Deep Learning

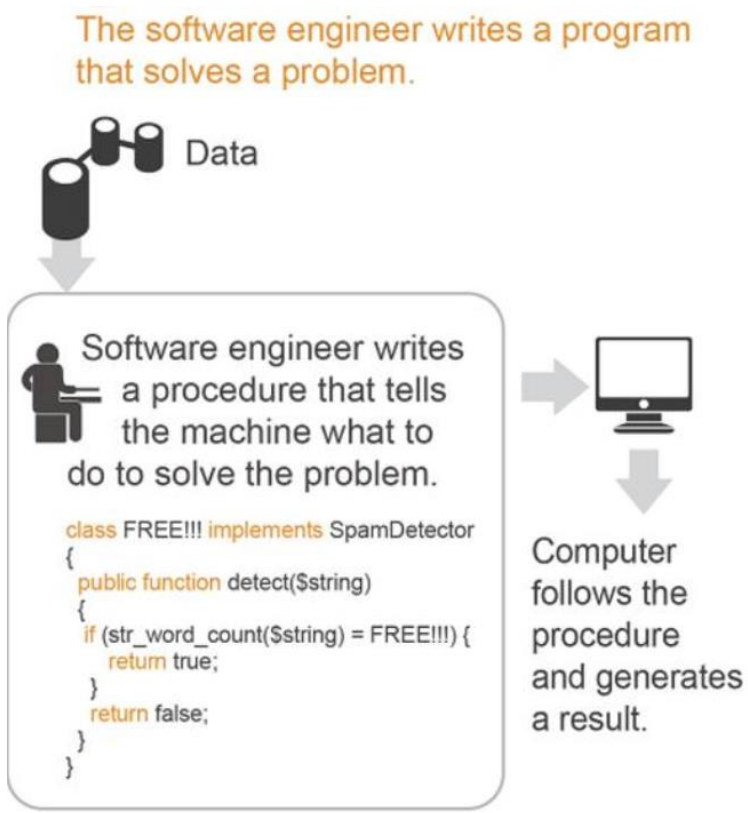
Presented by Hanchen Zhou

00 Contents

01 Introduction to Machine Learning

Machine learning is a field of computer science that gives computer systems the ability to **"learn"** with data, without being explicitly programmed.

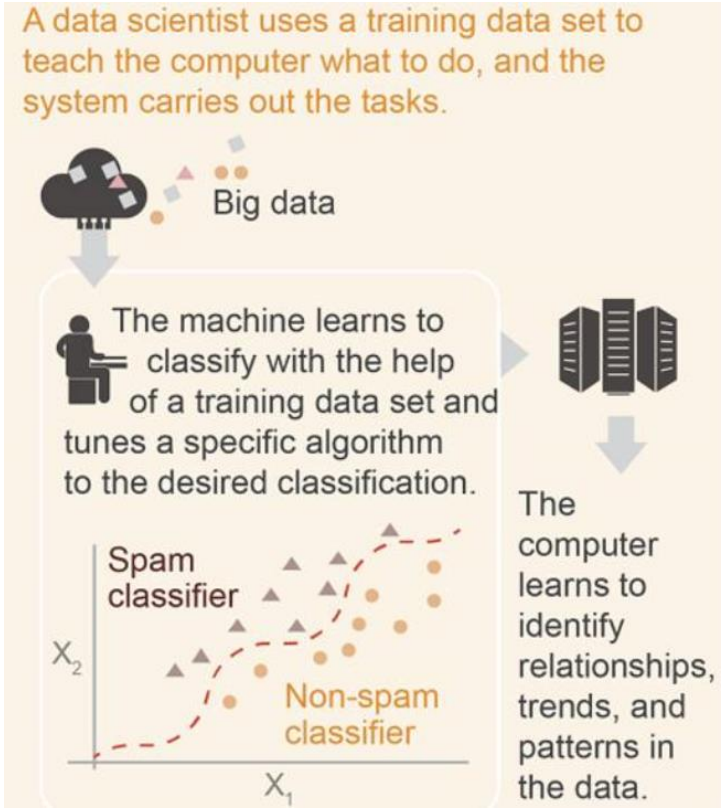
1.1 What is Machine Learning



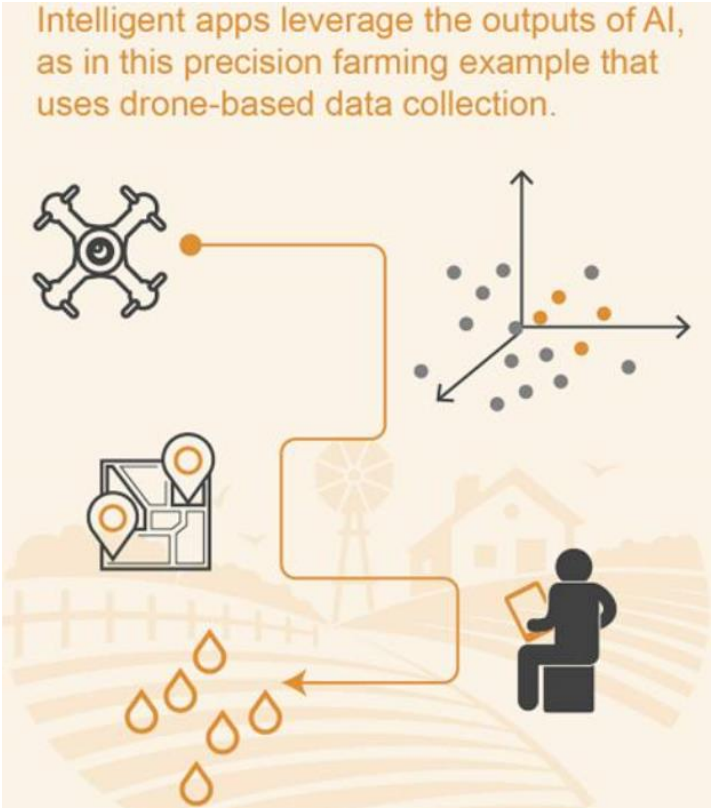
Traditional Programming

Statistics

1.1 What is Machine Learning

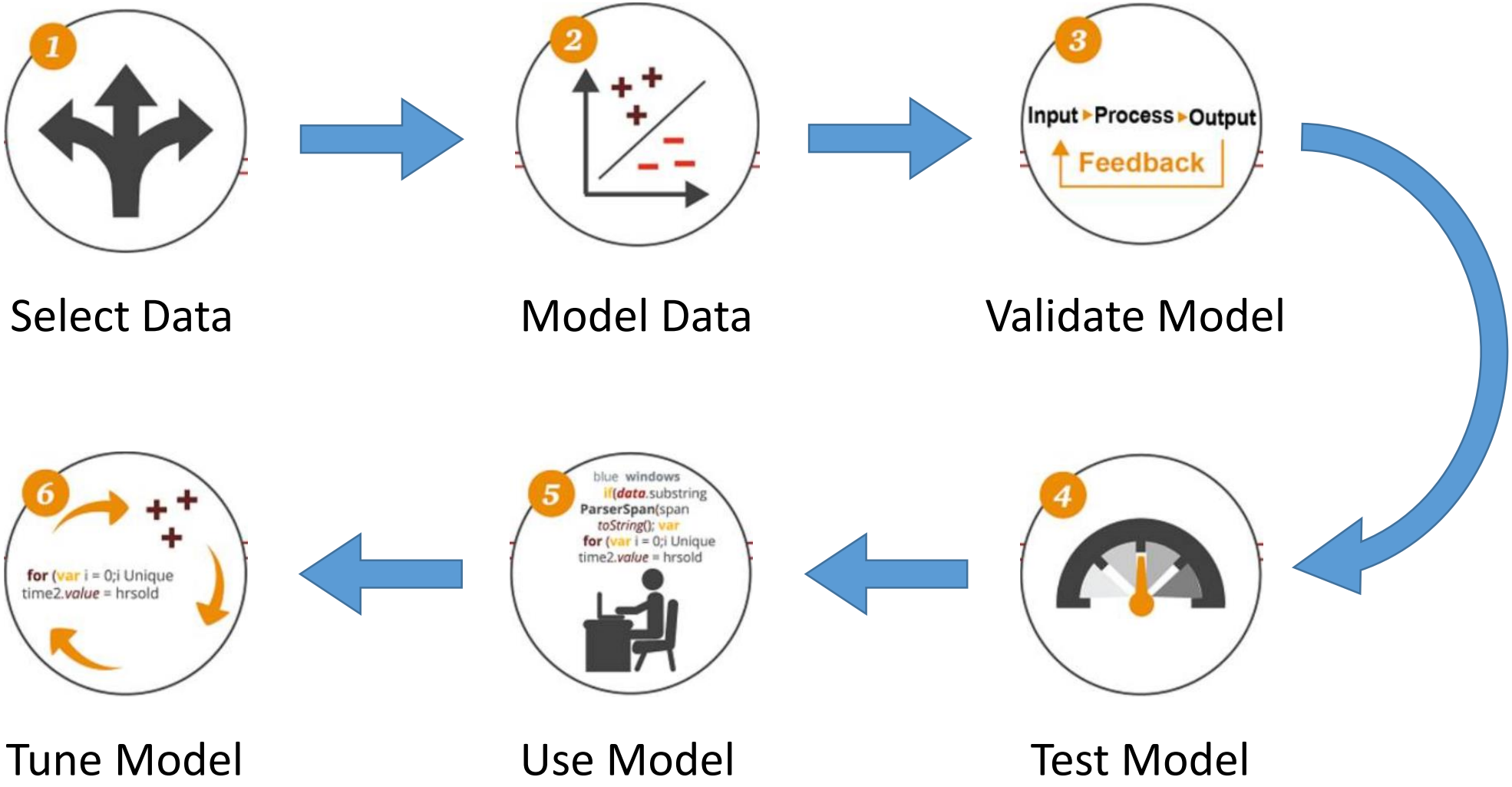


Machine Learning



Intelligent Apps

1.2 How Machine Learning Works



1.3 Formal Definition

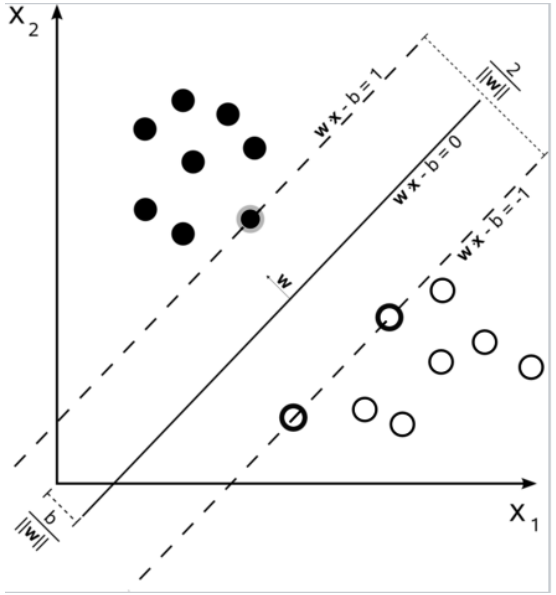
- some data: $x \rightarrow y$
- using data to build a model: m
- other data: $x' \rightarrow y'$
- using model to achieve:

$$m(x') = \tilde{y} \rightarrow y'$$

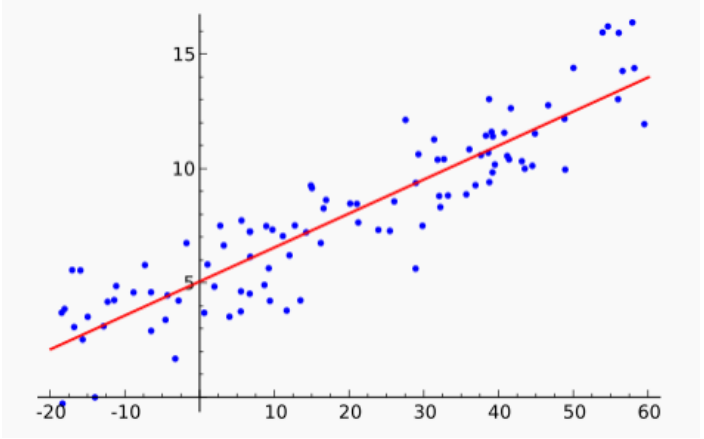
1.4 Classification

ML

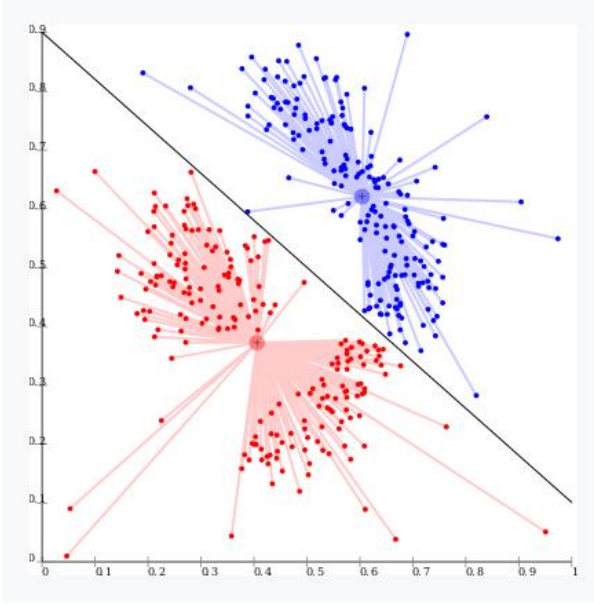
- Supervised learning → Classification
- Supervised learning → Regression
- Unsupervised learning → Clustering



Classification

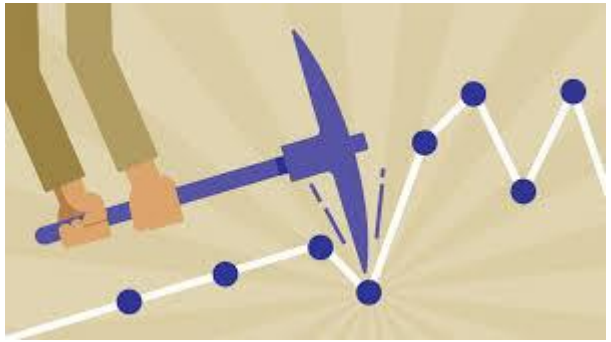


Regression



Clustering

1.5 Application



Data Mining

Google
YAHOO!
bing

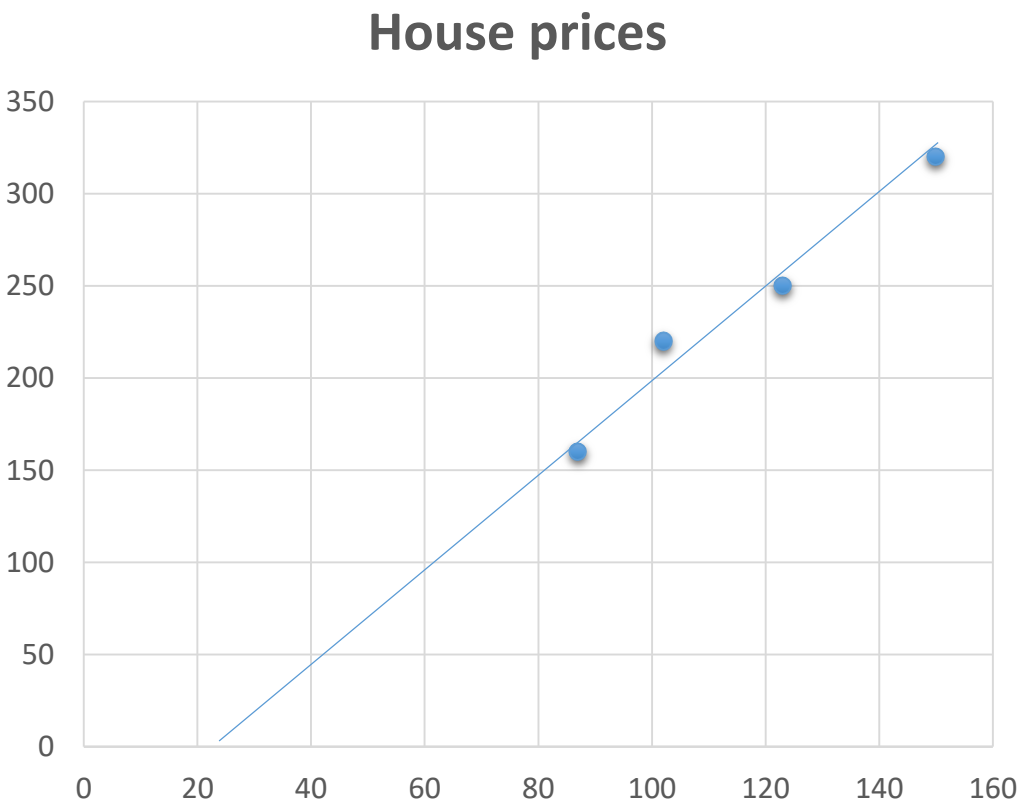
Search Engines

Handwriting Recognition

...

02 Linear Model

Some data about house prices

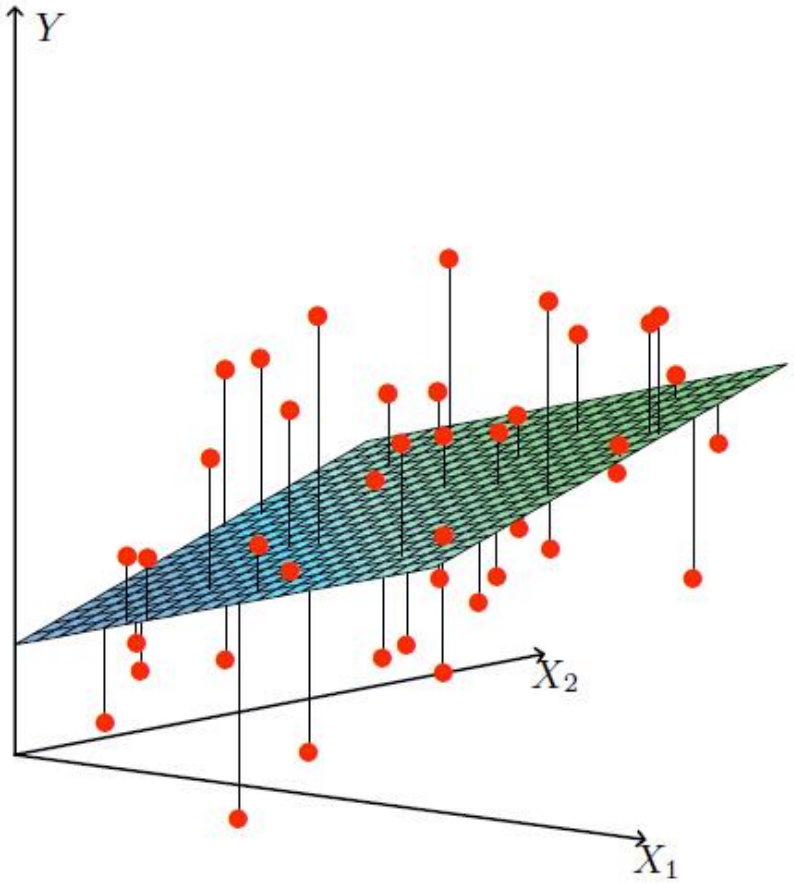


Area(m^2)	Prices
123	250
150	320
87	160
102	220

$$y = wx + b$$

02 Linear Model

With more dimensions of data



$$x = (x_1, x_2, \ldots, x_n)^T$$

$$y = w_1 x_1 + w_1 x_1 + \cdots + w_1 x_1 + b$$

$$y = \boldsymbol{w}^T \boldsymbol{x} + b$$

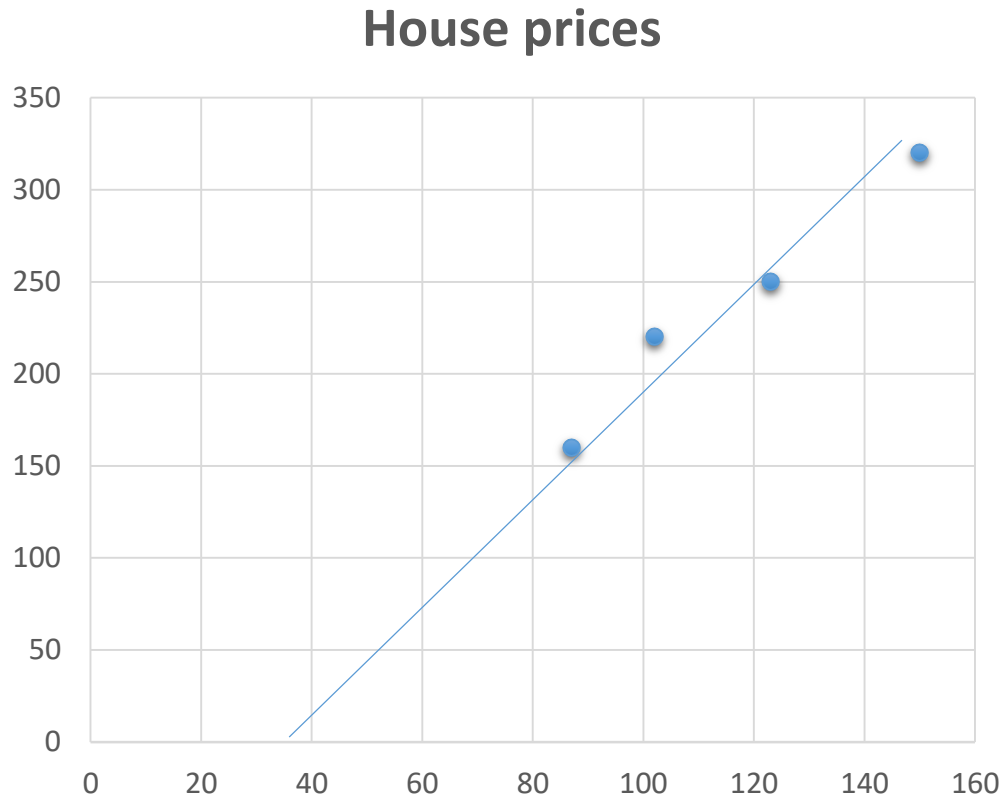
2.1 Linear Regression

Back to the house prices problem

$$f(x) = wx + b$$

Loss function is minimized

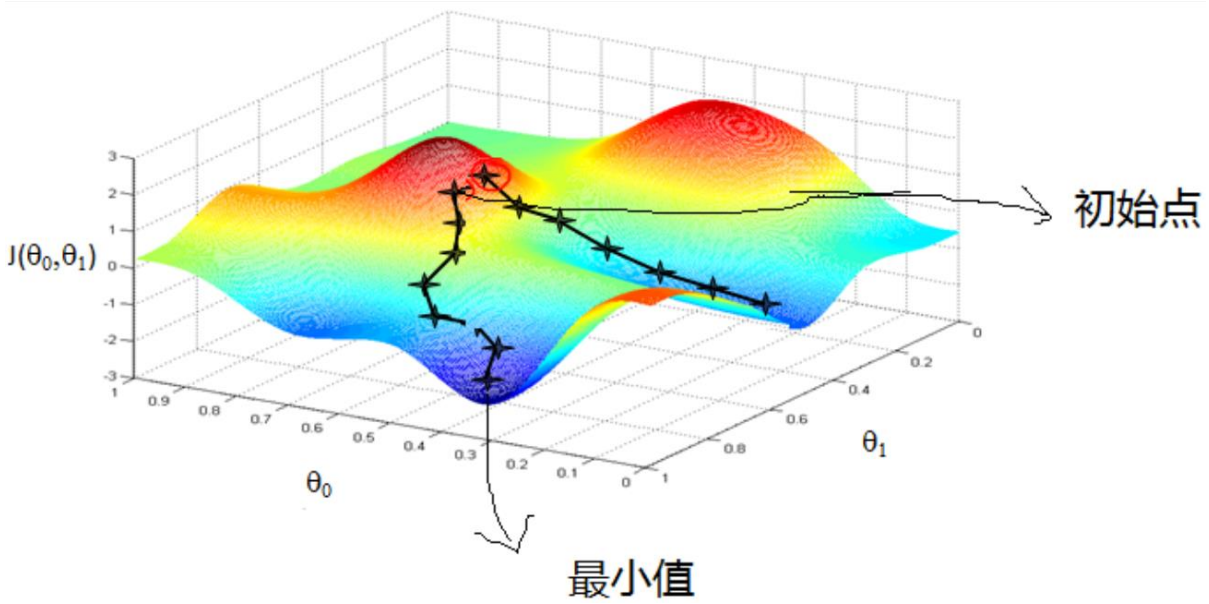
$$L(w,b) = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2$$



? How to make Loss function to a minimum

2.1 Linear Regression

Gradient Descent



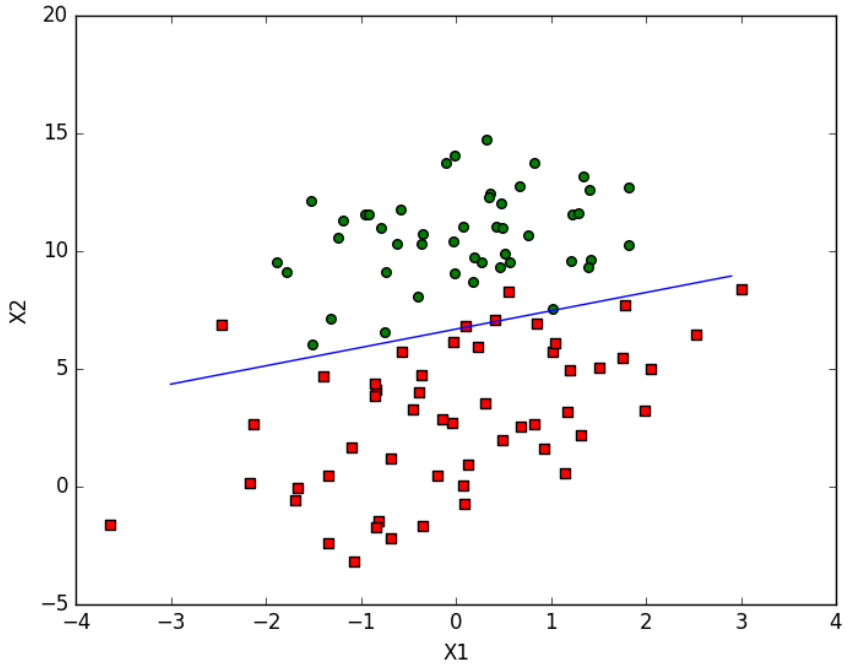
$$\nabla_i = \frac{\partial}{\partial \mathrm{w}_i} L(\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_n)$$

$$\mathrm{w}_i = \mathrm{w}_i - \alpha \nabla_i$$

2.2 Logistic Regression

Data needs to be classified

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

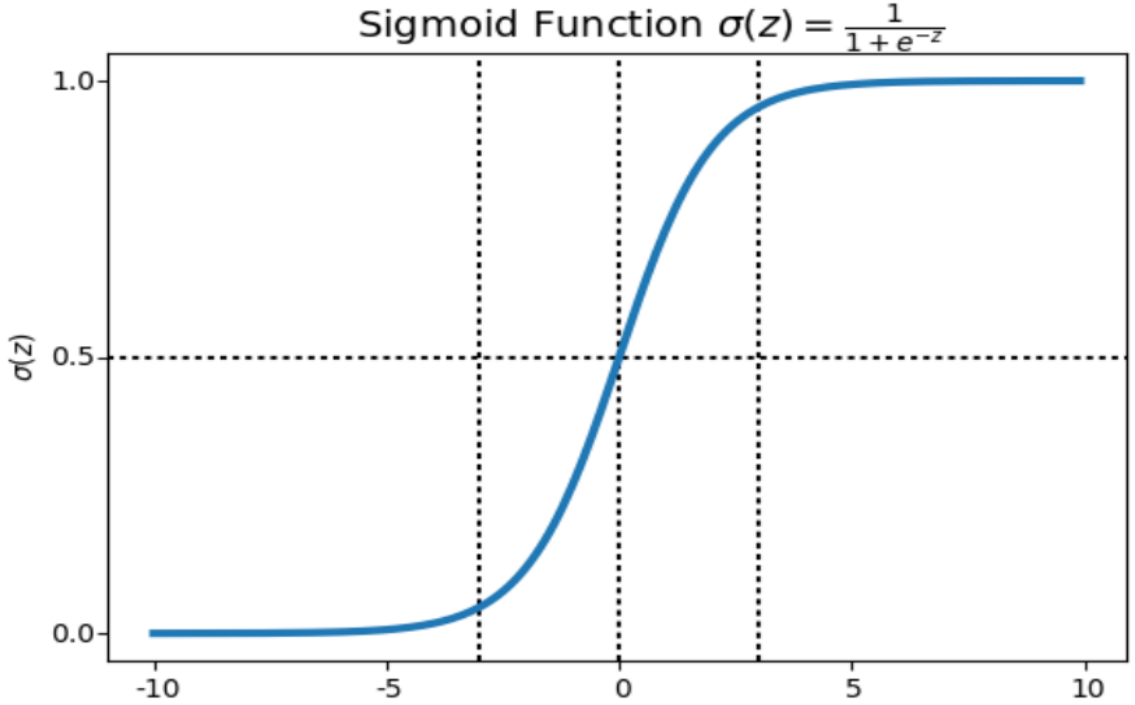


? How to quantify

2.2 Logistic Regression

Sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x} + b)}}$$



Why **Sigmoid** function

- result is compressed to 0-1
- result is converted to a probability value
- derivable and it's derived function is very easy

$$\nabla\sigma(z) = \sigma(z) \times \big(1 - \sigma(z)\big)$$

2.2 Logistic Regression

Loss function

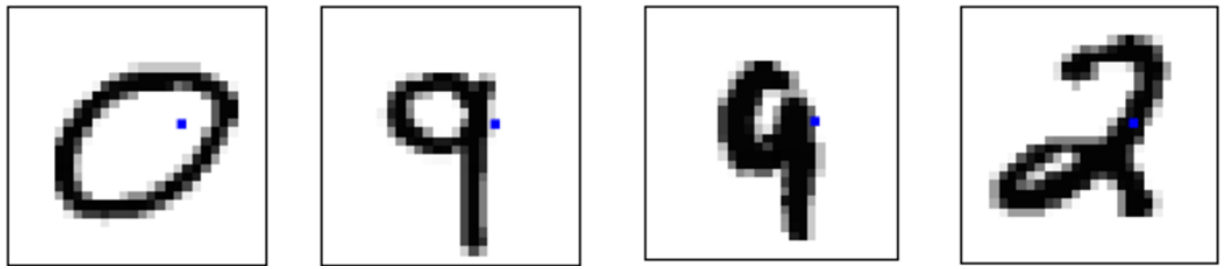
$$p(y=1|x)=\frac{e^{\boldsymbol{w}^T\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^T\boldsymbol{x}+b}}$$

$$p(y=0|x)=\frac{1}{1+e^{\boldsymbol{w}^T\boldsymbol{x}+b}}$$

$$\mathrm{L}(w,b)=\ln\left(\prod_{i=1}^{m}p\left(y^i|x^i;w,b\right)\right)=-\sum_{i=1}^{m}\left(y^i\times\ln\left(\sigma\left(x^i\right)\right)+(1-y^i)\times\ln\left(1-\sigma\left(x^i\right)\right)\right)$$

Better than other Machine Learning algorithm

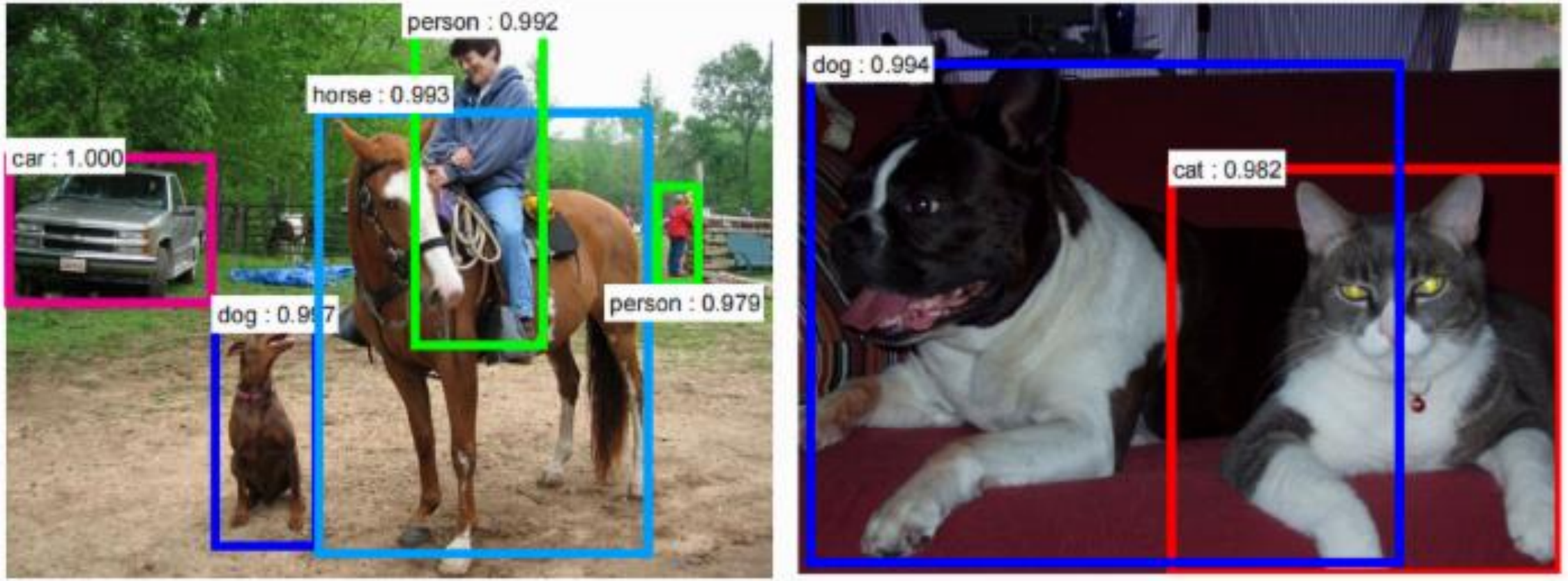
THE MNIST DATABASE



- 28×28
- 0-9

http://yann.lecun.com/exdb/mnist/

3.1 Application



object detection

3.1 Application

A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background

A little **girl** sitting on a bed with a teddy bear.

A group of **people** sitting on a boat in the water.

A giraffe standing in a forest with **trees** in the background.

From image to text

3.1 Application

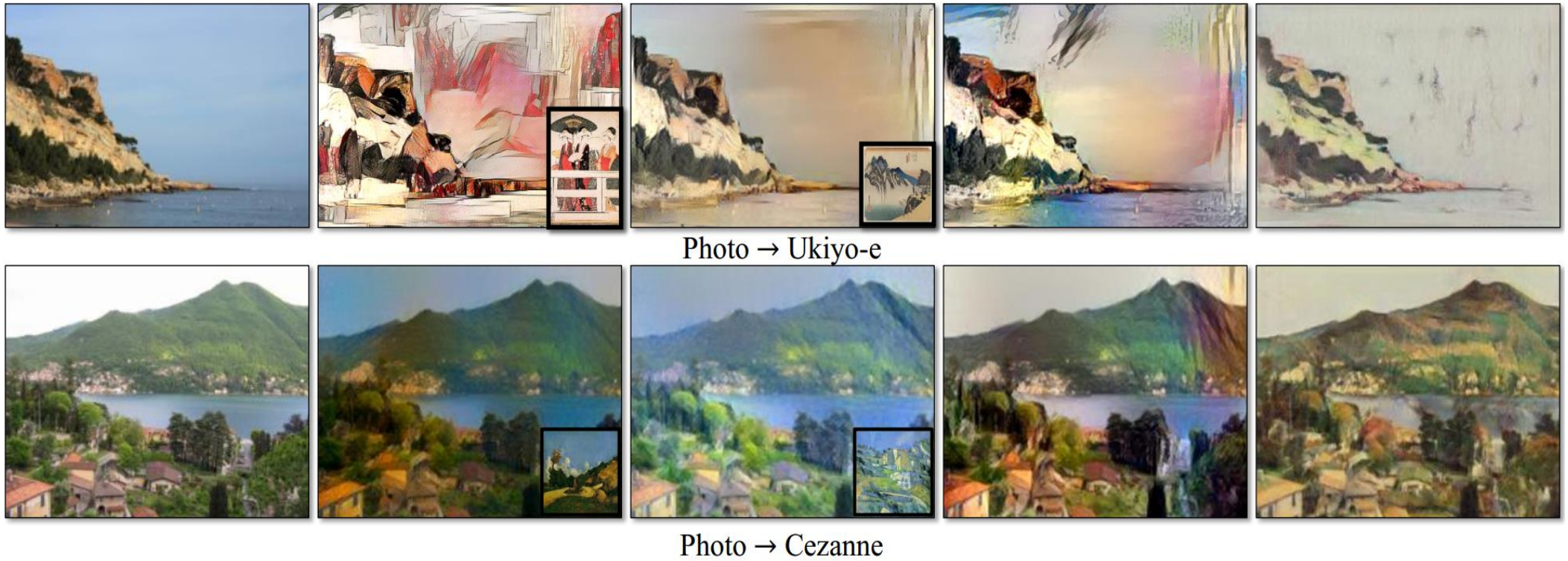
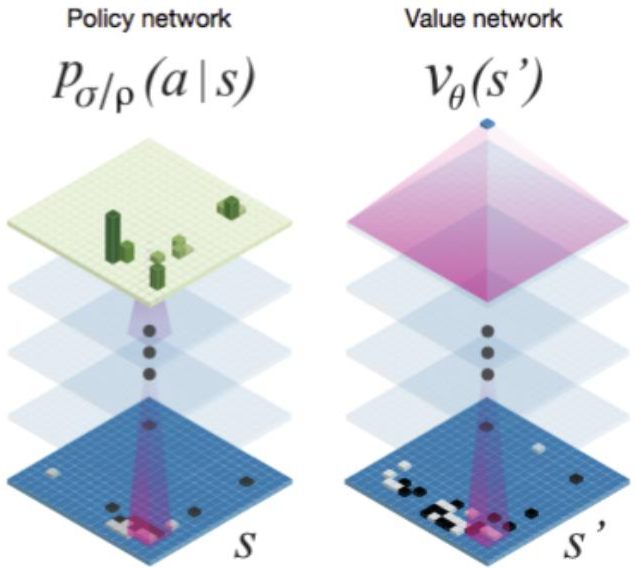


photo stylization

3.1 Application

Twitch anchor Charles

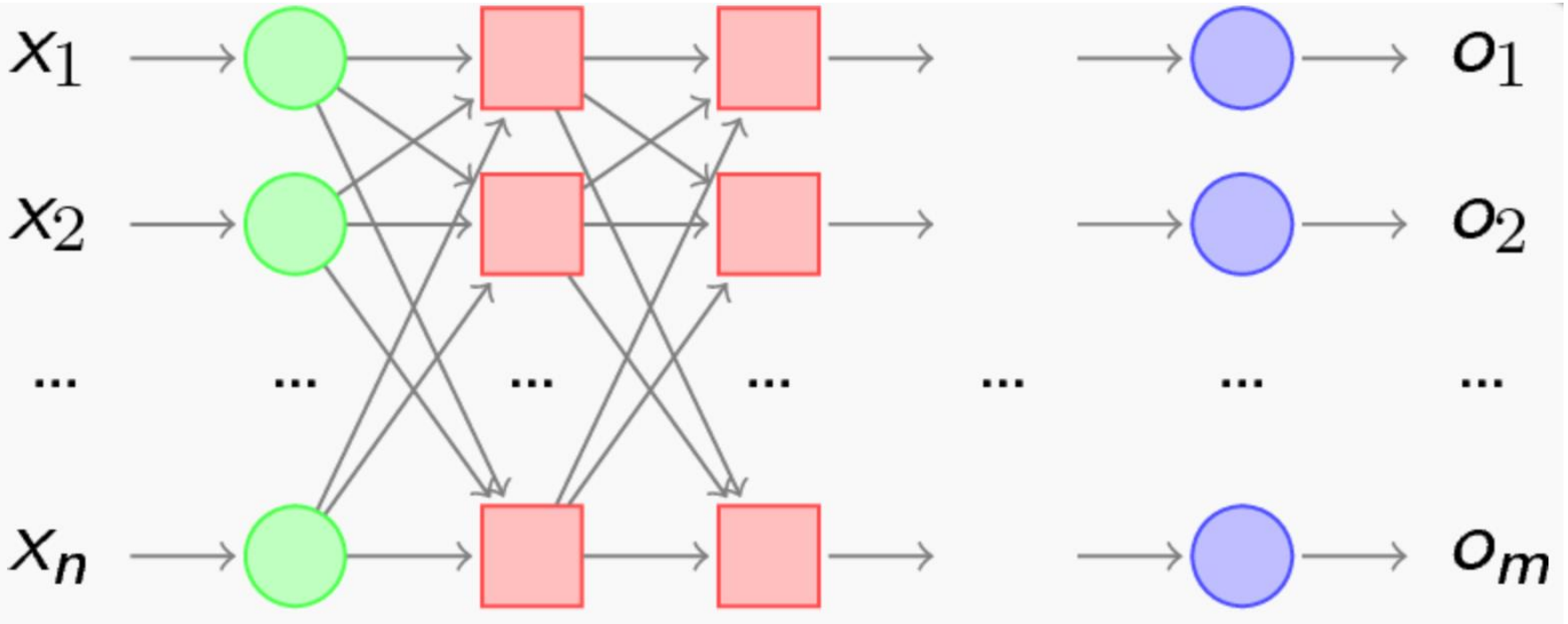


mastering the game of go

Even better than human!

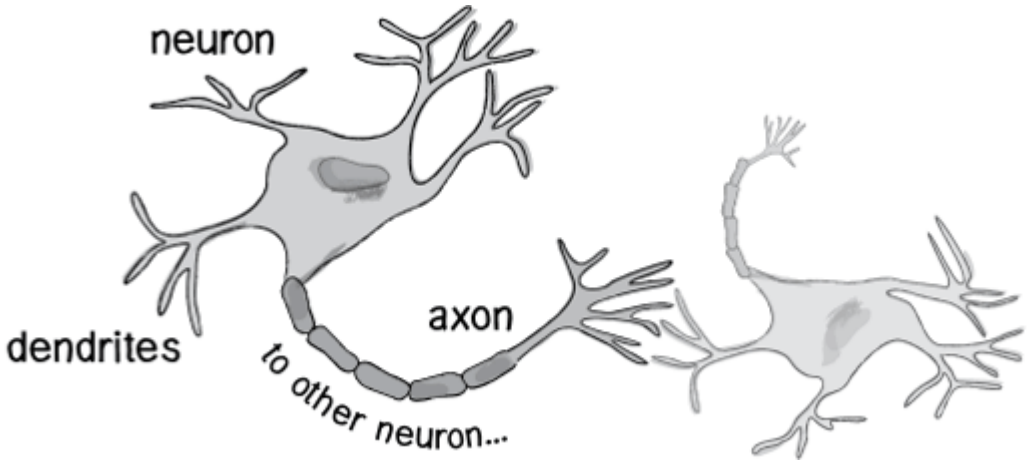
04 Neural Network

Basic structure of Artificial Neural Network



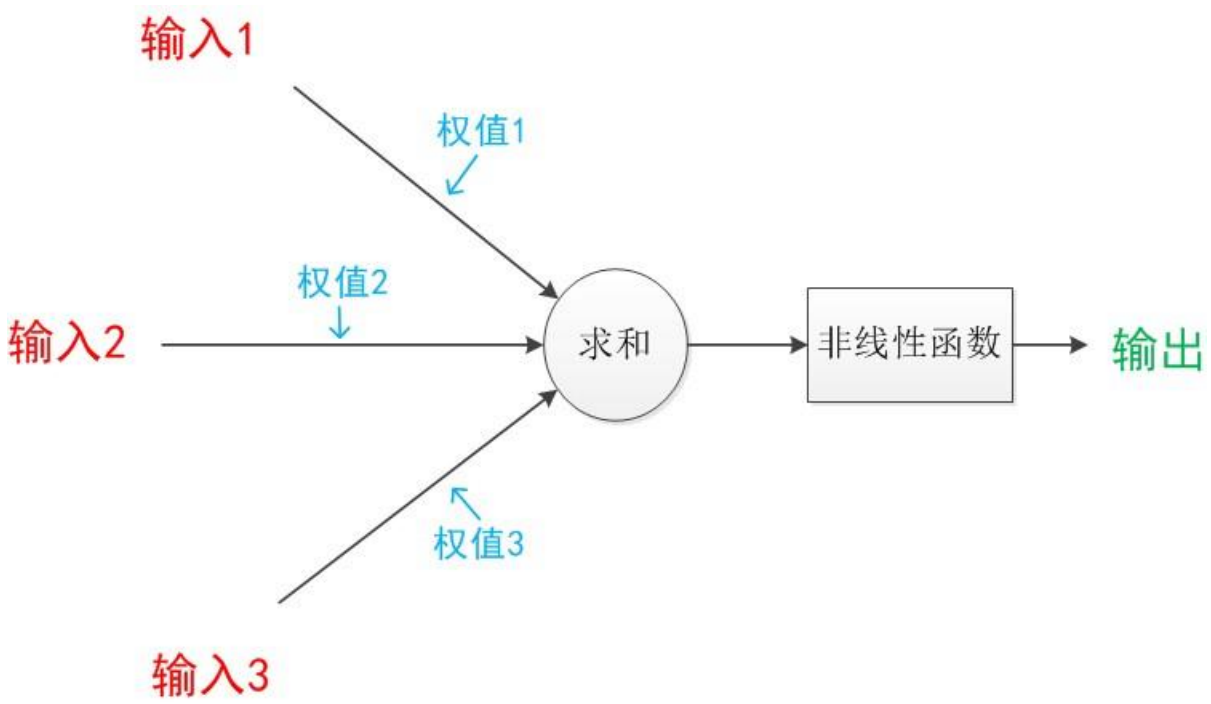
Input Layer Hidden Layer Output Layer

4.1 perceptron



1. Input (external stimulation)
2. Weights (Quantitative stimulation)
3. Activation function (generate signal)
4. Output (signal transmission)

4.1 perceptron

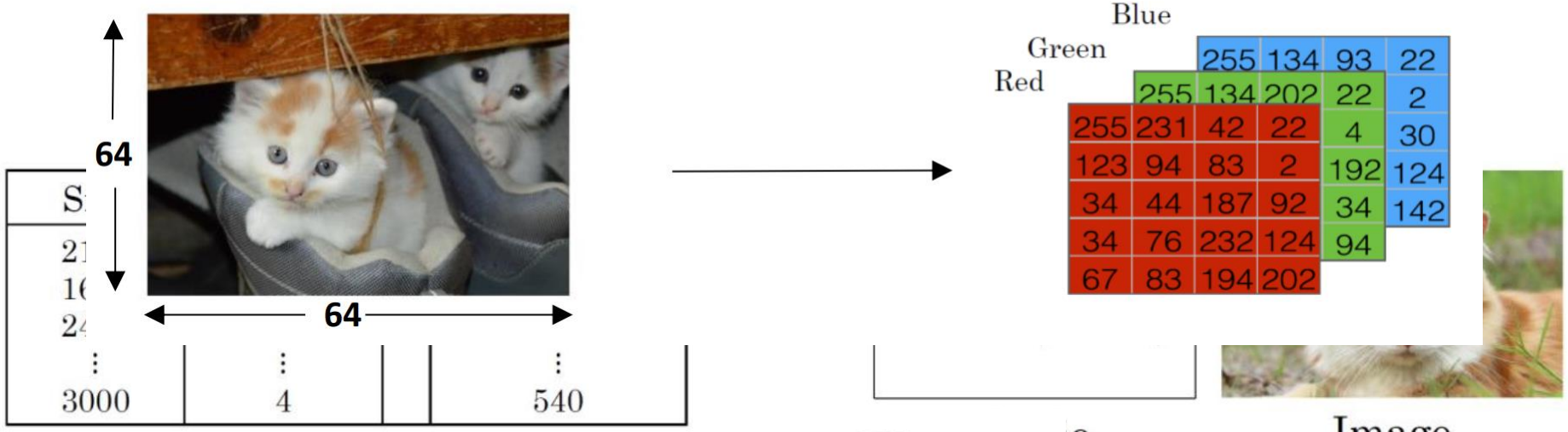


$$\vec{x} = (a_1, a_2, a_3)$$

$$\sigma(z) = \frac{1}{1 + e^{-(z)}}$$

$$\hat{y} = \sigma\left(w^T \boldsymbol{x} + b\right)$$

4.2 Input Layer



User Age	Ad Id	···	Click
41	93242		1
80	93287		0
18	87312		1
⋮	⋮		⋮
27	71244		1

$$x = \begin{bmatrix} 255 \\ 231 \\ 42 \\ \vdots \\ 255 \\ 134 \\ 202 \\ \vdots \\ 255 \\ 134 \\ 93 \\ \vdots \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \text{red} \\ \\ \\ \\ \text{green} \\ \\ \\ \\ \text{blue} \\ \\ \end{matrix}$$

4.3 Hidden Layer

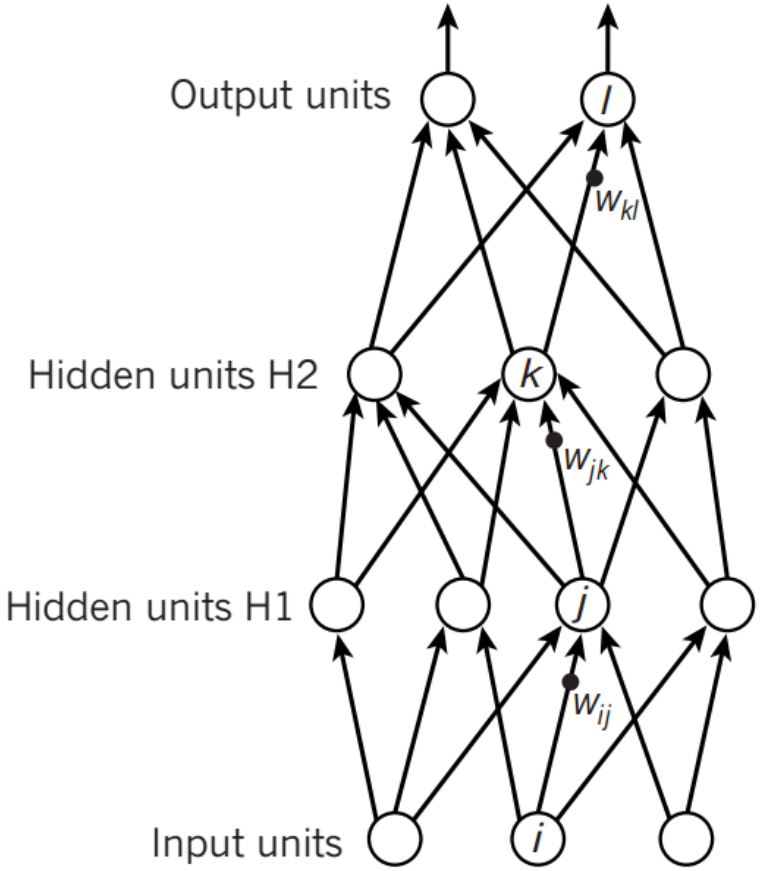
Loss functions

- $L\big(Y, f(X)\big) = \big(Y - f(X)\big)^2$
- $L\big(Y, f(X)\big) = -lnP(Y|X)$
- $L\big(Y, f(X)\big) = \max\big(0,1 - Y \times f(X)\big)$

Expected loss function

$$j(w,b) = \frac{1}{m}\sum_{i=1}^{m} L\big(Y, f(X)\big)$$

4.3 Hidden Layer



$$y_l = f(z_l)$$
$$z_l = \sum_{k \,\varepsilon\, \text{H2}} w_{kl}\, y_k$$

$$y_k = f(z_k)$$
$$z_k = \sum_{j \,\varepsilon\, \text{H1}} w_{jk}\, y_j$$

$$y_j = f(z_j)$$
$$z_j = \sum_{i \,\varepsilon\, \text{Input}} w_{ij}\, x_i$$

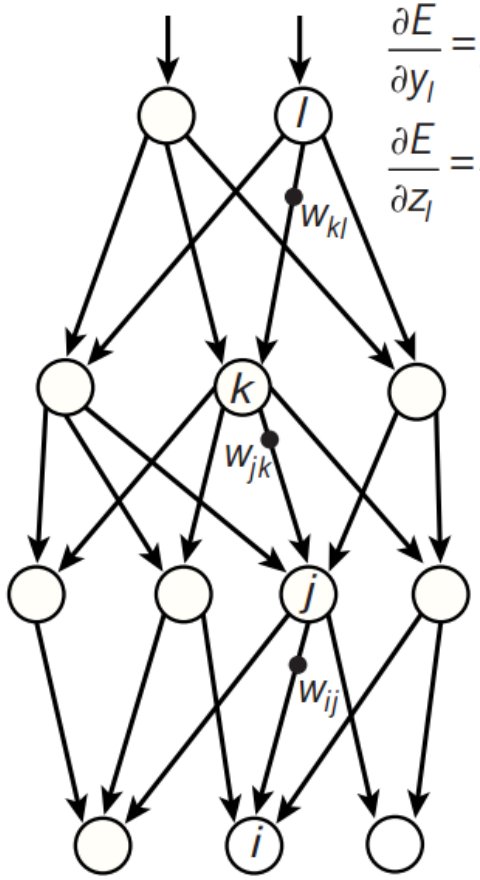
forward-propagation

Compare outputs with correct answer to get error derivatives

$$\frac{\partial E}{\partial y_l} = y_l - t_l$$
$$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l}\frac{\partial y_l}{\partial z_l}$$

$$\frac{\partial E}{\partial y_k} = \sum_{l \,\varepsilon\, \text{out}} w_{kl}\frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial z_k}$$



$$\frac{\partial E}{\partial y_j} = \sum_{k \,\varepsilon\, \text{H2}} w_{jk}\frac{\partial E}{\partial z_k}$$

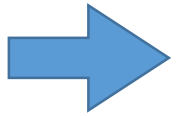
$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial z_j}$$

back-propagation

4.3 Output Layer

encoding of data： One-Hot

{0~9}

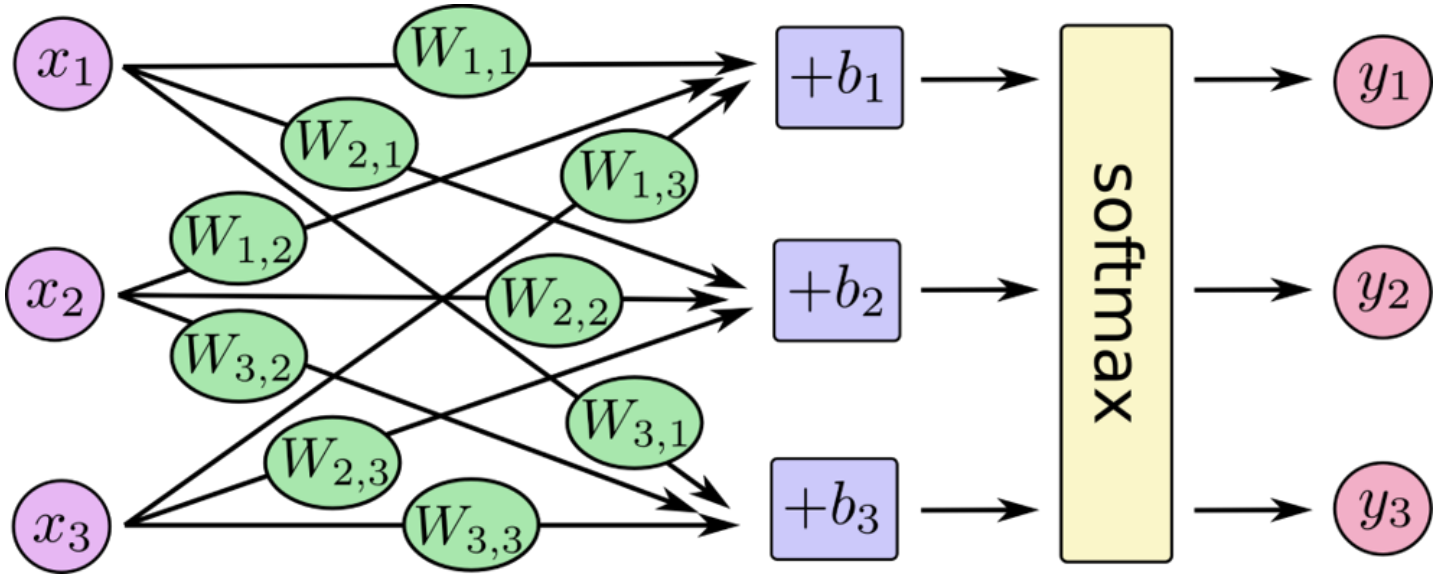


$$\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}\quad\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}\quad\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}\quad\cdots\quad\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}$$

0 1 2 9

4.3 Output Layer

Softmax



easy to calculate

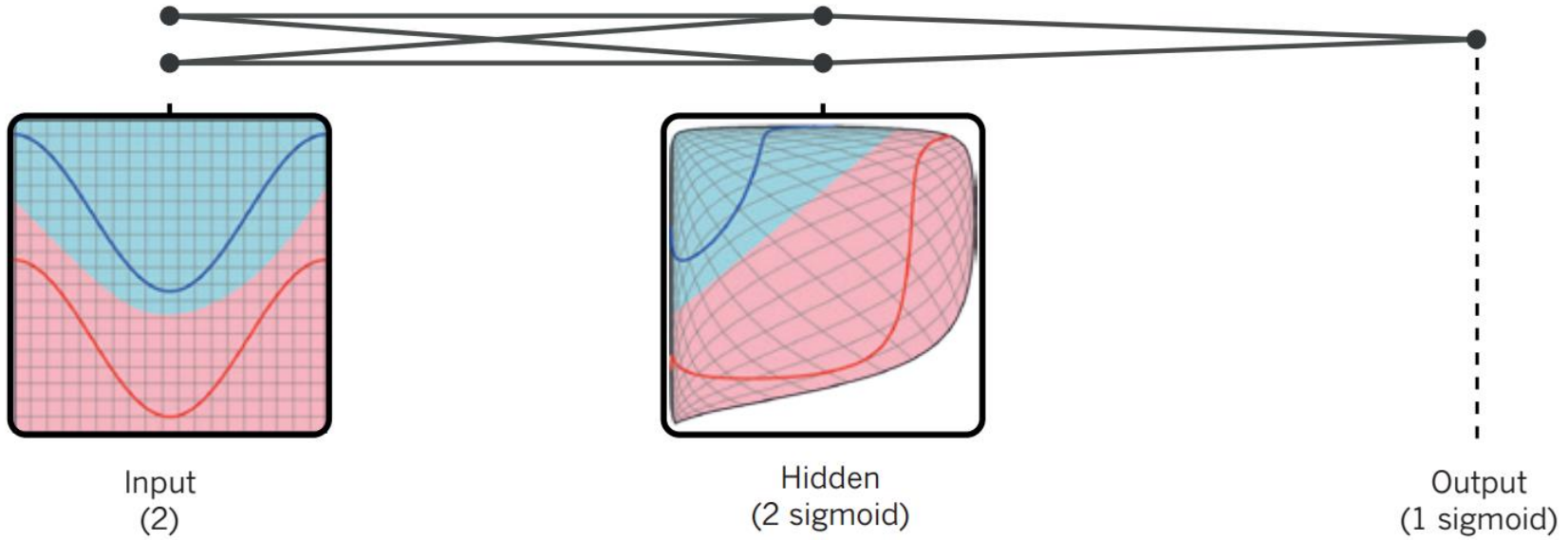
- $S_i = \frac{e^{v_i}}{\sum_j e^{v_j}}$

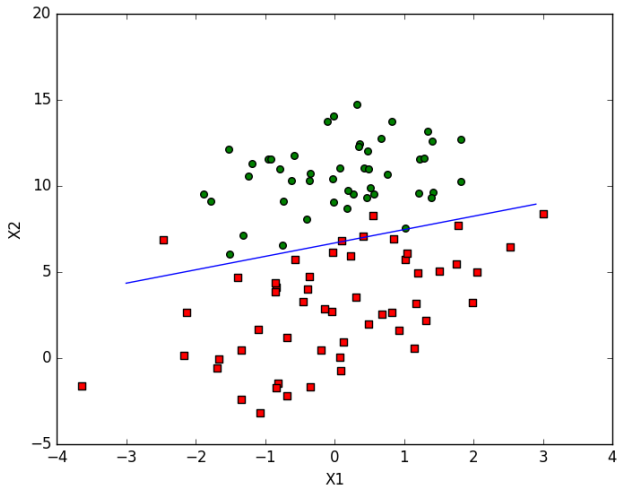
derivable and it's derived function is very easy

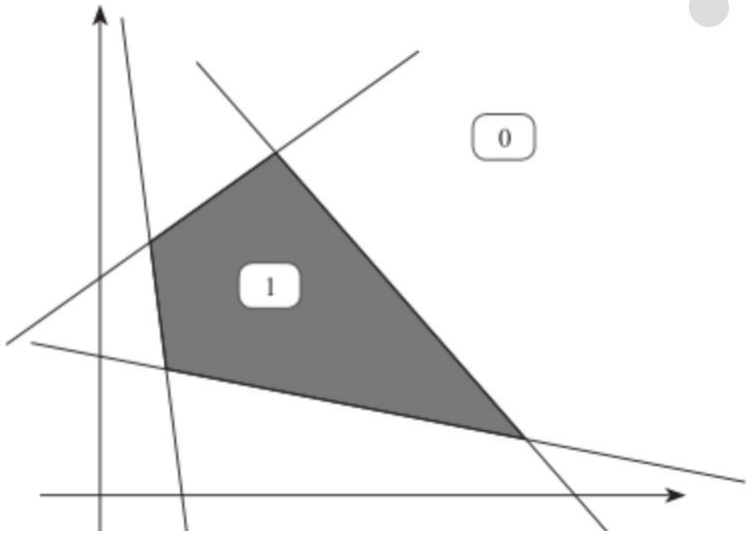
- $\nabla = S_i - 1$

convert data to **One-Hot** encoding

4.3 Why so accurate

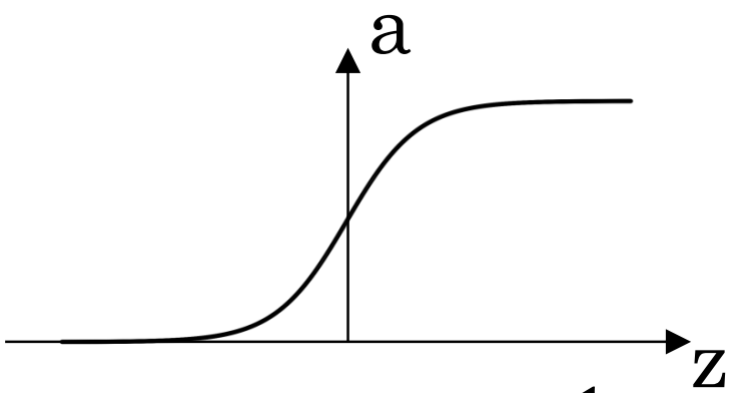




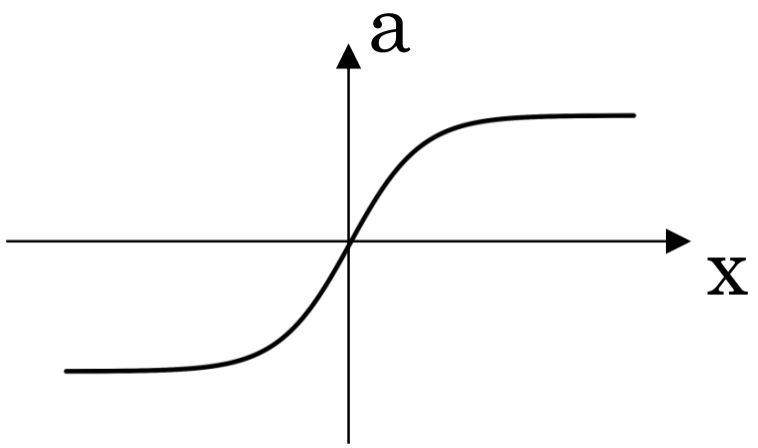


05 Improvement of Neural Network

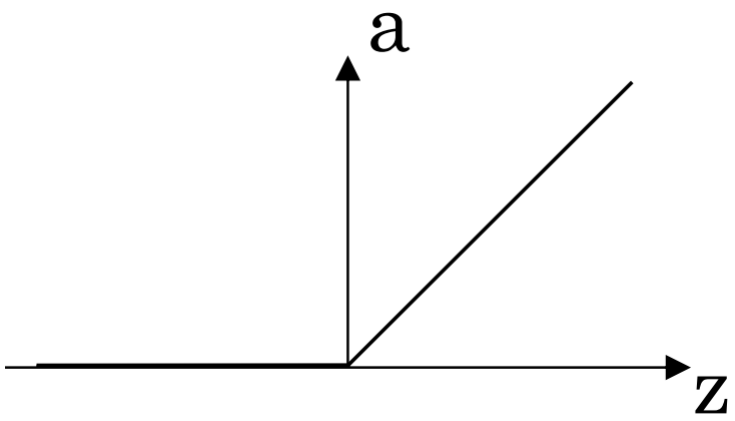
Activation functions



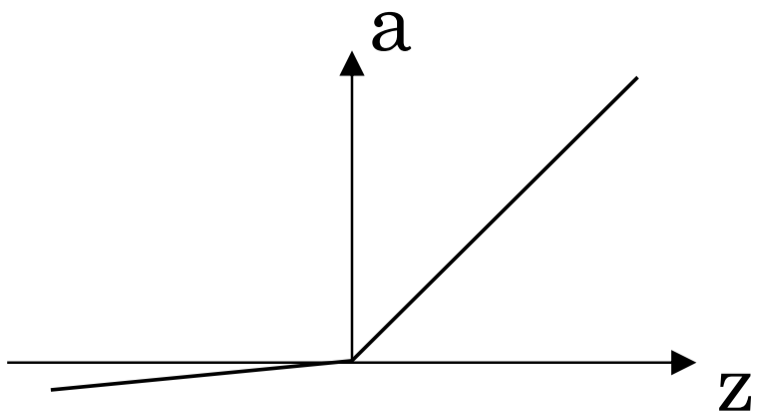
Sigmoid



Tanh



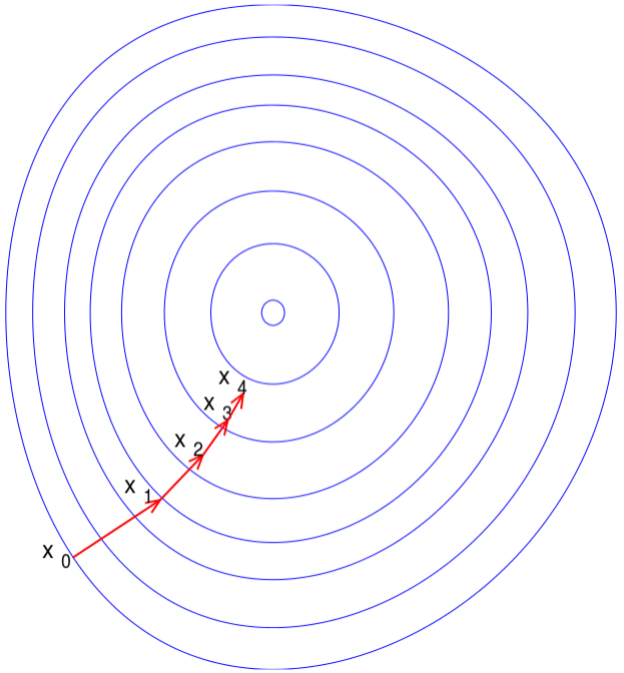
ReLU



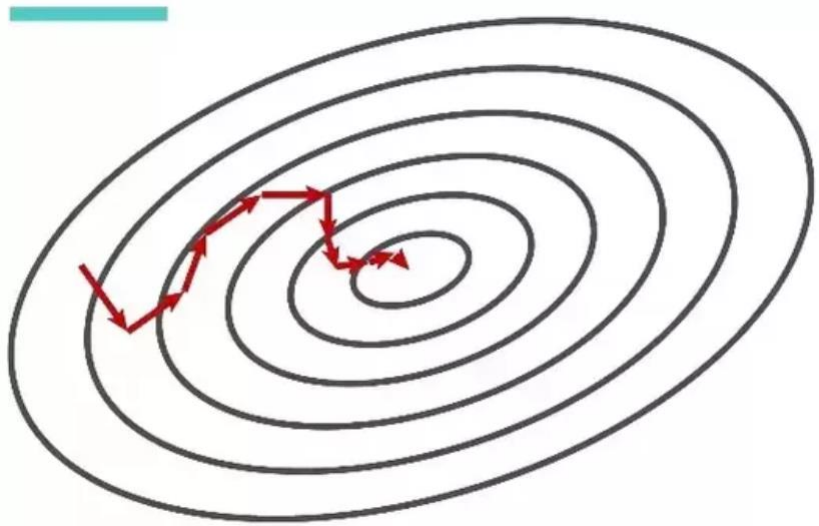
ReLU

05 Improvement of Neural Network

Gradient Descent



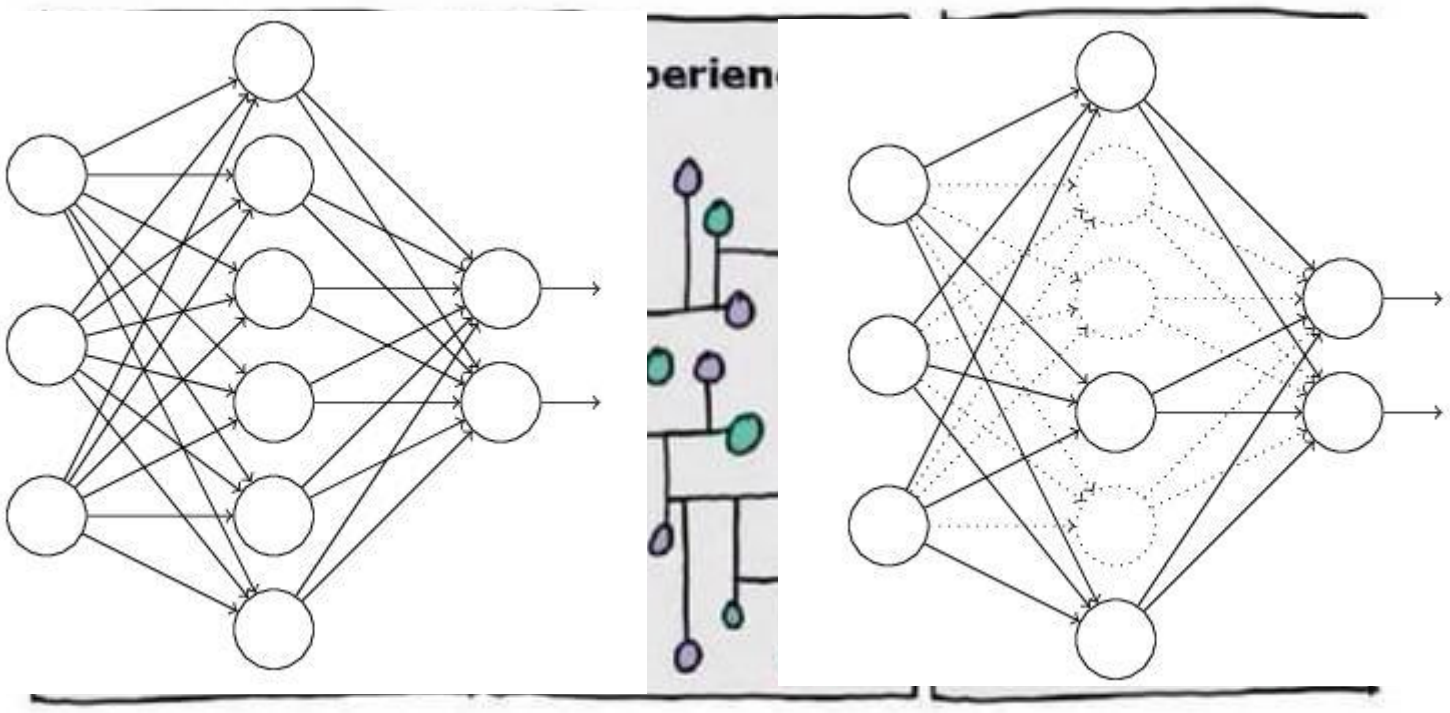
Batch Gradient Descent



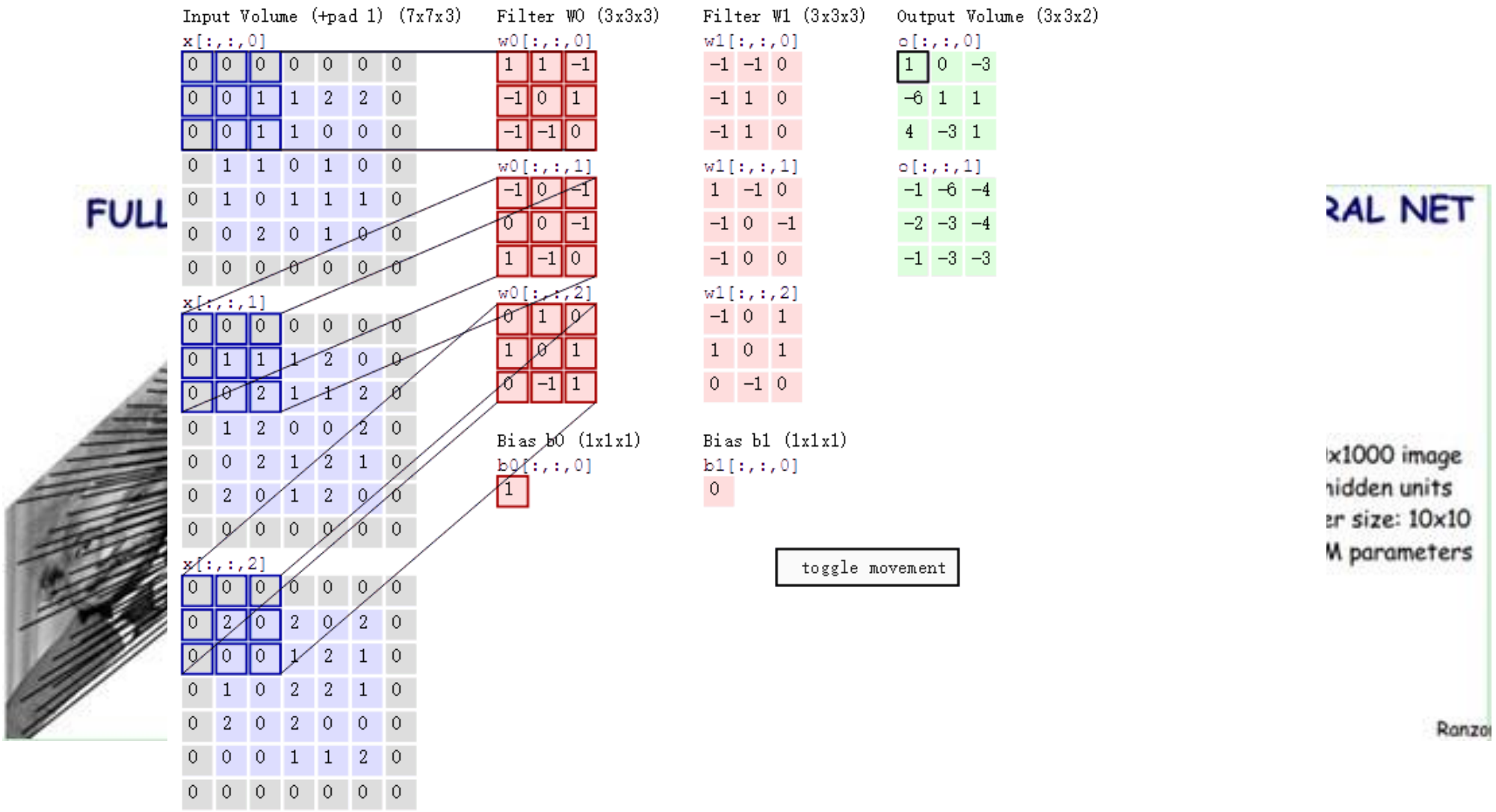
Stochastic Gradient Descent

05 Improvement of Neural Network

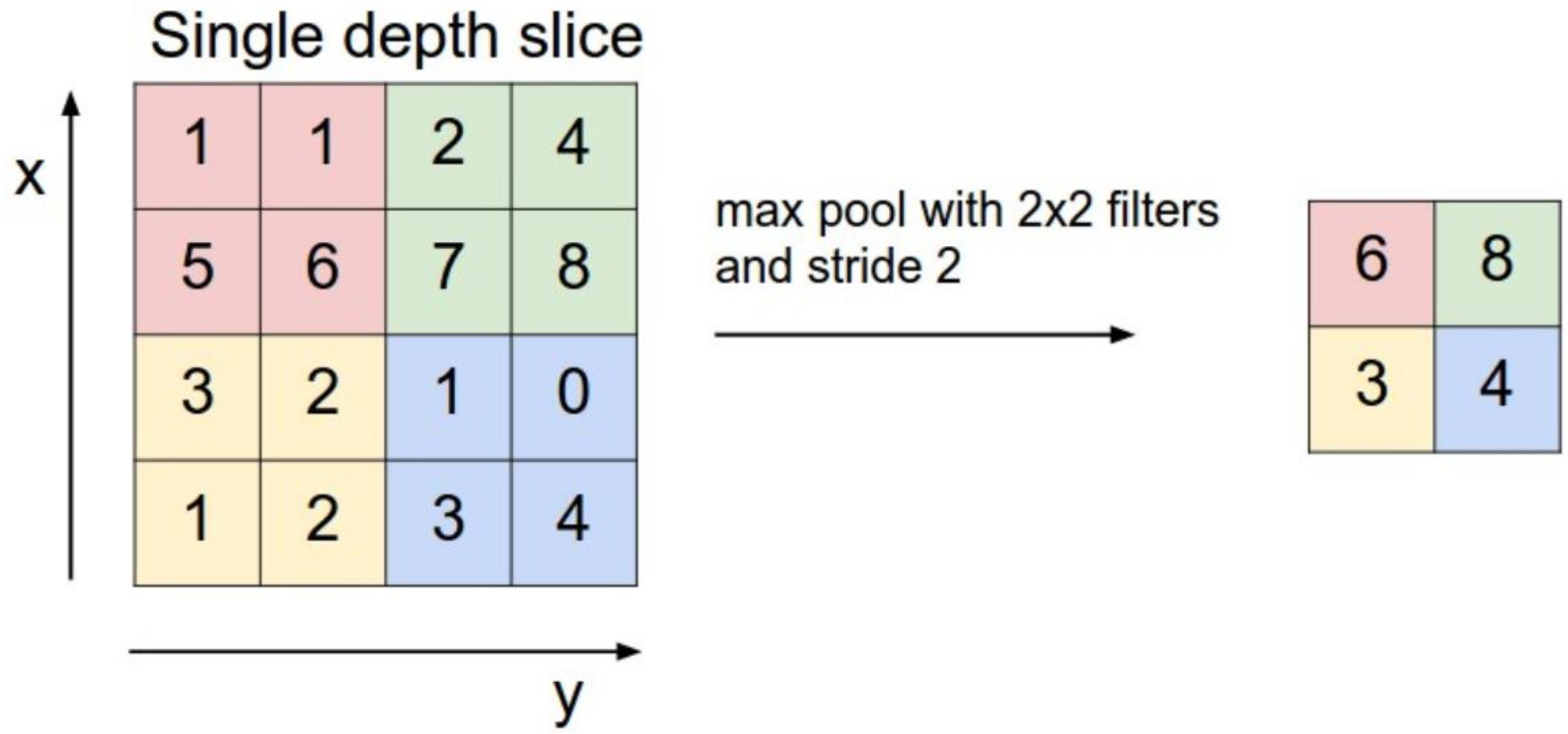
Dropout



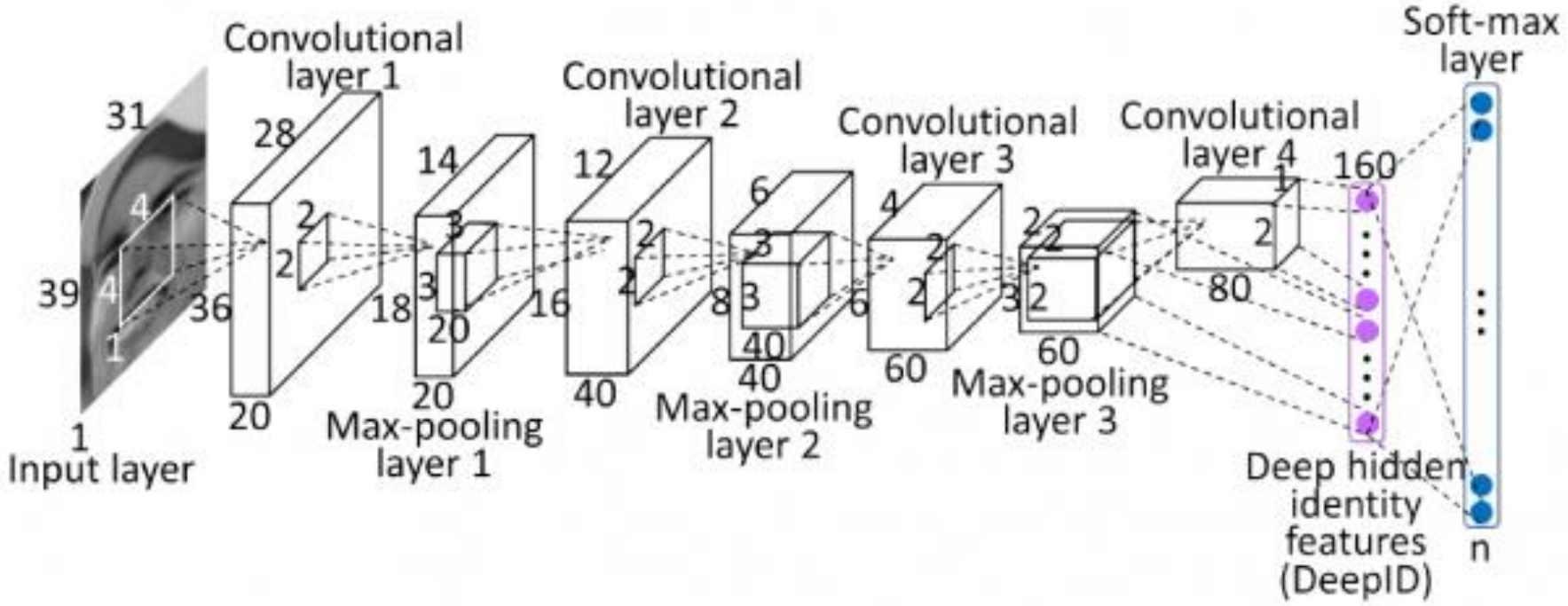
5.1 Convolutional Neural Networks



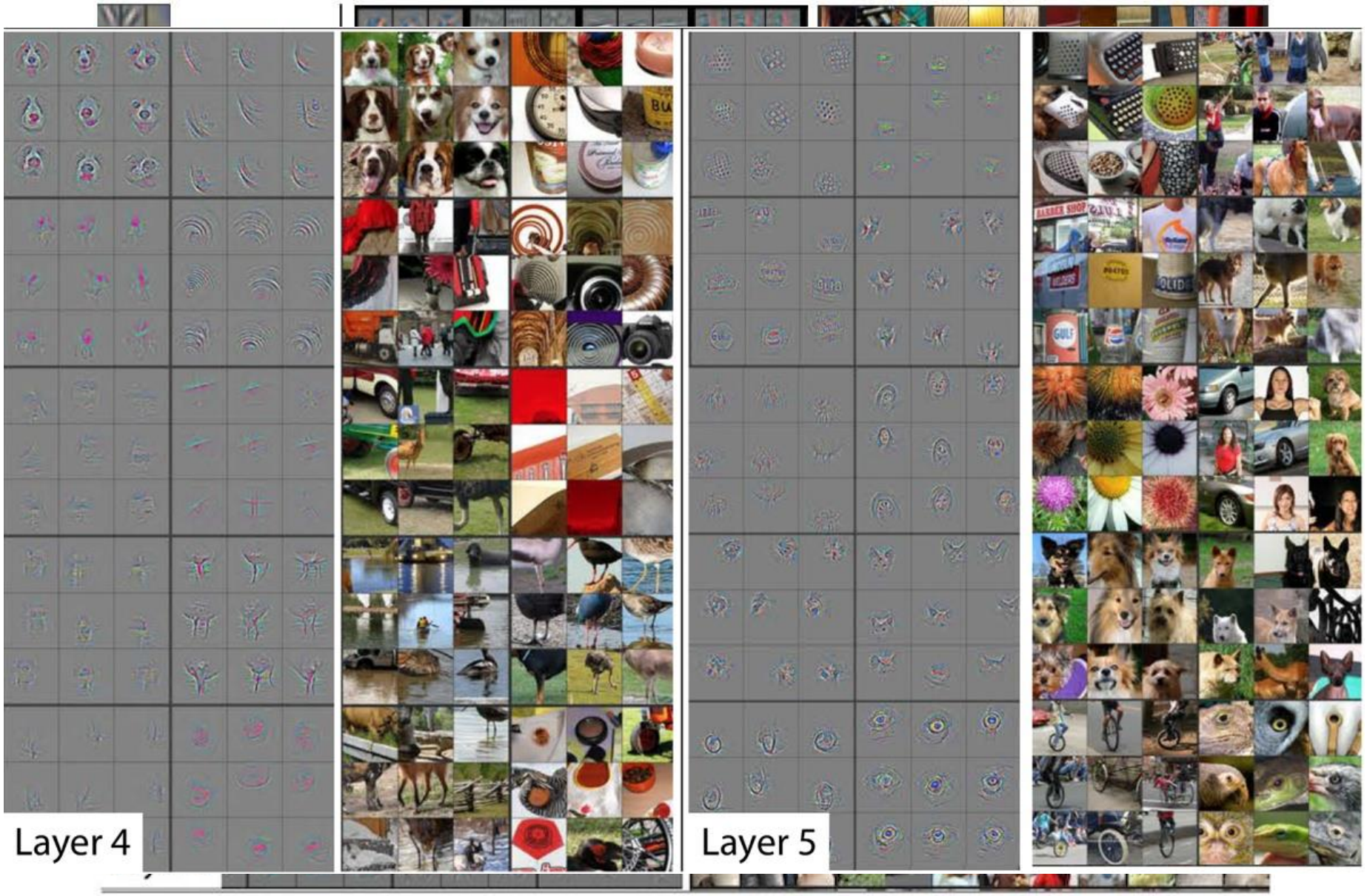
5.1 Pooling


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

5.1 Structure of a Convolutional Neural Networks



5.1 How does Convolutional Neural Networks work



5.2 Recurrent Neural Networks

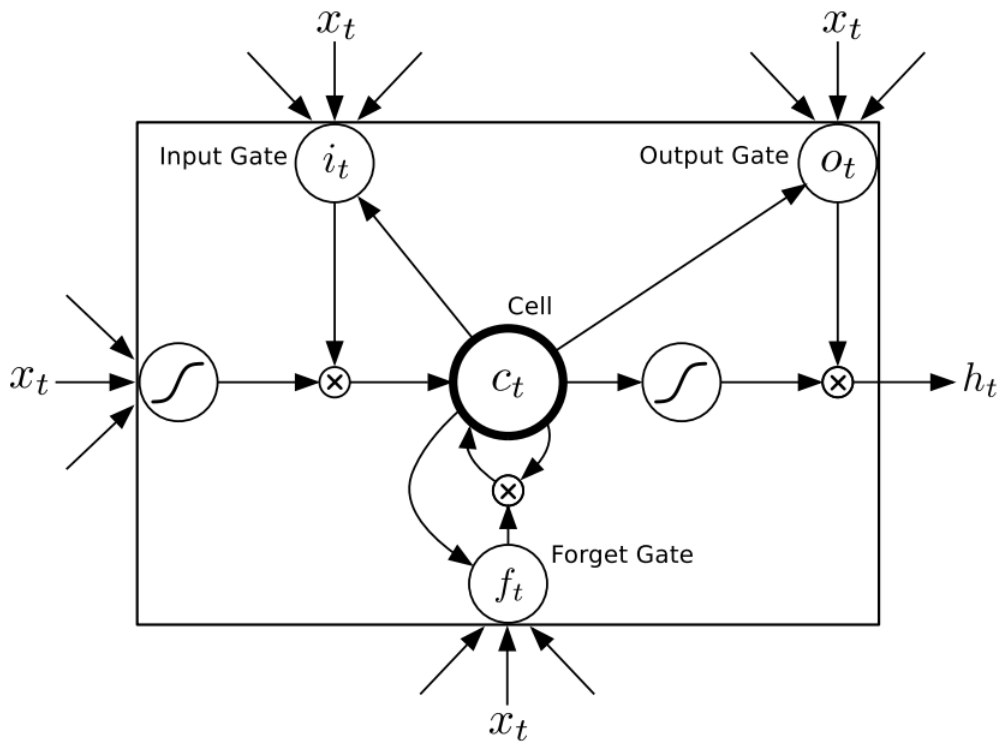


Fig. 1. Long Short-term Memory Cell

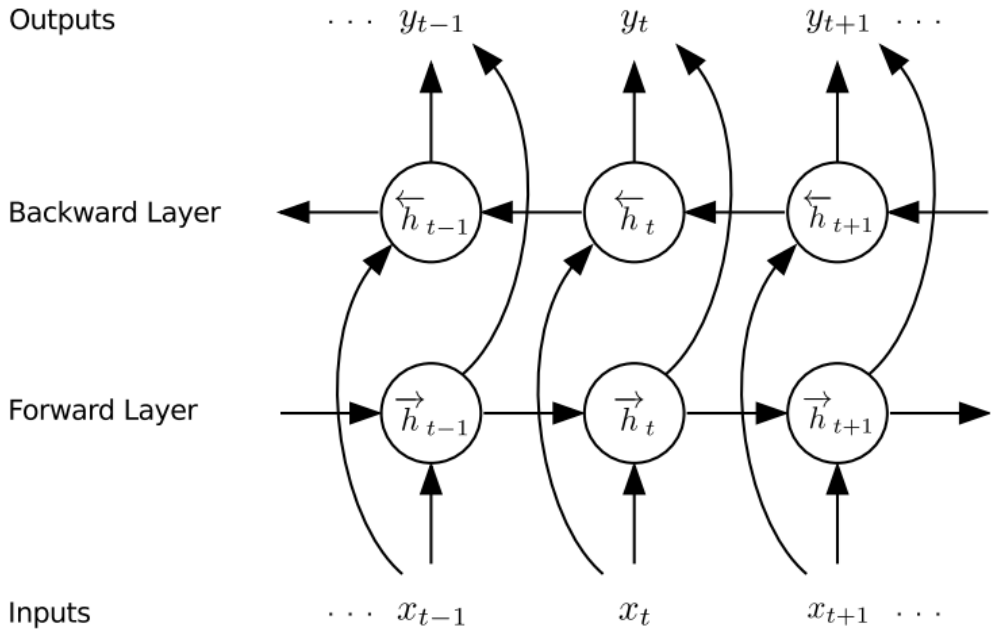
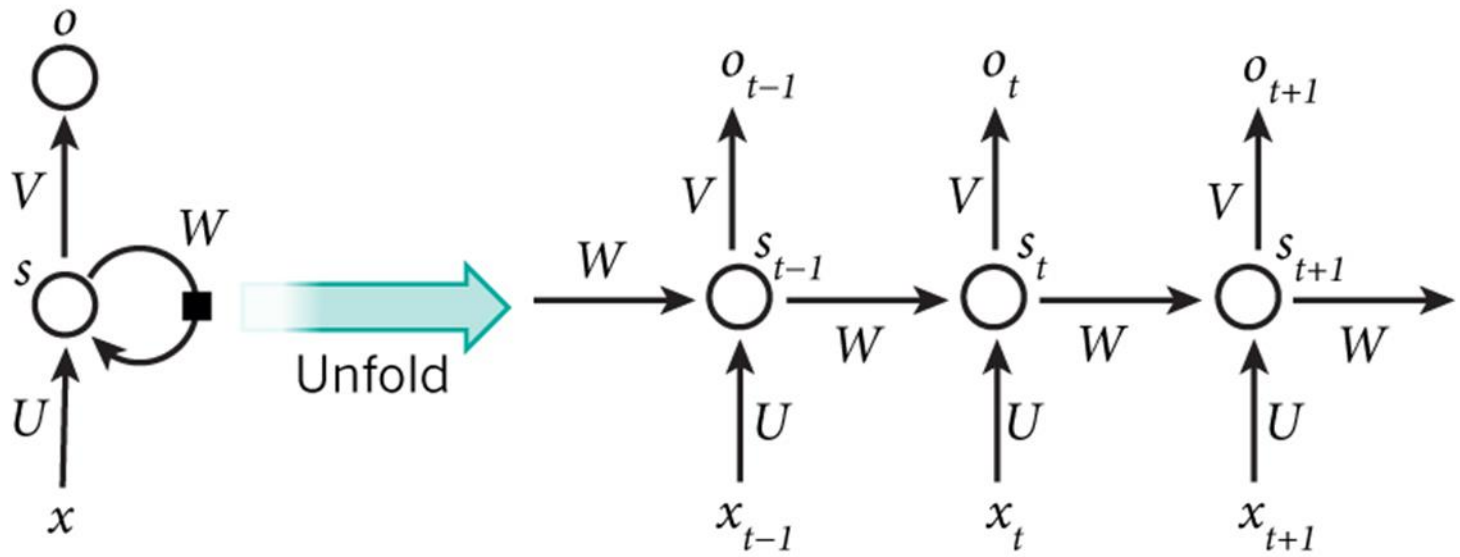
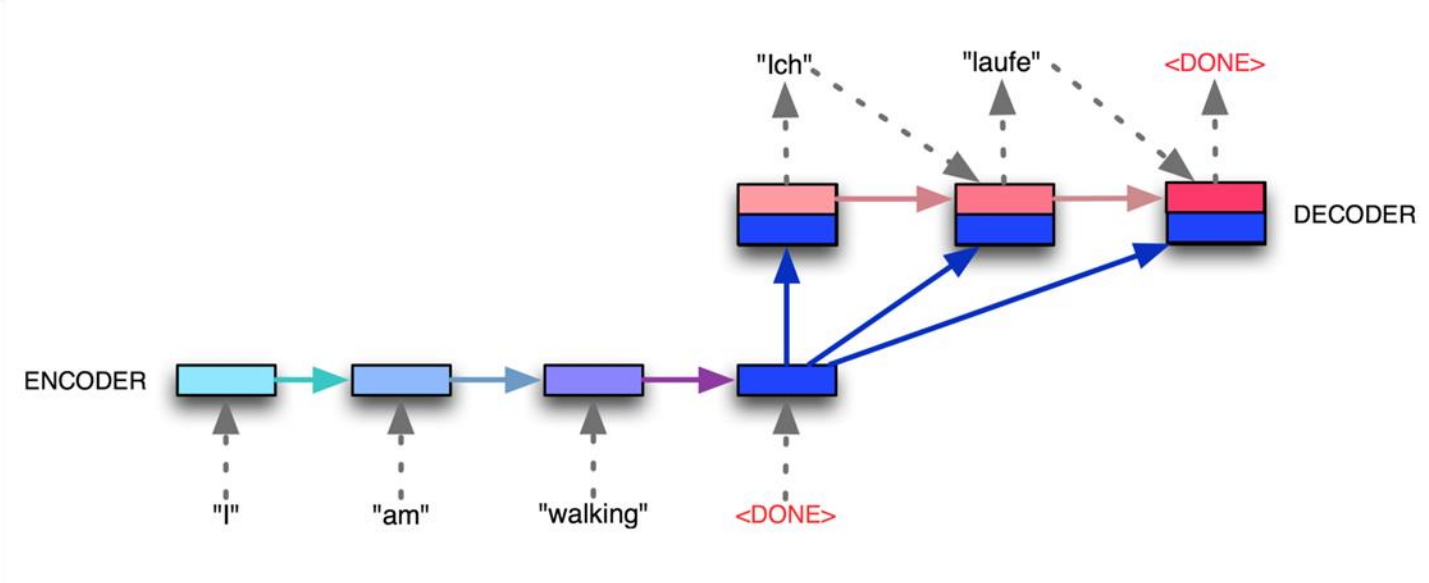


Fig. 2. Bidirectional RNN

5.2 Recurrent Neural Networks





Citation

[1] Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." An MIT Press book. (2015).

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks.“ Advances in neural information processing systems. 2012.

[3] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.

[4] Zhu, Jun Yan, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." (2017).

[5] deeplearning.ai, https://www.deeplearning.ai/. (2018)

[6] 机器学习: 基本概念, 五大流派与九种常见算法, http://wwv.cyzone.cn/a/20170422/310196.html. (2017.04.22)

[6] 黄安埠. 深入浅出深度学习[M]. 北京： 电子工业出版社, 2017

[7] 高扬, 卫峥. 白话深度学习与TensorFlow[M]. 北京： 机械工业出版社, 2017

THANK YOU!

Q & A