IOT DATA ANALYSIS
(INTRODUCTION TO MACHINE LEARNING)

Assoc Prof Dr Chaodit Aswakul
Wireless Network and Future Internet Research Unit
Chulalongkorn University, Thailand

What is Machine Learning?

[REF: Jake VanderPlas, Python Data Science Handbook
Essential Tools for Working with Data, O'Reilly, 2017]

- Means of building **models** of data
- **Learning** ability by tuning **model parameters**
- **To fit** with previously seen data
- **To predict** and **understand** future unseen data

*"To learn **without** explicitly being **programmed**"*

ML Categories

- **Supervised learning**

  Modelling **relationship** between **input features** of data and **output labels** associated with data

- **Unsupervised learning**

  Modelling input features of data **without** reference to any **output labels**

- **Reinforcement learning**

  Semi-supervised learning with only **incomplete label** available in terms of **state, action, reward**

[REF: Jake VanderPlas, Python Data Science Handbook
Essential Tools for Working with Data, O'Reilly, 2017]

SUPERVISED LEARNING

Data: (x, y)
x is data, y is label

Goal: Learn function to map
$$x \rightarrow y$$

Apple example:

This thing is an apple.

[REF: ©MIT 6.S191
Introduction to Deep Learning
introtodeeplearning.com]

IF output target is real-valued
THEN “Regression Analysis”

<u>Example</u>

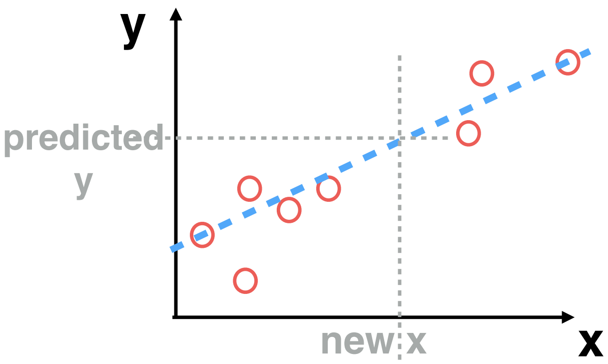
Wanted to predict

y = house price

from input feature vector

x = (x1, x2, x3)

x1 = #bed rooms (integers)
x2 = house area (reals)
x3 = location (strings)



SUPERVISED
LEARNING

IF output target belongs to a few discrete set, or *classes*

THEN “Classification Analysis”

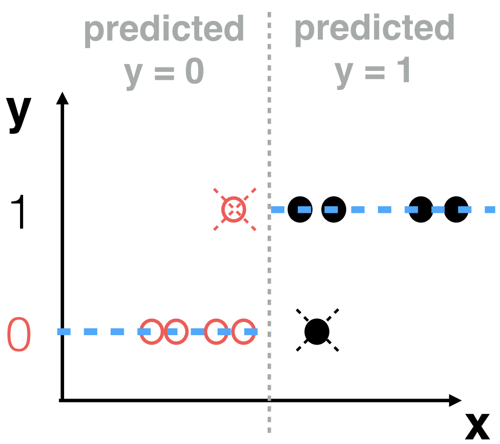
<u>Example</u>

Wanted to predict

y = 1, system is UP
y = 0, system is DOWN

from input features

x1, x2, … = system measurements



SUPERVISED
LEARNING

UNSUPERVISED LEARNING

[REF: ©MIT 6.S191
Introduction to Deep Learning
introtodeeplearning.com]

Data: x

x is data, no labels!

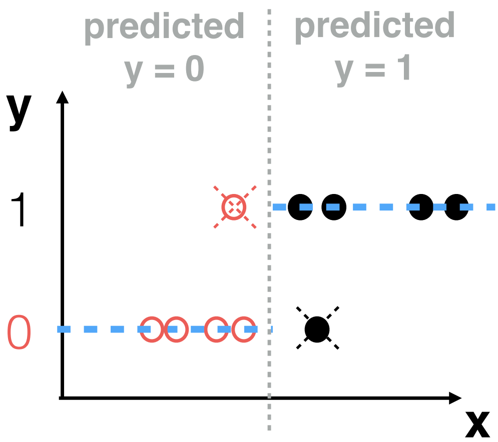
Goal: Learn underlying structure

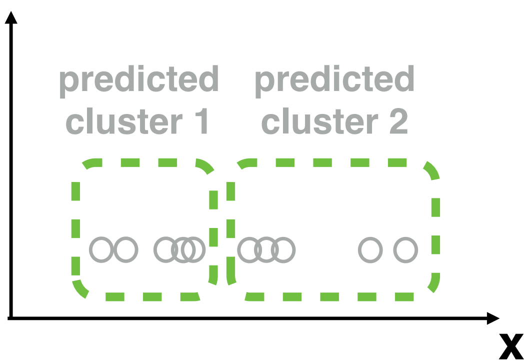
Apple example:

This thing is like the other thing.

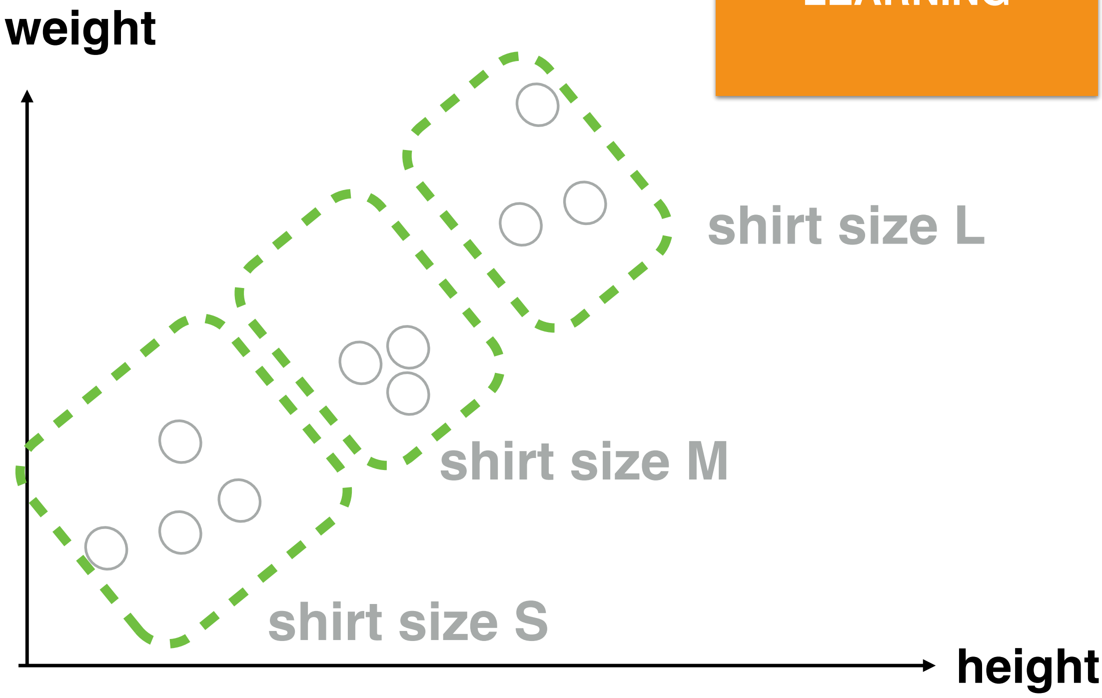
SUPERVISED LEARNING

UNSUPERVISED LEARNING





UNSUPERVISED LEARNING



REINFORCEMENT LEARNING

[REF: ©MIT 6.S191
Introduction to Deep Learning
introtodeeplearning.com]

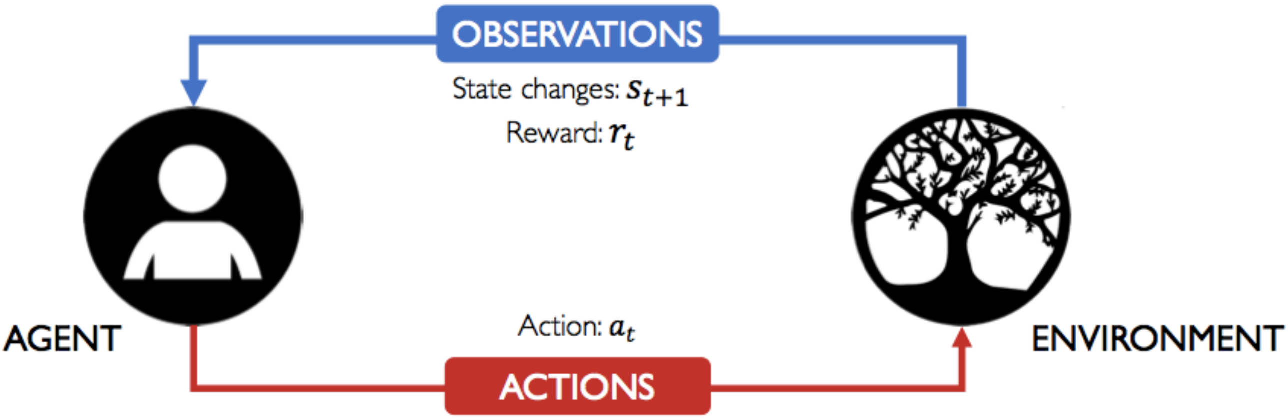
Data: state-action pairs

Goal: Maximize future rewards over many time steps

Apple example:

Eat this thing because it will keep you alive.

Reinforcement Learning



Reward: feedback that measures the success or failure of the agent's action.

[REF: ©MIT 6.S191
Introduction to Deep Learning
introtodeeplearning.com]

AlphaGo

https://youtu.be/8tq1C8spV_g Movie trailer 1.30 mins

https://youtu.be/jGyCsVhtW0M Documentary 1.30 hrs

AlphaGo

”The IBM chess computer Deep Blue, which famously beat grandmaster Garry Kasparov in 1997, was explicitly programmed to win at the game.

But AlphaGo was not preprogrammed to play Go.

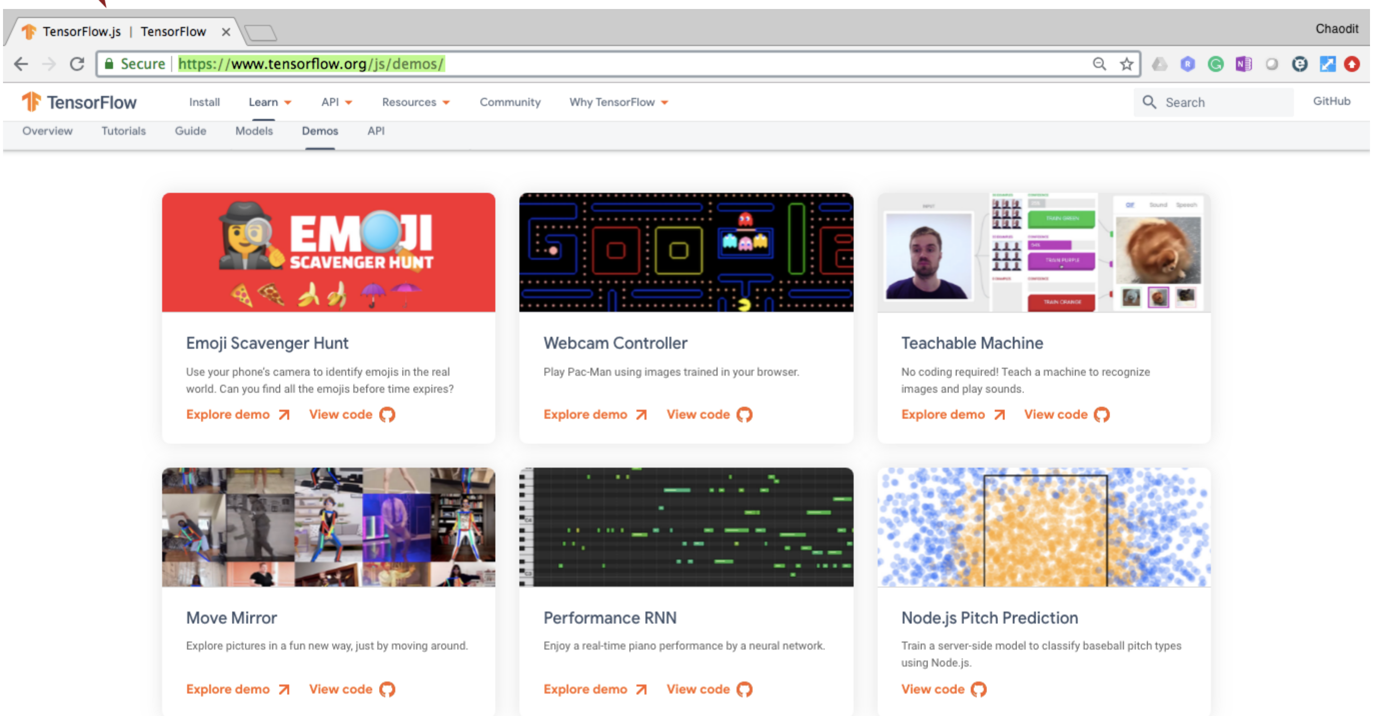
Rather, it learned using a general-purpose algorithm that allowed it to interpret the game’s patterns

in a similar way to how a DeepMind program learned to play 49 different arcade games”

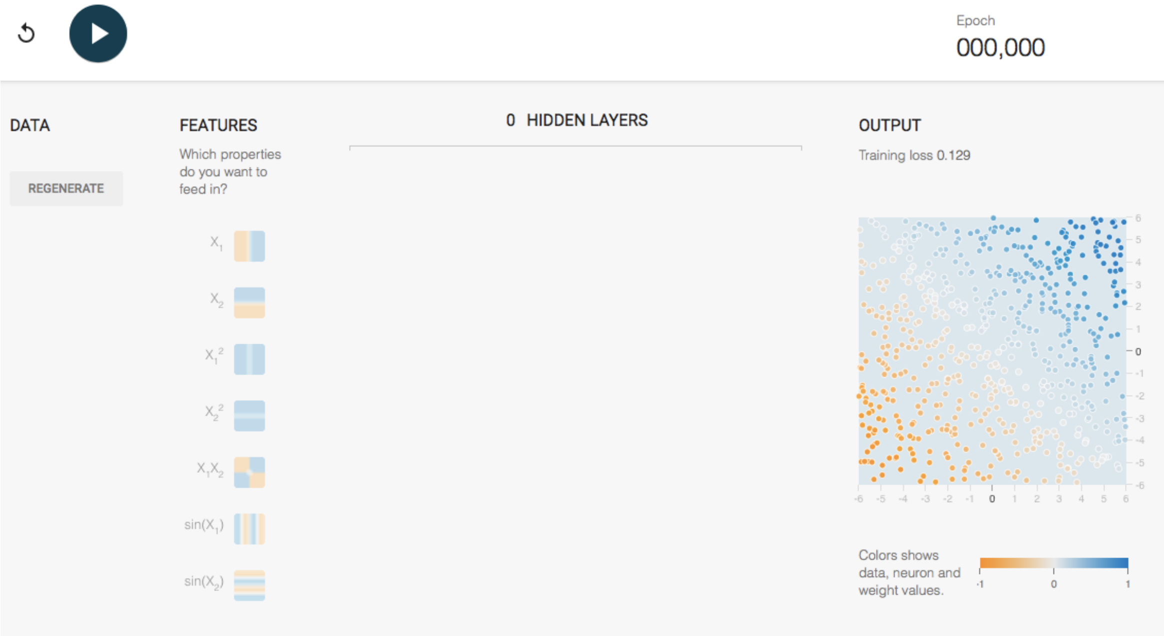
REF: Nature 529, 445–446 (28 January 2016)
https://www.nature.com/news/google-ai-algorithm-masters-ancient-game-of-go-1.19234

Hands-On Experiment

https://www.tensorflow.org/js/demos/



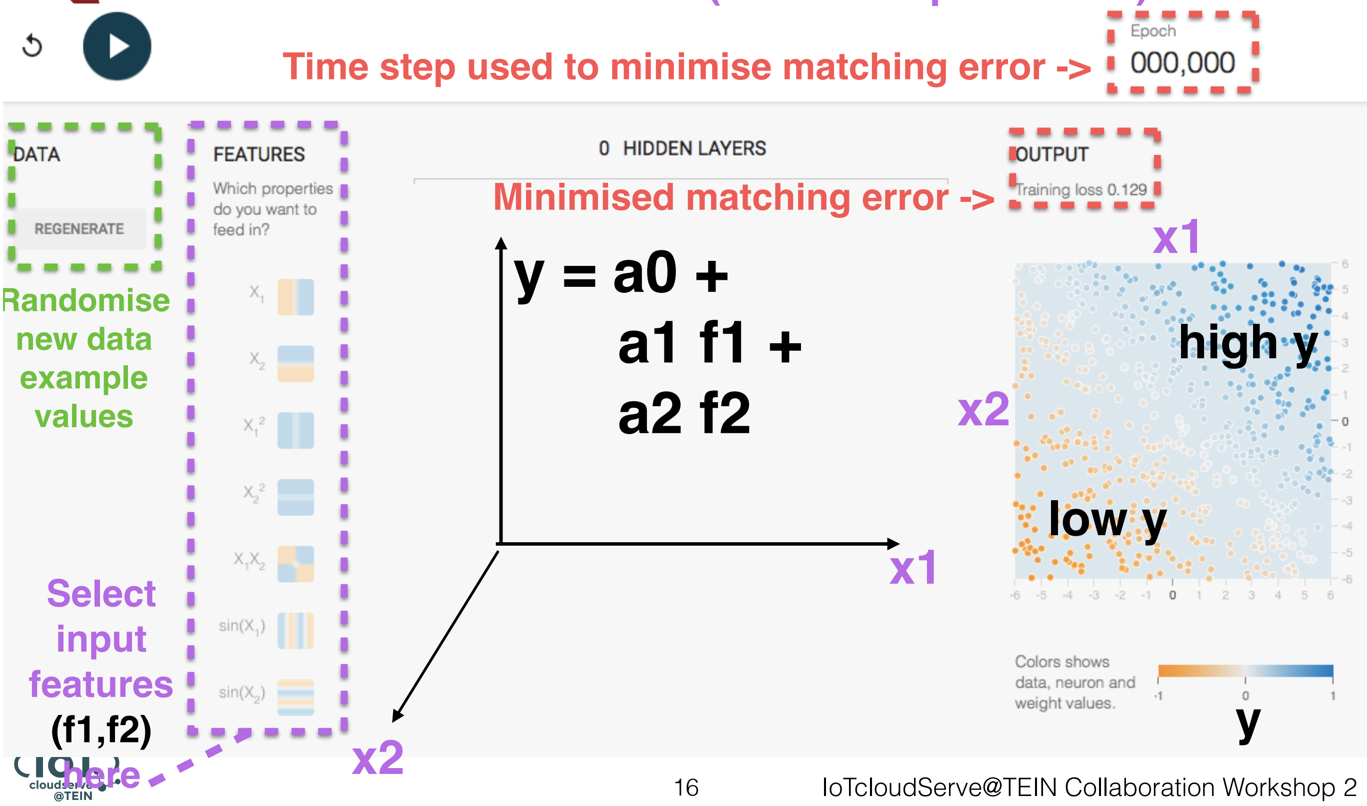
Linear Regression Experiment



Enter this experiment at bit.ly/2TQYYvf

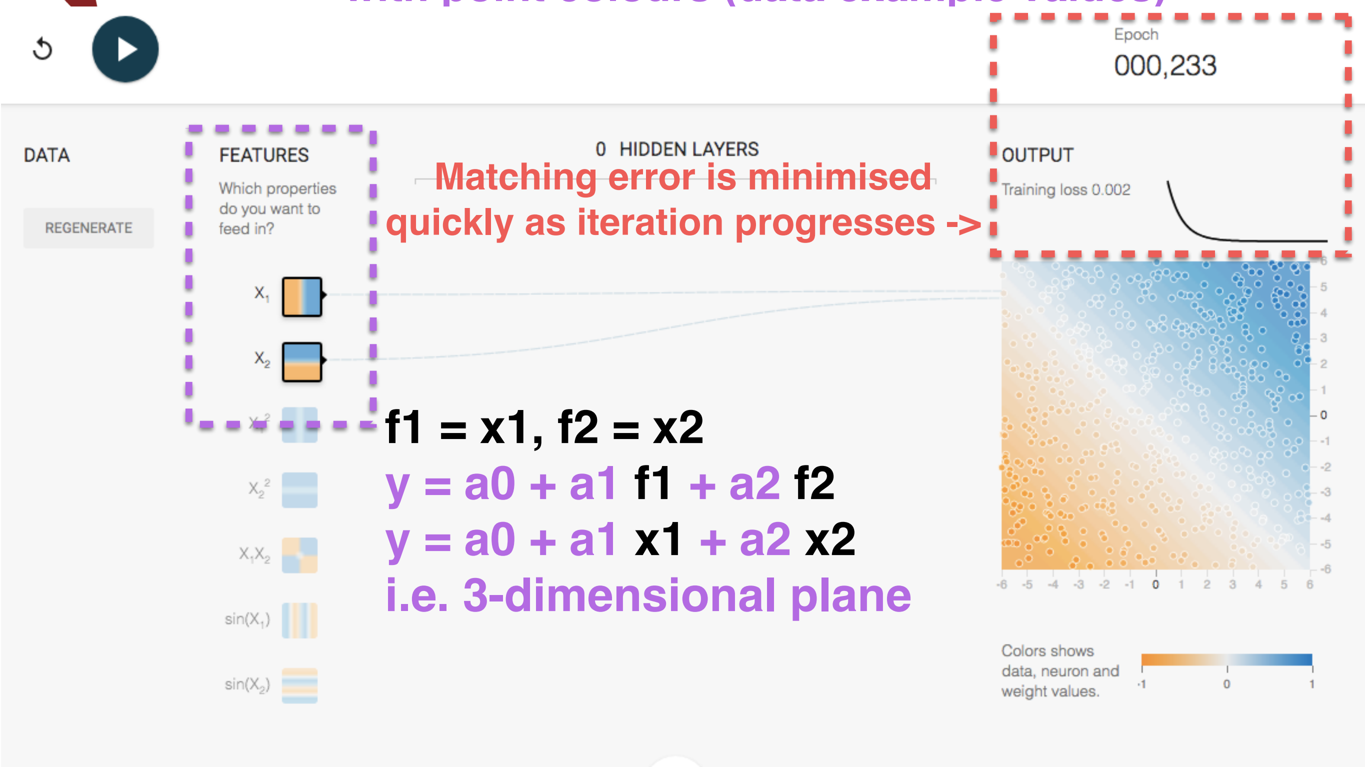
Regression Experiment

Goal: match background colours (predicted values) with dot colours (data example values)



Regression Experiment

Goal: match background colours (predicted values) with point colours (data example values)



Matching error is minimised quickly as iteration progresses ->

f1 = x1, f2 = x2

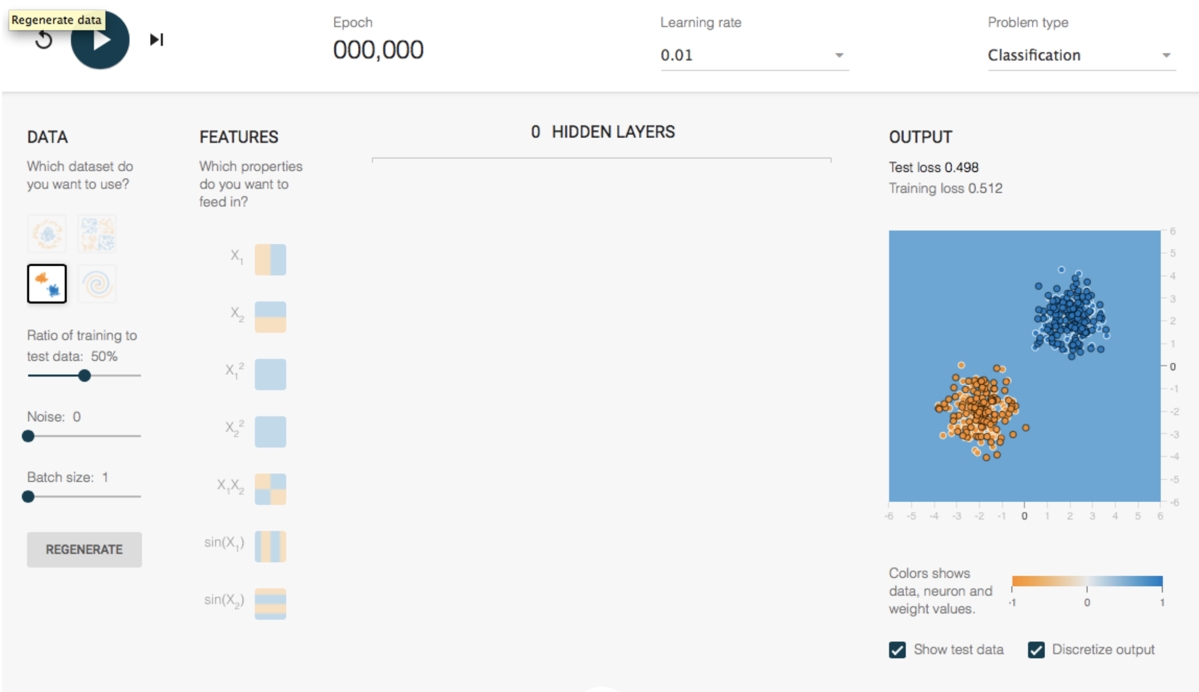
y = a0 + a1 f1 + a2 f2

y = a0 + a1 x1 + a2 x2

i.e. 3-dimensional plane

Classification Experiment

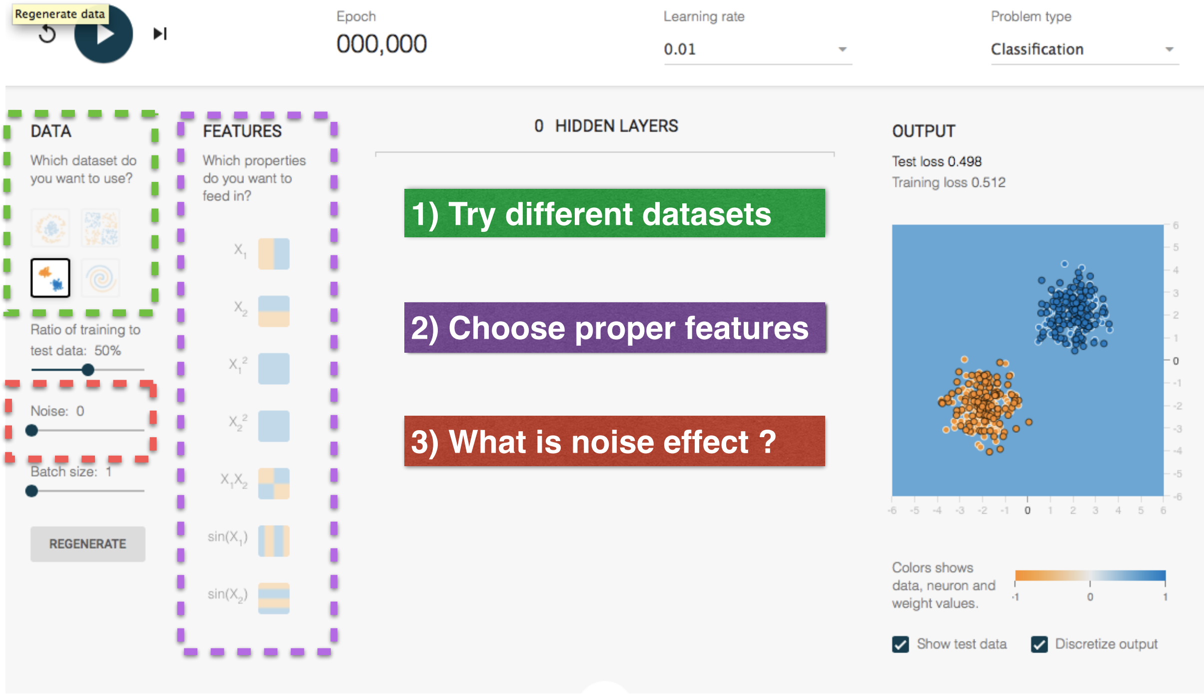
Goal: match <u>discretised</u> background colours (predicted values) with dot colours (data example values: -1 “orange” or 1 “blue”)



Enter this experiment at bit.ly/2OIT2Ep

Classification Experiment

Goal: match <u>discretised</u> background colours (predicted values) with dot colours (data example values: -1 "orange" or 1 "blue")



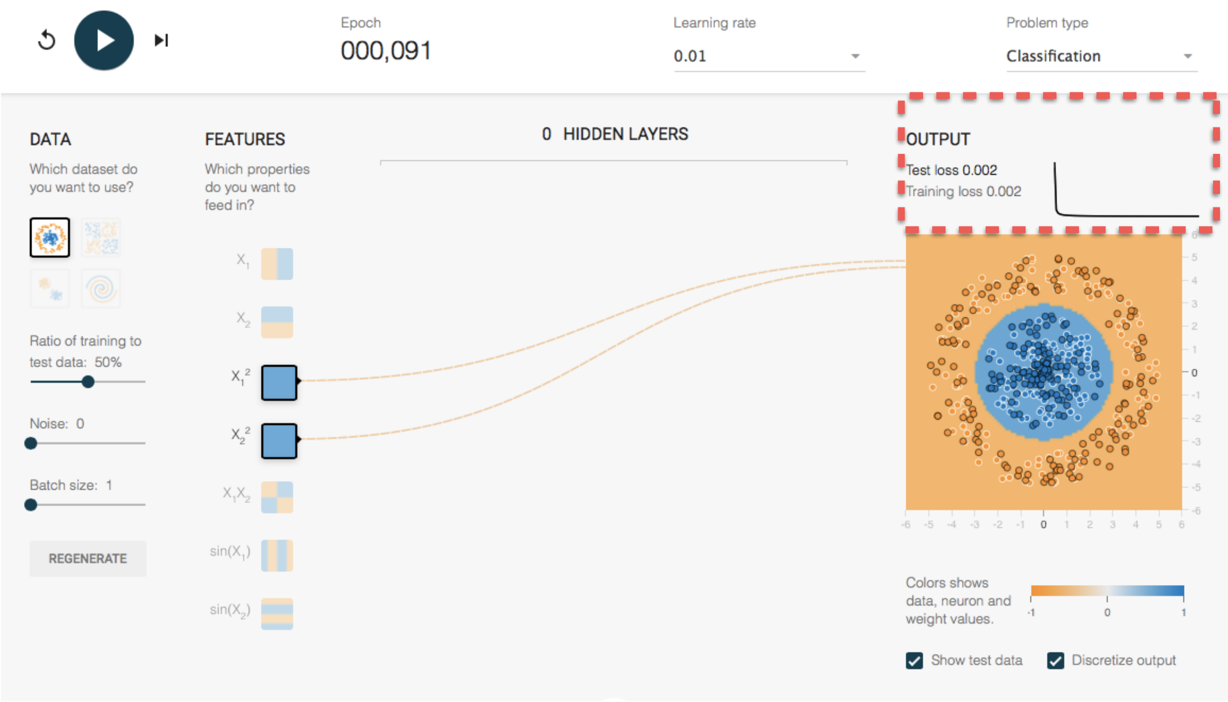
Enter this experiment at bit.ly/2OlT2Ep

Classification Experiment

Data: Two Gaussian Distributions (No Noise)

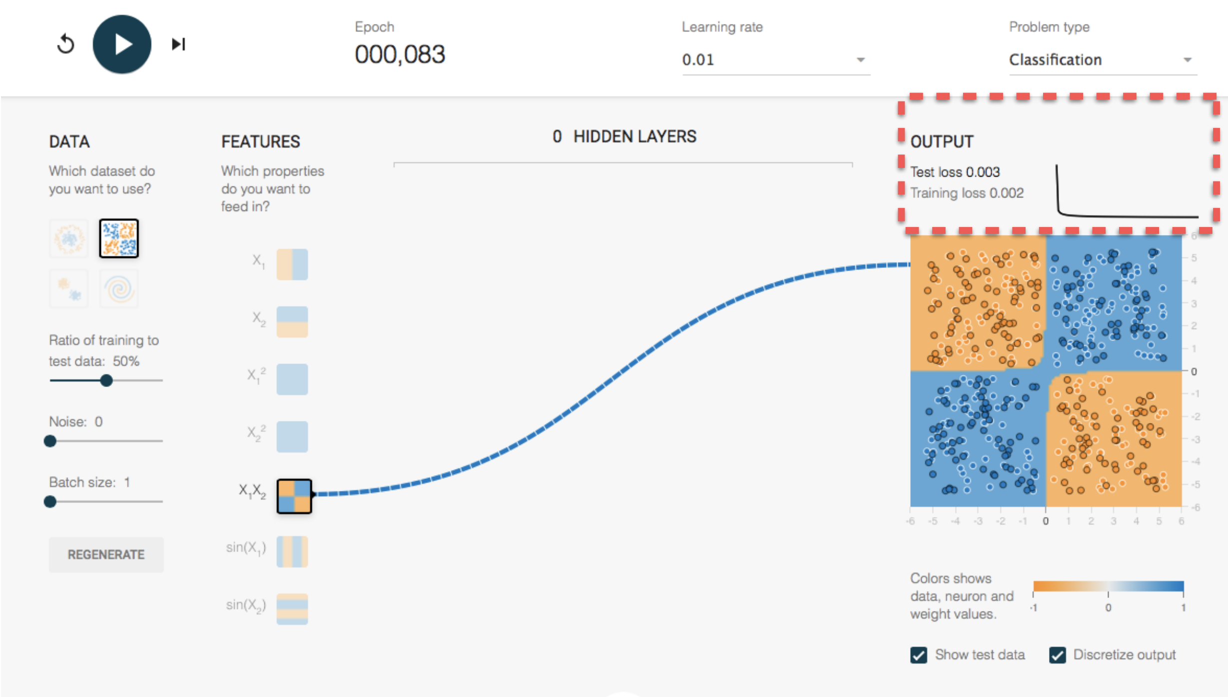
Classification Experiment

Circle Data (No Noise)



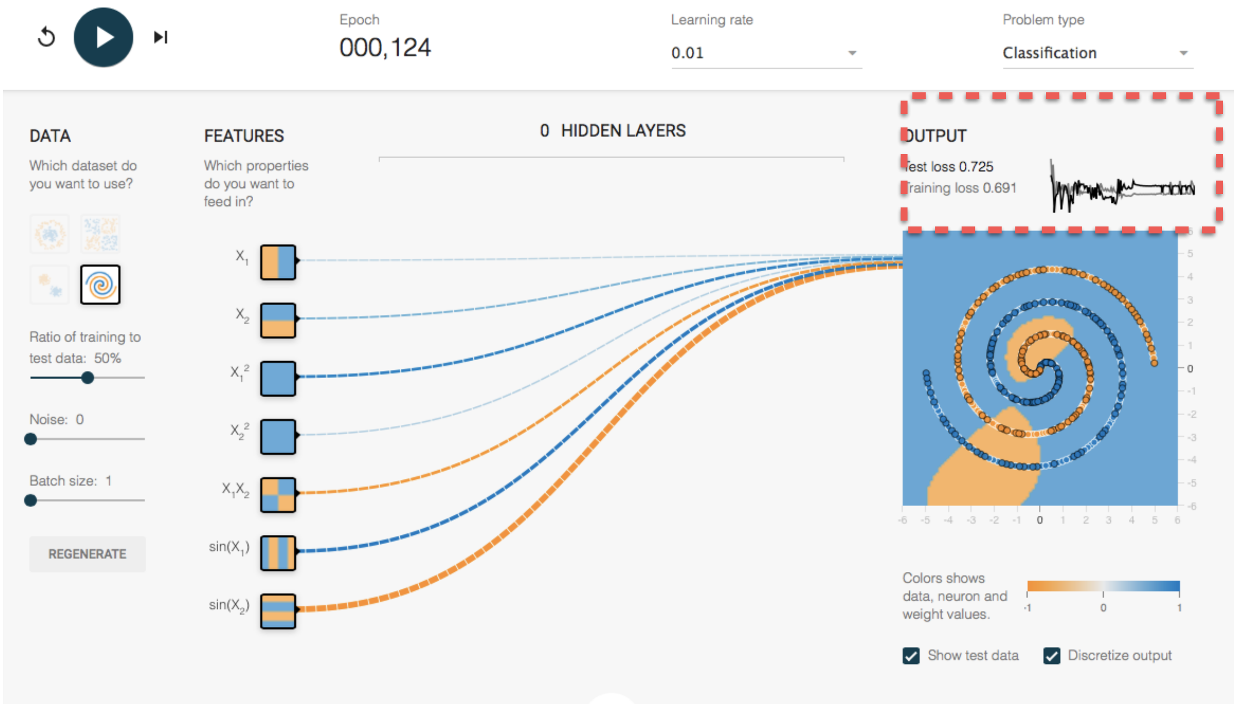
Classification Experiment

XOR Data (No Noise)



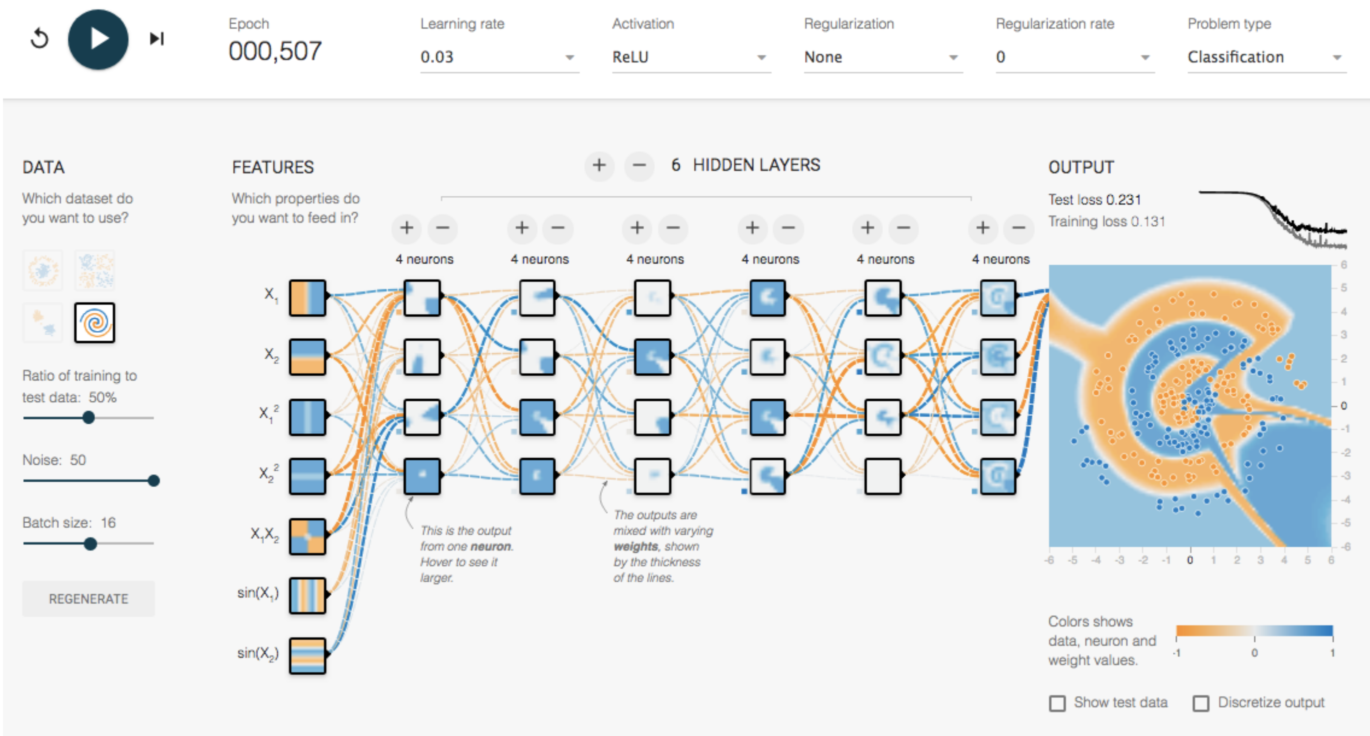
Classification Experiment

Spiral Data (No Noise)



Artificial (Deep) Neural Network Experiment

Spiral Data (No Noise)



NEXT: Scikit-Learn Package Python Code Examples

Import virtual machine into Virtual Box. Turn it on.

Open Jupyter notebook, in terminal, just type

>> jupyter notebook

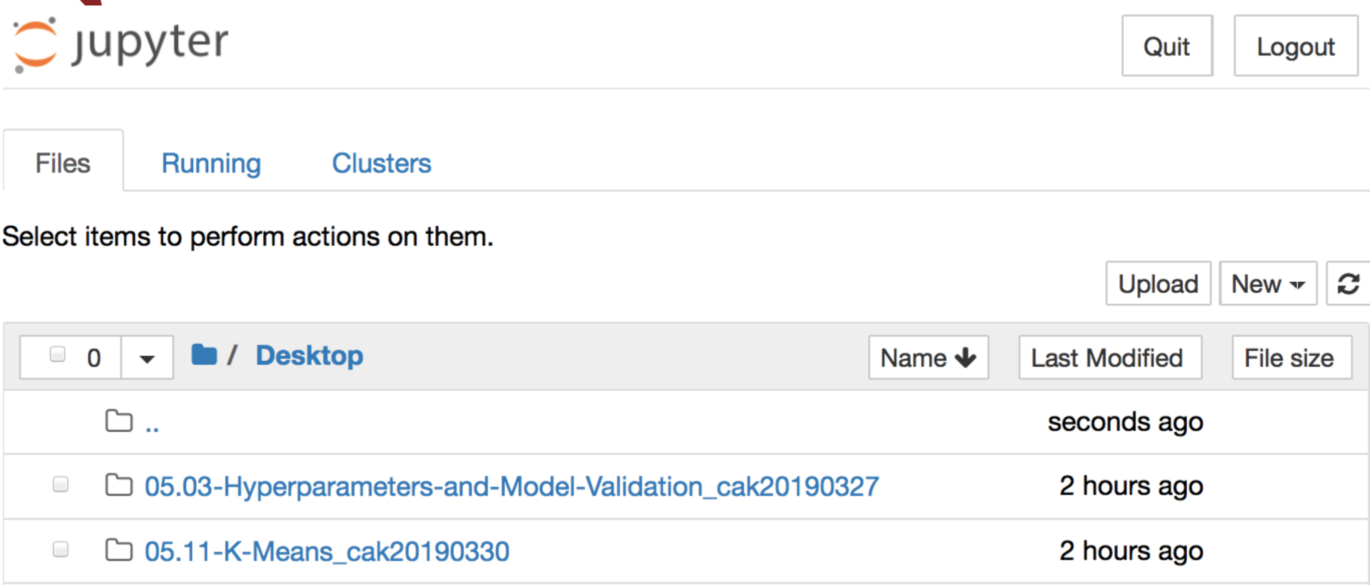
Browse to two provided folders on Desktop:

- 05.03-Hyperparameters-and-Model-Validation_cak20190327
- 05.11-K-Means_cak20190330

Click to open .ipynb file to load python codes

REF: Jake VanderPlas, Python Data Science Handbook Essential Tools for Working with Data, O'Reilly, 2017
https://jakevdp.github.io/PythonDataScienceHandbook

jupyter notebook



jupyter notebook

File Edit View Insert Cell Kernel Help Trusted Python 3

Run Markdown

This notebook contains an excerpt from the [Python Data Science Handbook](#) by Jake VanderPlas; the content is available [on GitHub](#).

The text is released under the [CC-BY-NC-ND license](#), and code is released under the [MIT license](#). If you find this content useful, please consider supporting the work by [buying the book](#)!

< Introducing Scikit-Learn | Contents | Feature Engineering >

Open in Colab

Hyperparameters and Model Validation

In the previous section, we saw the basic recipe for applying a supervised machine learning model:

1. Choose a class of model
2. Choose model hyperparameters
3. Fit the model to the training data
4. Use the model to predict labels for new data

jupyter notebook

File Edit View Insert Cell Kernel Help Trusted Python 3

Run Markdown

This notebook contains an excerpt from the [Python Data Science Handbook](#) by Jake VanderPlas; the content is available [on GitHub](#).

The text is released under the [CC-BY-NC-ND license](#), and code is released under the [MIT license](#). If you find this content useful, please consider supporting the work by [buying the book](#)!

< In-Depth: Manifold Learning | Contents | In Depth: Gaussian Mixture Models >

Open in Colab

In Depth: k-Means Clustering

In the previous few sections, we have explored one category of unsupervised machine learning models: dimensionality reduction. Here we will move on to another class of unsupervised machine learning models: clustering algorithms. Clustering algorithms seek to learn, from the properties of the data, an optimal division or discrete labeling of groups of points.

IOT DATA ANALYSIS

Reinforcement Learning

2102541 IoT Fundamentals (2/2018)
Assoc Prof Dr Chaodit Aswakul
chaodit.a@chula.ac.th

ML Categories

- Supervised learning
  - Regression models (linear regression, neural network)
  - Classification models (logistic regression, neural network)
- Unsupervised learning
  - K-means clustering
- Reinforcement learning
  - Q-learning

fitting model with past data to predict unseen cases

finding structural patterns in data

taking control of system

REINFORCEMENT LEARNING

[REF: ©MIT 6.S191
Introduction to Deep Learning
introtodeeplearning.com]

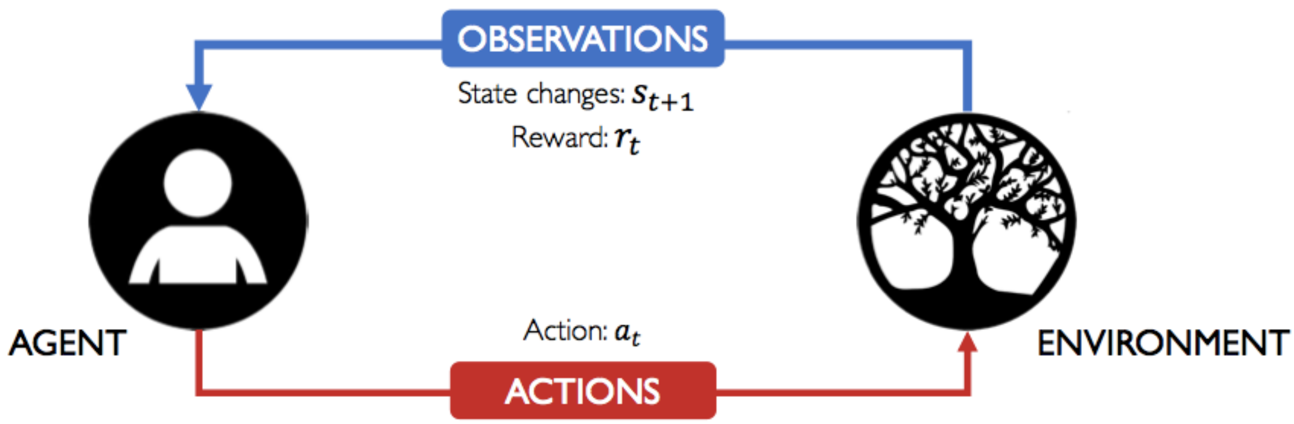
Data: state-action pairs

Goal: Maximize future rewards over many time steps

Apple example:

Eat this thing because it will keep you alive.

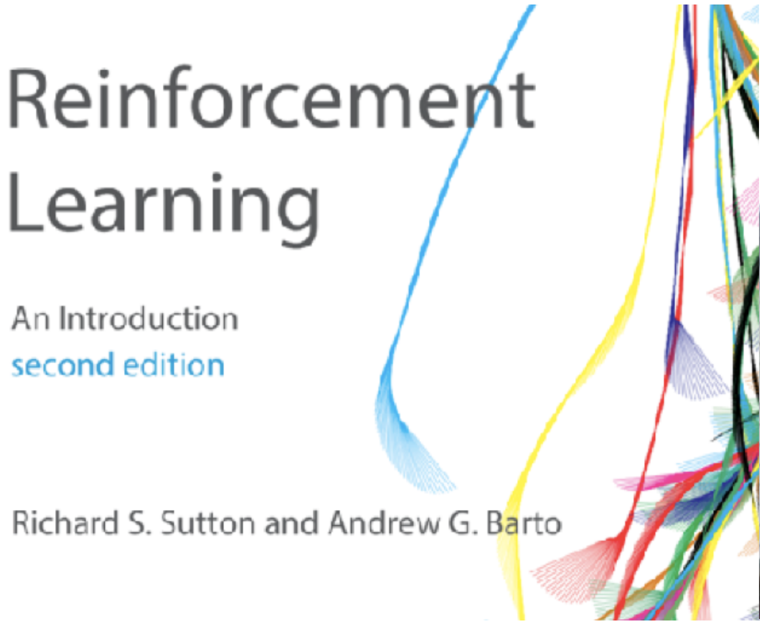
Reinforcement Learning

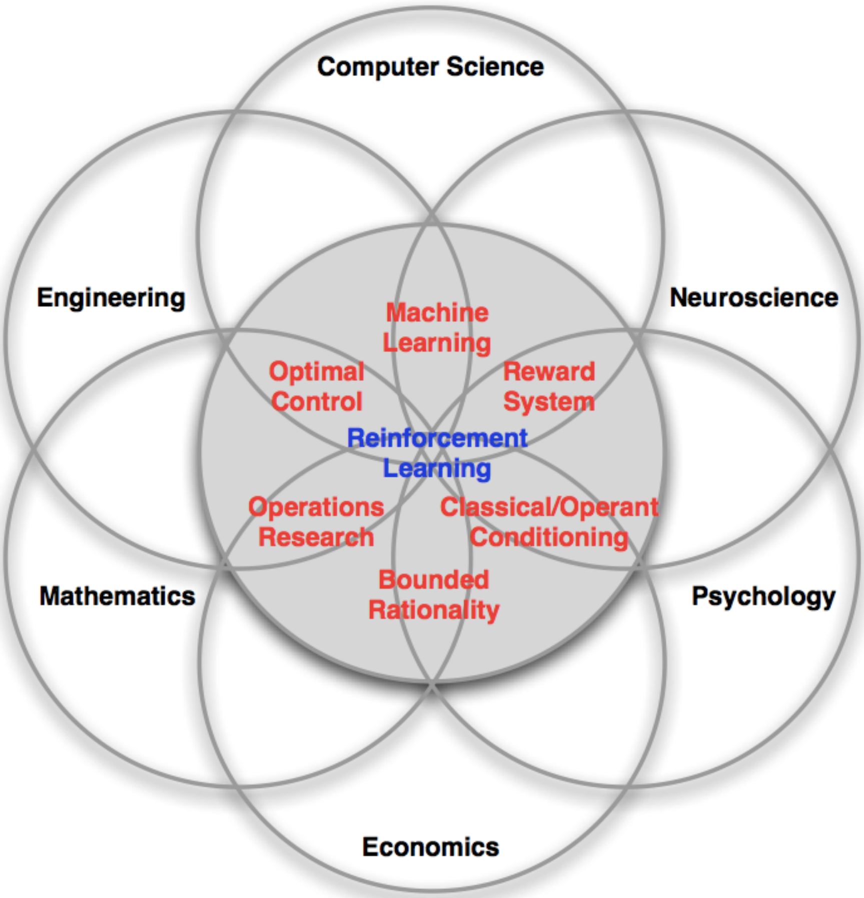


Reward: feedback that measures the success or failure of the agent's action.

[REF: ©MIT 6.S191
Introduction to Deep Learning
introtodeeplearning.com]

Classic Reference Textbook





REF: David Silver (2015)

Q-Learning

When system is in state s …

… agent must choose an action a

Define:

Q(s,a) =

'quality' or 'reward' given to agent after agent takes action a when system is in state s

EXAMPLE 1

system of road network

state : congestion level

action : traffic signal light

reward : travel time reduction

EXAMPLE 2

system of network firewall

state : info of captured packets

action : decision to let-go or drop

reward : malicious packet dropped

good packets let-go

EXAMPLE 3

system of Alpha GO

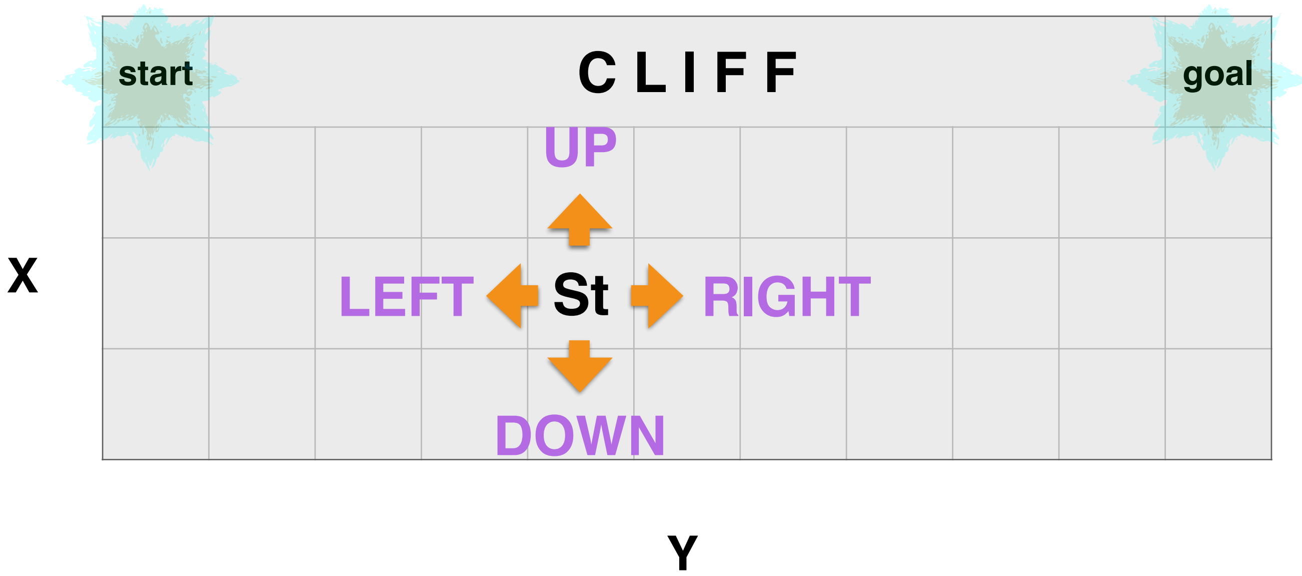
state : play positions on board

action : where to play next

reward : winning chance

TODAY PROBLEM

Cliff Walking Scenario



REF: Sutton & Barto, Reinforcement Learning Book

Today's Class : Cliff-Walking

Choose path from START to GOAL

state : current location

action : next movement directions

(LEFT, RIGHT, UP, DOWN)

reward : -1 (if still on the way but reaching goal)

-100 (if falling off cliff)

reward recorded after reaching goal
i.e. "completed one learning episode"

REF: Sutton & Barto, Reinforcement Learning Book

Q-Learning

- Initialise $Q(s, a)$ for all (s, a) arbitrarily
- Choose actions in any way (based on $Q(s, a)$) such that all actions are taken in all states (infinitely often in the limit)

$$\ldots S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, S_{t+2}, \ldots$$

REF: Sutton & Barto, Reinforcement Learning Book

Q-Learning

"epsilon-greedy algorithm"

- with probability = 1 - epsilon
  choose action that maximises $Q(s, a)$
  "exploitation strategy"

- with probability = epsilon
  choose any other actions randomly
  "exploration strategy"

REF: Sutton & Barto, Reinforcement Learning Book

Q-Learning

- On each time step t, update $Q(s, a)$ by

$$Q(S_t, A_t) \leftarrow (1 - \alpha)\underbrace{\boxed{Q(S_t, A_t)}}_{\text{old value}} + \alpha\,(\underbrace{\boxed{R_{t+1}}}_{\text{immediately gained reward}} + \gamma\,\underbrace{\boxed{\max_a Q(S_{t+1}, a)}}_{\text{future expected maximum reward}})$$

REF: Sutton & Barto, Reinforcement Learning Book

Hands-on Q-Learning Experiment

File Edit View Insert Cell Kernel Help Trusted Python 3

Reinforcement Learning (Q-Learning)

Remark: The original file has been simplified and reformatted to Jupyter Notebook for teaching in 2102541 IoT Fundamentals. (C. Aswakul 2 Apr 2019).

In [ ]:

```
# REF: github.com/ShangtongZhang/reinforcement-learning-an-introduction
#      /blob/master/chapter06/cliff_walking.py

#######################################################################
# Copyright (C)                                                       #
# 2016-2018 Shangtong Zhang(zhangshangtong.cpp@gmail.com)             #
# 2016 Kenta Shimada(hyperkentakun@gmail.com)                         #
# Permission given to modify the code as long as you keep this        #
# declaration at the top                                              #
#######################################################################
```

REF: Sutton & Barto, Reinforcement Learning Book

THANK YOU

Assoc Prof Dr Chaodit Aswakul
Wireless Network and Future Internet Research Unit
Chulalongkorn University, Thailand