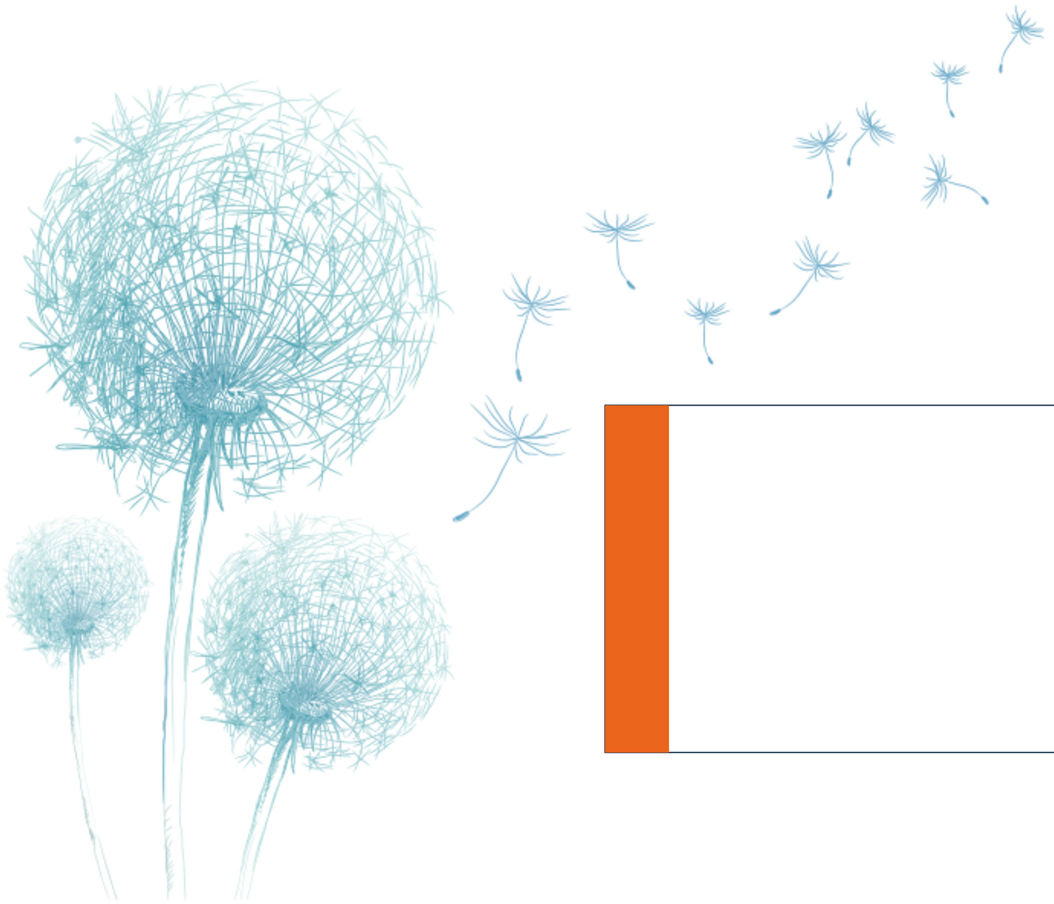
Neural networks and deep learning



Overview

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

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）

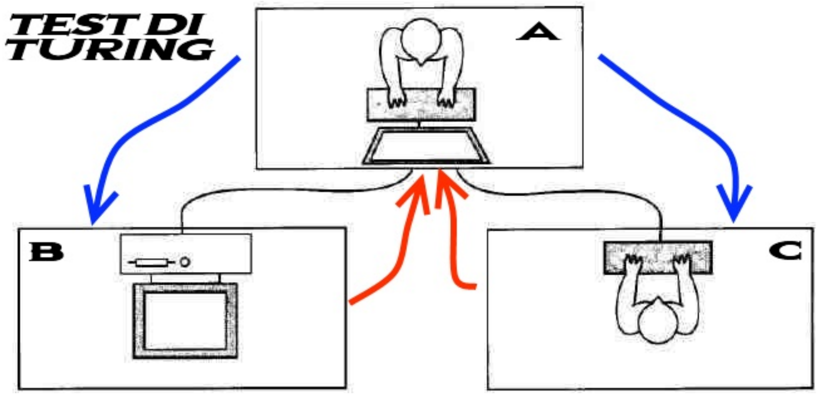
Turing test

Alan Turing

"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 [1950]
《Computing Machinery and Intelligence》

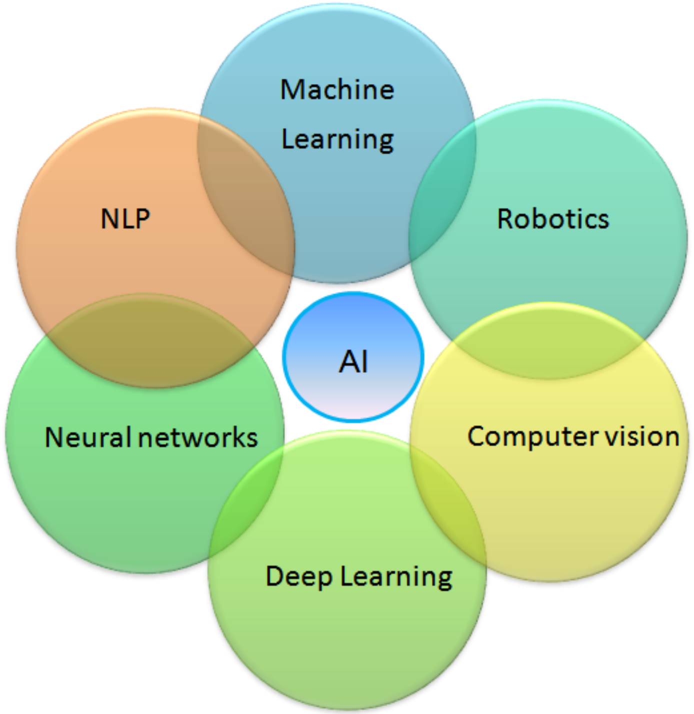
three terminals



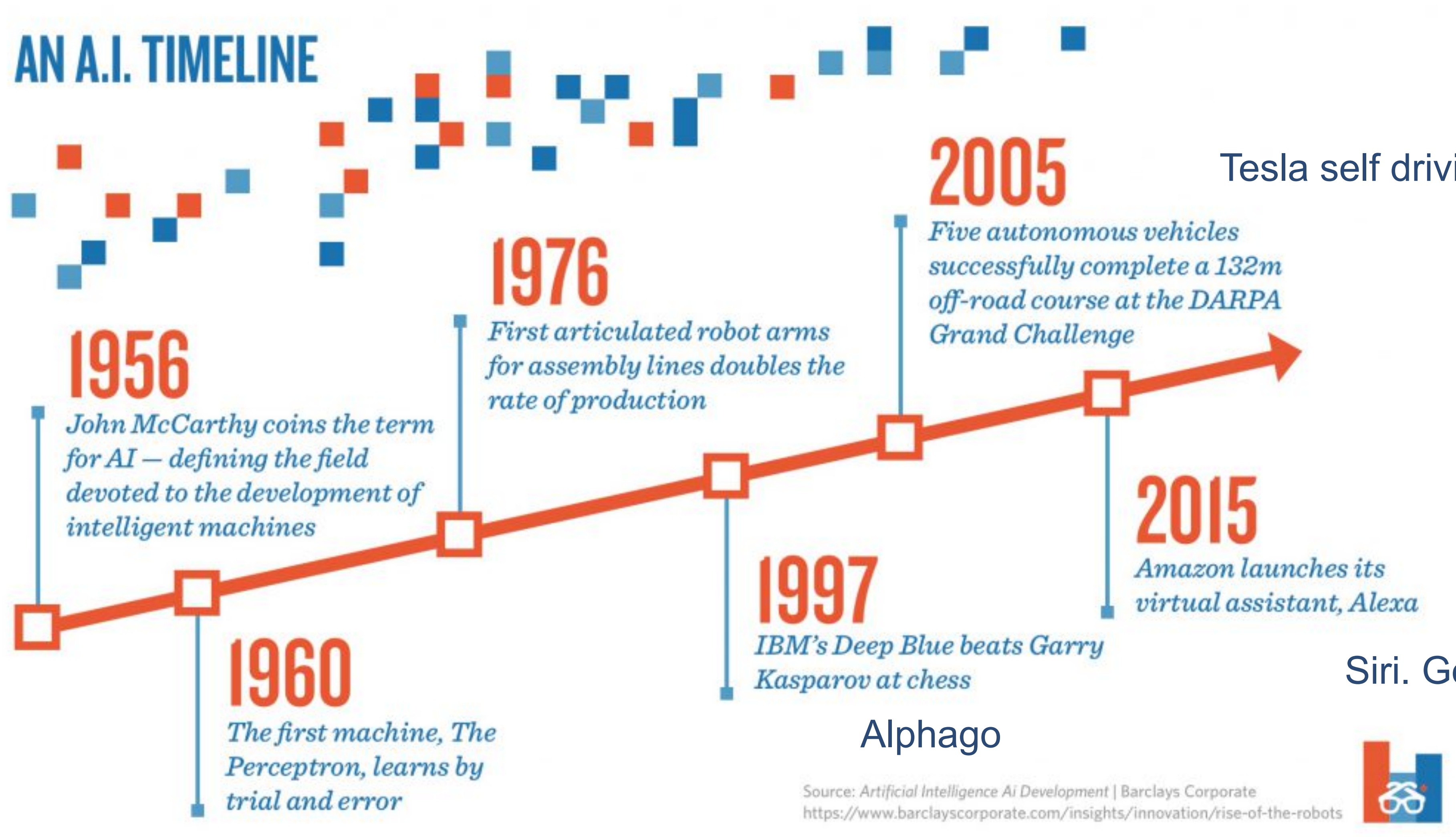
To pass Turing test, the computer must have the ability to ***understand language, learning, memory, reasoning, decision making, etc.***

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



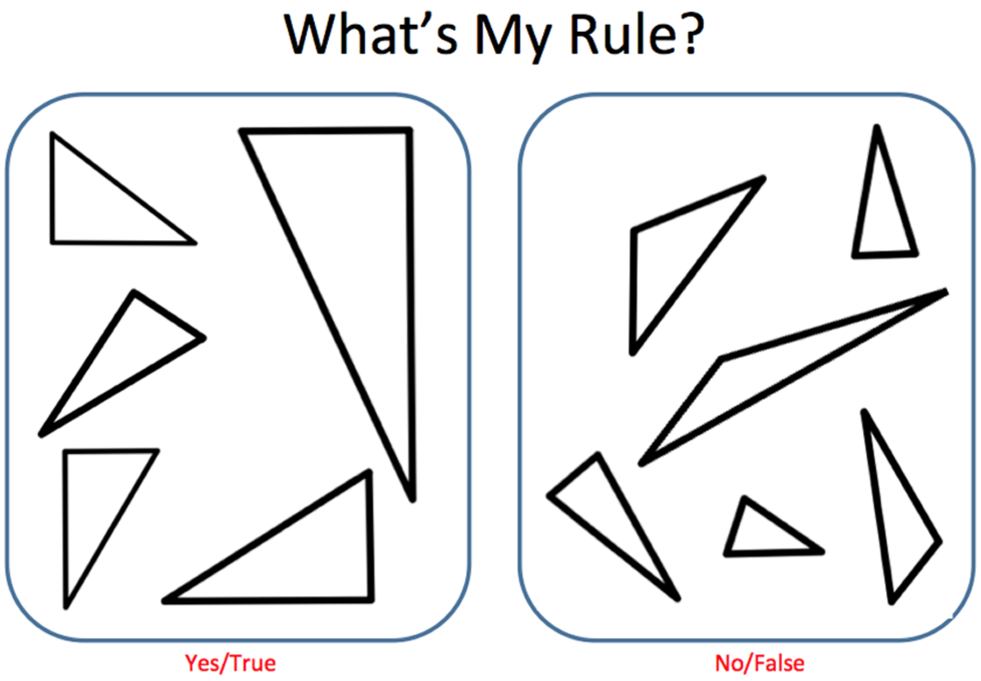
Tesla self driving

Siri. Google assistant…

Alphago

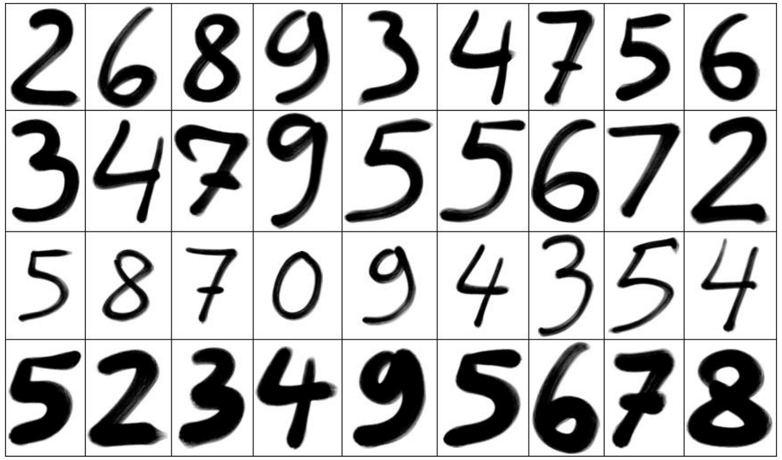
How to develop an artificial intelligence system?

- Expert knowledge (manual rules)



Right angle

What is the Rule?



Machine learning → find the rules

Machine learning ≈ build a mapping function

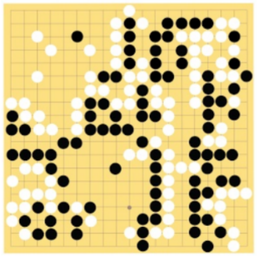
- Speech Recognition

$f($  $)=$ "Hello"

- Image Identification

$f($  $)=$ "9"

- Go

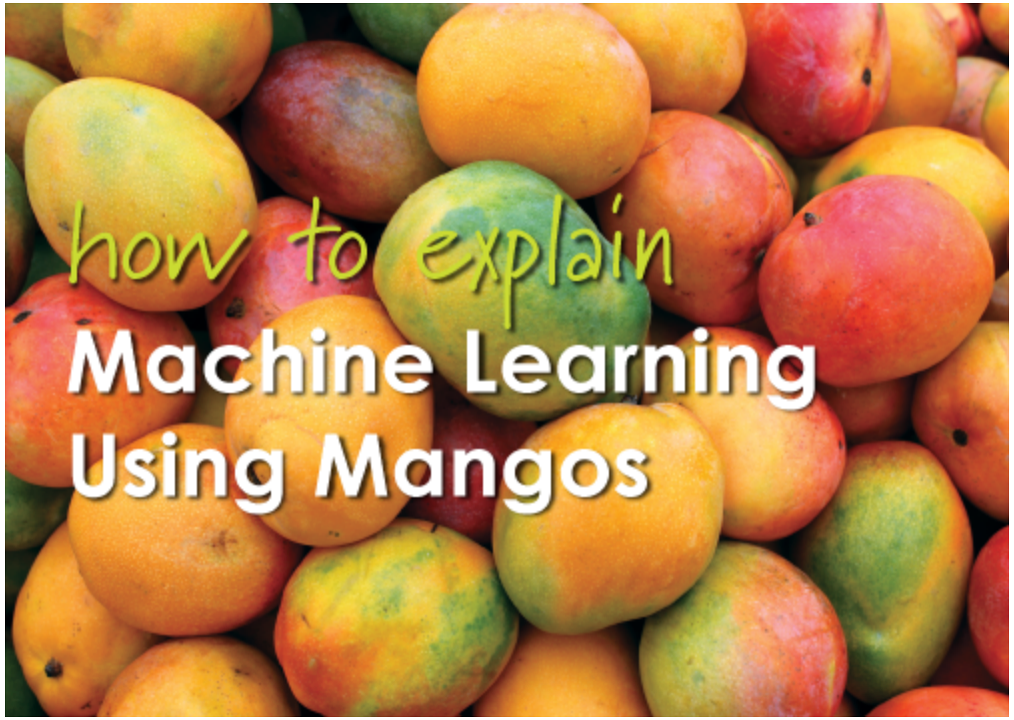
$f($  $)=$ "6-5" (position)

- Machine translation

$f($ "你好！" $)=$ "Hello!"

Example: Mango machine learning

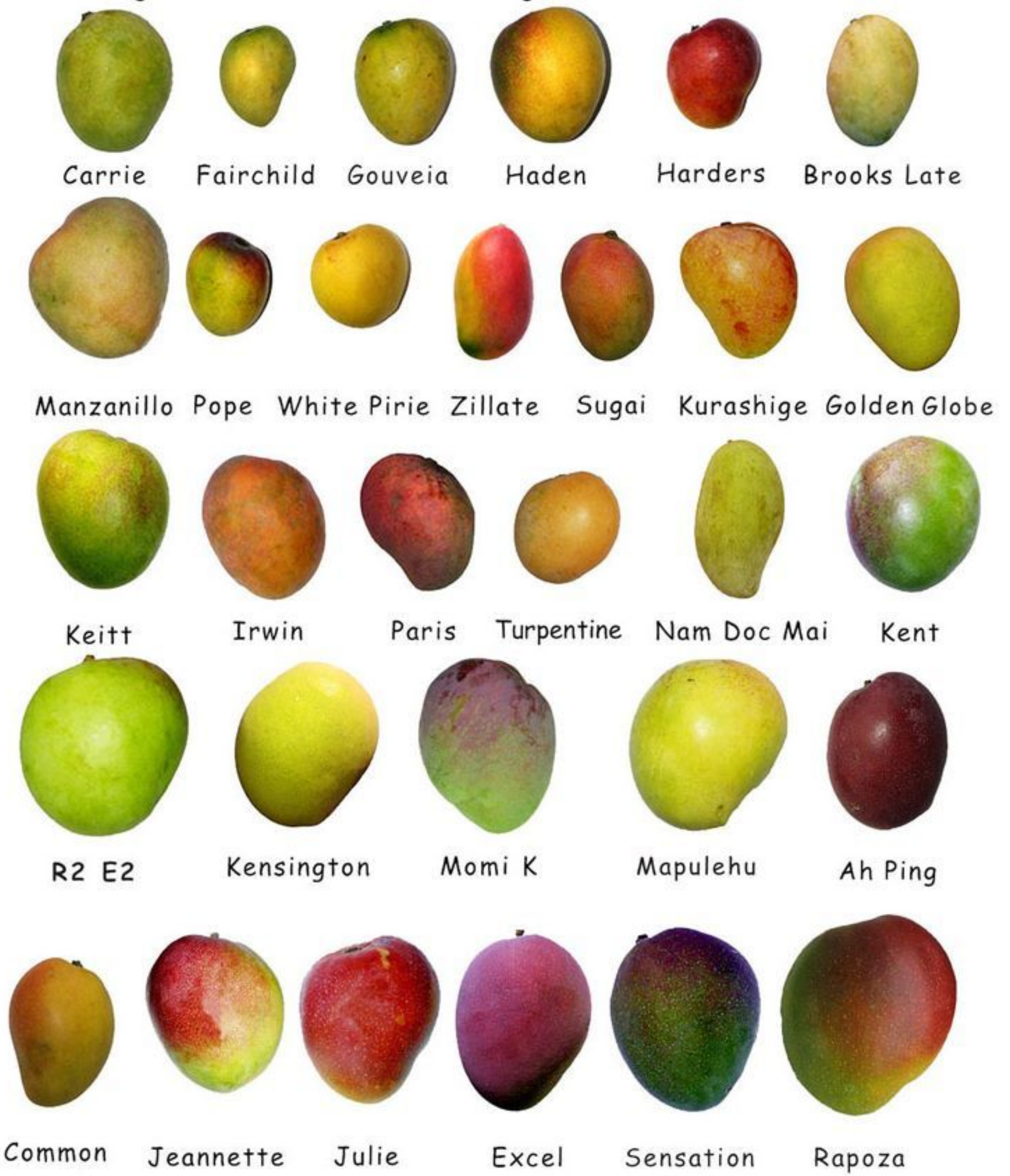
Q: How to determine whether mangoes are sweet?



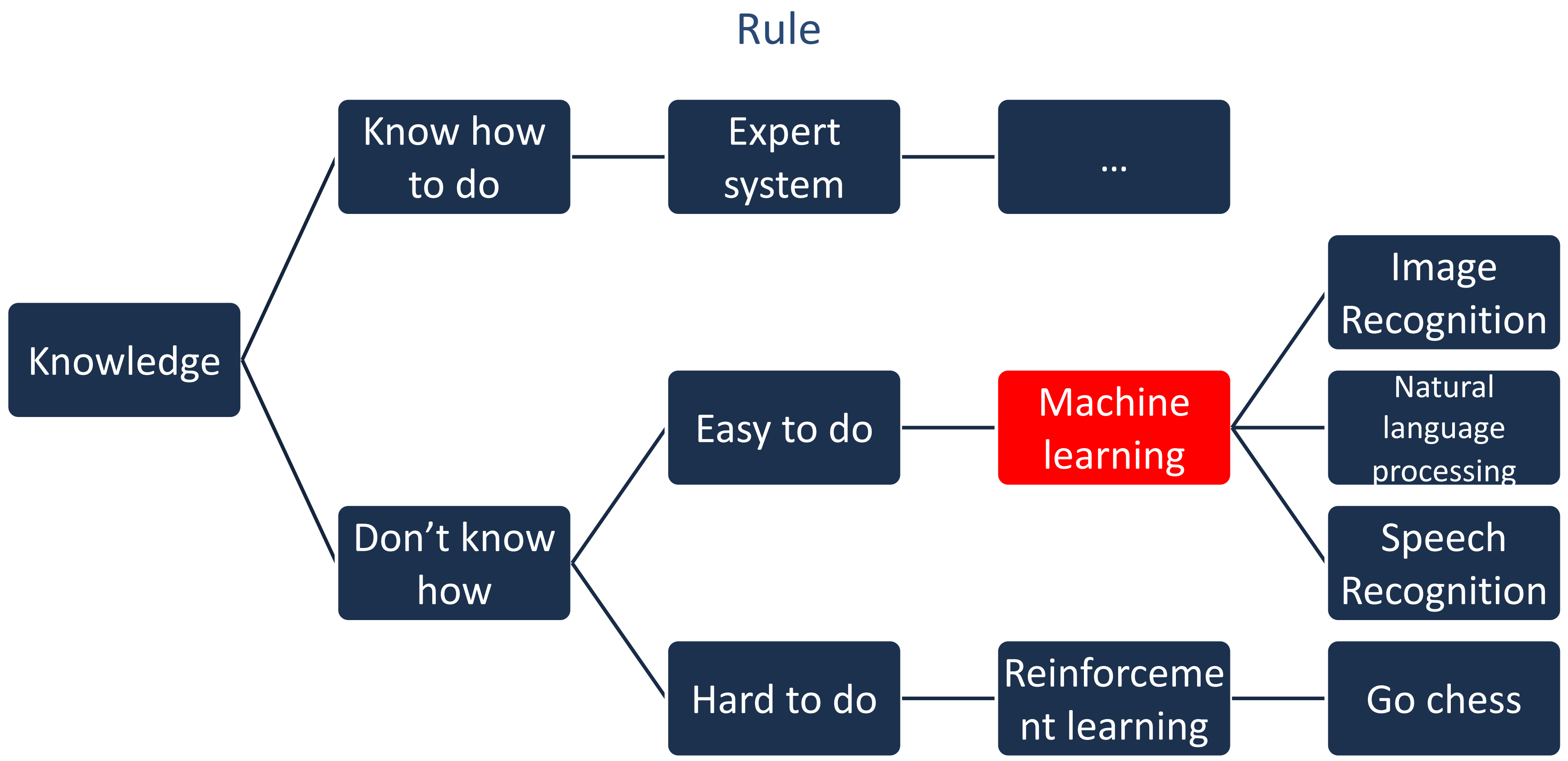
https://www.quora.com/How-do-you-explain-Machine-Learning-and-Data-Mining-to-non-Computer-Science-people

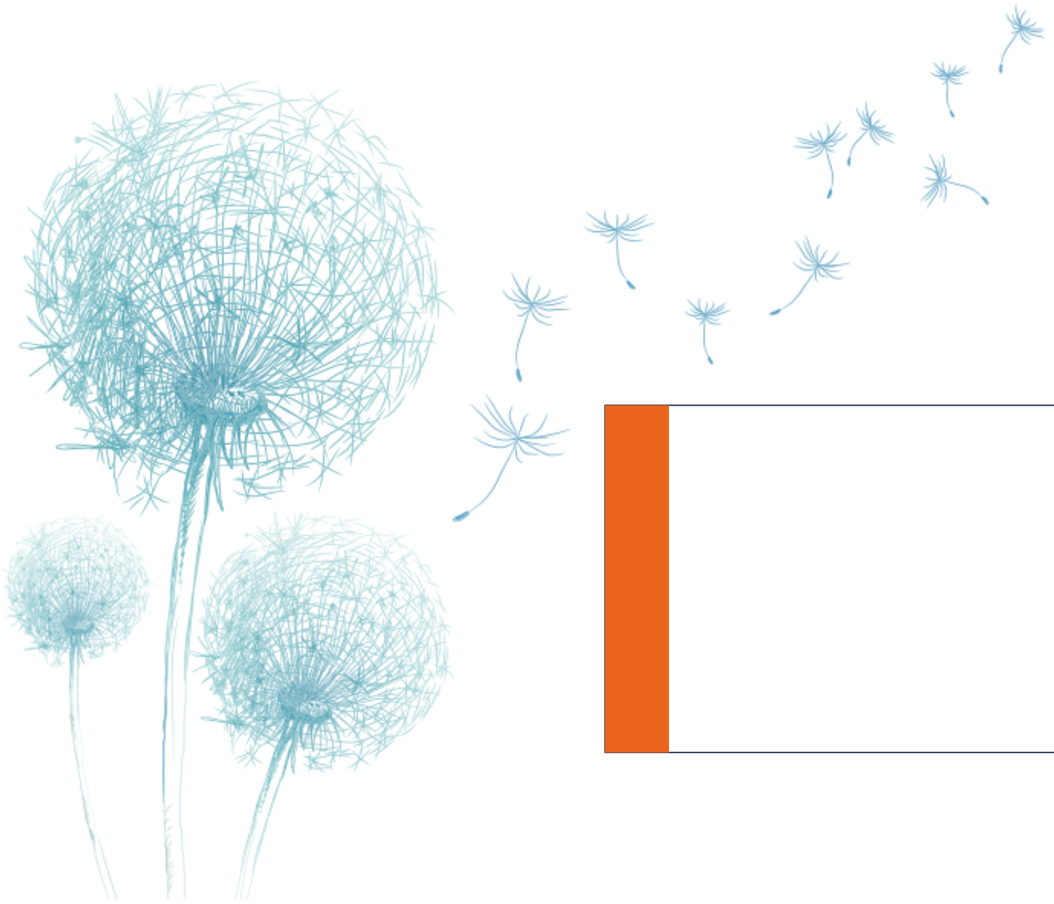
Mango machine learning

- From a randomly selected mango sample (
  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 cc
  characteristics and output variables.
  - f(input) = output
- Next time you buy mangoes from the marl
  model to predict the quality of mangoes b
  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.

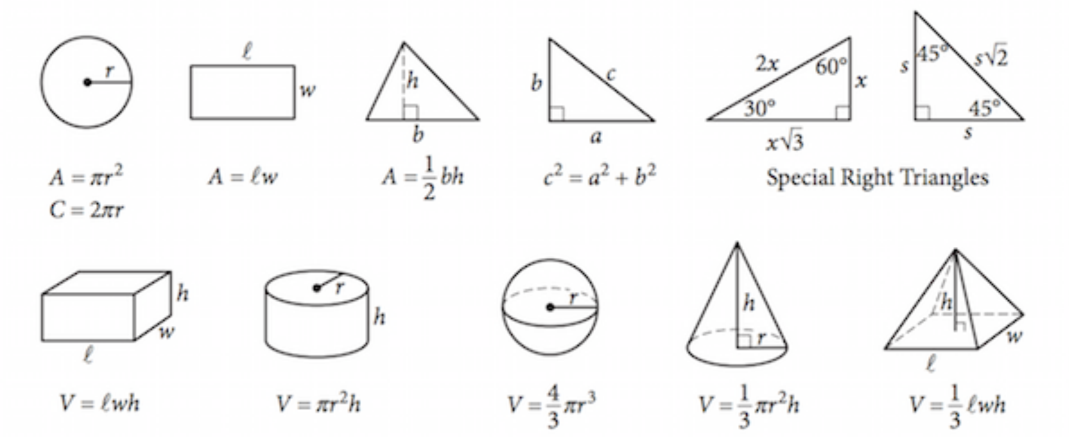
Representation Learning (new)

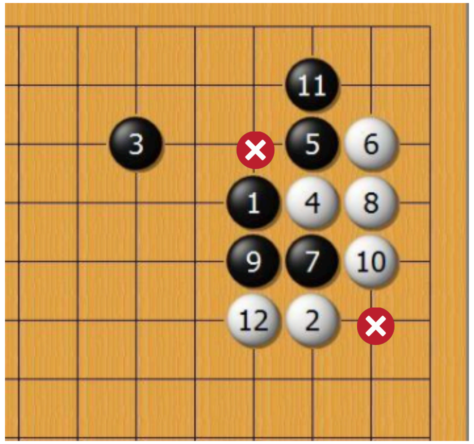
- Data representation is the core issue of machine learning.
  - Feature engineering: requires human intelligence
- Representation learning
  - How to automatically learn good representations from data
- Difficulty
  - No clear goal

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

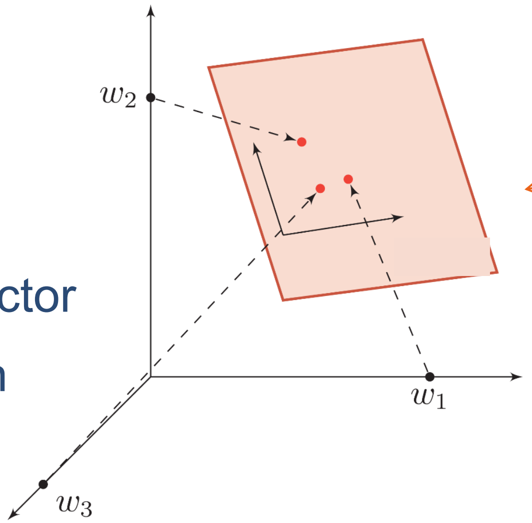
- How to represent semantics in a computer?

Local representation

Distributed representation

Knowledge base, rules

Embedding: compressed, low-dimensional, dense vector



Low dimension

One-Hot vector

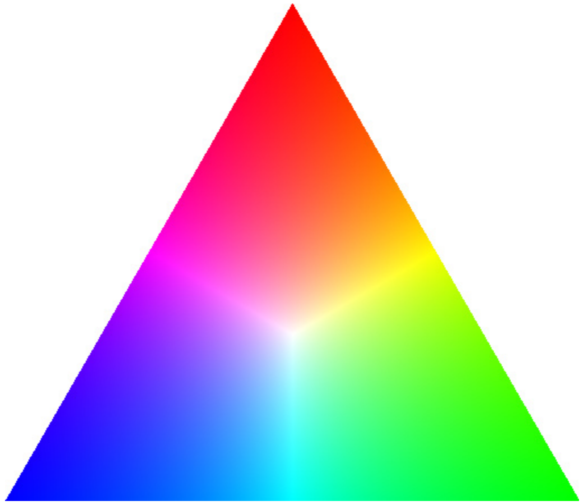
High dimension

Local representation

- Each element represents an entity

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

Color	Local representation
Amber	$[1,\ 0,\ 0,\ 0]^{\mathrm{T}}$
Sky blue	$[0,\ 1,\ 0,\ 0]^{\mathrm{T}}$
Red	$[0,\ 0,\ 1,\ 0]^{\mathrm{T}}$
Brown	$[0,\ 0,\ 0,\ 1]^{\mathrm{T}}$



Local representation

- Advantage

  1) Easy to model

  2) High calculation efficiency for linear model

- Disadvantage

  1) The dimension is high and cannot be expanded

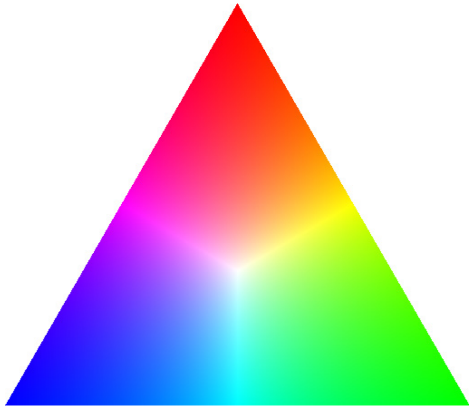
  2) Unable to calculate the similarity between (e.g., color)

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

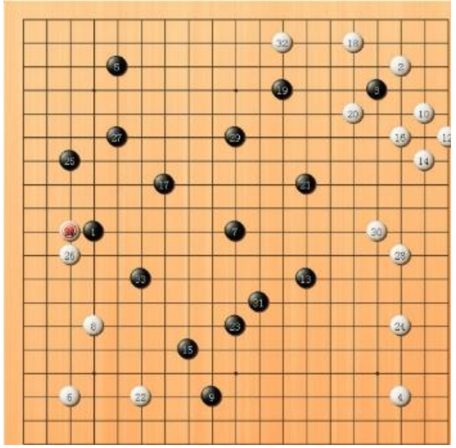
3) Able to calculate the similarity between (e.g., color comparison, tool)

Local vs. Distributed 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]

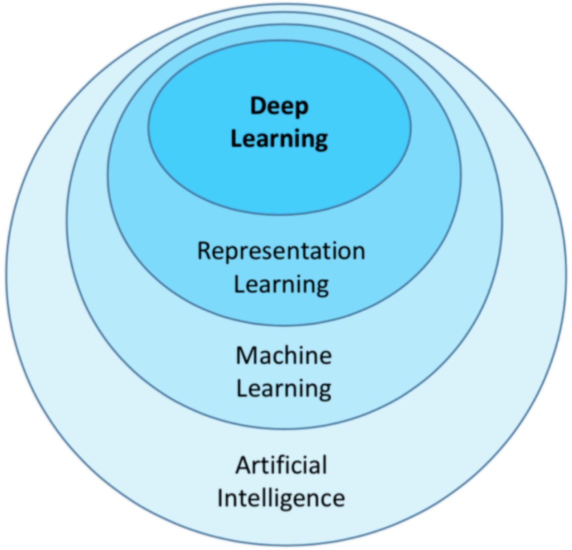
- Distributed representation
  - Compressed, low-dimensional, dense vector

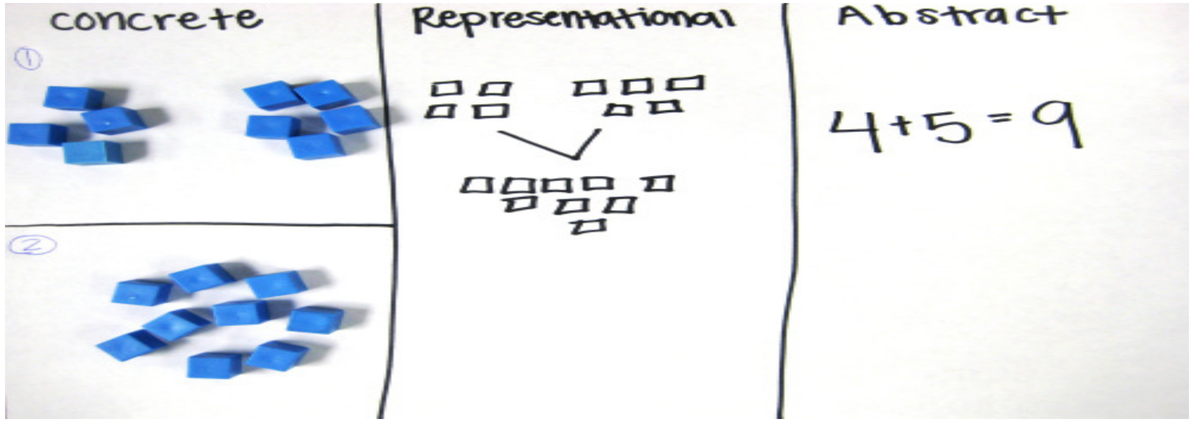


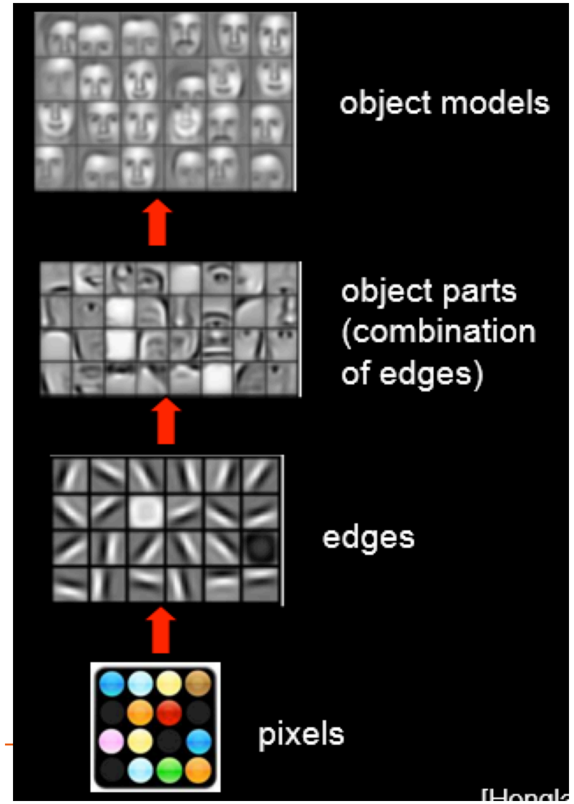
Distributed representation

Representation learning vs. 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

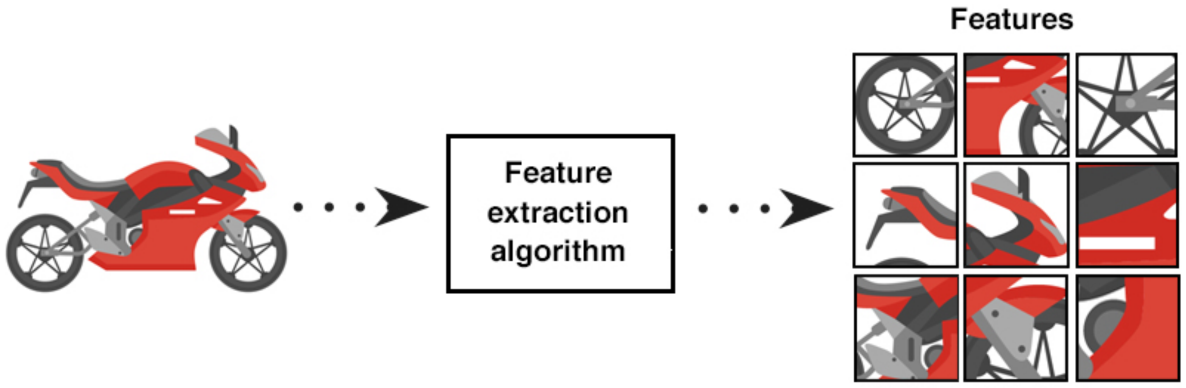






Traditional feature extraction

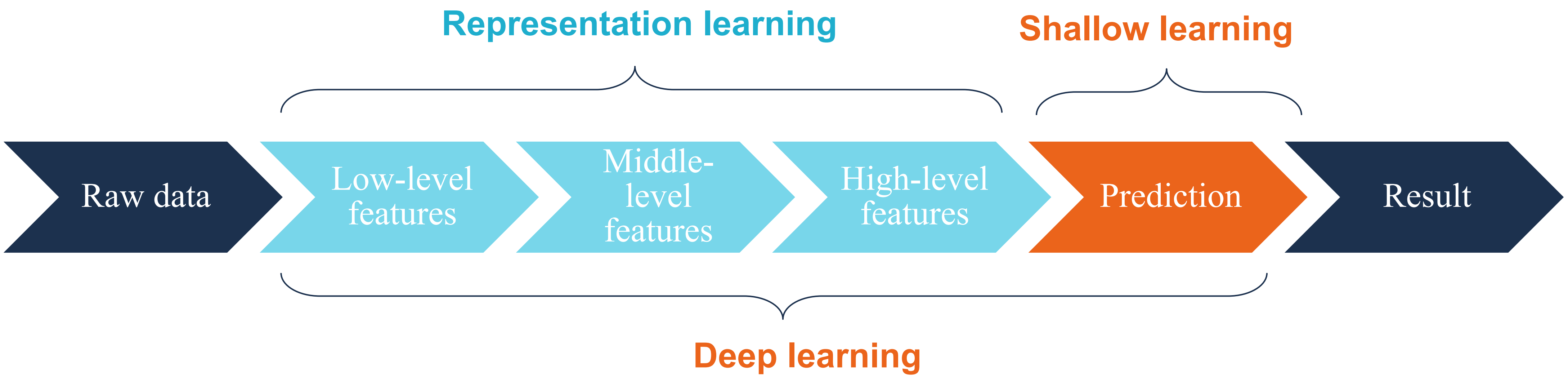
- Feature extraction algorithm
  - Linear projection (subspace)
  - Nonlinear embedding
  - Autoencoder
  - …



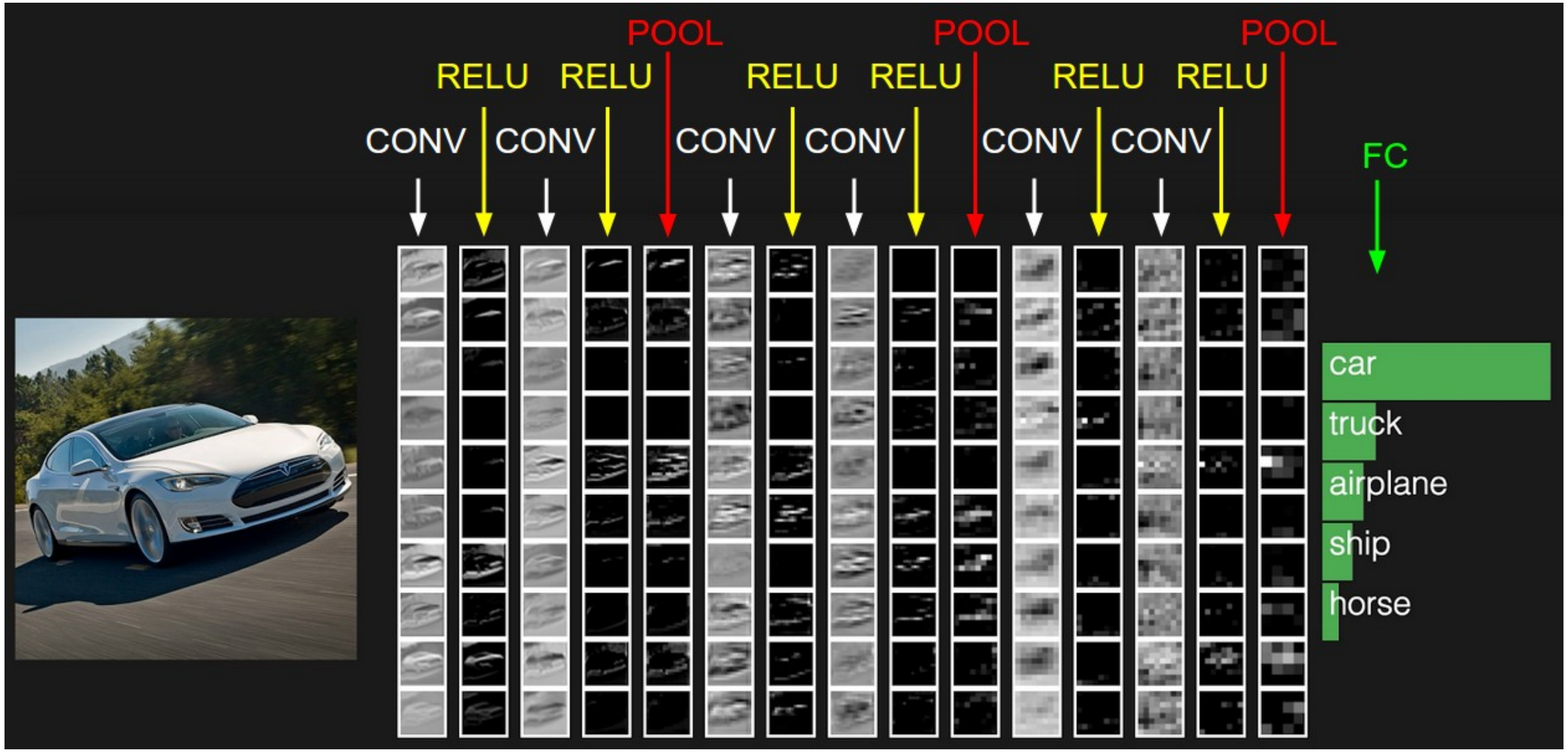
- Feature extraction vs. representation learning
  - Feature extraction: remove useless features based on task or prior pair
  - Representation learning: learning *high-level semantic features* through deep models

Deep learning

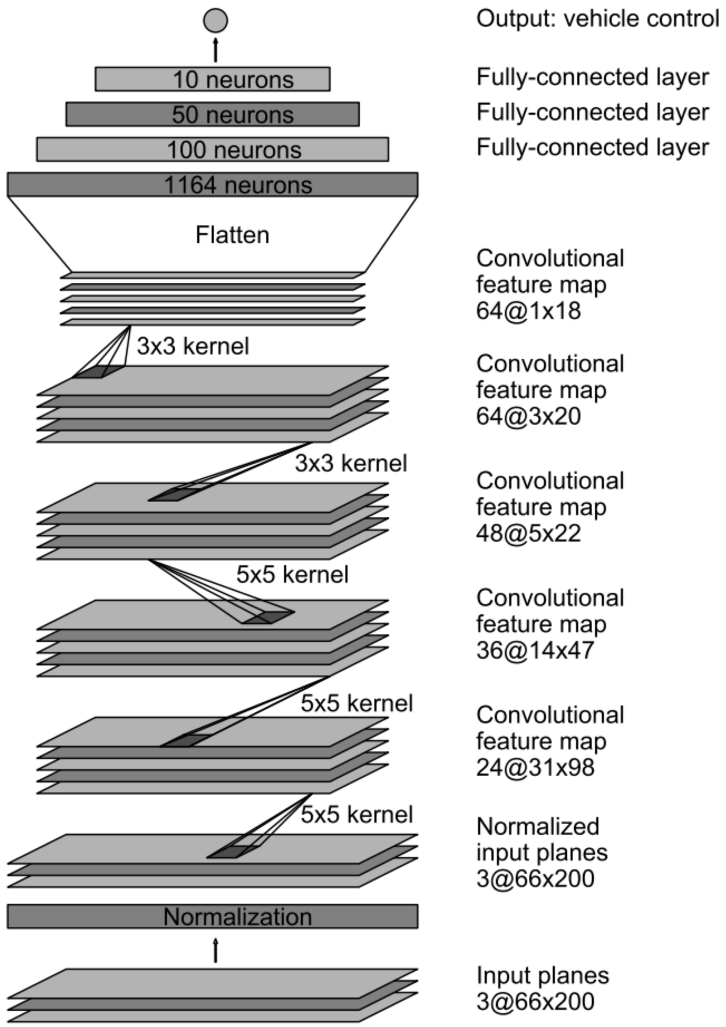
- By building a model with a certain "depth", the model can automatically learn a good feature representation (from low-level features, to middle-level features, and then to high-level features), thereby ultimately improving the accuracy of prediction or recognition



Deep learning: image recognition



Deep learning: self driving car



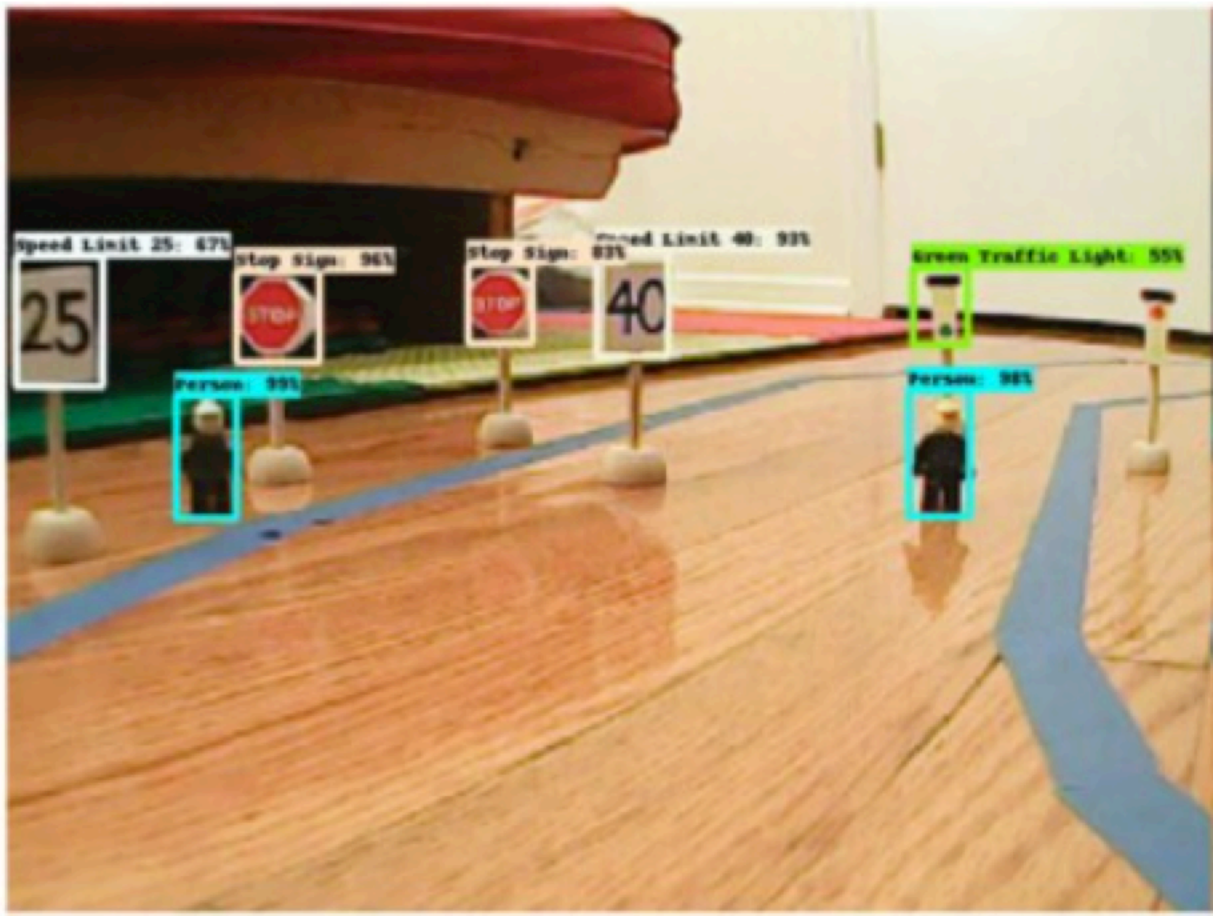


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

Deep learning: self driving car

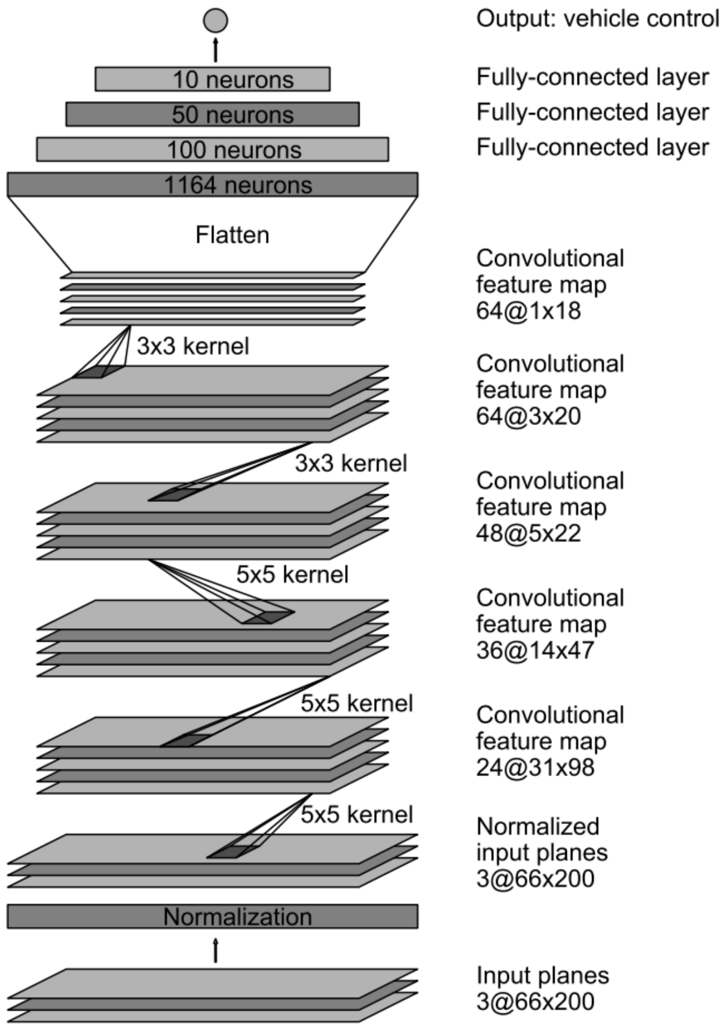
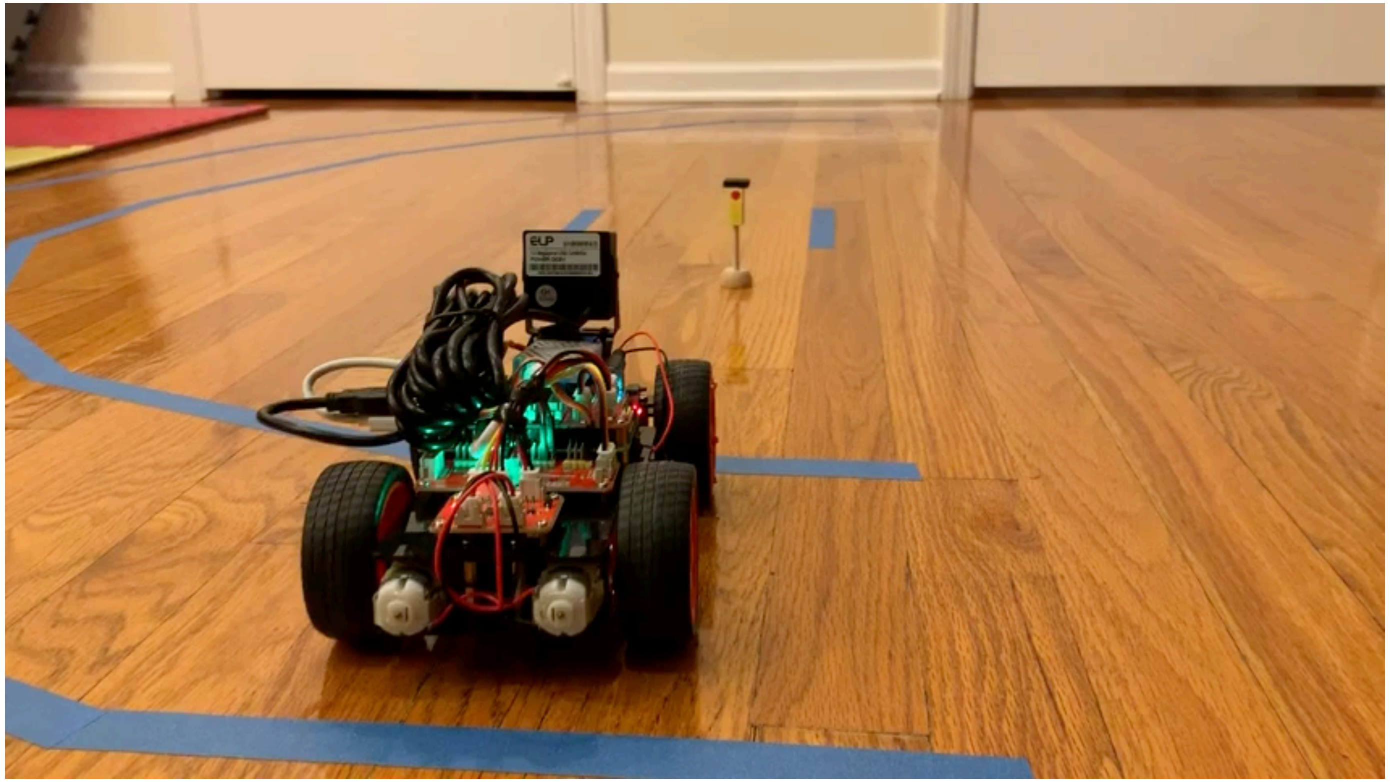


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.



Mathematical description of deep learning

Output of 1st layer

Input

$$y = f(x)$$

Low-level learning

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

Output of nth layer

$$\dots$$

$$y = f^5(f^4(f^3(f^2(f^1(x)))))$$

Deep-level learning

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

Widely used deep learning frameworks

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

PaddlePaddle
Microsoft CNTK
TensorFlow
Chainer
NumPy
Caffe2
Keras
APACHE mxnet
PYTORCH
theano

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

```python
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

```
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
```

```
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()
```

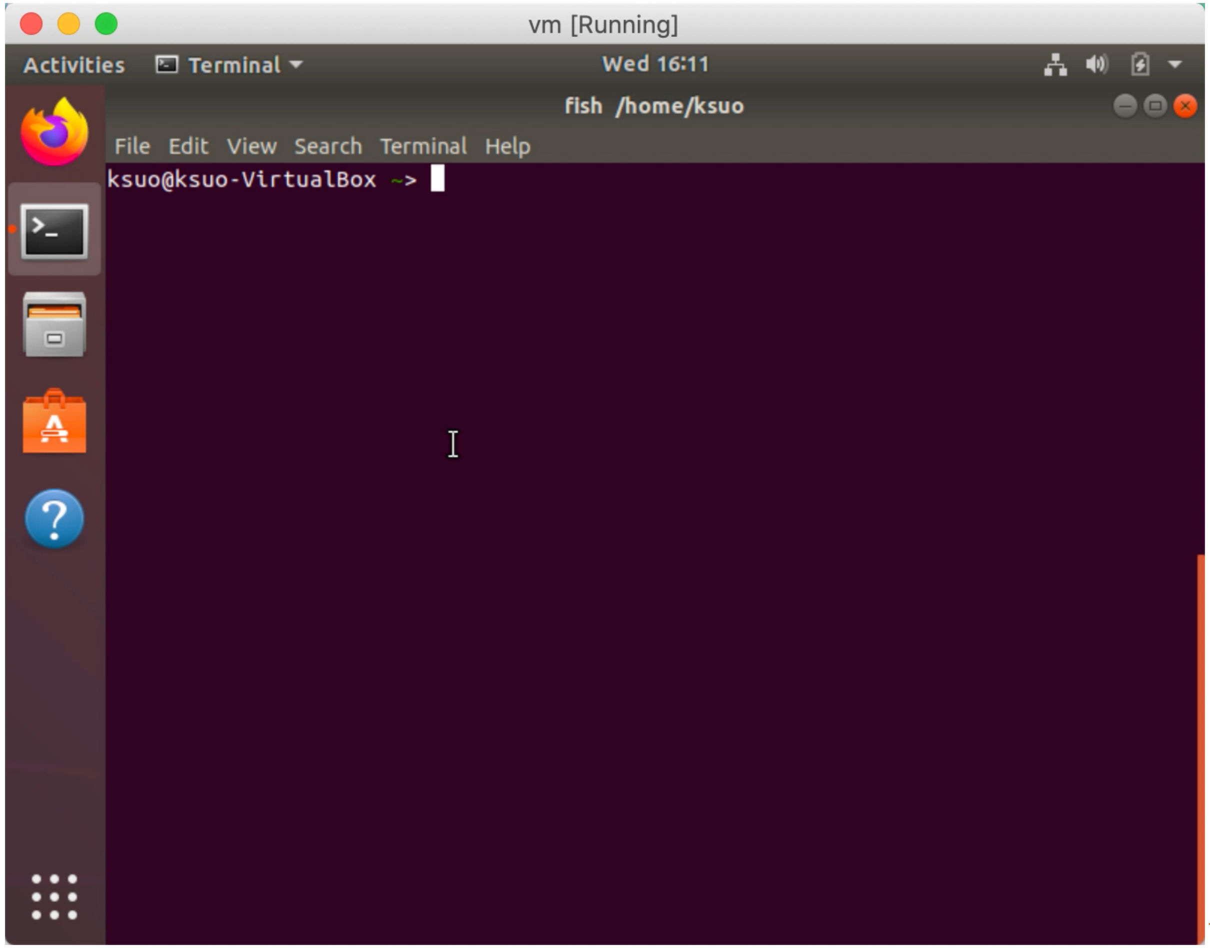


Figure sample

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

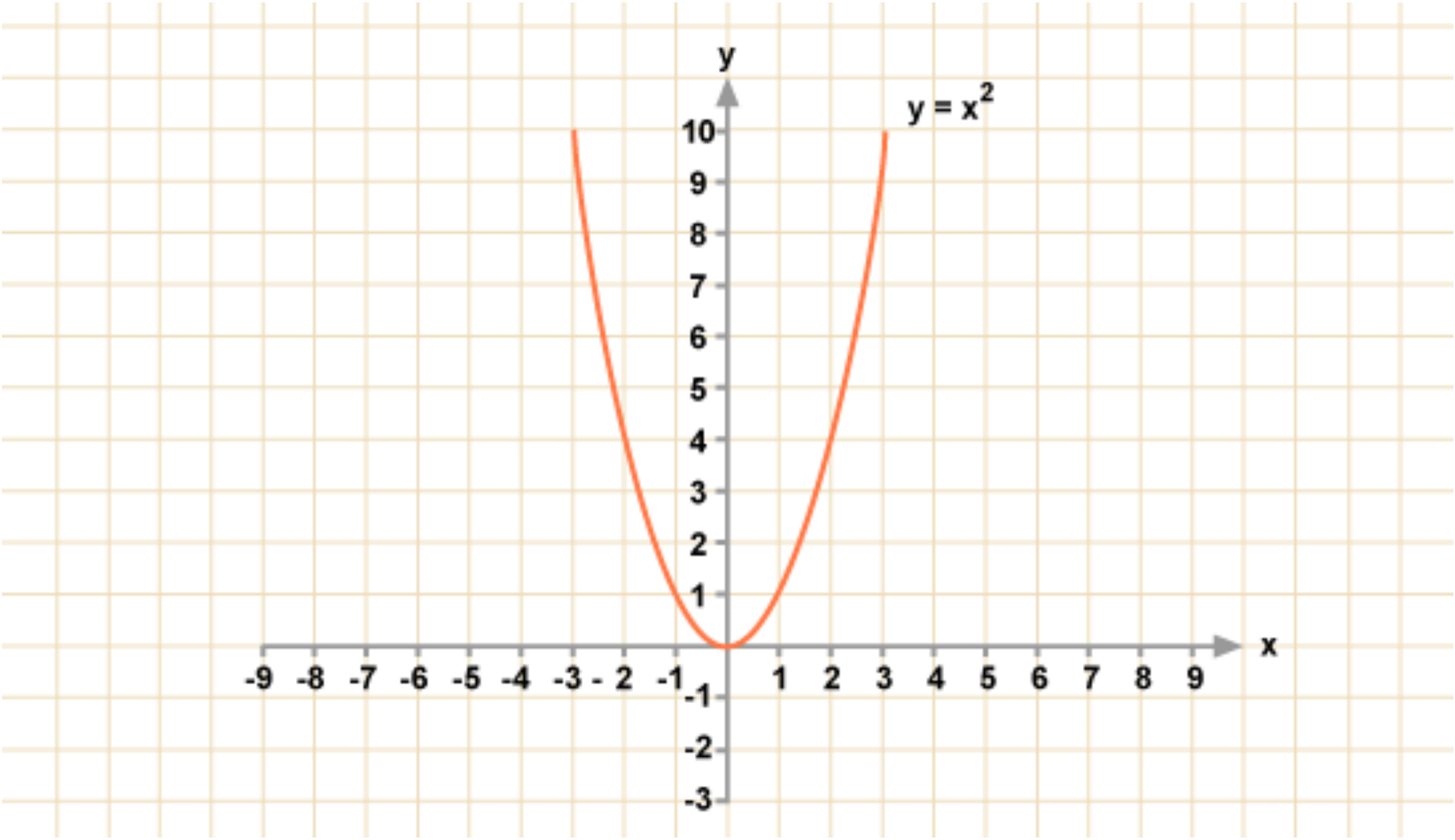


Figure sample

- 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()
```

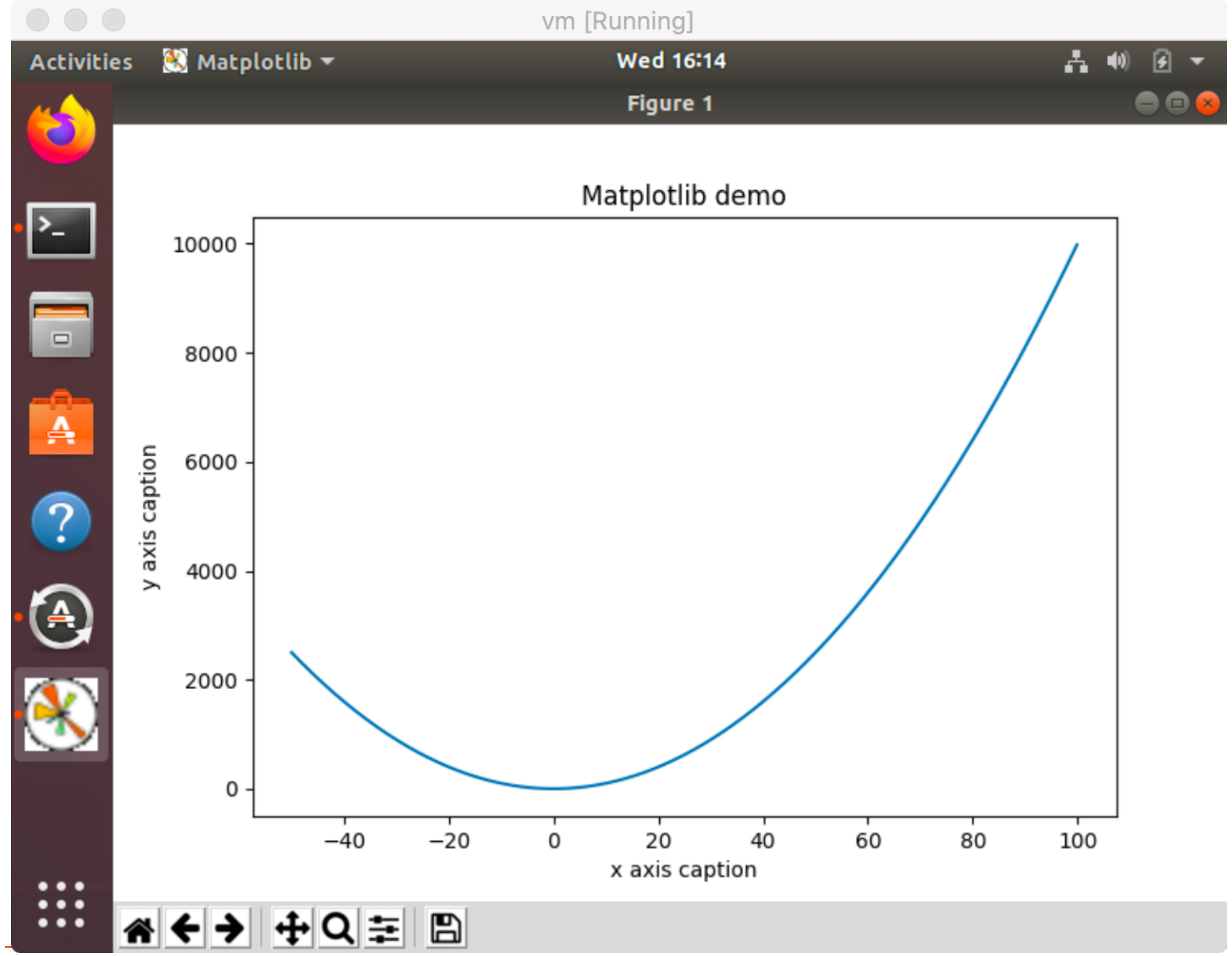


Figure sample

- 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)

```python
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()
```

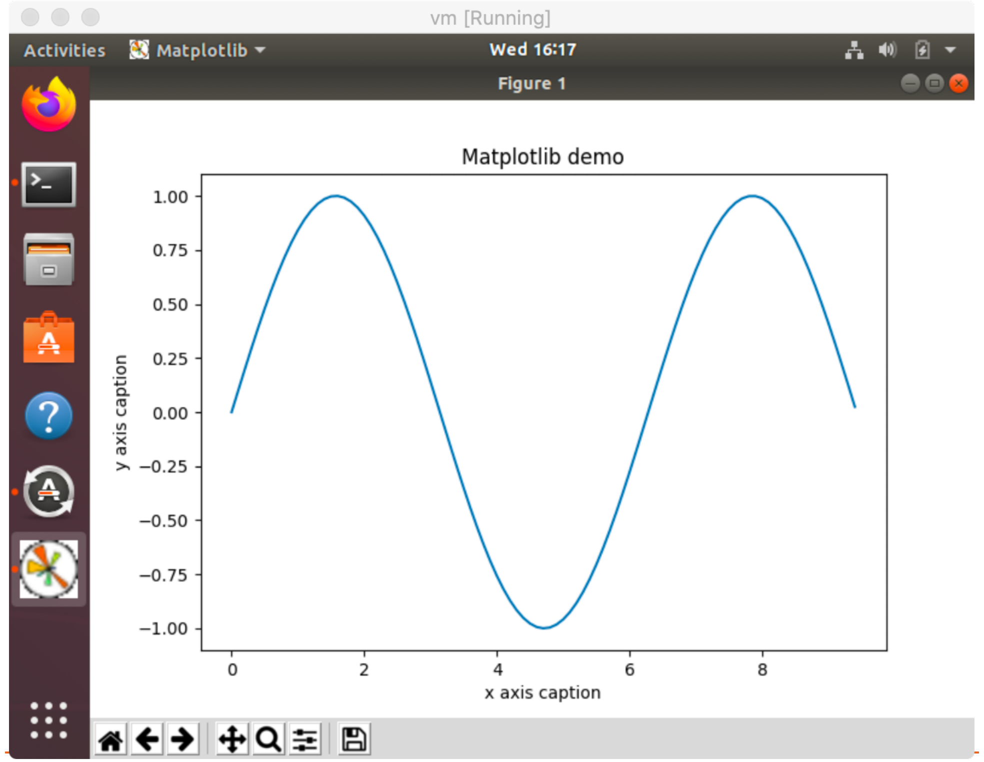
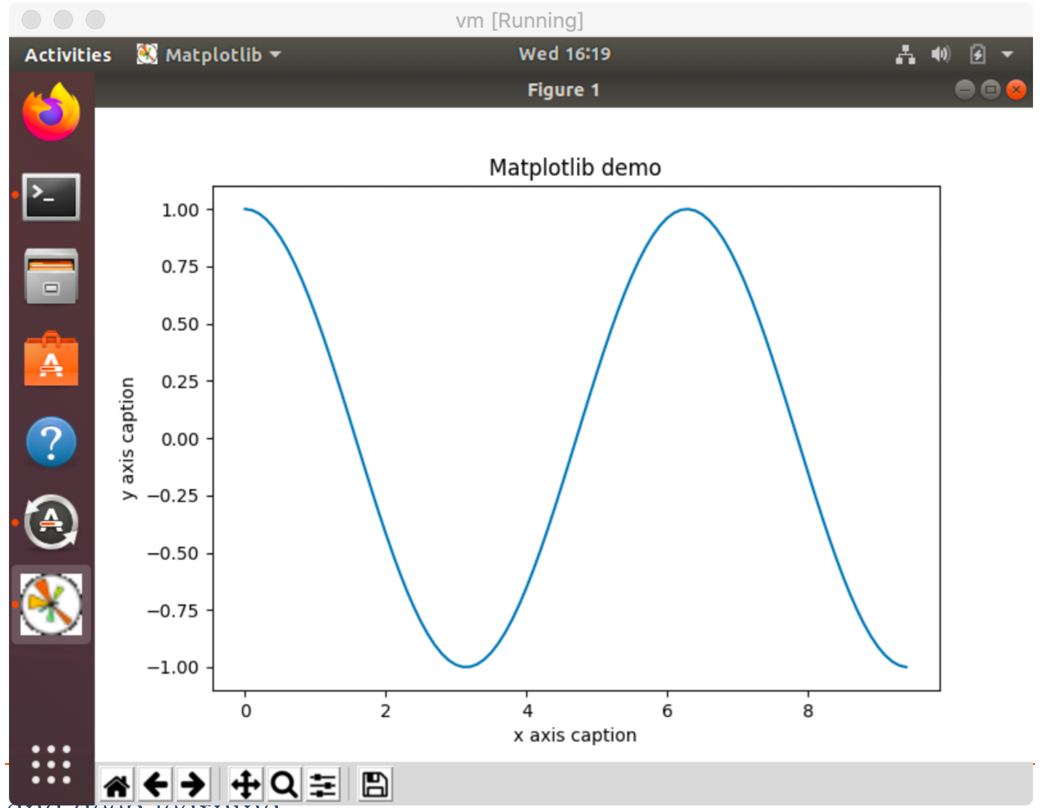


Figure sample

- 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)

```python
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()
```



Summary

- **Artificial intelligence**
  - Definition, Turing test, Rule, Machine learning
- **Deep learning**
  - Feature processing, Representation learning
  - Local representation vs. Distributed representation
- **Neural networks**
  - Neuron, artificial neural networks
  - History of neural networks
- **Frameworks**