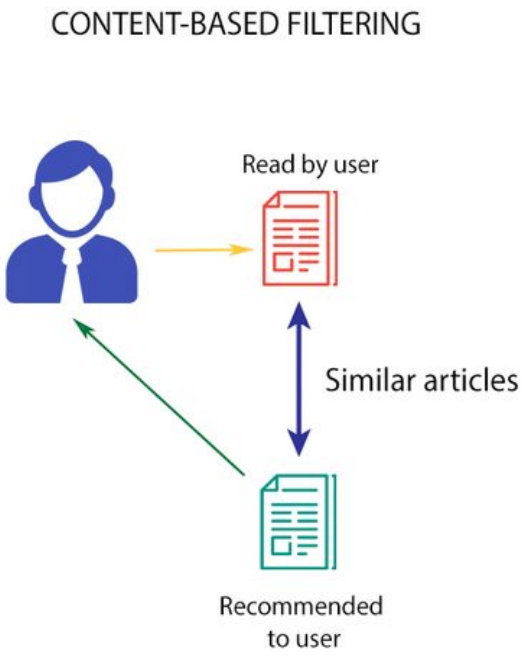
Voice Recognition



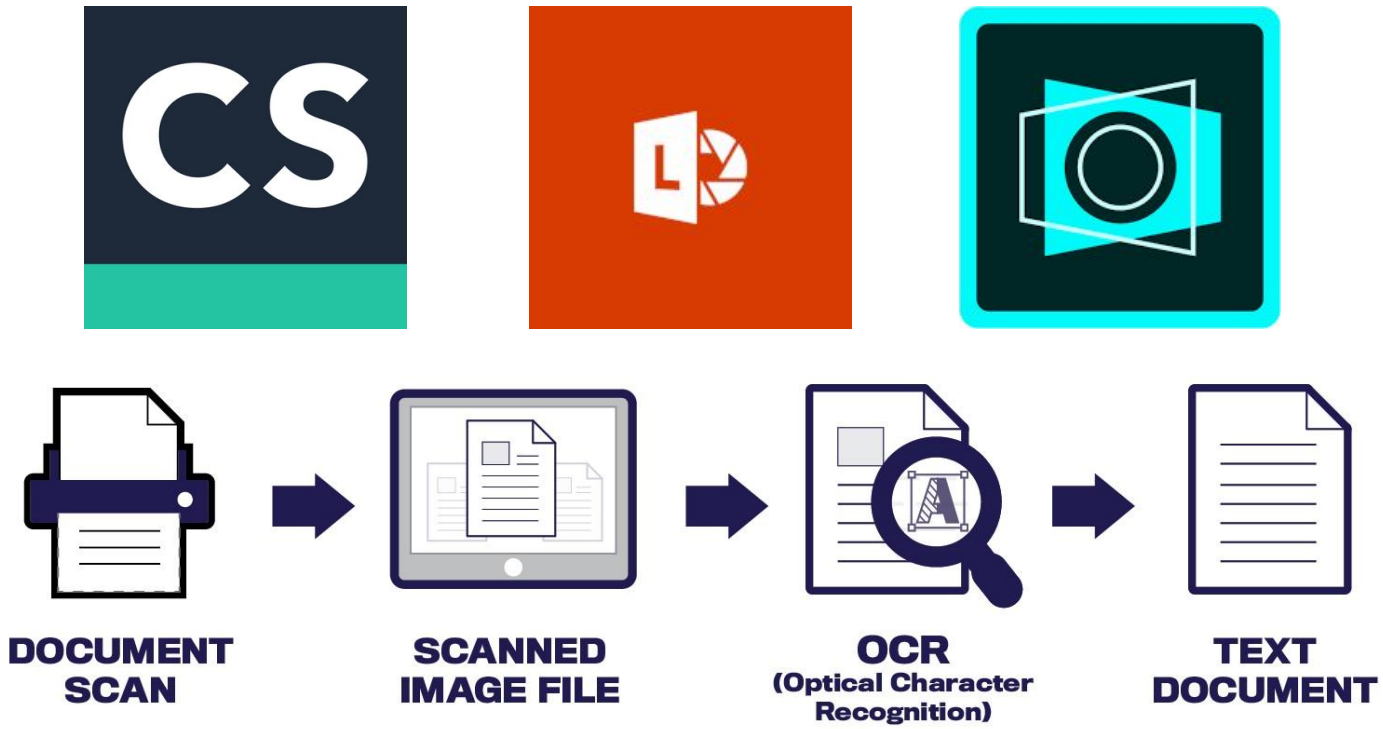


Recommender Systems





Optical Character Recognition



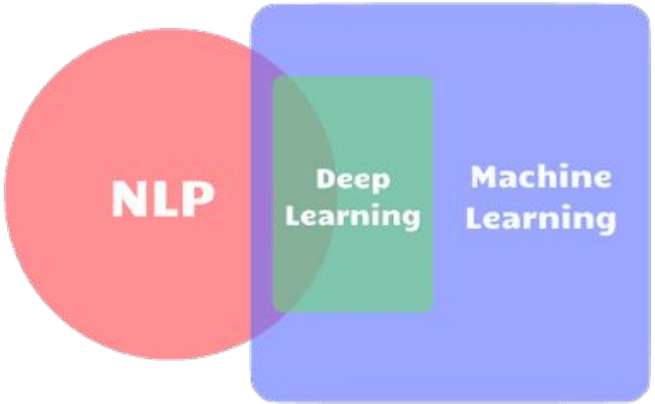
Self Driving Cars

Natural Language Processing

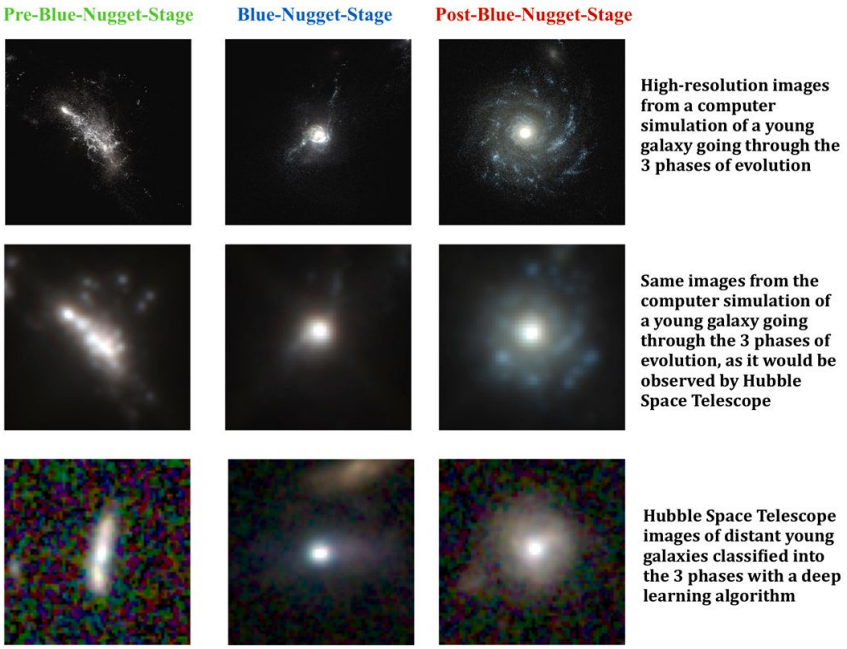
text

ML

Web search
Anti-spam
Machine translation
...



Machine Learning in Astrophysics



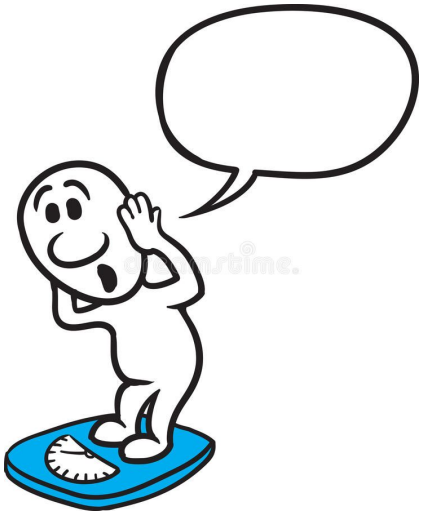
Examples of simulated galaxy evolution through the 3 stages: Pre-Blue Nugget (often elongated, i.e. pickle shaped), Blue Nugget (compaction phenomenon: gas infall leads to central starburst), and Post-Blue Nugget (often with star-forming disk), with similar galaxies observed by Hubble Space Telescope. The width of each image is approximately 100,000 light years. Credits: simulations Daniel Ceverino and Joel Primack, simulated images Greg Snyder and Marc Huertas-Company, HST observations CANDELS.

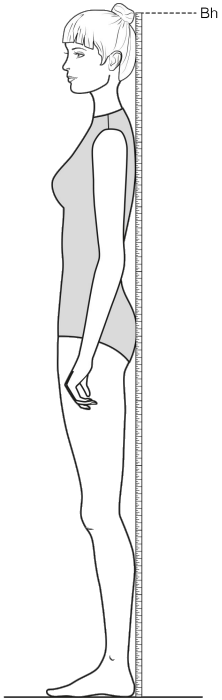
Before starting let's look at a couple of real life problems

Classification problem

Healthcare problems

- Given a patient's height and weight
- A decision is needed: is the patient diabetic or not
- Cant have a medical test every week





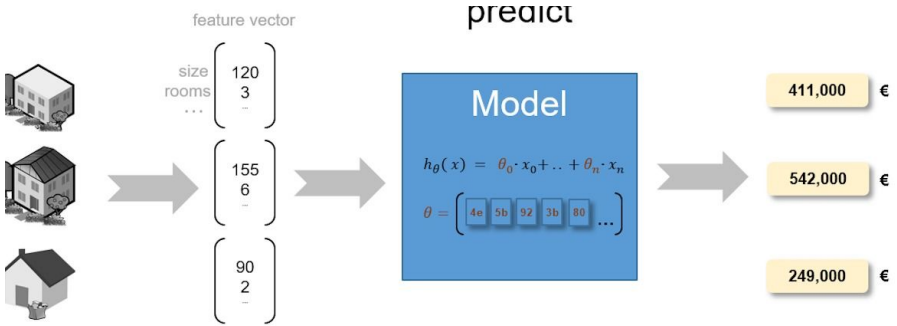
Regression problem

Approved or not

Real Estate

Size of House	Lot Size (acre)	# of Bedrooms	# of Bathrooms	Price of House
950	2.5	2	1	$127,325
1,535	1.5	2	2	$156,570
1,605	2.25	3	1.5	$158,895
1,905	2.5	2	1.5	$200,025
2,057	2.25	3	2	$230,384
2,227	2.75	3	2	$233,835
3,150	1	4	2	$261,420
3,620	3	4	3	$433,500

- Learn a regression model from the data
- Use the model to predict prices of newly built house
- What is the class for following case/instance?



Test Data:

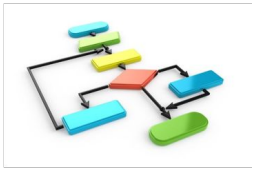
Size of House	Lot Size (acre)	# of Bedrooms	# of Bathrooms	Price of House
3,620	3	4	3	$433,500

Machine learning: **Supervised Learning**

- Like human learns from examples
- A computer also learns the same way
- **So it learns from data**

Orange

Apple

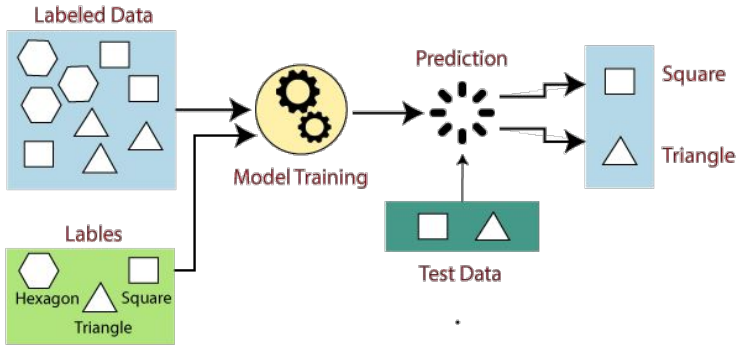


- Our focus: To learn a model that can predict something

The data and the goal

- Data: *some* features
  - a target variable: pre-defined label, discrete or continuous.

- Goal: To learn a model from the data that can be used to predict the classes of new (future, or test) cases/instances.



Using Python for ML

Let’s look at some benefits that Python has over other programming languages when it comes to ML:

- Easy to learn; offers concise and readable code
- Broad choice of AI explicit libraries and structures
- Platform independence
- Popular language; flaunts a vast, dynamic network of designers who are glad to offer assistance and backing

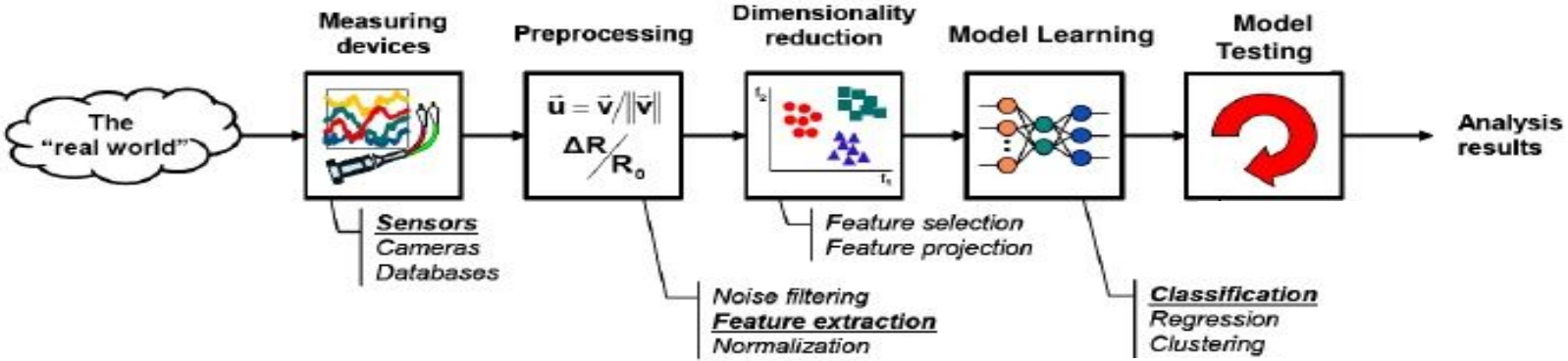
Documentation links

The following are the links to the official websites of the libraries we will be using today. You can visit them to know more about them and explore them on your own.

- NumPy: NumPy.org
- Pandas: pandas - Python Data Analysis Library
- Matplotlib: Matplotlib: Python plotting — Matplotlib 3.3.3 documentation
- Seaborn: seaborn: statistical data visualization — seaborn 0.11.1 documentation
- Scikit-learn: scikit-learn: machine learning in Python — scikit-learn 0.24.1 documentation

LET'S CODE!

A Generalized Machine Learning System



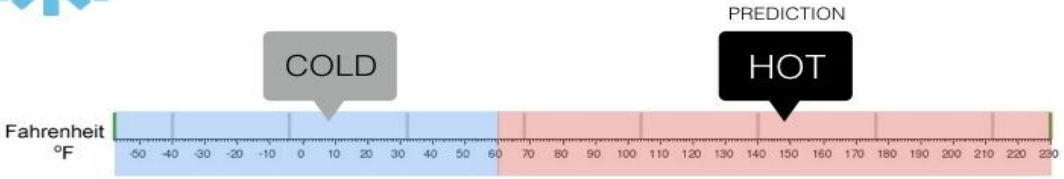
Types of Machine Learning Problems

Regression

What is the temperature going to be tomorrow?

Classification

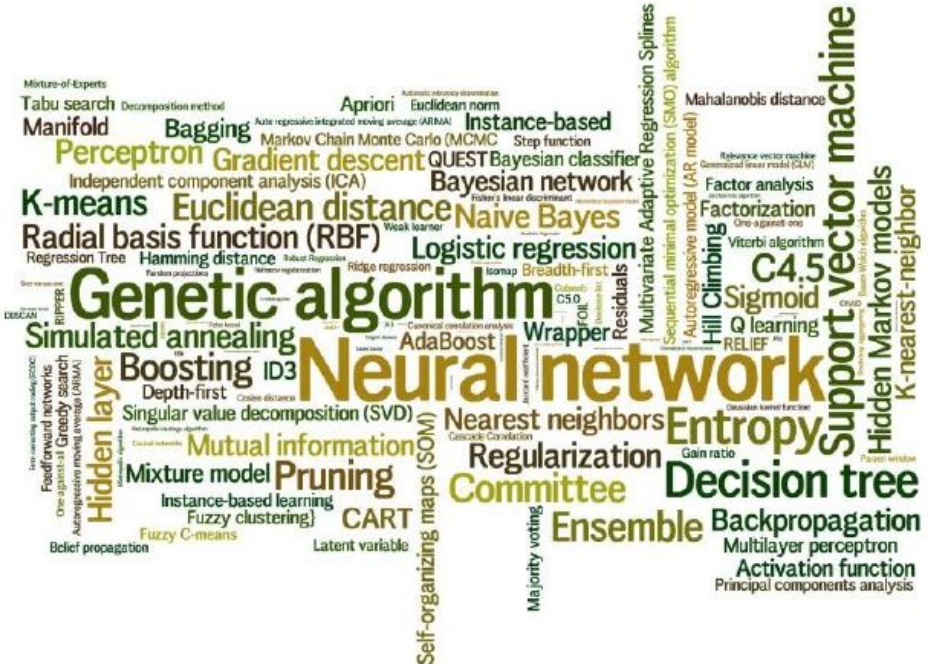
Will it be Cold or Hot tomorrow?



Solutions to Machine Learning Problems

- **Supervised -** the data is labeled to tell the machine exactly what patterns it should look for
- **Un-supervised -** the data has no labels. The machine just looks for whatever patterns it can find.
- **Reinforcement -** the latest frontier of machine learning. A reinforcement algorithm learns by trial and error to achieve a clear objective. It tries out lots of different things and is rewarded or penalized depending on whether its behaviors help or hinder it from reaching its objective.

Different Types of Learning Algorithms



Terminology and Notations

$\mathbf{x} \in \mathcal{X}$ – input variables ("features")

$t \in \mathcal{T}$ – a target variable

$\{\mathbf{x}_n\}, n = 1, \ldots, N$ – given N observations of input variables

$(\mathbf{x}_n, t_n)$ – a training example or y E Y

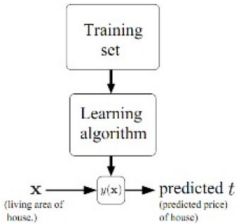
$(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)$ – a training set

Goal

Find a function $y(\mathbf{x}) : \mathcal{X} \to \mathcal{T}$ ("hypothesis") to **predict** the value of t for a new value of $\mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_d \end{bmatrix}$$

Feature vector



When the target variable t is continuous

$\Rightarrow$ a **regression** problem

In the case of discrete values

$\Rightarrow$ a **classification** problem

The following function table reflects a linear relationship. Write an equation in slope-intercept form (y = mx + b) to represent the relationship of values shown in the table.

x	y
4	36
6	56
9	86
15	146
20	196

2 3 6 5

20 30 60 50

$y = mx + b$

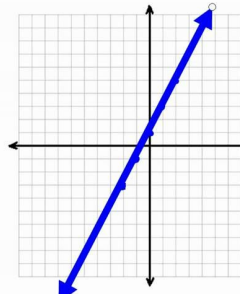
$\frac{\Delta Y}{\Delta X}$

$\frac{y_2 - y_1}{x_2 - x_1}$

20

function using a tabl

the equation is solved for y



Linear Regression

Simplest **'model'/mapping** from x to y.

where y is a continuous variable
It assumes a linear relationship between the input variables and output variables

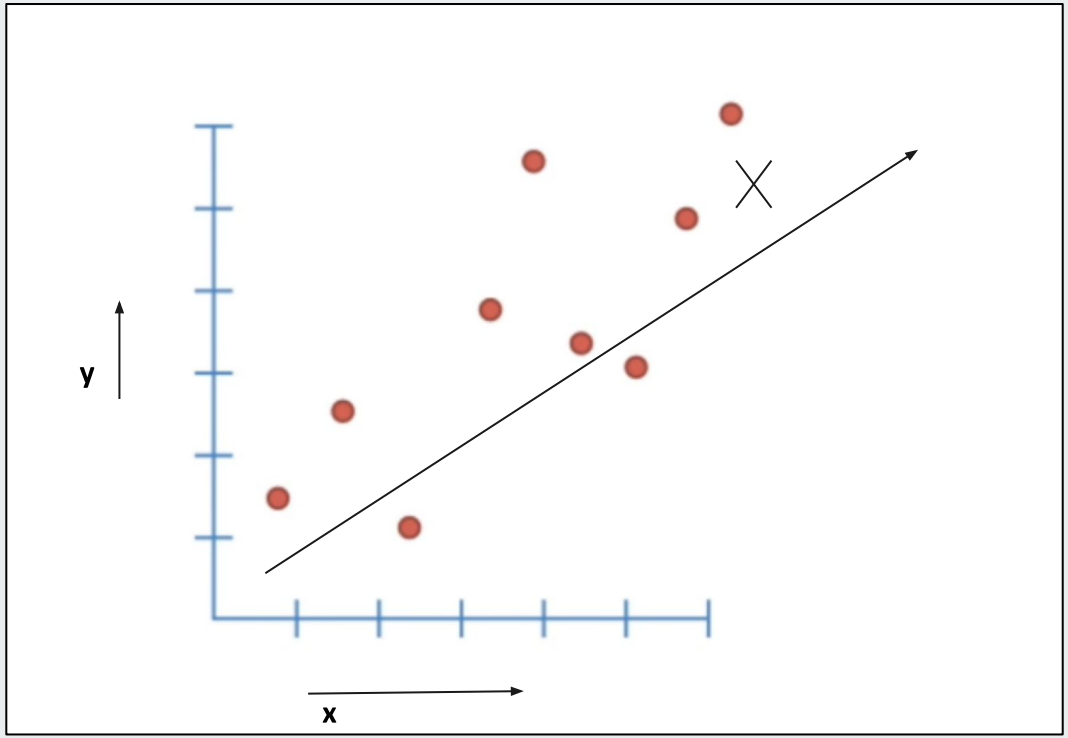
$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D,$$

where $\mathbf{x} = (x_1, \ldots, x_D)$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 \quad h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

How to choose these weights?

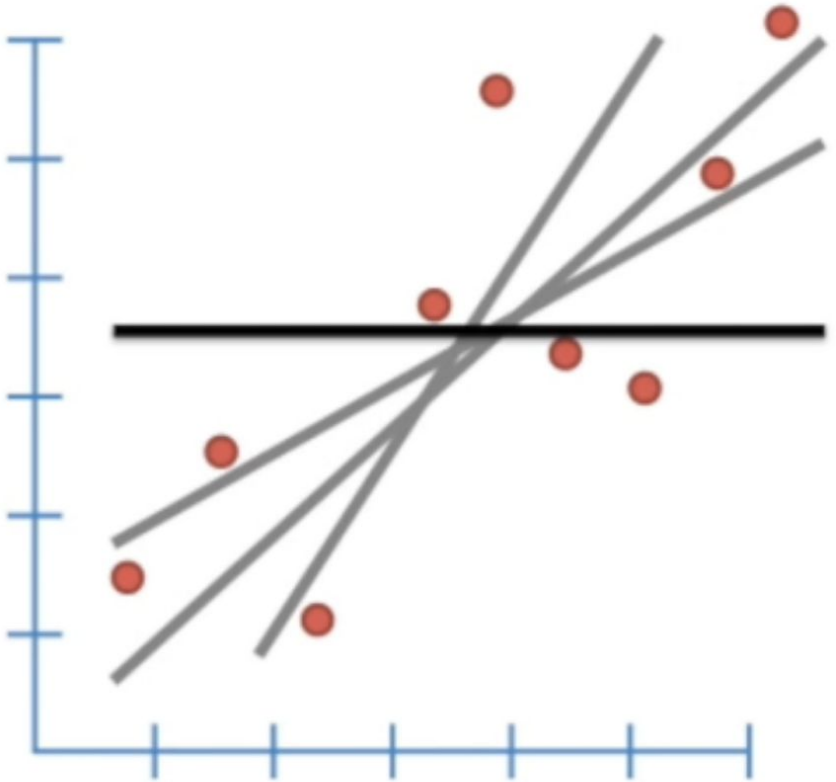
Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y)

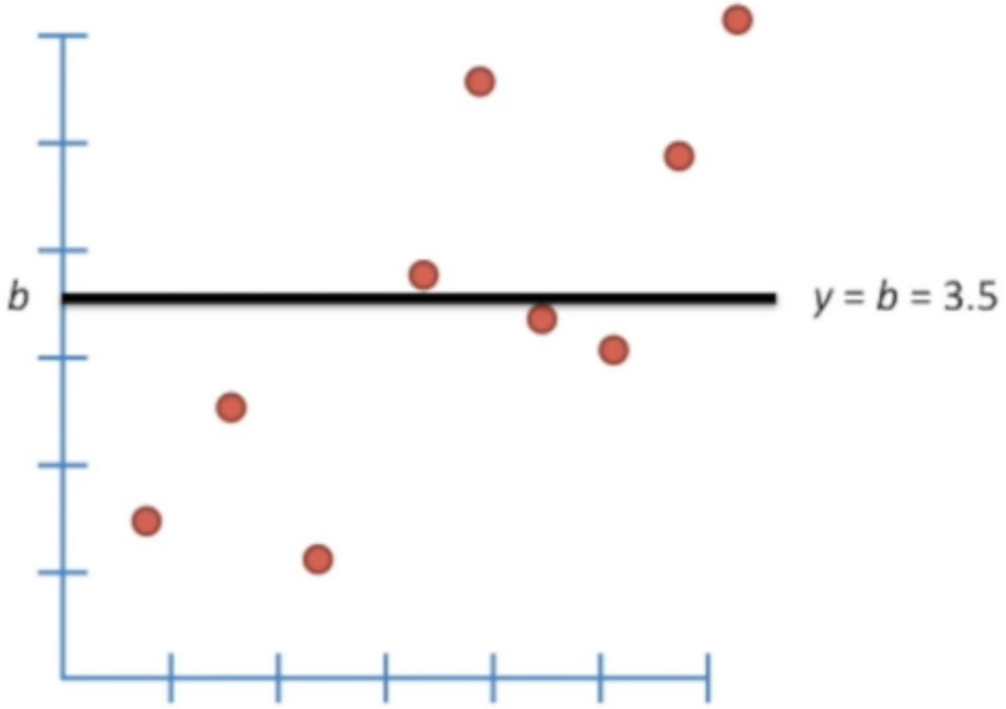


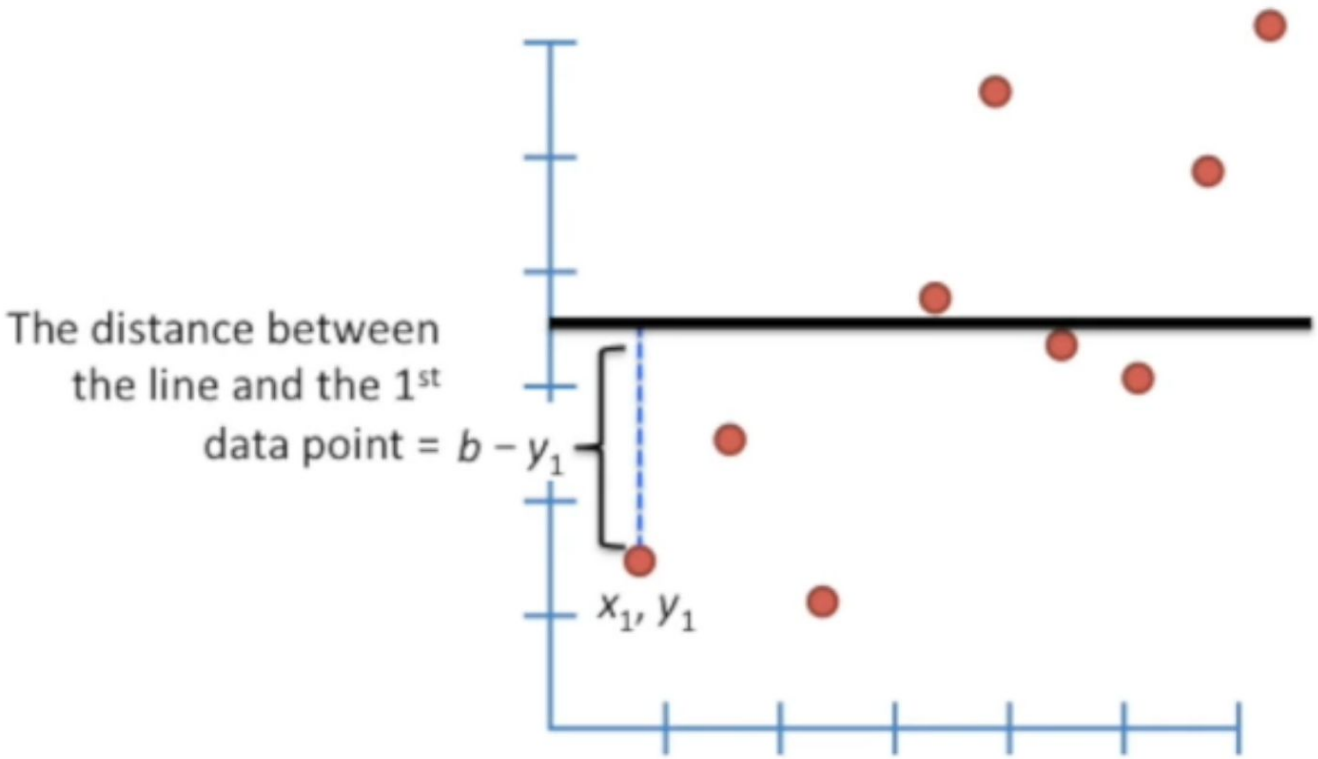
Step 1: Get the data and plot it

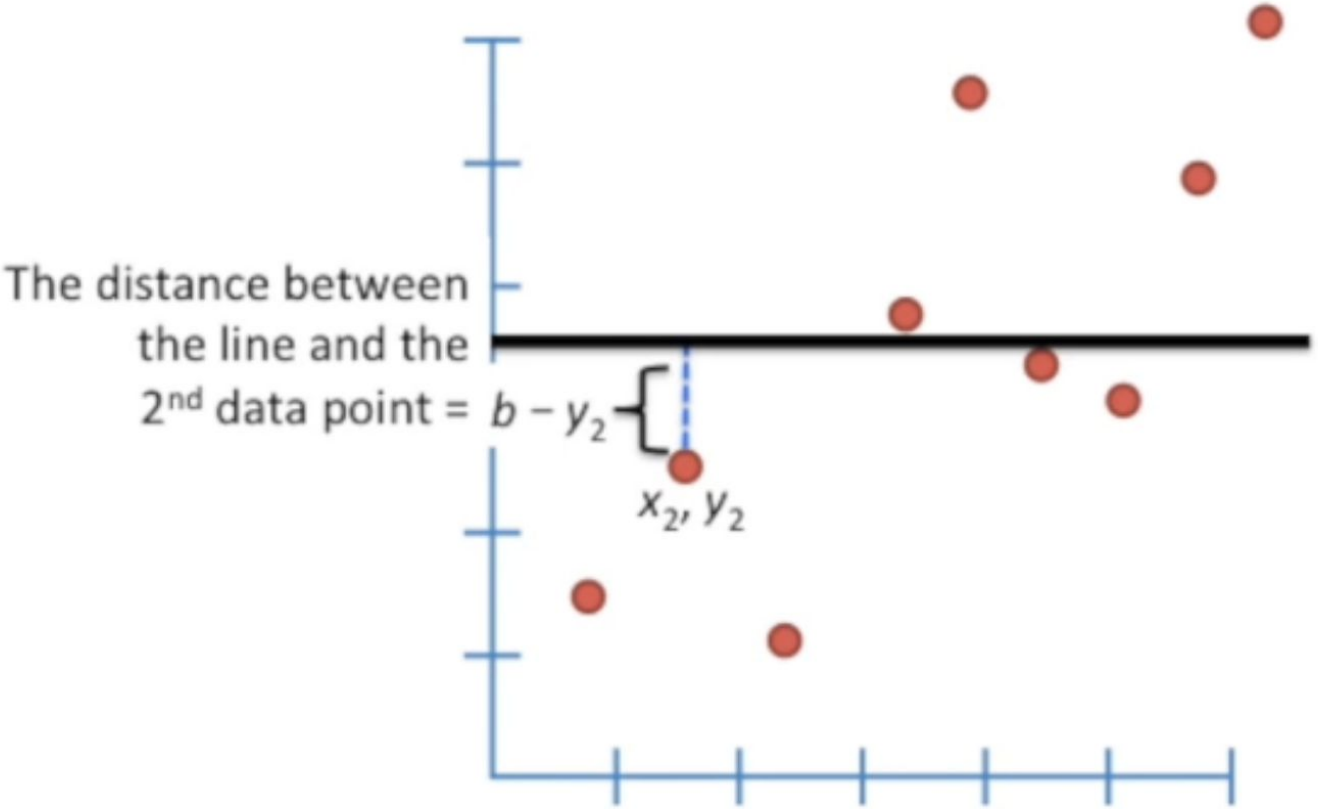
Plot size of house on x axis
Price on y axis

$h_\theta(x) = \theta_0 + \theta_1 x_1$



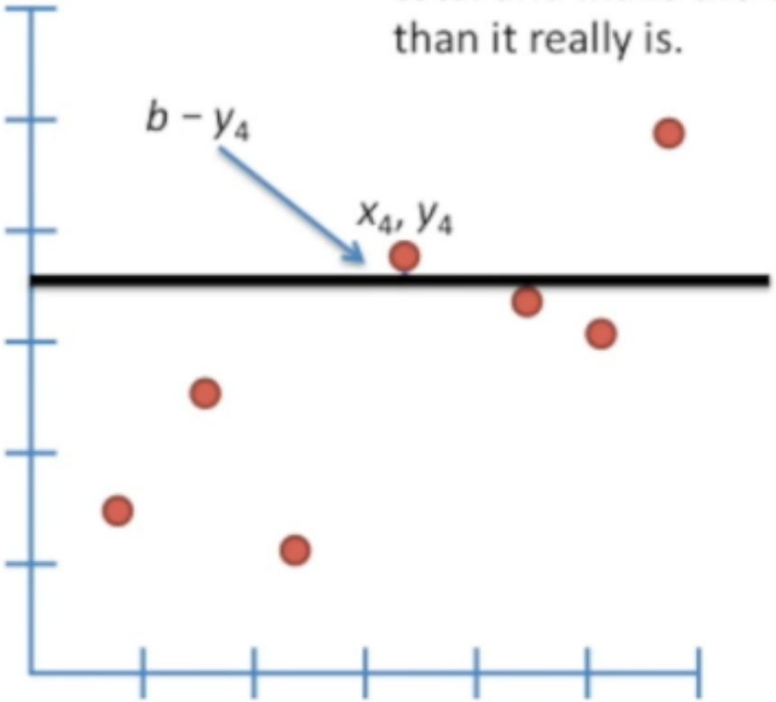

b
y = b = 3.5
w0


The distance between the line and the 1st data point = $b - y_1$
x_1, y_1

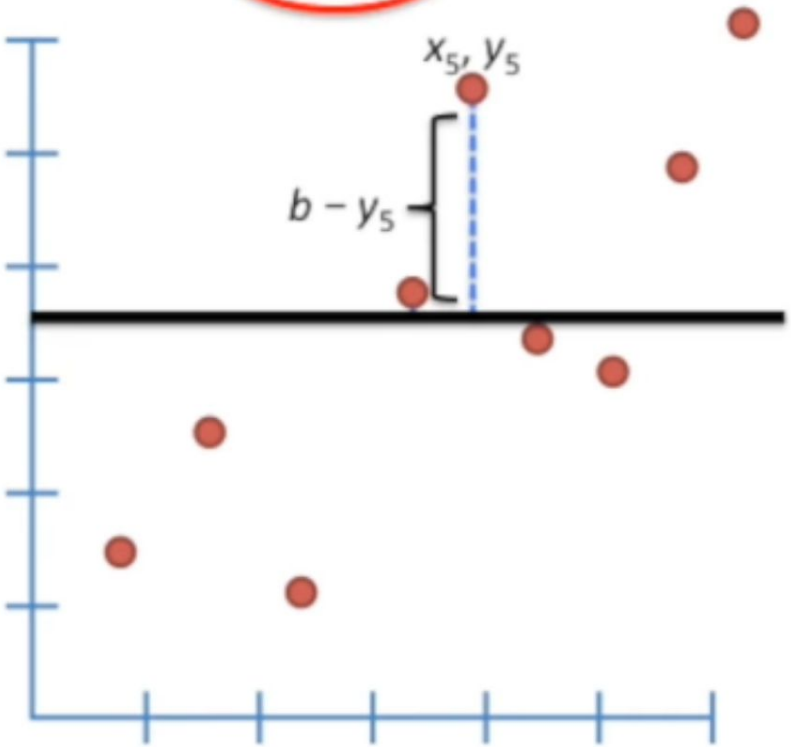

The distance between
the line and the
2nd data point = $b - y_2$
x_2, y_2

$$(b - y_1) + (b - y_2) + (b - y_3) + (b - y_4)$$

NOTE: $y_4 > b$, so this value will be negative. That's no good, since it will subtract from the total and make the overall fit appear better than it really is.

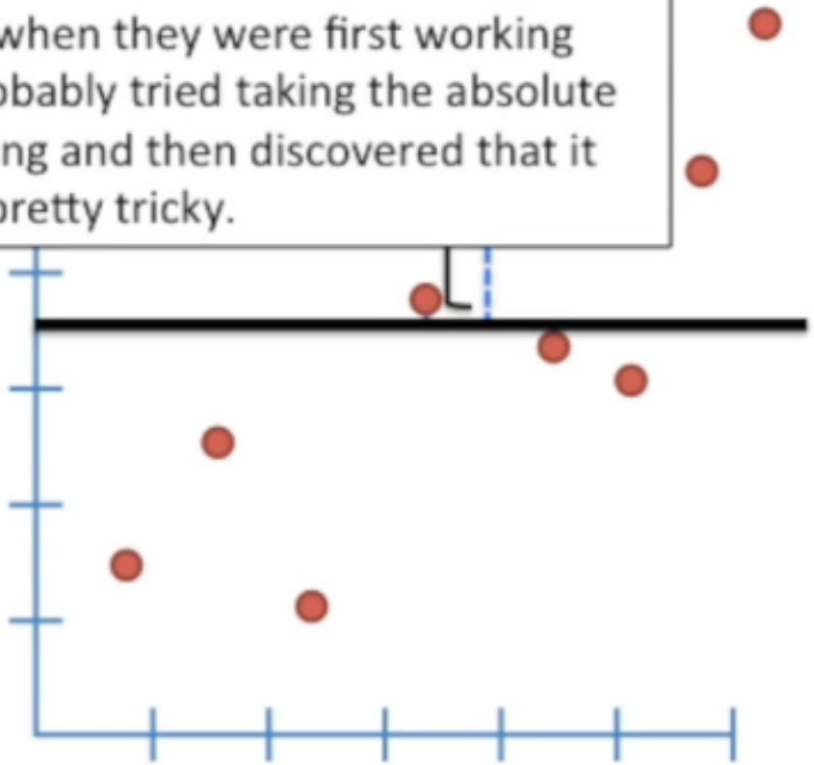


$(b - y_1) + (b - y_2) + (b - y_3) + (b - y_4) + (b - y_5)$ ← This is even worse!!!

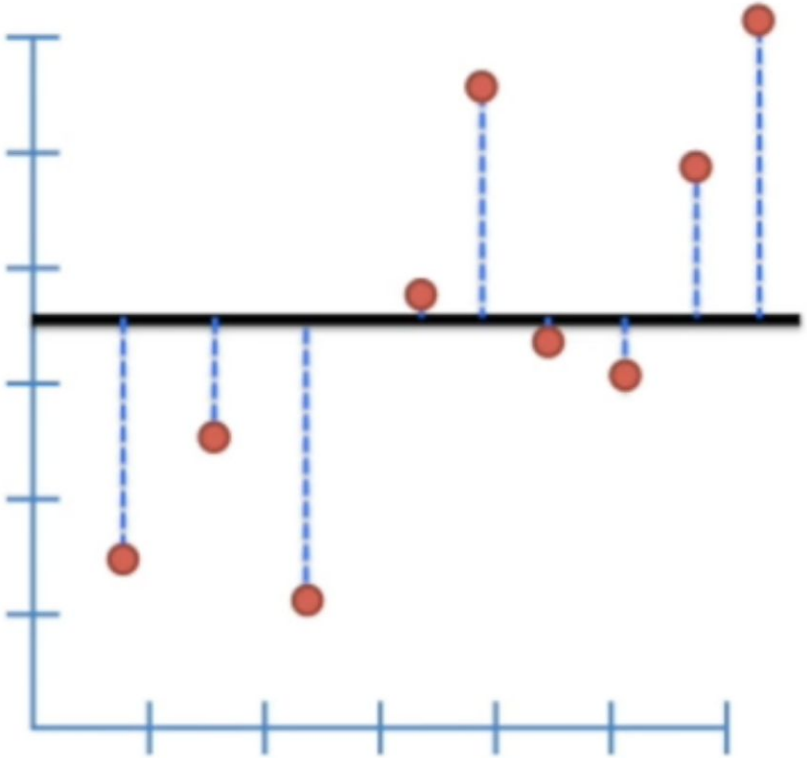


$$|(b - y_1)| + |(b - y_2)| + |(b - y_3)| + |(b - y_4)| + |(b - y_5)|$$

Back in the day, when they were first working this out, they probably tried taking the absolute value of everything and then discovered that it made the math pretty tricky.



$$(b - y_1)^2 + (b - y_2)^2 + (b - y_3)^2 + (b - y_4)^2 + (b - y_5)^2 + (b - y_6)^2 + (b - y_7)^2 + (b - y_8)^2 + (b - y_9)^2$$

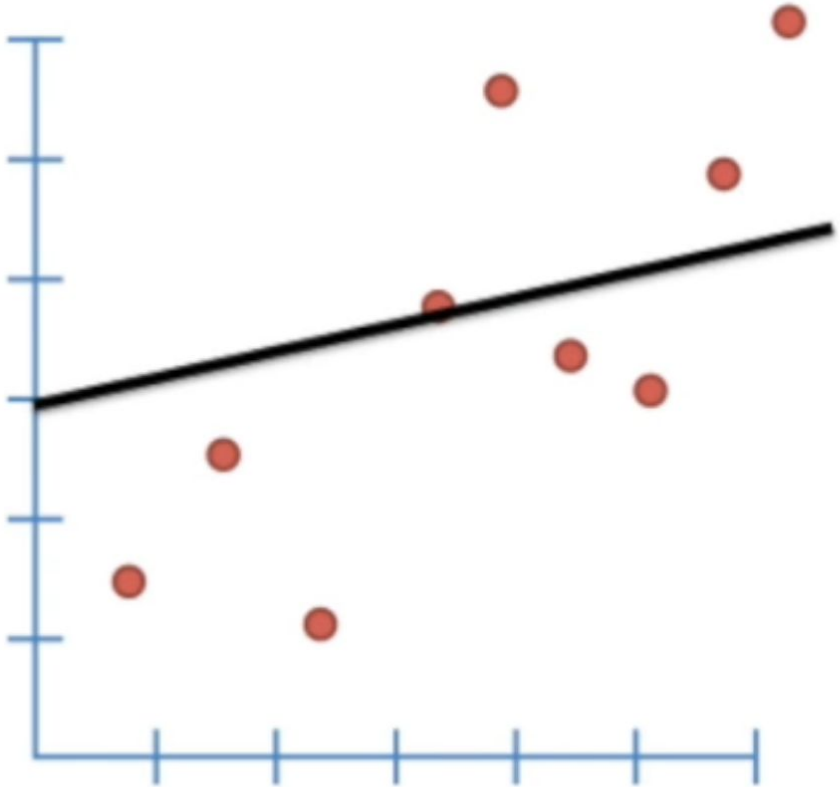


= 24.62

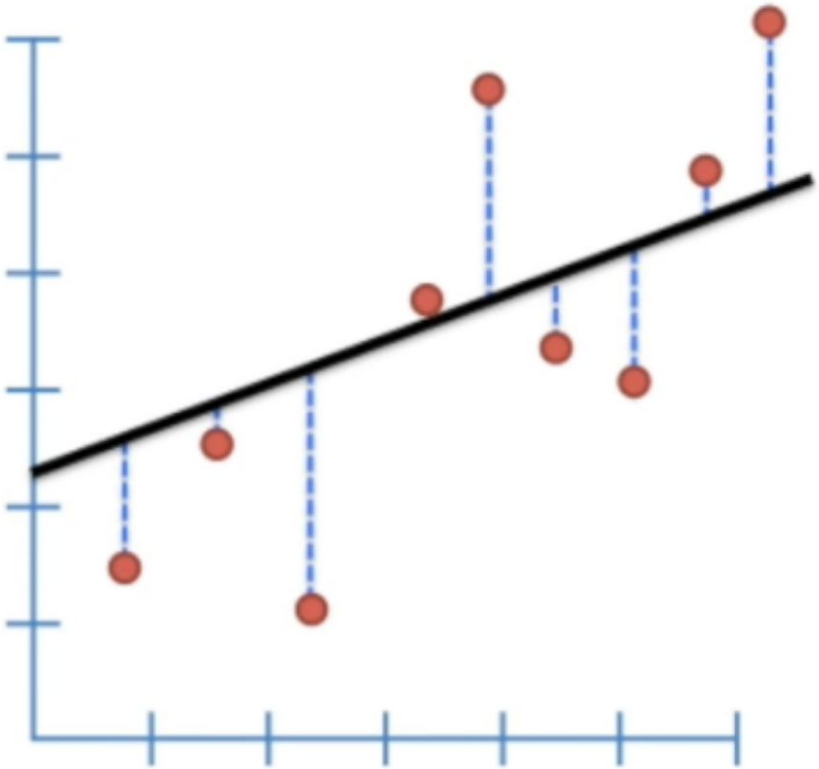
This is our measure of how well this line fits the data.

It's called the "sum of squared residuals, because the residuals are the differences between the real data and the line, and we are summing the square of these values.

Now let's see how good the fit is if we rotate the line a little bit.

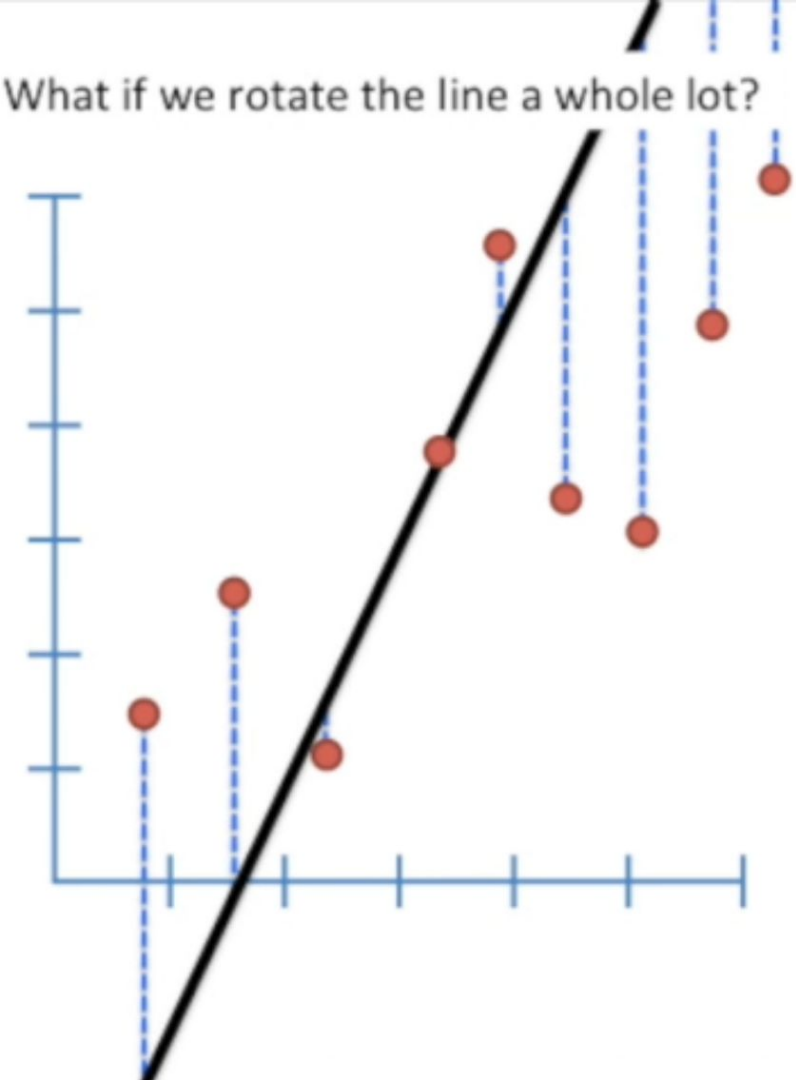


Does this fit improve if we rotate a little more?

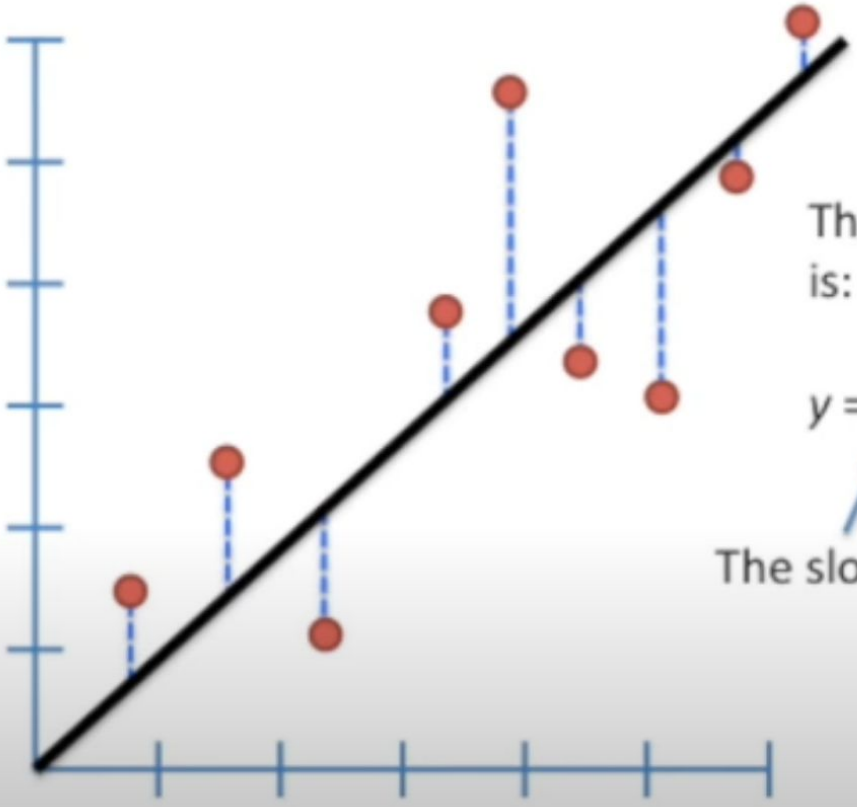


YES! The sum of squared residuals = 14.05

What if we rotate the line a whole lot?



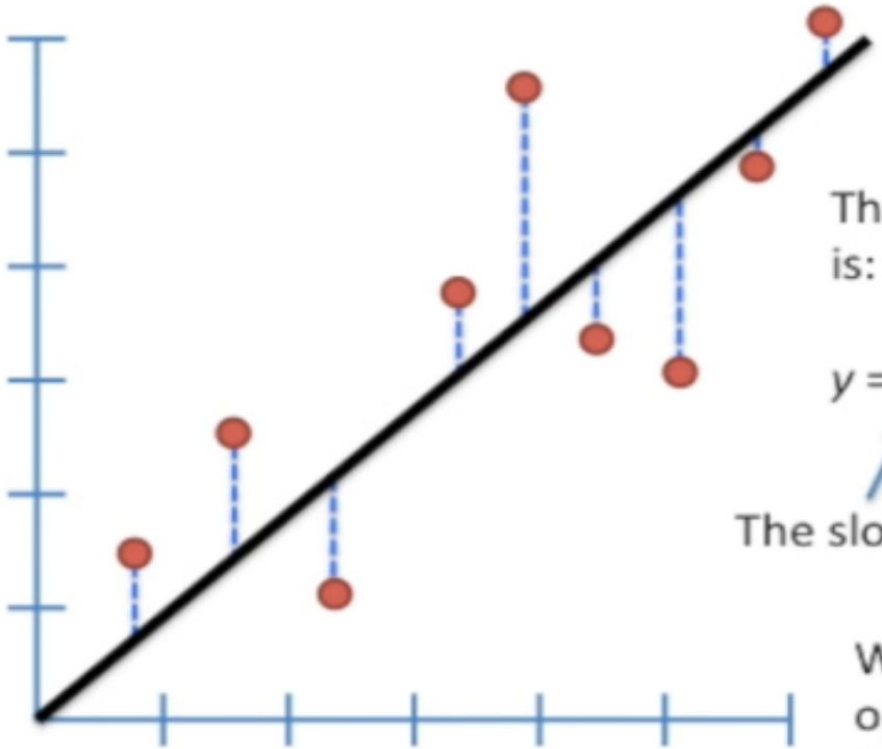
The fit gets worse. In this case the sum of squared residuals = 31.71



The generic line equation is:

$y = a^*x + b$

The slope… …the intercept



The generic line equation is:

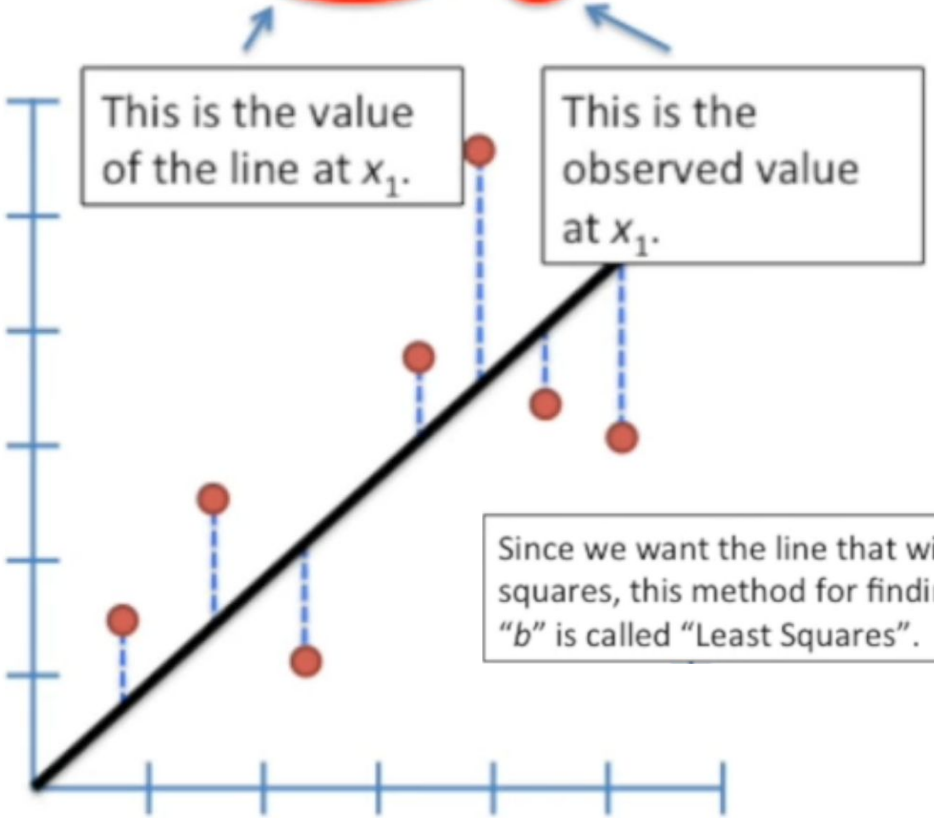
$y = a*x + b$

The slope… …the intercept

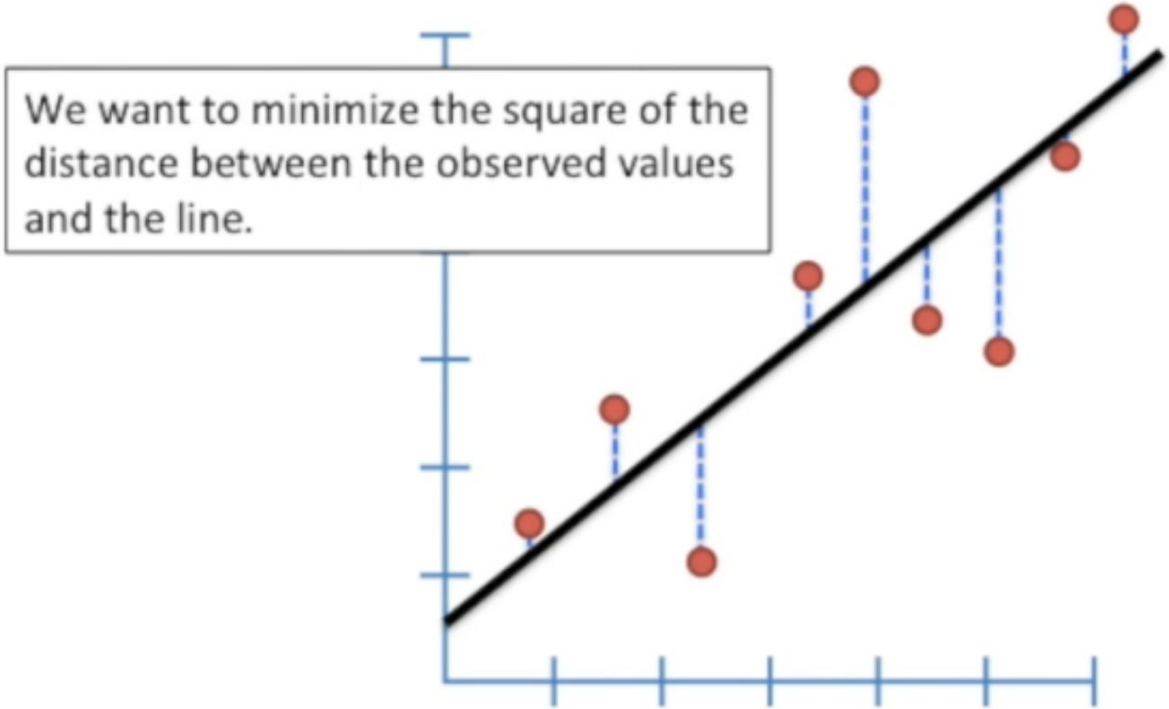
We want to find the optimal values for "*a*" and "*b*" so that we minimize the sum of squared residuals.

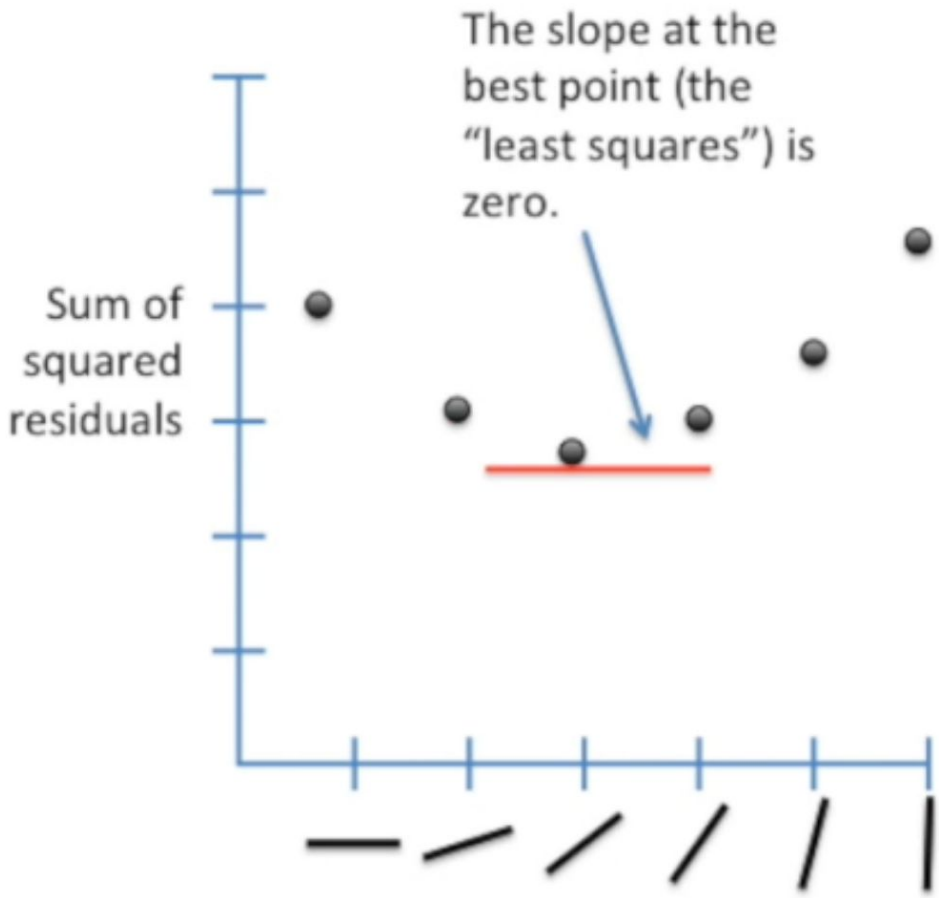
In more general math terms…

Sum of squared residuals = $((a*x_1 + b) - y_1)^2 + ((a*x_2 + b) - y_2)^2 + \ldots$



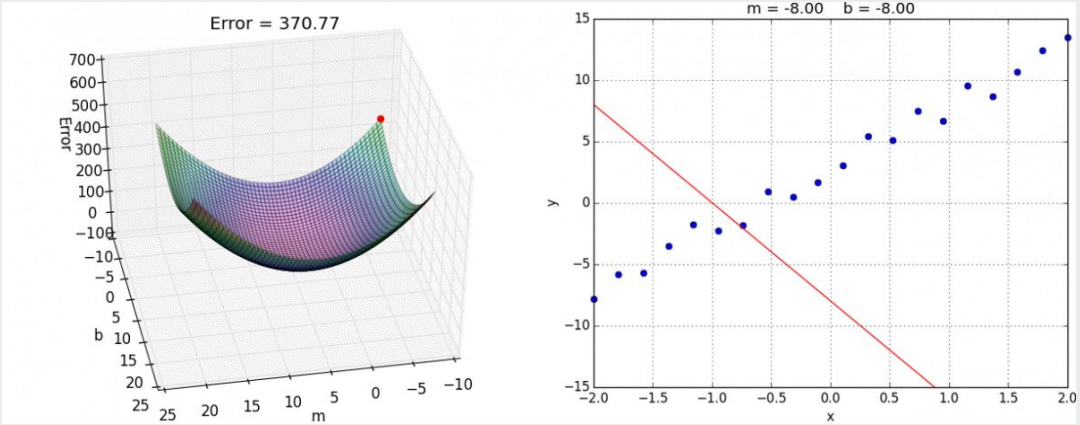
Since we want the line that will give us the smallest sum of squares, this method for finding the best values for "*a*" and "*b*" is called "Least Squares".


We want to minimize the square of the distance between the observed values and the line.

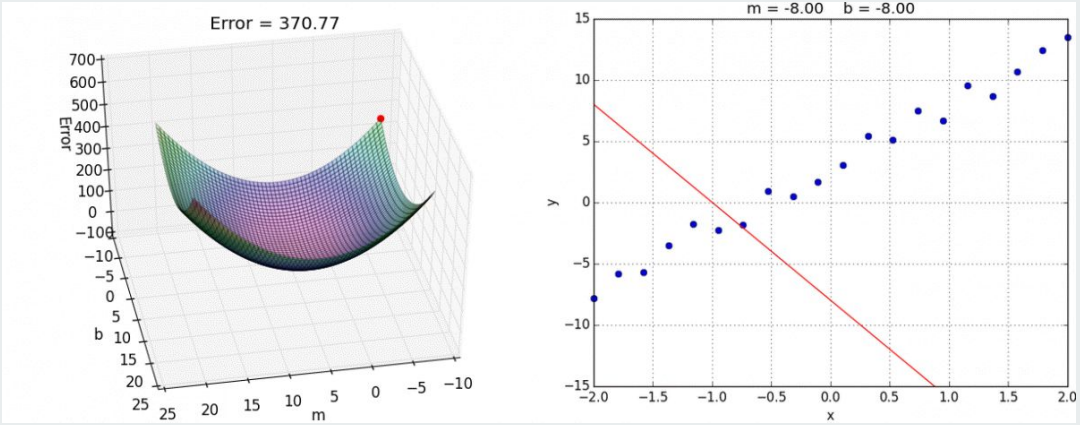


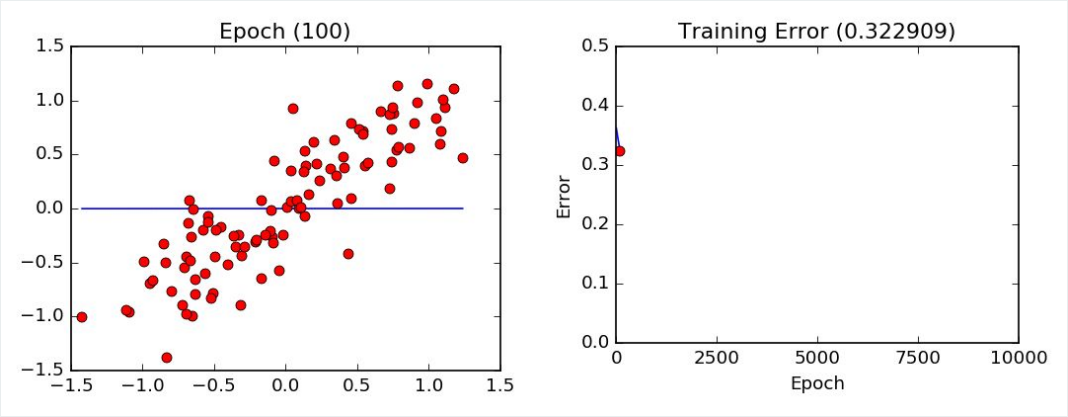
Only varying one parameter (slope)

Cost Function Graphical Representation



Squared error does not have local minimum


Error = 370.77
Error
b
m
m = -8.00 b = -8.00
y
x


Epoch (100)
Training Error (0.322909)
Error
Epoch

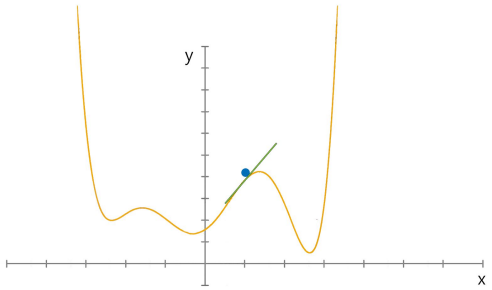
How to minimize the loss function?

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2.$$

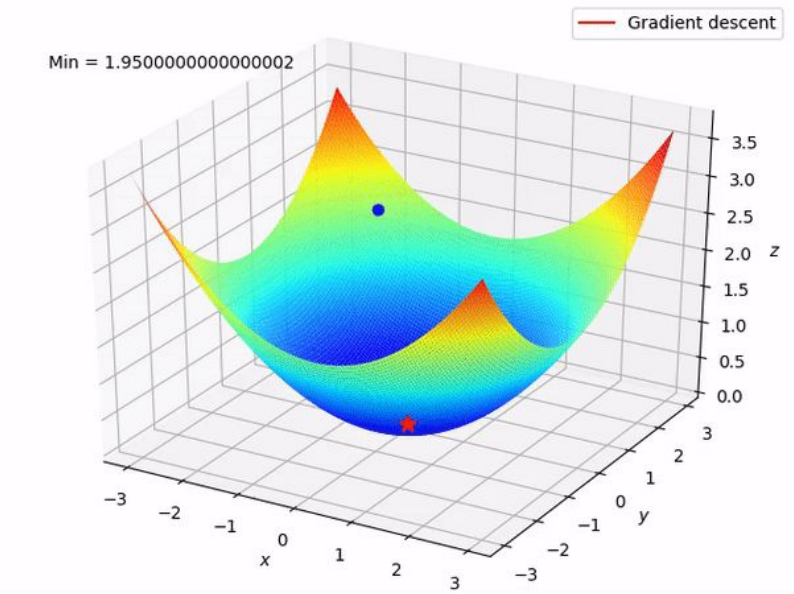
The normal equations

we will minimize J by explicitly taking its derivatives with respect to θ_j's, and setting them to zero

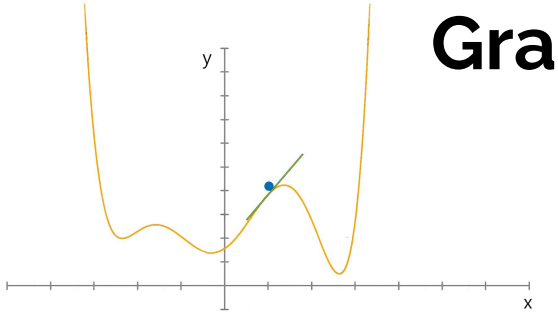
$$\theta = (X^T X)^{-1} X^T \vec{y}.$$



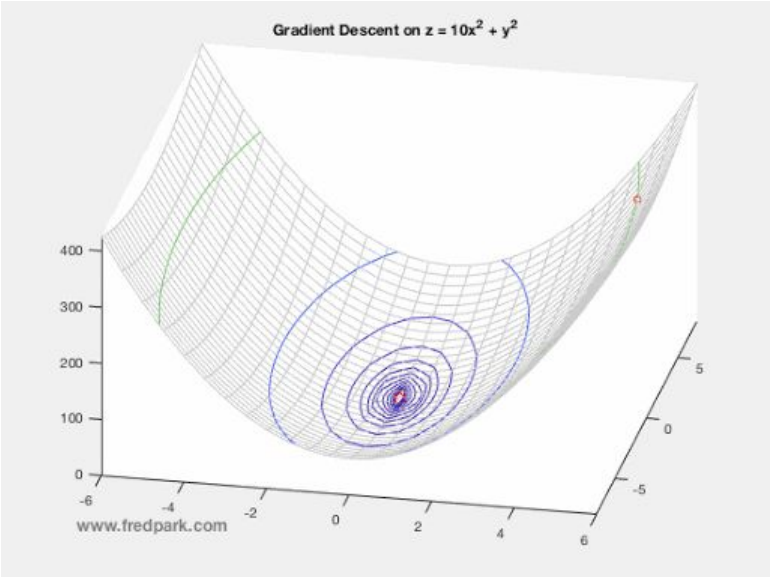
gradient descent



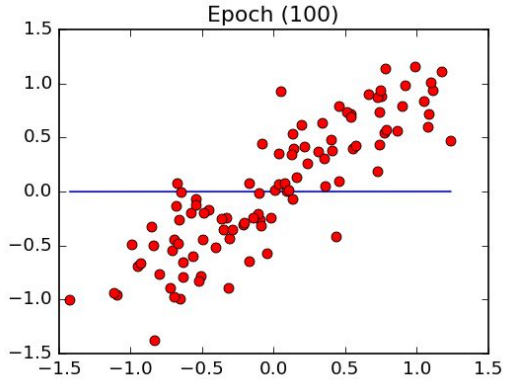
Gradient Descent

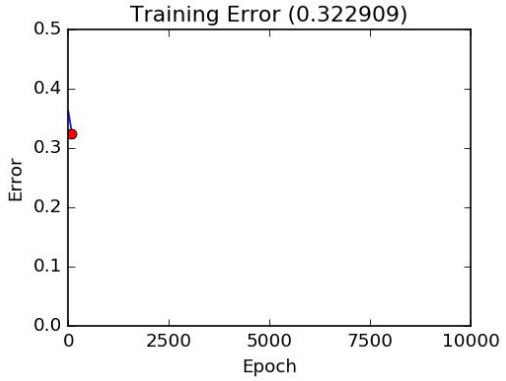


$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$



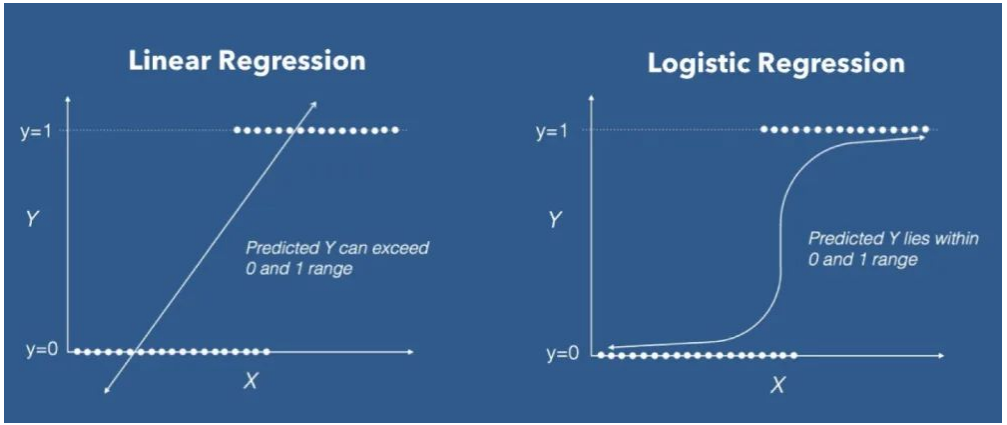
- This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of J.
- Gradient Descent





Classification and logistic regression

- For now, we will focus on the binary classification problem in which y can take on only two values, 0 and 1.
- eg - email- Spam/Not Spam



Why not use linear regression?

use our old linear regression algorithm to try to predict y given x. However, it is easy to construct examples where this method performs very poorly. Intuitively, it also doesn't make sense

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

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

Loss Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

Streamlit

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps

- Interactive
- Turn Data scripts into webapp
- Easy to Deploy
- No frontend Experience Required

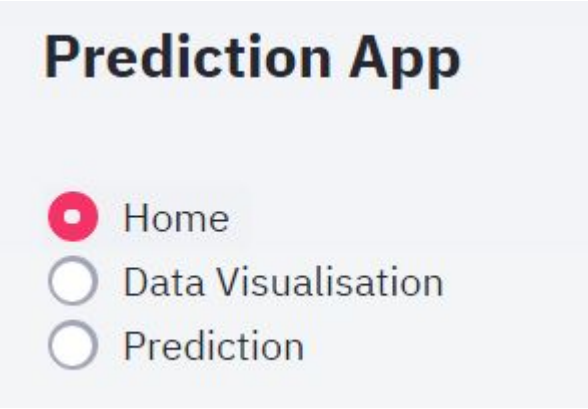
Installing Streamlit

Streamlit can easily be installed with the following command:

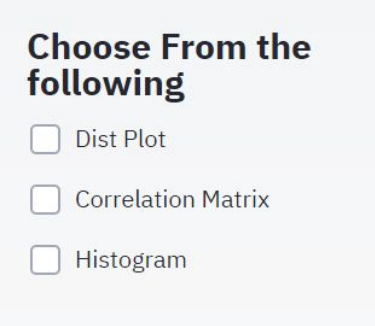
```
pip install streamlit
```

Widgets

Radio Button



Check Box



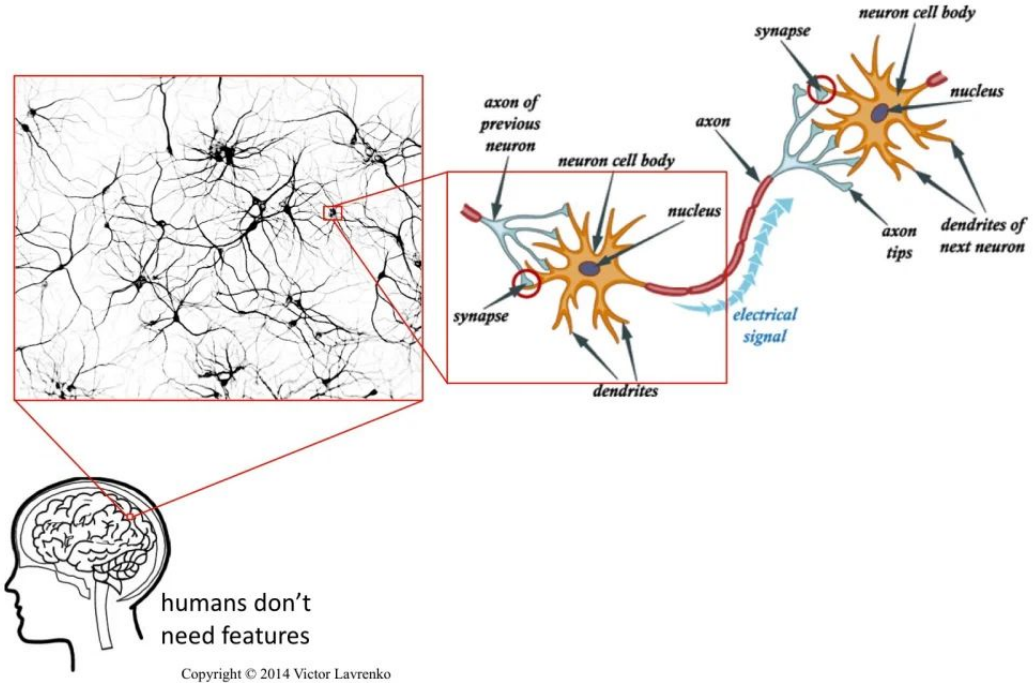
Sliders

NEURAL NETWORKS

Inspiration

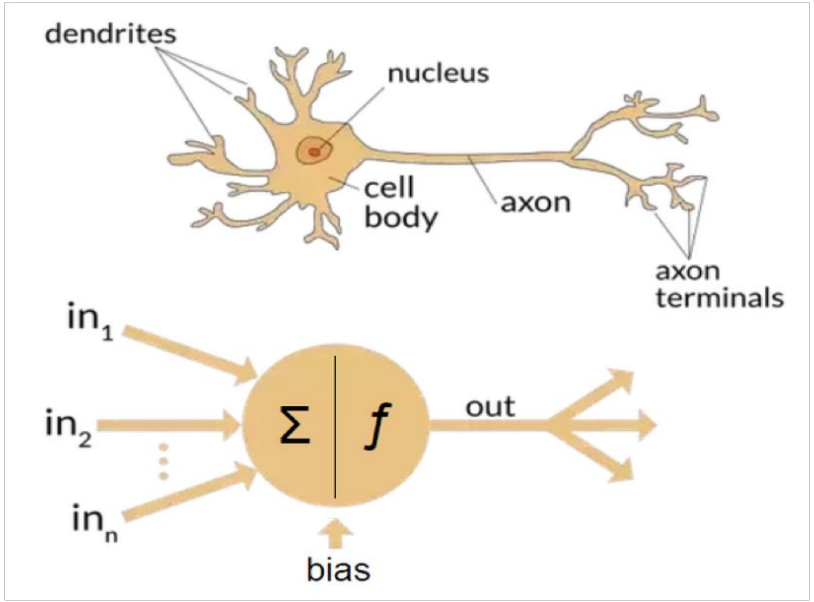
Brain Neurons

Neurons and the brain

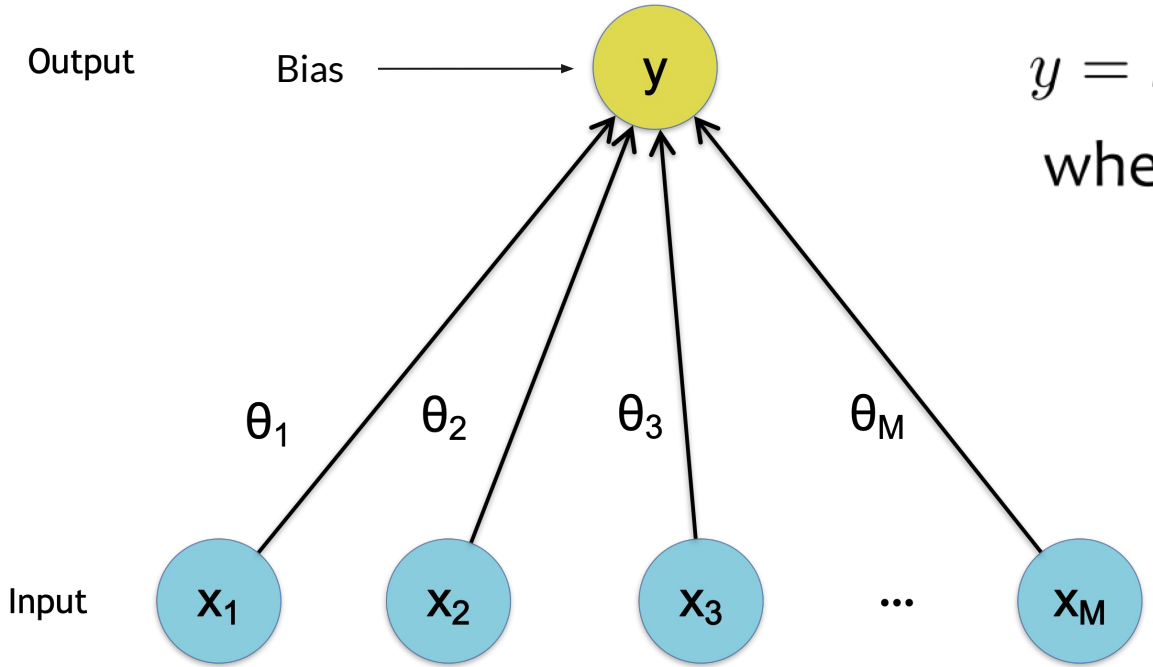


Inspiration

Brain Neurons



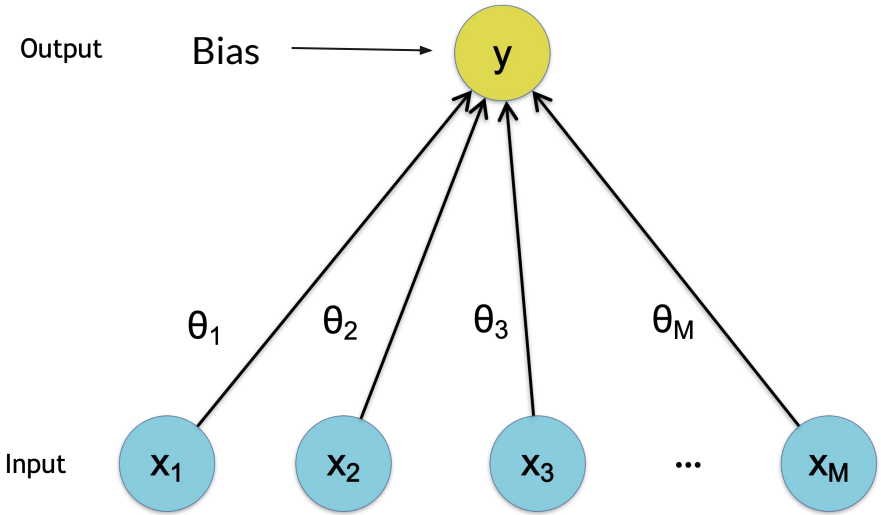
Linear Regression



$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$$

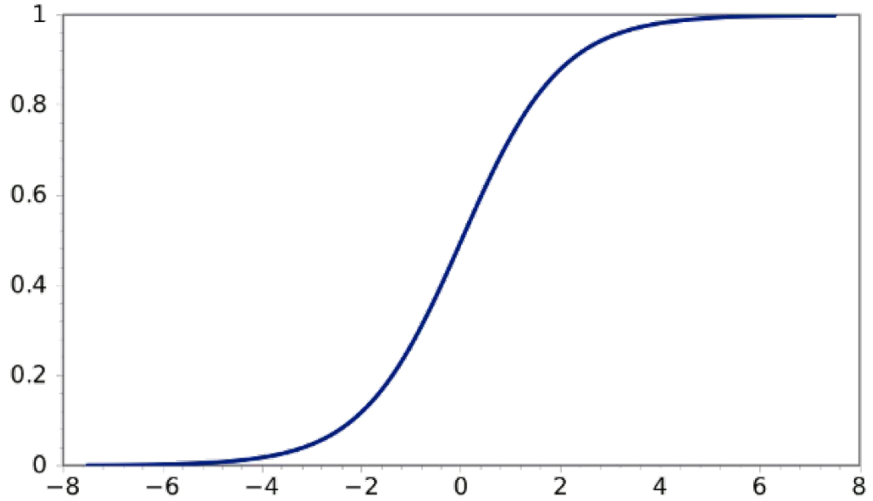
where $\sigma(a) = a$

Logistic Regression

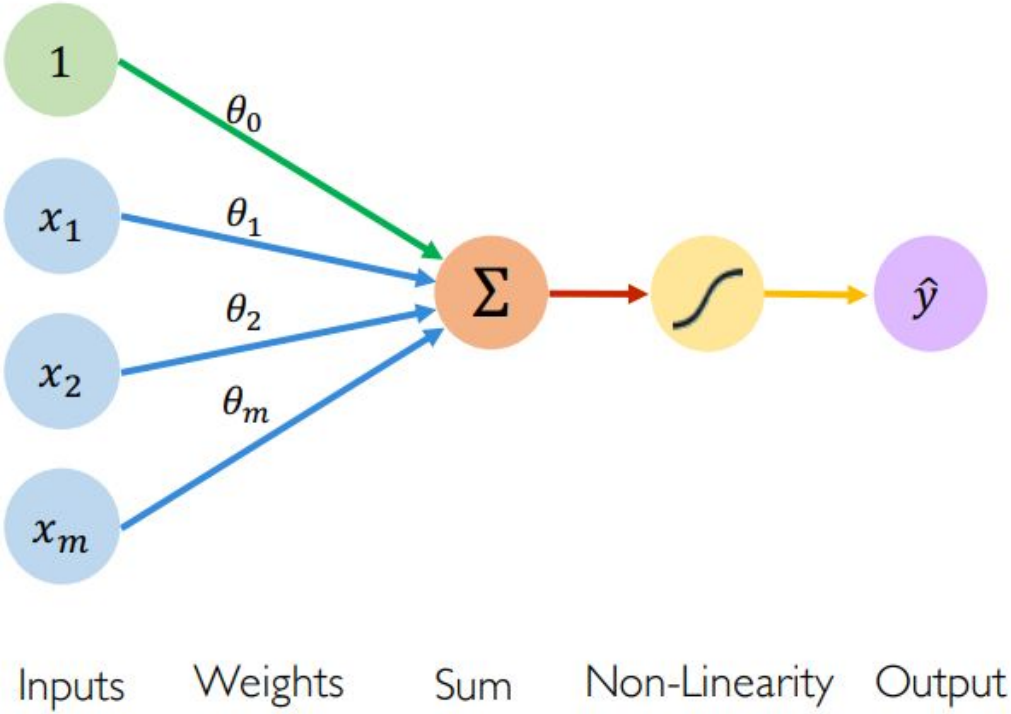


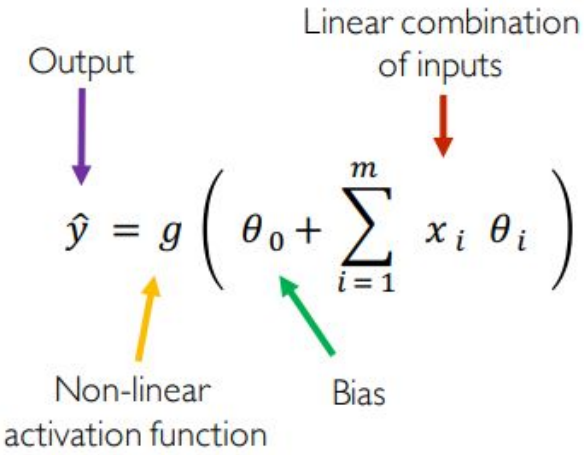
$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$$

$$\text{where } \sigma(a) = \frac{1}{1 + \exp(-a)}$$



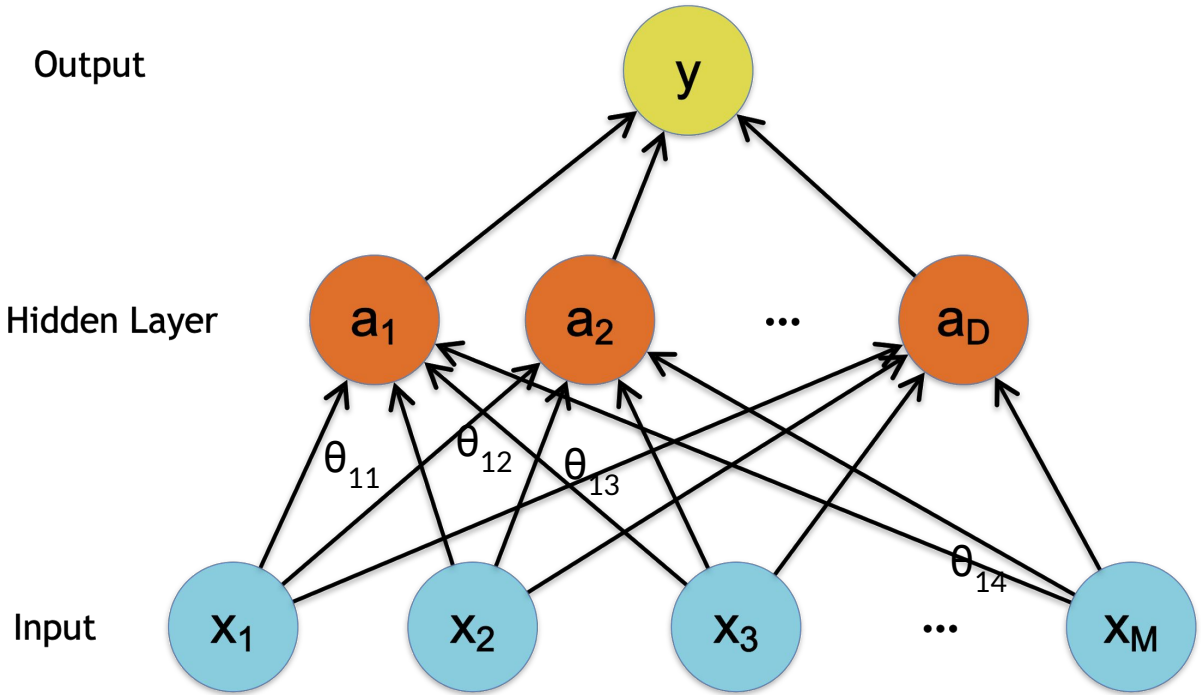
The Neuron

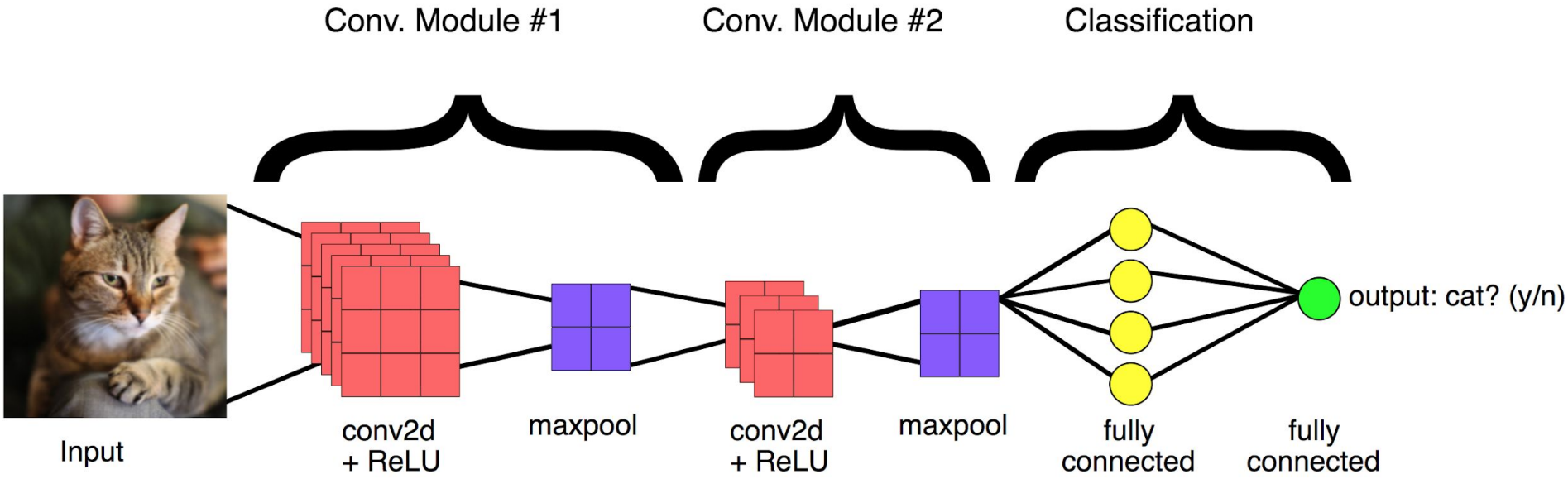

1
x_1
x_2
x_m
θ_0
θ_1
θ_2
θ_m
Σ
$\hat{y}$
Inputs
Weights
Sum
Non-Linearity
Output


Output
Linear combination of inputs
$\hat{y} = g\left(\theta_0 + \sum_{i=1}^{m} x_i\ \theta_i\right)$
Non-linear activation function
Bias

Deep learning

Neural Network




Conv. Module #1
Conv. Module #2
Classification
Input
conv2d
+ ReLU
maxpool
conv2d
+ ReLU
maxpool
fully
connected
fully
connected
output: cat? (y/n)

Echo

an IEEE DTU publication

AN IEEE DTU PUBLICATION

To join IEEE DTU, you can contact any of the following Membership coordinators:

- Ritwik:- 9873388660, 7903449147
- Mridul Gupta:- 9654419074, 9716101491
- Arun:- 7065783844
- Shubham:- 9953569302
- Abhishek :- 9958540744, 7011875112
- Manali:- 9205257150
- Raushan :- 9599627732

For questions regarding today’s session and the upcoming SIGs (exclusively for IEEE members), contact (us):

- Sakshi:- 8586016510
- Pranay:- 8076699075
- Jatin:- 9650408541
- Priyansh:- 8826277763
- Varang:- 8800566387


Kahoot!

SOME BASIC INSTRUCTIONS

- The top 3 position holders are requested to take a screenshot of their results screen and share it as a story on Instagram.
- Don't forget to tag @ieee.dtu in your Instagram Stories in order to claim your prize.

Prize Reveal

- 1st Position Holder : 90% off Coding Blocks Course Coupon
- 2nd Position Holder : 75% off Coding Blocks Course Coupon
- 3rd Position Holder : 50% off Coding Blocks Course Coupon
- Everyone gets access to Course Discount Coupons from Edudictive

THANK YOU SO MUCH

Contact the Membership Coordinators

Ritwik : wa.me/919873388660
Arun : wa.me/917065783844
Shubham : wa.me/919953569302
Mridul Gupta : wa.me/919654419074
Abhishek : wa.me/919958540744
Manali : wa.me/919205257150
Raushan : wa.me/919599627732

IEEE DTU
Student Branch

FEEDBACK FORM: bit.ly/ml_feedback