Artificial Intelligence and Machine Learning

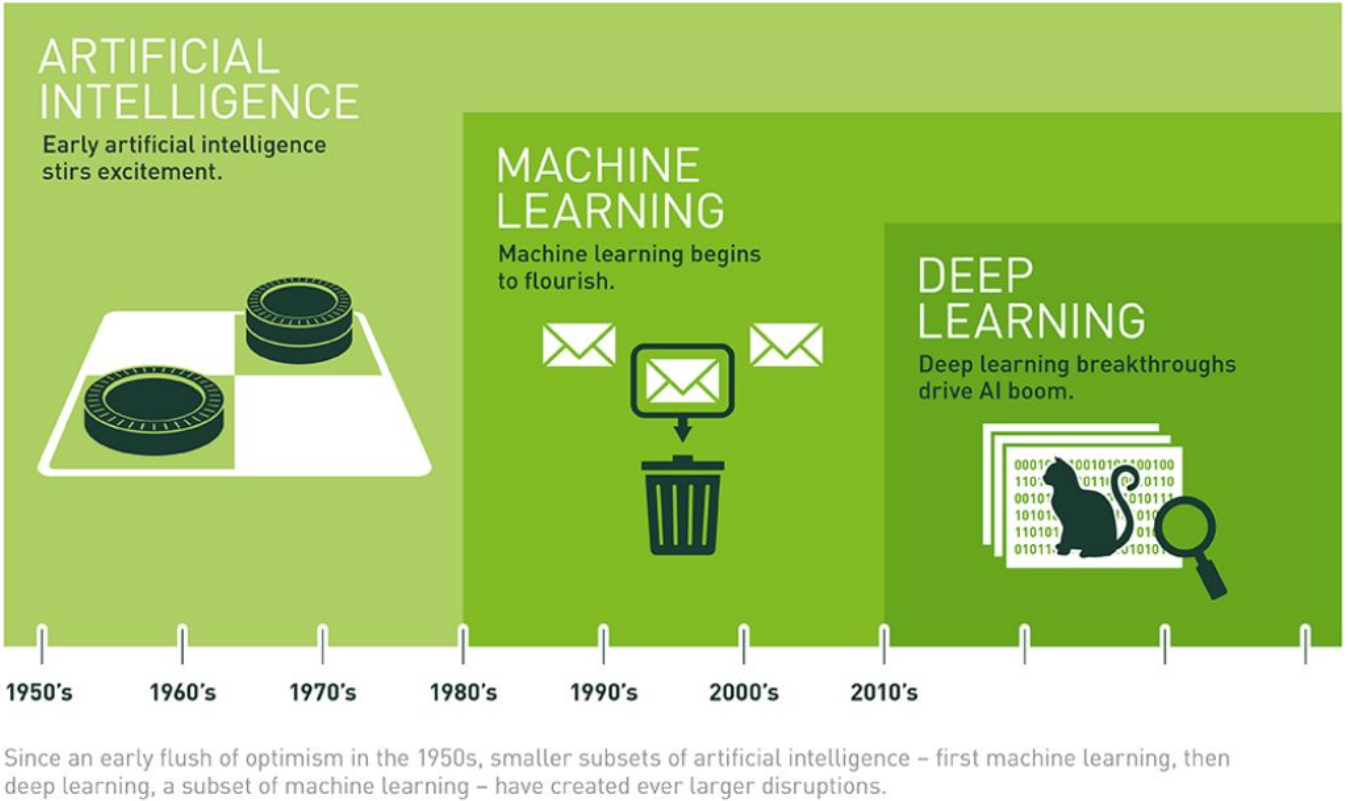
Cheng-Kuan Ou

X-Village

Outline

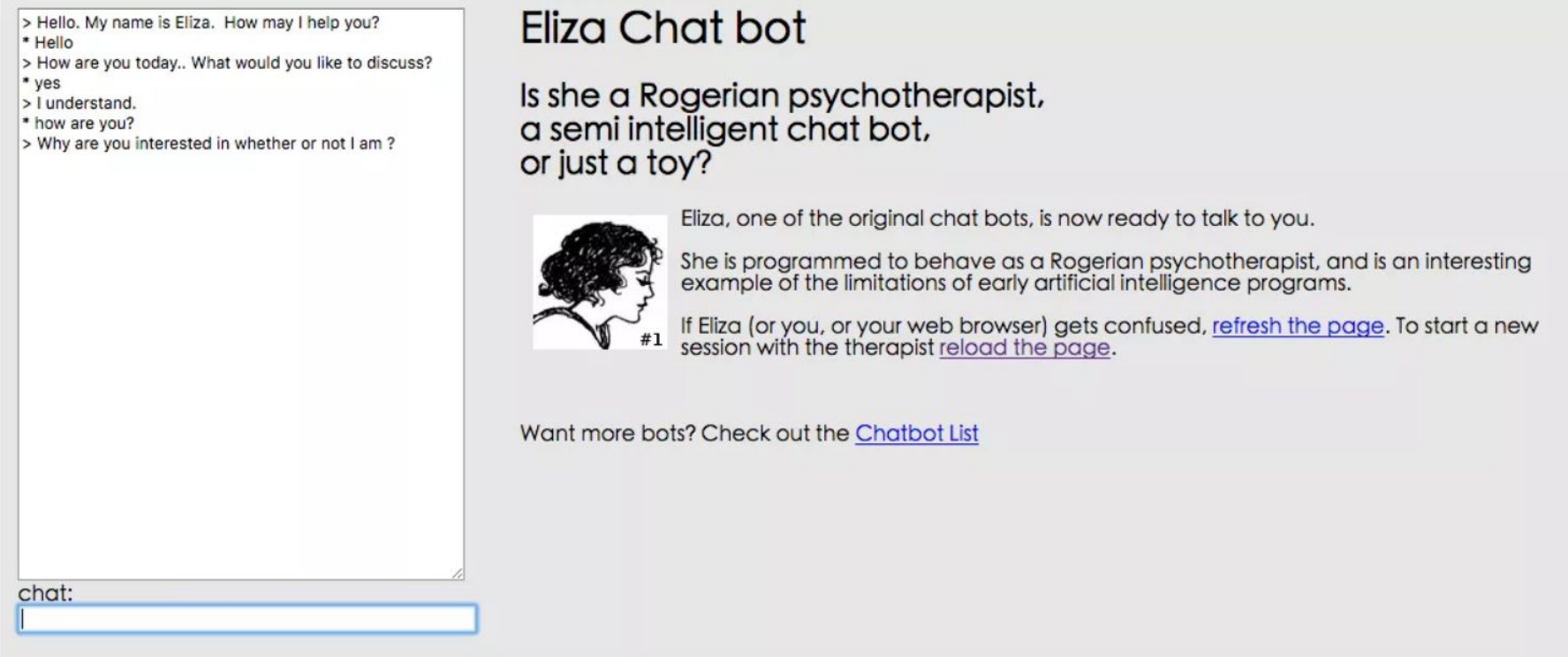
- Introduction to AI & Machine Learning
- Machine Learning Framework
- Types of Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Machine Learning Workflow
  - Data Preprocessing
  - Train & Testing Data
  - Evaluation

The development of Artificial Intelligence



reference: https://blogs.nvidia.com.tw/2016/07/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

First progress (1956-1974)



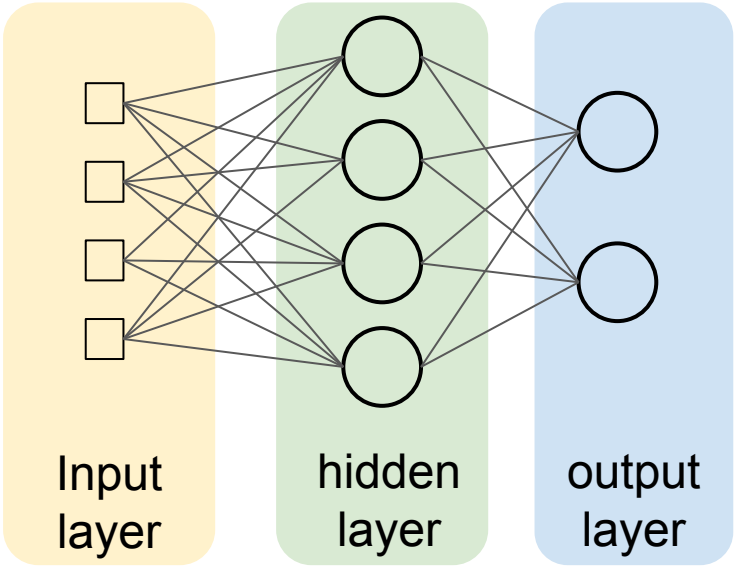
preference: https://www.inside.com.tw/2016/05/26/history-of-chat-bot-instant-messaging

Second progress (1980-1987)

- **Expert System**

MYCIN:
Medical system for diagnosing blood disorders.
First used in 1979

- **Artificial Neural Network**



Third progress (2012 ~)

IMAGENET

8 layers

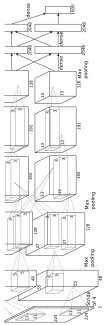
16-19 layers

22 layers

16.4%

7.3%

6.7%



AlexNet (2012)

VGGNet-16 (2014)

GoogleNet-16 (2014)

Third progress (2012 ~)

152 layers!!

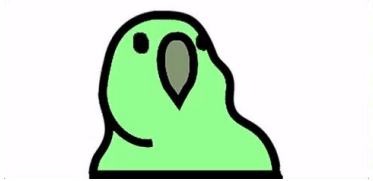
Human

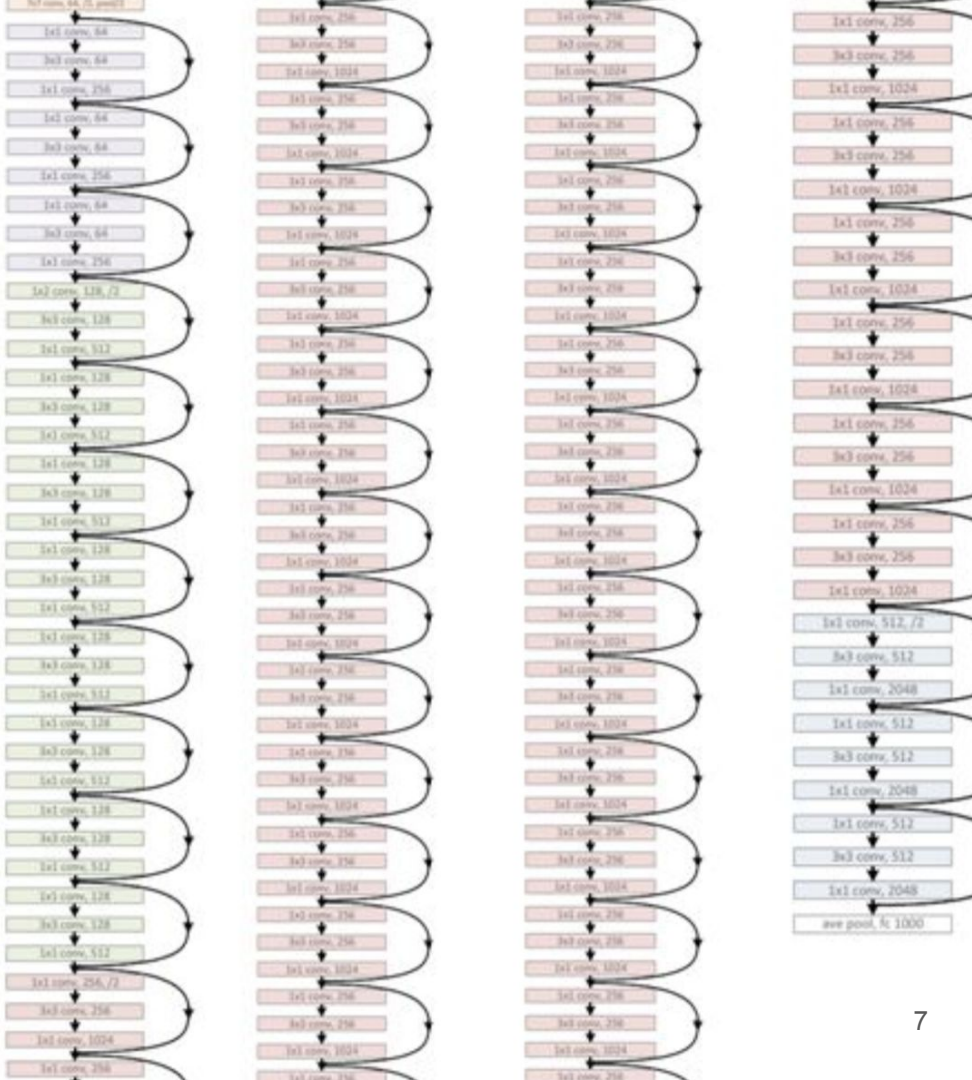
5.1%

Top-5 error rate

ResNet-152 (2015)

3.7%



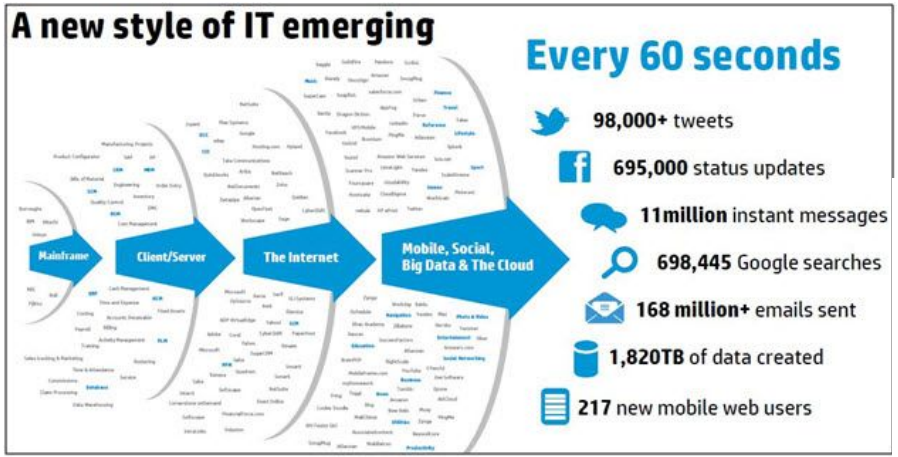


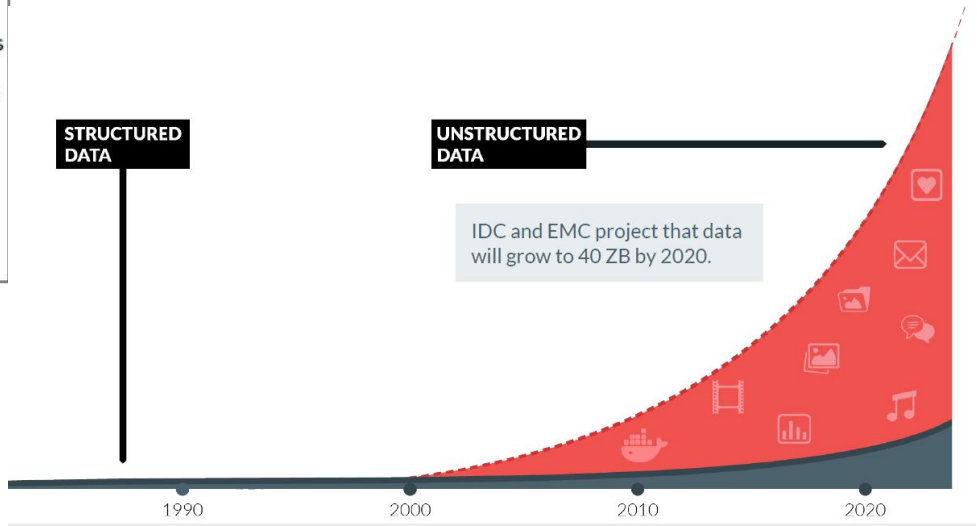
AlphaGo

What makes AI success in this era?

What makes AI success in this era?

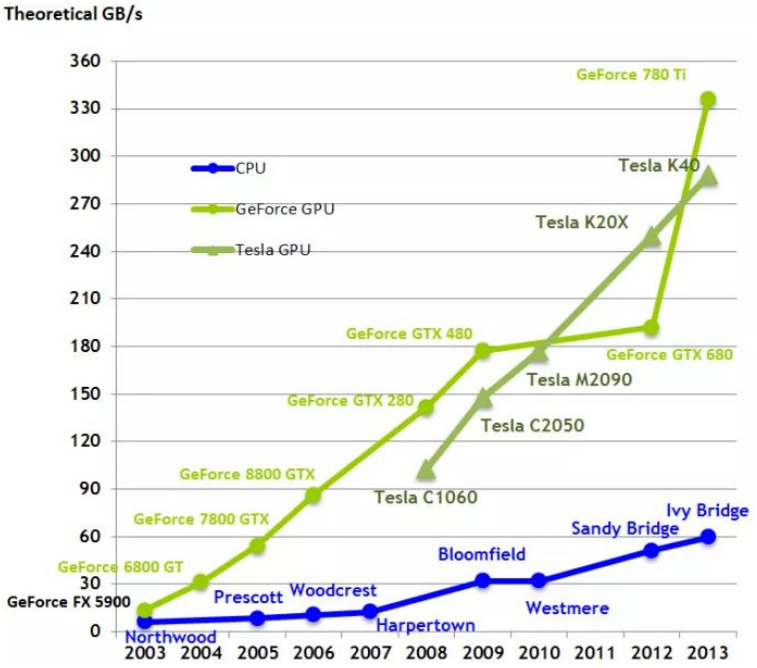
- Explosive increase in **data**



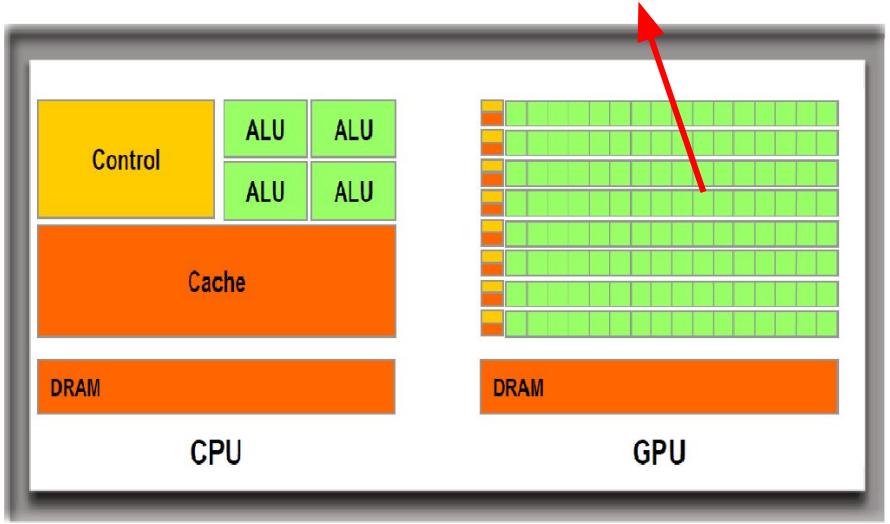


What makes AI success in this era?

- Breakthrough in **computing - GPU**

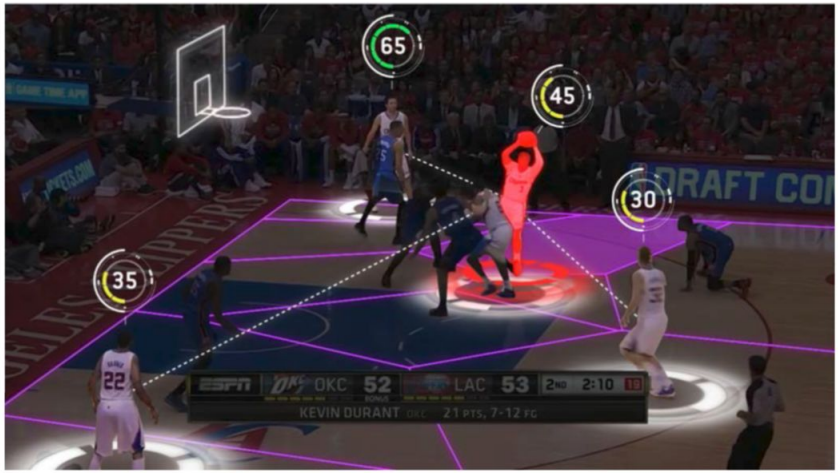


GPU consists of more ALU than CPU in a chip.



video: https://www.youtube.com/watch?v=-P28LKWTzrI

AI Application



(Photo Credit: Jia-Bin Huang)

video:
https://www.ted.com/talks/joseph_redmon_how_a_computer_learns_to_recognize_objects_instantly?language=zh-tw#t-395546

Why do we need machine learning?

先試想以下問題:

1. 我們要怎麼判斷一封電子郵件是垃圾郵件還是重要郵件？
2. 我們要推薦什麼樣的商品給我們的顧客？
3. 我們可以用先前學到的資料結構、演算法處理這樣的問題嗎?

- Weakness of hand-crafted rules
  - hard to consider all possibilities
- Data-driven approach
  - **Machine Learning**

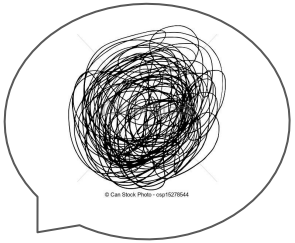
What is Machine Learning

cat

dog

bird

Learning



It is a cat!

Write a program for learning.

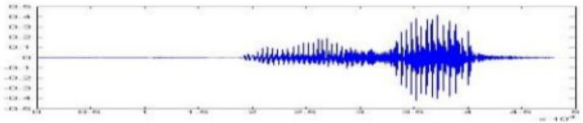
Machine Learning

Learning $\approx$ Looking for a **function** from **data**

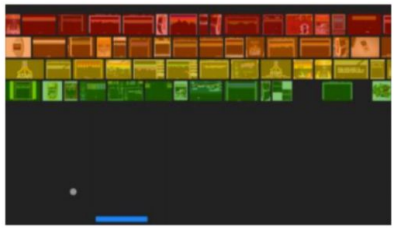
- Image Recognition

$f($  $) =$ “Cat”

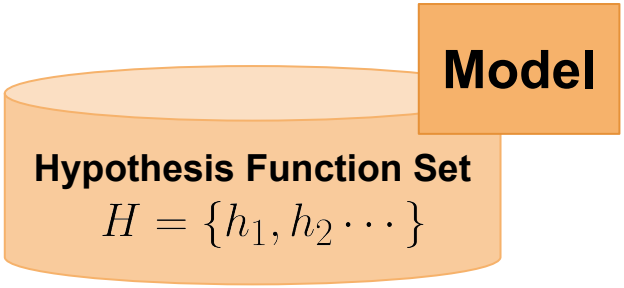
- Speech Recognition

$f($  $) =$ “你好”

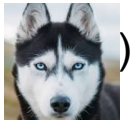
- Play Games

$f($  $) =$ “move left”

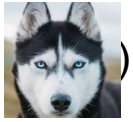
Framework



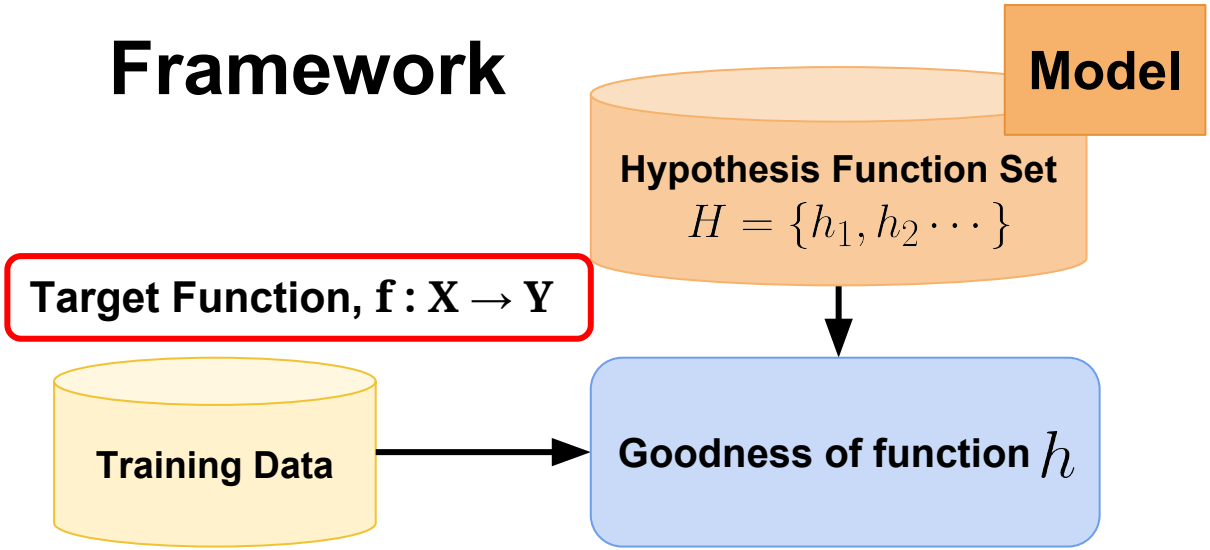
h_1() = “cat”

h_1() = “dog”

h_2() = “cat”

h_2() = “bird”

Framework

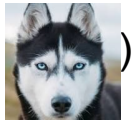


f() = “cat”

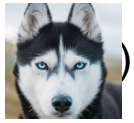
f() = “dog”

f() = “bird”

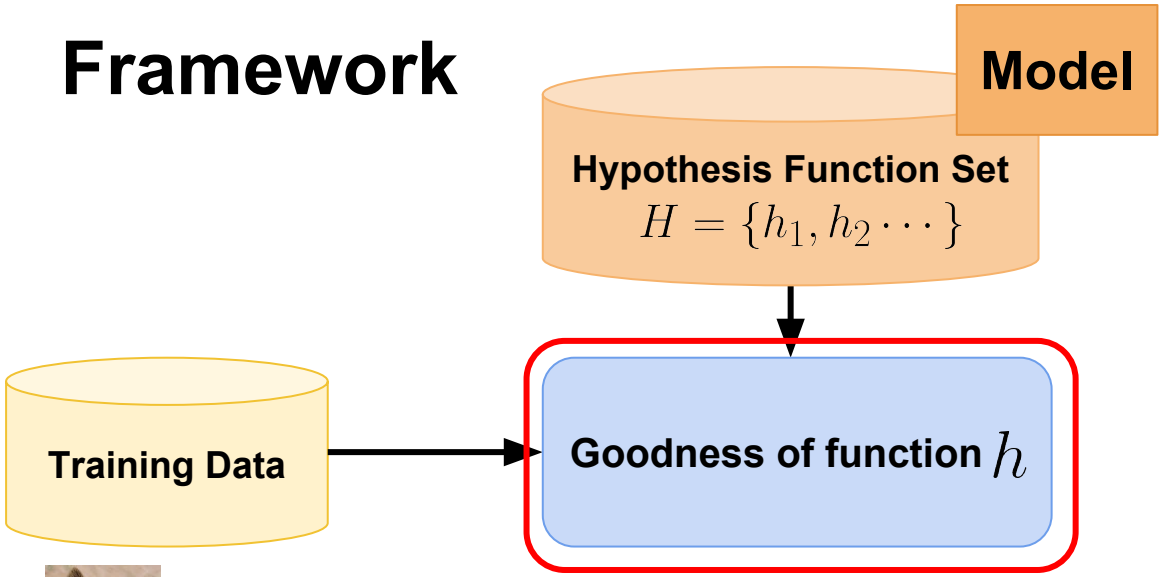
h_1() = “cat”

h_1() = “dog”

h_2() = “cat”

h_2() = “bird”

Framework



f( ) = “cat”

f( ) = “dog”

f( ) = “bird”

h$_1$( ) = “cat”

h$_1$( ) = “dog”

h$_2$( ) = “cat”

h$_2$( ) = “bird”

Framework

Model

Hypothesis Function Set

$H = \{h_1, h_2 \cdots\}$

Target Function, $f : X \rightarrow Y$

Training Data

Goodness of function h

$h^* \approx$ Target Function

Pick the best function h^*

f() = “cat”

f() = “dog”

f() = “bird”

Framework

Model

Hypothesis Function Set

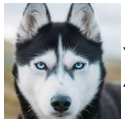
$H = \{h_1, h_2 \cdots\}$

Training Data

Goodness of function h

Pick the best function h^*

f() = “cat”

f() = “dog”

f() = “bird”

h*() = “cat”

$h^*\left(x'\right) = y'$

Testing Data

Framework

Model

Hypothesis Function Set

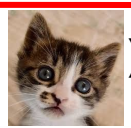
$H = \{h_1, h_2 \cdots\}$

Training

Training Data

Goodness of function h

Pick the best function h^*

f() = “cat”

f() = “dog”

f() = “bird”

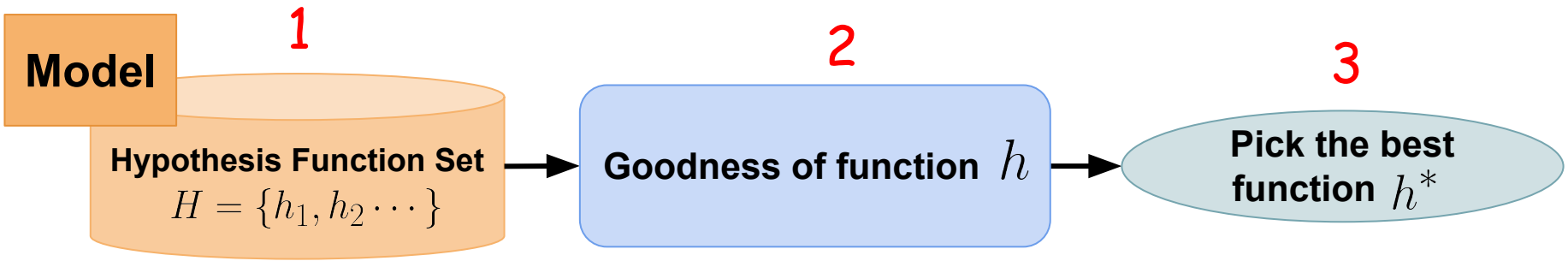
$h^*\left(x'\right) = y'$

Testing Data

Testing

h*() = “cat”

Machine Learning Step



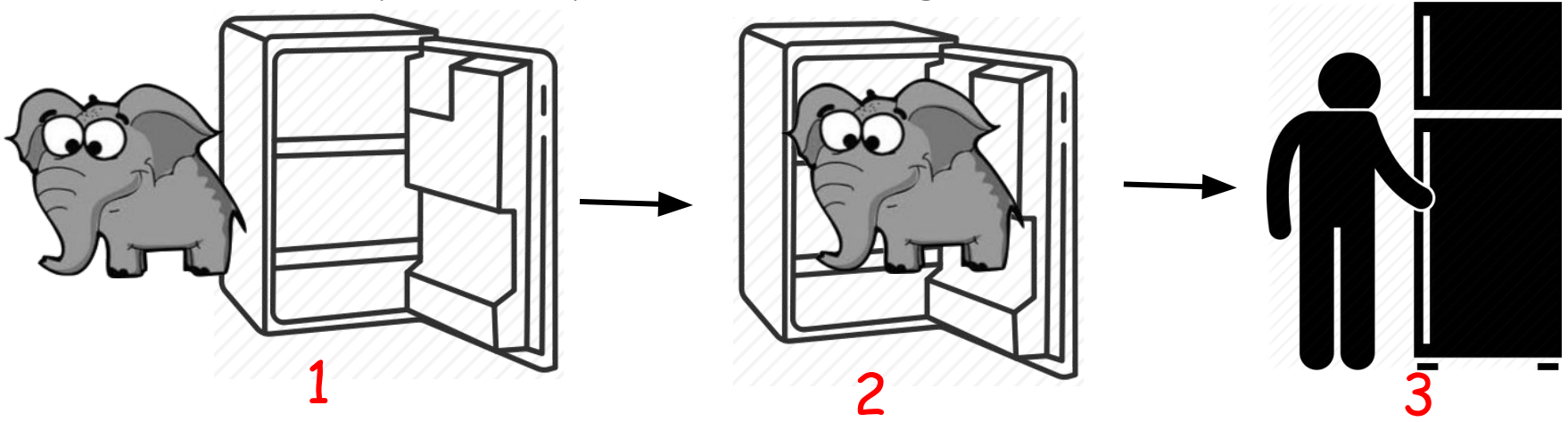
And it just like...

Model

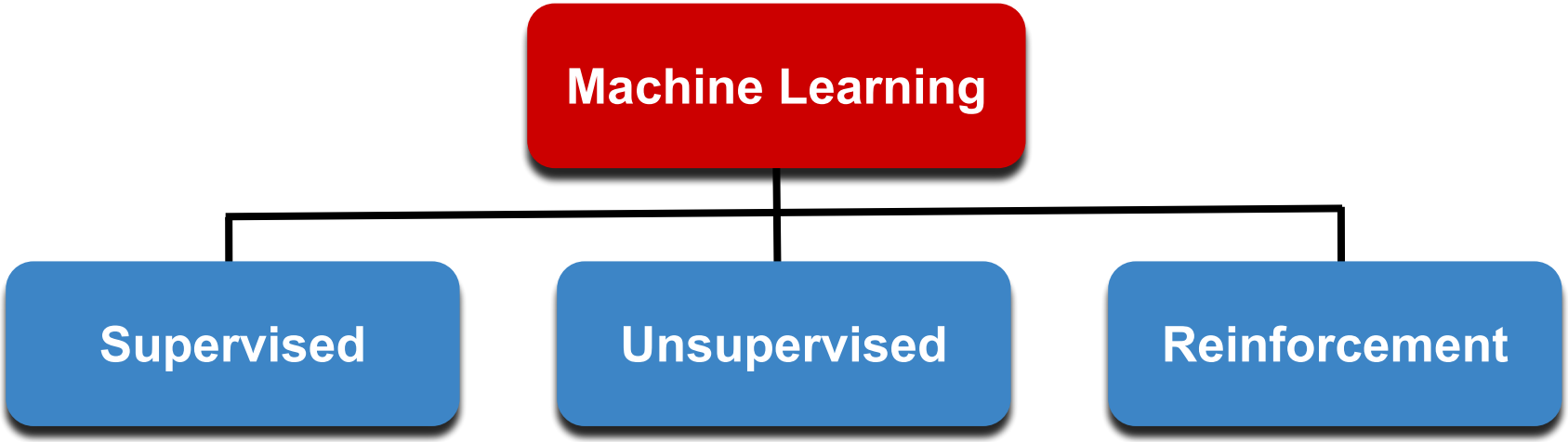
Hypothesis Function Set
$H = \{h_1, h_2 \cdots\}$

→ **Goodness of function** h → **Pick the best function** h^*

when we want to put an elephant into a refrigerator



Types of Machine Learning

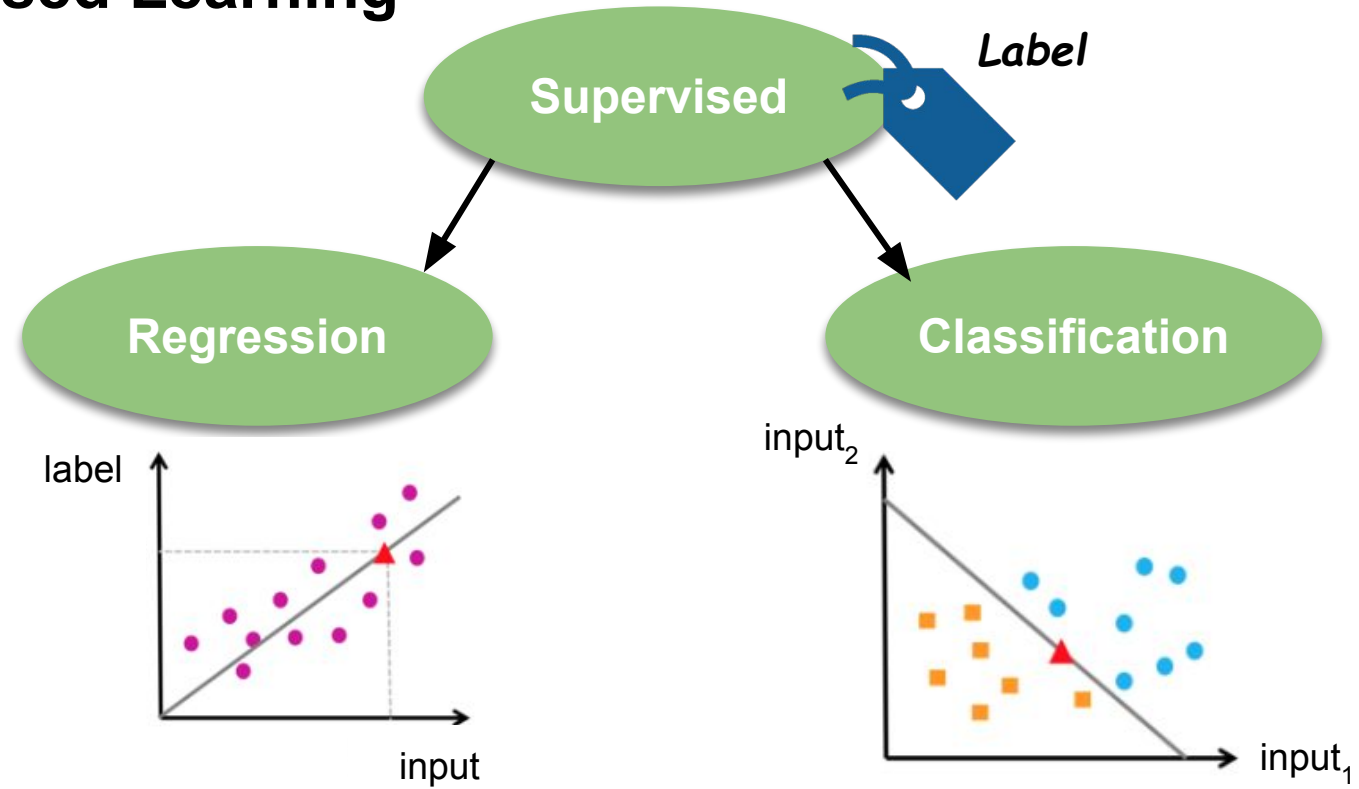


- Labeled data
- Direct feeback
- Predict outcome /future

- Unlabeled data
- No feeback
- Find hidden structure

- Markov Decision Process
- Reward function
- Learn policy

Supervised Learning

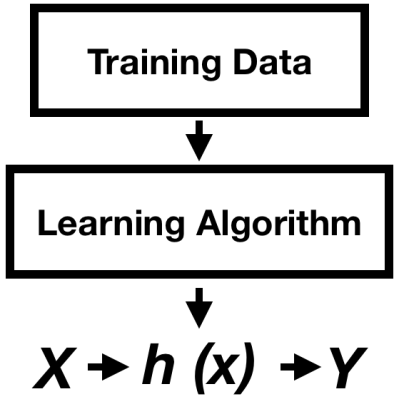


The class label is **continuous**.

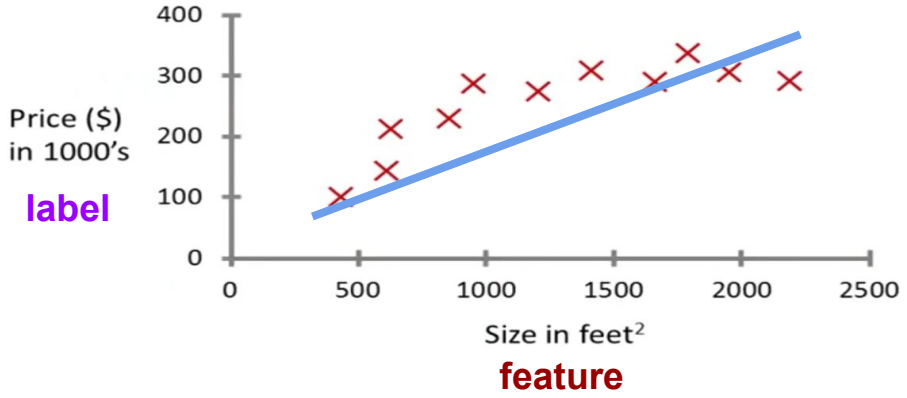
The class label is **discrete**.

Supervised Learning - Regression

- Goal : Learning a function for a **continuous** output.
  (scalar)
  - Housing price prediction



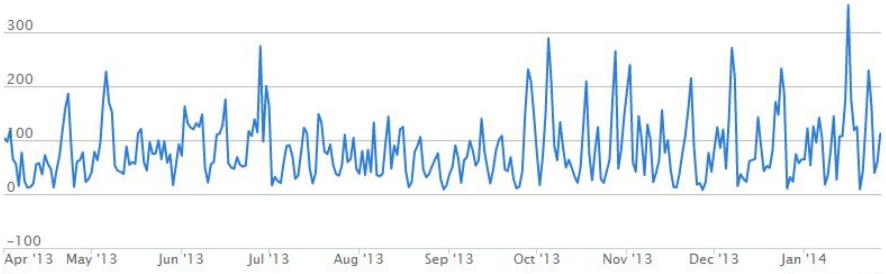
	Frontage	Lot size	…	Sale price
x1	65	8450	…	208500
x2	80	9600	…	181500
x3	68	11250	…	223500
x4	60	9550	…	140000
x5	84	14260	…	250000
x6	65	8450	…	143000
…				



Supervised Learning - Regression

- Stock Price Prediction

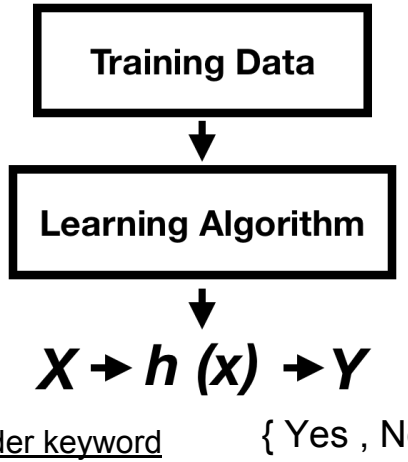
- PM2.5 Prediction



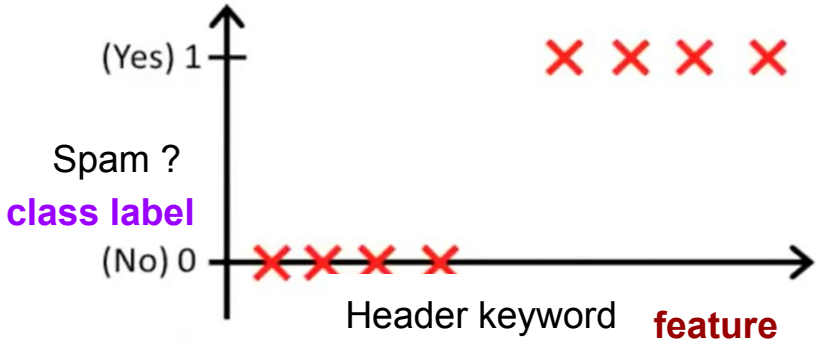
Supervised Learning - Classification

- Goal : Learning a function for a **discrete** output.
  (categorical)

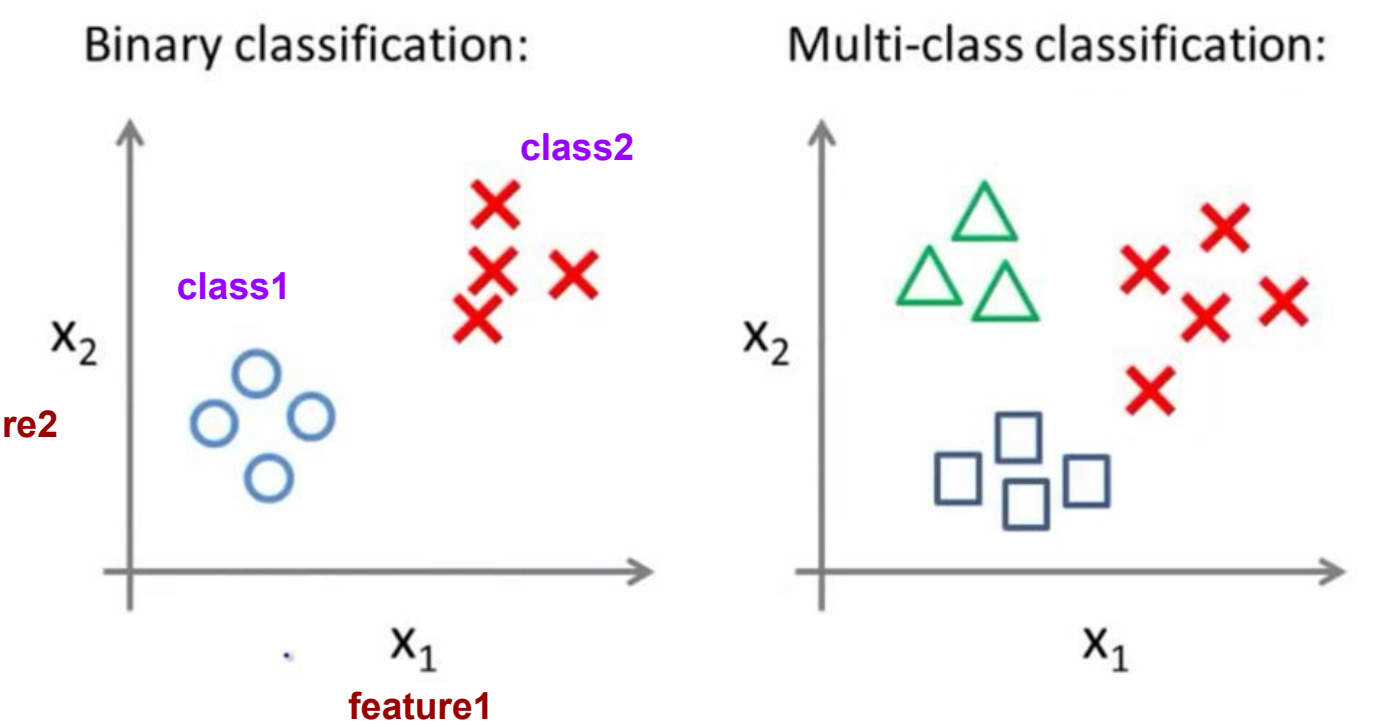
  - Email Spam filtering (Binary Classification)



	Sender in address book?	Header keyword	…	Spam?
x1	Yes	Schedule	…	No
x2	Yes	meeting	…	No
x3	No	urgent	…	Yes
x4	No	offer	…	Yes
x5	No	cash	…	Yes
x6	No	comp-551	…	No
…				



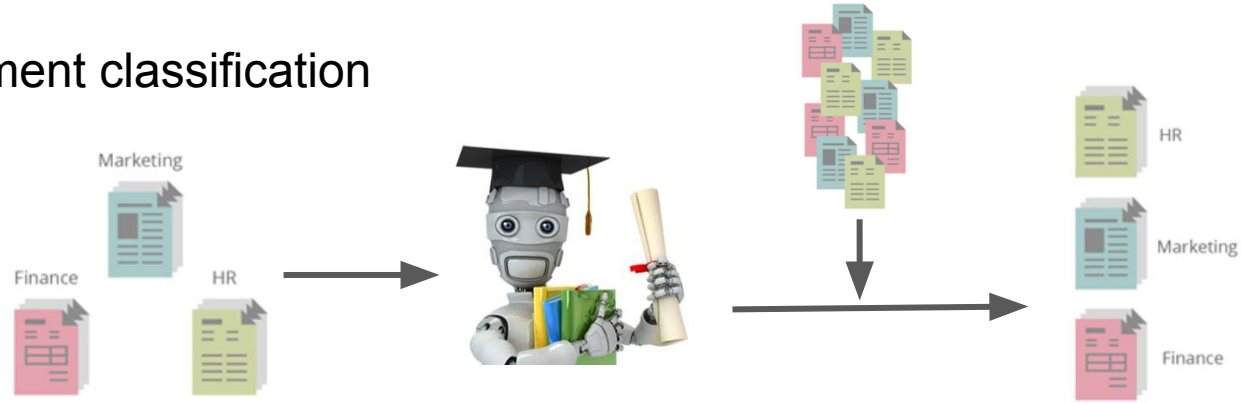
Supervised Learning - Classification



(slides by Cheng-Te Li)

Supervised Learning - Classification

- Document classification

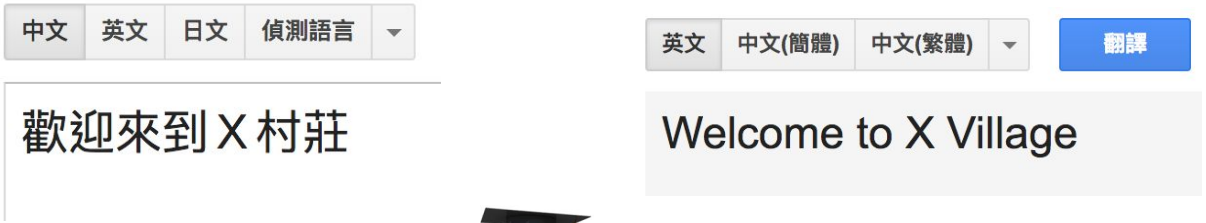


- Medical diagnosis
  - Categorical : not ill, cold, flu

- Weather
  - Categorical : sunny, cloud, rain, snow

Supervised Learning - Structured Learning

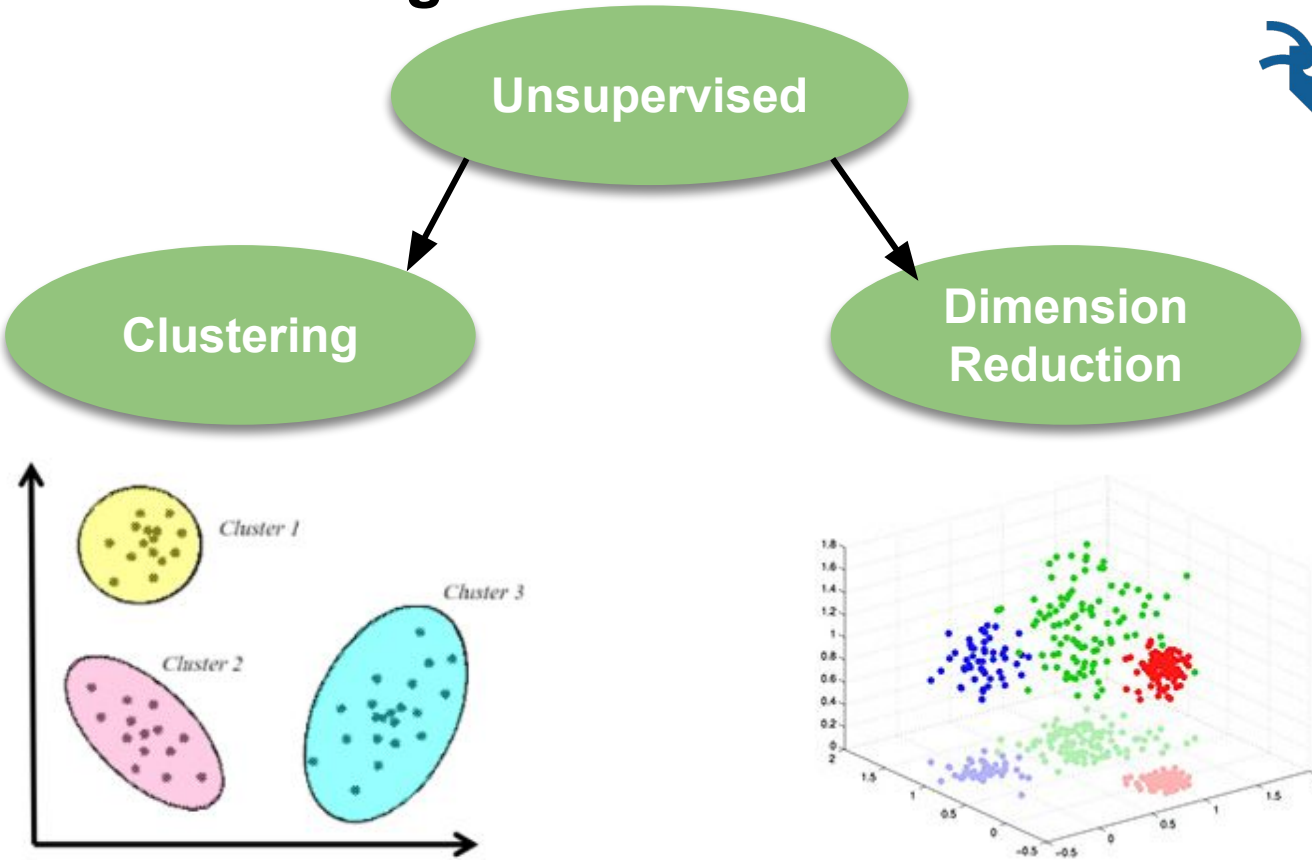
- Structured Data - beyond classification
  - Machine Translation



  - Speech Recognition

歡迎來修AI課程

Unsupervised Learning



Classifcation

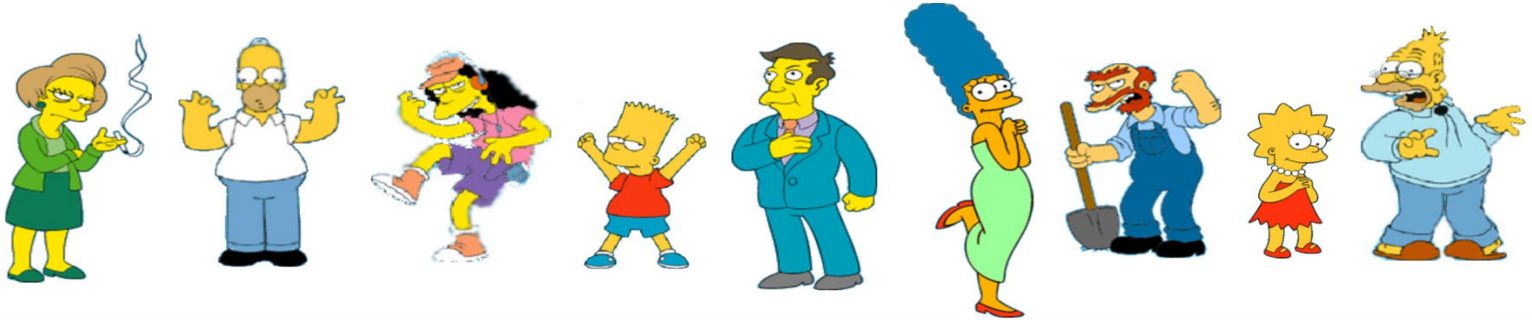
- Given a training set of labeled objects, learn a decision rule.

Clustering

- Given a collection of (unlabeled) objects, find meaningful groups.

Unsupervised Learning - Clustering

- Grouping



Clustering is subjective

(Slide Credit: Eamonn Keogh)

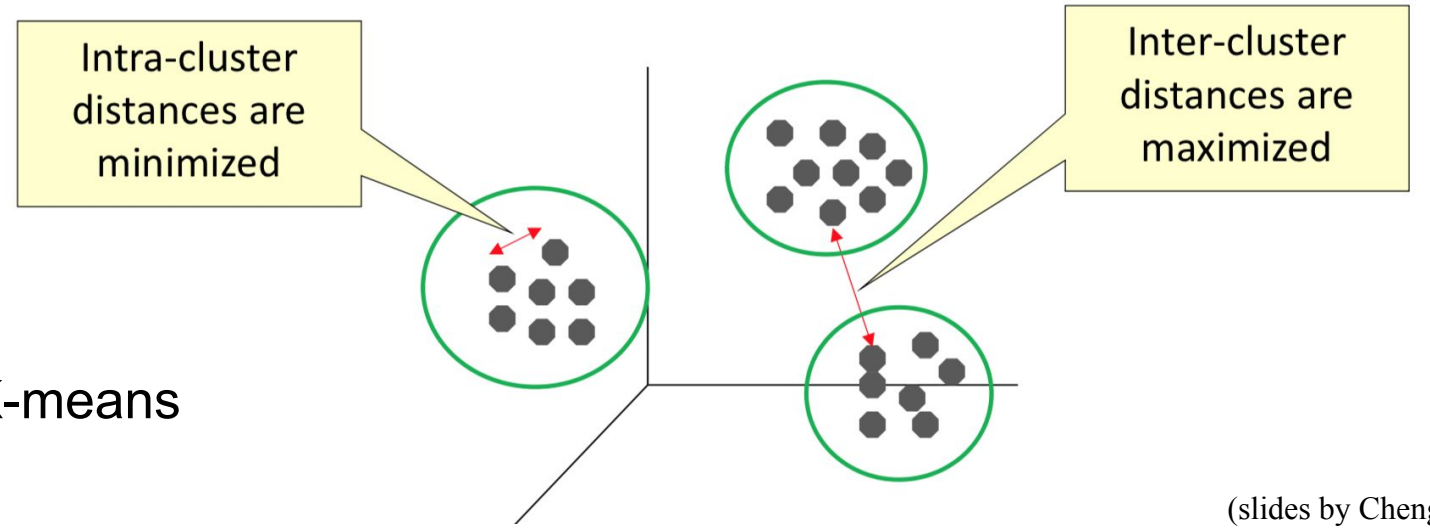
Unsupervised Learning - Clustering

- Similarity

Chihuahua Or Muffin?

Unsupervised Learning - Clustering

- A grouping of objects such that the objects in a **group (cluster)** are
  - **Similar** to one another in a group
  - **Different** from the objects in other groups

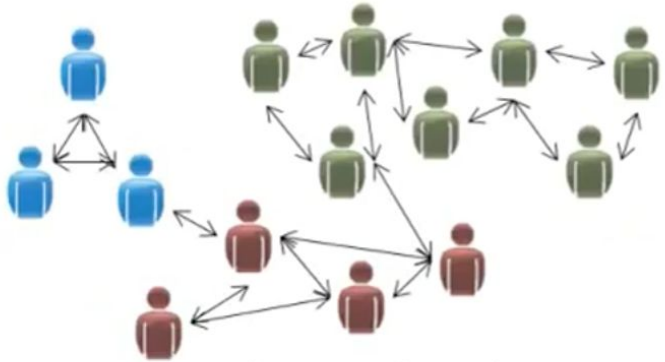


- K-means

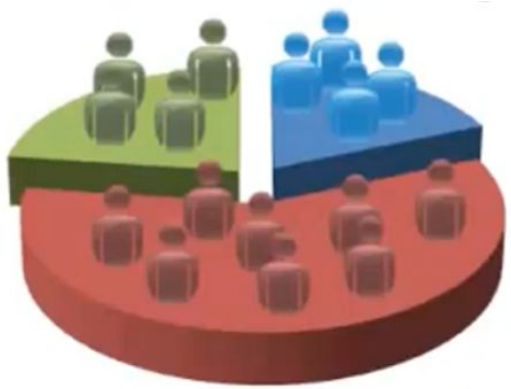
(slides by Cheng-Te Li)

Unsupervised Learning - Clustering

- Application



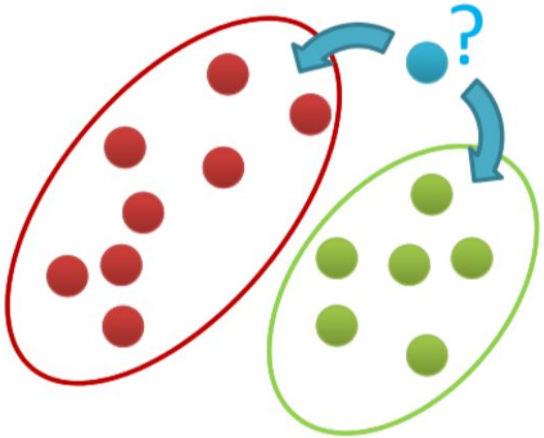
Social network analysis



Market segmentation

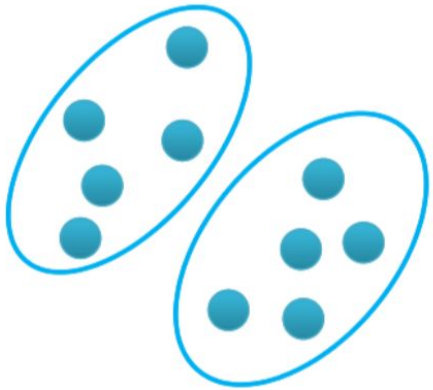
Comparison

Classification



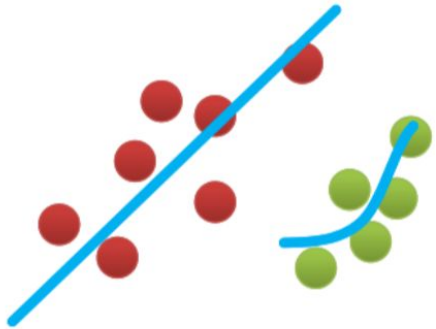
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

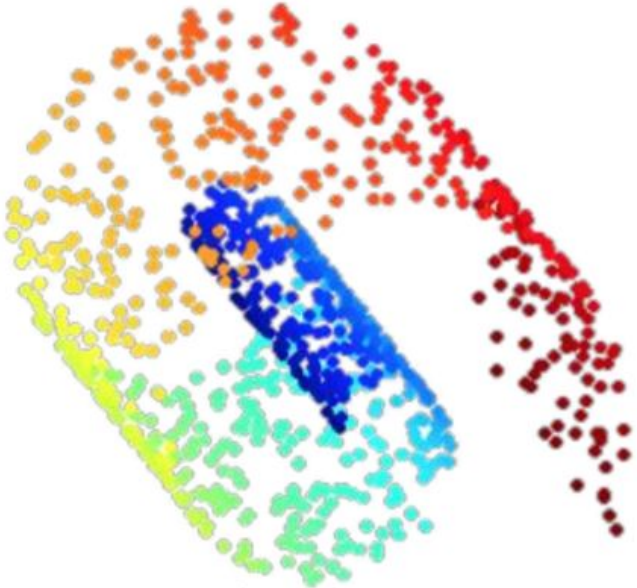
Regression



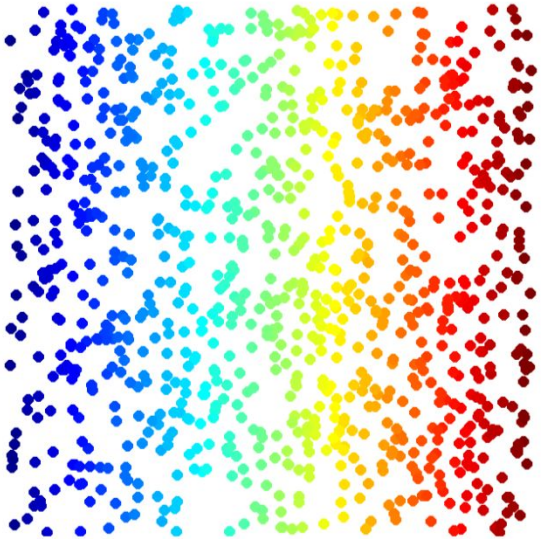
- Identify a line with a certain slope describing the data

(slides by Cheng-Te Li)

Unsupervised Learning - Dimension Reduction



Looks like 3-D

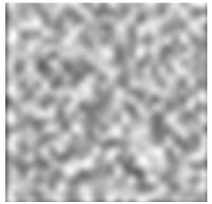


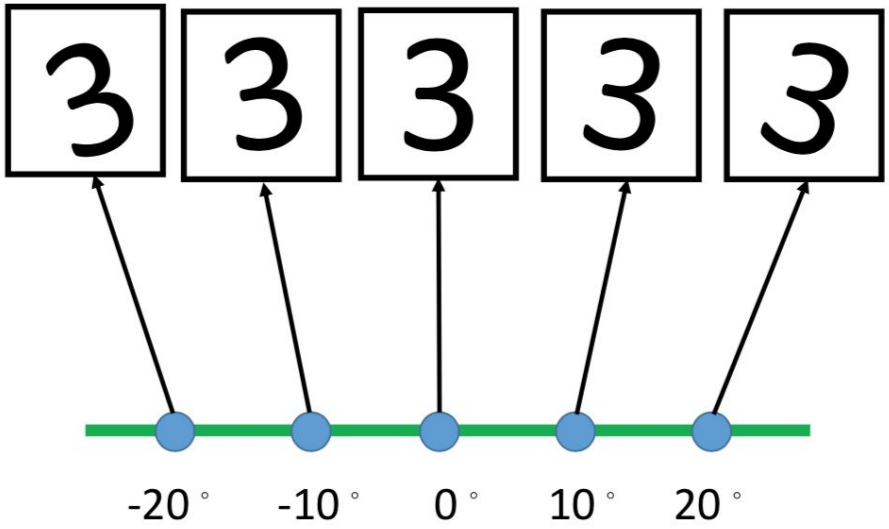
Actually, 2-D

http://reuter.mit.edu/blue/images/research/manifold.png

Unsupervised Learning - Dimension Reduction

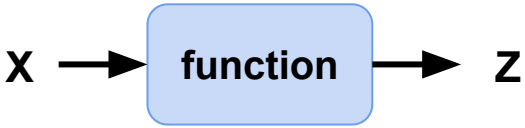
- A digit is 28 x 28 dimensions.





(quoted from Hung-yi Lee's slides)

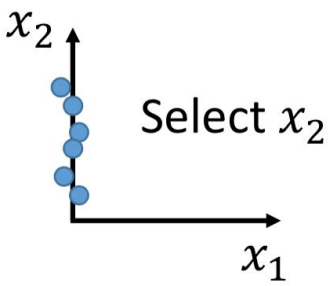
Unsupervised Learning - Dimension Reduction

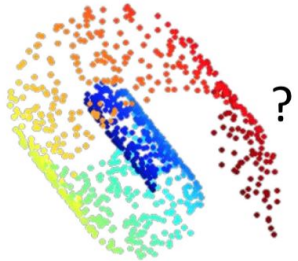


High Dimension

Low Dimension

- Feature Selection





- Principal Component Analysis (PCA)

$z = Wx$

(quoted from Hung-yi Lee's slides)

Unsupervised Learning

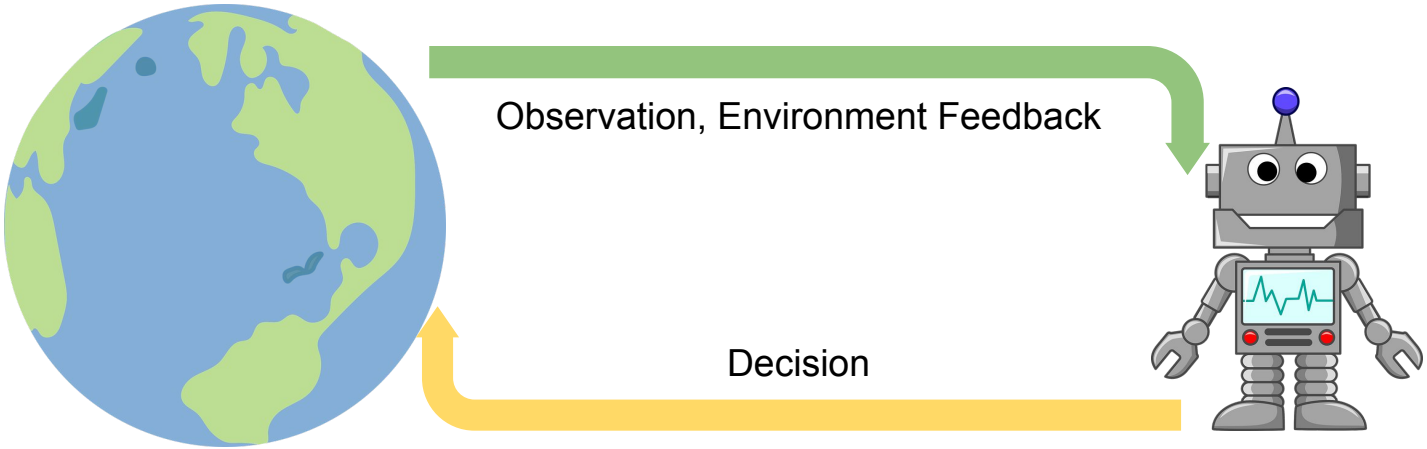
- Generation

http://ttic.uchicago.edu/~ klivescu/MLSLP2016/ (slides of Ian Goodfellow)

Reinforcement Learning

Reinforcement learning(強化學習) is another learning paradigm in machine learning.

- Inspired from control theory and animal learning.
- The learning agent will look around the environment and make a decision.



Reinforcement Learning - inspiration

How do we teach animal to learn expected reaction? Give reward

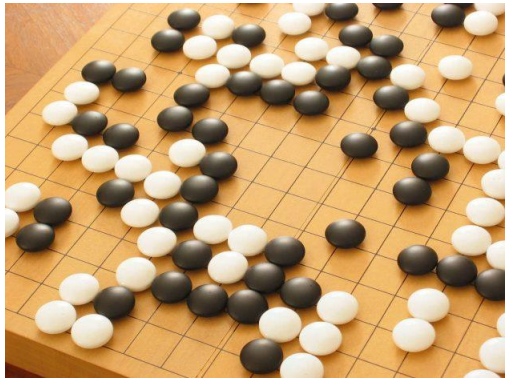
Reinforcement Learning - inspiration

When it do something wrong, just give it penalty.

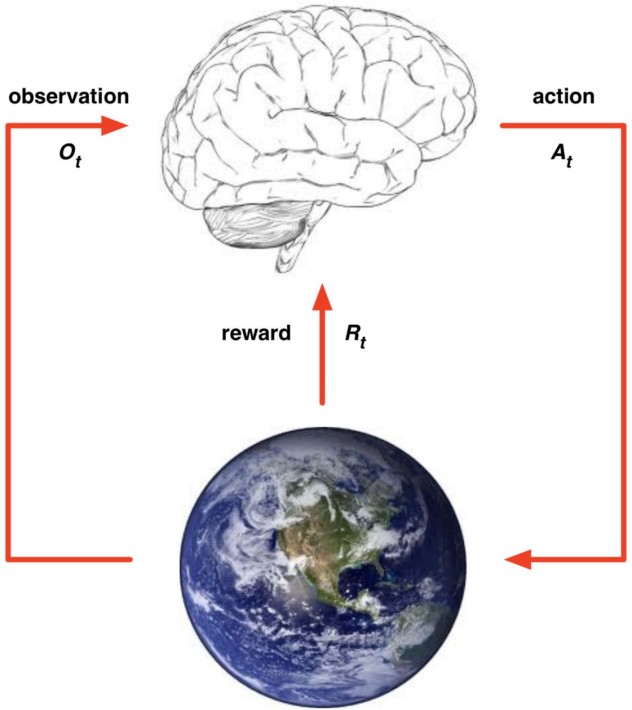
Reinforcement Learning - MDP

In theory, if we can modeled a decision problem as a Markov Decision Process (MDP) and we can apply reinforcement learning to handle such problem. we need to define the following component in MDP:

- State Space: How many possibilities in observation?
- Action Space: What can I do with the environment?
- Reward function: How good or bad about the decision?
- Transition: Which state do I transfer to in the next time?



Reinforcement Learning - MDP



At each <u>step t</u> the agent:

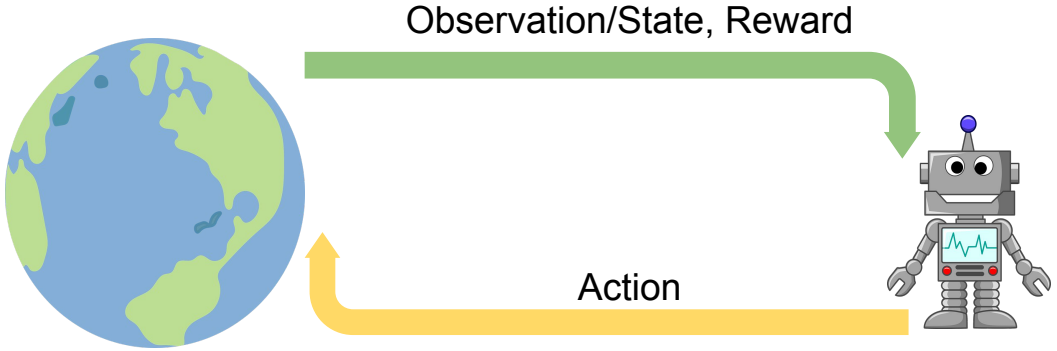
- Receives scalar reward R_t
- Receives observaiotn O_t
- Executes action A_t

The environment:

- Receives action A_t
- Emits observation O_{t+1}
- Emits scalar reward R_{t+1}

Image credits: David Silver's reinforcement Learning course in UCL, UK.

Reinforcement Learning - MDP



A **MDP** can be recorded as the following sequence:

$$S_1, A_1, R_1, \ldots, S_t, A_t, R_t, S_{t+1}, A_{t+1}, R_{t+1}, \ldots, S_{terminal}$$

We often call this sequence as **Trajectory** or **Episode**, denoted by τ

Reinforcement Learning - objective

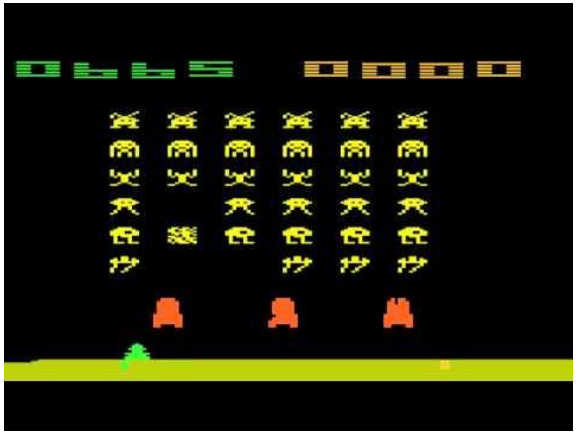
The objective of reinforcement learning is to learn the policy that can **maximize expected total rewards** in different MDPs.

$$S_1, A_1, R_1, \ldots, S_t, A_t, R_t, S_{t+1}, A_{t+1}, R_{t+1}, \ldots, S_{terminal}$$

$$\mathbb{E}_{\tau \sim P(\tau)} \left[\sum_{t=1}^{T} R_t \right]$$

Reinforcement Learning - algorithms development

Currently, the reinforcement learning algorithms are under development and have been tested in different kinds of environments:



Video Games

Board Games

(The state space of 19x19 Go is 10^{171})

Robotics

Reinforcement Learning - resources

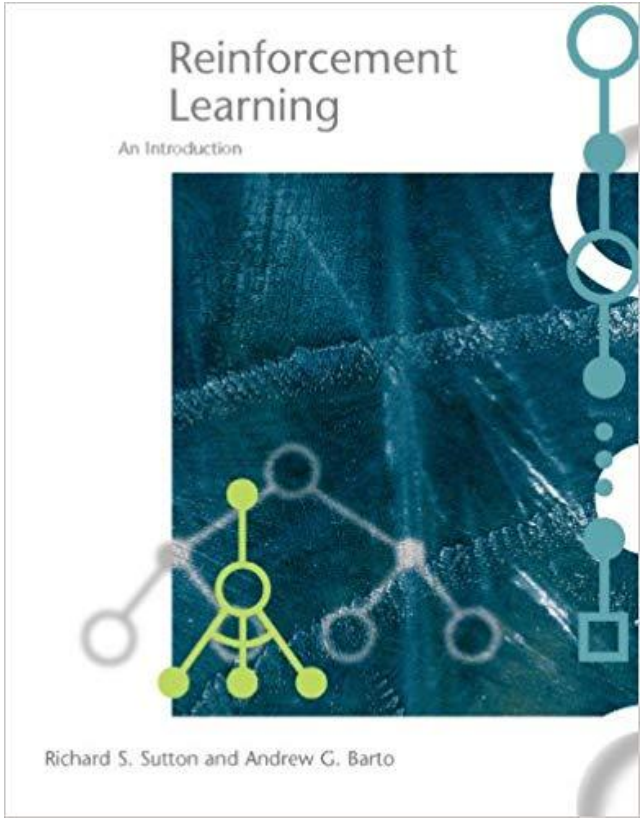
Course:

- The best choice to engaged in deep reinforcement learning is CS294, UCB.
- Another famous course is David Silver's reinforcement learning in UCL.
- The deep learning course from 吳尚鴻老師 in NTHU.
- The reinforcement learning/deep learning course from 李宏毅老師 in NTU. You can find wonderful materials on Youtube.
- We also have short courses about reinforcement learning in NCKU:
  - https://netdbncku.github.io/dsai/2018/

Reinforcement Learning - resources

Currently, there are just few reading materials about reinforcement learning:

- Reinforcement Learning: An Introduction
  - The second edition will be published recently, and you can find the online draft.
- Algorithms for Reinforcement Learning
  - This book describes more details about reinforcement learning algorithms.
- Top Conference Papers
  - ICLR, NIPS, ICML, AAAI, ...

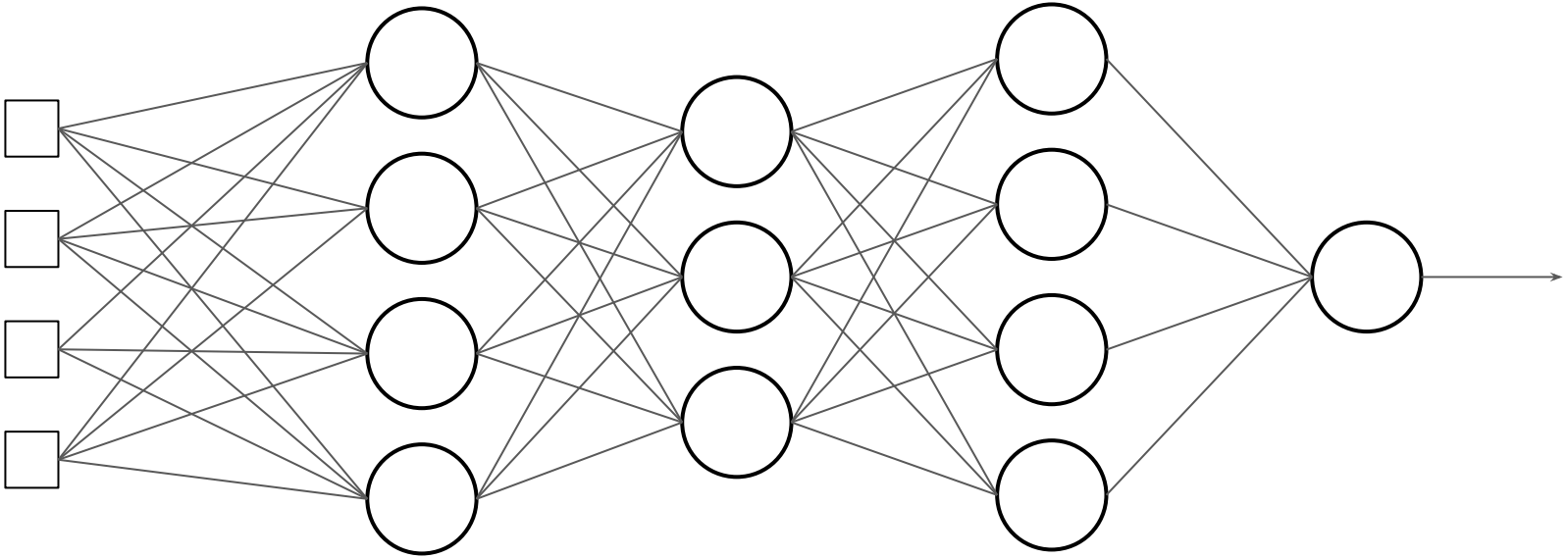


BTW, before you learn reinforcement learning, you had better to learn the basics of machine learning.

What is deep learning?

Deep Learning

Deep learning just means using deep neural network as your model(function approximator) in machine learning.



Deep Learning

Recently, the deep learning won big success in many domains and it can be applied in different kinds of learning paradigm to handle unstructured data.

- Supervised learning: object detection ...
- Unsupervised learning: generative adversarial network(GAN), auto-encoder
- Reinforcement learning: robotics, power distribution, chatbot, NLP ...

Object detection

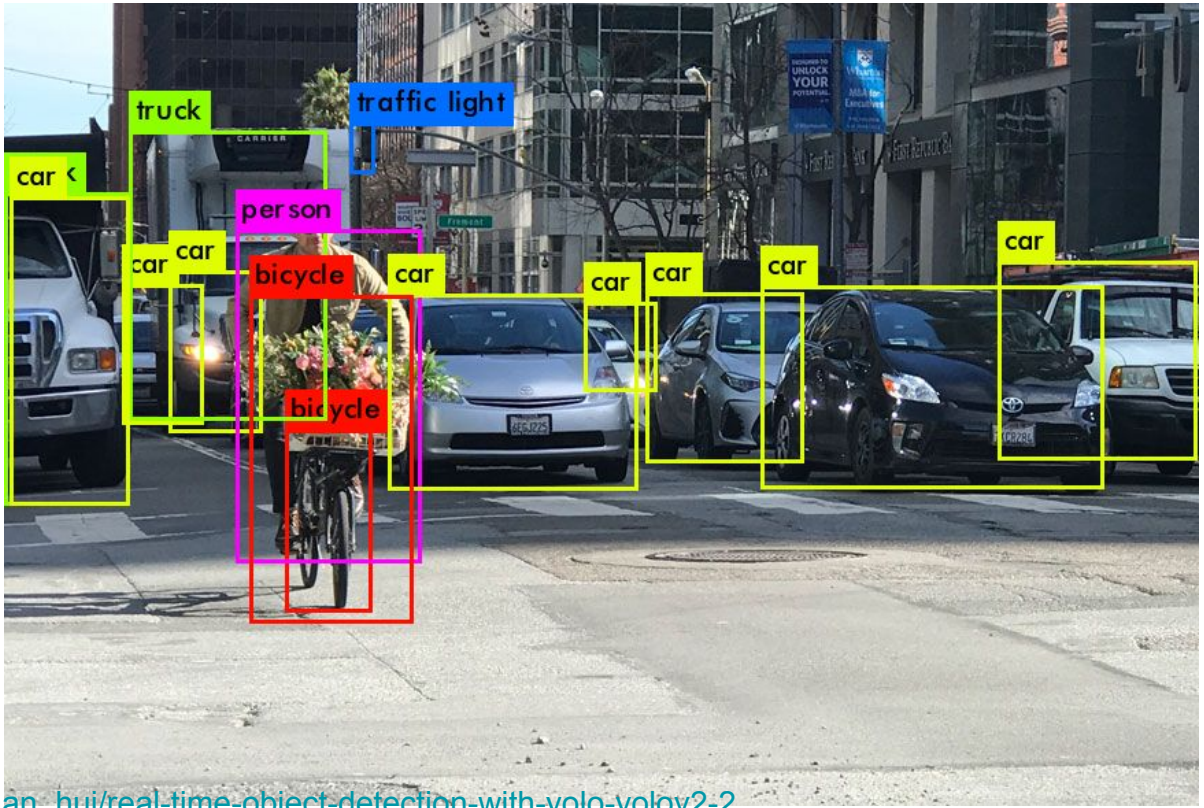


Image Credits:
https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

GAN - 二次元妹子生成器

生成

+1 -1

Share on Twitter

模型

Camellia 256x256 Ver.171219 (9.9MB)

发色 随机

发型 随机

眼睛颜色 随机

褐色肌肤 关 随机 开

脸红 关 随机 开

微笑 关 随机 开

张嘴 关 随机 开

帽子 关 随机 开

丝带 关 随机 开

眼镜 关 随机 开

风格 随机

噪声 随机 固定

当前噪声

导入或导出噪声 导入 导出

控制 导入 导出 重置

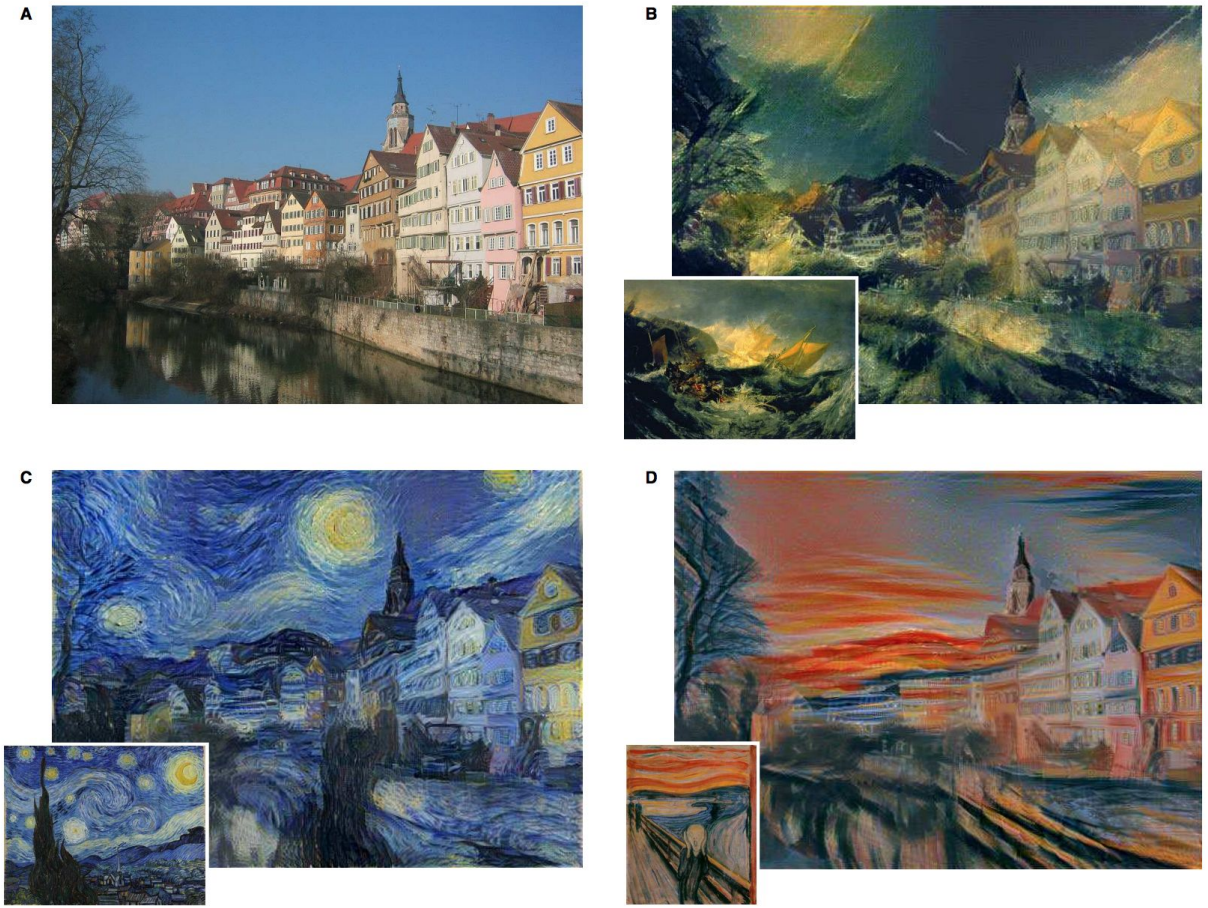
WebGL加速 禁用 启用

https://make.girls.moe/#/

GAN - image-to-image translation

Quoted from Isola et al., “Image-to-Image Translation with Conditional Adversarial Networks”

Style Transfer



Qutoed from Gatys et al., “Image Style Transfer Using Convolutional Neural Networks” (CVPR2016)

Style Transfer

https://www.youtube.com/watch?v=Khuj4ASldmU

Deep Reinforcement Learning - BiCNet

https://www.youtube.com/watch?v=kW2q15MNFug

Deep Reinforcement Learning - AlphaGO Zero

https://deepmind.com/blog/alphago-zero-learning-scratch/

Deep Reinforcement Learning - OpenAI Five

Train the agents to play DOTA2 and they can cooperate with each others. (2018/8/6)

https://blog.openai.com/openai-five-benchmark-results/

Deep Learning

We will cover the basic contents of deep learning in the 3rd lecture, but there are some prerequisite:

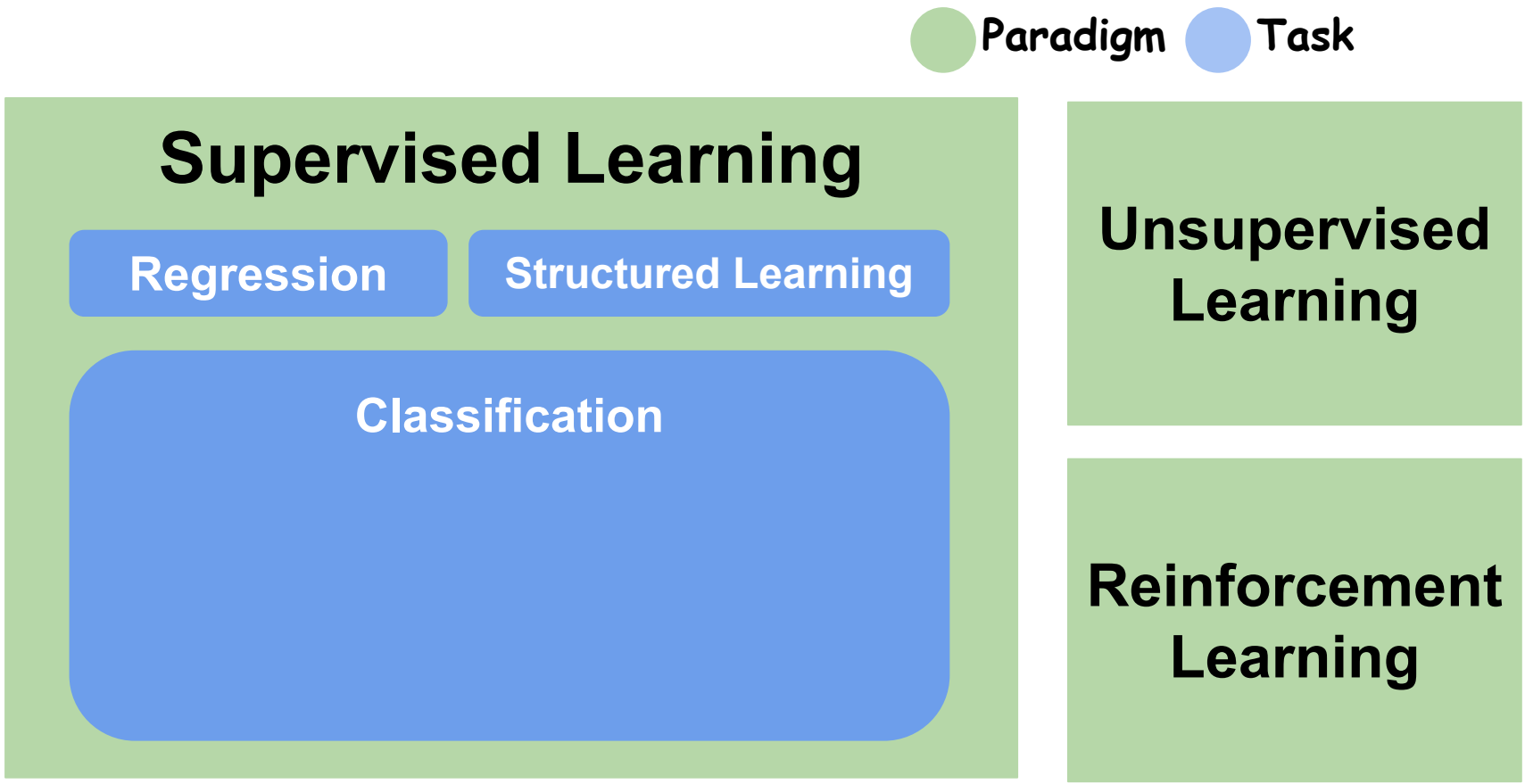
- The concept of Matrix, vector multiplication.
- The concept of partial derivative and chain rule (knowing derivative is OK.)

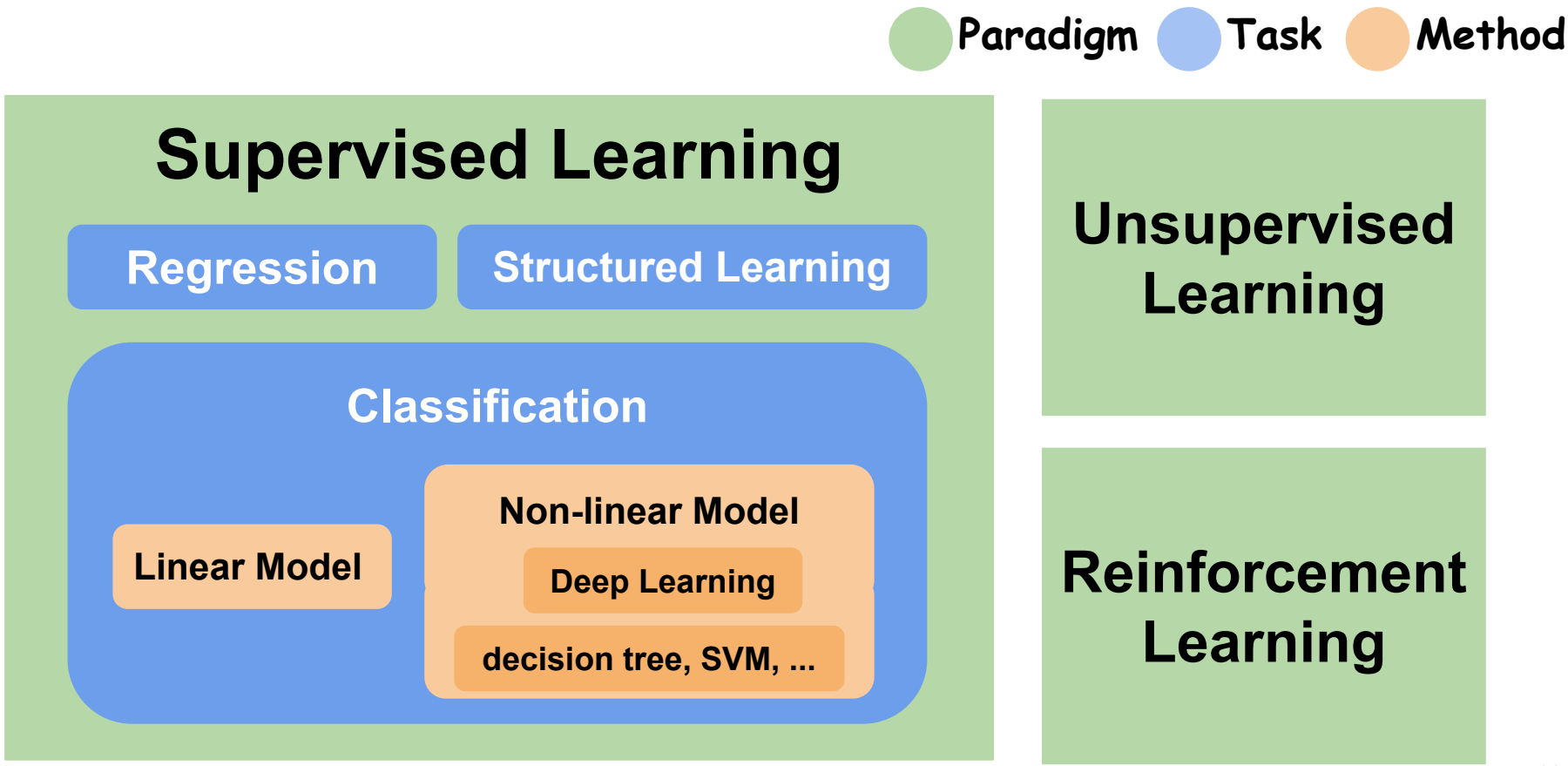
Paradigm

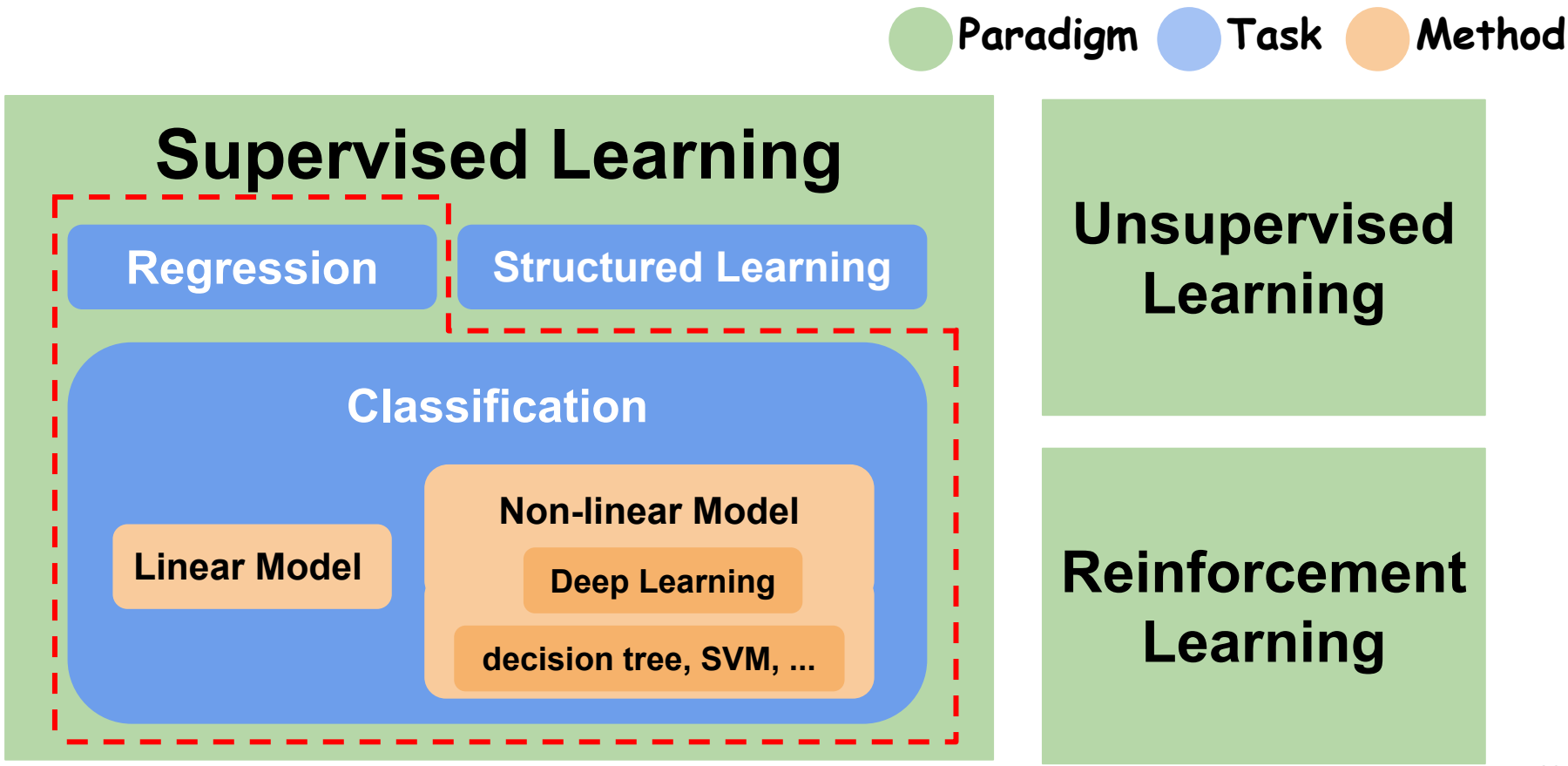
Supervised Learning

Unsupervised Learning

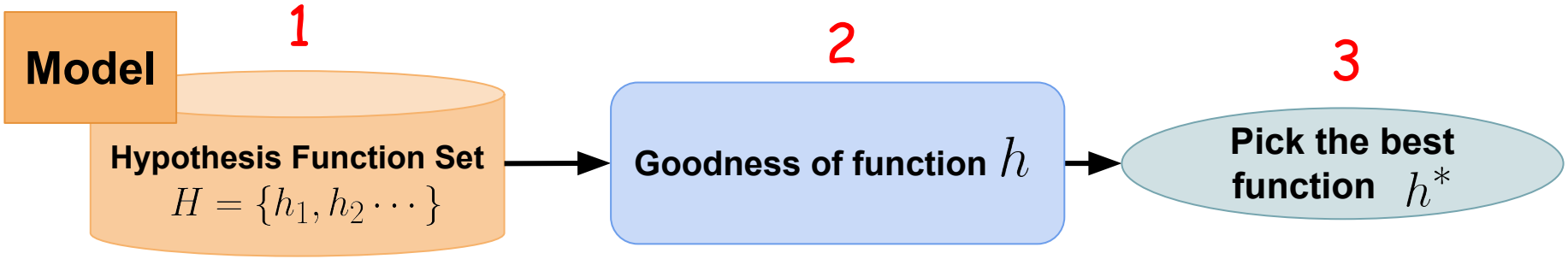
Reinforcement Learning


Paradigm
Task
Supervised Learning
Regression
Structured Learning
Classification
Unsupervised Learning
Reinforcement Learning

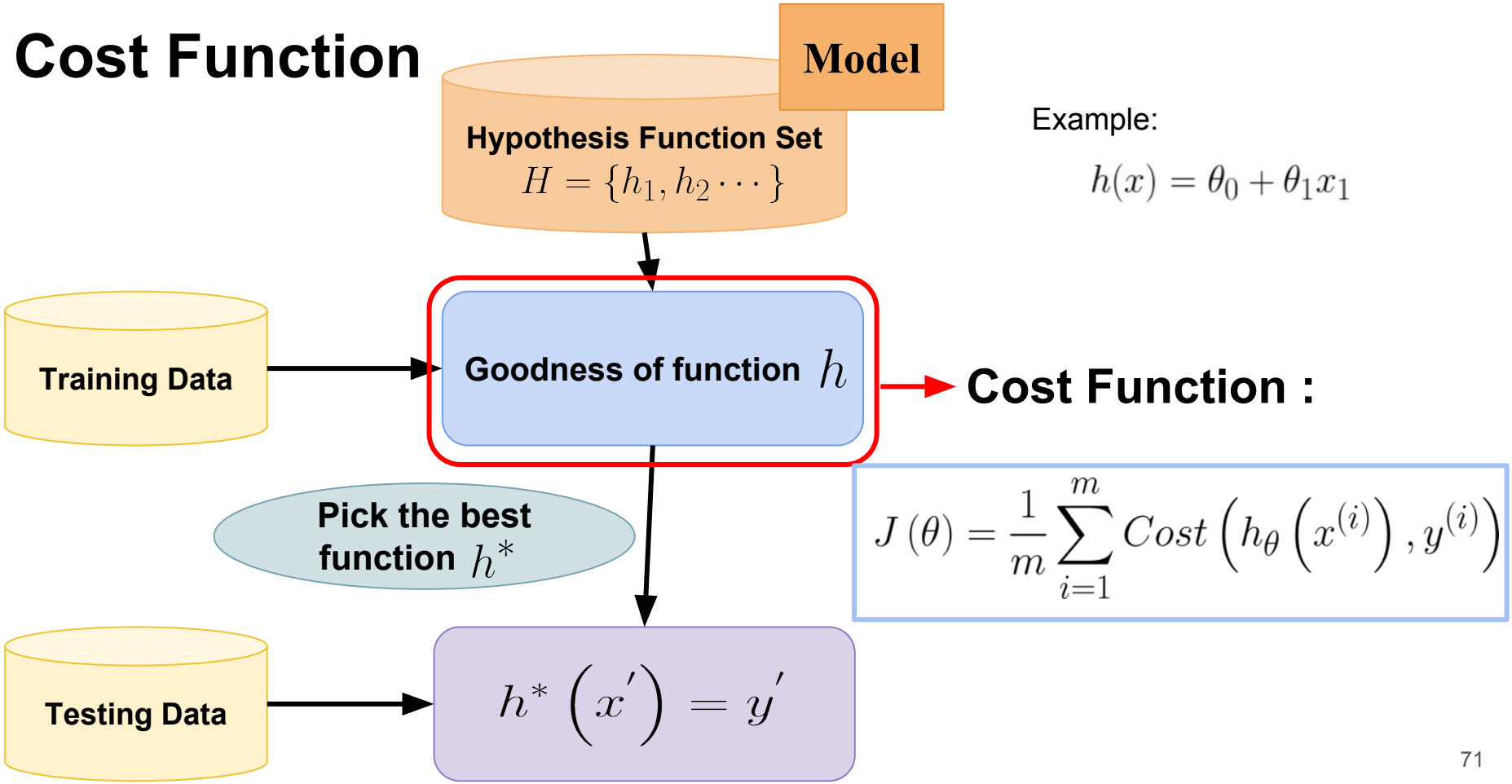

Paradigm
Task
Method
Supervised Learning
Regression
Structured Learning
Classification
Linear Model
Non-linear Model
Deep Learning
decision tree, SVM, ...
Unsupervised Learning
Reinforcement Learning


Paradigm
Task
Method
Supervised Learning
Regression
Structured Learning
Classification
Linear Model
Non-linear Model
Deep Learning
decision tree, SVM, ...
Unsupervised Learning
Reinforcement Learning

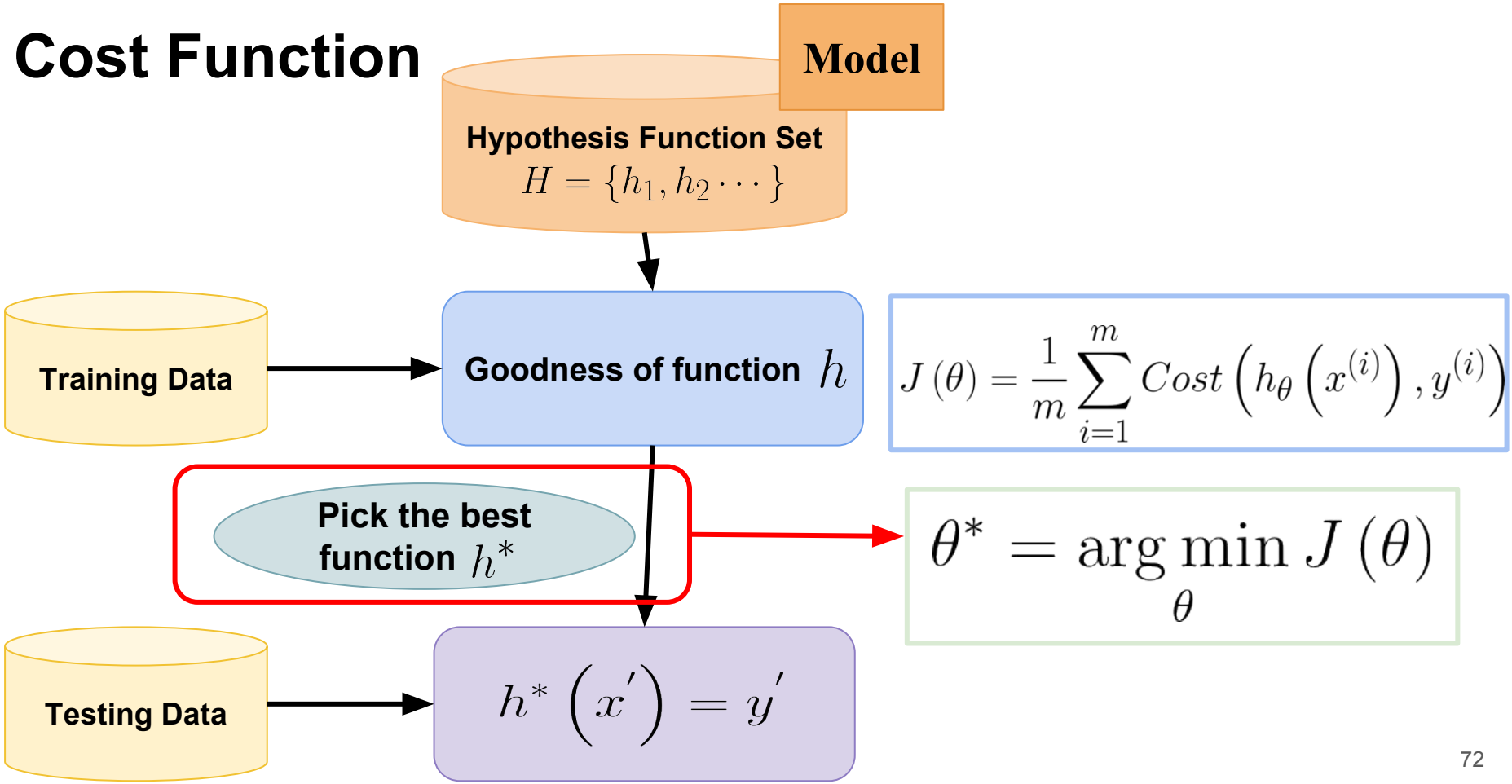
Machine Learning Step



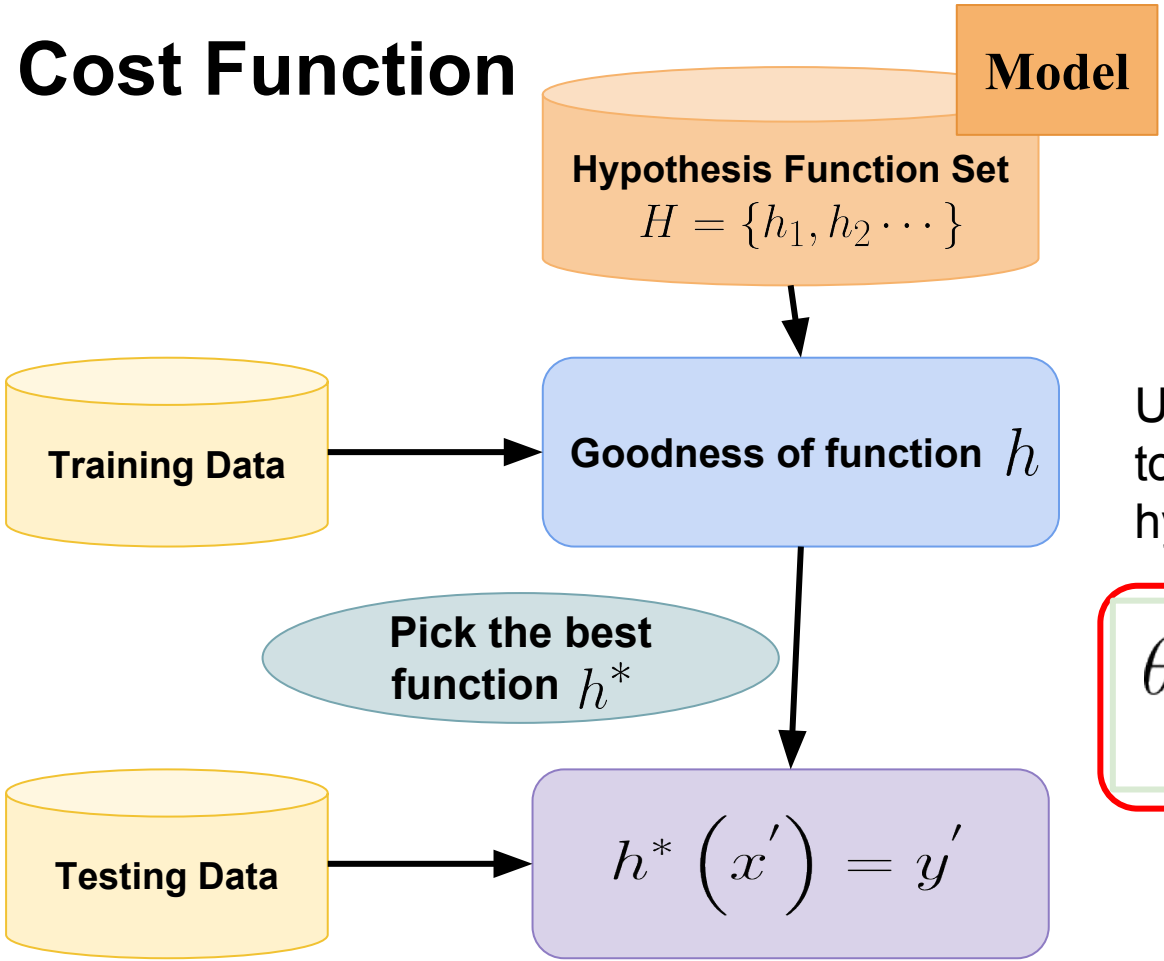
Cost Function



Cost Function



Cost Function



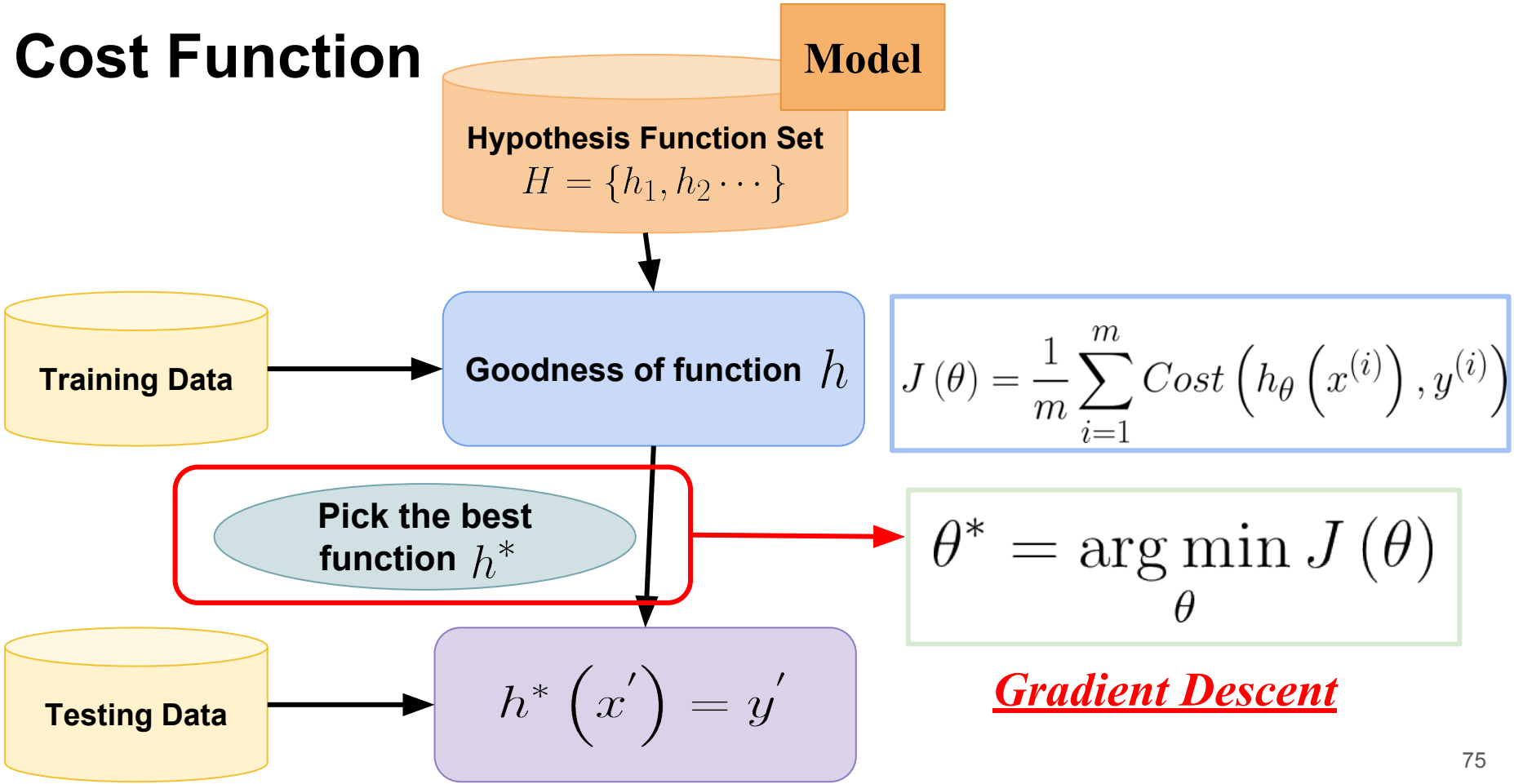
Use **optimization method** to find optimal parameters of hypothesis!

$$\theta^* = \arg\min_{\theta} J(\theta)$$

Optimization Method

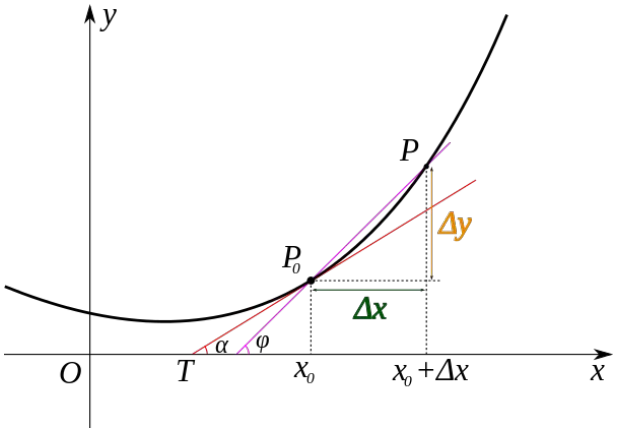
- There are many **optimization method** to find optimal parameters of hypothesis.
  - Gradient Descent/Ascent
  - Least Square Method
  - Linear programming/Quadratic Programming
  - Newton’s Method
  - Genetic Algorithms

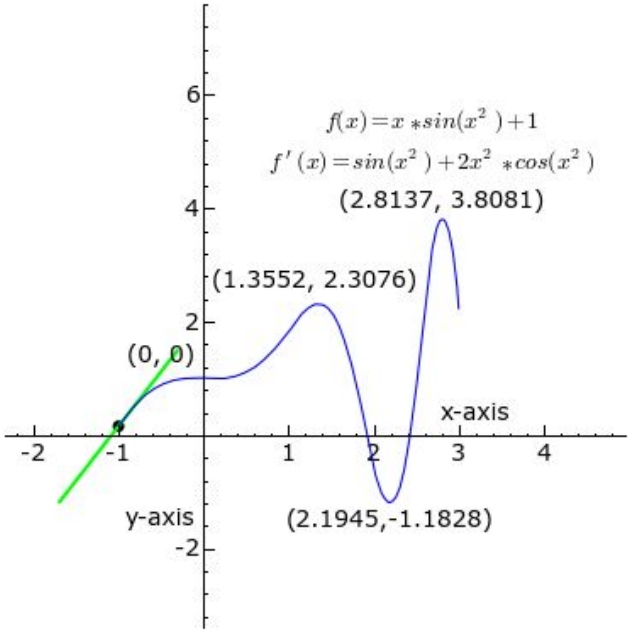
Cost Function



Gradient Descent

微分: $f'(x_0) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$



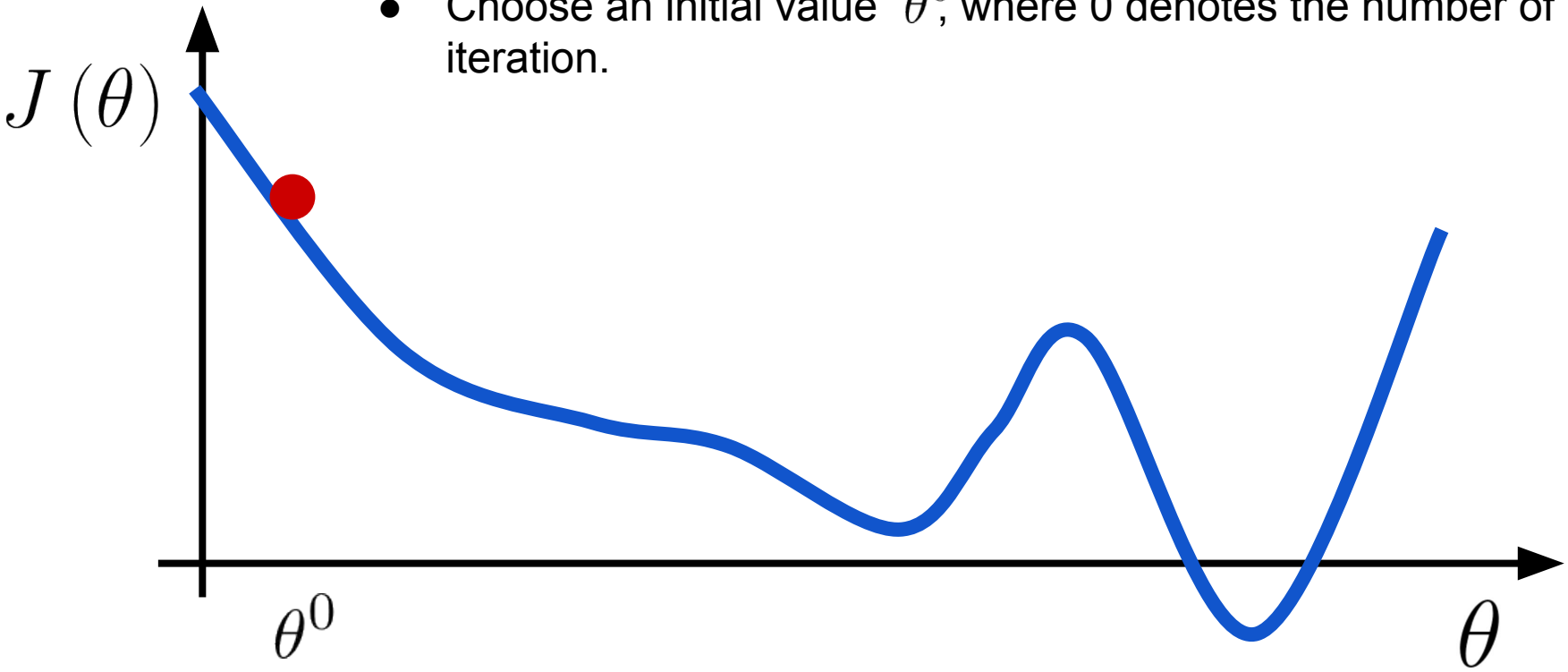


某一點的微分代表的是該點的斜率，也就是局部的變化量。

https://zh.wikipedia.org/wiki/%E5%AF%BC%E6%95%B0

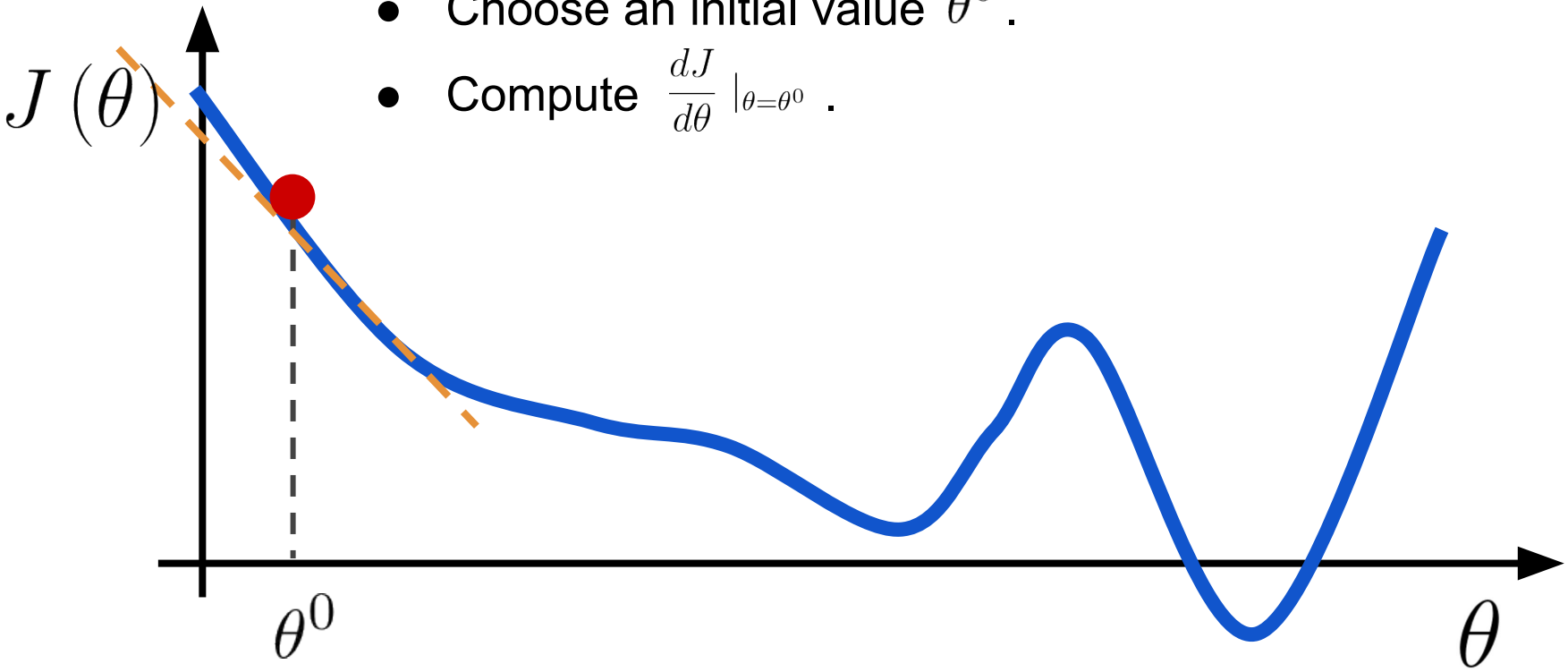
Gradient Descent

- Choose an initial value θ^0, where 0 denotes the number of iteration.



Gradient Descent

- Choose an initial value θ^0 .
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.

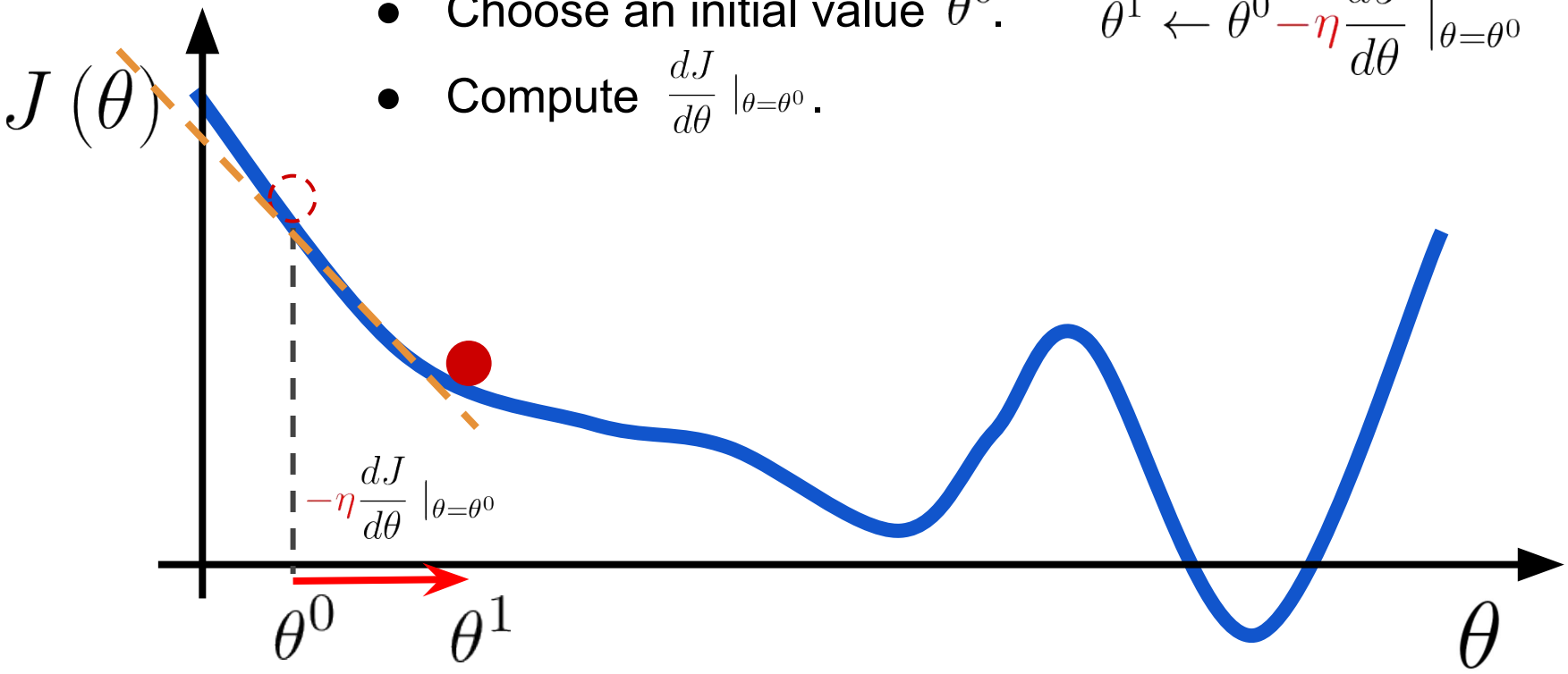


Gradient Descent

learning rate : η

- Choose an initial value θ^0.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.

$$\theta^1 \leftarrow \theta^0 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^0}$$

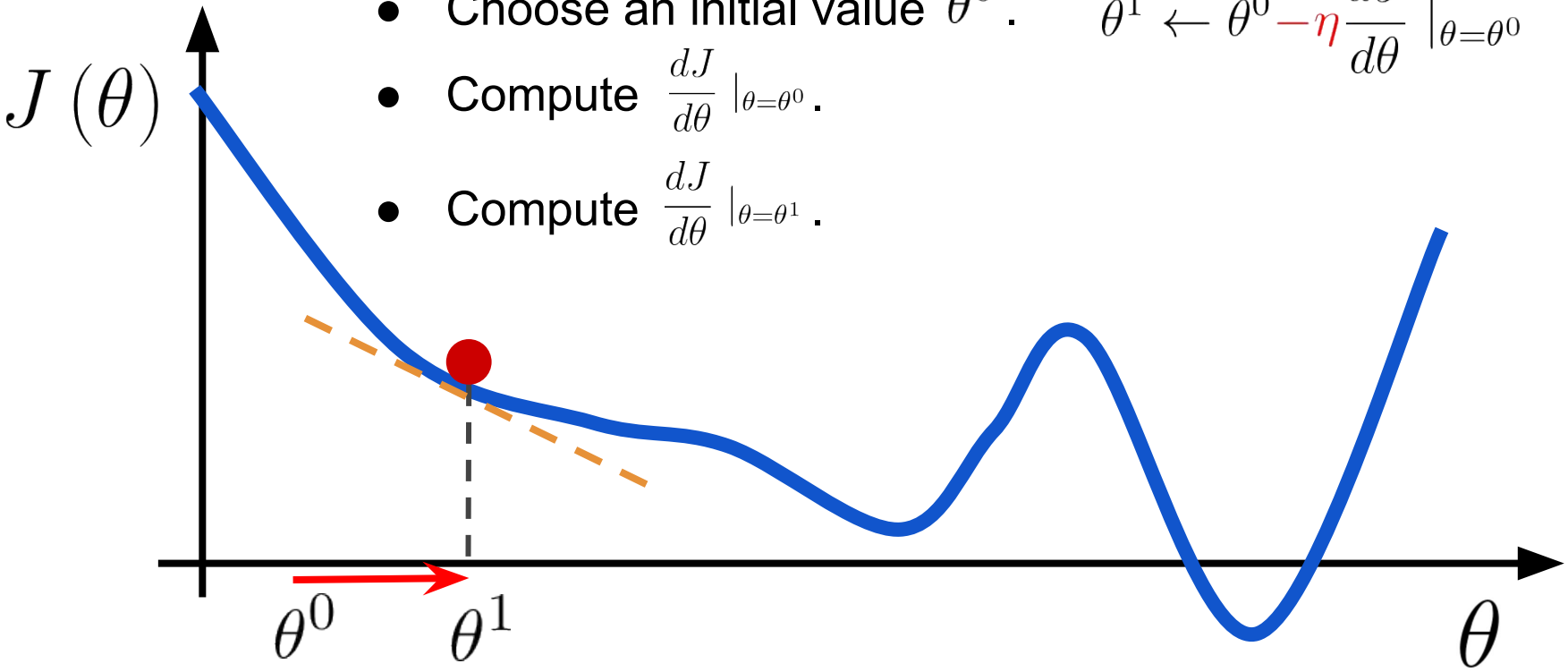


Gradient Descent

learning rate : η

- Choose an initial value θ^0 .
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^1}$.

$$\theta^1 \leftarrow \theta^0 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^0}$$



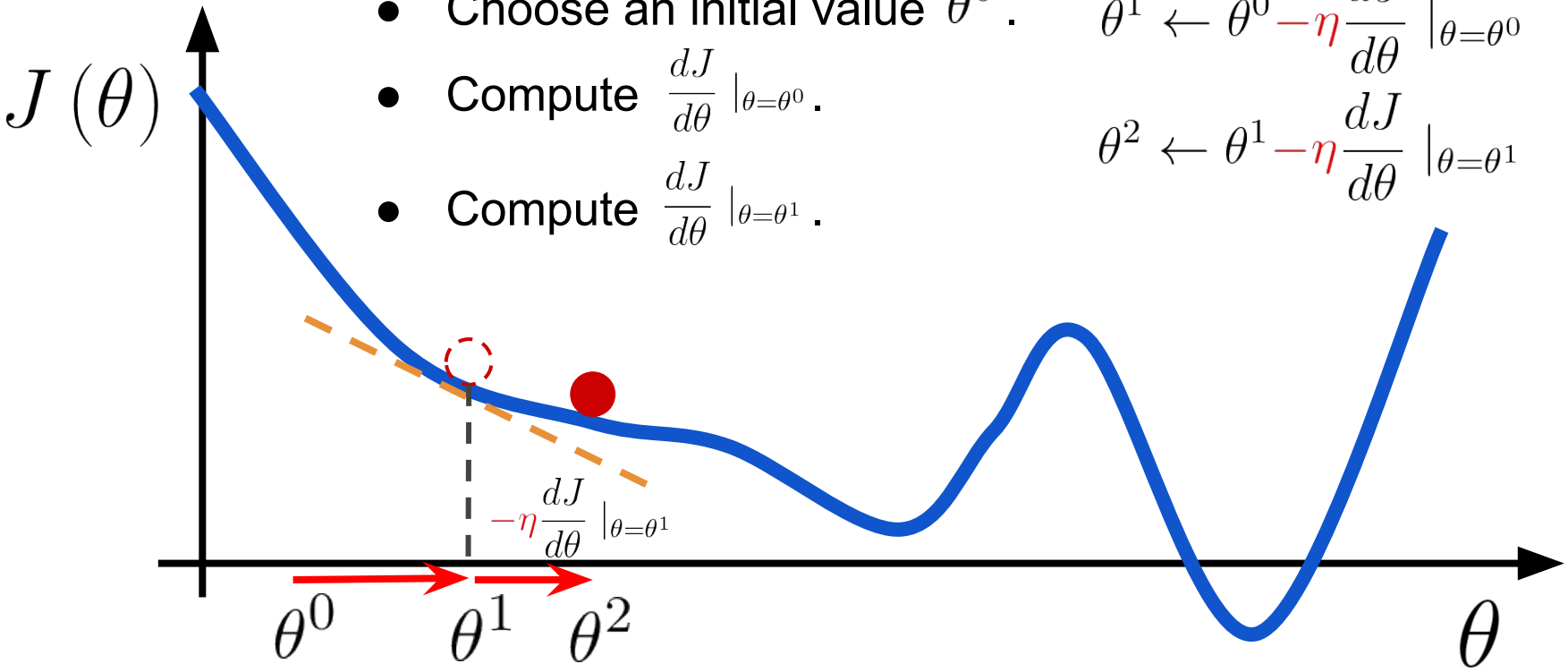
Gradient Descent

learning rate : η

- Choose an initial value θ^0 .
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^1}$.

$$\theta^1 \leftarrow \theta^0 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^0}$$

$$\theta^2 \leftarrow \theta^1 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^1}$$



Gradient Descent

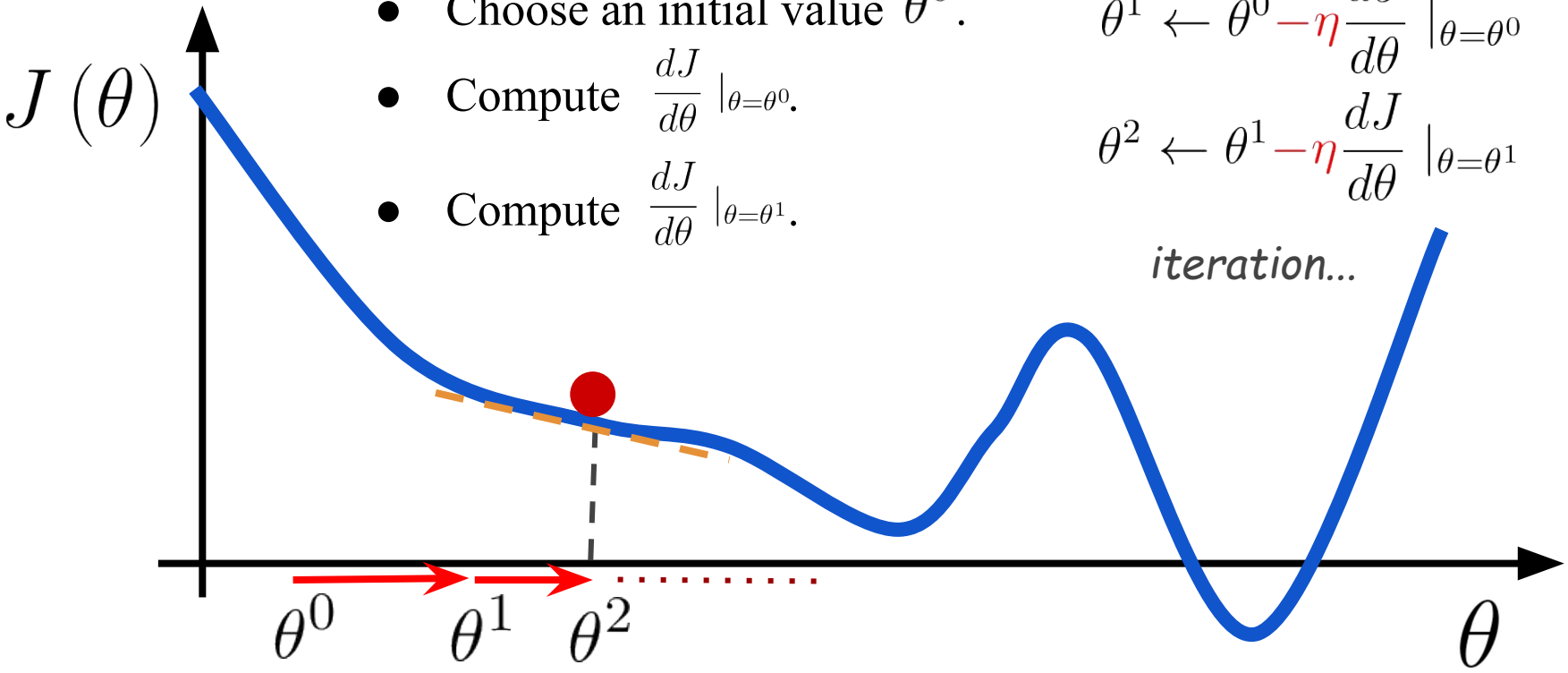
learning rate : η

- Choose an initial value θ^0.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^1}$.

$$\theta^1 \leftarrow \theta^0 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^0}$$

$$\theta^2 \leftarrow \theta^1 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^1}$$

iteration...



Gradient Descent

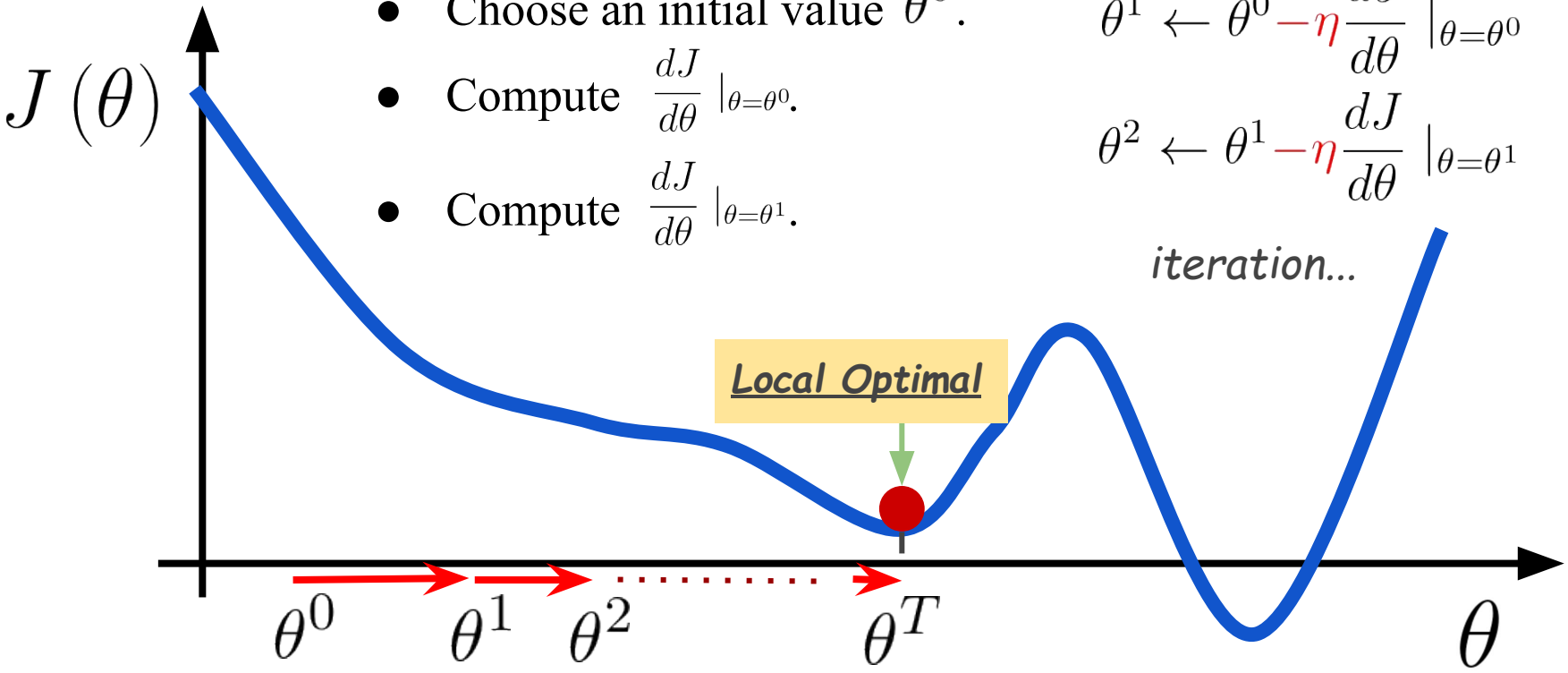
learning rate : η

- Choose an initial value θ^0.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^1}$.

$$\theta^1 \leftarrow \theta^0 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^0}$$

$$\theta^2 \leftarrow \theta^1 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^1}$$

iteration...



Gradient Descent

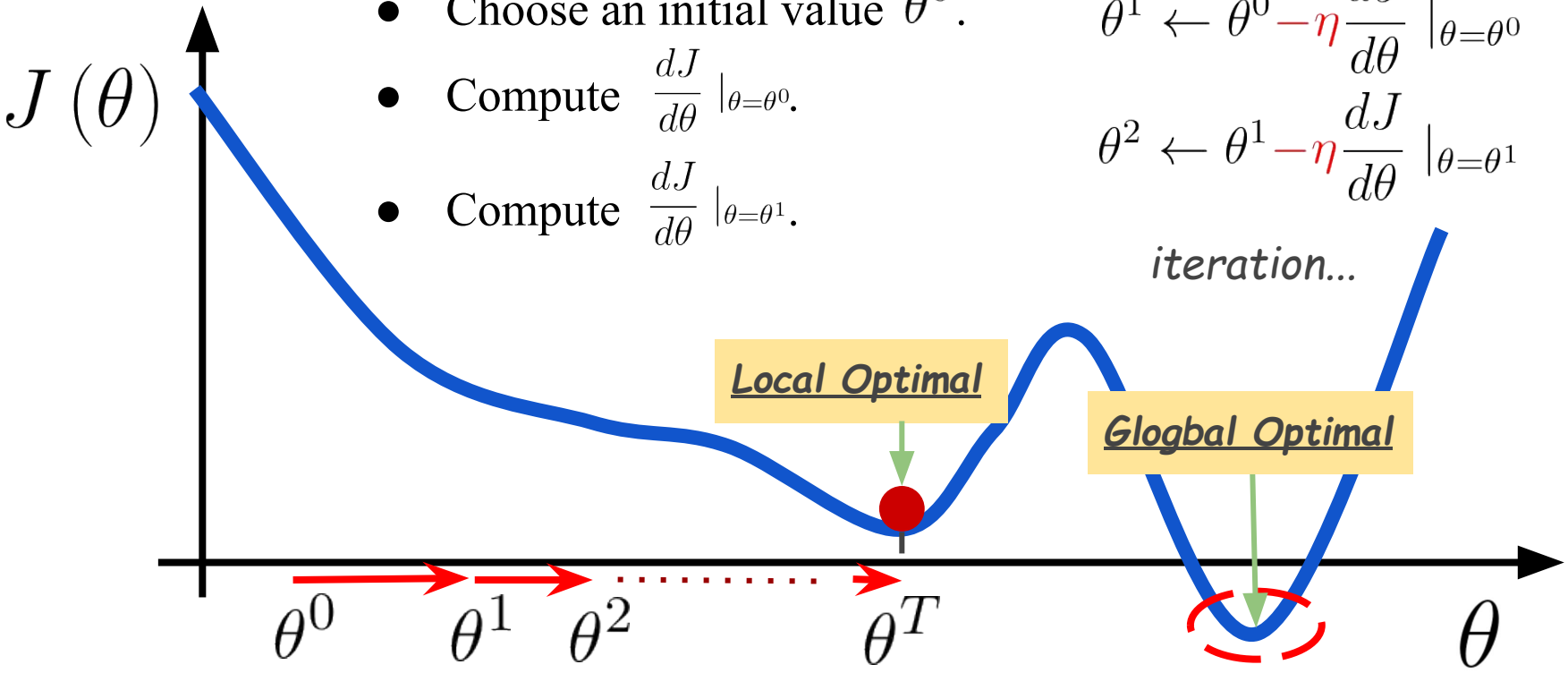
learning rate : η

- Choose an initial value θ^0.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$.
- Compute $\frac{dJ}{d\theta}|_{\theta=\theta^1}$.

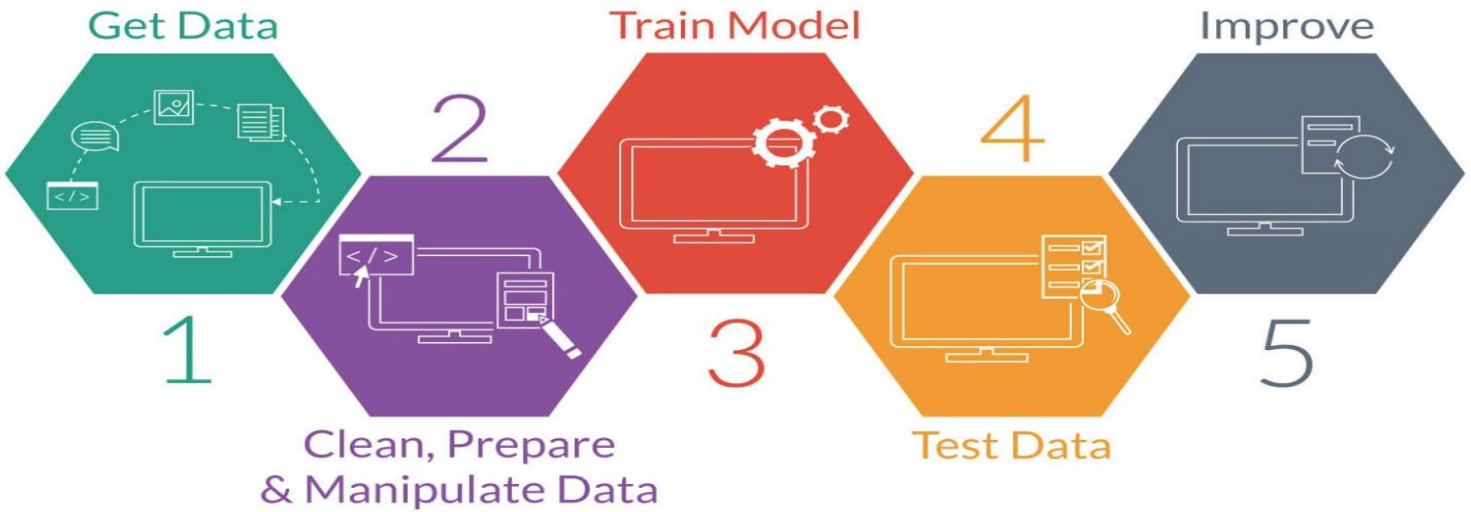
$$\theta^1 \leftarrow \theta^0 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^0}$$

$$\theta^2 \leftarrow \theta^1 - \eta \frac{dJ}{d\theta}|_{\theta=\theta^1}$$

iteration...

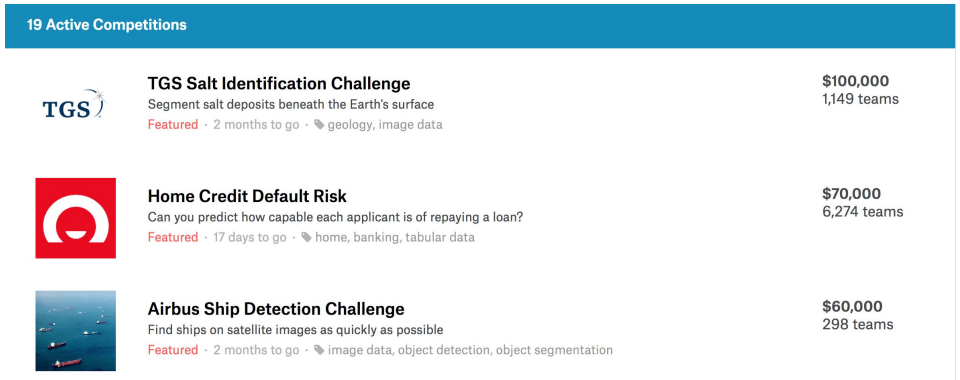


Machine Learning Workflow



https://www.linkedin.com/pulse/machine-learning-lets-learn-neeraj-sabharwal/

Get Data

- Data analysis competition platform
  - kaggle
- Open data
  - 政府資料開放平臺
  - 

Data Pre-processing

- Why should we pre-process data?

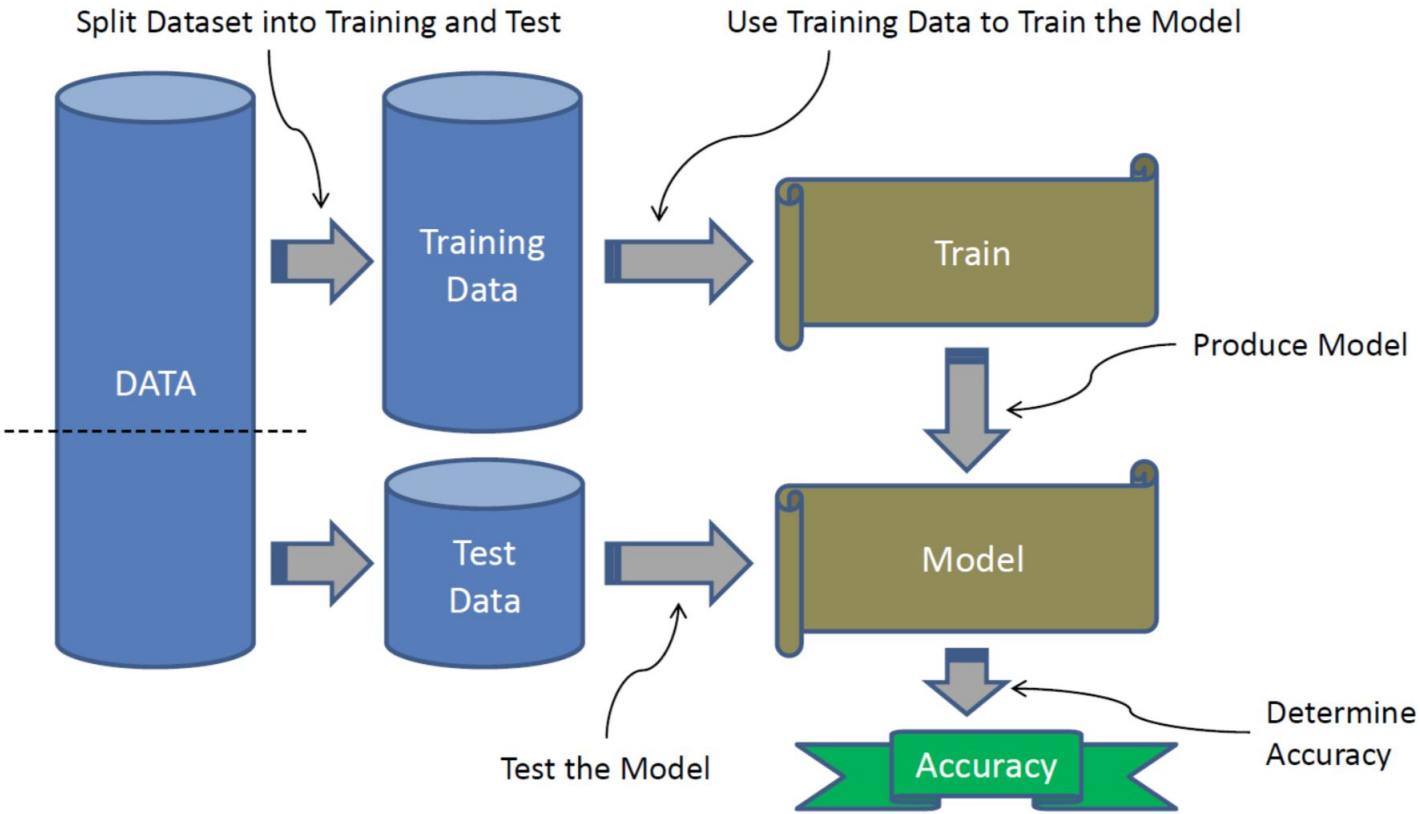
"Garbage in, garbage out."

Data Pre-processing

- Data type constraints: int, string, boolean
- Accuracy: outlier, noise…
- Completeness: missing value…
- Feature Scaling

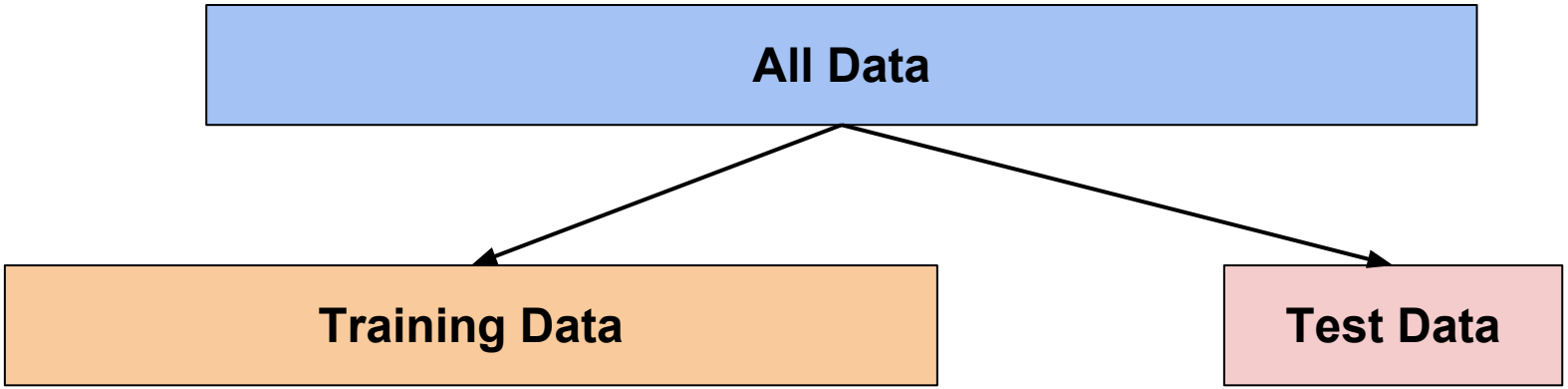
	0	1	2	3	4	5	6	7	8
0	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
5	5	116.0	74.0	NaN	NaN	25.6	0.201	30	0
6	3	78.0	50.0	32.0	88.0	31.0	0.248	26	1
7	10	115.0	NaN	NaN	NaN	35.3	0.134	29	0
8	2	197.0	70.0	45.0	543.0	30.5	0.158	53	1
9	8	125.0	96.0	NaN	NaN	NaN	0.232	54	1
10	4	110.0	92.0	NaN	NaN	37.6	0.191	30	0
11	10	168.0	74.0	NaN	NaN	38.0	0.537	34	1
12	10	139.0	80.0	NaN	NaN	27.1	1.441	57	0
13	1	189.0	60.0	23.0	846.0	30.1	0.398	59	1
14	5	166.0	72.0	19.0	175.0	25.8	0.587	51	1
15	7	100.0	NaN	NaN	NaN	30.0	0.484	32	1
16	0	118.0	84.0	47.0	230.0	45.8	0.551	31	1
17	7	107.0	74.0	NaN	NaN	29.6	0.254	31	1
18	1	103.0	30.0	38.0	83.0	43.3	0.183	33	0
19	1	115.0	70.0	30.0	96.0	34.6	0.529	32	1

Training & Test Data



(slides by Cheng-Te Li)

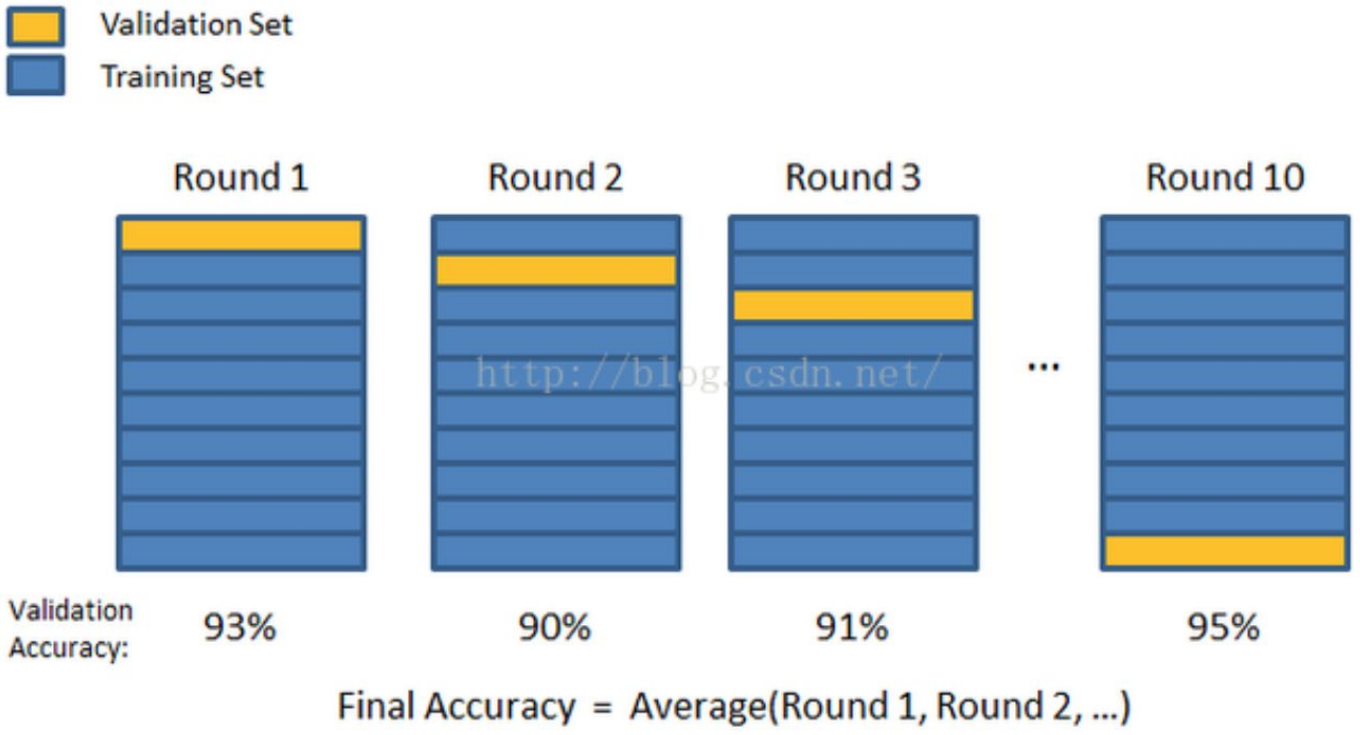
Training & Test Data



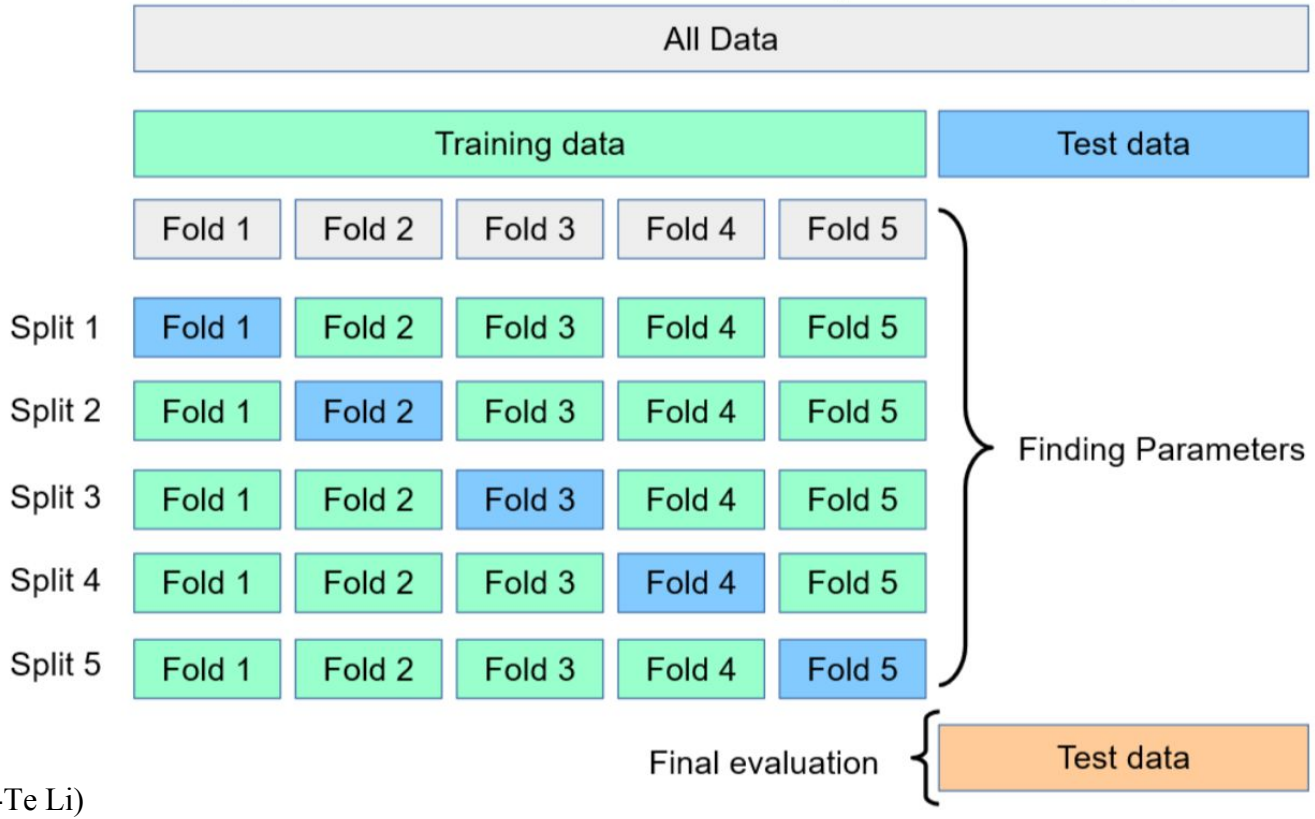
Training : Test
80% : 20%
75% : 25%

Cross-Validation

- Please look at the following diagram which clearly shows how the k-fold CV works:



Cross-Validation



(slides by Cheng-Te Li)

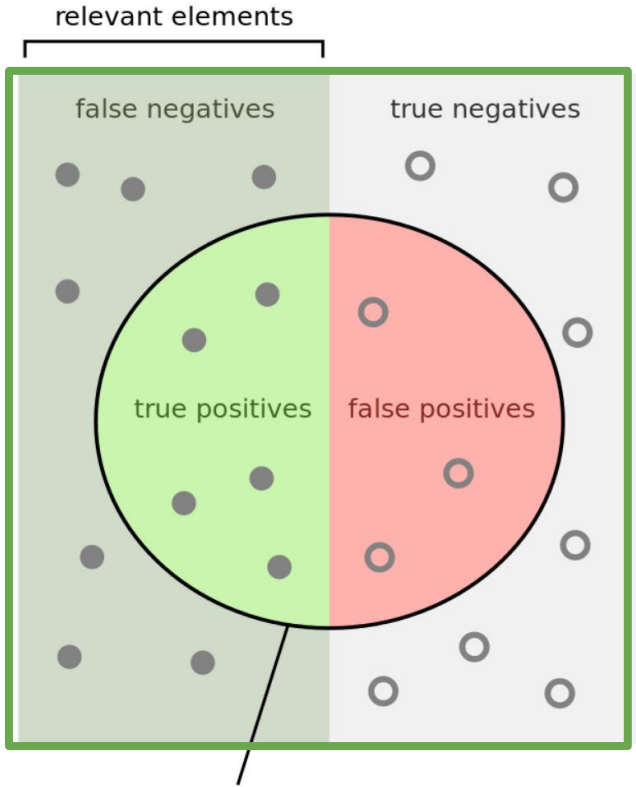
Evaluation

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR$$

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} = 1 - ACC$$

Evaluation



= what model says true

How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

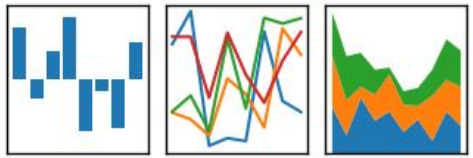
$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Data Science in Python

matplotlib

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

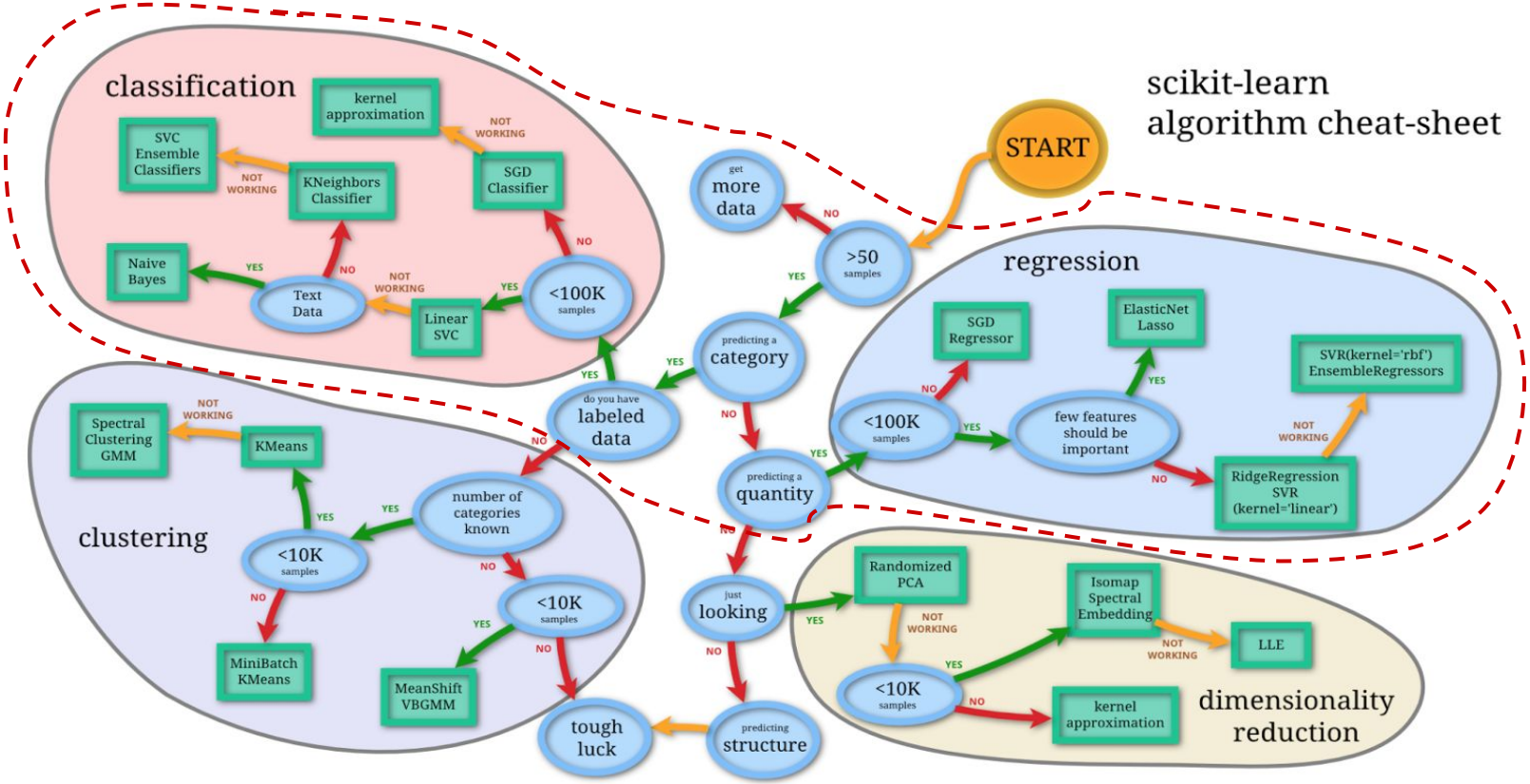


SciPy

NumPy

scikit learn

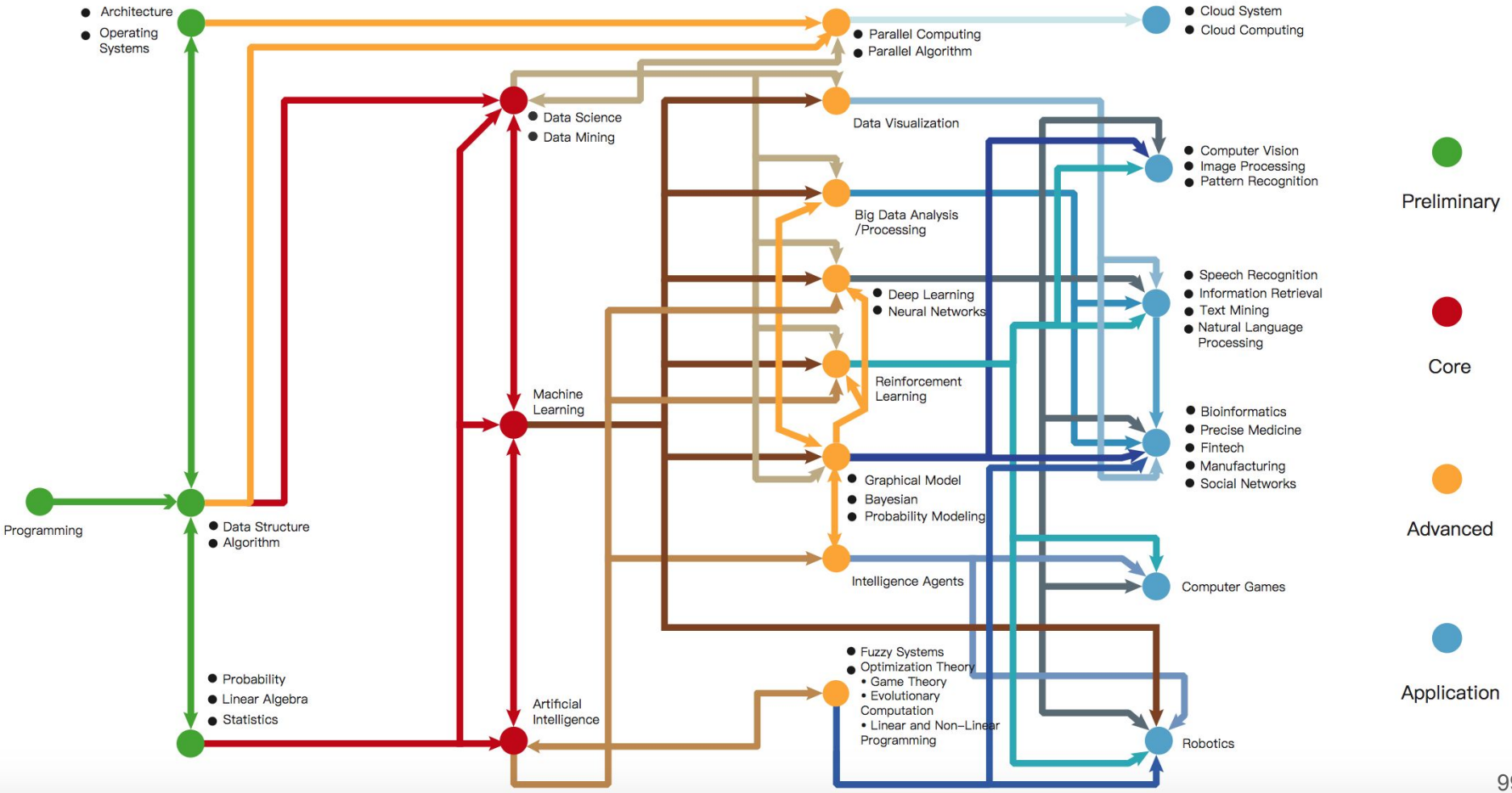
Scikit-Learn



Thank you for listening.

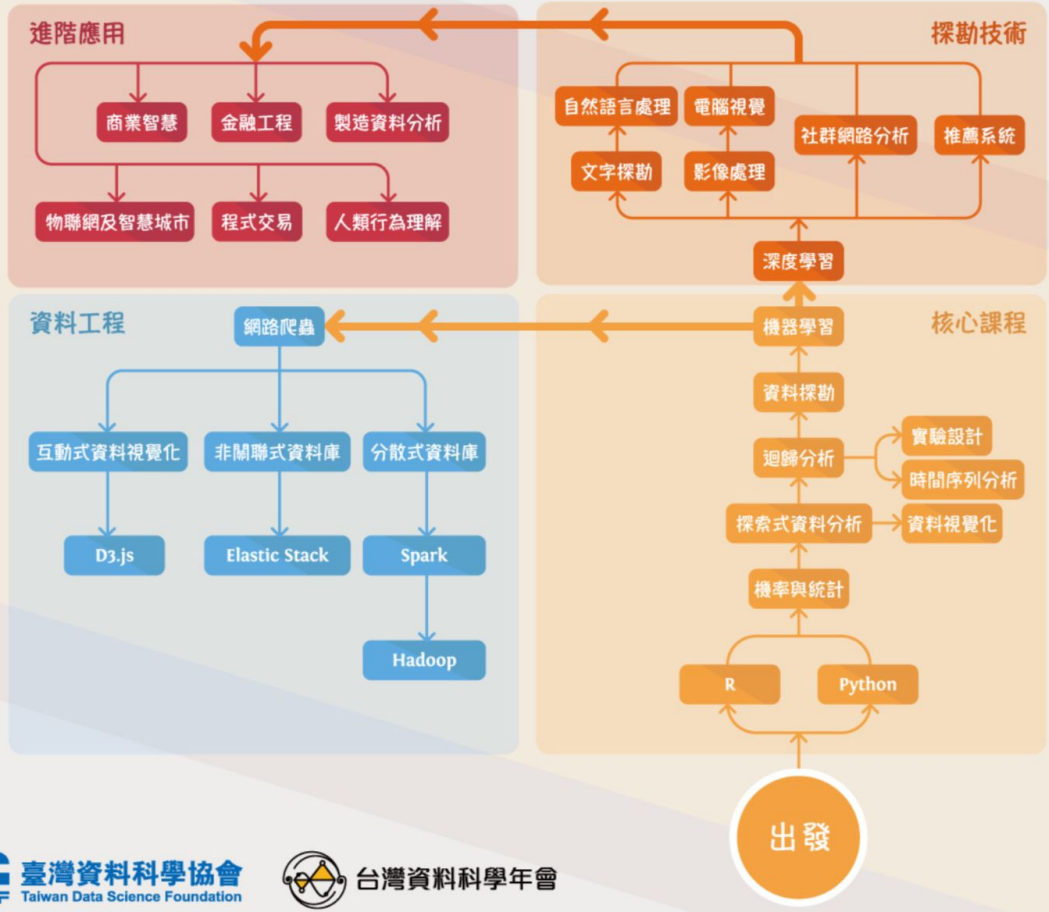
Reference

- Hung-yi Lee Course
- Andrew Ng Course
- Cheng-Te Li Course
- Tinglo’s figure


Architecture
Operating Systems
Programming
Data Structure
Algorithm
Probability
Linear Algebra
Statistics
Data Science
Data Mining
Machine Learning
Artificial Intelligence
Parallel Computing
Parallel Algorithm
Data Visualization
Big Data Analysis /Processing
Deep Learning
Neural Networks
Reinforcement Learning
Graphical Model
Bayesian
Probability Modeling
Intelligence Agents
Fuzzy Systems
Optimization Theory
• Game Theory
• Evolutionary Computation
• Linear and Non-Linear Programming
Cloud System
Cloud Computing
Computer Vision
Image Processing
Pattern Recognition
Speech Recognition
Information Retrieval
Text Mining
Natural Language Processing
Bioinformatics
Precise Medicine
Fintech
Manufacturing
Social Networks
Computer Games
Robotics
Preliminary
Core
Advanced
Application

資料科學學習地圖

http://foundation.datasci.tw/learning-map/



臺灣資料科學協會 Taiwan Data Science Foundation TDSF

台灣資料科學年會