人工智能之自動化光學檢測實務

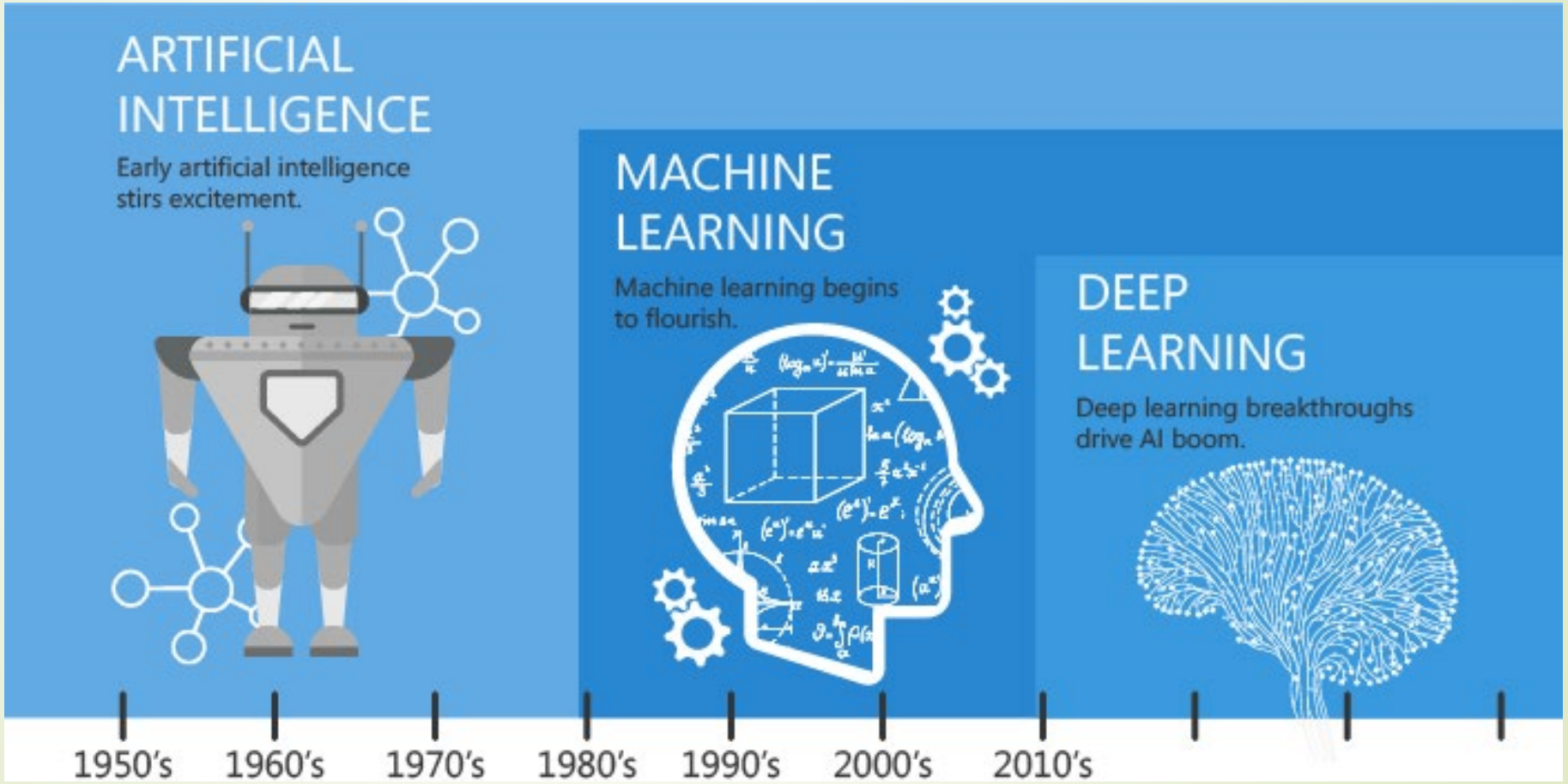
Yi-Yung Chen

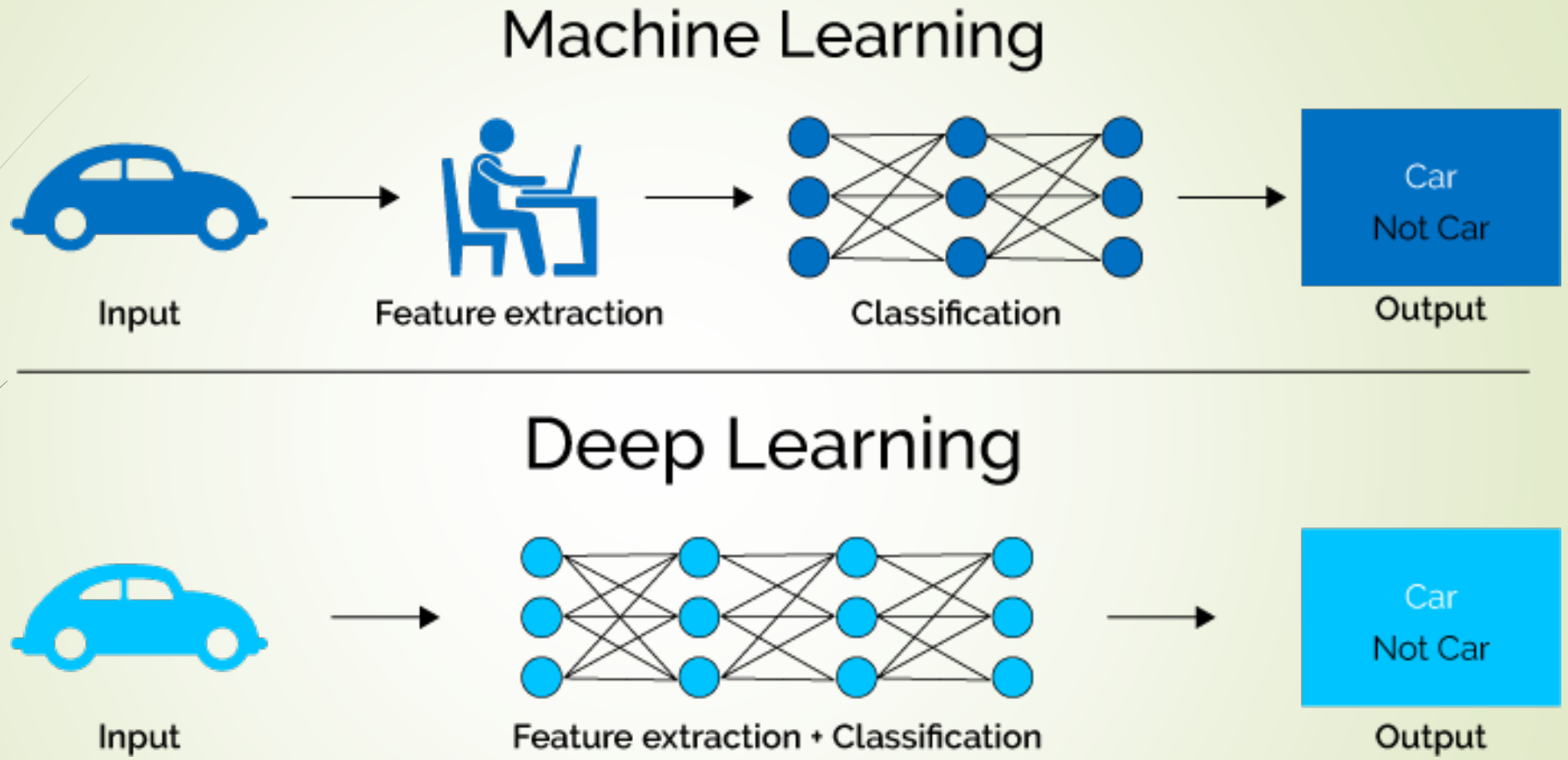
適用影片單元10~單元16

深度學習 (Deep Learning)

- 人工神經網絡
- 卷積神經網絡

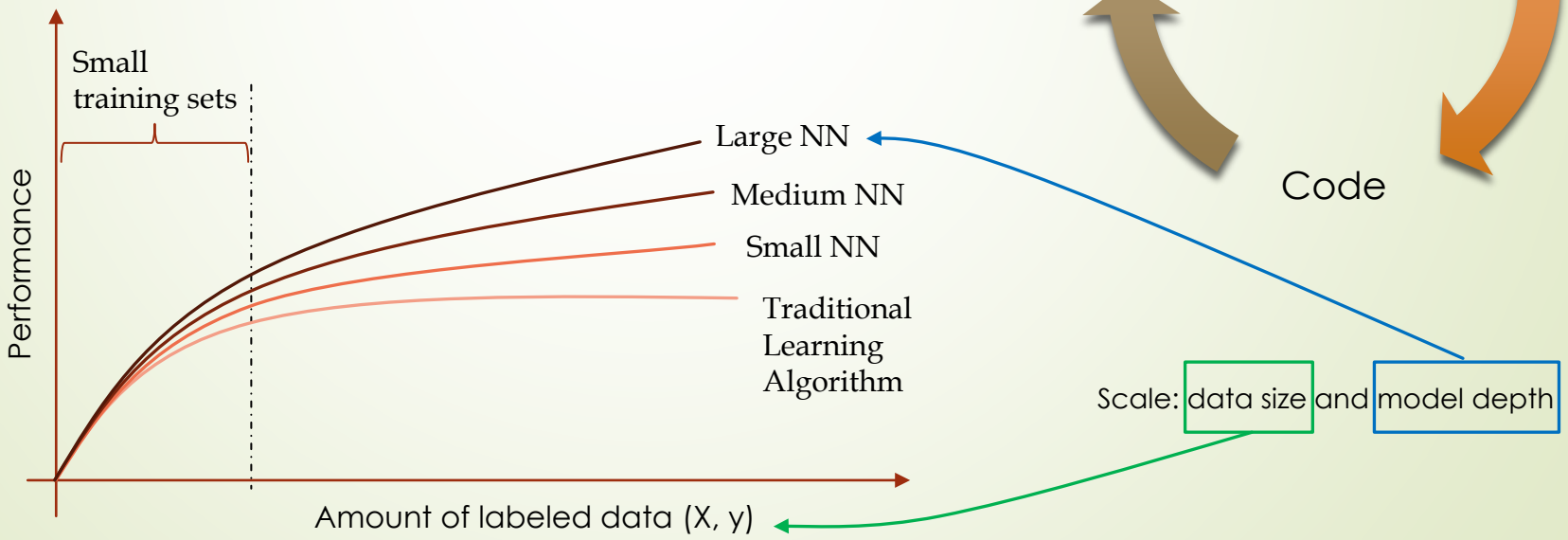
Development of AI




Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output

Scale Drives Deep Learning Progress

- Data
- Computation
- Algorithms
  - Sigmoid -> ReLU

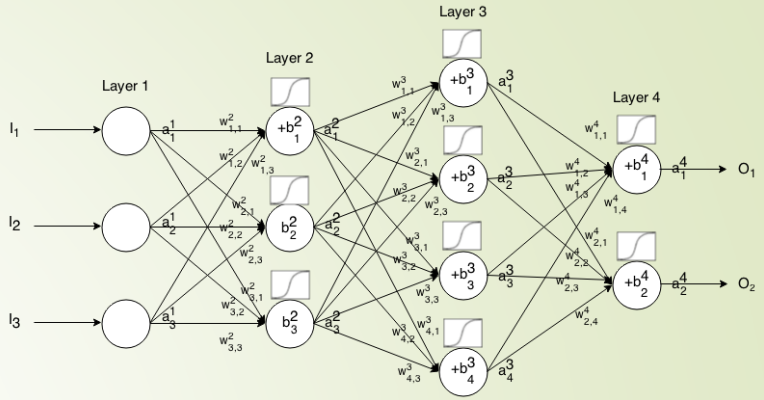


人工神經網絡

人工神經網路 (Artificial Neural Networks)

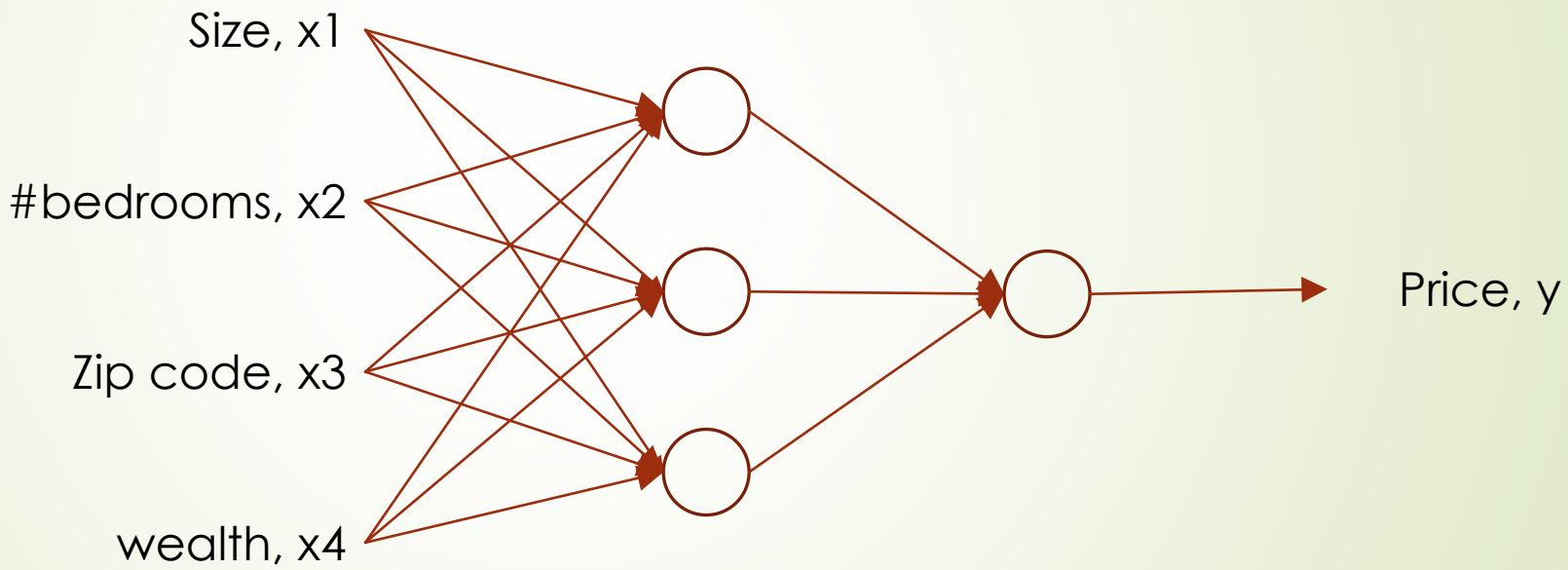
- 神經元 (The Neuron)
- 激勵函數 (Activation Function)
- 神經網路如何運作
- 神經網路如何訓練
- 梯度下降 (Gradient Descent)
- 隨機梯度下降 (Stochastic Gradient Descent)
- 反向傳播 (Back Propagation)

Training the ANN

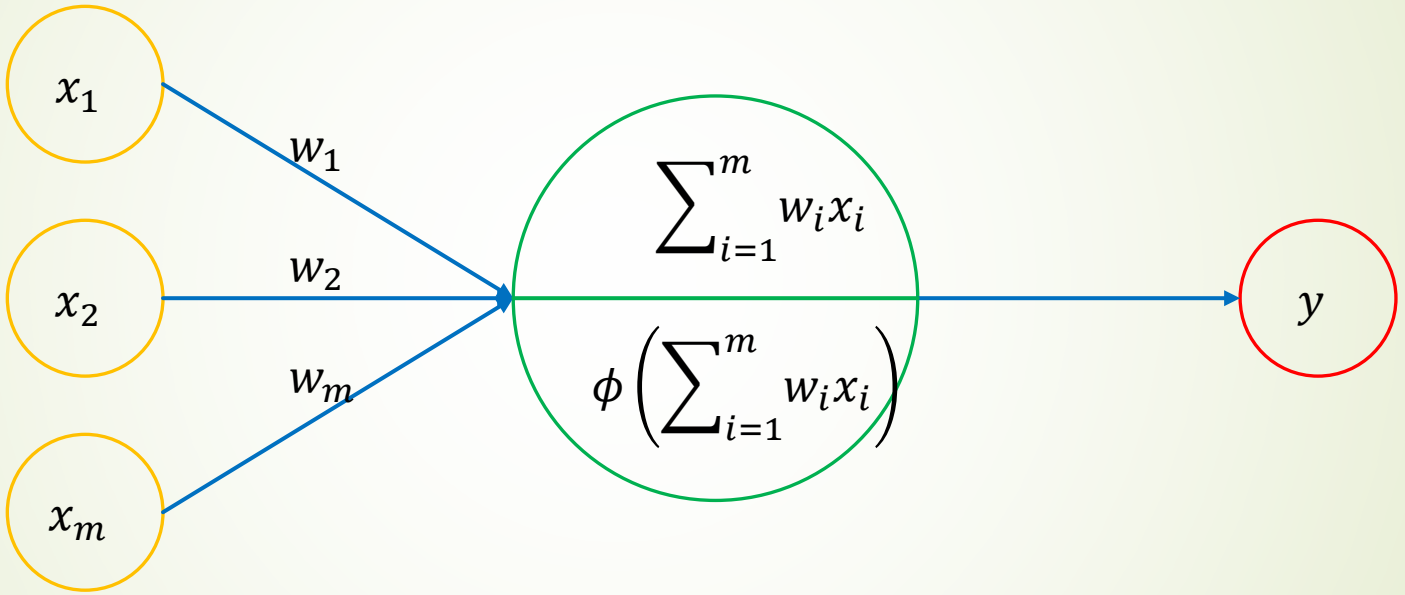


1. 隨機初始化權重，使得權重接近0但不等於0
2. 將第一個觀察數據輸入“輸入層”，每個自變量特徵輸入一個神經元
3. 正向傳播：神經網路從左至右進行計算，每個神經元得到來自上一層神經元的輸入與權重的運算
4. 誤差函數：計算預測值與實際值的差異 (**loss function**)
5. 反向傳播：神經網路從右至左進行計算，依據誤差函數相對於權重的梯度，對每個權重進行更新。學習速率與梯度將決定更新的速度 (**optimizer**)
6. 對每一組新的觀察數據，重複step 1到step 5 (**batch size**)
7. 當所有數據都輸入神經網路後，稱之為一期(**epoch**)的訓練

Housing Price Prediction



The Neuron



自變量

標準化：平均=0、標準差=1
歸一化：範圍0~1

Standardisation	Normalisation
$x_{stand} = \frac{x - mean(x)}{StandardDeviation(x)}$	$x_{norm} = \frac{x - min(x)}{max(x) - min(x)}$

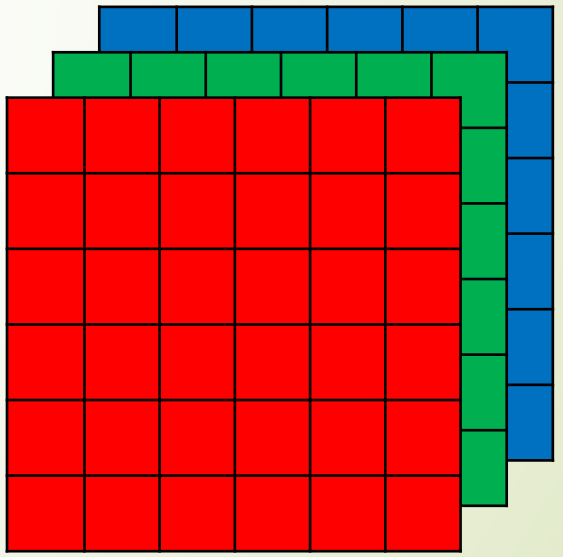
應變量

1. Continuous (price)
2. Binary (will exit yes/no)
3. Categorical

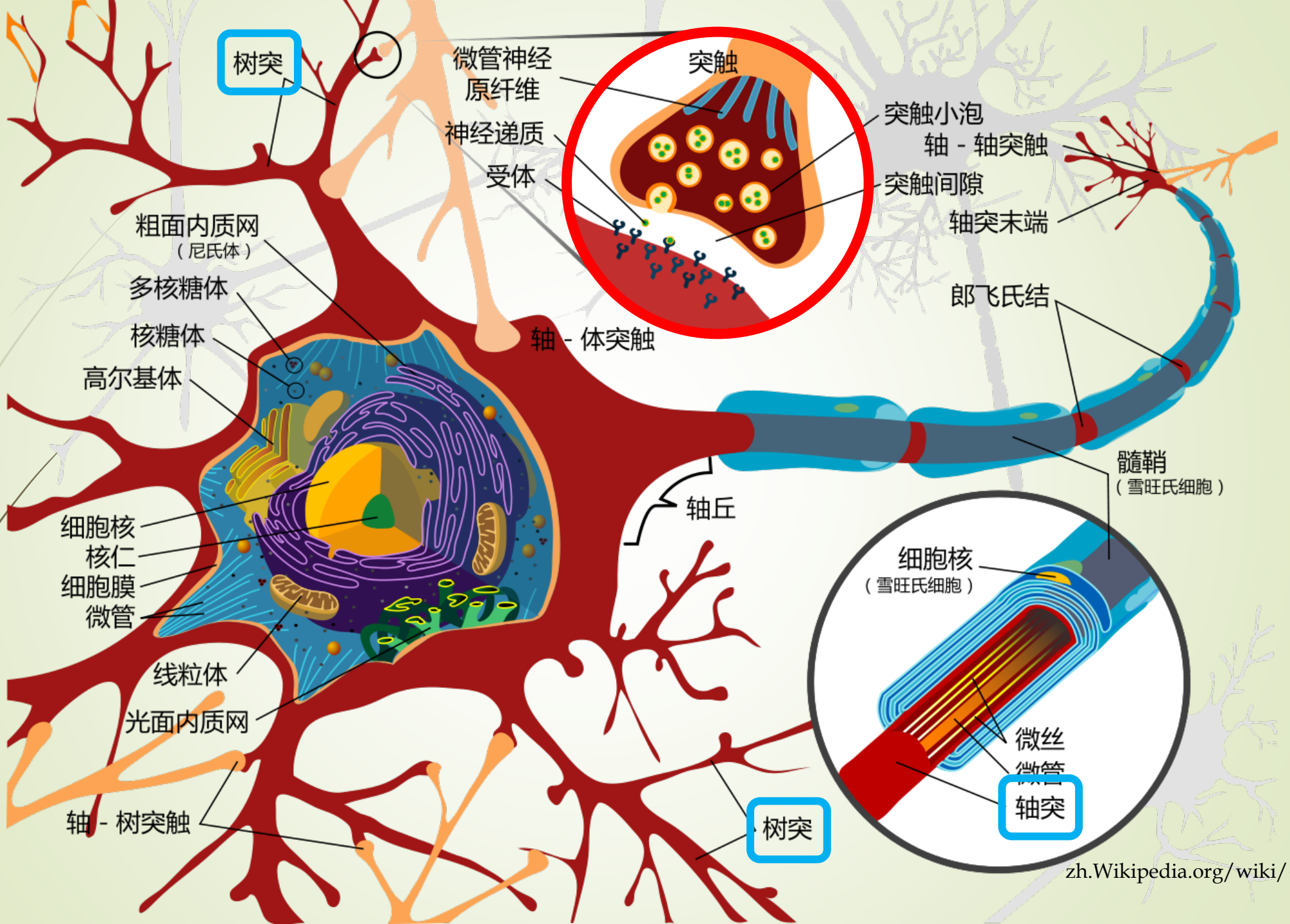
Notation

- (x, y)
  - $x \in \mathbb{R}^{n_x}, y \in \{0, 1\}$
- m training examples
  - $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$
- $X \in \mathbb{R}^{n_x \times m}, Y \in \mathbb{R}^{1 \times m}$
  - $X = \begin{bmatrix} | & & | \\ x^{(1)} & \cdots & x^{(m)} \\ | & & | \end{bmatrix}$
  - $Y = [y^{(1)}, \cdots, y^{(m)}]$

Input



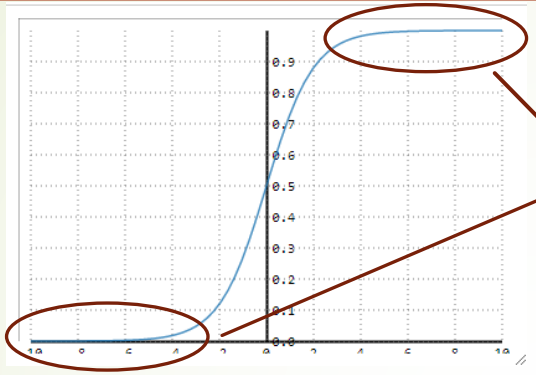
Ex. 64 x 64 x **3**, $n^x = 64 \times 64 \times 3 = 12{,}288$



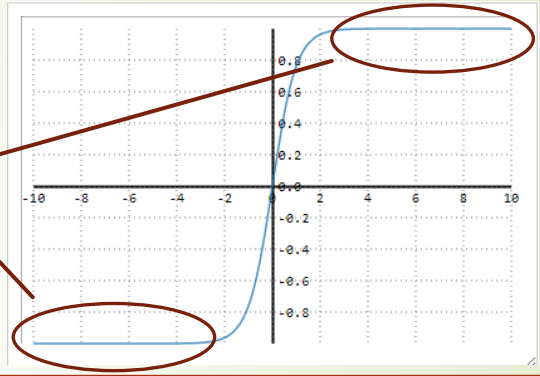
zh.Wikipedia.org/wiki/神經元

Activation function

Homework:
- Why do you need non-linear activation functions?



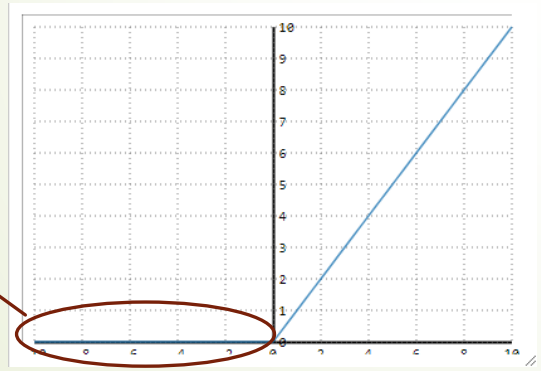
梯度消失

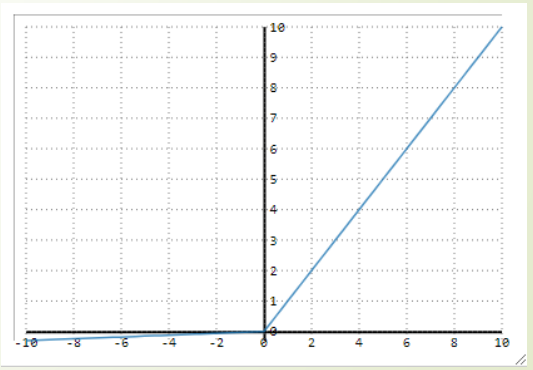


$$sigmoid: a = g(z) = \frac{1}{1+e^{-z}}$$

$$tanh: a = g(z) = \frac{(e^{z}-e^{-z})}{(e^{z}+e^{-z})}$$

Dead neurons



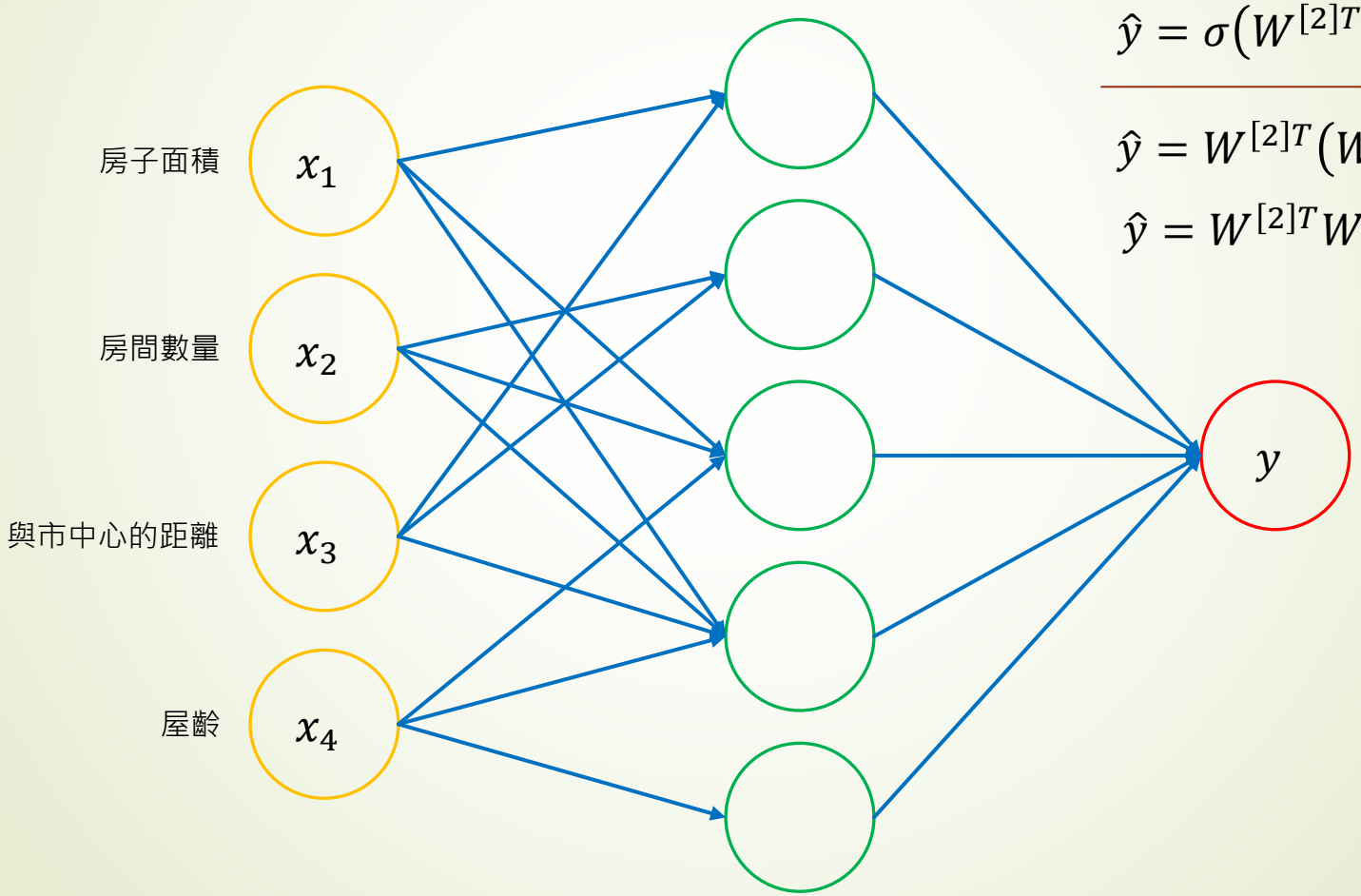


$$ReLU: a = g(z) = Max(0, z)$$

$$Leaky\ ReLU: a = g(z) = Max(0.01z, z)$$

https://www.sympy.org/en/index.html

How do Neural Networks work?



$$\hat{y} = \sigma\left(W^{[2]T}\sigma\left(W^{[1]T}X + b^{[1]}\right) + b^{[2]}\right)$$

$$\hat{y} = W^{[2]T}\left(W^{[1]T}X + b^{[1]}\right) + b^{[2]}$$

$$\hat{y} = W^{[2]T}W^{[1]T}X + W^{[2]T}b^{[1]} + b^{[2]}$$

已訓練完的網路，故無全連接

Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations (2009)

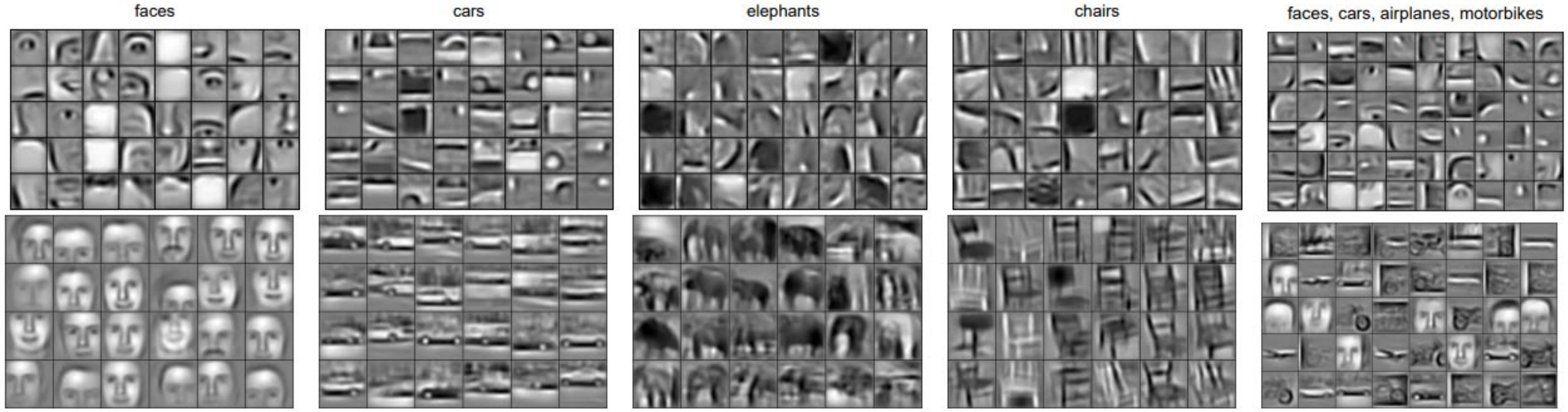
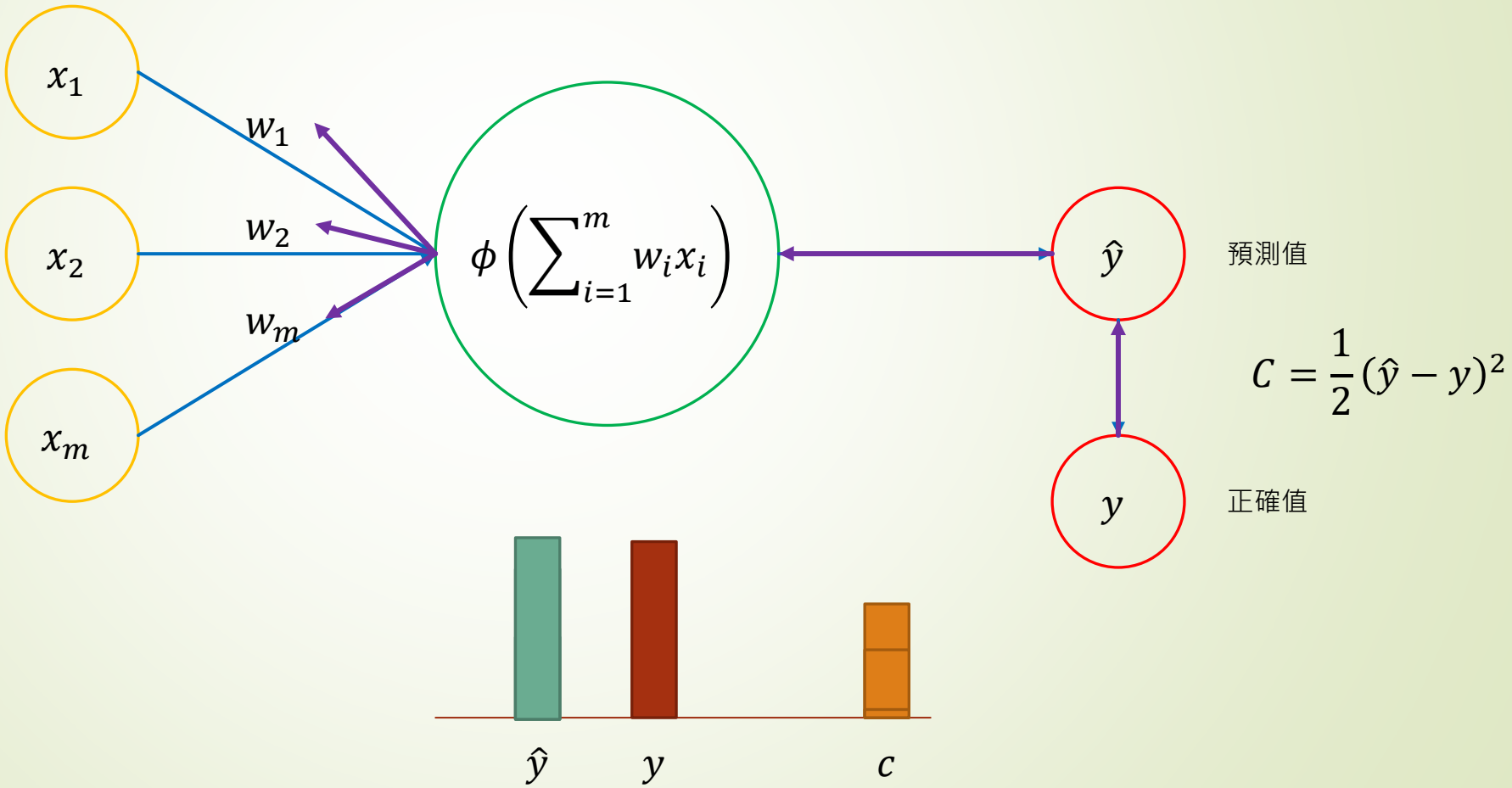


Figure 3. Columns 1-4: the second layer bases (top) and the third layer bases (bottom) learned from specific object categories. Column 5: the second layer bases (top) and the third layer bases (bottom) learned from a mixture of four object categories (faces, cars, airplanes, motorbikes).

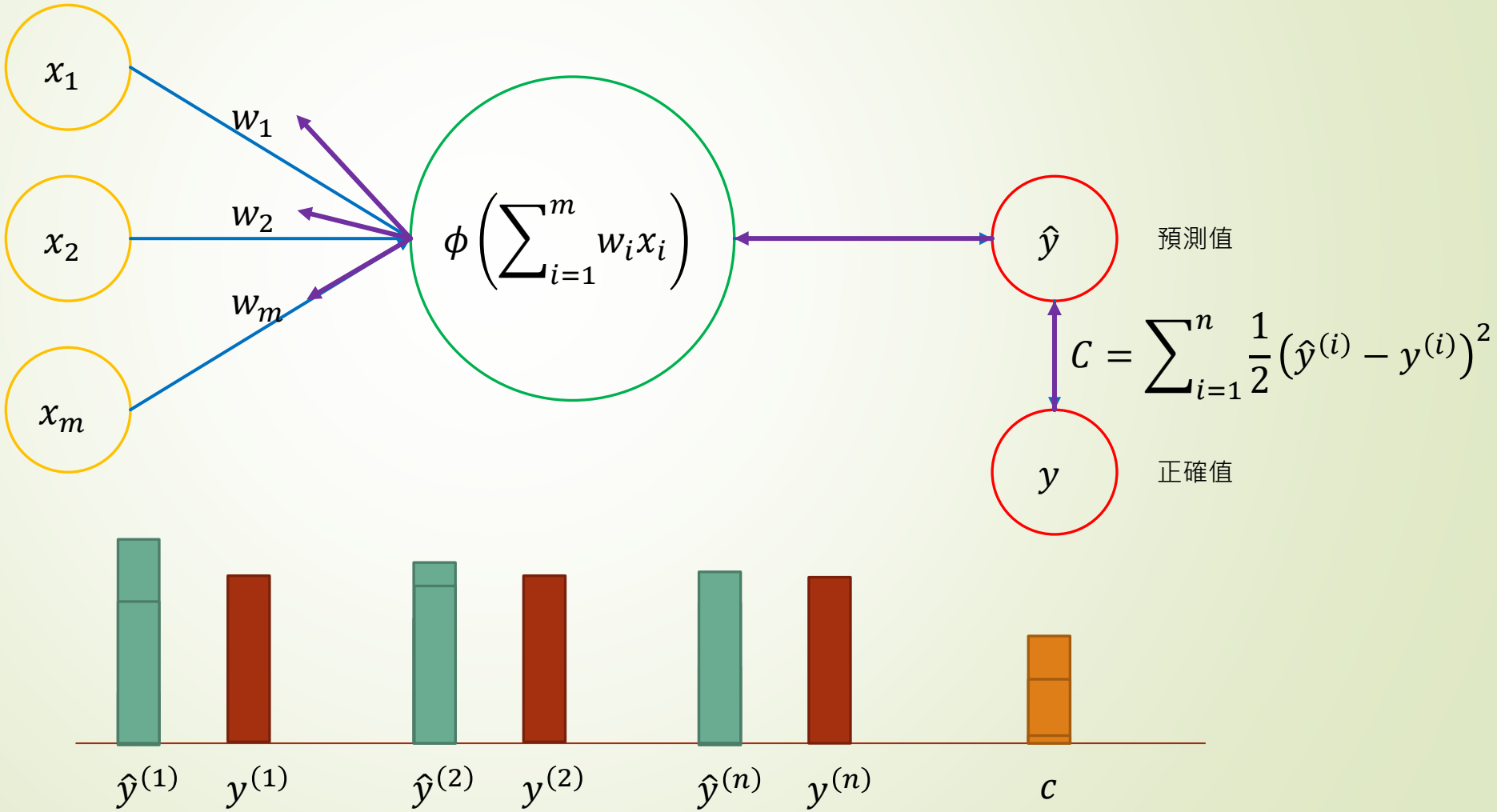
Output

How do Neural Networks learn?



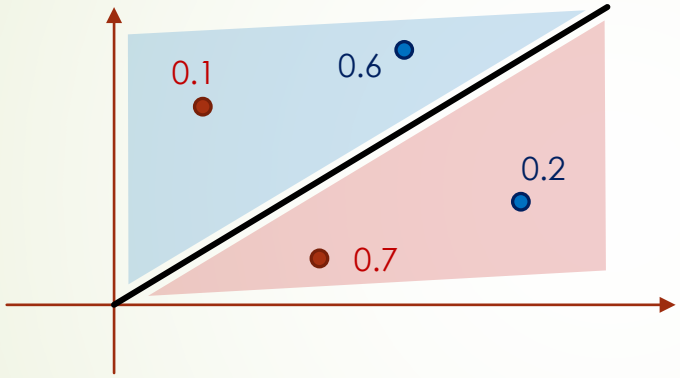
Output

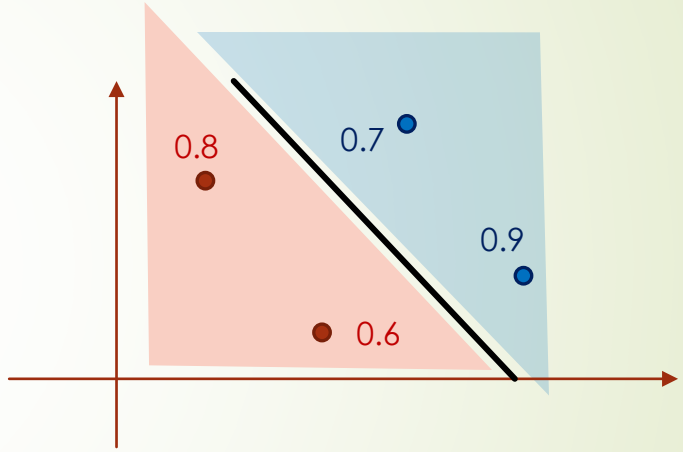
How do Neural Networks learn?



Output

Maximum Likelihood





$0.6 \times 0.2 \times 0.1 \times 0.7 = 0.0084$

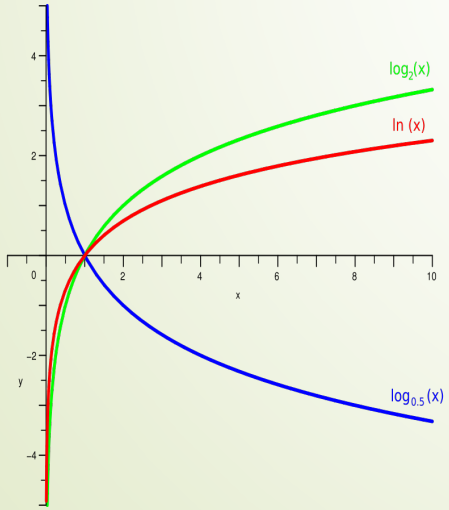
$0.7 \times 0.9 \times 0.8 \times 0.6 = 0.3024$

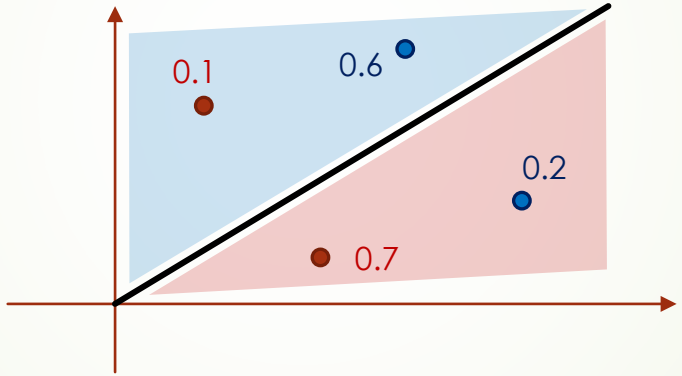
Output

Cross-Entropy

Goal: Minimize the Cross Entropy

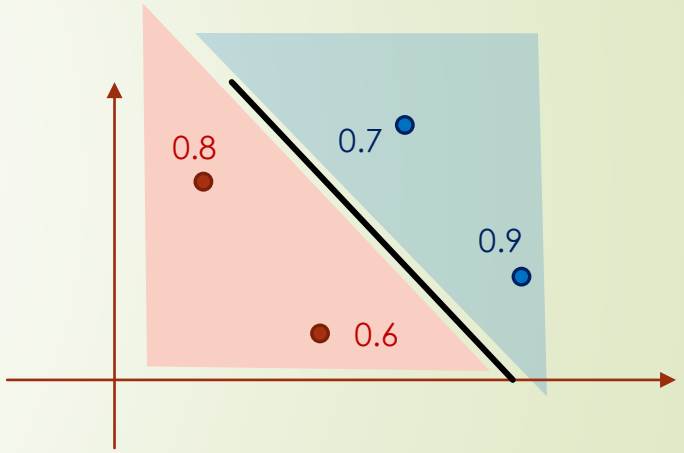
- What function turns products into sums?
  - Sin()
  - Cos()
  - Log()
  - Exp()





$0.6 \times 0.2 \times 0.1 \times 0.7 = 0.0084$

$-\ln(0.6) - \ln(0.2) - \ln(0.1) - \ln(0.7) = 4.8$



$0.7 \times 0.9 \times 0.8 \times 0.6 = 0.3024$

$-\ln(0.7) - \ln(0.9) - \ln(0.8) - \ln(0.6) = 1.2$

$$\text{Cross Entropy} = -\sum_{i=1}^{m} y^{(i)} \ln \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \ln\left(1 - \hat{y}^{(i)}\right)$$

Logistic Regression cost function

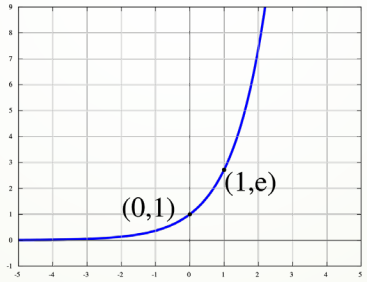
- Loss (error) function
  - $L(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$
    - If y=1, $L(\hat{y}, y) = -\log \hat{y}$, want $\hat{y}$ large
    - If y=0, $L(\hat{y}, y) = -\log(1 - \hat{y})$, want $\hat{y}$ small
- Cost function
  - $J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right]$
    - $\hat{y}^{(i)} = \sigma\left(w^T x^{(i)} + b\right)$

Output

The Softmax Function

- 二元分類
- 多標籤二元分類
- 多類別分類

- We want to classify cat, dog, and pig
  - If the prediction of cat is 1, dot is 0, and pig is -1, what should we do to calculate the probability?
  - What function turns every number into a positive number?
    - Sin()
    - Cos()
    - Log()
    - Exp()



$$\text{Cat} => \frac{e^1}{e^1+e^0+e^{-1}} = 0.67$$

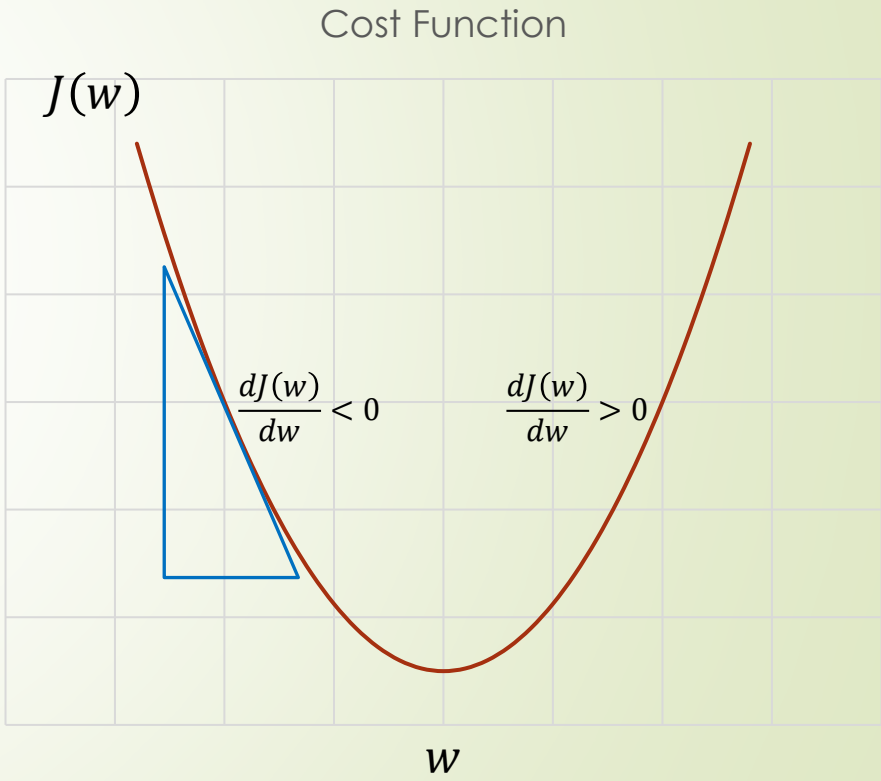
$$\text{Dot} => \frac{e^0}{e^1+e^0+e^{-1}} = 0.24$$

$$\text{Pig} => \frac{e^{-1}}{e^1+e^0+e^{-1}} = 0.09$$

- Softmax Function
  -

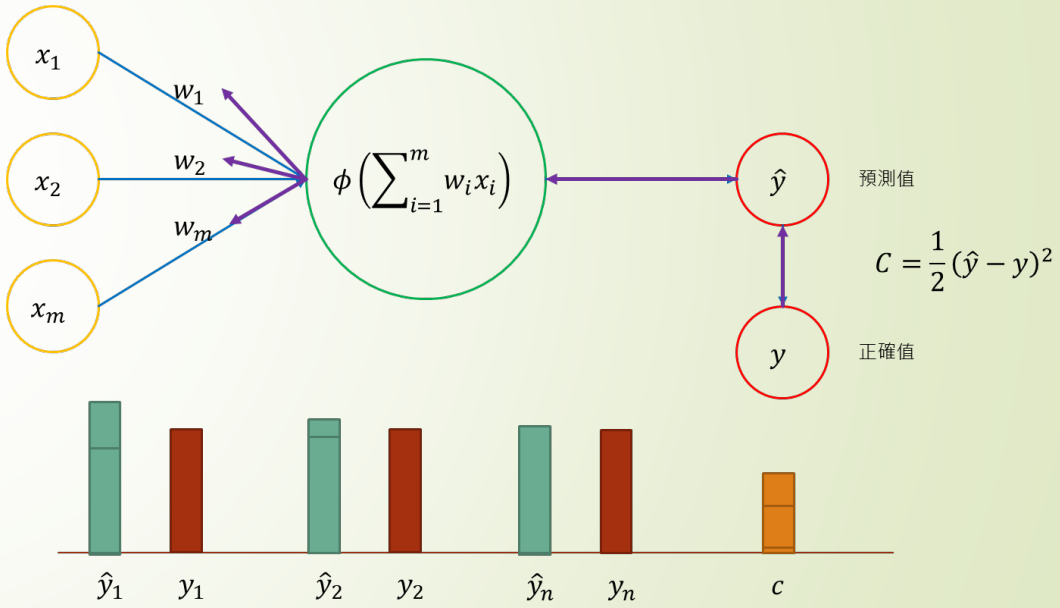
Gradient Descent

- Want to find w, b that minimize J(w, b)
  - $w := w - \alpha \frac{\partial J(w,b)}{\partial w}$
    - $\frac{\partial J(w,b)}{\partial w}$ is the slope of J(w, b) on w direction
  - $b := b - \alpha \frac{\partial J(w,b)}{\partial b}$
    - $\frac{\partial J(w,b)}{\partial b}$ is the slope of J(w, b) on b direction
  - α: learning rate

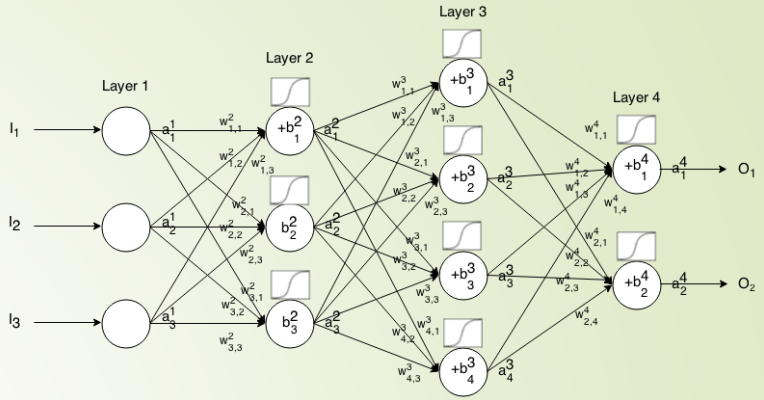


Stochastic Gradient Descent

- (Batch) Gradient Descent
  - $w^{(t+1)} = w^{(t)} - \alpha \nabla f\left(w^{(t)}\right)$
  - Local Minimum
- Stochastic Gradient Descent
  - Large Vibration -> Global Minimum
  - Including Random
  - Higher Speed
- Mini-batch Gradient Descent
  - Mixed
- Momentum
  - $w^{(t+1)} = w^{(t)} - v^{(t)}$
  - $v^{(t)} = mv^{(t-1)} + \alpha \nabla f\left(w^{(t)}\right)$ when $t \geq 1$



Training the ANN



1. 隨機初始化權重，使得權重接近0但不等於0
2. 將第一個觀察數據輸入“輸入層”，每個自變量特徵輸入一個神經元
3. 正向傳播：神經網路從左至右進行計算，每個神經元得到來自上一層神經元的輸入與權重的運算
4. 誤差函數：計算預測值與實際值的差異 (**loss function**)
5. 反向傳播：神經網路從右至左進行計算，依據誤差函數相對於權重的梯度，對每個權重進行更新。學習速率與梯度將決定更新的速度 (**optimizer**)
6. 對每一組新的觀察數據，重複step 1到step 5 (**batch size**)
7. 當所有數據都輸入神經網路後，稱之為一期(**epoch**)的訓練

卷積神經網絡

About CNN

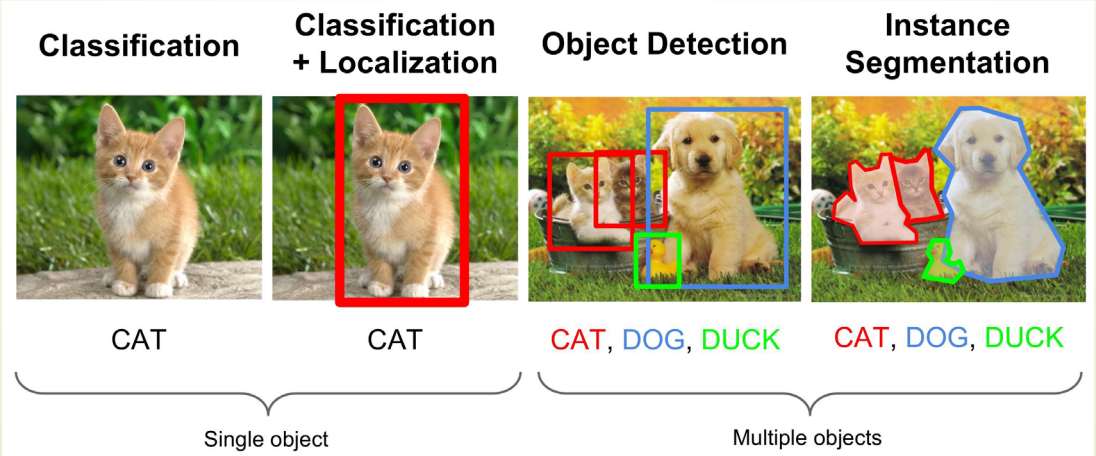
- What are Convolutional Neural Networks?
- Step 1: Convolution Operation
  - Step 1.1: ReLU Layer
- Step 2: Pooling
- Step 3:Flattening
- Step 4: Full Connection
- Summary

應用

- 應用領域
  - 電腦視覺
  - 語音辨識
  - 自然語言處理
  - 音訊辨識
  - 生物資訊學等領域
- 影片
  - 駕駛：https://vimeo.com/192179727
  - 倉儲：https://www.youtube.com/watch?v=rKHFPsA8JjM
  - 機器人：https://www.youtube.com/watch?v=rVlhMGQgDkY&start_radio=1&list=RDQMaBMndpuioWw

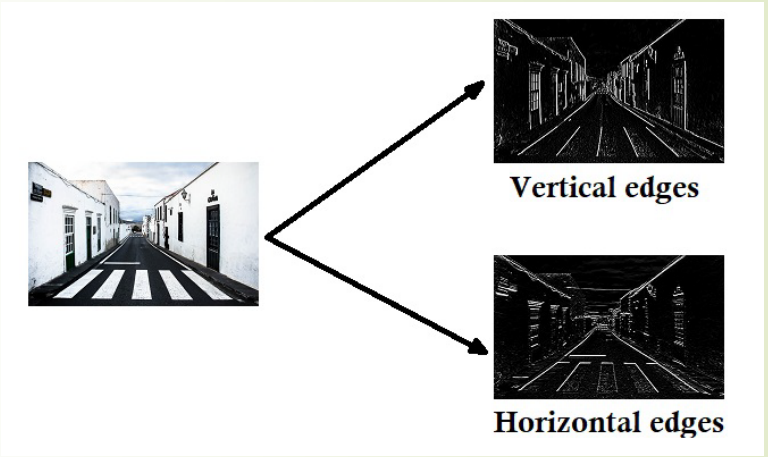
Computer Vision

- 圖像分類(image classification)
  - 識別圖像中存在的內容;
- 物體識別和檢測(object recognition and detection)
  - 識別圖像中存在的內容和位置（通過邊界框）；
- 語義分割(semantic segmentation)
  - 識別圖像中存在的內容以及位置（通過查找屬於它的所有像素）



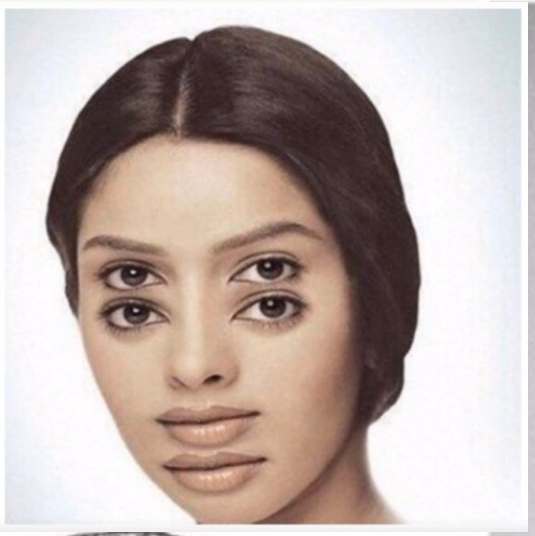
Why Convolution

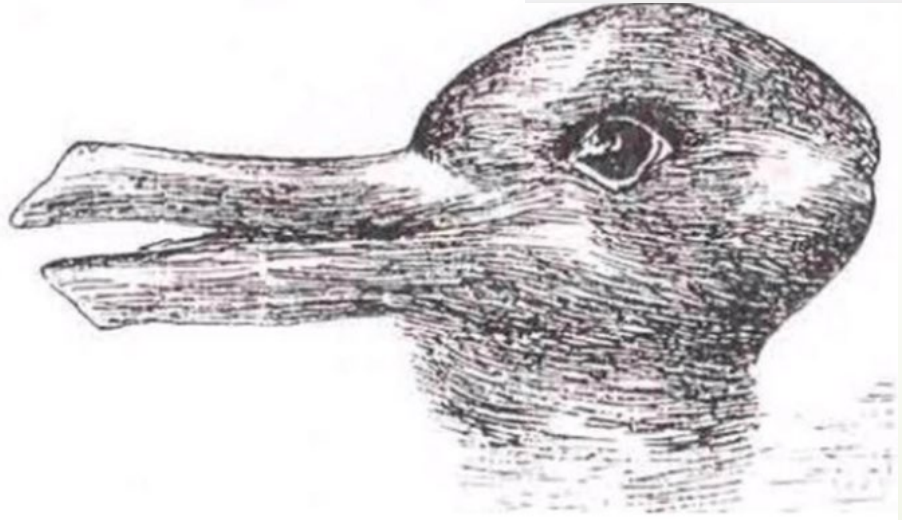
- Reduce the dimension of input
  - If the image is 1,000 x 1,000
  - If the neuron of hidden layer is 1,000
  - The parameters is around 3 billion
- Pick the feature of input

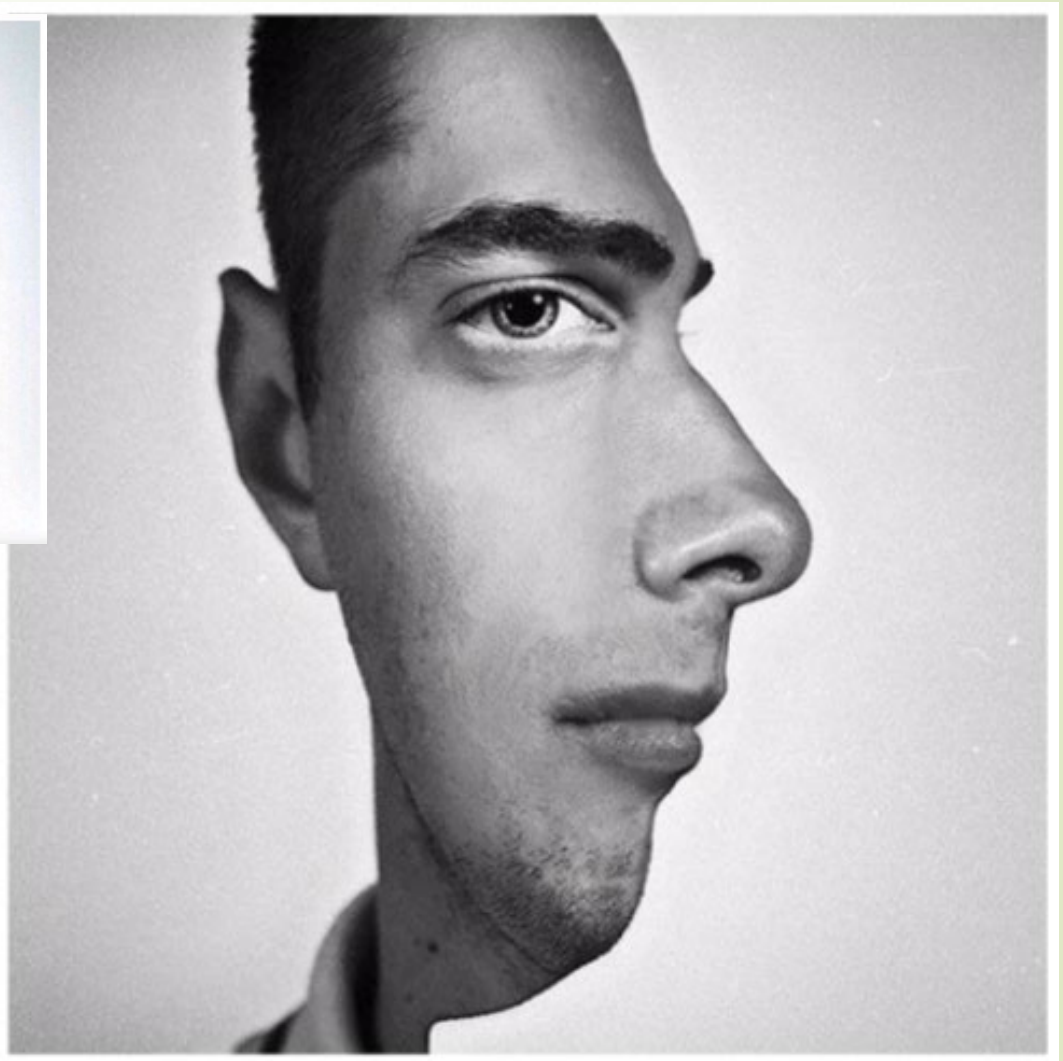


datahacker.rs/edge-detection/

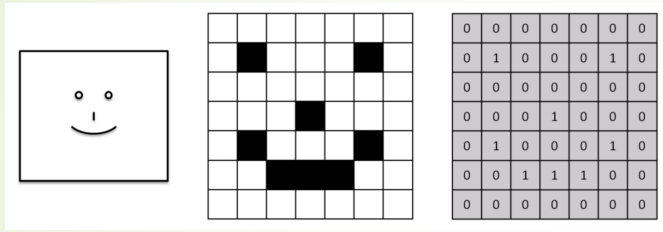
尋找特徵

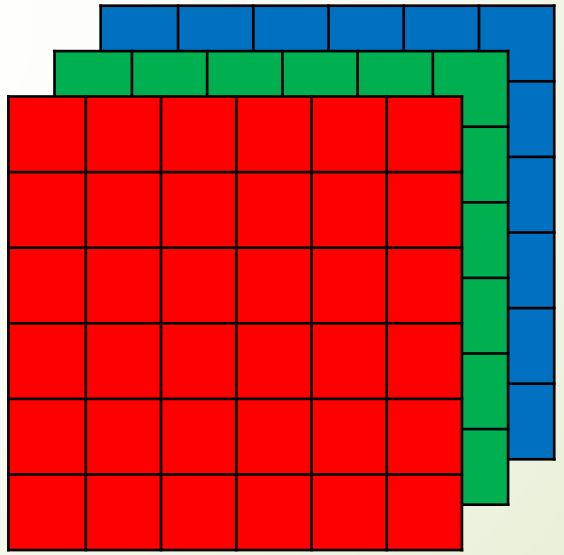






圖片數字化





Ex. 64 x 64 x **3**,
$n^x = 64 \times 64 \times 3 = 12{,}288$

卷積神經網路 (Convolutional Neural Networks)

- 卷積 (Convolution)
  - 線性整流層 (ReLU Layer)
- 最大池化 (Max Pooling)
- 扁平化 (Flattening)
- 全連接 (Full Connection)

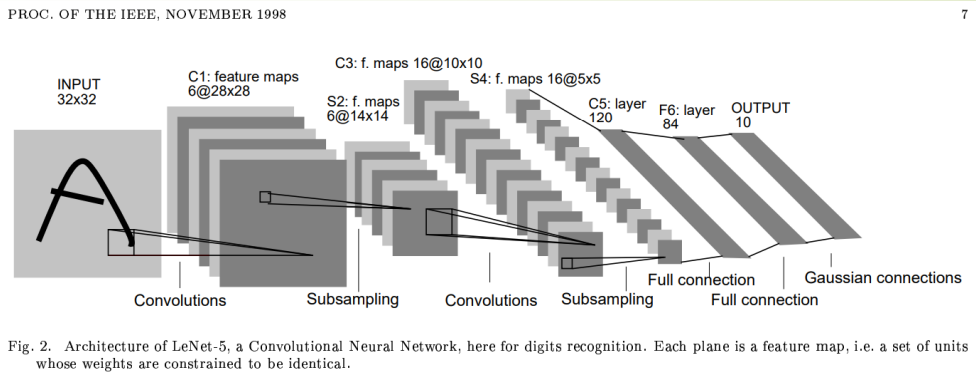


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

1. 卷積

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau) d\tau$$

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

輸入圖片

0	0	1
1	0	0
0	1	1

過濾器

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

特徵圖

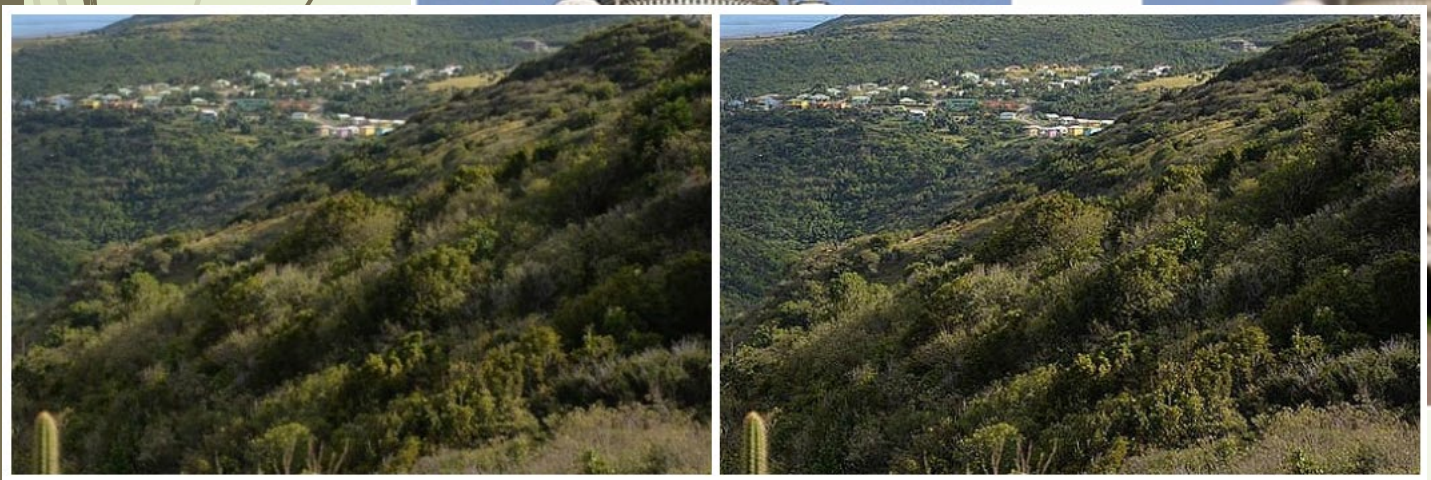
Size of Filter
Step of Filter

1. 卷積的功能
2. 如何決定filter的數值？

濾鏡

0	-1	0
-1	5	-1
0	-1	0

尖銳化
Sharpen



1	1	1
1	1	1
1	1	1

模糊化
Blur

0	1	0
1	-4	1
0	1	0

邊緣凸顯
Edge Detect

Padding

- Valid convolutions: no padding
  - Shrink output
    - Input (n)
    - Filter (f)
    - Output = $n - f + 1$
  - Throw array info from edge
- Same convolutions: Pad so that output size is the same as the input size
  - Padding (P)
  - Output = $n + 2p - f + 1$
    - $p = \frac{f-1}{2}$
    - f is usually odd

0	0	0	0	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	0	0	0	10	10	10	0
0	0	0	0	10	10	10	0
0	0	0	0	10	10	10	0
0	0	0	0	0	0	0	0

Stride

- Input (n=6)
- Padding (p=1)
- Filter (f=3)
- Stride (s=2)
- Output = $\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor = \left\lfloor \frac{6+2-3}{2} + 1 \right\rfloor = 3$
  - $\lfloor z \rfloor = floor(z)$

PS. Cross-correlation vs. Convolution

3	4	5
1	0	2
-1	9	7

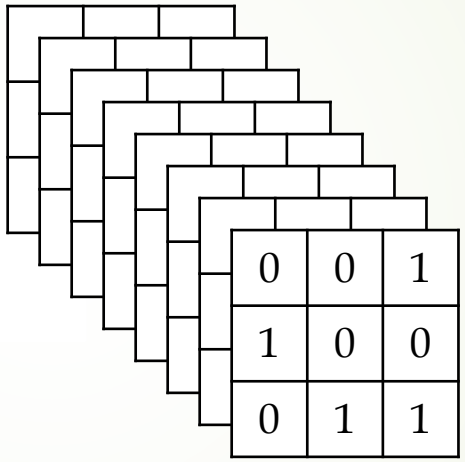
7	9	-1
2	0	1
5	4	3

0	0	0	0	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	10	10	10	0	0	0	0
0	0	0	0	0	0	0	0

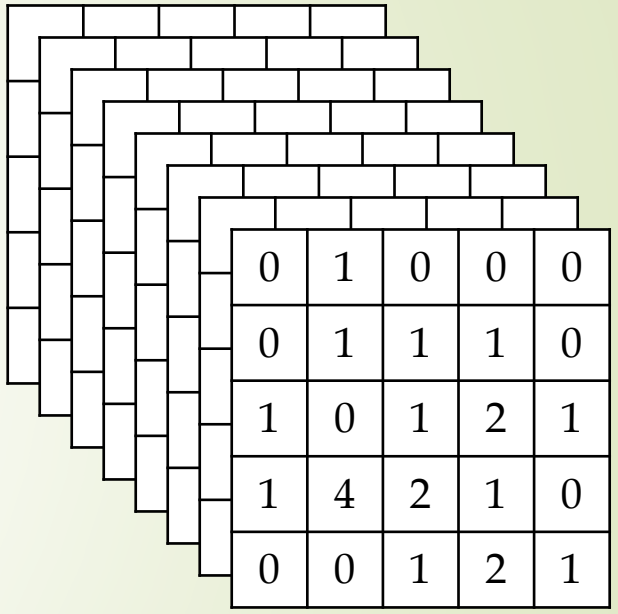
卷積層

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

輸入圖片

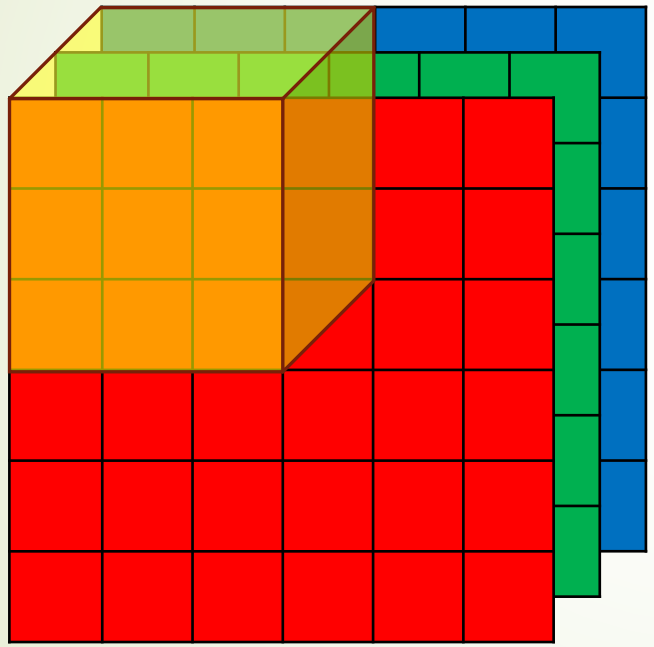


過濾器



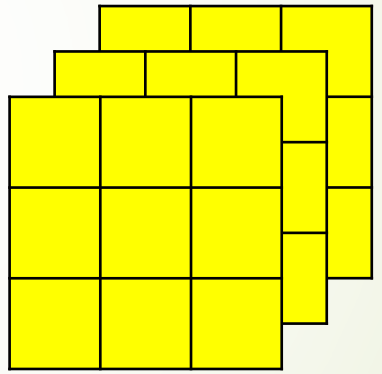
特徵圖

Convolutions on RGB image



6 x 6 x **3**

*



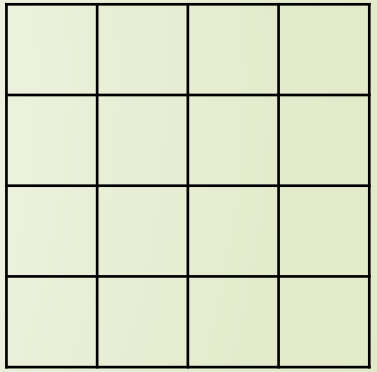
Width

Height

Channels

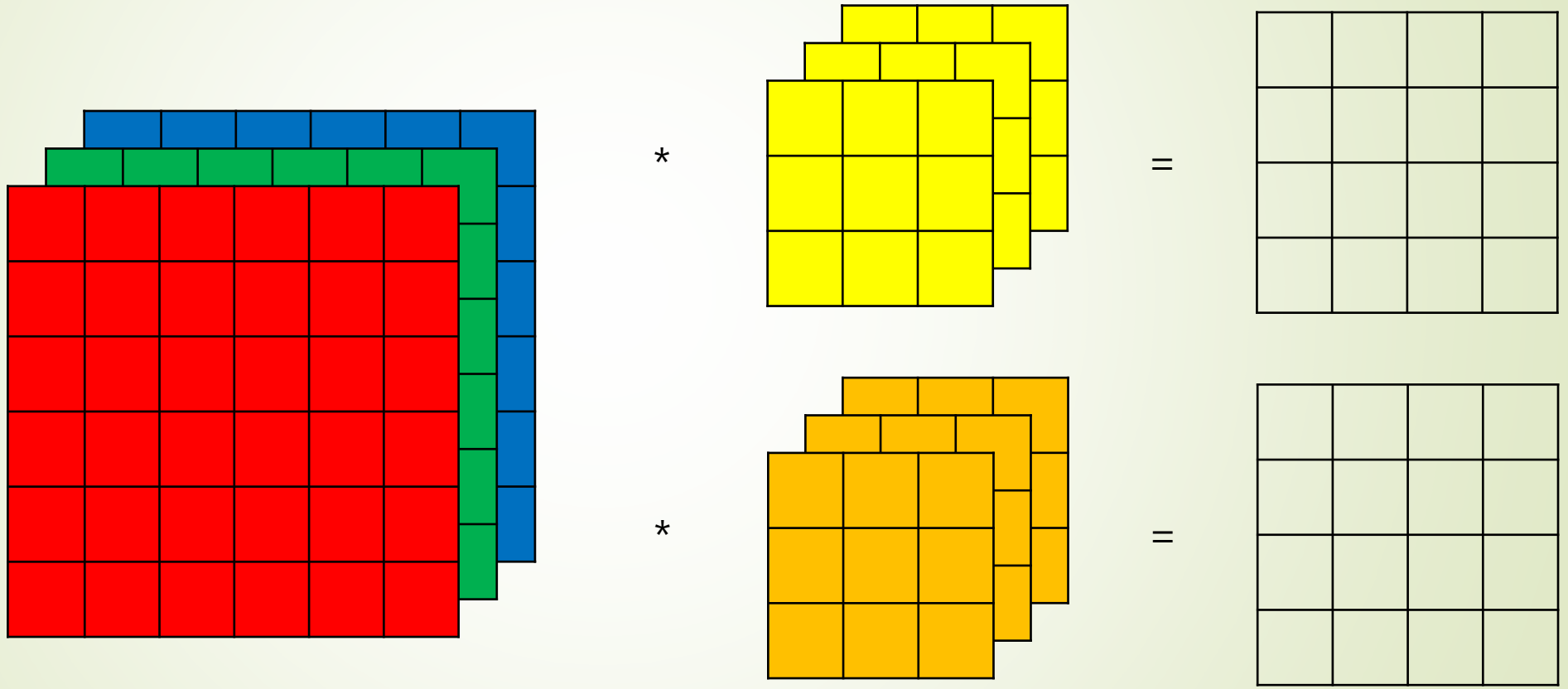
3 x 3 x **3**

=



4 x 4 x 1

Multiple Filters

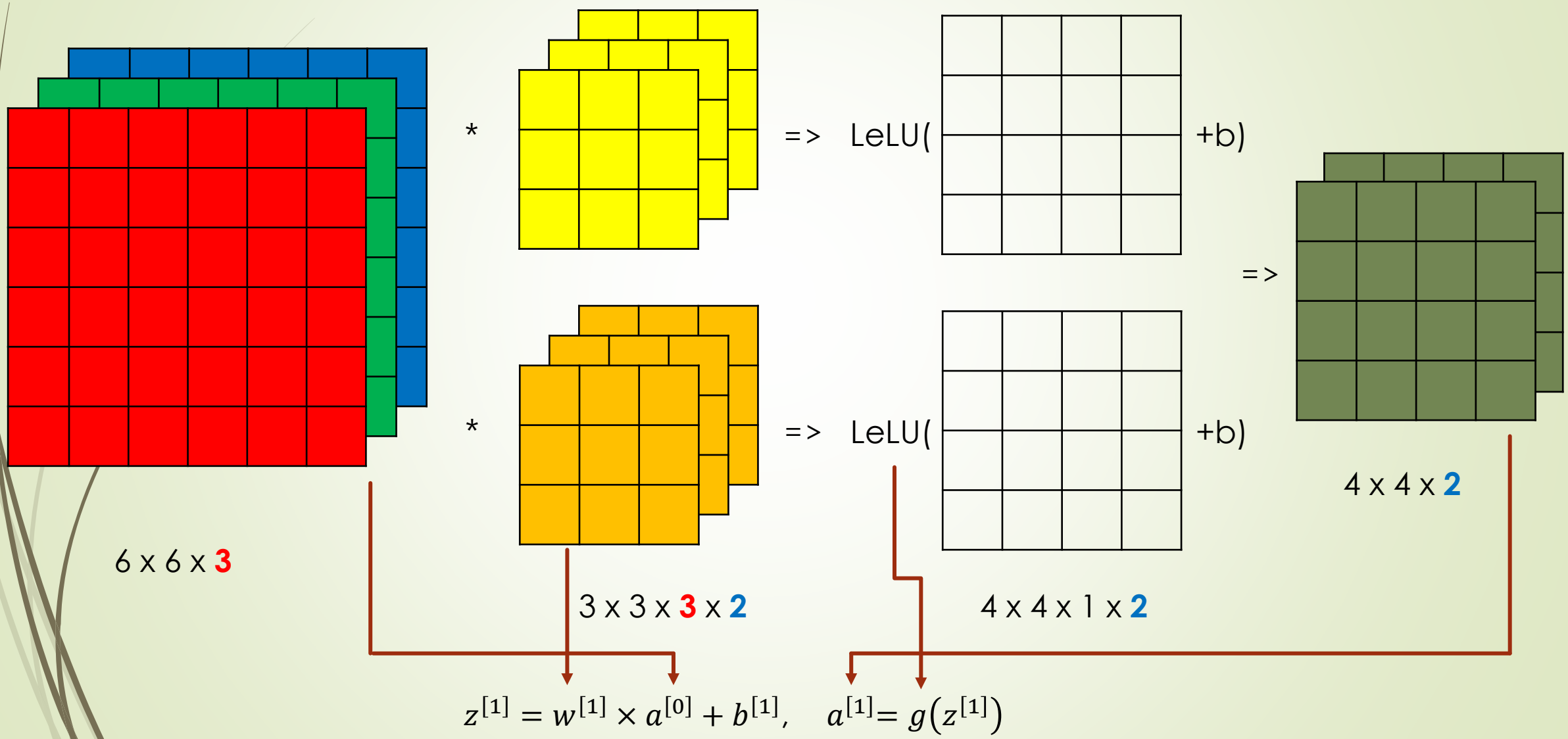


6 x 6 x **3**

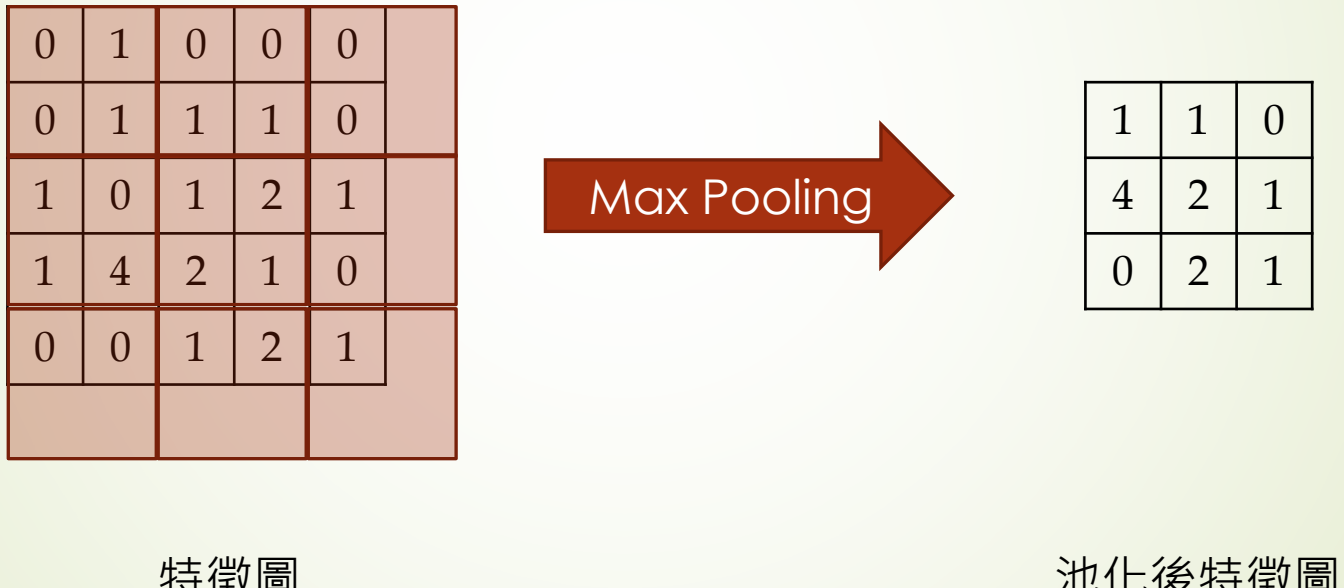
3 x 3 x **3** x **2**

4 x 4 x 1 x **2**

Example of a Layer



2. 最大池化



特徵圖

池化後特徵圖

為何池化？

1. 特徵的相對位置，非絕對位置
2. 降低維度，可避免過度擬合

5獵豹捕獵圍攻羚羊完食後血面具怵目驚心- ...
chinatimes.com

Our Planet】獵豹生擒角馬超震撼令人類置身弱肉強...
hk01.com

猎豹- 维基百科，自由的百科全书
zh.wikipedia.org

成年男子能不能單殺非洲獵豹- 每日頭條
kknews.cc

獵豹跑得快！快不過絕種的速度| DQ 地球圖...
dq.yam.com

地表最快的獵豹逝世－－－同欣賞牠奔馳的...
natgeomedia.com

獵豹，花豹，美洲豹，它們到底有什麼區別...
kknews.cc

你聽過獵豹的叫聲嗎？這反差也太大了！ * 阿波...
tw.aboluowang.com

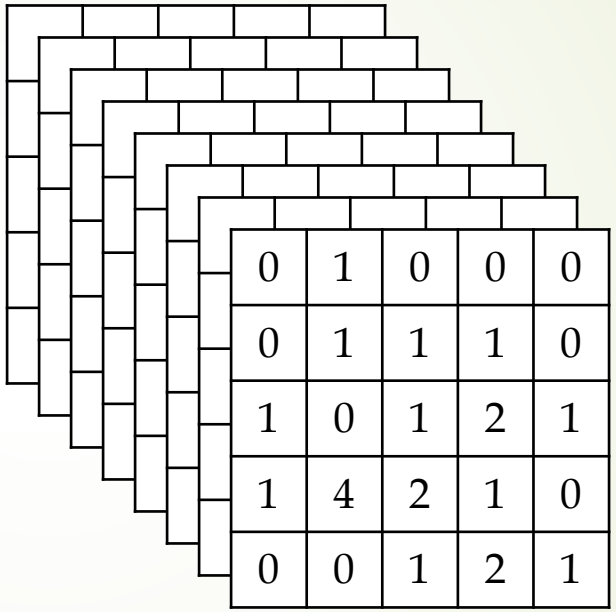
哪隻獵豹不偷腥？ - 國家地理雜誌中文網
natgeomedia.com

獵豹寶寶太害羞！園方派出「療癒犬拉拉」...
lookerpets.com

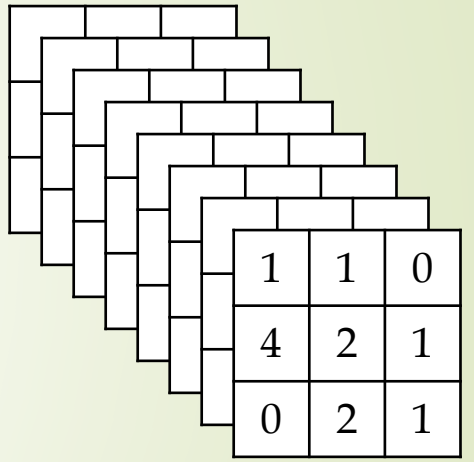
池化層

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

輸入圖片

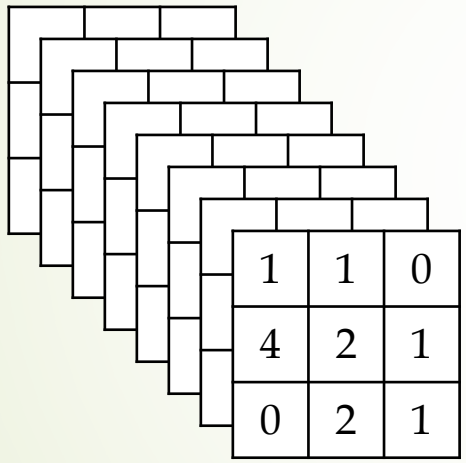


卷積層



池化層

3. 扁平化

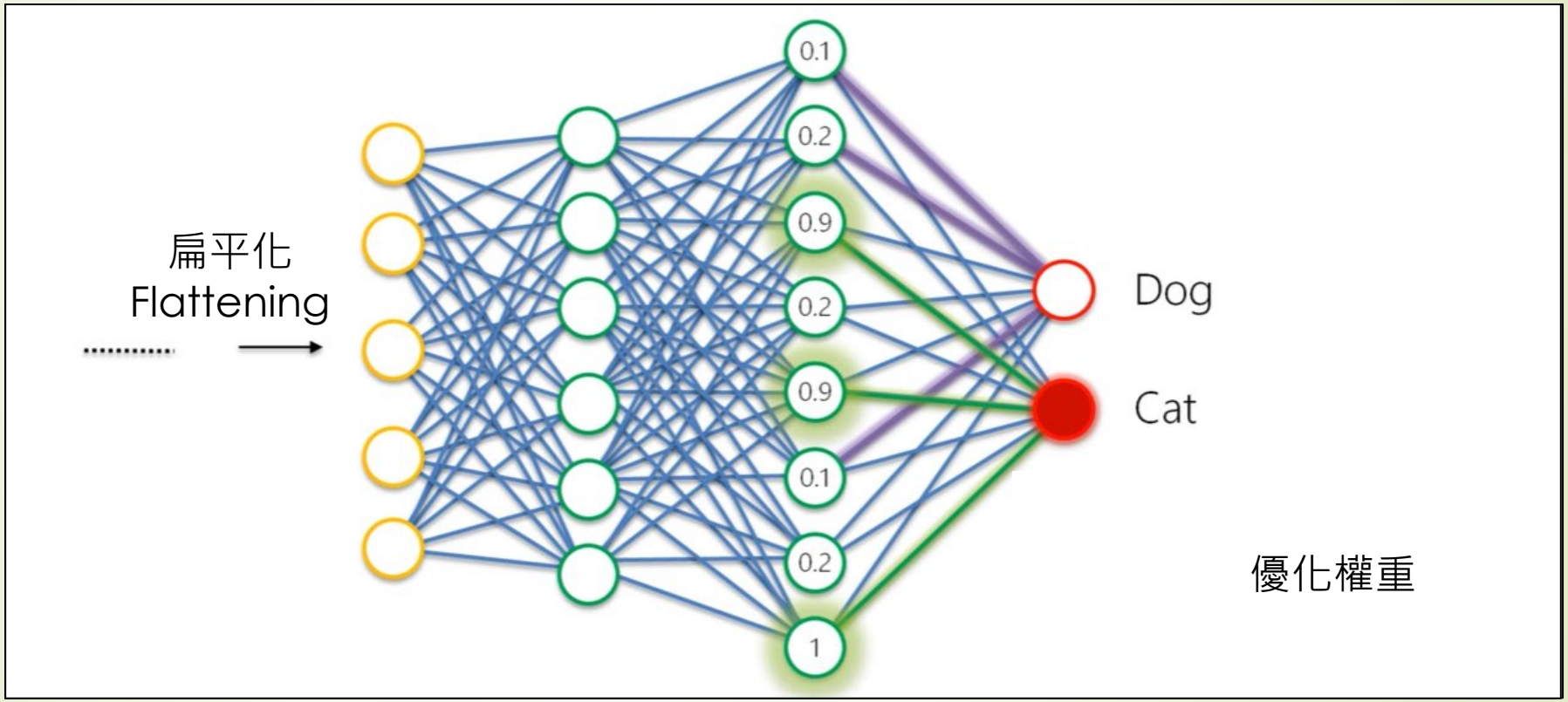


池化層

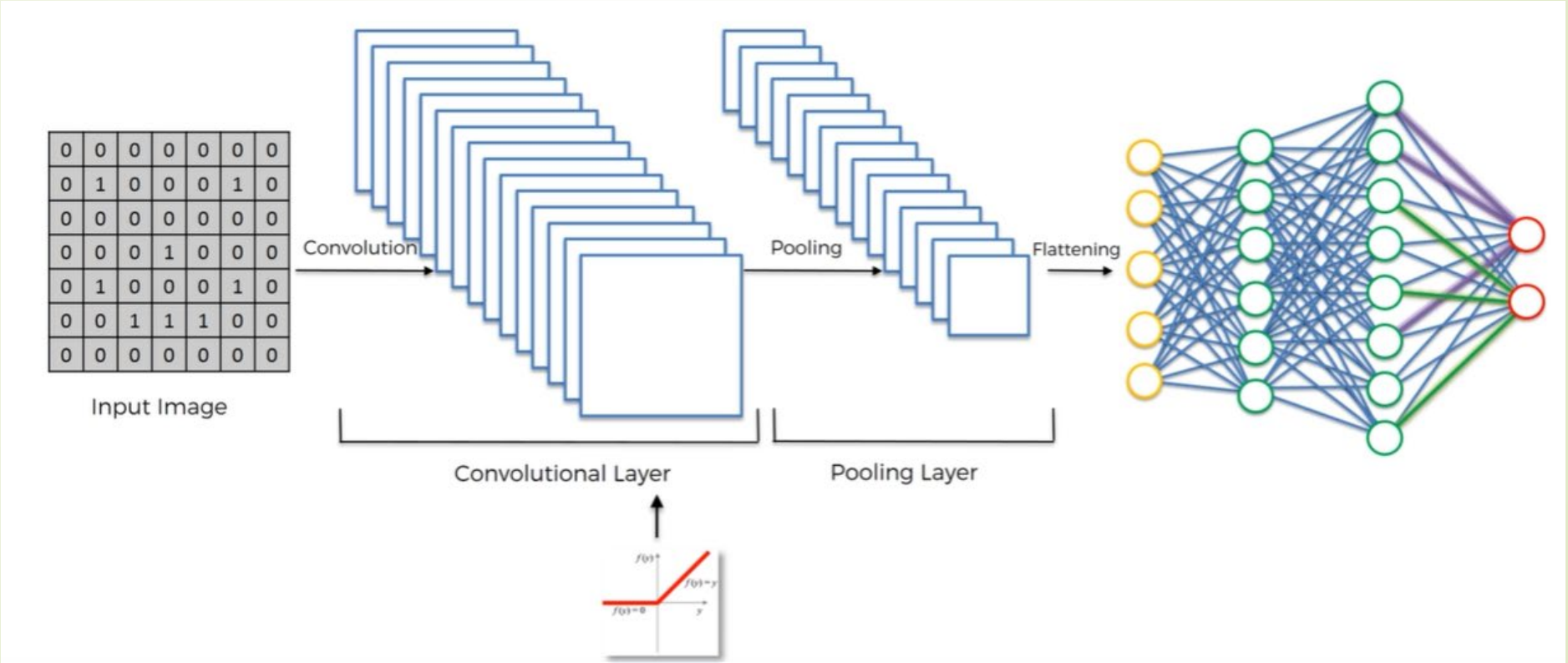
Flattening

ANN 輸入層

4. 全連接

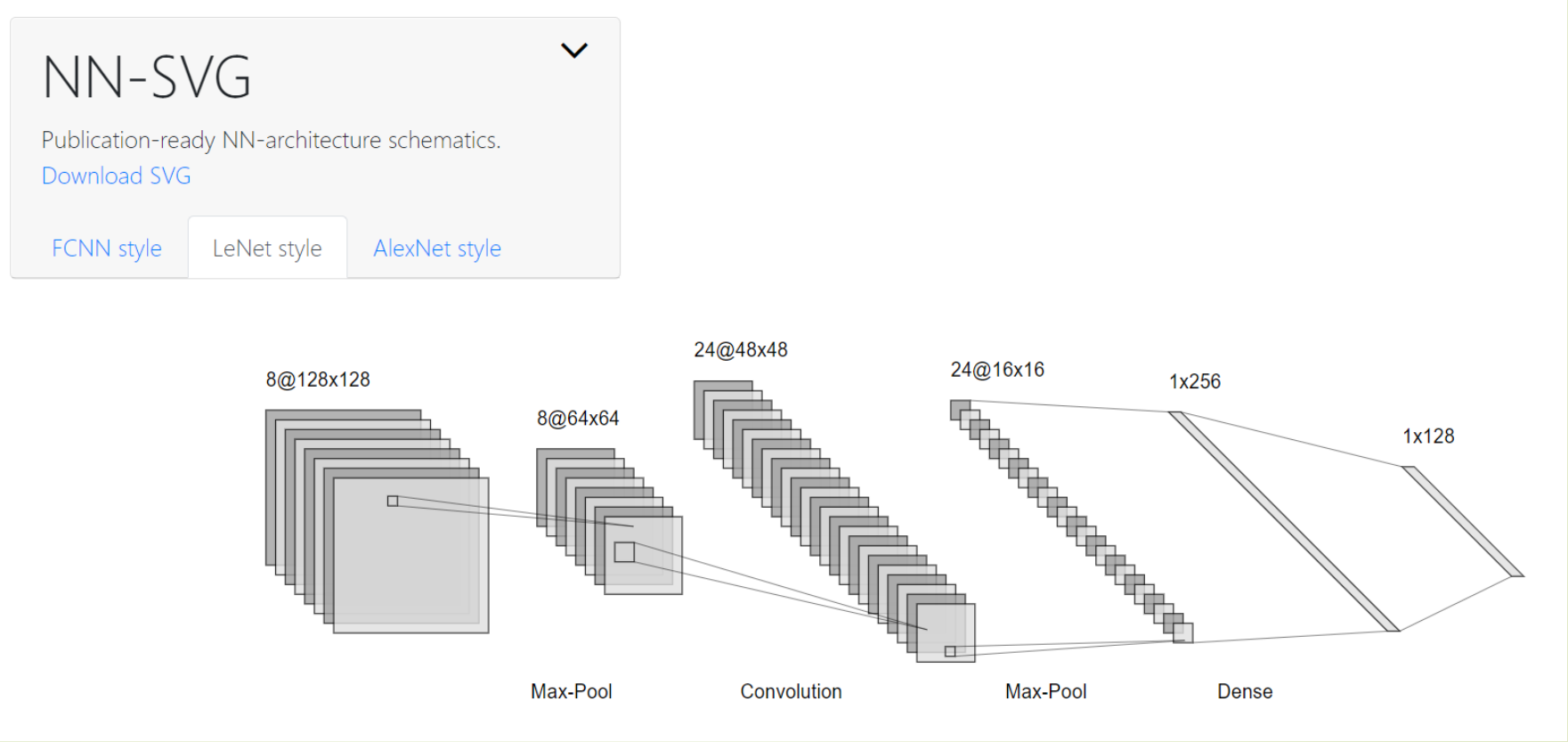


整體架構

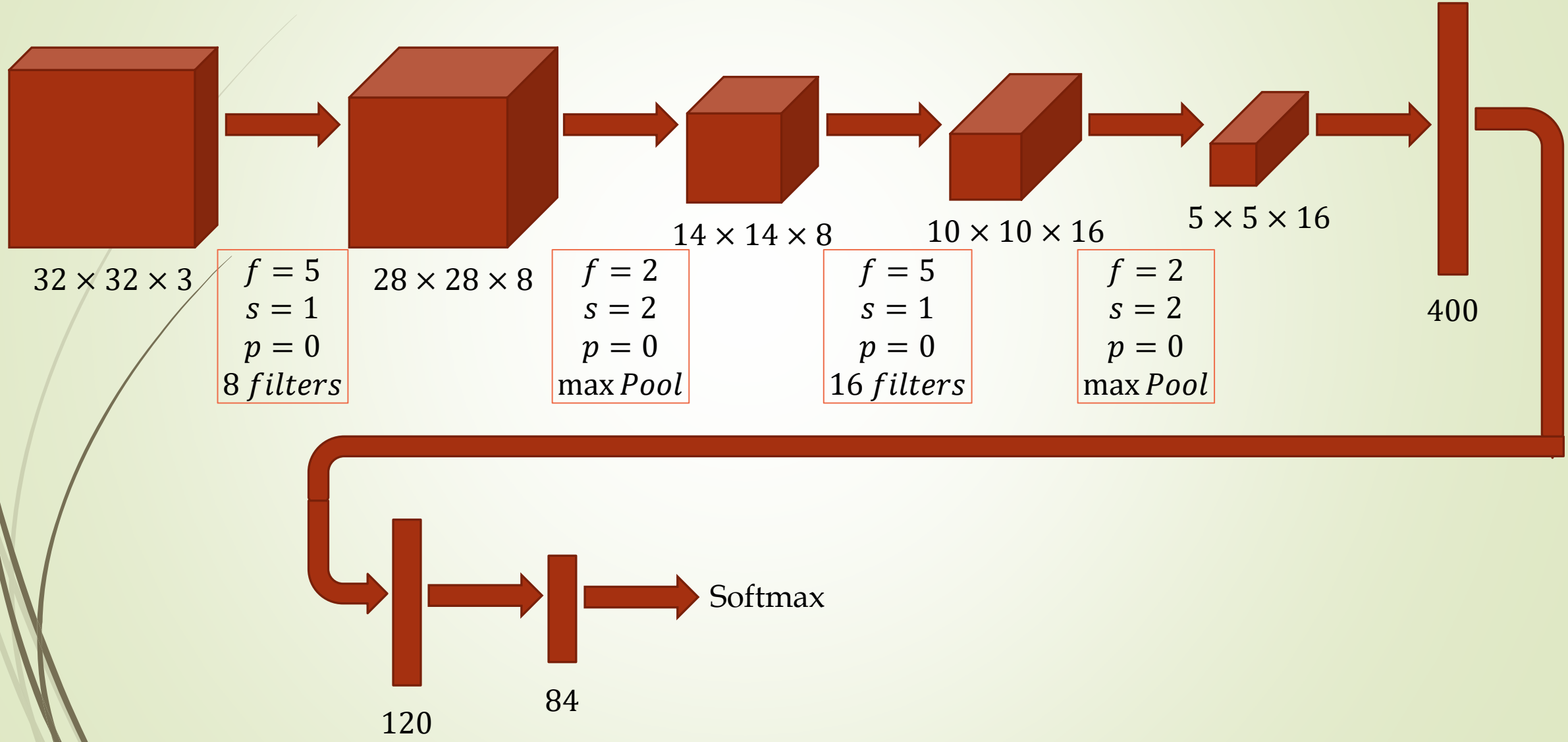

0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
Input Image
Convolution
Pooling
Flattening
Convolutional Layer
Pooling Layer

A Tool for Publication

- http://alexlenail.me/NN-SVG/LeNet.html



Neural Network Example (~LeNet-5)



Neural Network Example

	Activation Shape	Activation Size	Parameters Shape	Parameters Size
Input	(32, 32, 3)	3,072		0
CONV 1(f=5, s=1)	(28, 28, 8)	6,272	(5, 5, 3) x 8	608
POOL1	(14, 14, 8)	1,568		0
CONV2 (f=5, s=1)	(10, 10, 16)	1,600	(5, 5, 8) x 16	3,216
POOL2	(5, 5, 16)	400		0
FC3	(120, 1)	120	(120, 400)	48,120
FC4	(84, 1)	84	(84, 120)	10,164
Softmax	(10, 1)	10	(10, 84)	850

感謝聆聽