Tam Wing Fan Innovation Wing

Welcome to

# Astar!

AI and Robotics: An Introduction by HKU Astar

HKU Astar · 2023 · 10 · 8 · HelloElwin & Nebula

Astar's

Rough Introduction to

# Machine Learning

*Astar's Rough Introduction to*

# Machine Learning

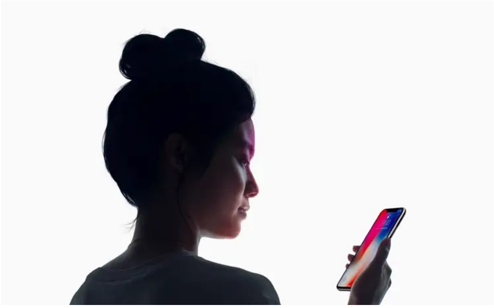
Astar's Rough Introduction to

# Machine Learning

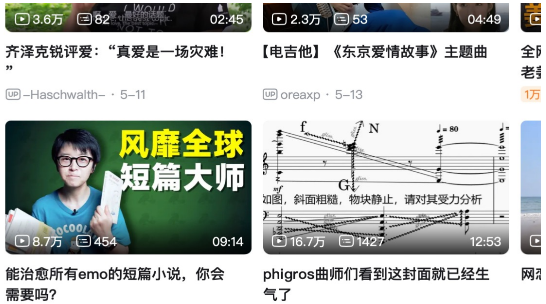
1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
   a. Multilayer Perceptron (MLP)
   b. Convolutional Neural Networks (CNN)
   c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
   a. Representation Learning
   b. The Encoder-Decoder Architecture and Autoencoder
   c. Generative Adversarial Network (GAN)
   d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

# Where is Artificial Intelligence?

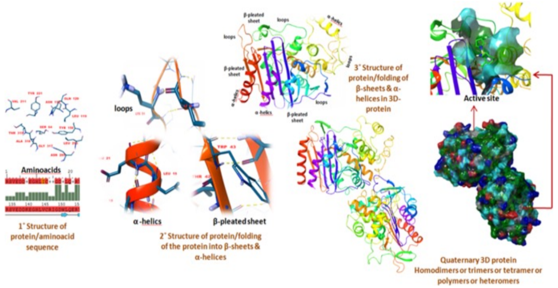

Alpha Go



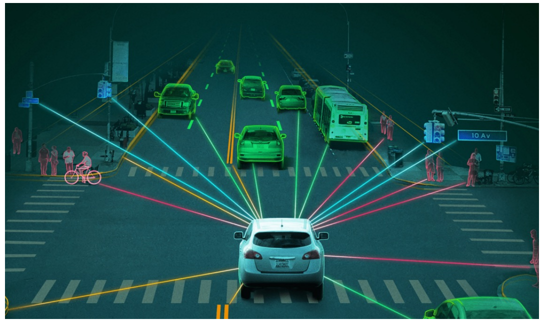
Face Recognition



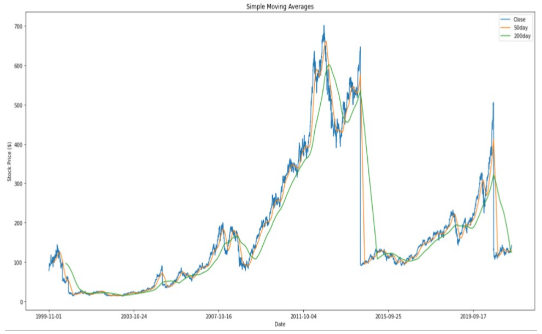

Recommender Systems



Drug Discovery

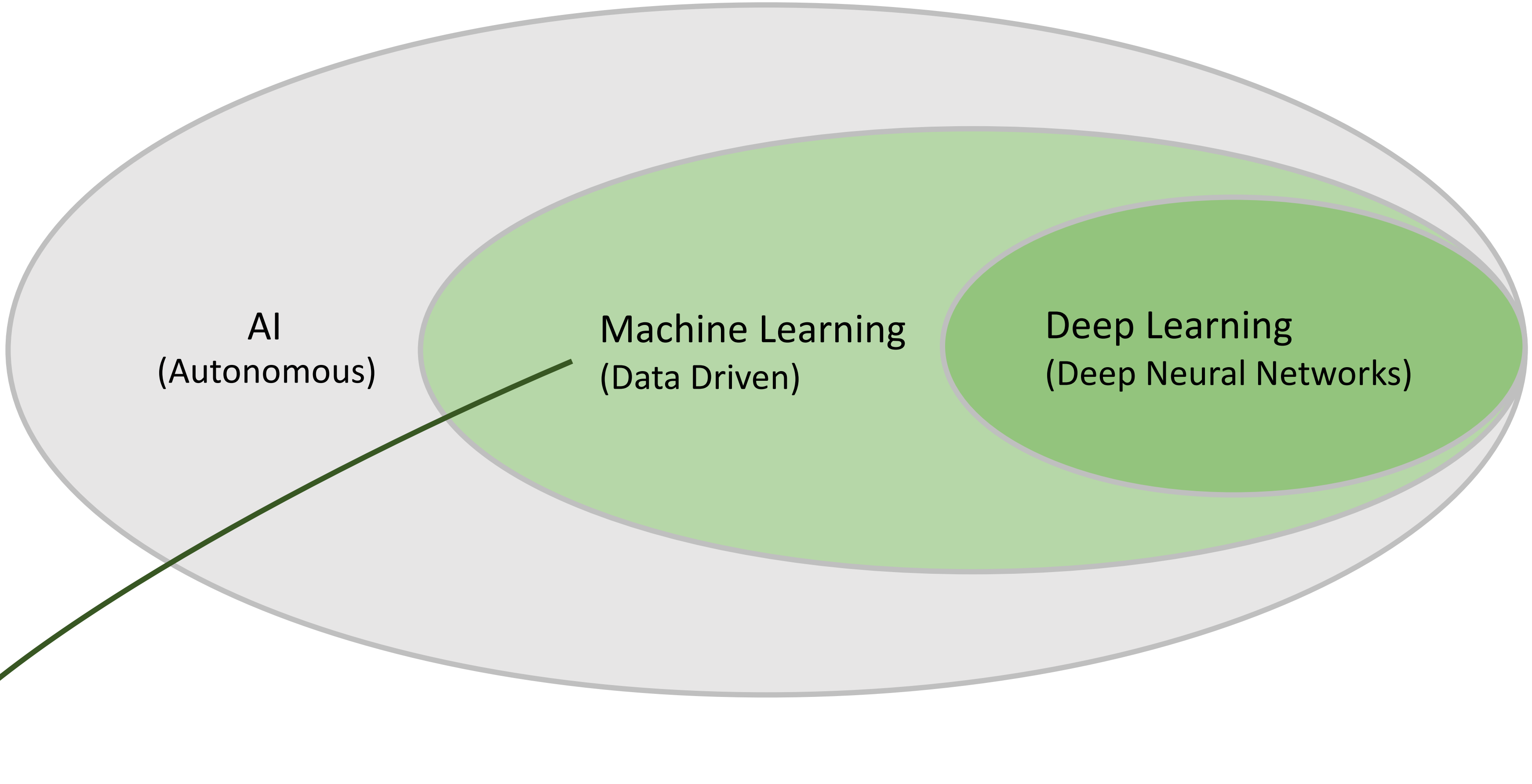


Autonomous Driving



Market Prediction

# What is Machine Learning?




Field of study that gives computers the ability to learn without being explicitly programmed. *Arthur Samuel*

Astar's Introduction to

# Machine Learning

1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

A General Machine Learning Paradigm : Supervised Learning

# How can a machine learn ?

数据 Dataset

*本课关注：
有监督学习
Supervised Learning

这是什么问题的数据？
- 一个小车识别的数据集？（看一看）
- 一个西瓜评估的数据集？
- 一个推荐系统的数据集？

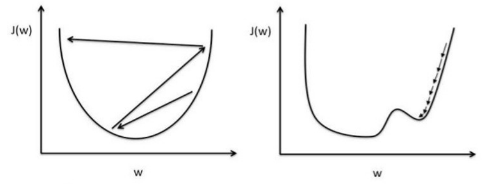
模型 Model

这个问题适合什么模型？
- 线性模型？( Linear Model )
- 多层感知机？( Multilayer Perceptron )
- 卷积神经网络？( Convolutional Neural Network )

A General Machine Learning Paradigm : Supervised Learning

# How can a machine learn ?

Learning rate




Overshooting    Learn too slow

| 训练集<br>Train Set | 验证集<br>Validation Set | 测试集<br>Test Set |
|---|---|---|

拟合训练集
优化模型参数

模型 Model

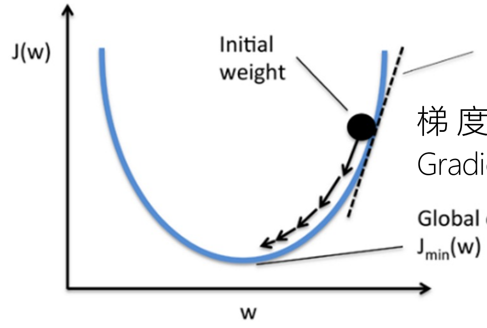
X    Y'    Y

$$\nabla\theta = \frac{\partial}{\partial\theta}\frac{1}{n}\sum_{i=0}^{n}||f^{\theta}(x) - y||_2^2$$

$$\theta = \theta - \alpha\nabla\theta$$

均方误差
Mean Square Error

损失函数
Loss Function




梯度下降法
Gradient Descent

Global cost minimum
$J_{min}(w)$

优化
Optimization

怎么选择/设计损失函数？
- 小车识别？
- 西瓜评估？
- 推荐系统？

A General Machine Learning Paradigm : Supervised Learning

# How can a machine learn ?

训练集
Train Set

验证集
Validation Set

测试集
Test Set

拟合训练集
优化模型参数

调整超参数，观察泛化性能

模型 Model

X

Y′

Y

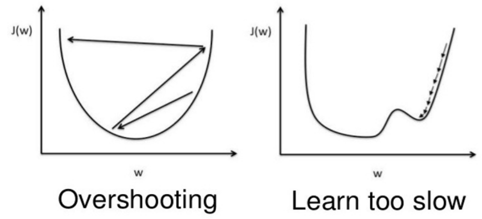
$$\nabla\theta = \frac{\partial}{\partial\theta}\frac{1}{n}\sum_{i=0}^{n}||f^{\theta}(x) - y||_2^2$$

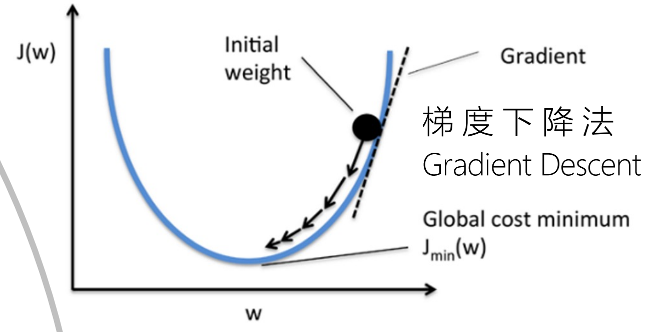
$$\theta = \theta - \alpha\nabla\theta$$

均方误差
Mean Square Error

损失函数
Loss Function

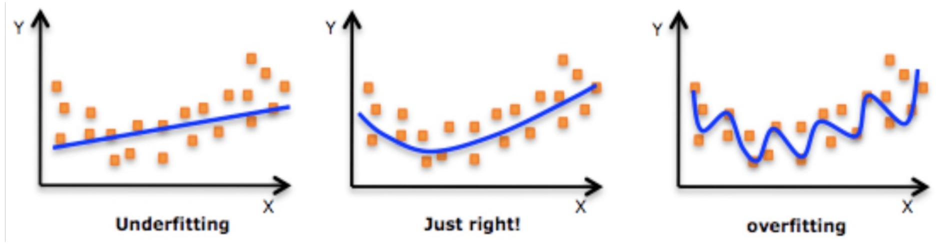
Learning rate




梯度下降法
Gradient Descent

优化
Optimization




泛化能力 Generalization

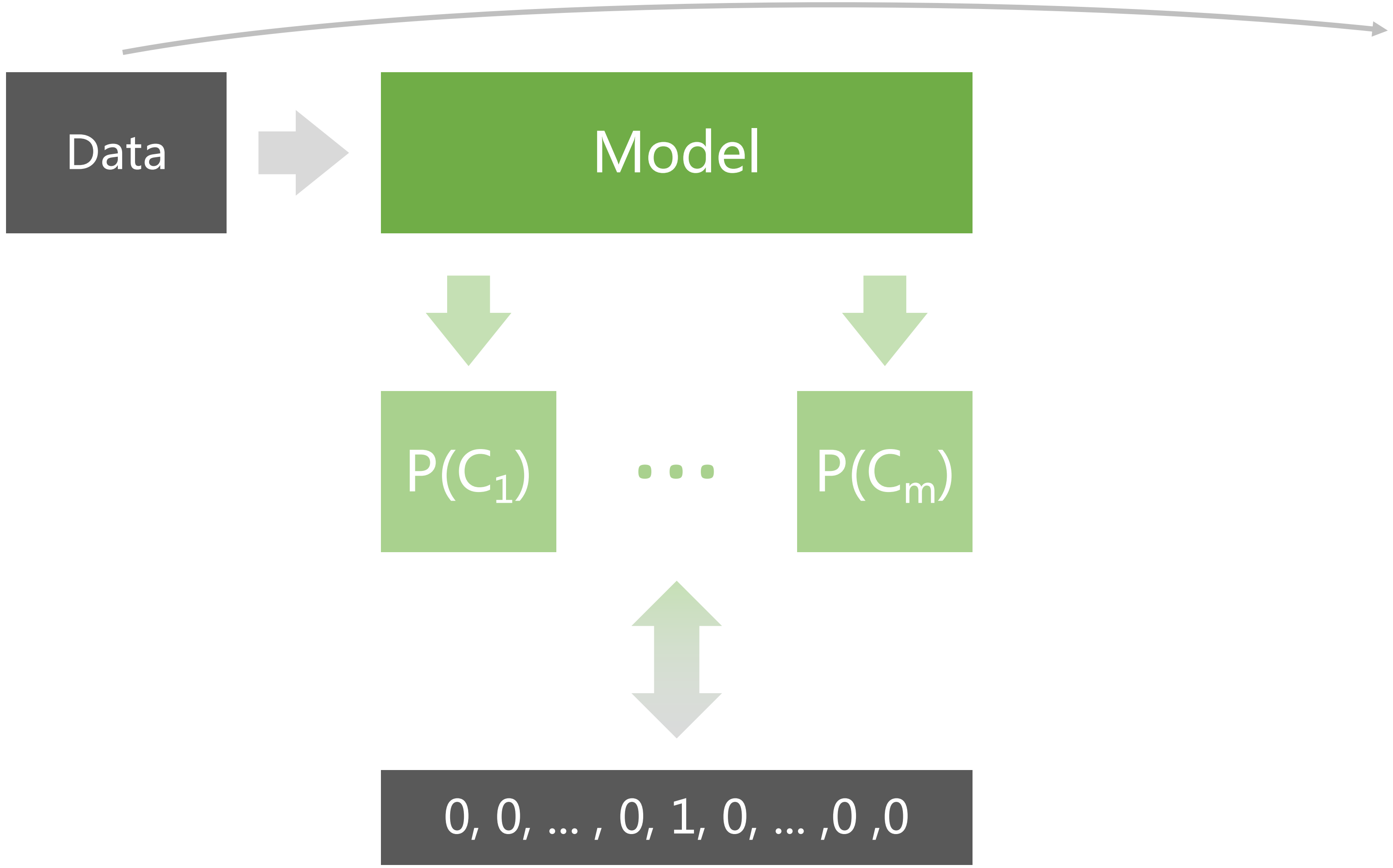
*Astar's Introduction to*

# Machine Learning

1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

# What is classification ?

$\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$



# What is a Linear Model ?

Data

Model

P(C$_1$) · · · P(C$_m$)

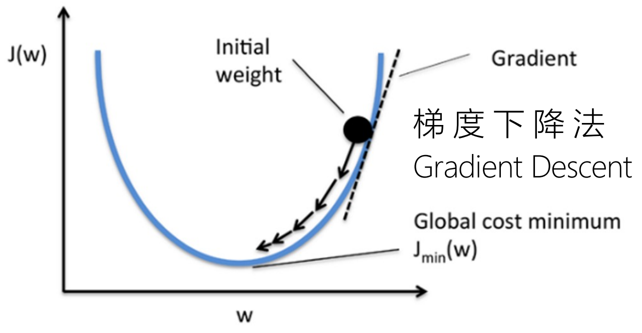
0, 0, … , 0, 1, 0, … ,0 ,0

$$\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$$

$$f(x) = \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{b} \in \mathbb{R}^m$$

$$\boldsymbol{w} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$$

$$\mathcal{L} = \sum (f(\boldsymbol{x}_i) - \boldsymbol{y}_i)^2$$

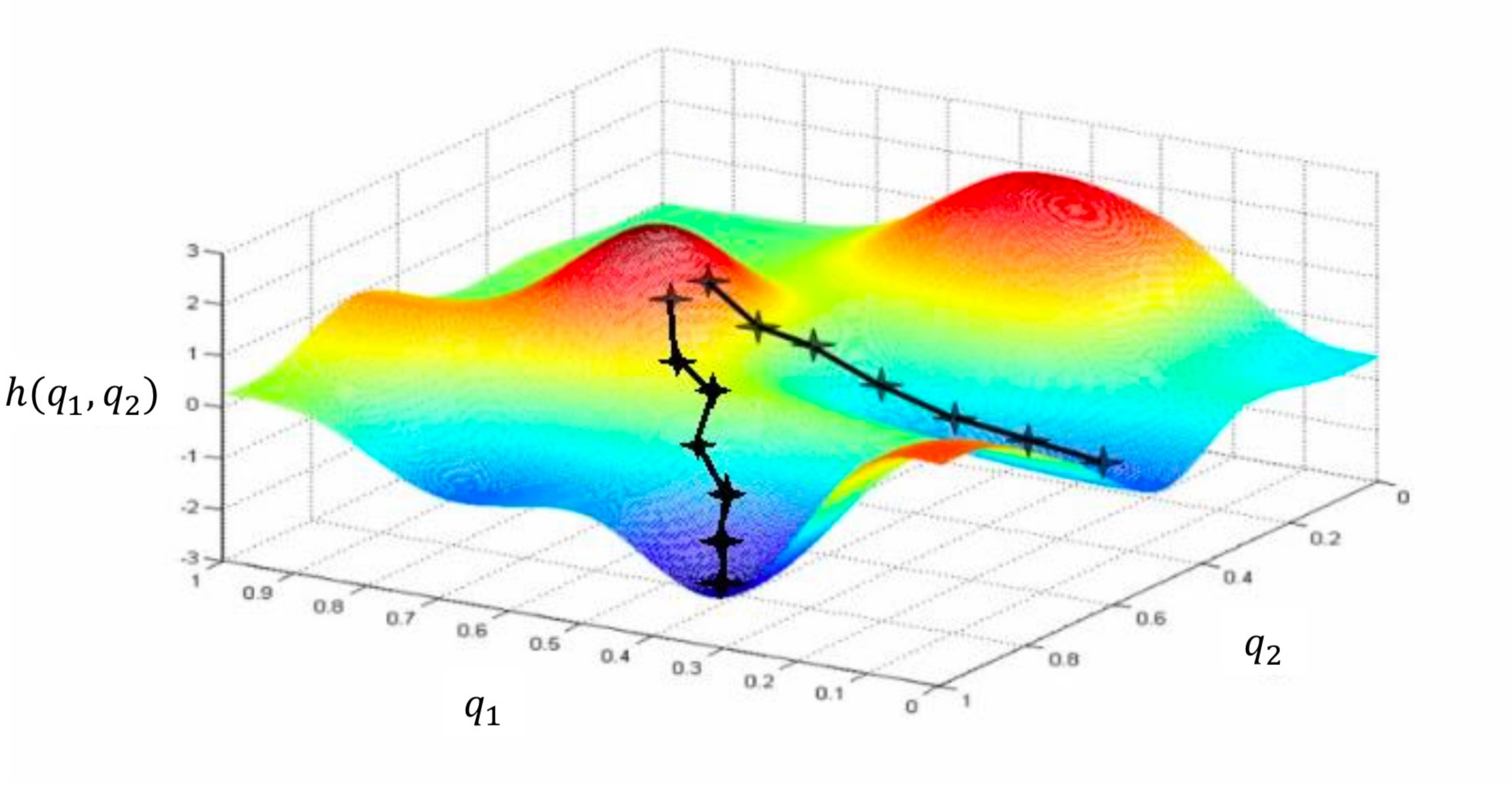



**举例！！**

- 西瓜分类
- EEG预测

# What is a Linear Model ?

Visualization of Gradient Descent:

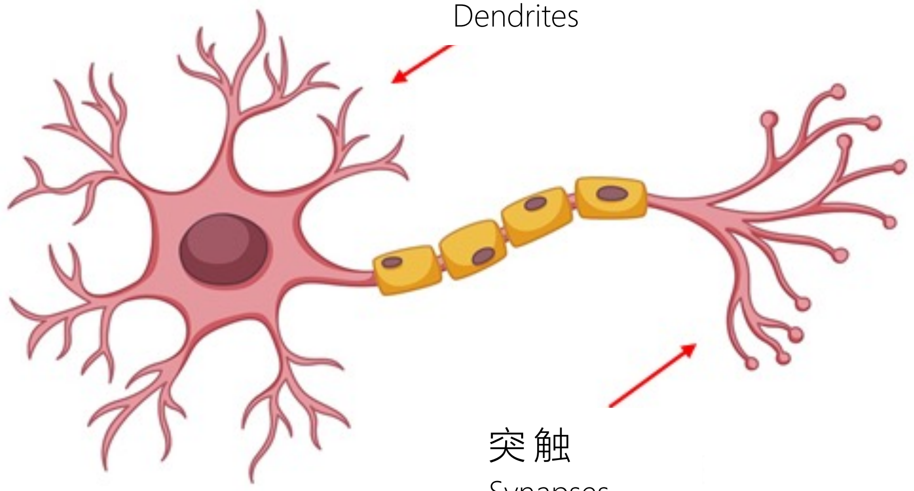


# Machine Learning

Neural Networks

# The simplest example : A perceptron

树突
Dendrites



突触
Synapses

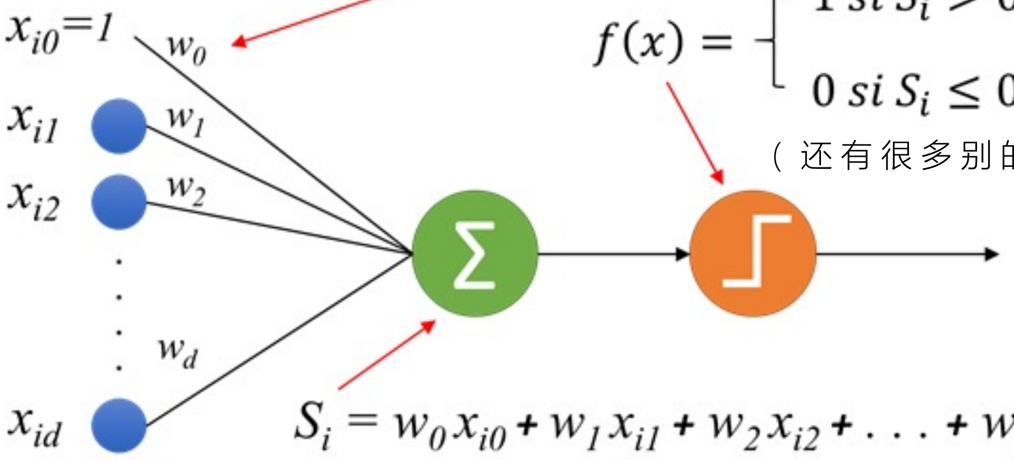
神经元
Neuron

模型参数/权重
Parameters / Weights

激活函数
Activation Function

$$f(x) = \begin{cases} 1 \ si \ S_i > 0 \\ 0 \ si \ S_i \leq 0 \end{cases}$$

（还有很多别的激活函数哦）



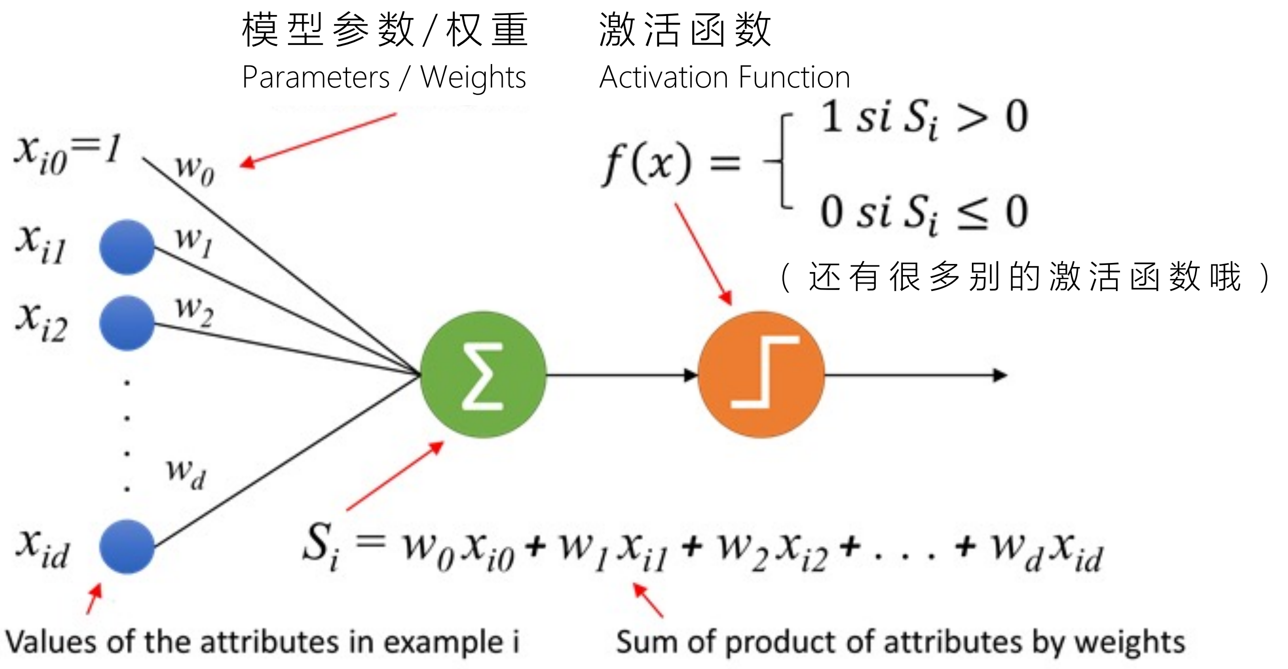

$$S_i = w_0 x_{i0} + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_d x_{id}$$

Values of the attributes in example i

Sum of product of attributes by weights

感知机
Perceptron

# The simplest example : A perceptron

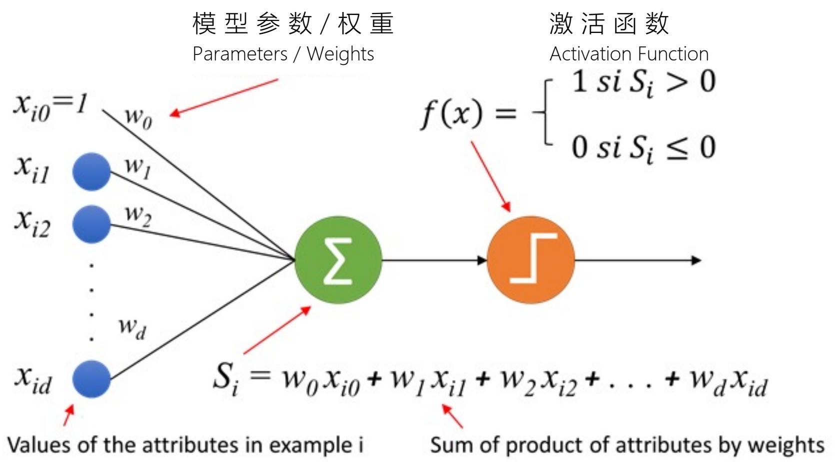



感知机
Perceptron

$$\boldsymbol{y} = f_{activation}\left(\sum \boldsymbol{w}^T \boldsymbol{x}\right)$$

$$f\left( \begin{bmatrix} w_0 & w_1 & \cdots & w_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \right) = y \in \mathbb{R}$$

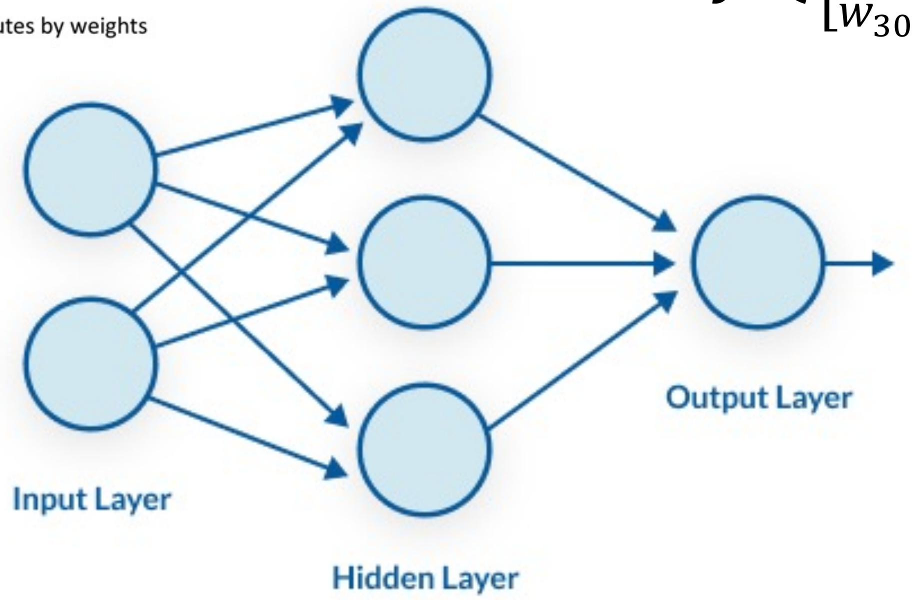
# More perceptrons !




感知机
Perceptron

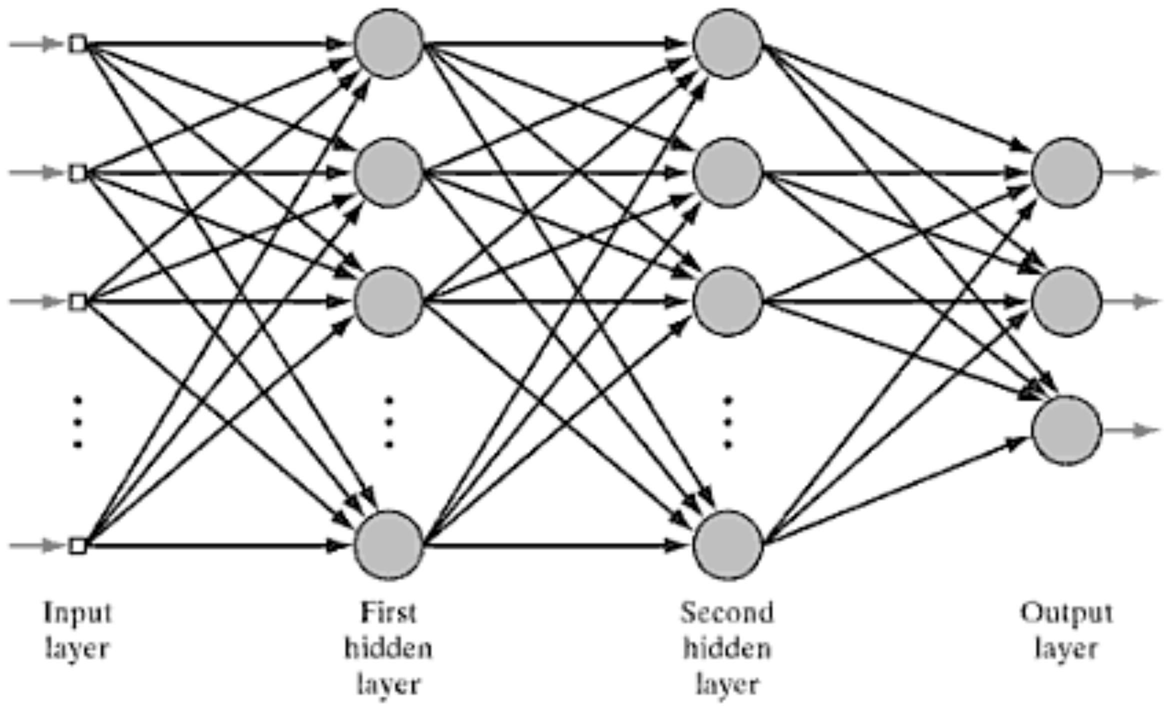
$$\boldsymbol{y} = f_{activation}\left(\sum \boldsymbol{w}^T \boldsymbol{x}\right)$$

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ w_{20} & w_{21} & \cdots & w_{2n} \\ w_{30} & w_{31} & \cdots & w_{3n} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$



Neural Networks

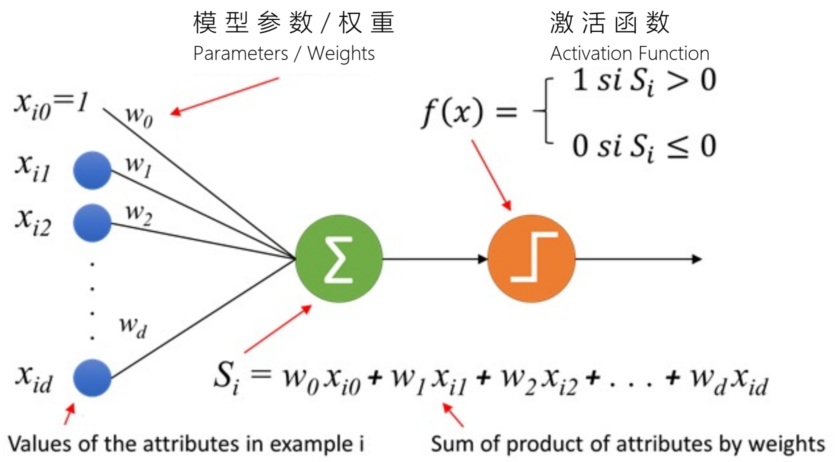
# Multilayer Perceptron \(≧▽≦)/




多层感知机 / 前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

**The same for every layer:**

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$




感知机
Perceptron

Neural Networks

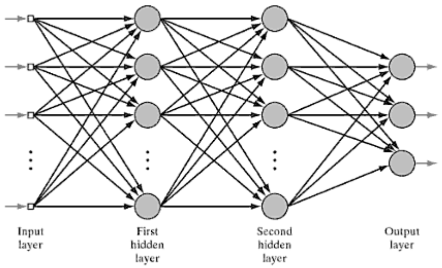
# Multilayer Perceptron for classification

Data → Model

$\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$

P(C$_1$) ... P(C$_m$)

0, 0, ... , 0, 1, 0, ... ,0 ,0

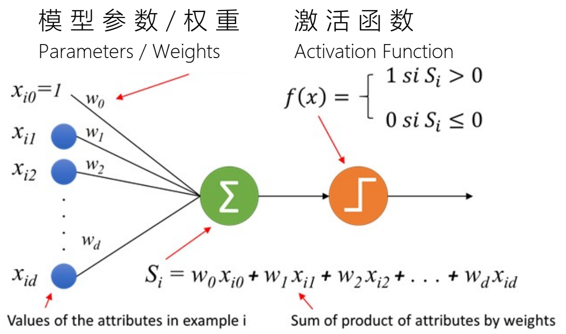



多层感知机 /
前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

$$\mathcal{L} = \sum (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2$$

**The same for every layer:**

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$




感知机 Perceptron

Neural Networks

# MLP vs. Linear Model

Data

$\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$

Model

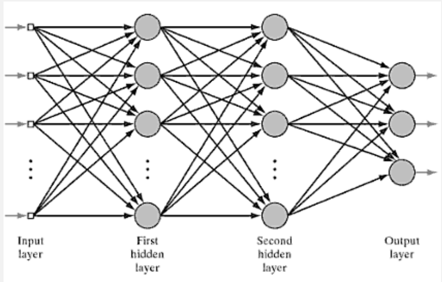
P($C_1$) ... P($C_m$)

0, 0, ... , 0, 1, 0, ... ,0 ,0

线性回归
Linear Regression

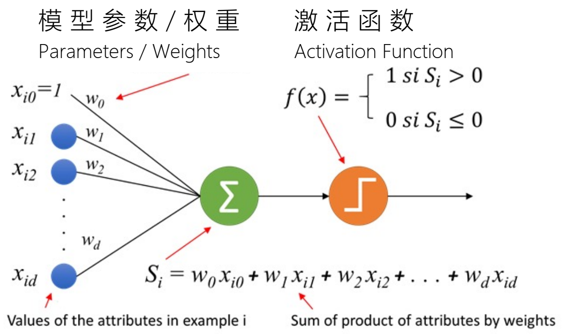
$$f(x) = \boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b} \in \mathbb{R}^m$$

$$\boldsymbol{w} \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m$$




多层感知机 /
前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x_0' \\ x_1' \\ \vdots \\ x_m' \end{bmatrix} \in \mathbb{R}^m$$




感知机 Perceptron

# Perceptron was invented in 1950s

Neural Networks

## MLP vs. Linear Regression

$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$

P($C_1$) ... P($C_n$)

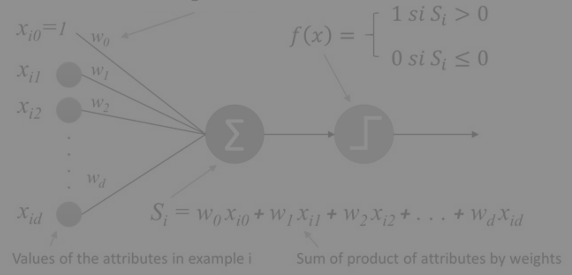
0, 0, ... , 0, 1, 0, ... , 0 ,0

线性回归
Linear Regression

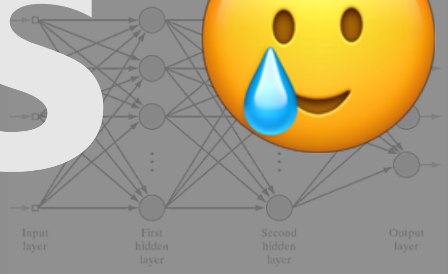
$\boldsymbol{w} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$

模型参数 / 权重
Parameters / Weights

激活函数
Activation Function



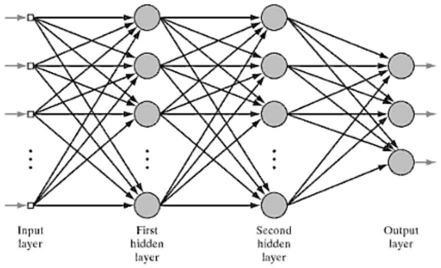

感知机 Perceptron




多层感知机 /
前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

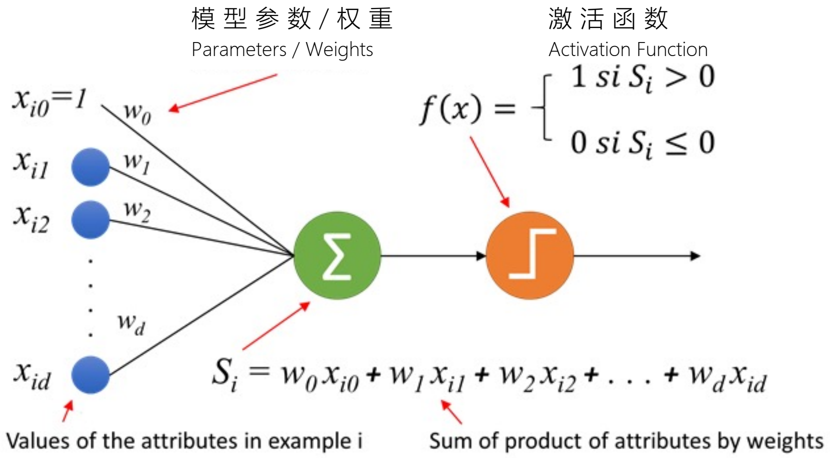
$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$

# Understanding MLP




多层感知机 /
前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$



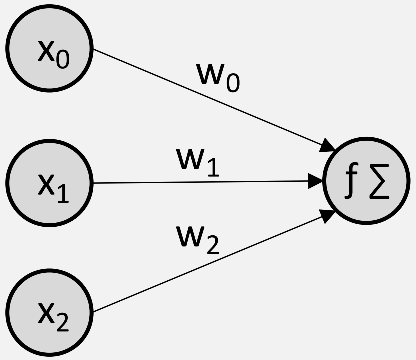

感知机 Perceptron

## 如何理解MLP？

1. 一个简单的感知机能完成一些逻辑运算

### Task 1

Find $w_0$, $w_1$, $w_2$ for AND, OR, and NOT (for $x_1$).
Note that the bias $x_0$=1.

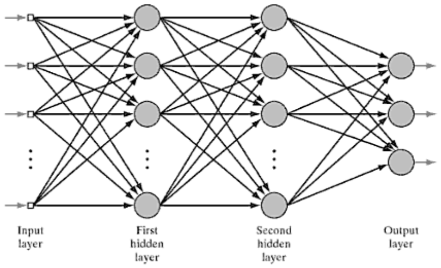



Answer:

| | |
|---|---|
| AND | [-1,1,1] |
| OR | [-0.5,1,1] |
| NOT | [0.5,-0.6,0] |

Neural Networks

# Understanding MLP

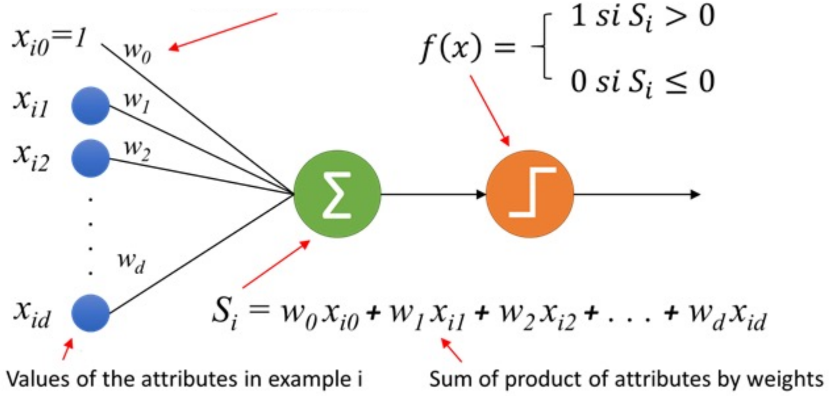



多层感知机 /
前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$

模型参数 / 权重
Parameters / Weights

激活函数
Activation Function



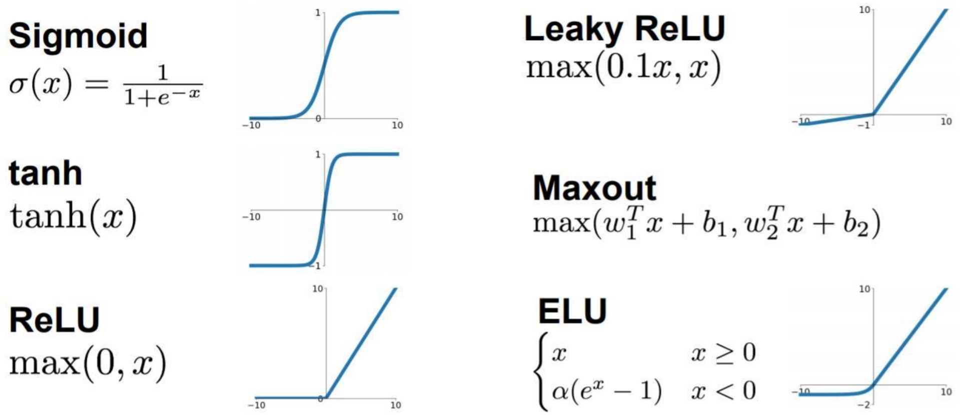

感知机 Perceptron

## 如何理解MLP？

1. 一个简单的感知机能完成一些逻辑运算
2. 线性变换 + 非线性变换去逼近输入到输出

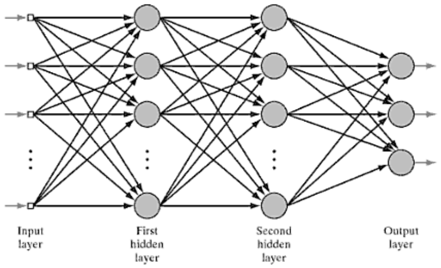
## Task 2

1. Go to *https://playground.tensorflow.org*
2. Set Dataset=2, LR=0.3, Activation=Linear
3. Set Dataset=2, LR=0.3, Activation=Sigmoid
4. Play around and see who can gets highest accuracy on dataset 4!



Neural Networks

# Understanding MLP

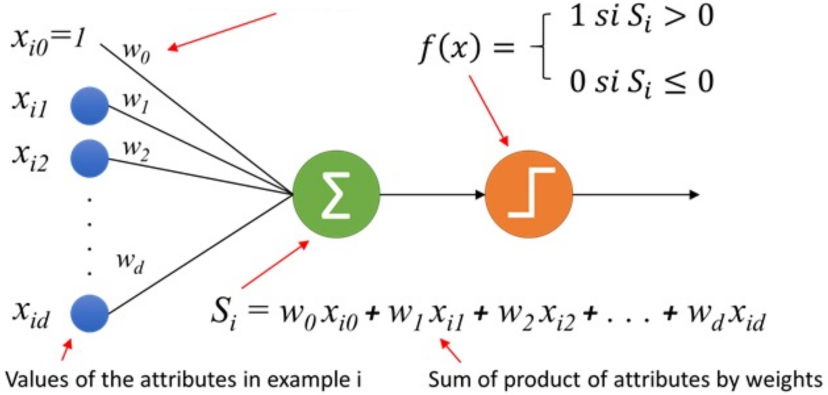



多层感知机 /
前馈神经网络
Multilayer Perceptron (MLP)
Feed Forward Network (FFN)

$$f\left(\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \cdots & w_{in} \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x'_0 \\ x'_1 \\ \vdots \\ x'_m \end{bmatrix} \in \mathbb{R}^m$$

模型参数/权重
Parameters / Weights

激活函数
Activation Function

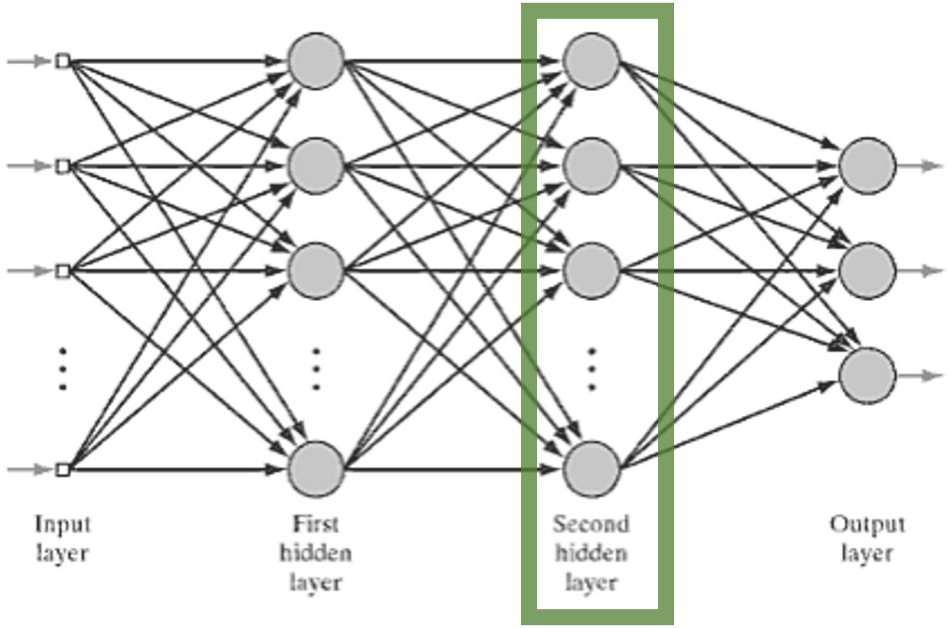



感知机 Perceptron

**如何理解MLP？**

1. 一个简单的感知机能完成一些逻辑运算
2. 线性变换 + 非线性变换去逼近输入到输出
3. 隐层形成的表征向量（表示学习）

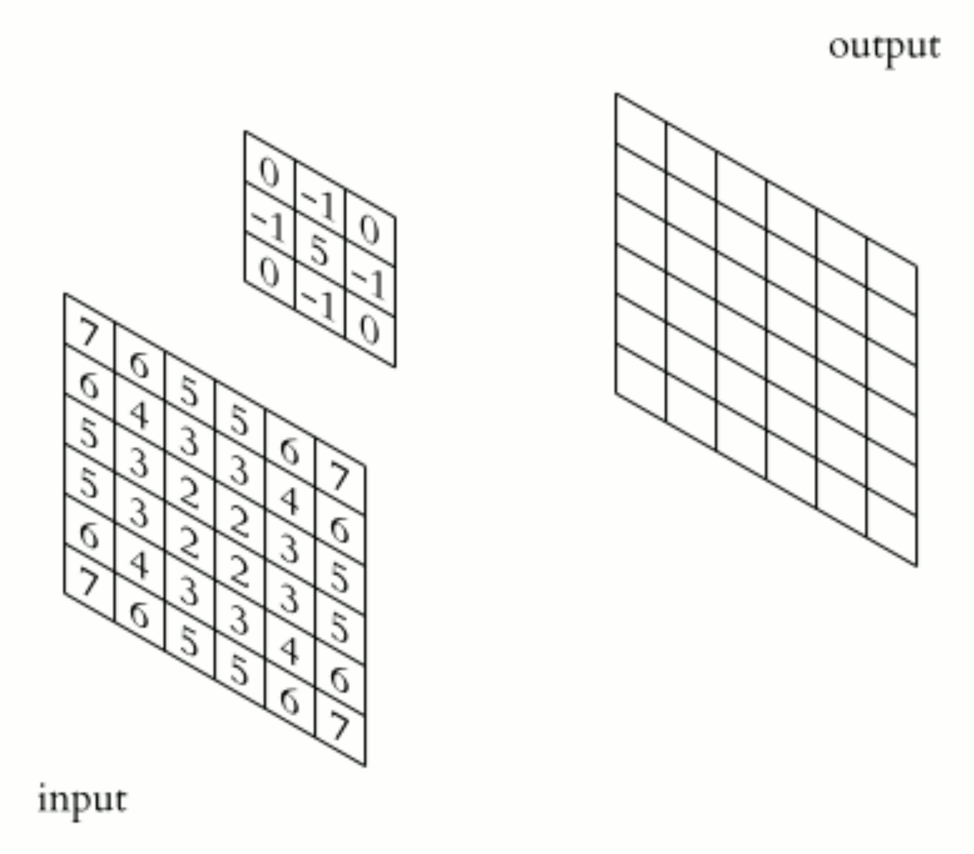
人工特征工程 → 表示学习学习特征

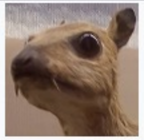
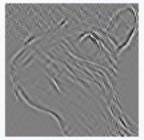
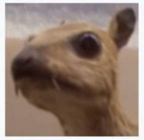
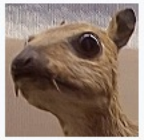


# Machine Learning

# What is convolution?

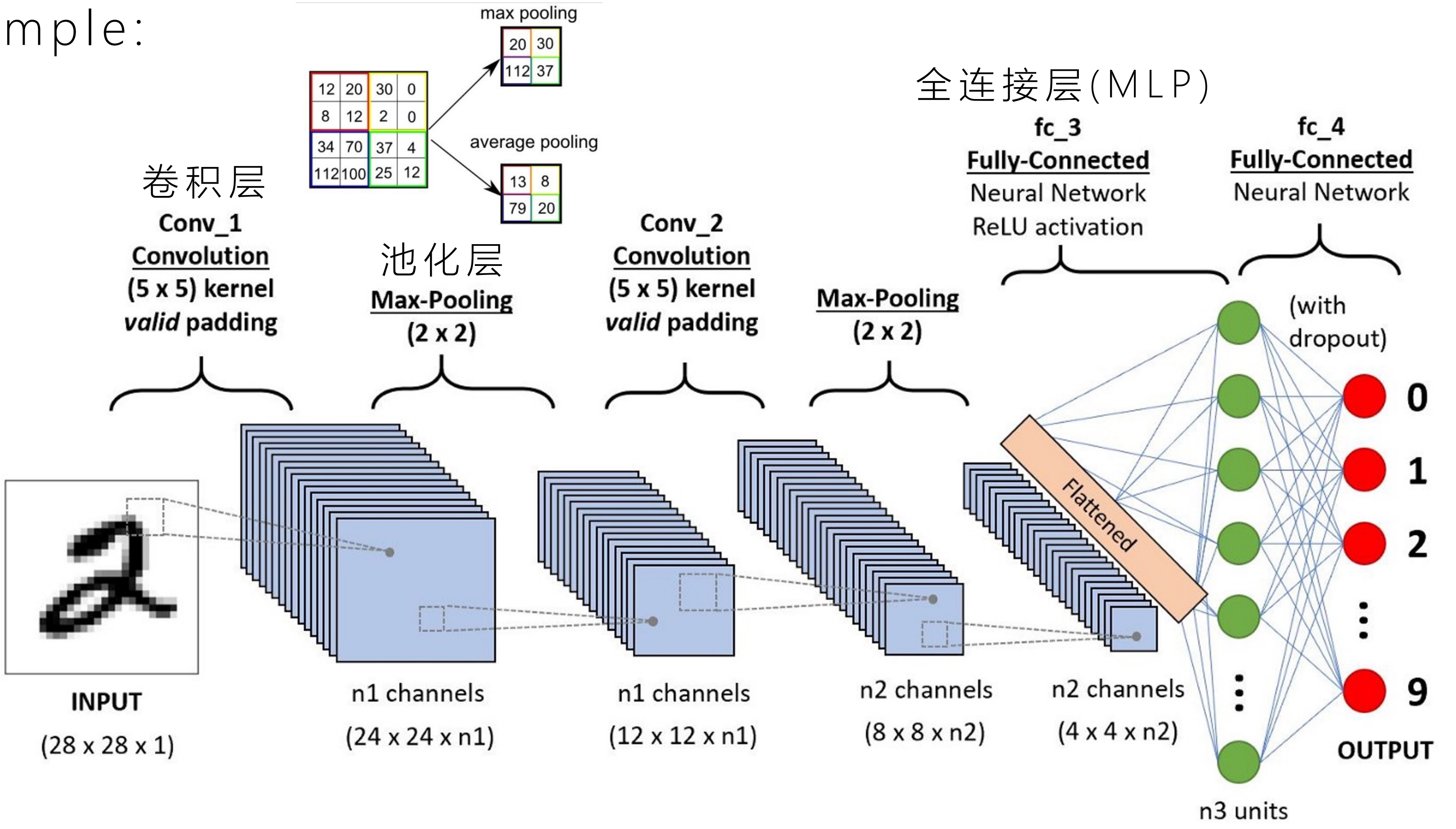
a convolution matrix is also called a filter matrix




| Operation | Kernel ω | Image result g(x,y) |
|---|---|---|
| Identity | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |  |
| Ridge detection | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ |  |
| | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ |  |
| Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |  |
| Box blur (normalized) | $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |  |
| Gaussian blur 3 × 3 (approximation) | $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |  |
| Gaussian blur 5 × 5 (approximation) | $\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$ |  |
| Unsharp masking 5 × 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask) | $\frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$ |  |

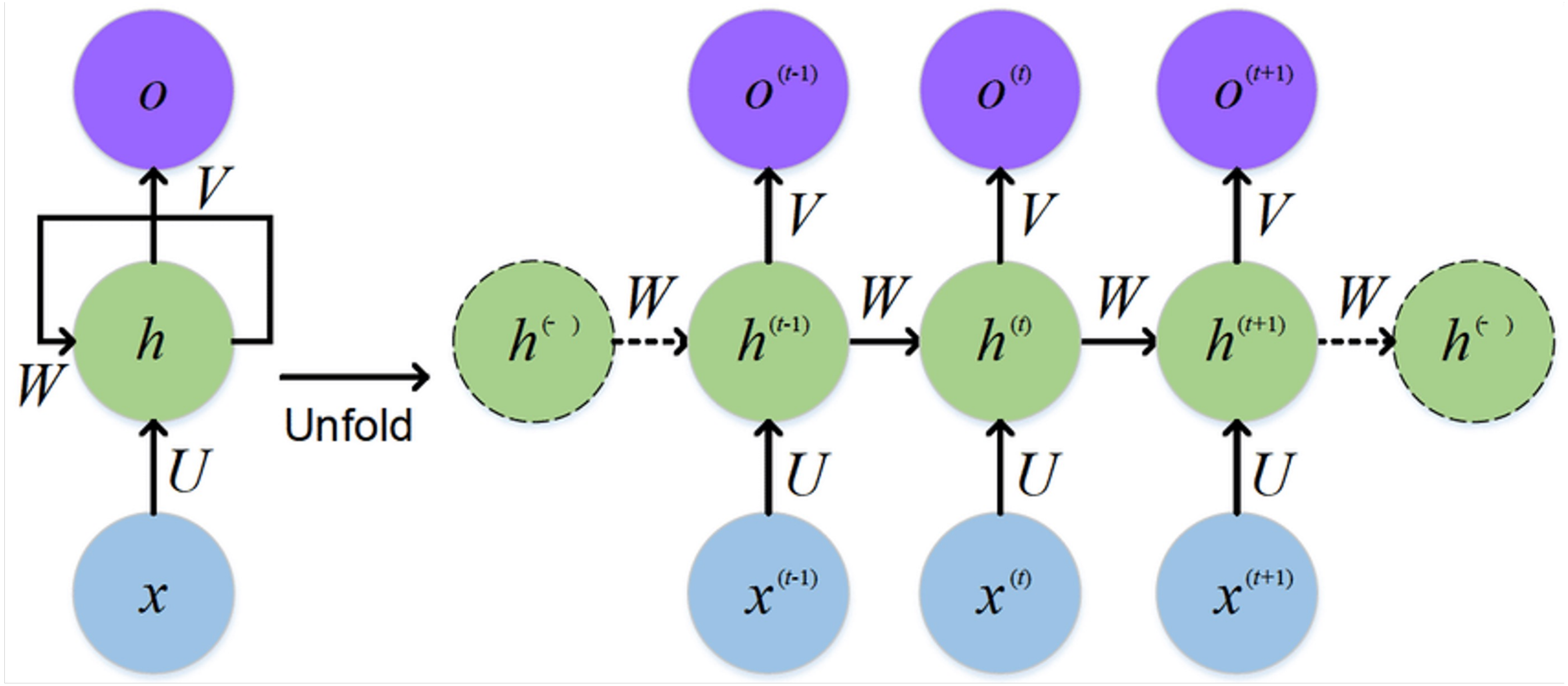
# A convolutional neural network

Example:



# Machine Learning

# Recurrent Neural Network



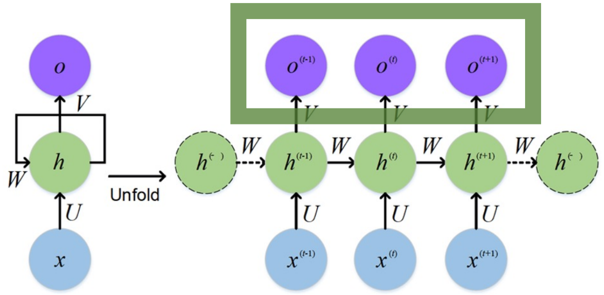
Astar's Introduction to

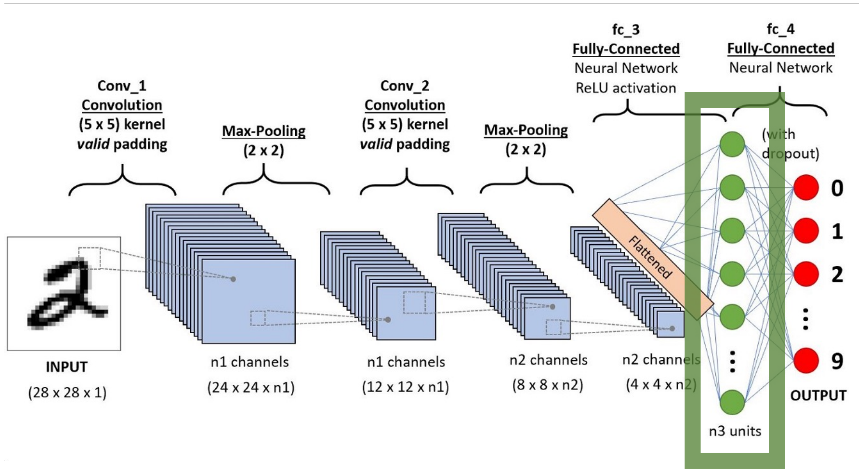
# Machine Learning

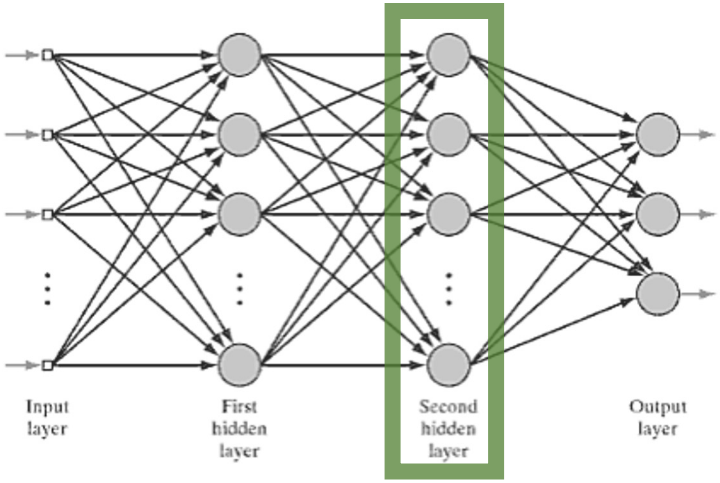
1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

# Representation Learning

1. What are representations?



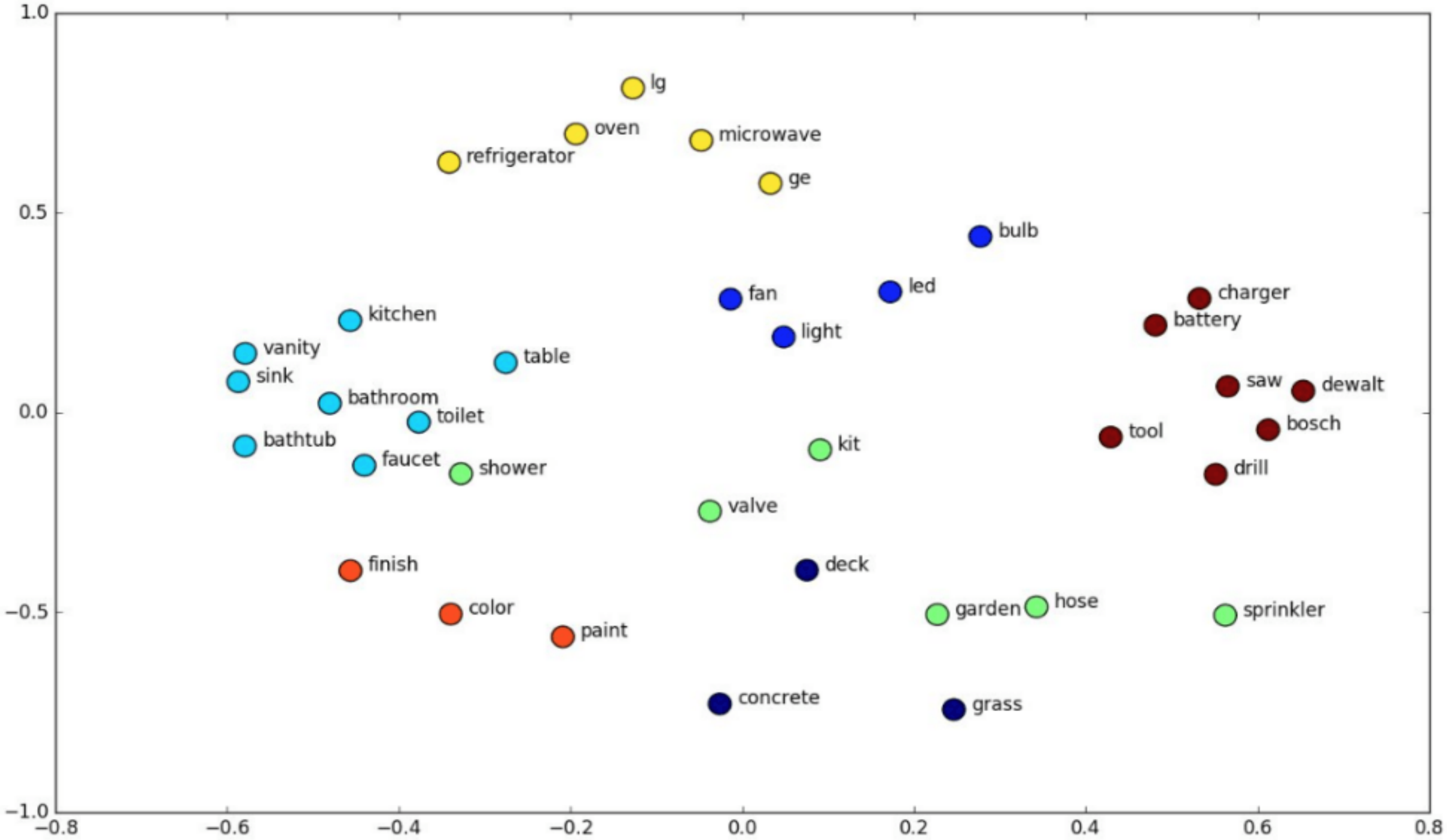



# Representation Learning

1. What are representations?
2. What is a latent space?

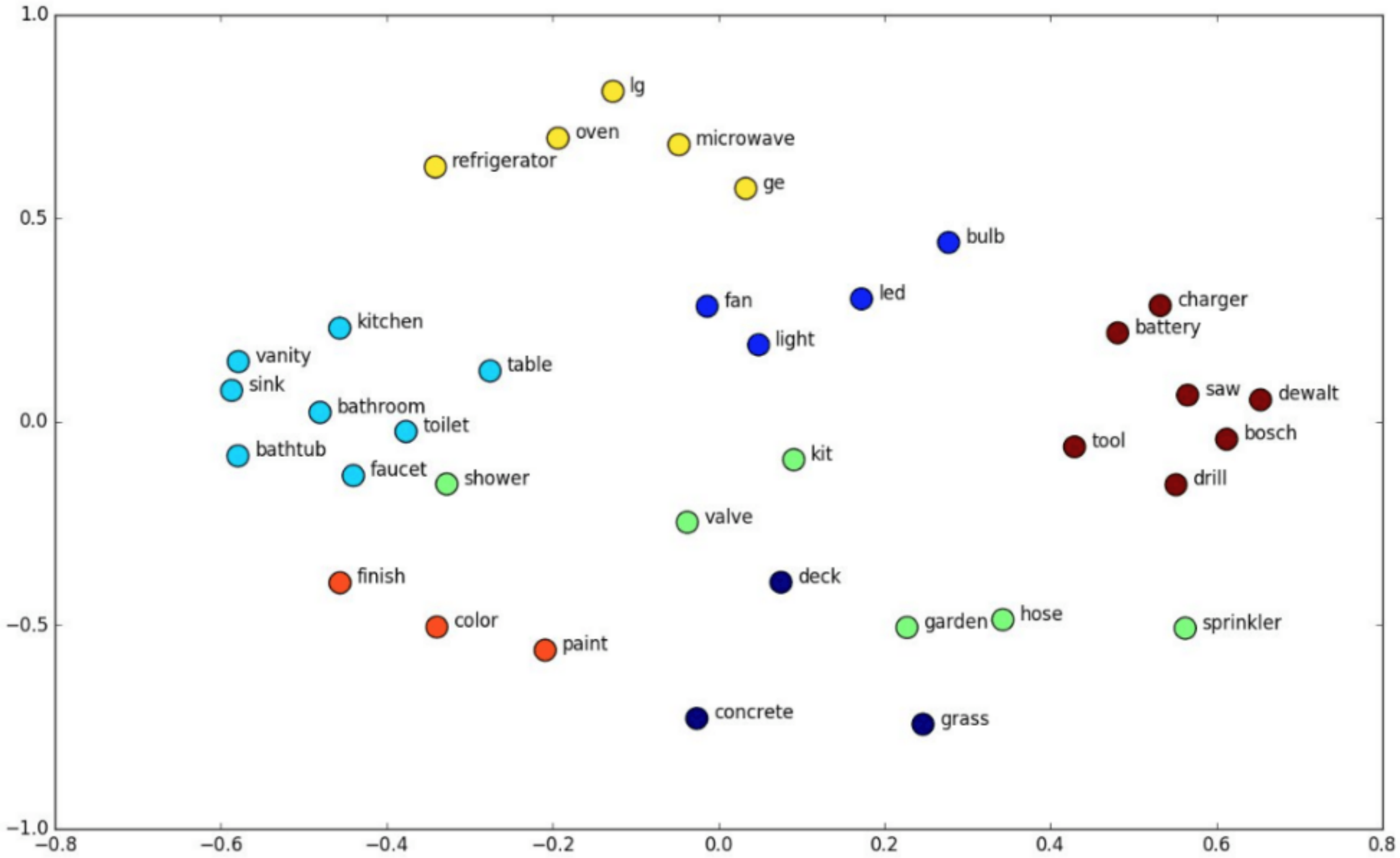
# Representation Learning

1. What are representations?
2. What is a latent space?



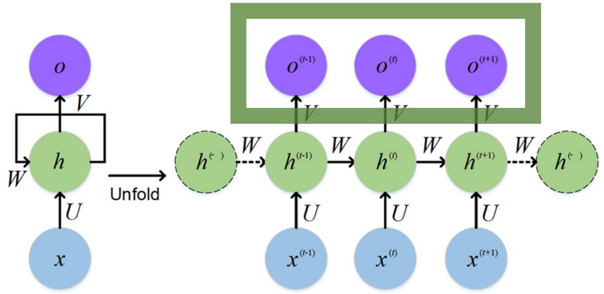
# Representation Learning

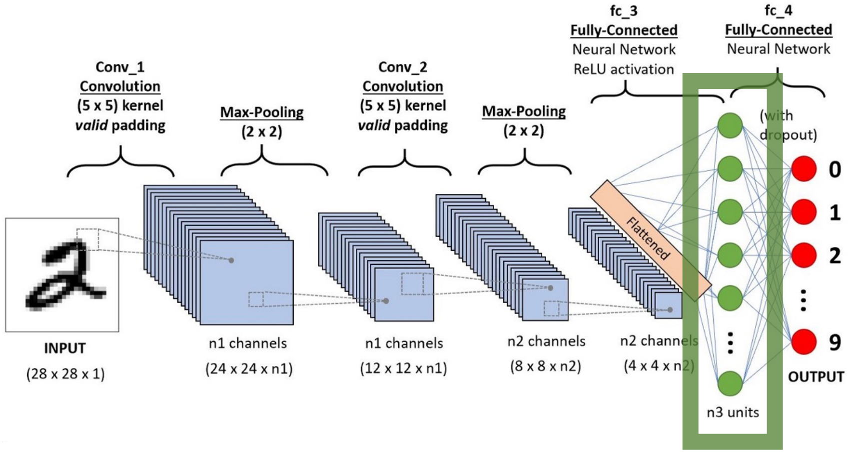
1. What are representations?
2. What is a latent space?
3. Relationship among data samples in the latent space?

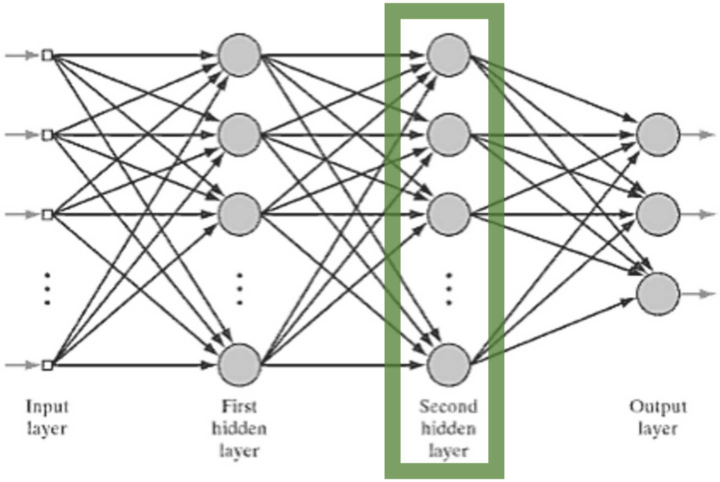


# Representation Learning

1. What are representations?
2. What is a latent space?
3. Relationship among data samples in the latent space?
4. What are supervision signals for?
5. Using different models on a same dataset?
6. How to get robust and informative representations?



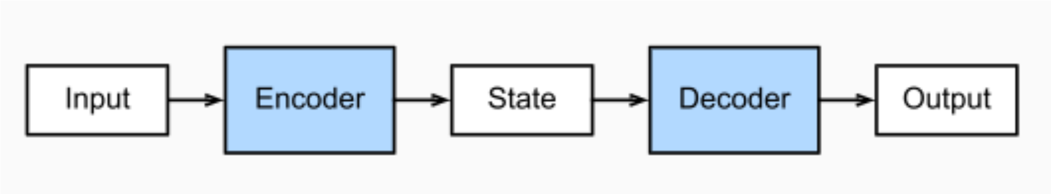



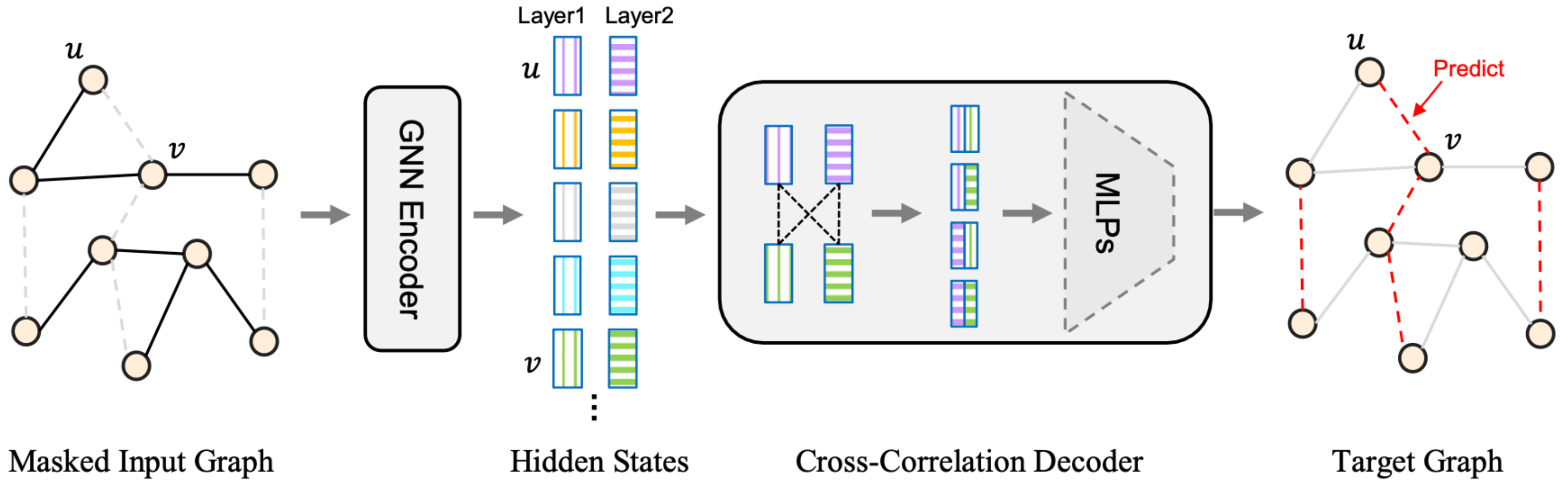
# Machine Learning

1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

# The Encoder-Decoder Architecture




Encoder-Decoder Architecture




Masked Autoencoder (an example on Graph Neural Networks)

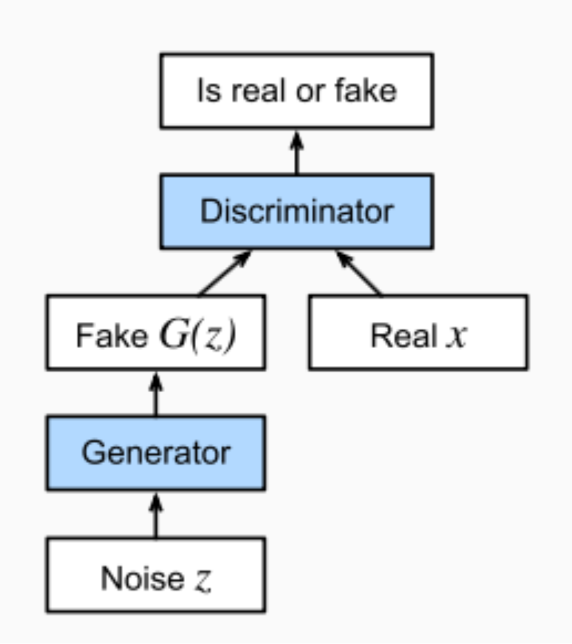
Astar's Introduction to

# Machine Learning

1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

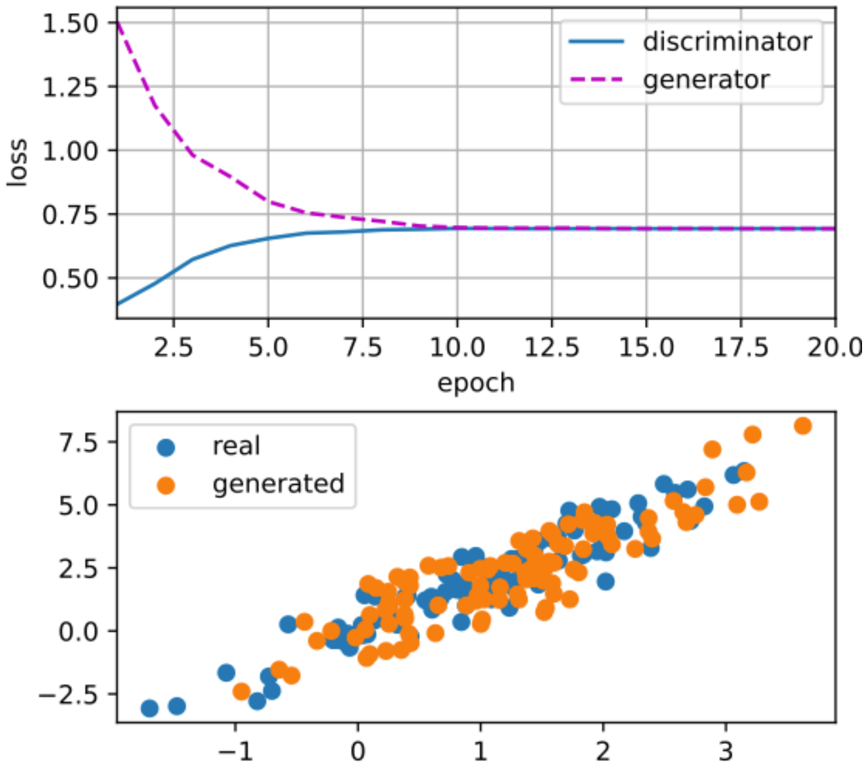
Neural Networks

# Generative Adversarial Network




生成式对抗网络

Generative Adversarial Network (GAN)




训练过程例

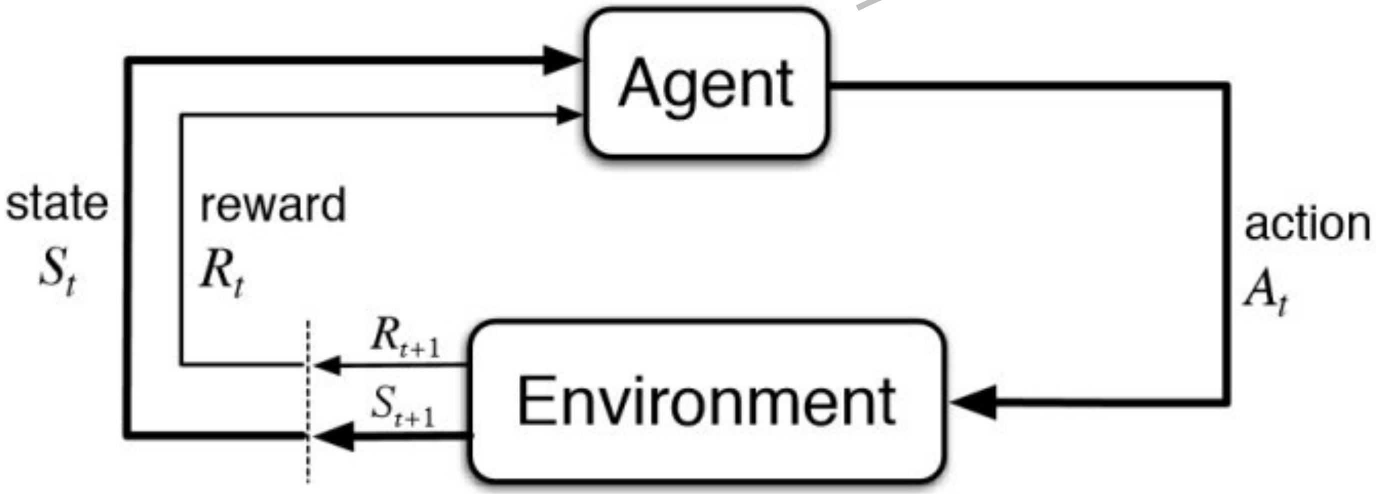
Astar's Introduction to

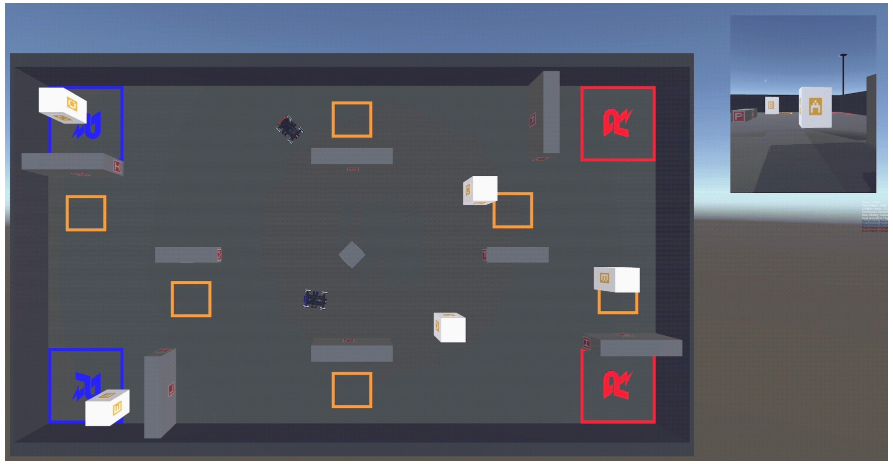
# Machine Learning

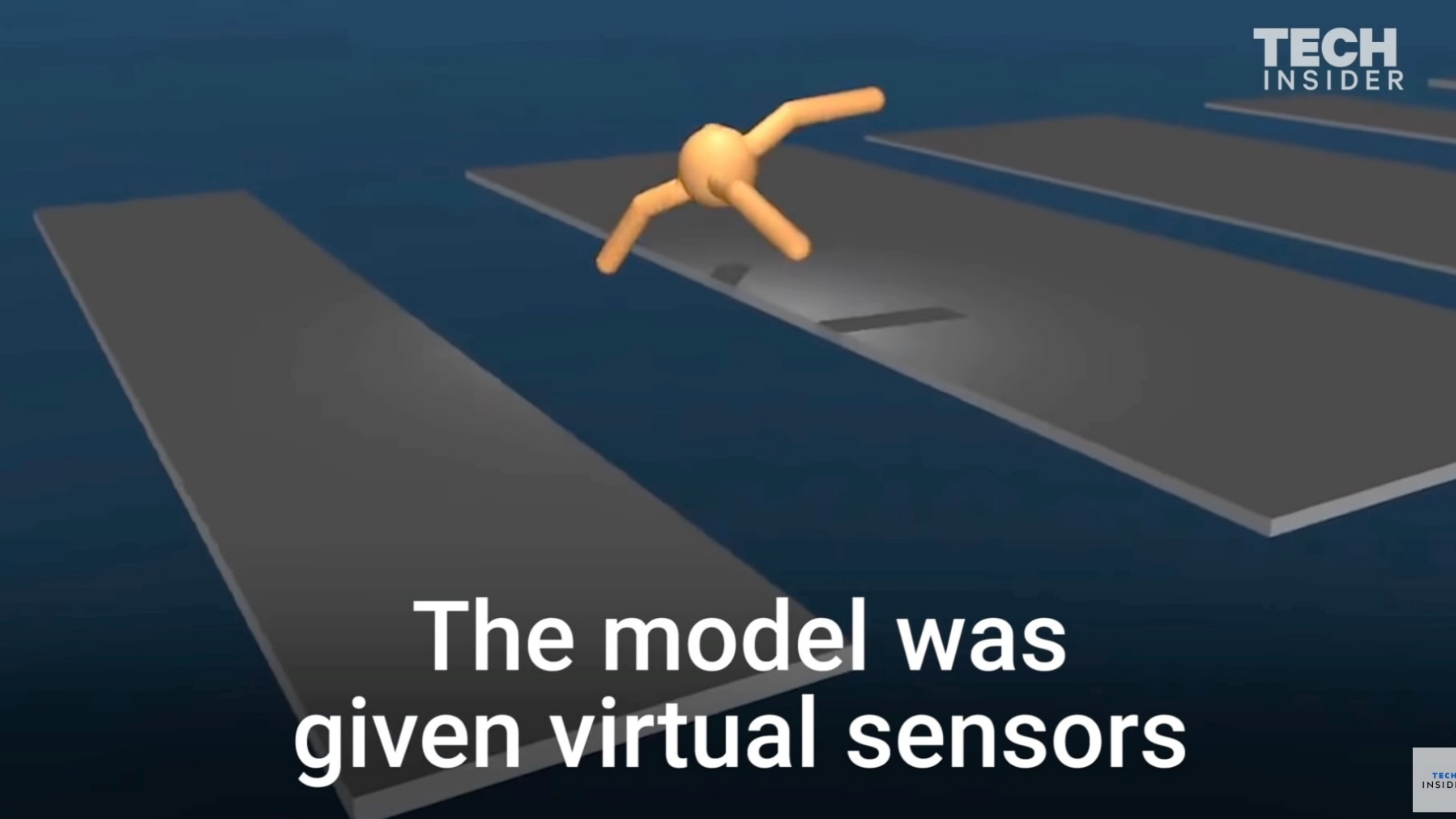
1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

# Reinforcement Learning

Can be a deep learning model




TECH INSIDER
The model was given virtual sensors

Astar's Introduction to

# Machine Learning

1. Artificial Intelligence and Machine Learning
2. One of the General Machine Learning Paradigms: Supervised Learning
3. Solving a Classification Task with the Linear Model
4. Neural Networks
    a. Multilayer Perceptron (MLP)
    b. Convolutional Neural Networks (CNN)
    c. Recurrent Neural Networks (RNN)
5. Some Ideas and Fancy Models
    a. Representation Learning
    b. The Encoder-Decoder Architecture and Autoencoder
    c. Generative Adversarial Network (GAN)
    d. Reinforcement Learning
6. Research Areas
7. Limitations
8. Resources

Astar's Introduction to

# Some Research Areas *

Recommender Systems
Computer Vision
Natural Language Processing
Medicine
Robotics
Internet of Things
Data Mining
...

Theoretical Analysis
Self-Supervised Learning
Reinforcement Learning
Attention Mechanism
Contrastive Learning
Diffusion Models
Transformers
...

* Due to limited knowledge, the lists might be incomplete and important areas of research might be missing

Astar's Introduction to

# Limitations of Deep Learning

- Explainability: When reliability is vital, DL methods might not be safe!
- Data Demanding: Training a DL model might require tremendous amount of data, which is usually hard to obtain.
- Computational Resources Demanding: Very often, larger model make things easier, but them might take months to train.

Astar’s Introduction to

# Resources for you

- 机器学习书籍：《机器学习》周志华（人称西瓜书）
- 机器学习课程：吴恩达机器学习 bilibili
- 深度学习课程：跟李沐学AI bilibili
- 常用代码框架：PyTorch (推荐)，TensorFlow 等
- 计算资源获取：可以使用社团的 GPU （仅有12G）或者 CS GPU Farm（CS专业可以用到两张 3090 卡）或者 Google Colab

End of
Astar's
Rough Introduction to
# Machine Learning

HKU Astar · 2023 · 10 · 8 · HelloElwin & Nebula