SUMMER SCHOOL 2020

MACHINE LEARNING DAY 2

DEEP LEARNING

Session I: Introduction to DL

Isaac Ye, HPTC @ York University

Isaac@sharcnet.ca

Schedule

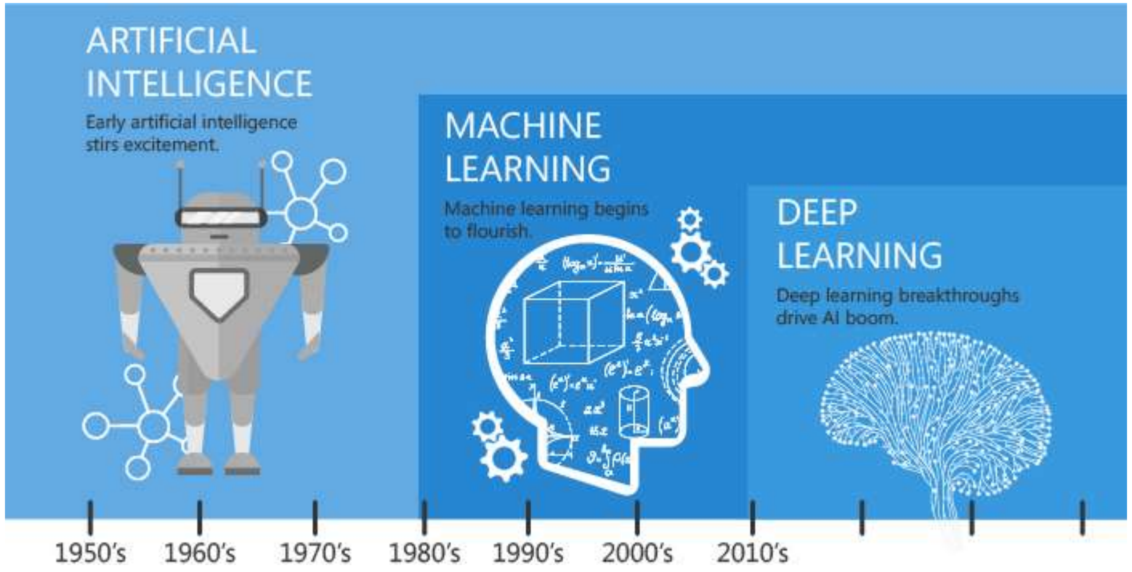
Session	Time	Topic	Hands-on
1	10:00 - 10:55	Introduction to Deep Learning Single variable linear regression problem	Lab 1
2	11:00 - 11:55	Multi-variable linear regression problem	Lab 2A, 2B
3	13:30 - 15:15	Multi-layer perceptron	Lab 3A, 3B
4	15:30 - 17:00	Convolutional neural network	Lab 4A, 4B

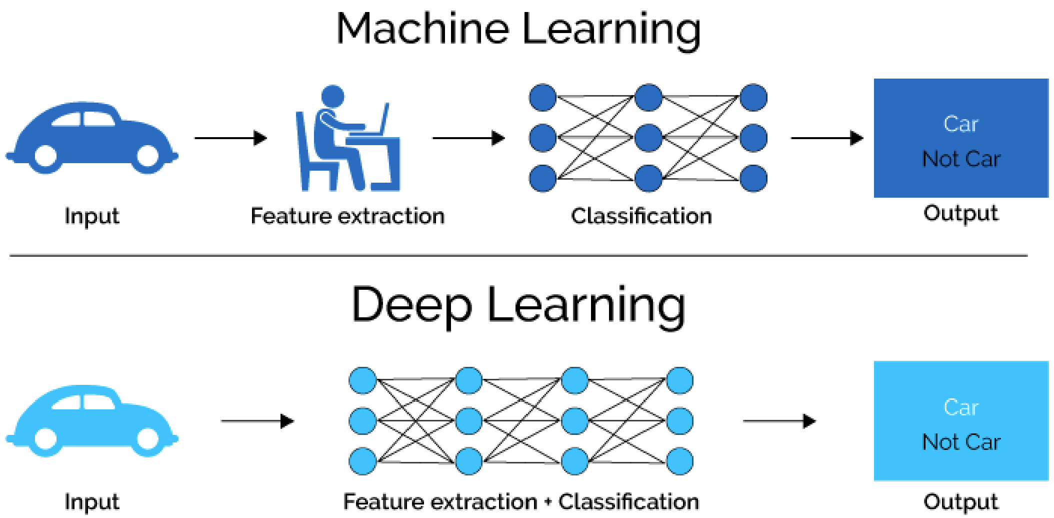
Session I

- What is AI/Deep Learning (DL)
- DL frameworks
- Categories of ML problems
- Linear regression (single variable)
- Cost function / gradient decent algorithm / learning rate
- *Lab 1: Single variable linear regression (vanilla)*

AI / Deep Learning

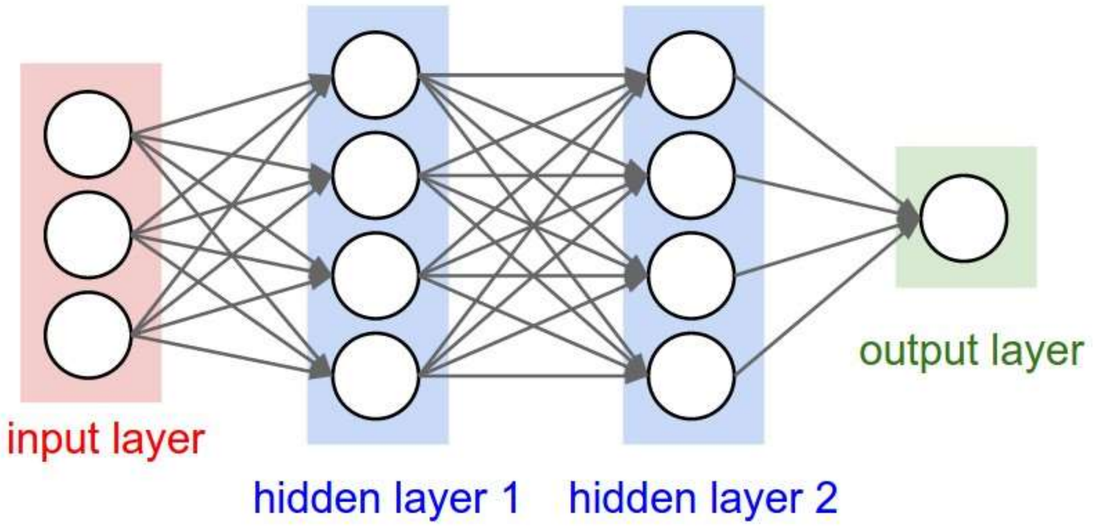
Deep learning (DL) is a class of machine learning algorithms in which multiple layers of nonlinear processing units are used for feature extraction and transformation, with each successive layer taking the output from the previous layer as input.





DL: Deep Neural Network (DNN)

“A family of parametric, non-linear and hierarchical representation learning functions, which are massively optimized with stochastic gradient descent to encode domain knowledge, i.e. domain invariances, stationarity.” -- Efstratios Gavves

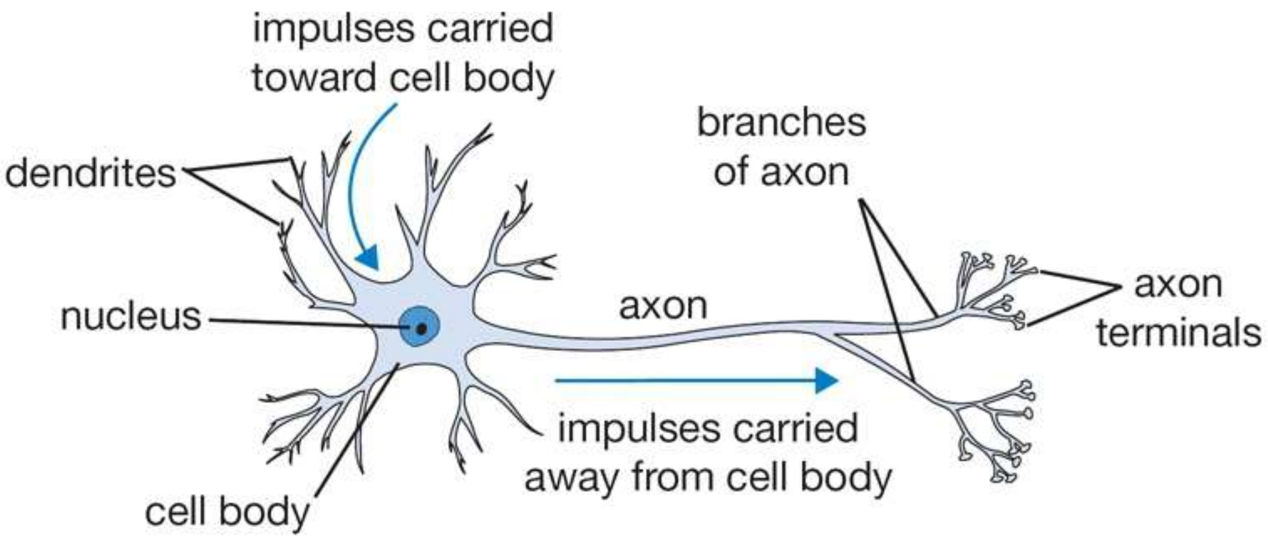


Example of a 3-layer Deep Neural Network (DNN)

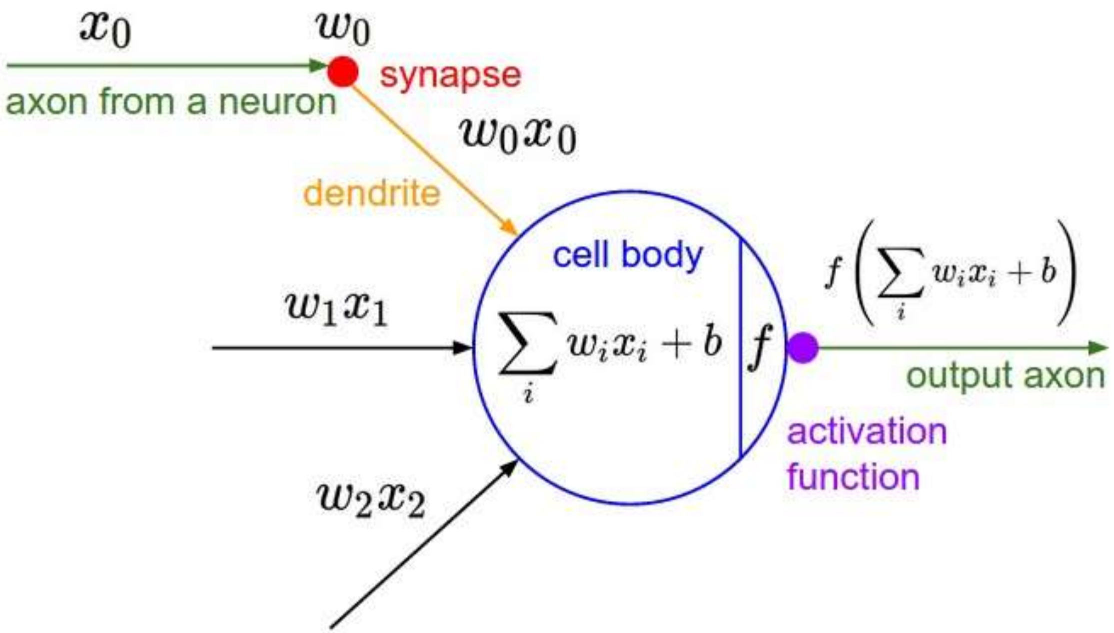
http://cs231n.github.io/neural-networks-1/

Neural Network

http://cs231n.github.io/neural-networks-1/

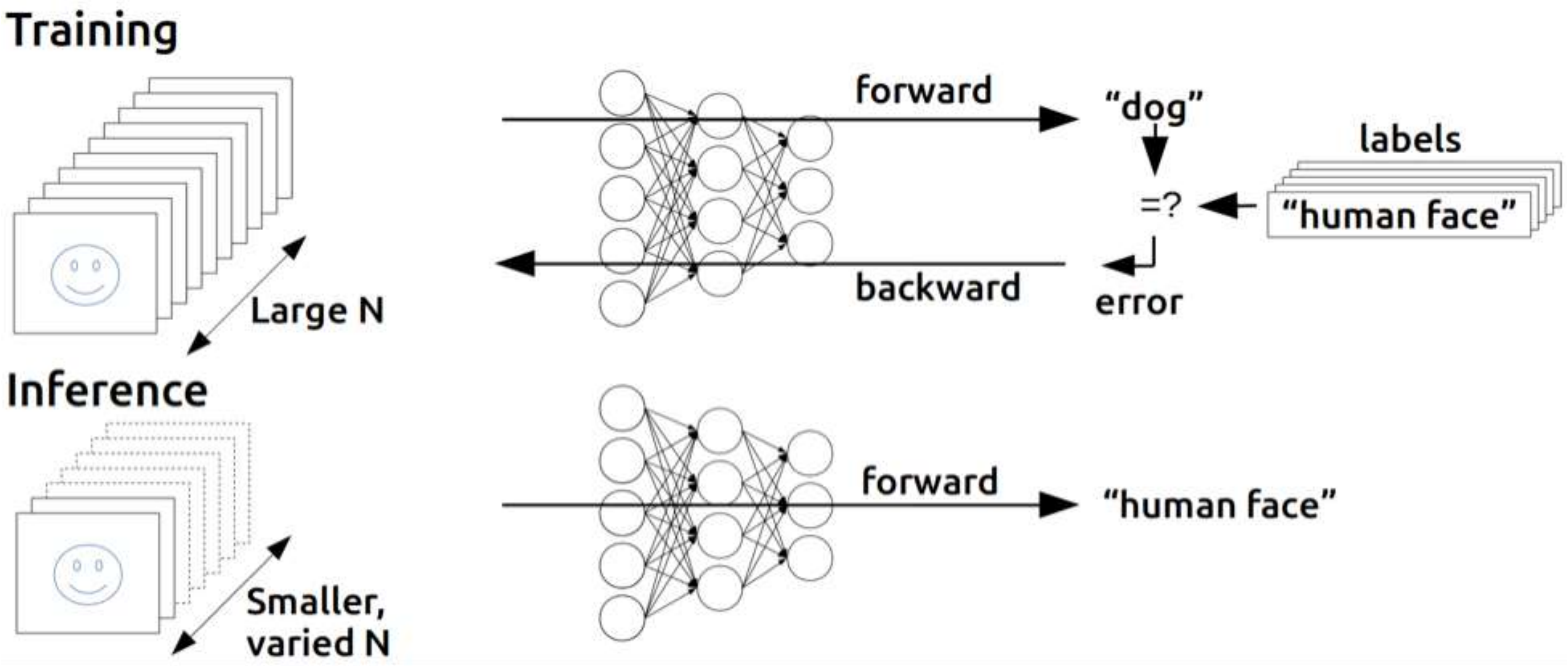


Biological neuron



Mathematical model

DL: Training / Inference



https://devblogs.nvidia.com/inference-next-step-gpu-accelerated-deep-learning/

DL frameworks

Caffe

Microsoft Cognitive Toolkit

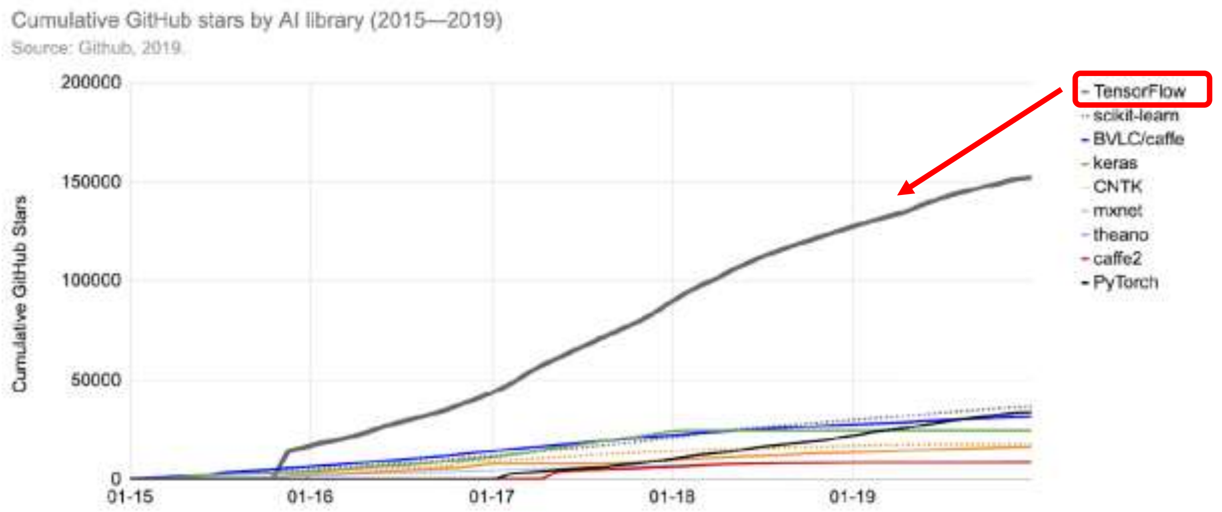
TensorFlow

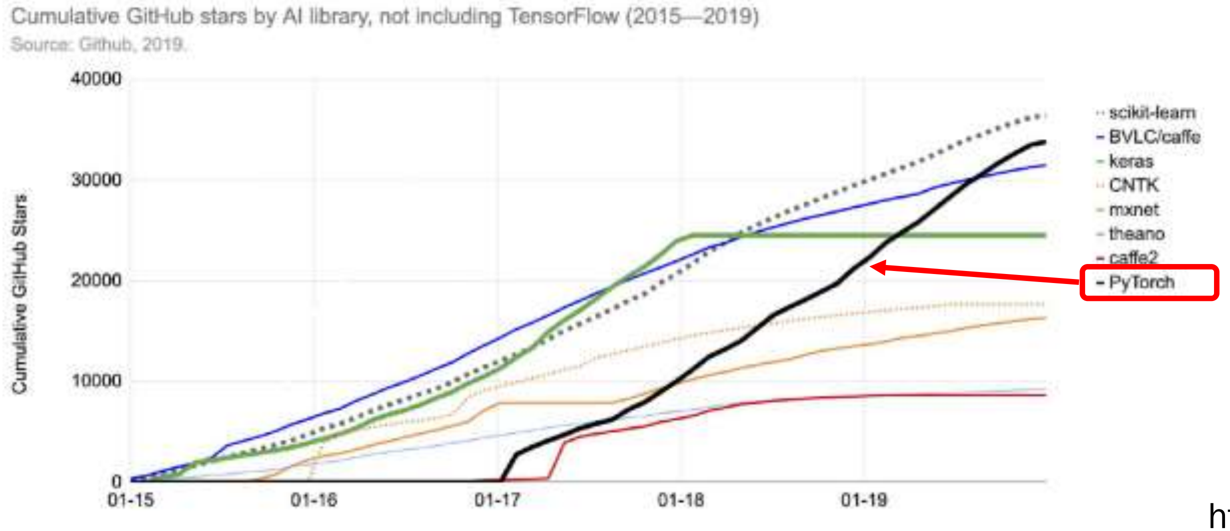
PyTorch

mxnet

julia

DL frameworks trend





https://aiindex.org

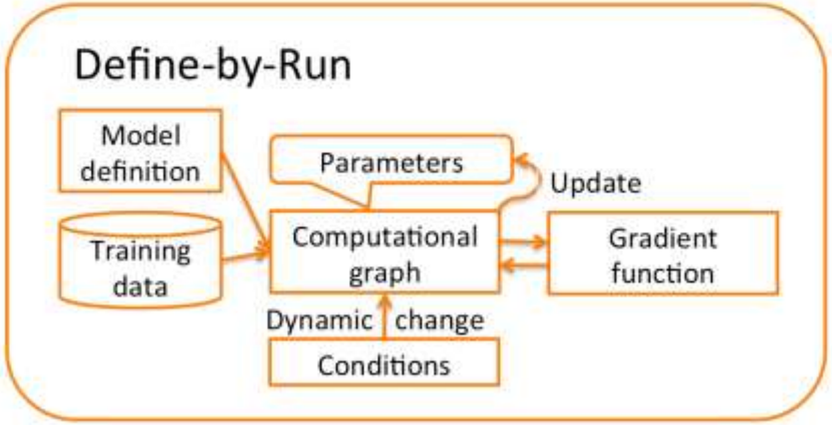
TensorFlow

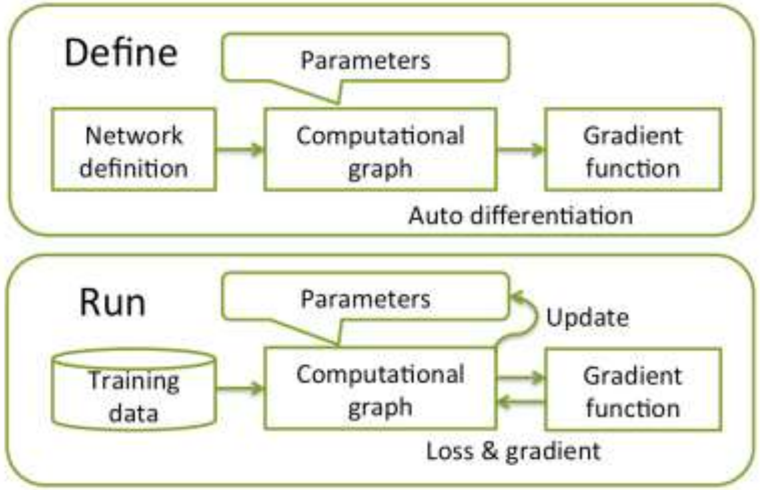
- The most widely used framework open-sourced by Google
- Replaced Google's DistBelief framework
- Runs on almost all architectures (CPU/GPU/TPU/etc)
- Define-and-Run type for neural networks
- Version 2.0 has Define-by-Run component(Eager execution)
- https://github.com/tensorflow/tensorflow

PyTorch

- Rapidly growing in research community for deep learning
  framework developed by Facebook
- A Python adaptation of Torch
- Caffe2 has been merged to PyTorch
- Define-by-Run type for neural networks
- Ease of expression and use
- https://github.com/pytorch/pytorch

	PyTorch	TensorFlow
Pros	Easy to use (Python support) Intuitive Dynamic graphs Research community prefers	Large community Heterogeneous architecture TF 2.0: Eager execution(Define-by-Run) Tensorboard (visualizing), Keras
Cons	Small community Less additional tools	Verbose Static graphs





https://www.oreilly.com/content/complex-neural-networks-made-easy-by-chainer/

Categories of ML problems

	Supervised	Unsupervised	Reinforcement
Discrete	Classification	Clustering	Action space agent
Continuous	Regression	Dimensionality reduction	Action space agent

Supervised learning

	Supervised
Discrete	**Classification**
Continuous	**Regression**

Supervised

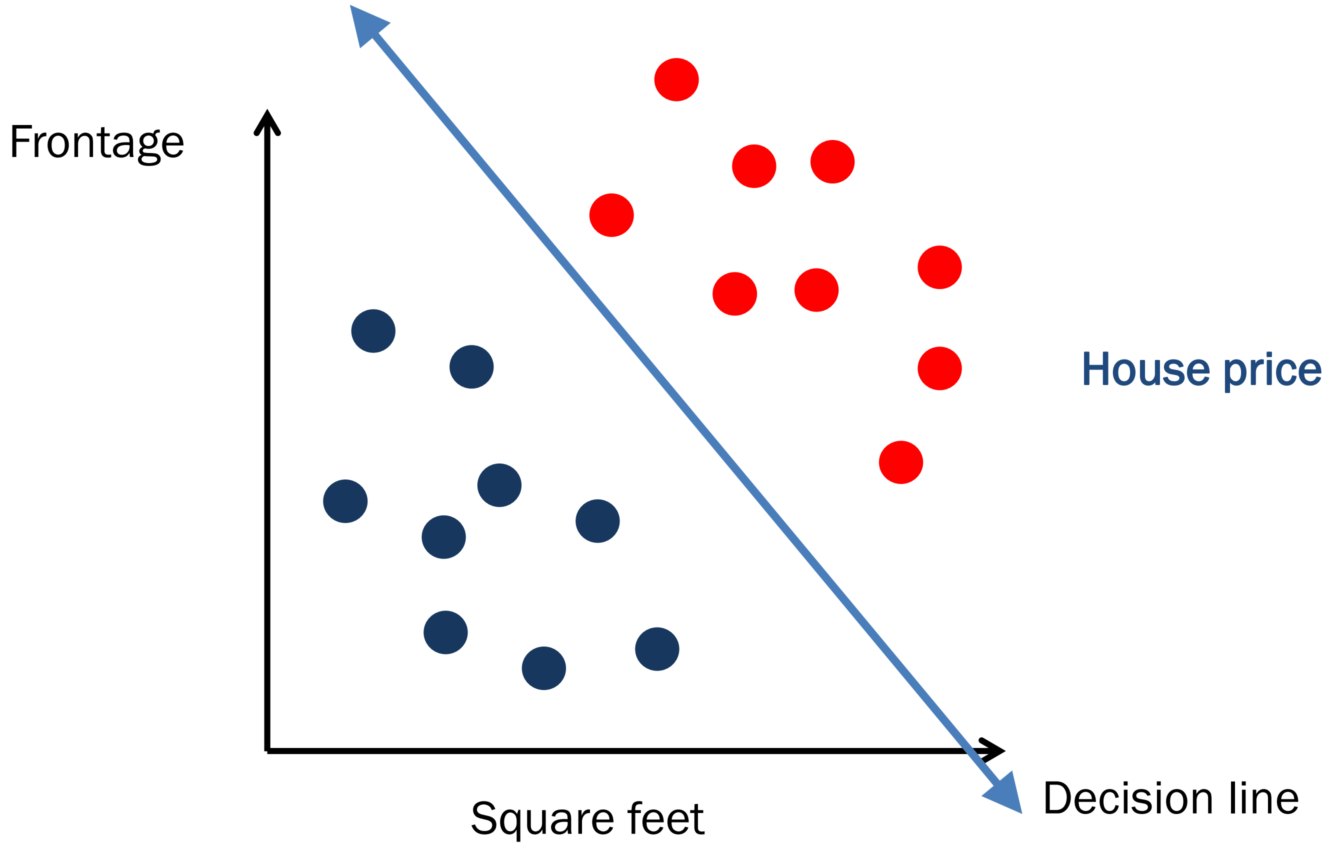
Input	Data (x, y) x: data, y: label
Goal	learn a function to map x to y
Examples	classification, regression, object detection semantic segmentation, image captioning

Supervised learning

Supervised

Input	Data (x, y) x: data, y: label
Goal	learn a function to map x to y
Examples	classification, regression, object detection semantic segmentation, image captioning

Label: Red(> $1M), Blue(< $1M)

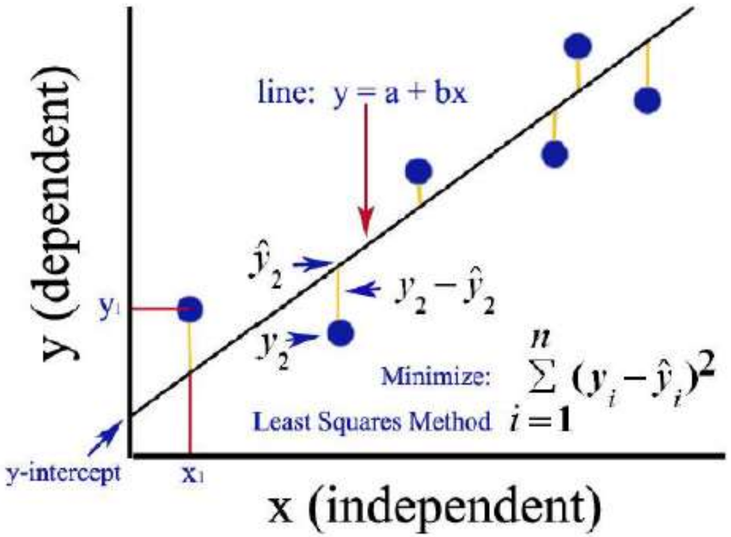


Categories of ML problems

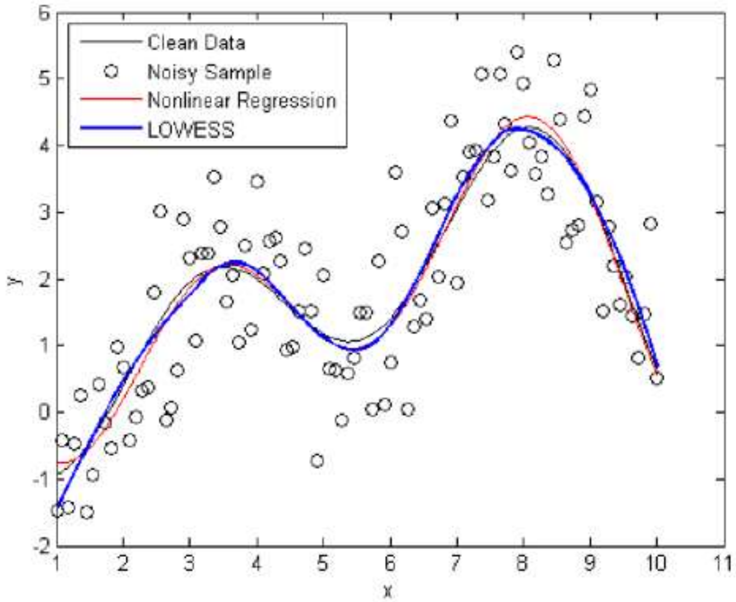
	Supervised	Unsupervised	Reinforcement
Discrete	Classification	Clustering	Action space agent
Continuous	Regression	Dimensionality reduction	Action space agent

Regression problem

Fit the prediction function f(x) to the training data to predict continuous real value



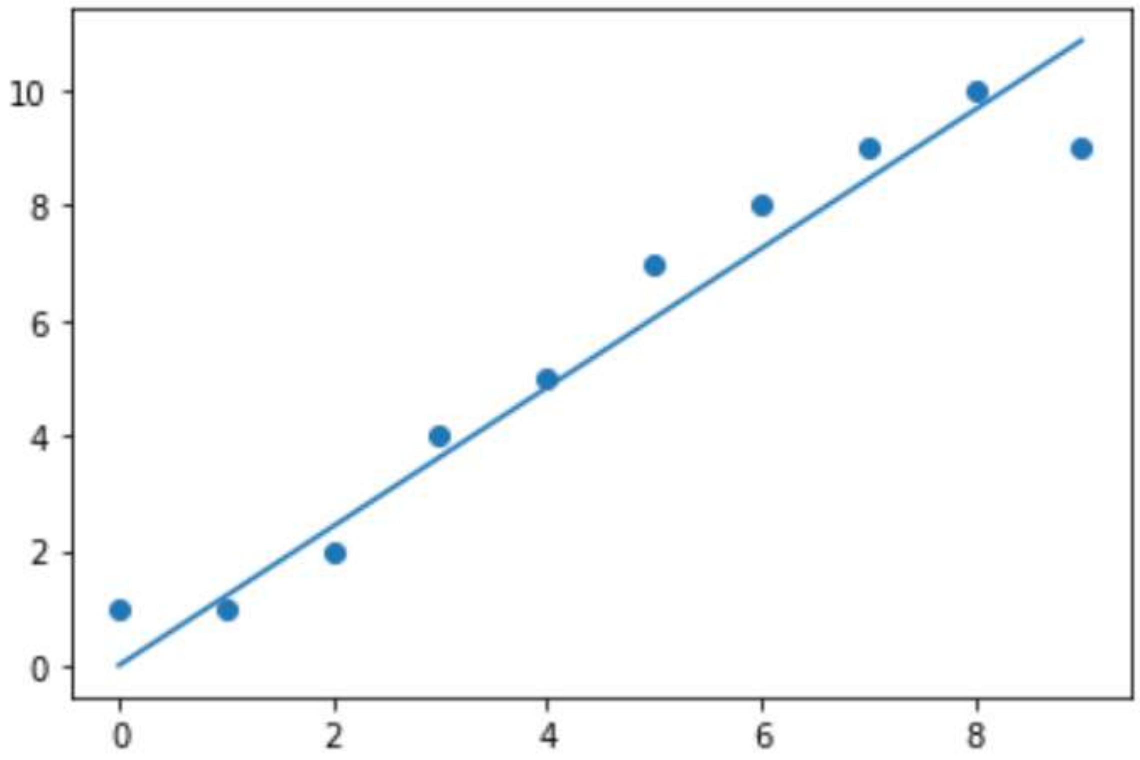
Linear regression



Nonlinear regression

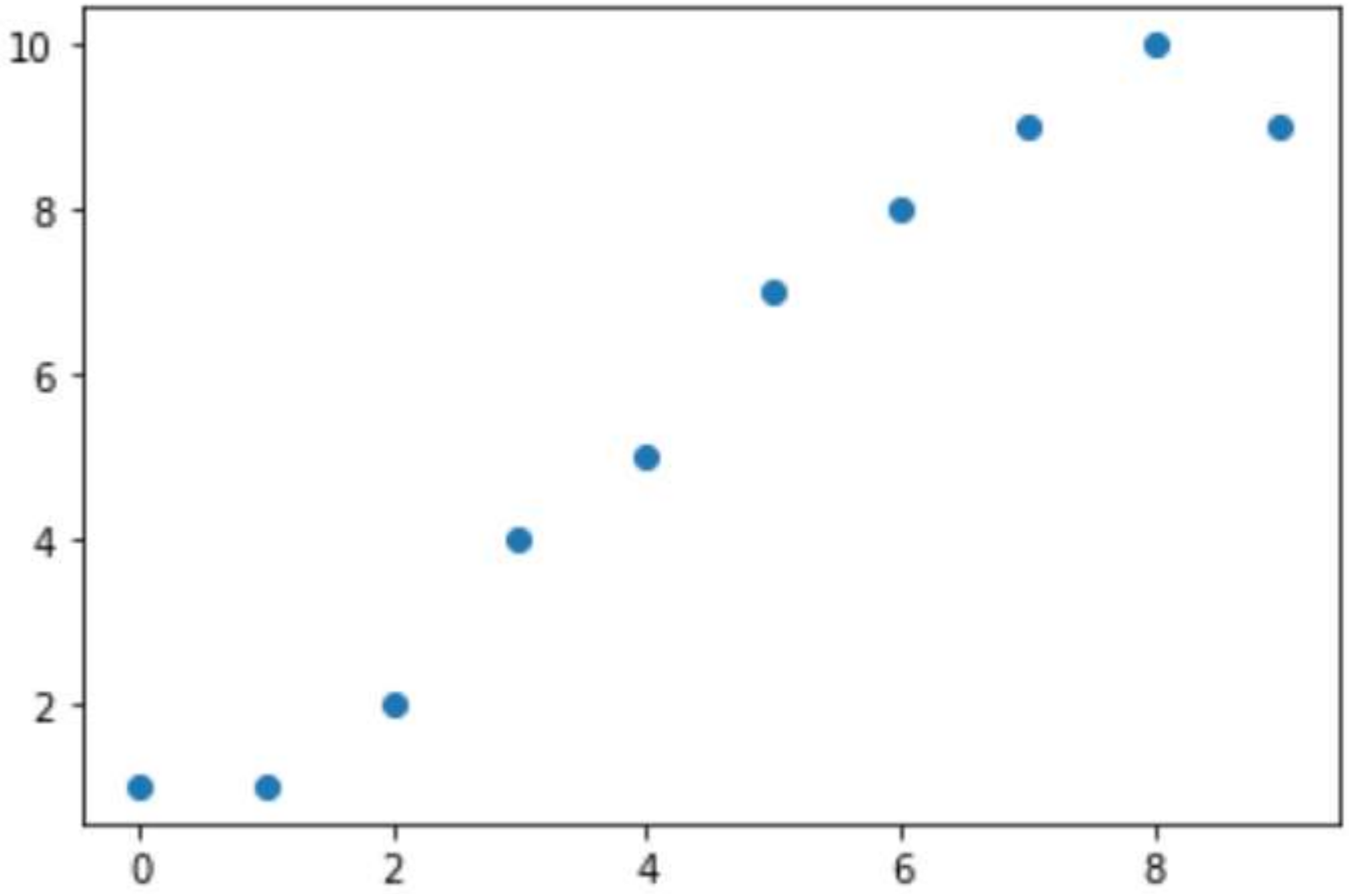
Linear regression : single variable

x	y
0	1
1	1
2	2
3	4
4	5
5	7
6	8
7	9
8	10
9	9

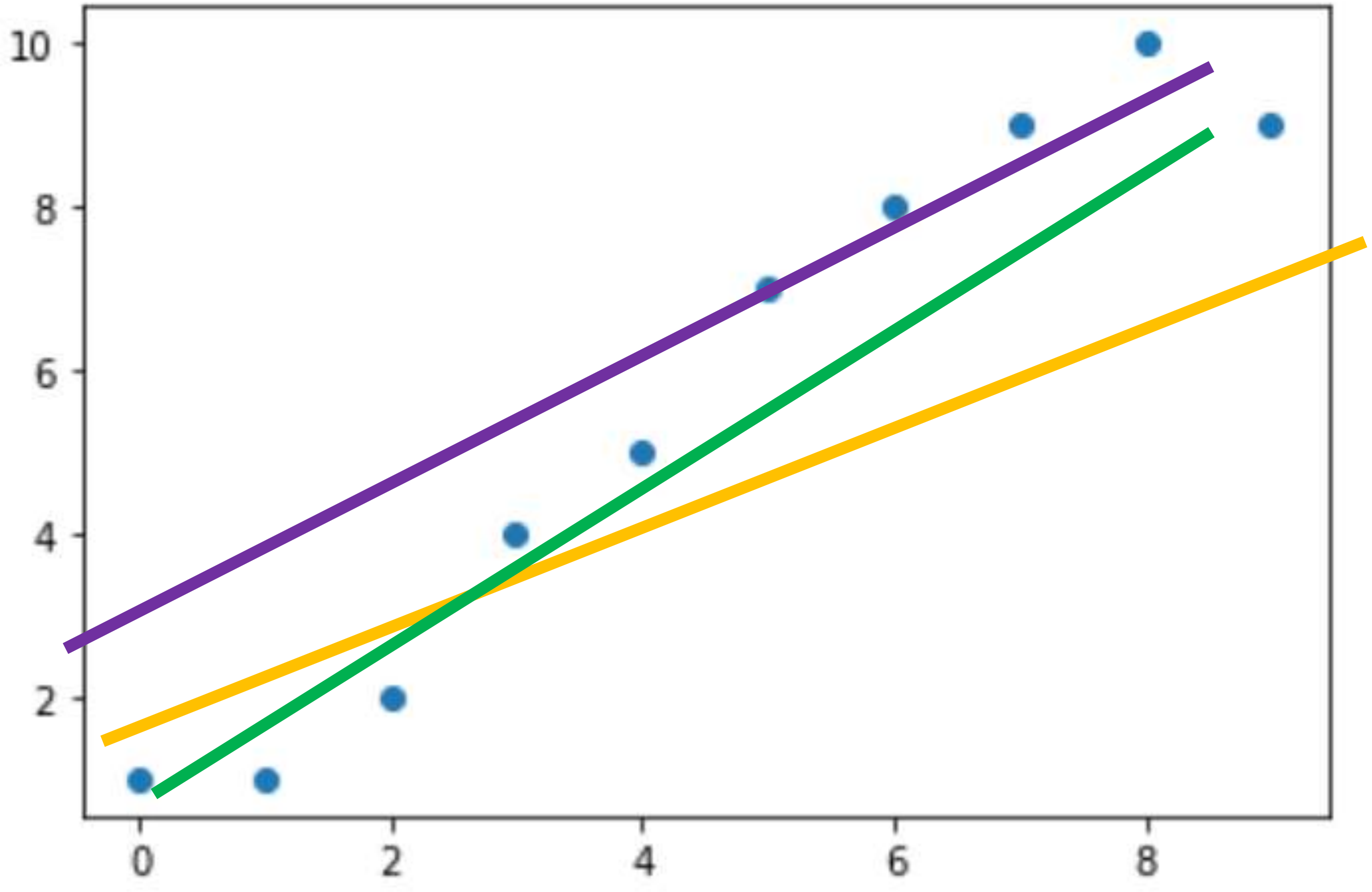


Data preparation: Input (x, y)

x	y
0	1
1	1
2	2
3	4
4	5
5	7
6	8
7	9
8	10
9	9

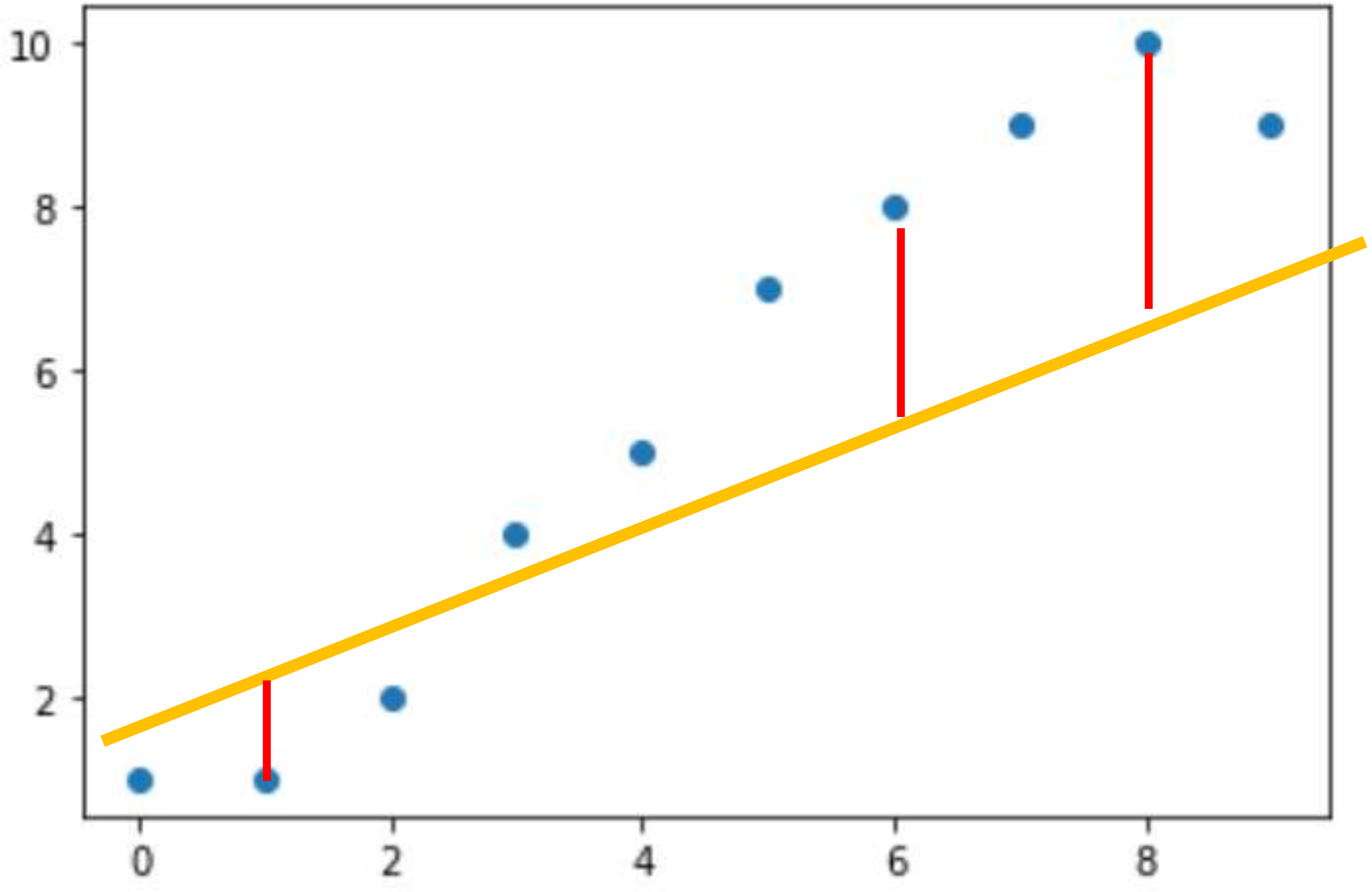


Model (Hypothesis)



$$H(x) = Wx + b$$

Which model is better?



How well fit the line to data?

$$\underbrace{H(x)}_{\text{Predicted}} - \underbrace{y}_{\text{True}}$$

Cost function

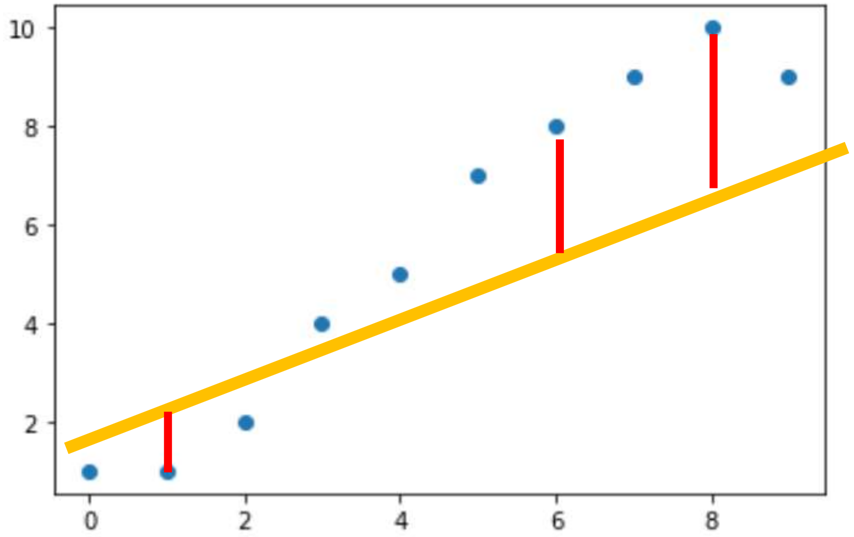
Model

$$H(x) = Wx + b$$

Mean Square Error

$$cost = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2$$

m is the number of data.



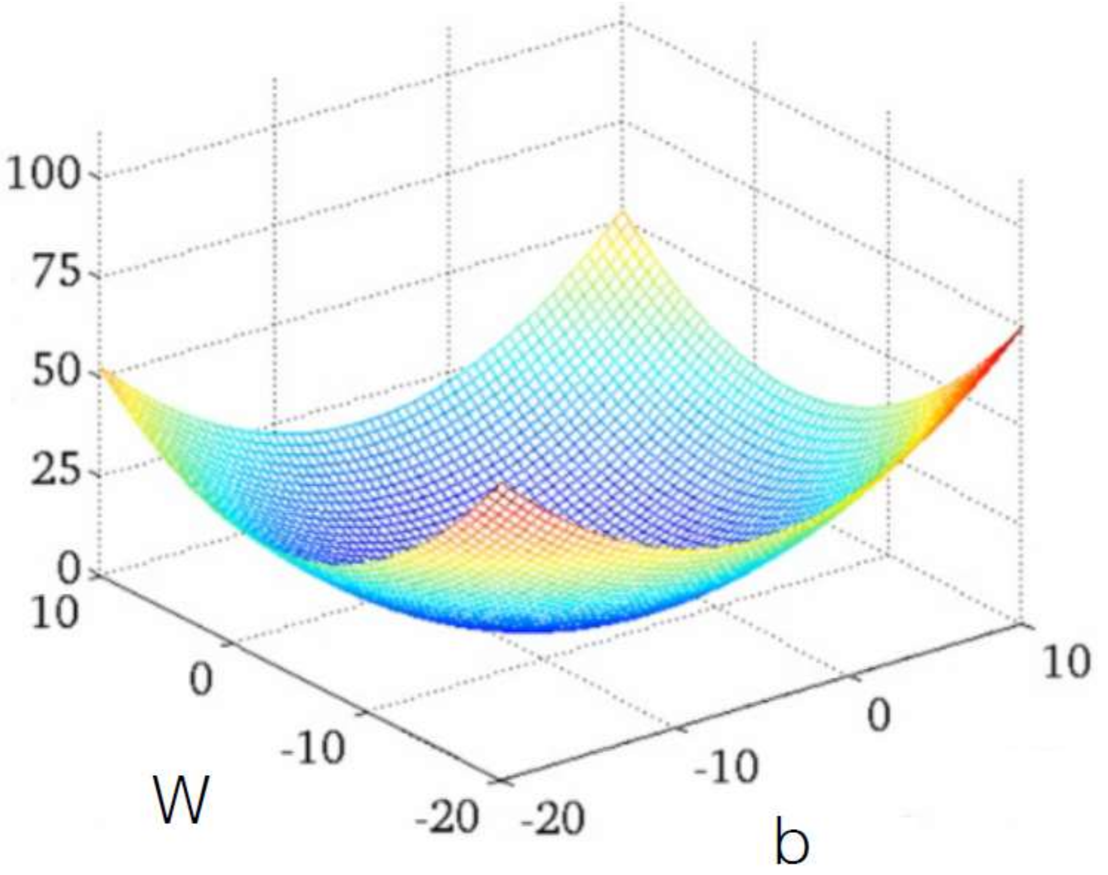
Now we can see the cost function as a function of W and b.

$$cost(W, b) = \frac{1}{m}\sum_{i=1}^{m}\left((Wx_i + b) - y_i\right)^2$$

Cost function: what we want?

We want to minimize the cost!

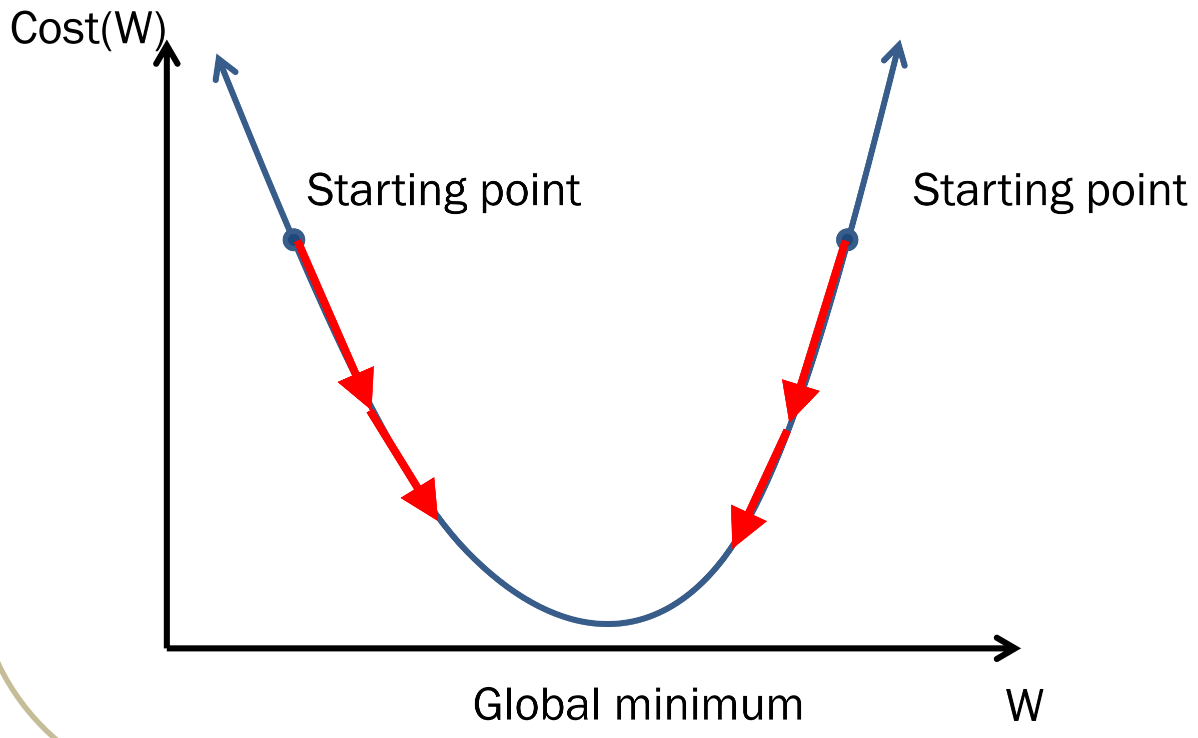
$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}\left((Wx_i + b) - y_i\right)^2$$



http://www.holehouse.org/mlclass

Gradient decent algorithm

Let's consider a simple case with W only.



$$cost(W) = \frac{1}{2m}\sum_{i=1}^{m}(Wx_i - y_i)^2$$

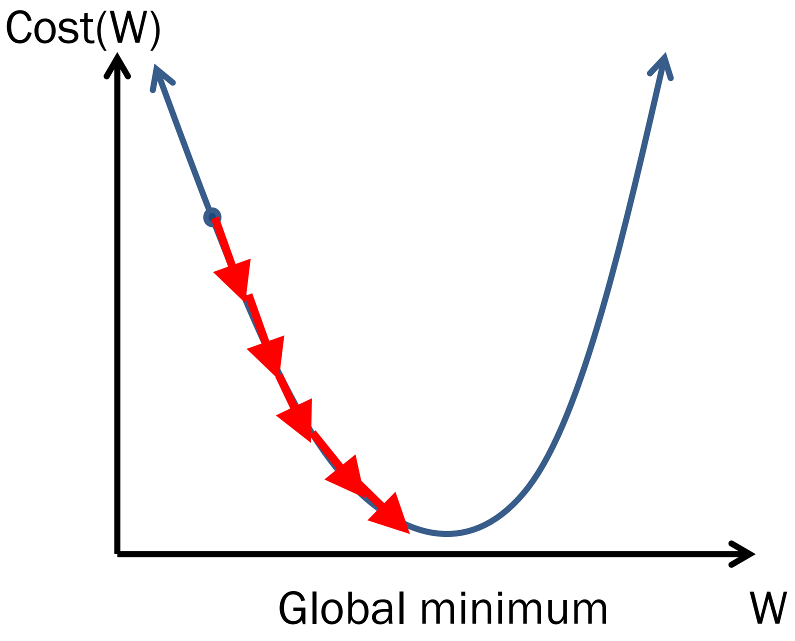
$$\frac{d}{dW}cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx_i - y_i)x_i$$

$$\underbrace{W}_{\text{New}} := W - \underbrace{\alpha}_{\text{Learning rate}}\frac{d}{dW}cost(W)$$

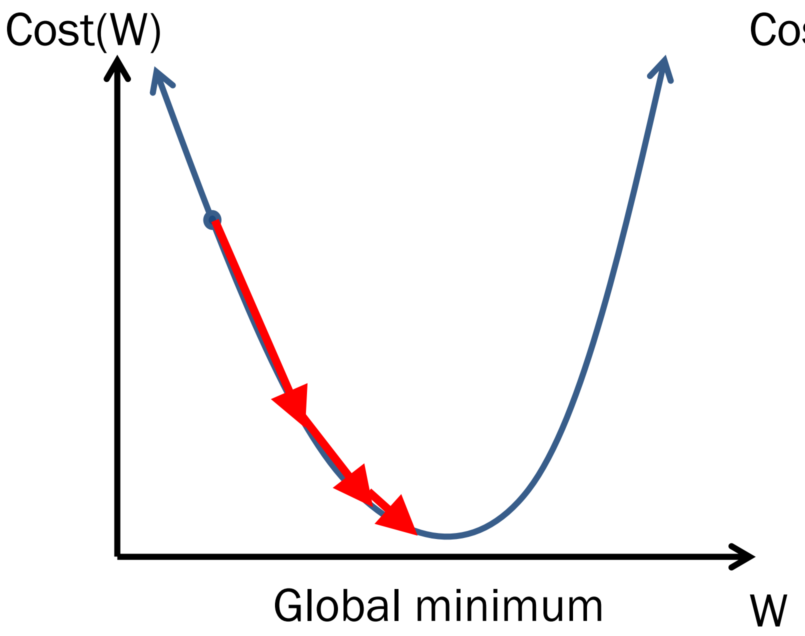
Learning rate

$$W := W - \alpha \frac{d}{dW} cost(W)$$

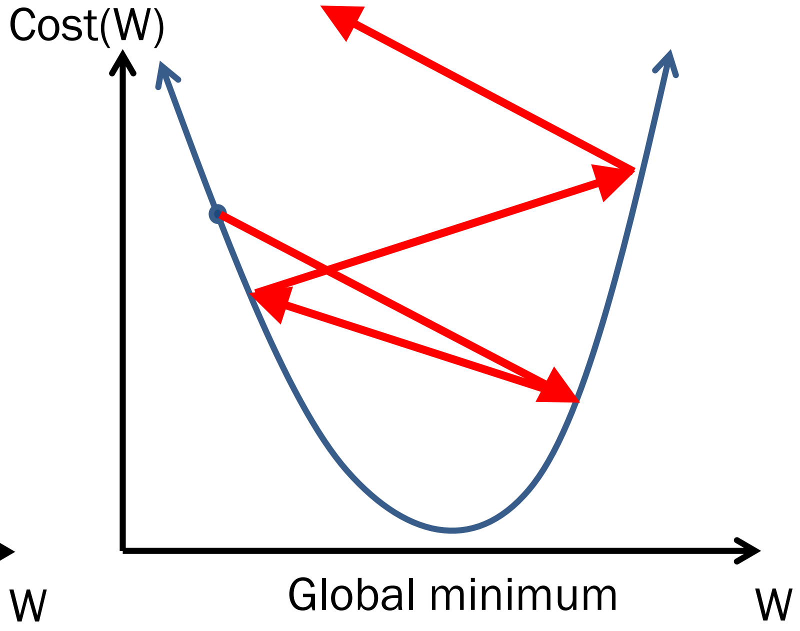
Learning rate



A small learning rate requires many steps



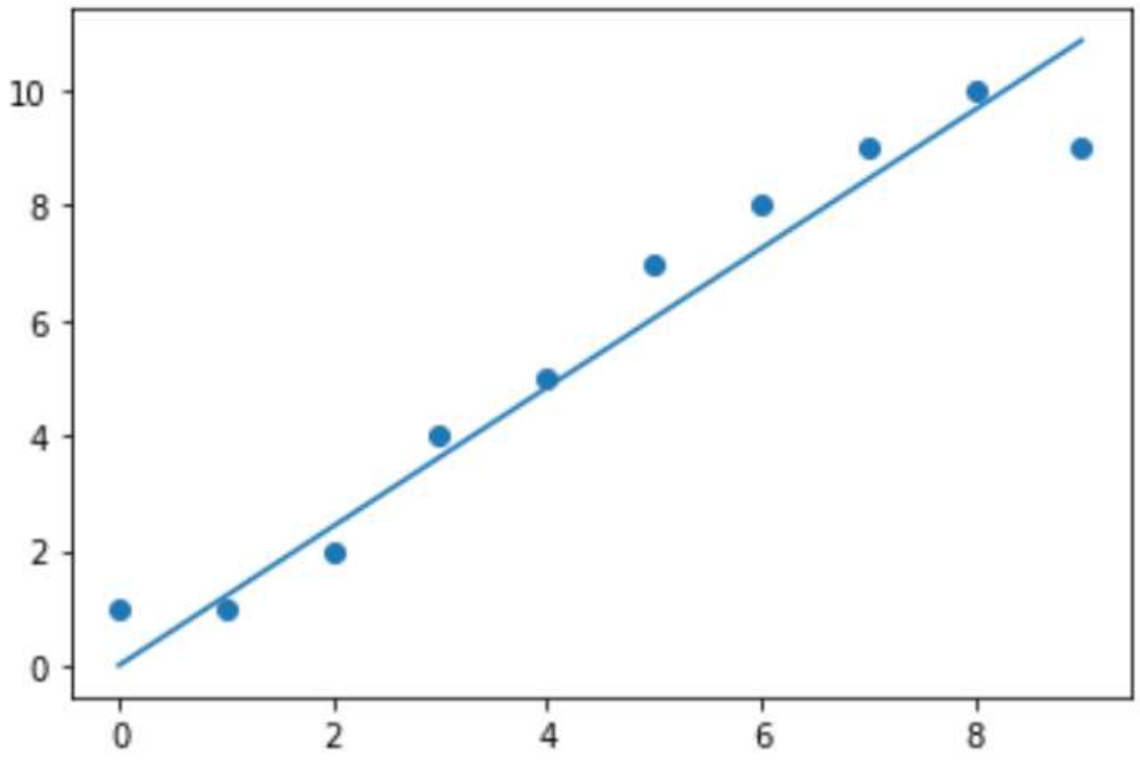
The optimal learning rate (possibly adaptive value)



Too large learning rate causes divergence

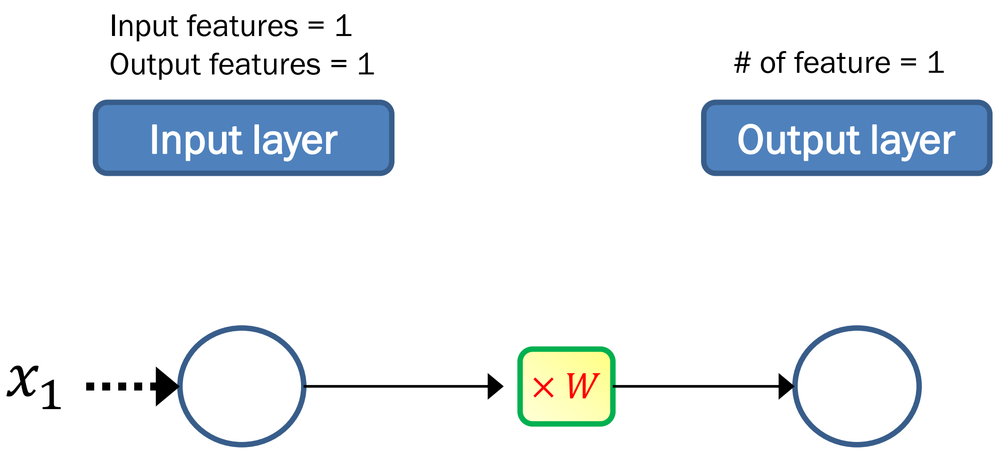
Lab 1: Linear regression (single variable)

x	y
0	1
1	1
2	2
3	4
4	5
5	7
6	8
7	9
8	10
9	9



Layout

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_9 \\ x_{10} \end{bmatrix}$$



$$H(x_i) = Wx_i$$

Let's consider a simple case with W only.

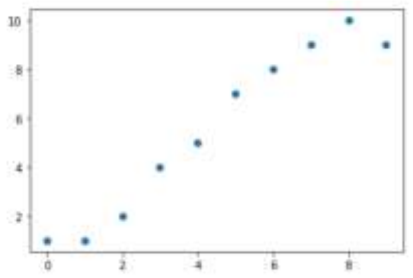
Algorithm structure

Data Preparation

Model define

Cost function + optimizer

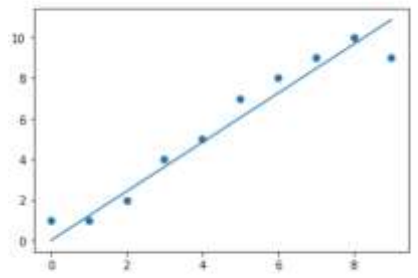
Model Test



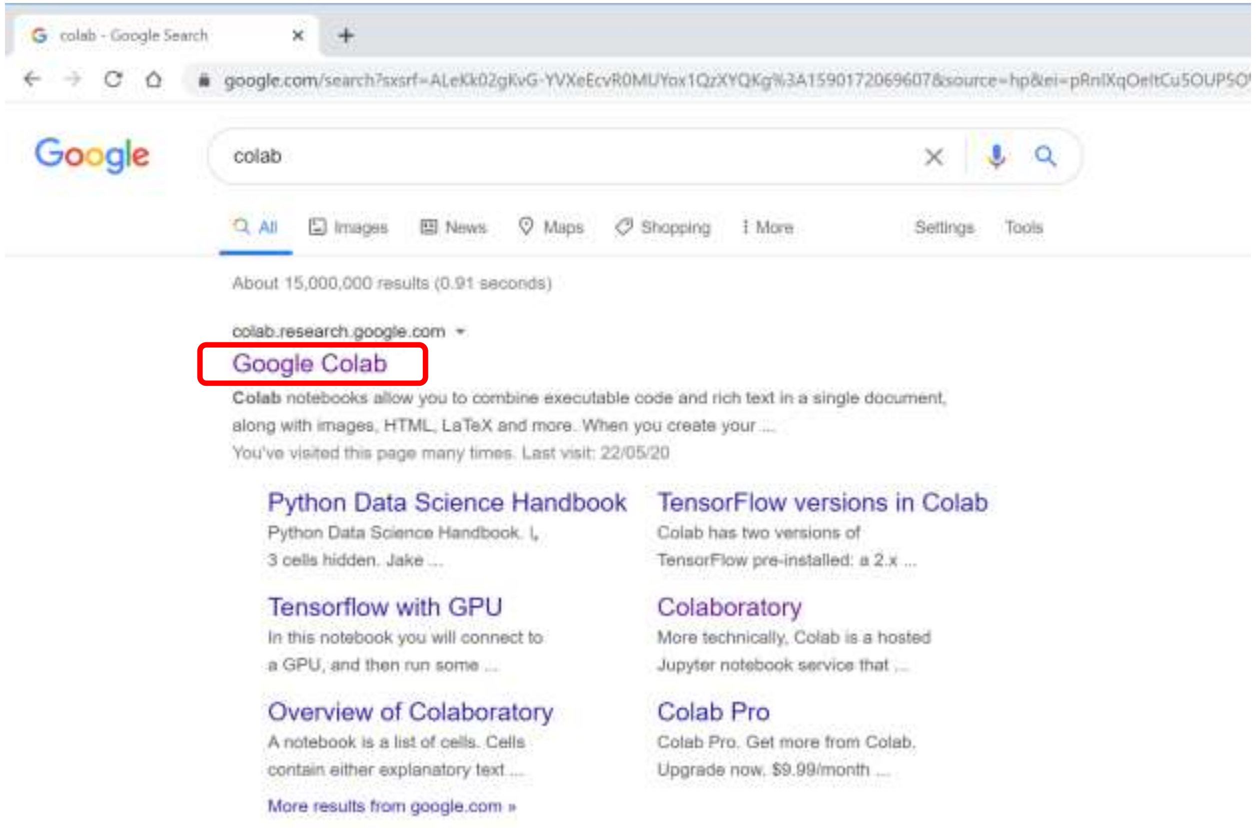
$$H(x) = Wx + b$$

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2$$

$$W := W - \alpha \frac{d}{dW} cost(W)$$



Lab 1: Working environment (Google Colab)



1. Go to https://colab.research.google.com
2. Open a new Jupyter notebook
3. Check Runtime type (GPU/TPU) and settings
4. Editor (Code/text block)
5. Be careful of running order
6. Make sure where you can find your code

Lab I: Linear regression - vanilla

Github:

https://github.com/isaacye/SS2020V2_ML_Day2/Session_1

What you may want to try :

1. Check the model define
2. Check the result by changing the starting point
3. Check the result by changing learning rate

Break room

Session break:

Please come back by 11:00 AM