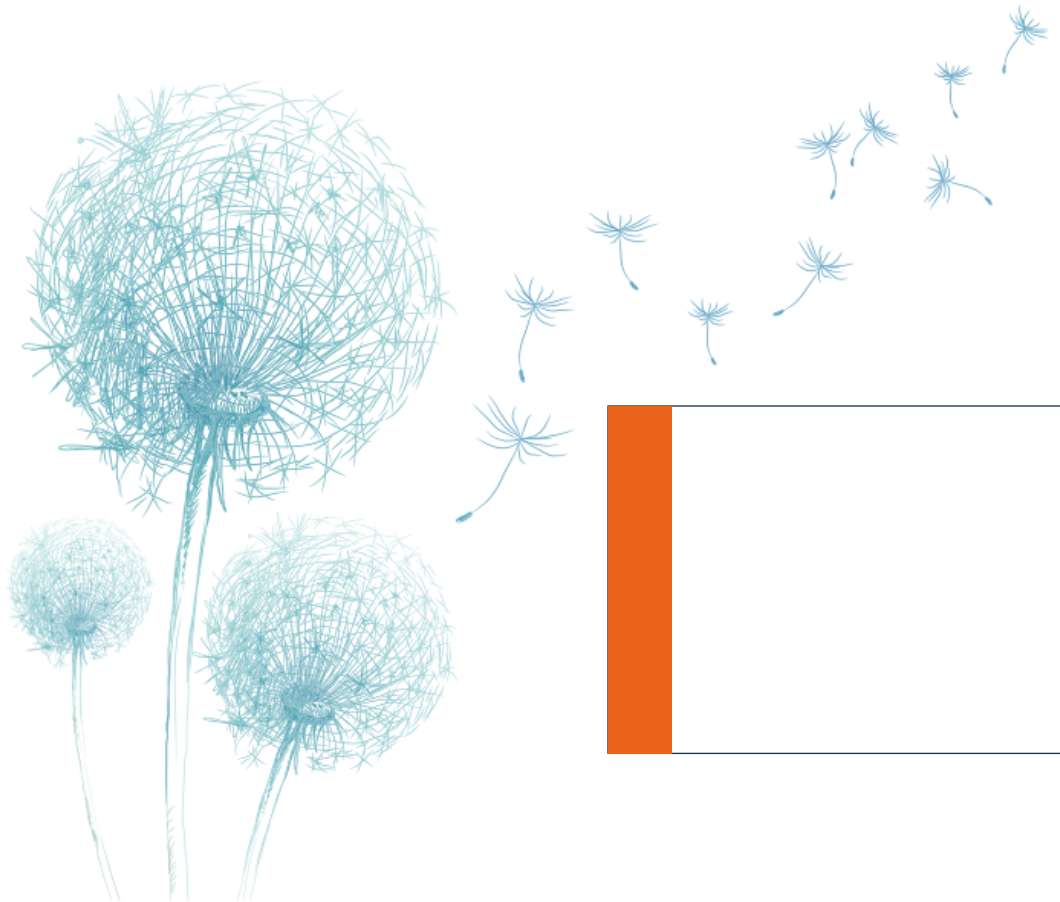
Neural networks and deep learning



Overview

Kun Suo
Computer Science, Kennesaw State University
https://kevinsuo.github.io/

Self Introduction

Kun Suo, Ph.D.

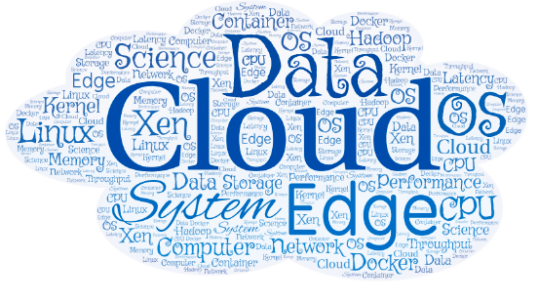
- Homepage, https://kevinsuo.github.io/

Research interests:

- Cloud computing and virtualization;
- Operating systems, containers and kubernetes;
- Software defined network (SDN) and network function virtualization (NFV)
- Big data systems and machine learning systems

Projects you may be interested in:

- Several projects in Cloud & Data & Edge
- https://kevinsuo.github.io/code-lab.html



Now it's your turn

- Name, program/year, where from
- Your interests in Computer Science
- What do you expect in the Neural networks and deep learning?

https://www2.eecs.berkeley.edu/Research/Areas/CS/

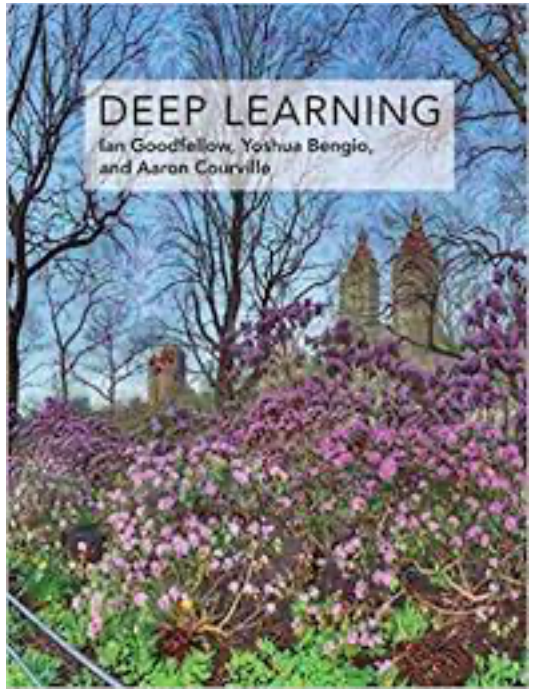
If you are in the online course, introduce yourself in D2L,
Discussions → Self-Introduction

Course Information

- Instructor: Dr. Kun Suo
- Office: J-318
- Email: ksuo@kennesaw.edu
  - Only reply to e-mails that are sent from KSU student email accounts and title the course number [CS7357]
- Office Hours:
  - T/Th, 2pm-3pm
  - By appointment
- Course Materials
  - Homework assignments, lecture slides, and other materials will be posted in the webpage (https://kevinsuo.github.io/teaching/2021Spring/7357/class.html) and D2L.

Reference Book

- Deep Learning (Adaptive Computation and Machine Learning series) :
  - Ian Goodfellow
  - The MIT Press, 2016
  - ISBN-13: 978-0262035613



Prerequisites

- Computer basics that are supposed to covered in *CS 5040 - Data Structures & Algorithms or equivalent*

- Python programming (code reading, development and debugging). (Famous projects in python: https://hackernoon.com/50-popular-python-open-source-projects-on-github-in-2018-c750f9bf56a0)

- Linux command line environment (debugging, simple shell programming).

For Python and Linux beginners

- Python tutorial
  - https://docs.python.org/3/tutorial/
  - https://www.w3schools.com/python/
  - https://www.tutorialspoint.com/python/index.htm
  - https://www.learnpython.org/

- Linux tutorial
  - https://ryanstutorials.net/linuxtutorial/
  - http://www.ee.surrey.ac.uk/Teaching/Unix/
  - https://www.tutorialspoint.com/unix/

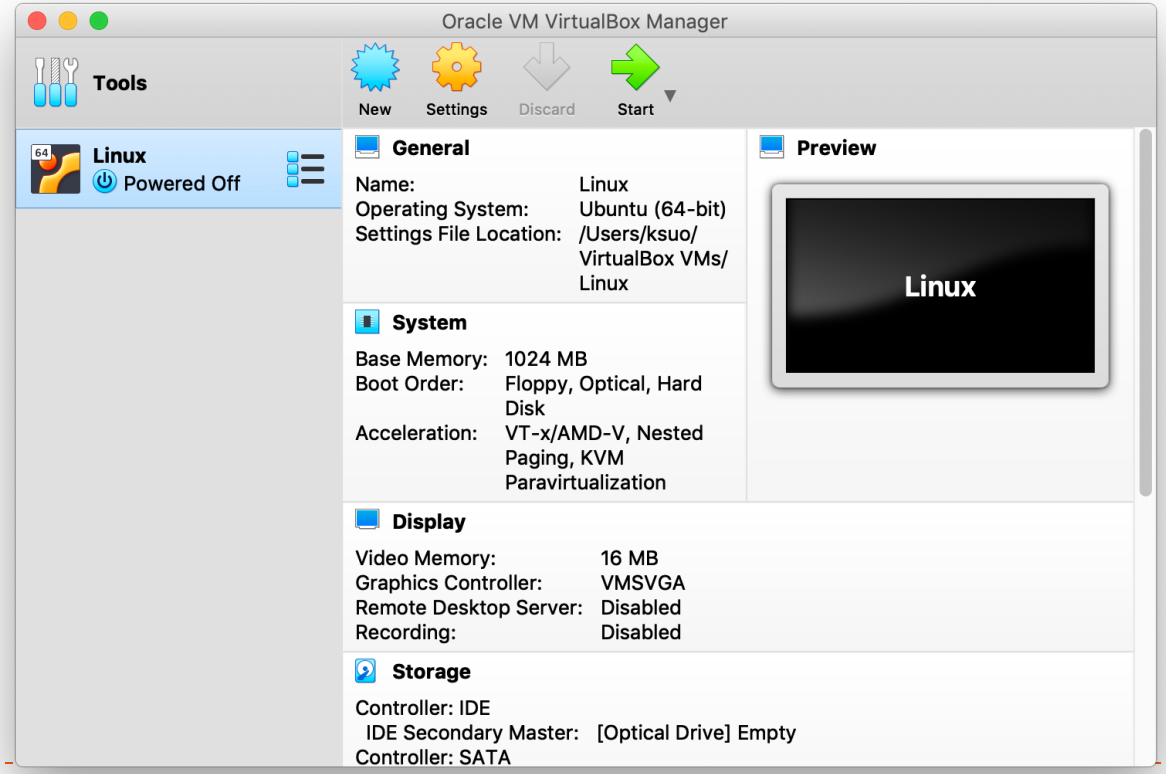
Project Environment

- Recommend project environment
  - VirtualBox + Ubuntu + Linux + Python

Virtual machine VM OS VM OS Kernel

https://www.virtualbox.org/

https://ubuntu.com/download/desktop



Project Environment

- Recommend project environment
  - VirtualBox + Ubuntu + Linux + Python

https://www.virtualbox.org/

https://ubuntu.com/download/desktop

- New to VirtualBox?
  - https://oracle-base.com/articles/vm/virtualbox-creating-a-new-vm
  - https://www.youtube.com/watch?v=sB_5fqiysi4

Project Environment

- If you like
  - Windows / Mac + Python

Course Structure

- Lectures
  - M/W 8:00 PM – 9:15 PM
  - Online
- Paper presentation
  - Second half term
- Projects
  - 3 programming assignments
- Exams (open books)
  - 3X : online, TBA.

Course Policy

- Grading scale

Percentage	Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	F

Grading Policy (cont.)

- Grading percentage
  - In-class discussion, attendance & paper presentation: 10%
  - Projects (x3): 30%
  - Exam 1: 20%
  - Exam 2: 20%
  - Final exam: 20%

Late submission policy: late submission will **not be accepted** and **no credits**

Academic Integrity

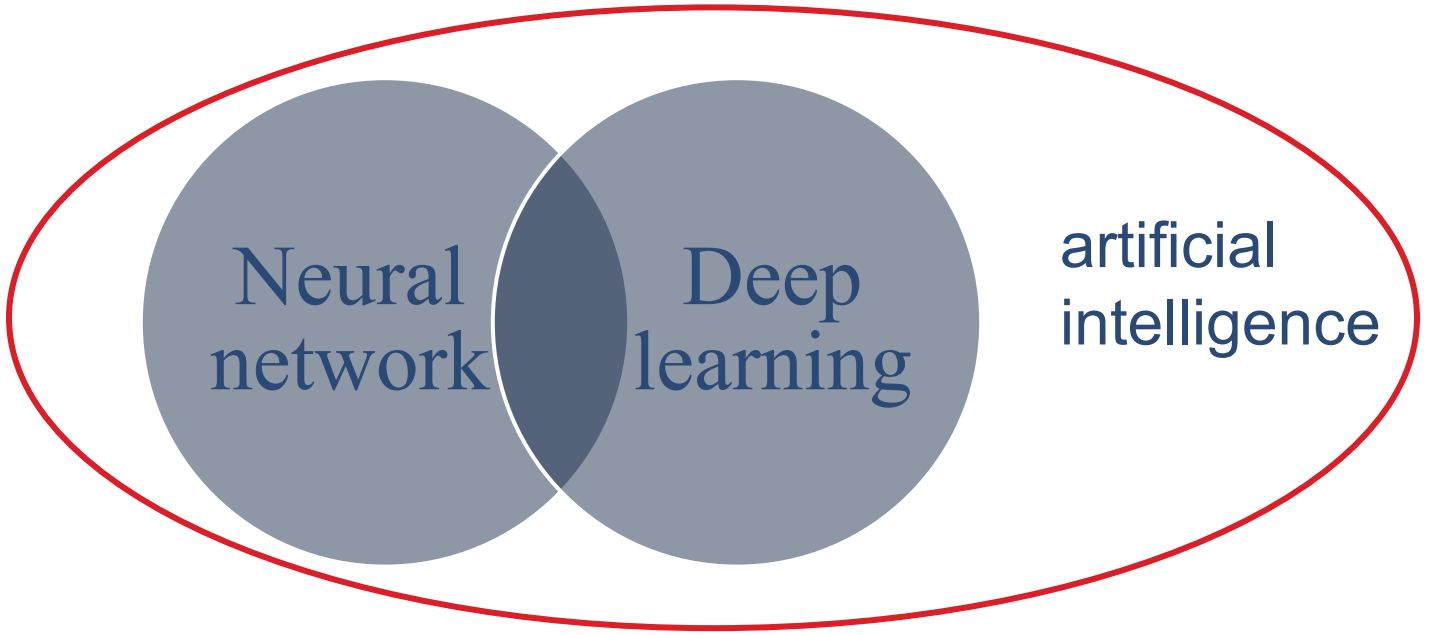
- Academic dishonesty
  - Cheating
  - Plagiarism
  - Collusion
  - The submission for credit of any work or materials that are attributable in whole or in part to another person
  - Taking an examination for another person
  - Any act designed to give unfair advantage to a student or the attempt to commit

Where to go for help ?

- Ask questions in class
- Ask questions outside class
  - Classmates and friends
- Attend office hours
  - Dr. Kun Suo: Tuesday/Thursday 2:00PM – 3:00PM, J-318
- Search on the web
  - Stand on the shoulder of giants

About this lecture

- A subfield of artificial intelligence
  - Neural network: a model with (artificial) neurons as the basic unit
  - Deep learning: a type of machine learning problem that mainly solves the problem of contribution allocation.



Prerequisite knowledge

- Linear algebra
- Calculus
- Mathematical optimization
- Probability theory
- Information theory

Recommended online courses

- CS224n: Deep Learning for Natural Language Processing
  - https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/
  - Chris Manning, mainly explain various deep learning models in the field of natural language processing
- CS231n: Convolutional Neural Networks for Visual Recognition
  - http://cs231n.stanford.edu/
  - Fei-Fei Li and Rej Karpathy, mainly explain the application of CNN and RNN in the image field
- CS 294: Deep Reinforcement Learning
  - http://rail.eecs.berkeley.edu/deeprlcourse/

Recommended top conferences

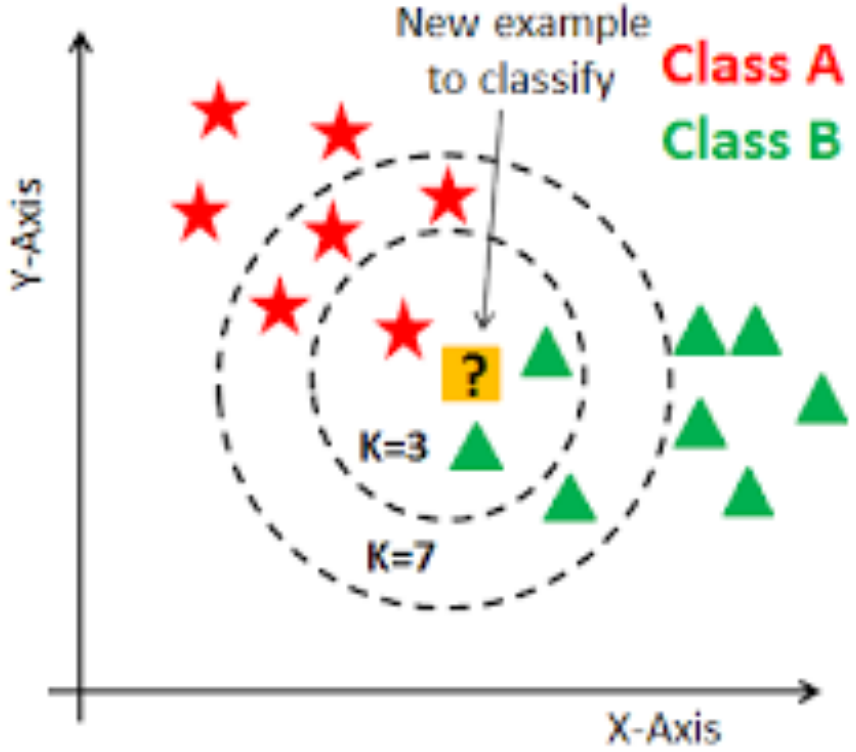
- NeurIPS、ICLR、ICML、AAAI、IJCAI
- ACL、EMNLP
- CVPR、ICCV
- …

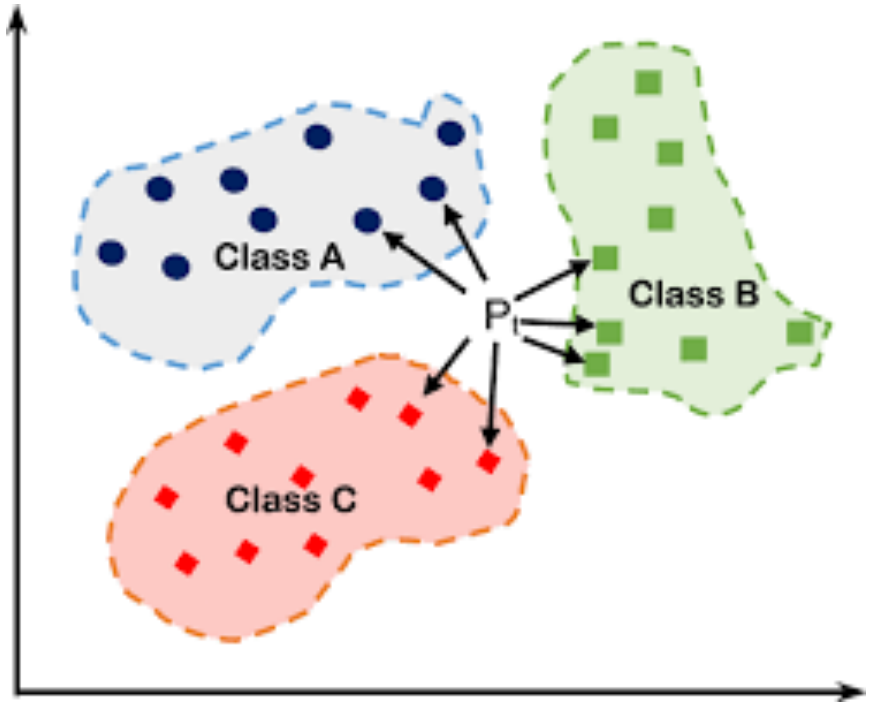
Course outline

Week	Topics	Homework
Week 1	Overview	
Week 2	No class, Machine learning	
Week 3	KNN	HW1
Week 4	Linear Regression	
Week 5	Exam 1	
Week 6	Logistic Regression	HW2
Week 7	Feedforward neural network	
Week 8	CNN	
Week 9	No class, Spring break	
Week 10	Exam 2	
Week 11	RNN	HW3
Week 12	Unsupervised learning	
Week 13	GAN	
Week 14	Resarch paper presentation	
Week 15	Resarch paper presentation	
Week 16	Resarch paper presentation	
Week 17	Final exam	

Project 1

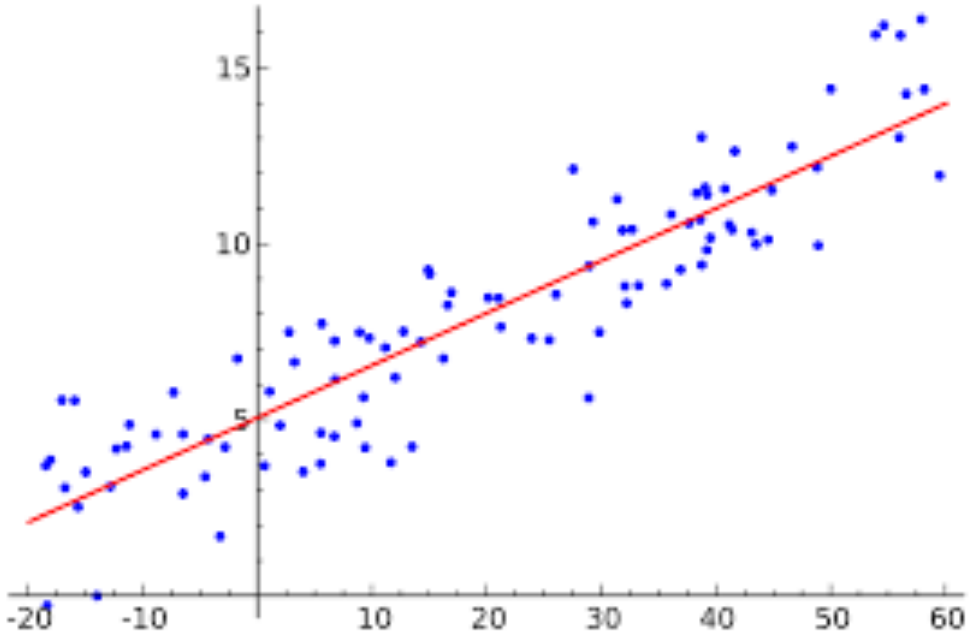
- KNN

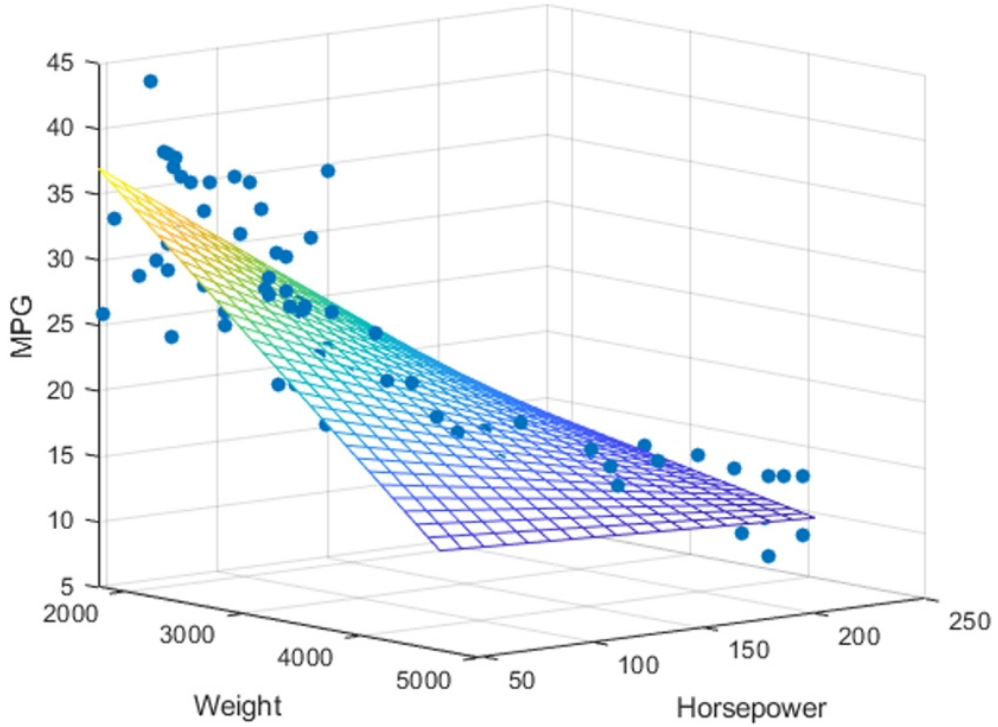




Project 2

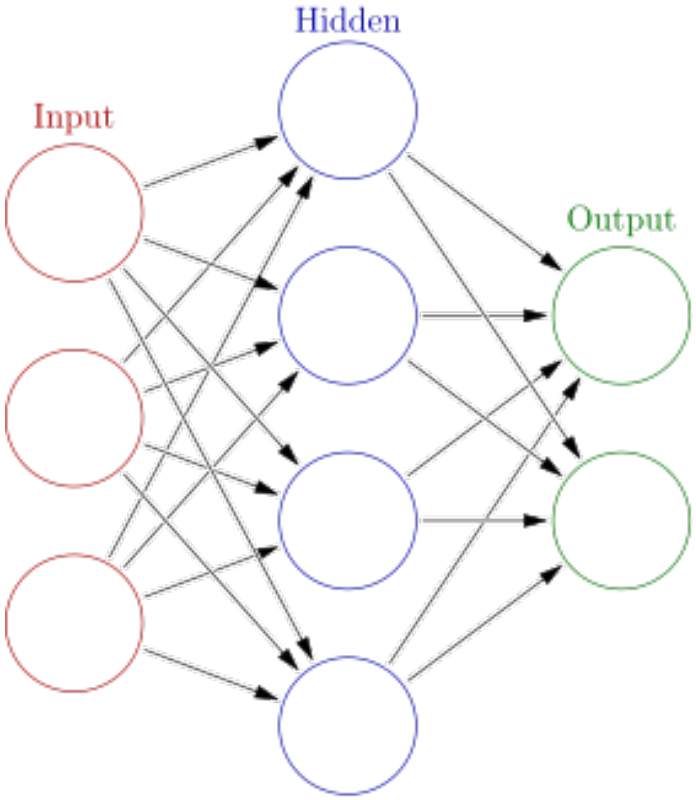
- Linear regression

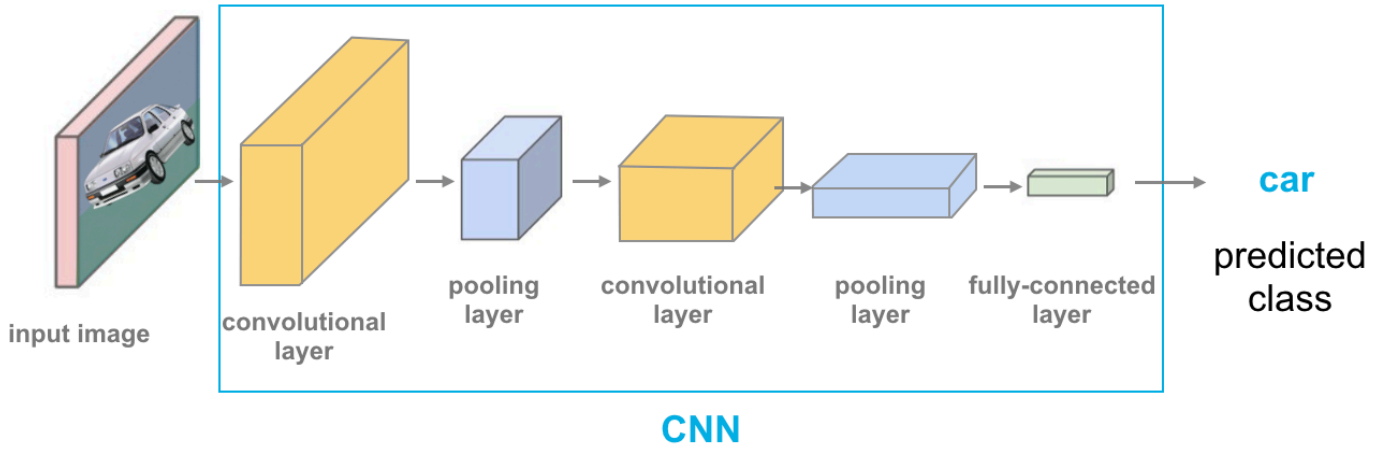




Project 3

- neural networks and convolutional neural network

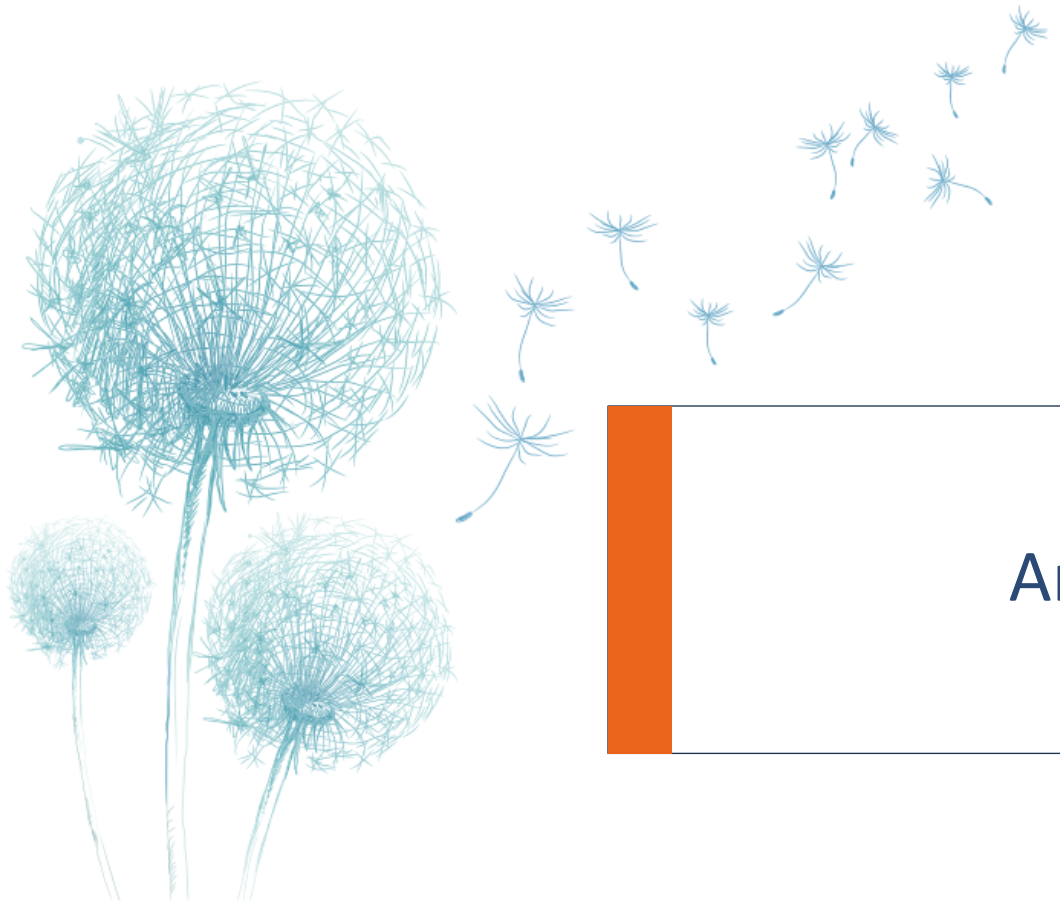




Paper presentation

- https://docs.google.com/spreadsheets/d/1kGszd_RWYSKGaXkznMFz3mBWH17bPJvnzU8haclL_Co/edit?usp=sharing

Name	Neural Network	Paper
	MobileNet v1	https://arxiv.org/pdf/1704.04861.pdf
	Inception v1	https://arxiv.org/pdf/1409.4842.pdf
	ResNet-50 v2	https://arxiv.org/pdf/1603.05027.pdf).
	VGG-16	https://arxiv.org/pdf/1409.1556.pdf%20http://arxiv.org/abs/1409.1556.pdf
	ResNet-SRGAN	https://openaccess.thecvf.com/content_cvpr_2017/papers/Ledig_Photo-Realistic_Single_
	SRCNN 9-5-5	https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7115171&casa_token=tysl9rSqf3
	ResNet-DPED	https://openaccess.thecvf.com/content_ICCV_2017/papers/Ignatov_DSLR-Quality_Photo
	U-Net	https://arxiv.org/pdf/1505.04597.pdf)%e5%92%8c%5bTiramisu%5d(https://arxiv.org/abs/
	Nvidia-SPADE	https://openaccess.thecvf.com/content_CVPR_2019/papers/Park_Semantic_Image_Synt
	ICNet	https://openaccess.thecvf.com/content_ECCV_2018/papers/Hengshuang_Zhao_ICNet_f
	PSPNet	https://openaccess.thecvf.com/content_cvpr_2017/papers/Zhao_Pyramid_Scene_Parsin
	DeepLab v1	https://arxiv.org/pdf/1412.7062.pdf
	Project 1	https://github.com/kevinsuo/CS7357/tree/master/project/1
	Project 2	https://github.com/kevinsuo/CS7357/tree/master/project/2
	Project 3	https://github.com/kevinsuo/CS7357/tree/master/project/3



Artificial intelligence

Artificial intelligence

- Artificial intelligence (AI) is to make machines have human intelligence.
  - "Computer Control (input/output)" + "Intelligent Behavior"

- The birth of the subject of artificial intelligence has a definite landmark event: the Dartmouth Conference in 1956.
  - At this conference, "artificial intelligence" was proposed and used as the name of this research field.

Artificial intelligence is to make the behavior of the machine look like the intelligent behavior shown by humans.

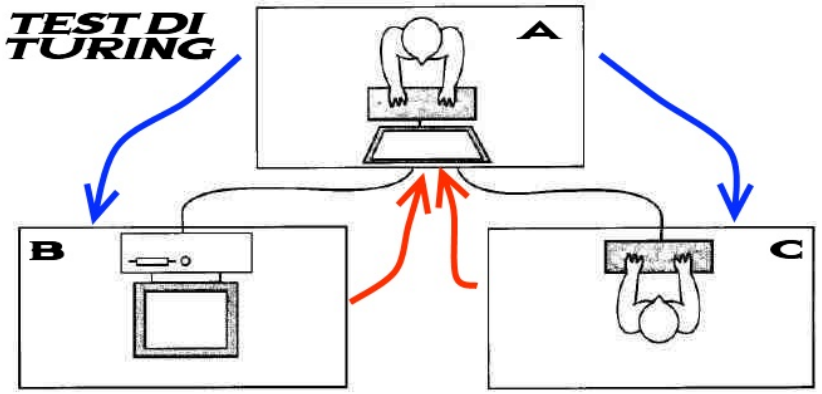
John McCarthy（1927-2011）

Turing test

"A person conducts a series of questions and answers with the other party in a special way without contacting the other party. If he cannot determine whether the other party is a human or a computer based on these questions for a long time, then he can think of this Computers are smart"

Alan Turing

---Alan Turing [1950]
《Computing Machinery and Intelligence》

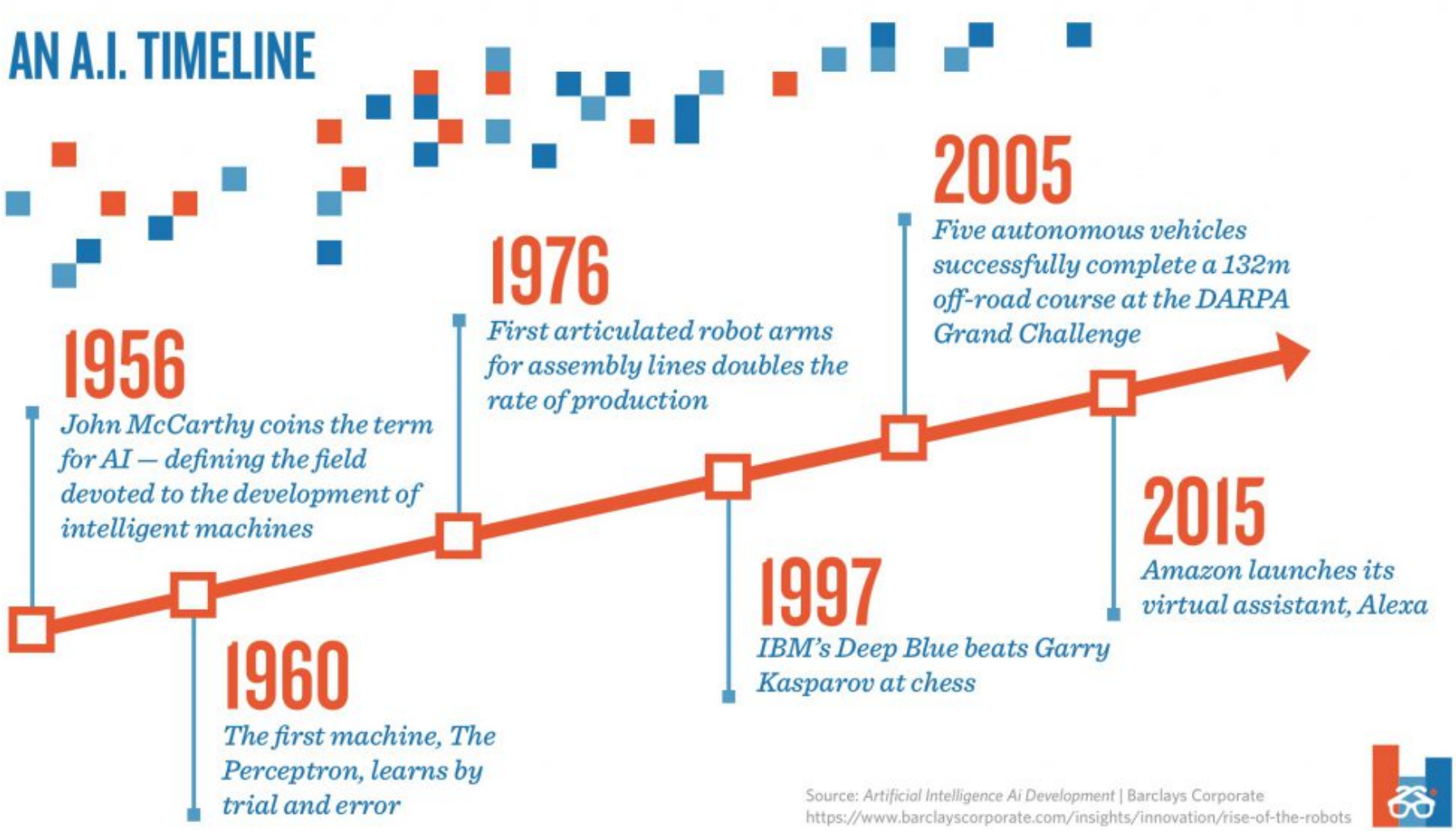


The computer must have the ability to ***understand language, learning, memory, reasoning, decision making, etc.***

Artificial Intelligence Research Field

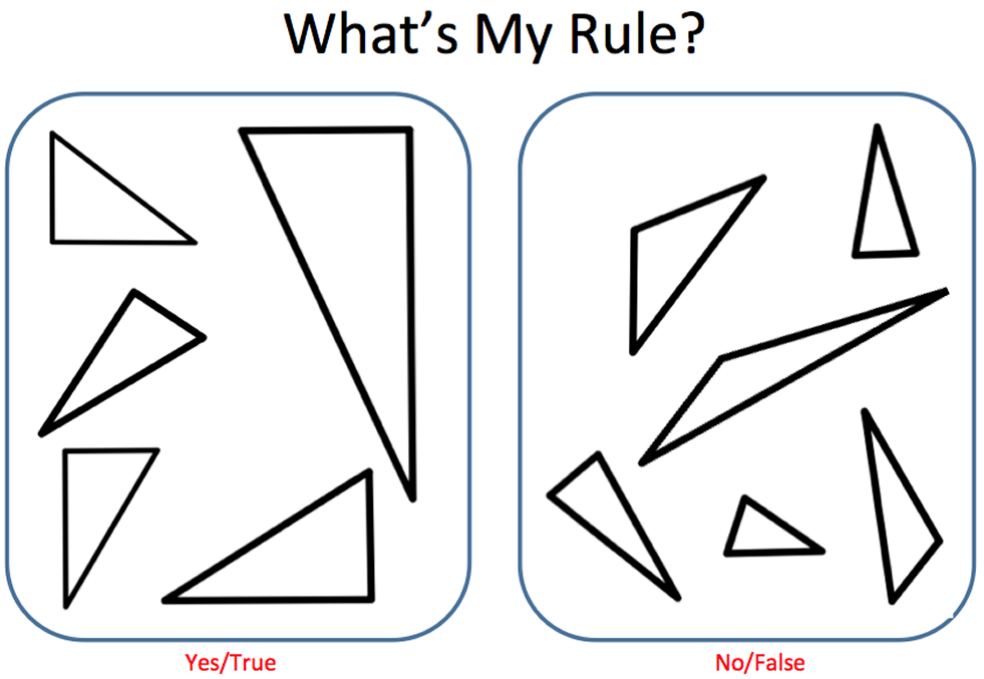
- Let machines have human intelligence
  - Machine perception (computer vision, voice information processing)
  - Learning (pattern recognition, machine learning, reinforcement learning)
  - Language (Natural Language Processing)
  - Memory (knowledge representation)
  - Decision-making (planning, data mining)

Artificial Intelligence history



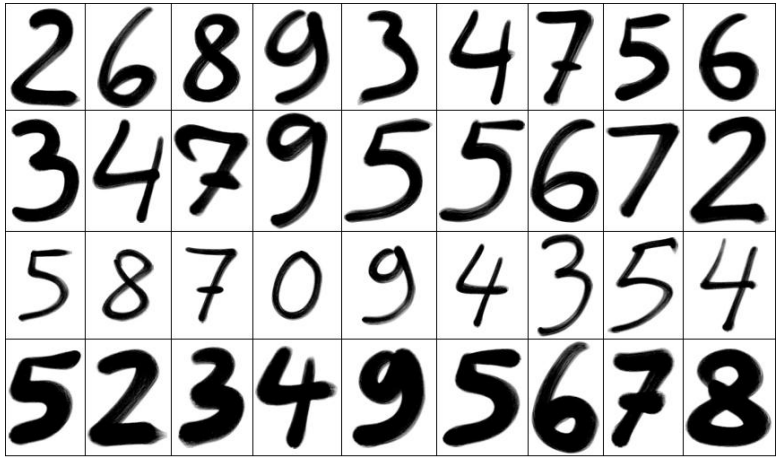
How to develop an artificial intelligence system?

- Expert knowledge (manual rules)



Right angle

What is the Rule?



Machine learning

Machine learning ≈ build a mapping function

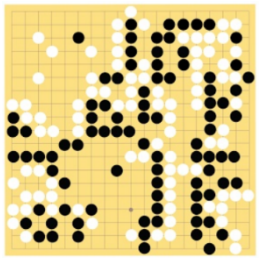
- Speech Recognition

$f($  $) =$ “Hello”

- Image Identification

$f($  $) =$ “9”

- Go

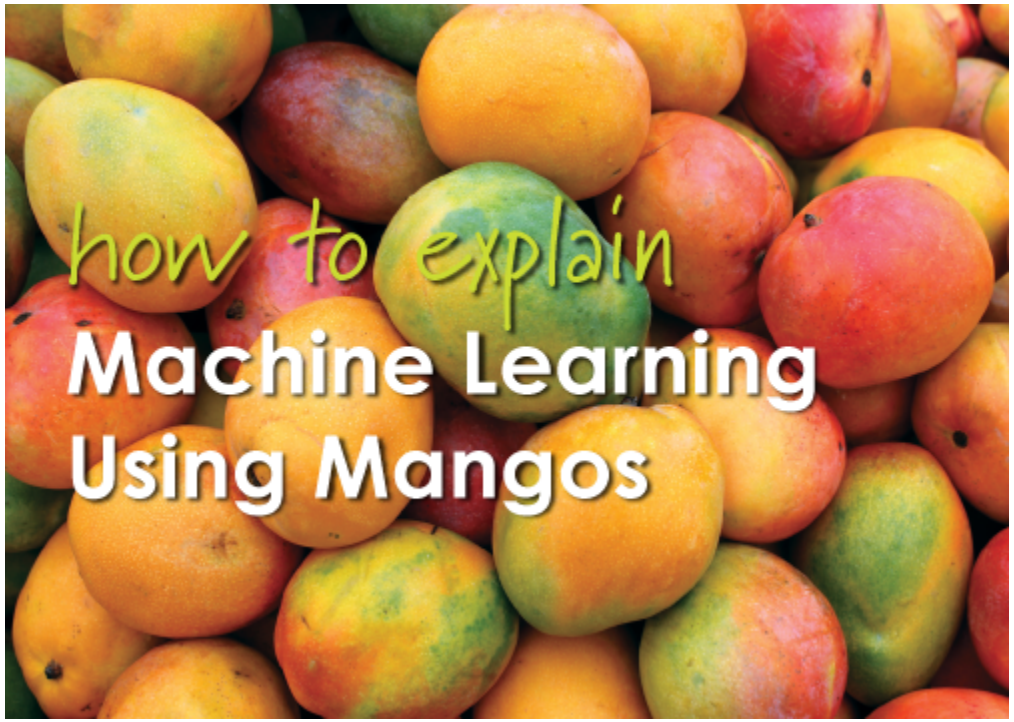
$f($  $) =$ “6-5” (position)

- Machine translation

$f($ “你好！” $) =$ “Hello!”

Mango machine learning

Q: How to determine whether mangoes are sweet?



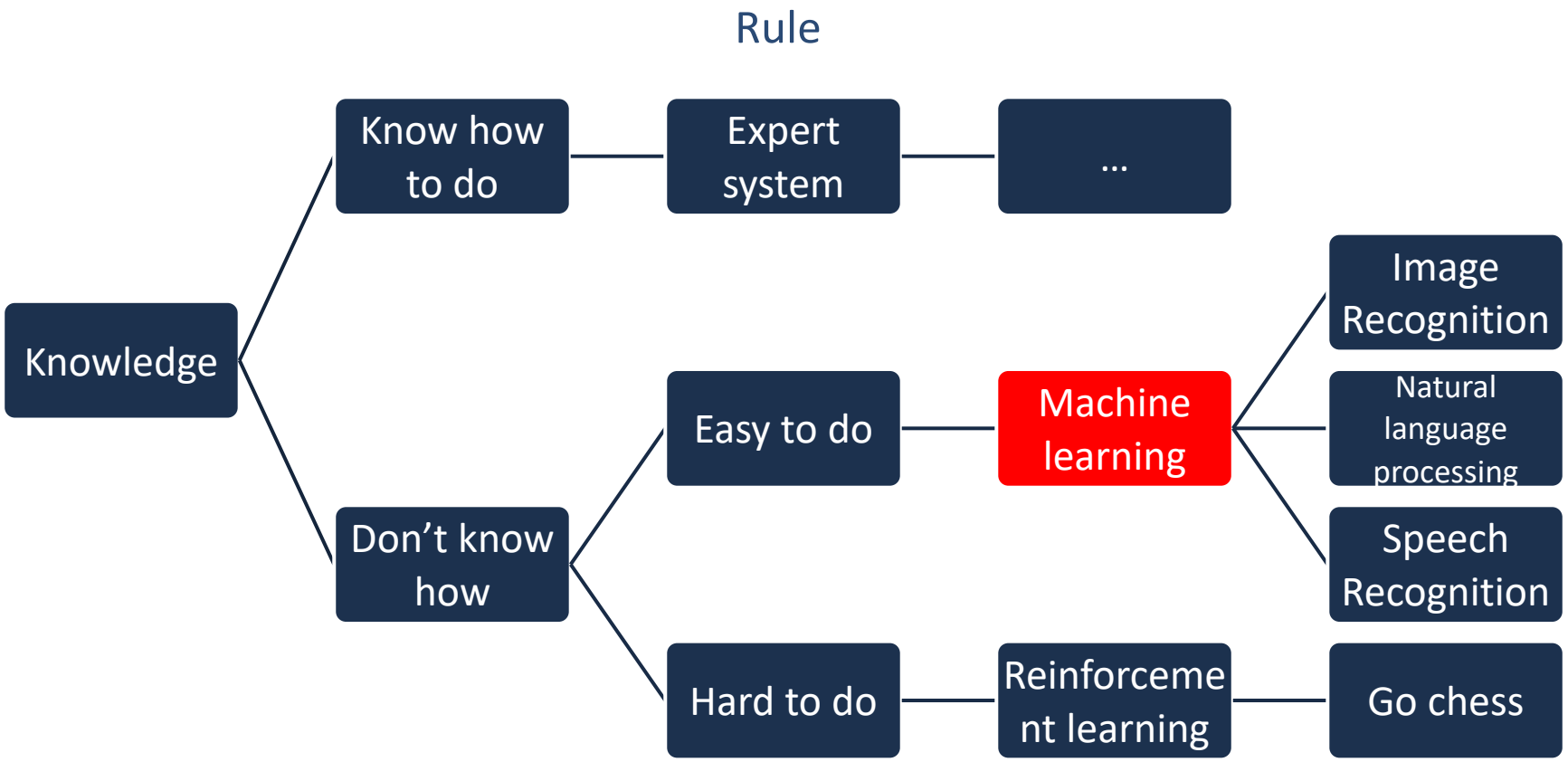
https://www.quora.com/How-do-you-explain-Machine-Learning-and-Data-Mining-to-non-Computer-Science-people

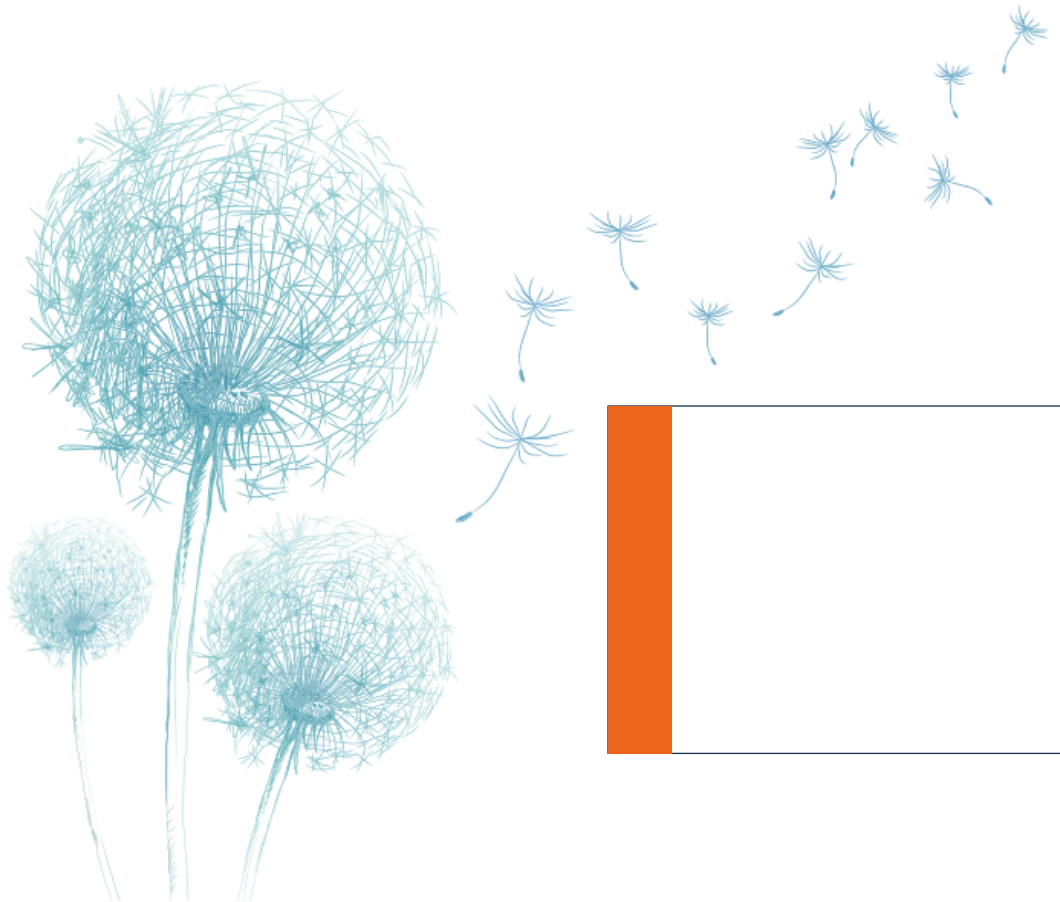
Mango machine learning

- From a randomly selected mango sample (training data) from the market, list all the characteristics of each mango:
  - Such as color, size, shape, origin, brand
- And mango quality (output variable):
  - Sweet, juicy, ripeness.

- Design a learning algorithm to learn the correlation model between mango characteristics and output variables.
- Next time you buy mangoes from the market, you can use the previously calculated model to predict the quality of mangoes based on the characteristics of the mangoes (test data).

How to develop an artificial intelligence system?

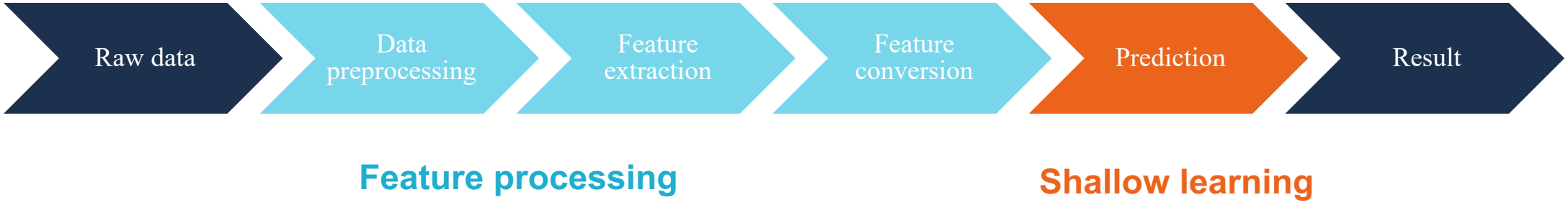




Deep learning

Machine learning

- When we use machine learning to solve some pattern recognition tasks, the general process includes the following steps:

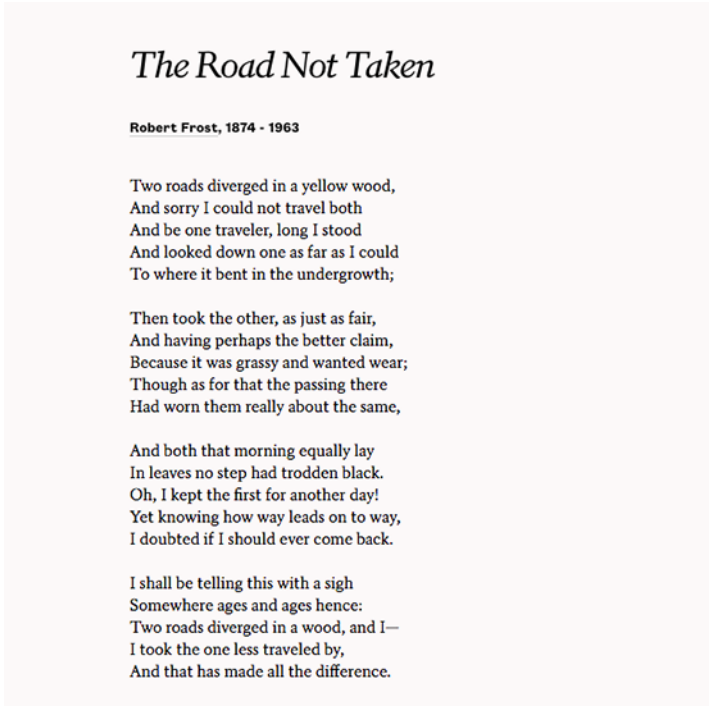


- Shallow Learning: Does not involve feature learning, and its features are mainly extracted by manual experience or feature conversion methods.

Feature processing: One of the Challenges of Artificial Intelligence

- Low-level features VS high-level semantics
  - People's understanding of text and images cannot be directly obtained from the underlying features of strings or images

VS



The Road Not Taken

Robert Frost, 1874 - 1963

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I—
I took the one less traveled by,
And that has made all the difference.

Representation Learning

- Data representation is the core issue of machine learning.
  - Feature engineering: requires human intelligence

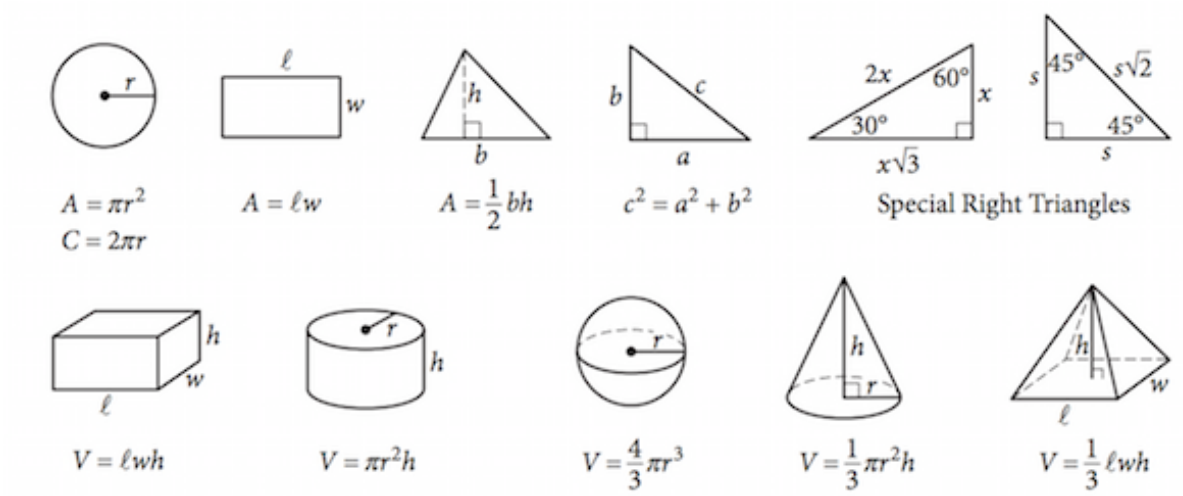
- Representation learning
  - How to automatically learn good representations from data

- Difficulty
  - No clear goal

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.

Data representation

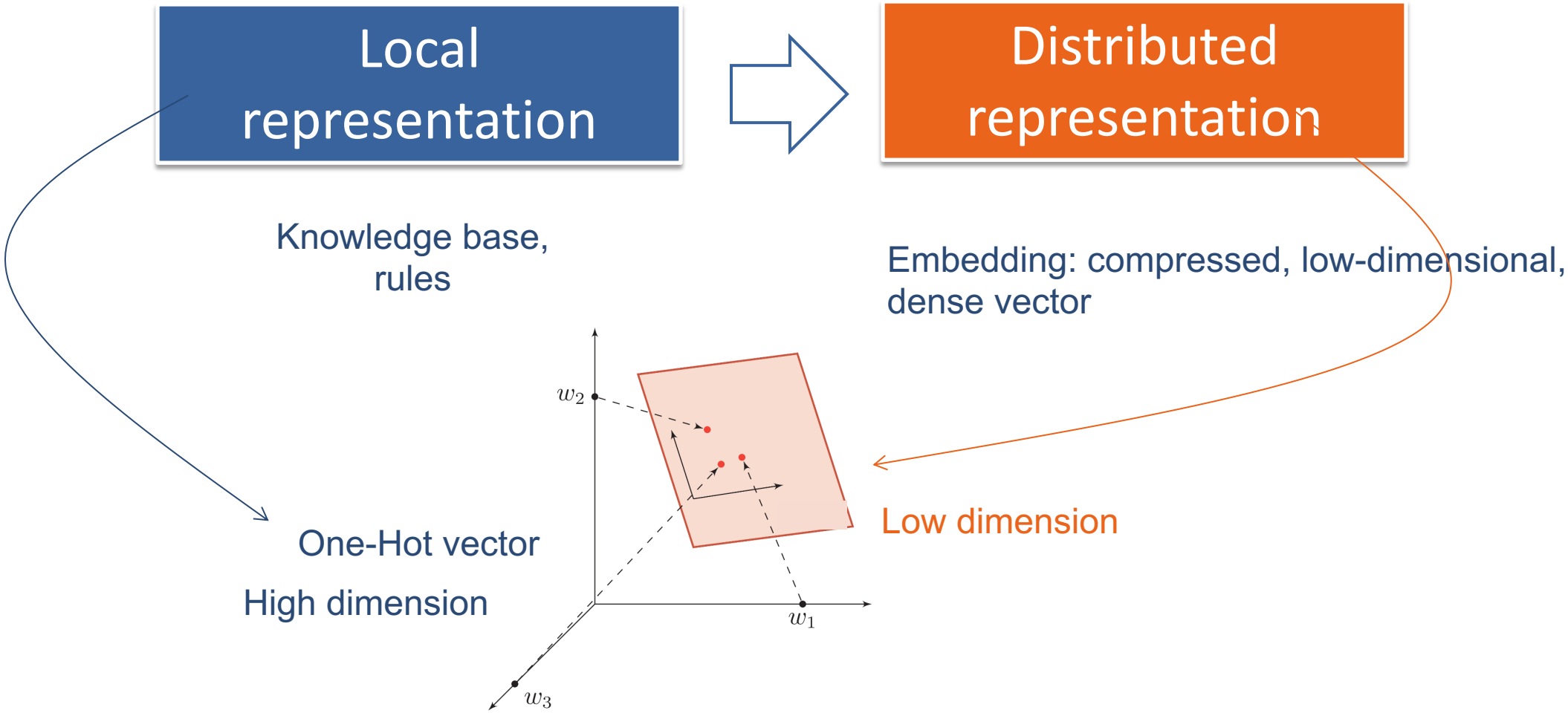
- "Good representation" is a very subjective concept and there is no clear standard.



- But generally speaking, a good representation has the following advantages:
  - Should have a strong presentation ability.
  - Should make subsequent learning tasks simple.
  - Should be general, task or field independent.

Semantic representation

- How to represent semantics in a computer?

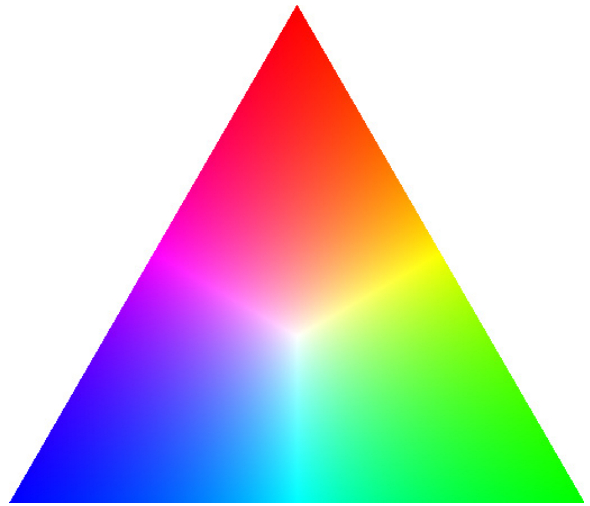


Local representation

- Each element represents an entity

- Example:
  - Take color as an example, name different colors with different names

Color	Local representation	Distributed representation
Amber	$[1,\ 0,\ 0,\ 0]^{\mathrm{T}}$	$[1.00,\ 0.75,\ 0.00]^{\mathrm{T}}$
Sky blue	$[0,\ 1,\ 0,\ 0]^{\mathrm{T}}$	$[0.00,\ 0.5,\ 1.00]^{\mathrm{T}}$
Red	$[0,\ 0,\ 1,\ 0]^{\mathrm{T}}$	$[0.67,\ 0.22,\ 0.12]^{\mathrm{T}}$
Brown	$[0,\ 0,\ 0,\ 1]^{\mathrm{T}}$	$[0.44,\ 0.31\ 0.22]^{\mathrm{T}}$



Local representation

- Advantage

  1) Easy to model

  2) High calculation efficiency for linear model

- Disadvantage

  1) The dimension is high and cannot be expanded;

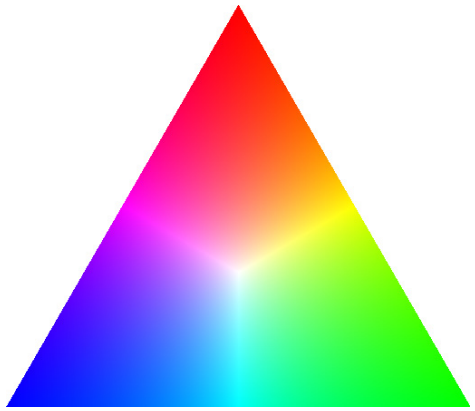
  2) Unable to calculate the similarity between

Distributed representation

- An entity is represented by multiple elements

- Example:
  - Take color as an example, use RGB values to represent colors, and different colors correspond to a point in the R, G, and B three-dimensional space

Color	Local representation	Distributed representation
Amber	$[1,\ 0,\ 0,\ 0]^{\mathrm{T}}$	$[1.00,\ 0.75,\ 0.00]^{\mathrm{T}}$
Sky blue	$[0,\ 1,\ 0,\ 0]^{\mathrm{T}}$	$[0.00,\ 0.5,\ 1.00]^{\mathrm{T}}$
Red	$[0,\ 0,\ 1,\ 0]^{\mathrm{T}}$	$[0.67,\ 0.22,\ 0.12]^{\mathrm{T}}$
Brown	$[0,\ 0,\ 0,\ 1]^{\mathrm{T}}$	$[0.44,\ 0.31\ 0.22]^{\mathrm{T}}$



Distributed representation

- Advantage

  1) Low vector dimensionality of distributed representation

  2) Use low-latitude vectors to represent all information (Three-dimensional vectors can represent all colors, and it is also easy to represent new colors)

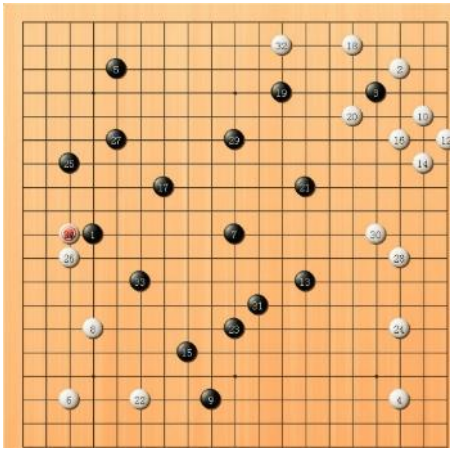
  3) Able to calculate the similarity between

Representation

- Local (symbol) representation
  - Discrete representation
  - Symbolic representation
  - One-Hot vector

	Local representation	Distributed representation
A	[1 0 0 0]	[0.25 0.5]
B	[0 1 0 0]	[0.2 0.9]
C	[0 0 1 0]	[0.8 0.2]
D	[0 0 0 1]	[0.9 0.1]

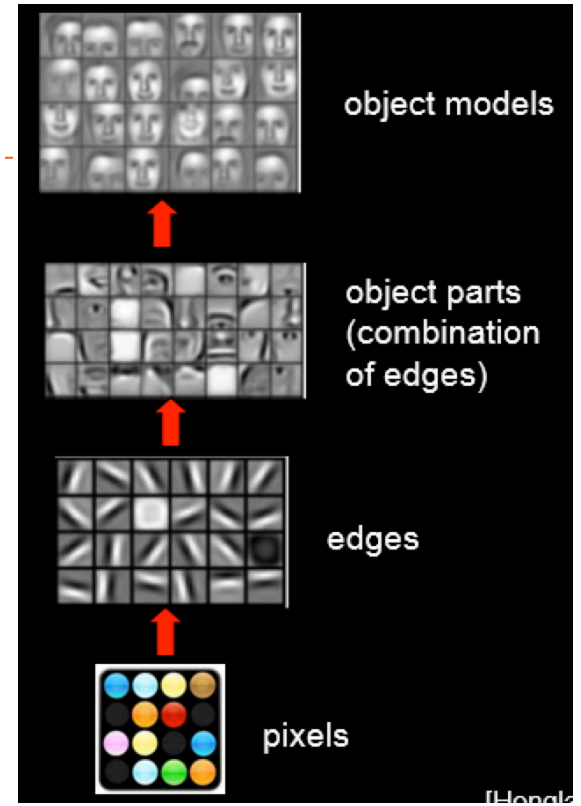
- Distributed representation
  - Compressed, low-dimensional, dense vector
  - Use O(N) parameters to represent O(2k) interval
    - k is a non-zero parameter, k<N



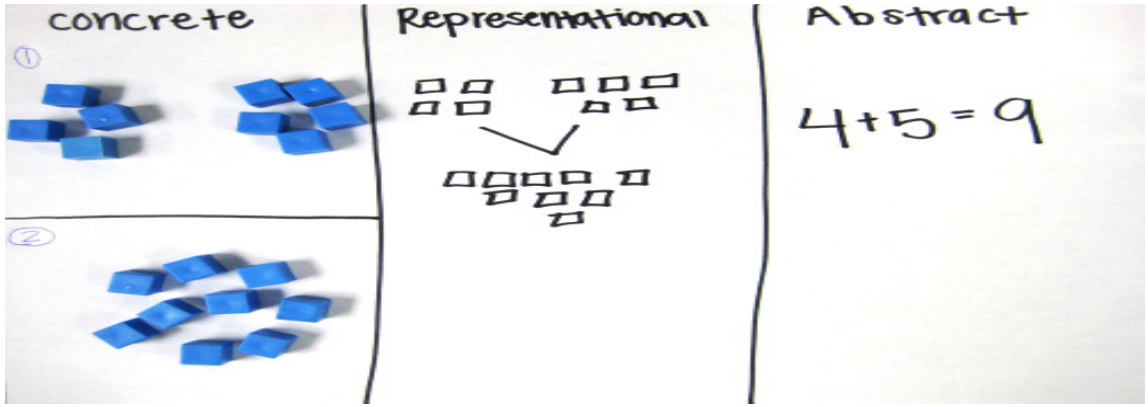
Distributed representation

Representation learning and deep learning

- A good representation learning strategy must have a certain depth
  - Feature reuse
    - Exponential representation



  - Abstract representation and immutability
    - Abstract representation requires multi-step construction



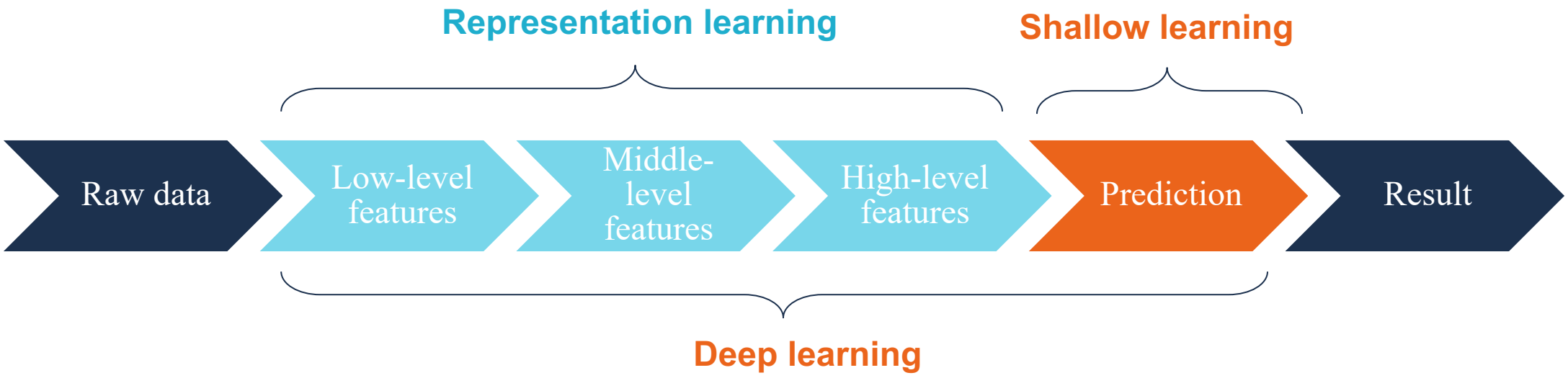
Traditional feature extraction

- Feature extraction
  - Linear projection (subspace)
  - Nonlinear embedding
  - Autoencoder

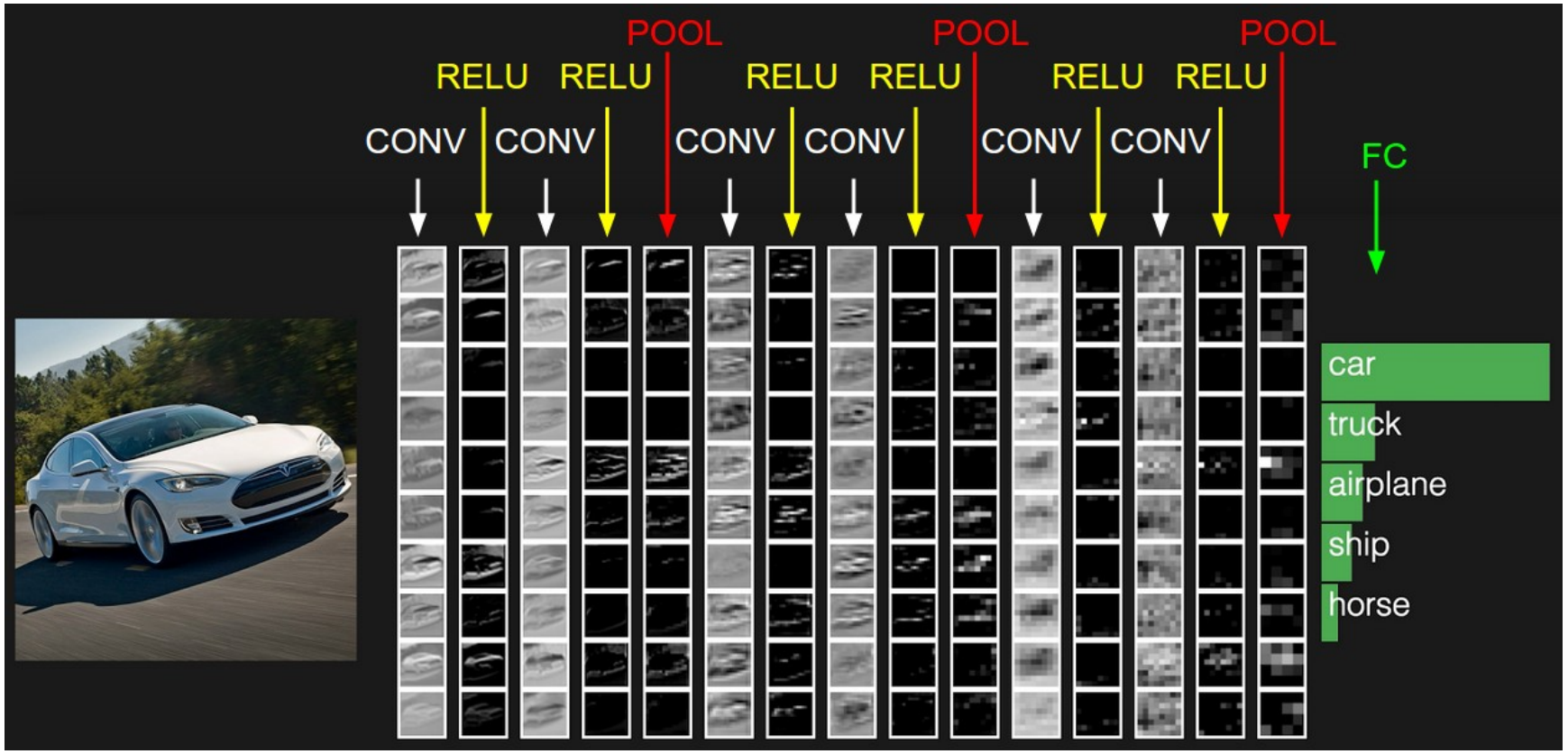
- Feature extraction VS representation learning
  - Feature extraction: remove useless features based on task or prior pair
  - Representation learning: learning high-level semantic features through deep models

Deep learning

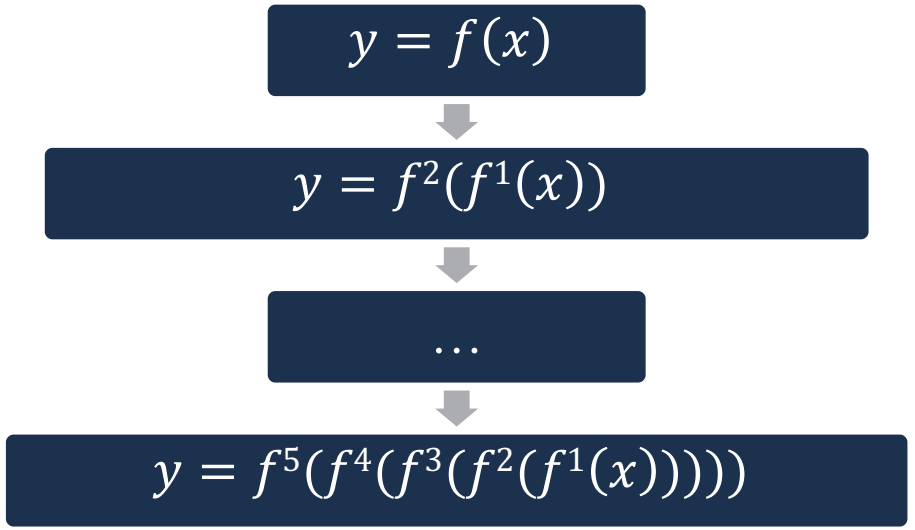
- By building a model with a certain "depth", the model can automatically learn a good feature representation (from low-level features, to middle-level features, and then to high-level features), thereby ultimately improving the accuracy of prediction or recognition



Representation learning and deep learning



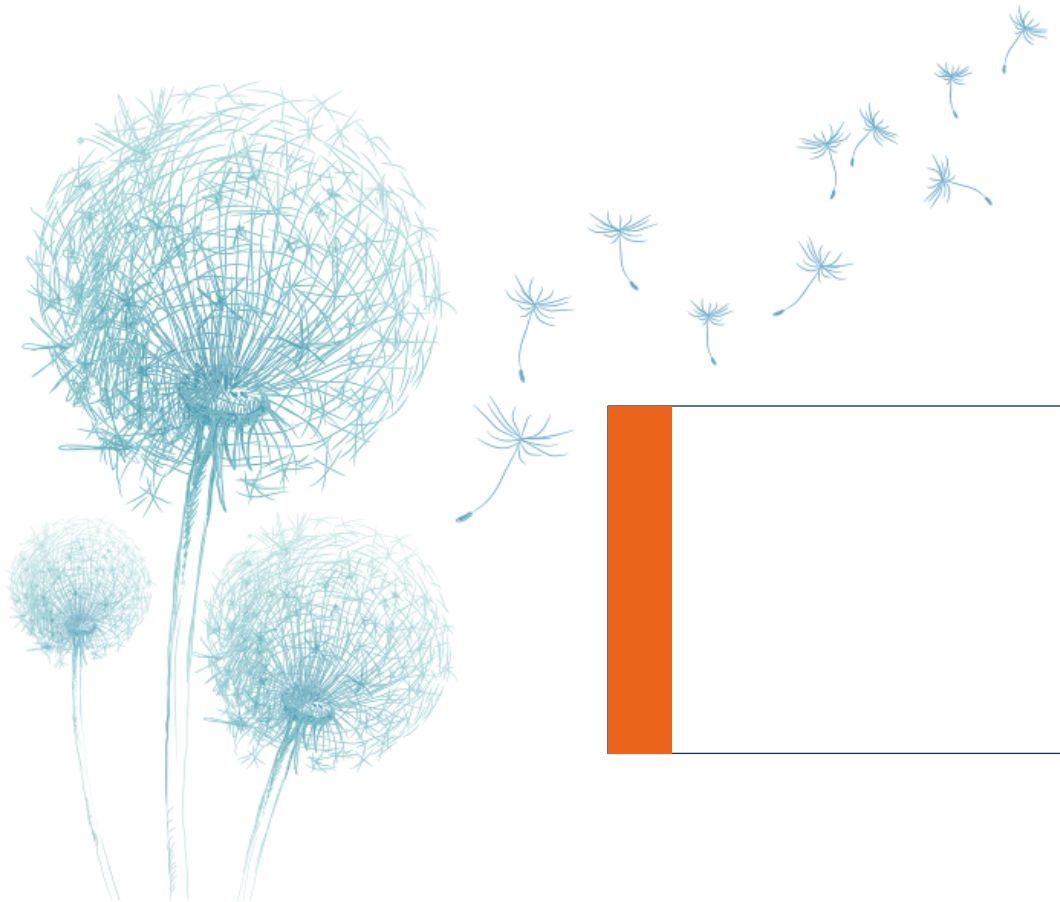
Mathematical description of deep learning



Low-level learning

Deep-level learning

$f^1(x)$ *is a nonlinear function, not necessarily continuous.*

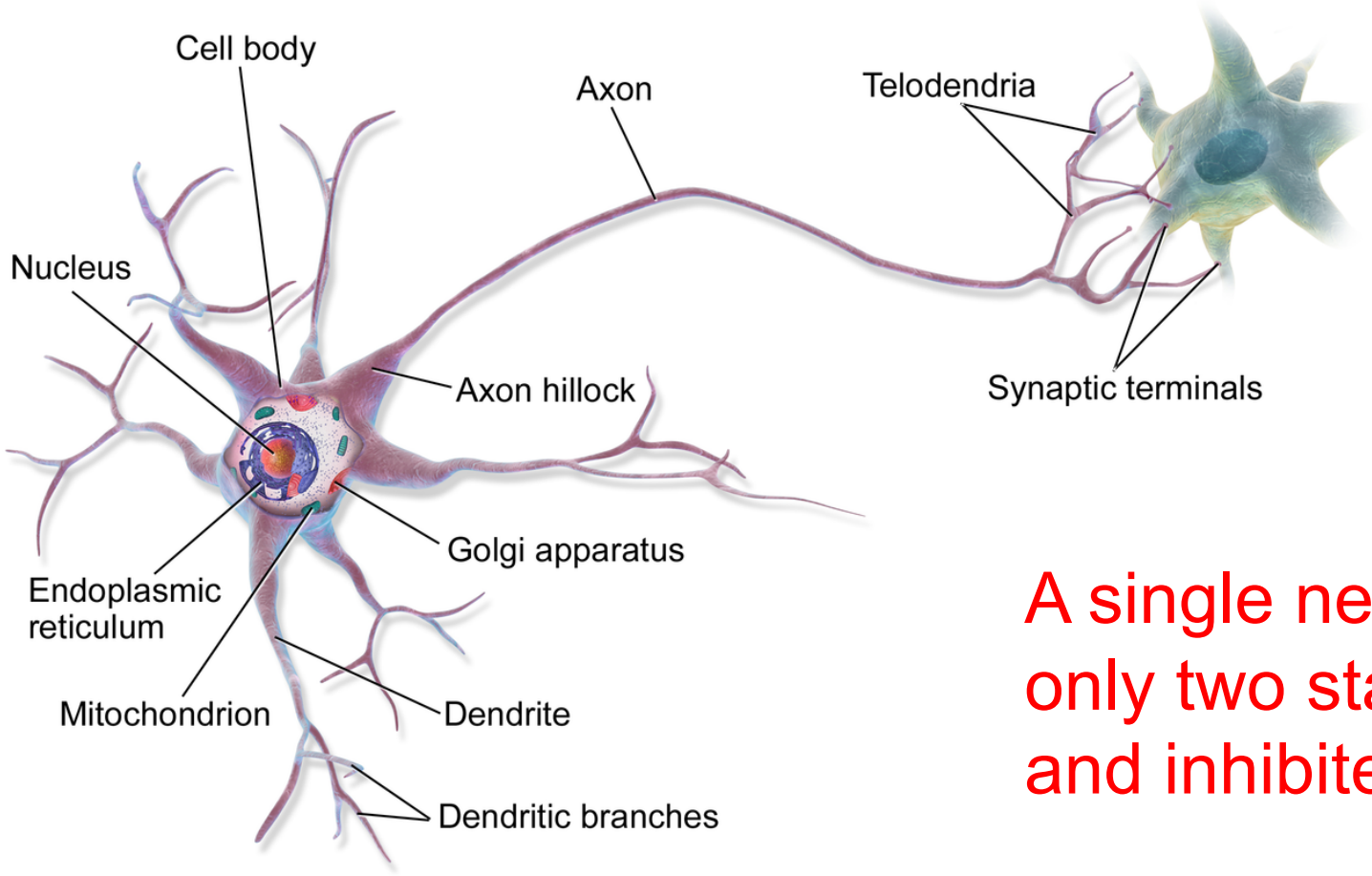


Neural Networks

Biological neuron

video: structure of brain

The human brain has 86 billion neurons



A single nerve cell has only two states: excited and inhibited

How do neural networks learn?

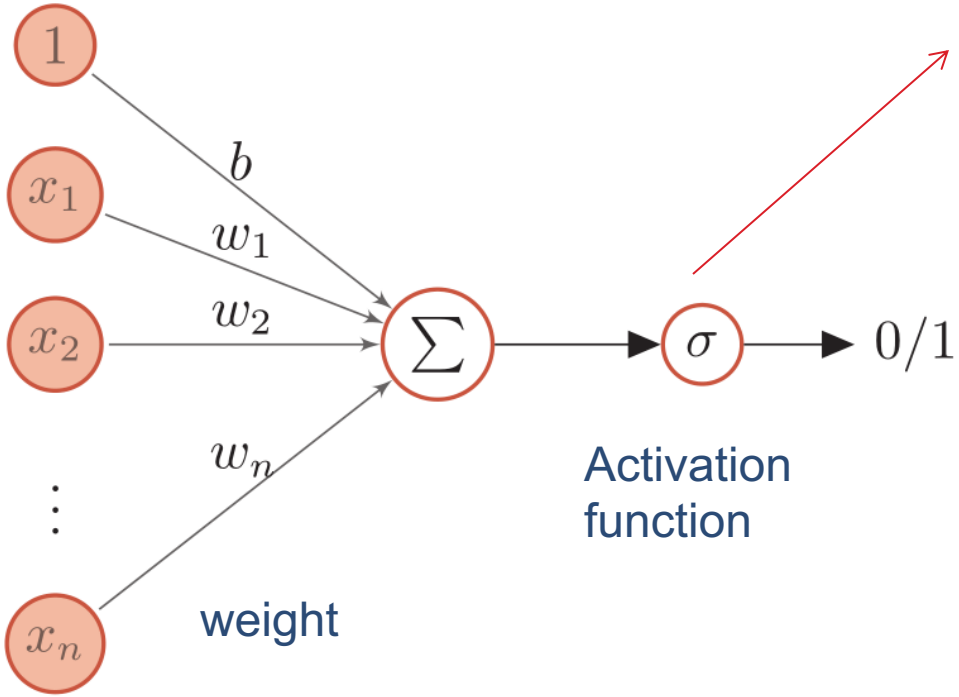
Hebb's Rule

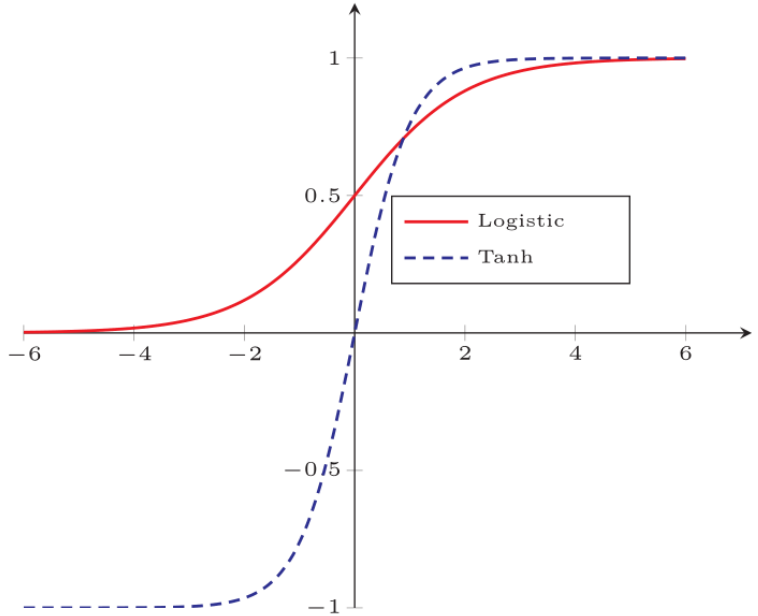
- "When an axon of neuron A and neuron B are close enough to affect it, and it continuously and repeatedly participates in the excitement of neuron B, then it will happen in these two neurons or one of them. Some kind of growth process or metabolic changes, so that neuron A as one of the cells that can excite neuron B, its effectiveness is enhanced."

---- Canadian psychologist Donald Hebb,
Behavioral organization, 1949

- The human brain has two types of memory: long-term memory and short-term memory. The short-term memory lasts no more than one minute. If an experience is repeated enough times, the experience can be stored in long-term memory.
- The process of transforming short-term memory into long-term memory is called coagulation.
- The hippocampus in the human brain is the core area for the coagulation of the brain structure.

Artificial neuron





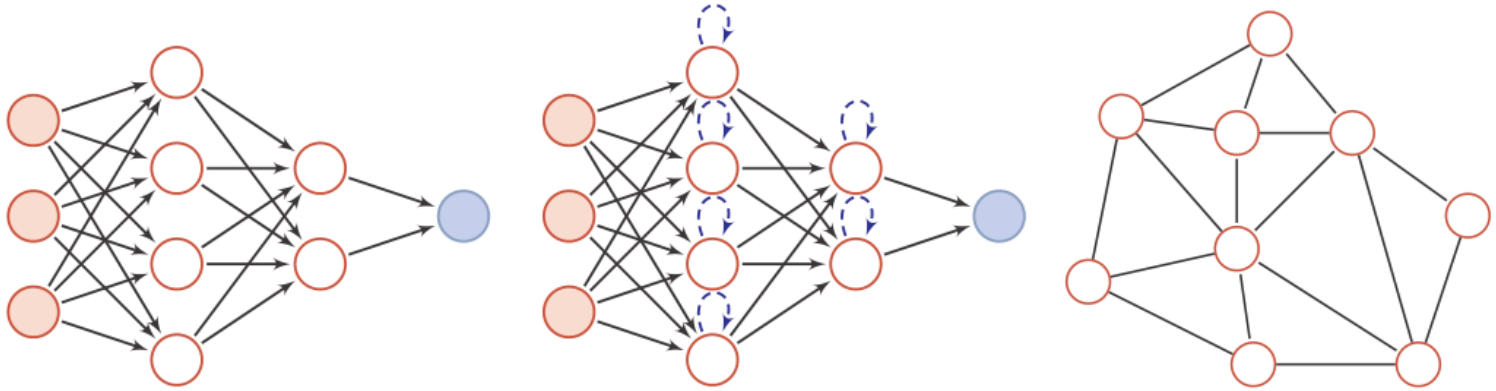
Artificial neural networks

- Artificial neural network is mainly composed of a large number of neurons and the directional connections between them. So consider three aspects:

  - Neuron activation rules
    - Mainly refers to the mapping relationship between neuron input and output, which is generally a non-linear function.
  - Network topology
    - The connection between different neurons.
  - Learning algorithm
    - Learn the parameters of the neural network through training data.

Artificial neural networks

- Artificial neural network is composed of neuron models, and this information processing network composed of many neurons has a parallel distributed structure.

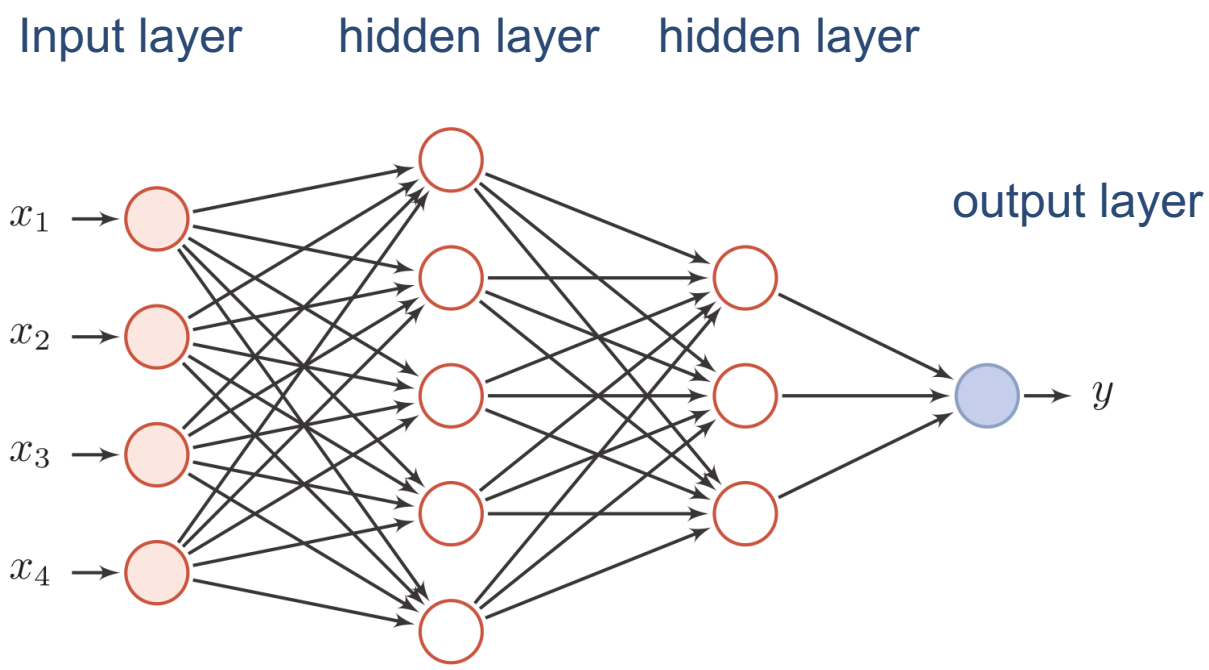


Feedforward network Memory network Graph network

  - Although neural network structures are roughly divided into three types here, most networks are composite structures, that is, a neural network includes multiple network structures.

Neural Networks

$$y = (f^3(f^2(f^1(x))))$$



The history of neural networks

- The development of neural network roughly goes through five stages.
- The first stage: the model is proposed
  - In 1943, psychologist Warren McCulloch and mathematician Walter Pitts first described an idealized artificial neural network and constructed a computer system based on simple logical operations. The neural network model they proposed is called the MP model.
  - Alan Turing described a "Type B Turing machine" in a 1948 paper. (Hebb-type learning)
  - In 1951, McCulloch and Pitts student Marvin Minsky built the first neural network machine called SNARC.
  - Rosenblatt [1958] first proposed a neural network model that can simulate human perception, and called it Perceptron, and proposed a learning algorithm that is close to the human learning process (iteration, trial and error).

The history of neural networks

- The second stage: ice age
  - In 1969, Marvin Minsky published the book "Perceptron", in which he concluded that the neural network was directly put into the cold palace, leading to the "ice age" of the neural network for more than ten years. They discovered two key problems with neural networks:
    - 1) The basic perceptron cannot handle XOR circuits.
    - 2) The computer does not have enough power to handle the long calculation time required by a large neural network.
  - In 1974, Paul Webos of Harvard University invented the backpropagation algorithm, but it did not receive the attention it deserves.
  - In 1980, Kunihiko Fukushima (Fukushima Kunihiko) proposed a multi-layer neural network with convolution and sub-sampling operations: Neocognitron (Neocognitron)

The history of neural networks

- The third stage: the revival caused by the backpropagation algorithm
  - In 1983, physicist John Hopfield introduced the concept of energy function to neural networks, and proposed a network for associative memory and optimized calculations (called Hopfield network), which achieved the best results at the time on the traveling salesman problem, causing a sensation.
  - In 1984, Geoffrey Hinton proposed a randomized version of Hopfield network, namely Boltzmann machine.
  - In 1986, David Rumelhart and James McClelland provided a comprehensive discussion on the application of connectionism in computer simulation of neural activities and reinvented the back propagation algorithm.
  - In 1986, Geoffrey Hinton et al. introduced the back propagation algorithm to the multilayer perceptron
  - In 1989, LeCun et al. introduced the backpropagation algorithm into the convolutional neural network and achieved great success in the recognition of handwritten digits.

The history of neural networks

- Stage 4: Decrease in popularity
  - In the mid-1990s, statistical learning theory and machine learning models represented by support vector machines began to emerge.
  - In contrast, the shortcomings of neural networks such as unclear theoretical basis, difficulty in optimization, and poor interpretability are more prominent, and the research of neural networks has once again fallen into a low ebb.

The history of neural networks

- The fifth stage: the rise of deep learning
  - In 2006, Hinton et al. found that a multilayer feedforward neural network can be effectively learned by pre-training layer by layer, and then fine-tuned with a backpropagation algorithm.
    - Deep neural networks have great success in tasks such as speech recognition and image classification.
  - In 2013, AlexNet: The first modern deep convolutional network model is the beginning of a real breakthrough in image classification by deep learning technology.
    - AlexNet does not require pre-training and layer-by-layer training, it uses many modern deep network technologies for the first time
  - With the popularity of massively parallel computing and GPU equipment, the computing power of computers has been greatly improved. In addition, the scale of data available for machine learning is also increasing. With the support of computing power and data scale, computers can already train large-scale artificial neural networks.

Milestone in short

- 1958 Rosenblatt → Sensor
- 1969 Minsky → XOR
- 1986 Hinton, LeCun → Artificial neural network (BP algorithm)
- 1998 LeCun → Convolutional Neural Network
- 2006 Hinton → Deep Network

Deep learning revolution

- AI field
  - Speech recognition: can reduce the word error rate from 1/4 to 1/8
  - Computer vision: target recognition, image classification, etc.
  - Natural language processing: distributed representation, machine translation, question answering, etc.
  - Information retrieval, social network

Early academic institutions

- Toronto University
  - Hinton 1975, Ph.D. of University of Edinburgh
- NYU
  - Lecun (Now Facebook) 1987, Hinton's postdoc
- Montreal University
  - Bengio 1991, M. Jordan's postdoc
- IDSIA
  - Jürgen Schmidhuber

2018 Turing Award winner


AAAI-20
Thirty-Fourth AAAI Conference on Artificial Intelligence
February 7-12 2020, Hilton New York Midtown, New York, New York USA

Widely used deep learning frameworks

- Easy and fast prototyping
- Automatic gradient calculation
- Seamless CPU and GPU switching

Exercise

- https://numpy.org/

- NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Install numpy

- sudo apt install python-pip –y
- pip install numpy

```
ksuo@ksuo-VirtualBox:~$ pip install numpy

Collecting numpy
  Downloading https://files.pythonhosted.org/packages/3a/5f/47e578b3ae79e2624e205445ab77a1848acdaa2929a00eeef6b16e
aaeb20/numpy-1.16.6-cp27-cp27mu-manylinux1_x86_64.whl (17.0MB)
    100% |                                | 17.0MB 37kB/s
Installing collected packages: numpy
Successfully installed numpy-1.16.6
```

https://numpy.org/install/

Helloworld using numpy

- test.py

```
import numpy as np
a = "hello world"
print a
```

- Run the code:
  - $ python test.py

```
ksuo@ksuo-VirtualBox:~$ python test.py
hello world
```

Q1: Add, Minus, Multiply and Divide

- Suppose x = np.array([[1, 2], [3, 4]], dtype=np.float64) y = np.array([[5, 6], [7, 8]], dtype=np.float64). Print out
  - x+y, np.add(x,y)¶
  - x-y, np.subtract(x,y)
  - x*y,np.multiply(x, y)
  - x/y, np.divide(x, y)

Add, Minus, Multiply and Divide

vim /home/ksuo

```
import numpy as np

x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)

print x+y
print np.add(x, y)

print x-y
print np.subtract(x, y)

print x*y
print np.multiply(x, y)

print x/y
print np.divide(x, y)
```

fish

```
ksuo@ksuo-VirtualBox ~> python test.py
[[ 6.  8.]
 [10. 12.]]
[[ 6.  8.]
 [10. 12.]]
[[-4. -4.]
 [-4. -4.]]
[[-4. -4.]
 [-4. -4.]]
[[ 5. 12.]
 [21. 32.]]
[[ 5. 12.]
 [21. 32.]]
[[0.2        0.33333333]
 [0.42857143 0.5       ]]
[[0.2        0.33333333]
 [0.42857143 0.5       ]]
```

Q2: Matrix transpose

- Suppose x = np.array([[1, 2], [3, 4]], dtype=np.float64). Perform matrix transpose on x and output the transposed result (Hint: x.T represents the transposition of x)

Q2: Matrix transpose

vim /hom

```
import numpy as np

x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)

print x.T
```

fish

```
ksuo@ksuo-VirtualBox ~> python test.py
[[1. 3.]
 [2. 4.]]
```

Figure sample

- Install python-matplotlib library
  - sudo apt-get install python-matplotlib

```
import numpy as np
from matplotlib import pyplot as plt

x = np.arange(1,11)
y =  2  * x +  5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```

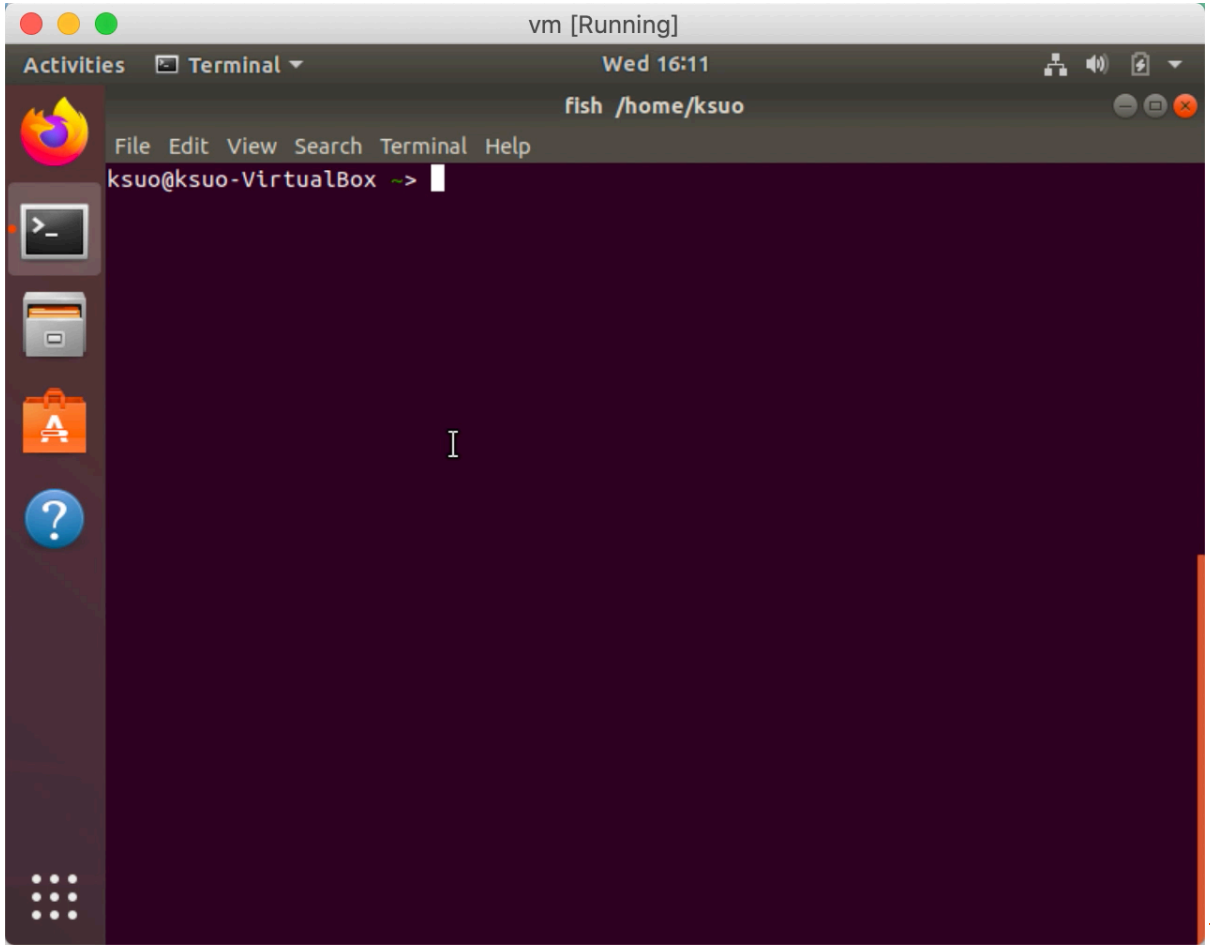


Figure sample

- Draw a figure of y=x*x, which x = np.arange(-50, 100, 0.1)

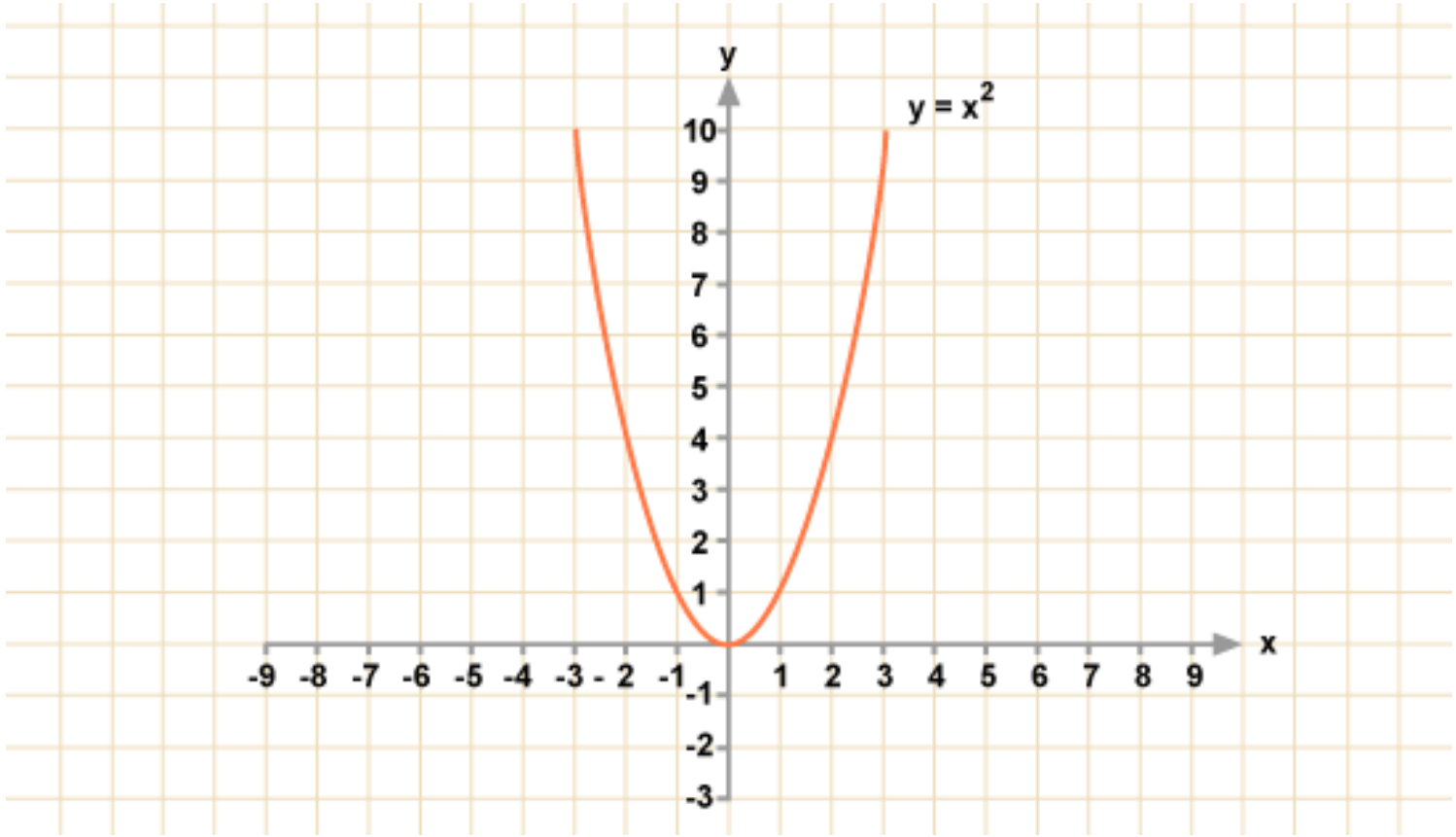


Figure sample

- Draw a figure of y=x*x, which x = np.arange(-50, 100, 0.1)

```
import numpy as np
from matplotlib import pyplot as plt

x = np.arange(-50,100,0.1)
y =  x * x
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```

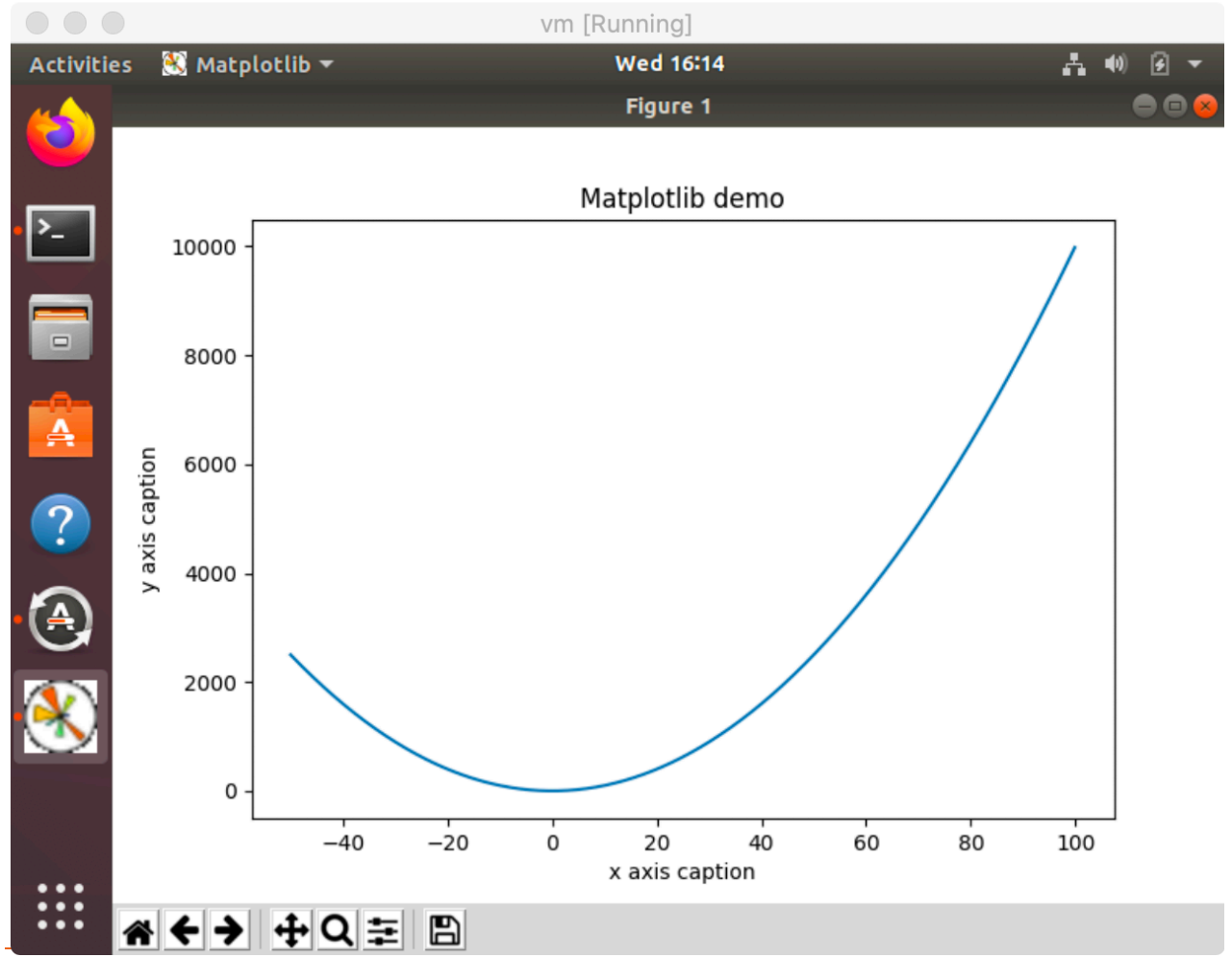


Figure sample

- Draw a figure of sine function and cosine function, which x = np.arange(0, 3 * np.pi, 0.1)
  - (Hint: np.sin(), np.cos() function is used here)

Figure sample

- Draw a figure of sine function and cosine function, which x = np.arange(0, 3 * np.pi, 0.1)
  - (Hint: np.sin(), np.cos() function is used here)

```
import numpy as np
from matplotlib import pyplot as plt

x = np.arange(0,3*np.pi,0.1)
y = np.sin(x)
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```

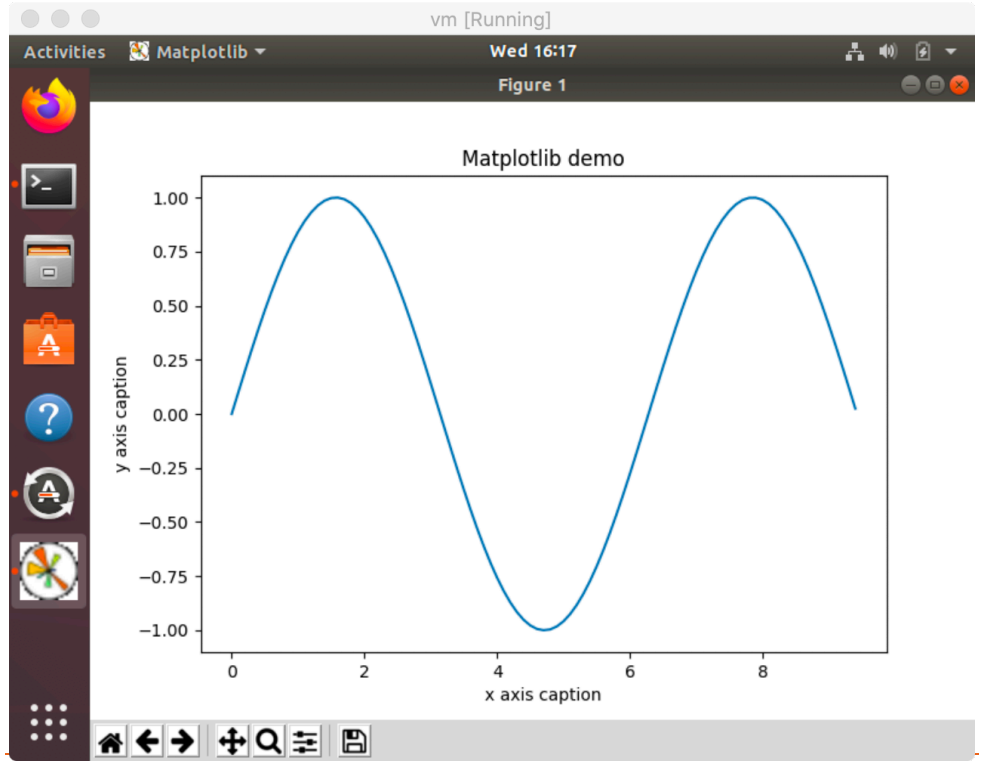


Figure sample

- Draw a figure of sine function and cosine function, which x = np.arange(0, 3 * np.pi, 0.1)
  - (Hint: np.sin(), np.cos() function is used here)

```
import numpy as np
from matplotlib import pyplot as plt

x = np.arange(0,3*np.pi,0.1)
y = np.cos(x)
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```