AI 新手村攻略

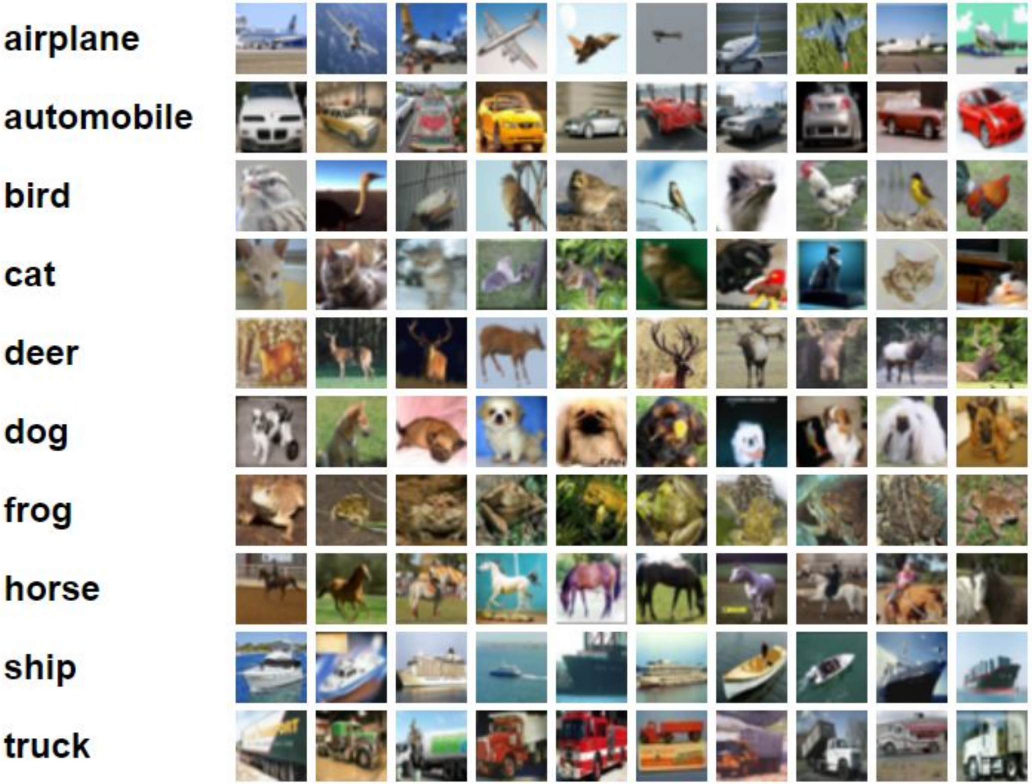
探險技能--影像視覺

隨堂複習

DNN with cifar10

Cifar10 ？哪位？

cifar10



這個 Alex 也是大有來頭的

- 由 Alex Krizhevsky, Geoffrey Hinton 收集的
- 有 10 個類別，每個類別有 6,000 張，總共有 60,000 張影像

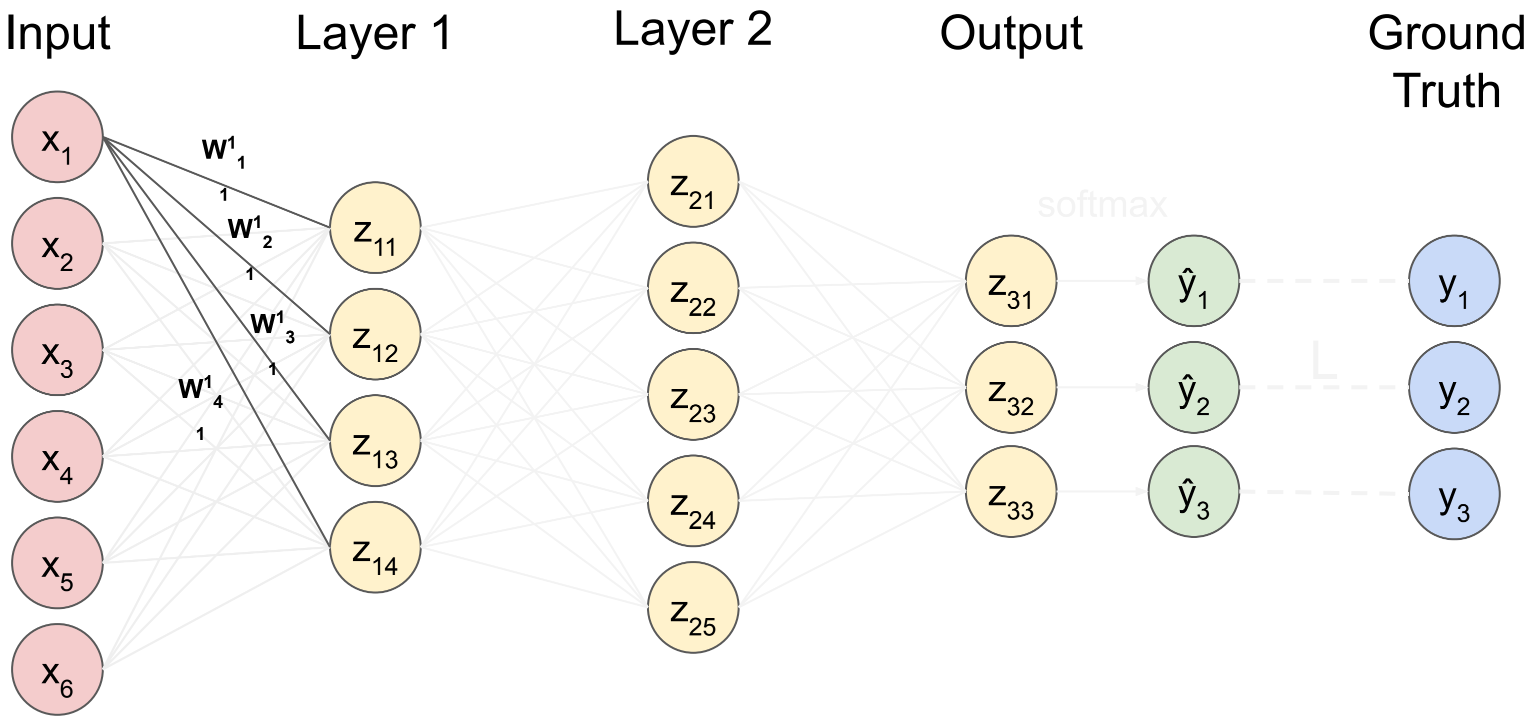
train 一發吧!!!

不好 train 齁~~

Convolutional Neural Network (CNN)

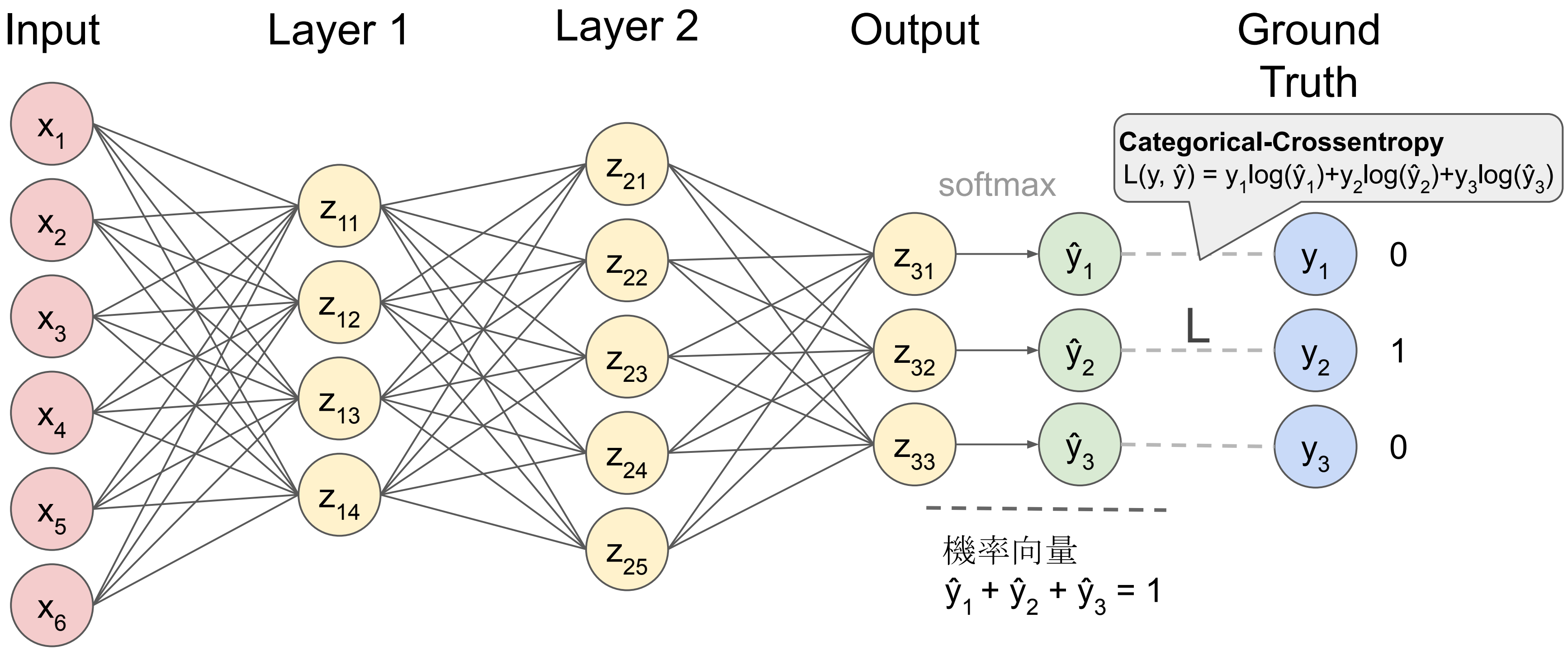
視覺神經網絡

複習一下 DNN

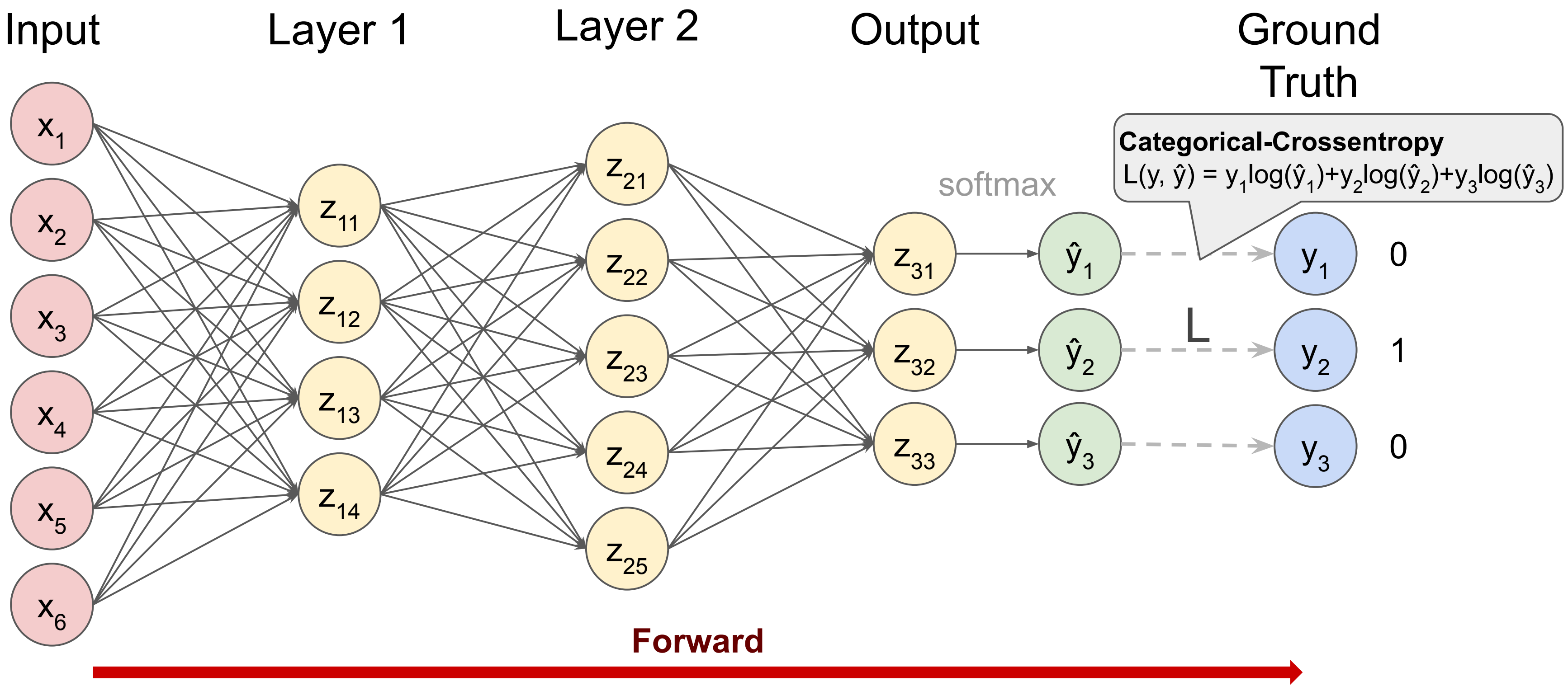


Input
Layer 1
Layer 2
Output
Ground Truth
softmax
L
x1
x2
x3
x4
x5
x6
w1 1 1
w1 2 1
w1 3 1
w1 4 1
z11
z12
z13
z14
z21
z22
z23
z24
z25
z31
z32
z33
ŷ1
ŷ2
ŷ3
y1
y2
y3

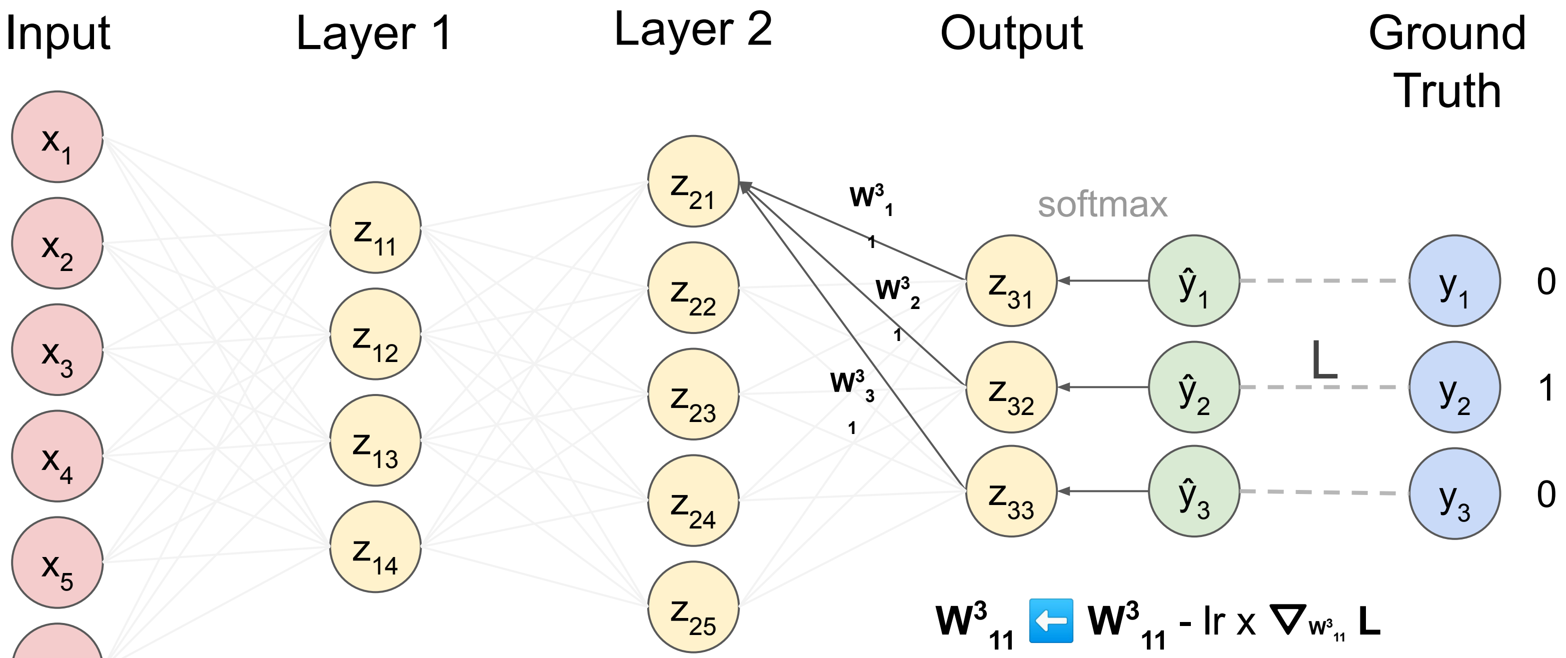
複習一下 DNN


Input
Layer 1
Layer 2
Output
Ground Truth
x_1 x_2 x_3 x_4 x_5 x_6
z_{11} z_{12} z_{13} z_{14}
z_{21} z_{22} z_{23} z_{24} z_{25}
z_{31} z_{32} z_{33}
softmax
$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$
Categorical-Crossentropy
$L(y, \hat{y}) = y_1\log(\hat{y}_1)+y_2\log(\hat{y}_2)+y_3\log(\hat{y}_3)$
L
y_1 0
y_2 1
y_3 0
機率向量
$\hat{y}_1 + \hat{y}_2 + \hat{y}_3 = 1$

Forward 計算 - 第一個人拿到訊息, 往後傳, 傳到最後對答案



Backward 計算 - 最後一個人對完答案，請前面的人修正訊息



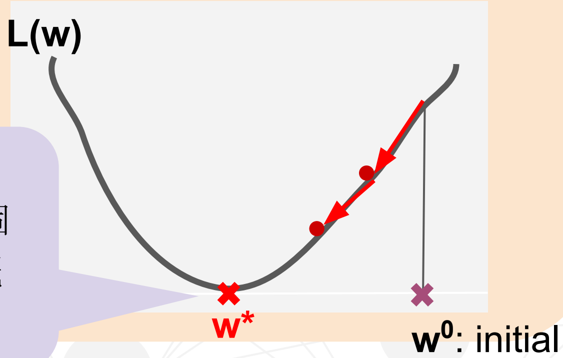
$$W^3_{11} \Leftarrow W^3_{11} - lr \times \nabla_{w^3_{11}} L$$

終極目標

更新 w，找到可以使得答對最多的傳遞過程

Backward 計算 - Gradient Descent

因為....
L 之於 **w** 就像是....凹曲面
目標找到最好的 **w*** 使得 loss 最小



Q: 怎麼找到最低點 w*？
A: 沿著""**切線**""方向找，就不會錯

切線就要先算，切線斜率，要算 Gradient

Q: 找到 **w*** 很難嗎??
A: 實際上的 **L(w)** 是一個很複雜的抽象曲面，大概長.....

Q: 沿著切線走，那一步要走多遠 ??
A: 設定 lr 決定，一步距離

$$W^3_{11} \leftarrow W^3_{11} - lr \times \nabla_{W^3_{11}} L$$

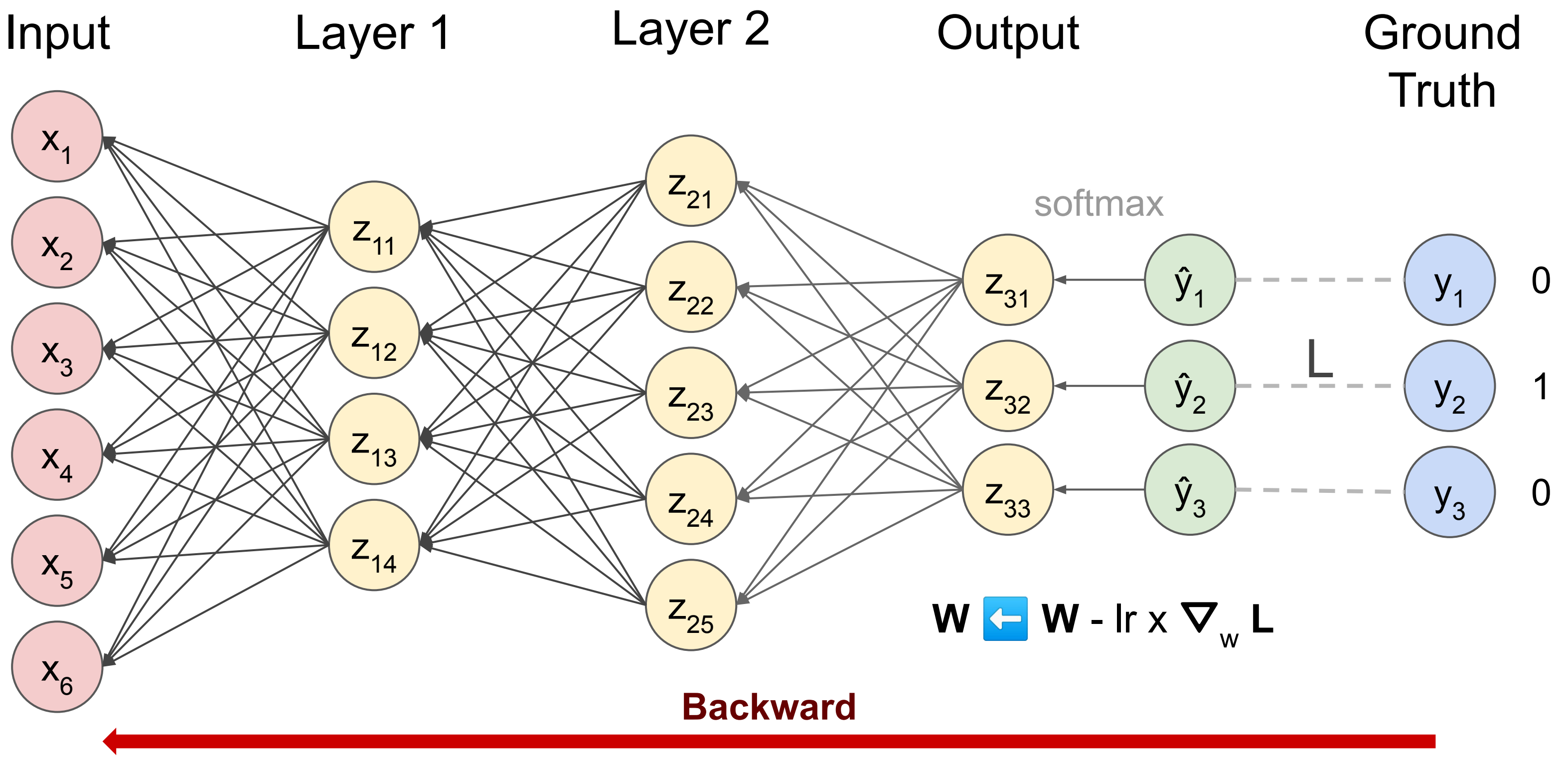
加分密技請看
Appendix A

為什麼更新的數學式長這樣？

終極目標

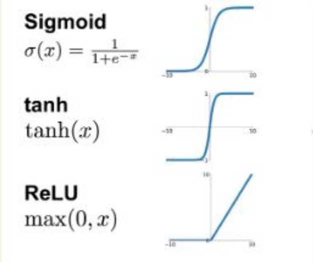
更新 w，找到可以使得答對最多的傳遞過程

Backward 計算 - 最後一個人對完答案，請前面的人修正訊息



Backward 計算 - Backward + Chain Rule = Backpropagation

Activation Function 基本款



$\sigma'(x) = \sigma(x)(1 - \sigma(x))$

$\sigma'(x) = 1 - \sigma(x)^2$

$\sigma'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$

叫什麼 Chain Rule 老掉牙了~~
現在要叫 Backpropagation

當然是 Chain Rule 呀!!!

$$\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{\partial L(\sigma(\boldsymbol{W}x + \boldsymbol{b}))}{\partial \boldsymbol{W}}$$

$$= \frac{\partial L}{\partial \sigma} \boxed{\frac{\partial \sigma(z)}{\partial z}} \frac{\partial(\boldsymbol{W}x + \boldsymbol{b})}{\partial \boldsymbol{W}}$$

什麼年代, 微分用 limit 算

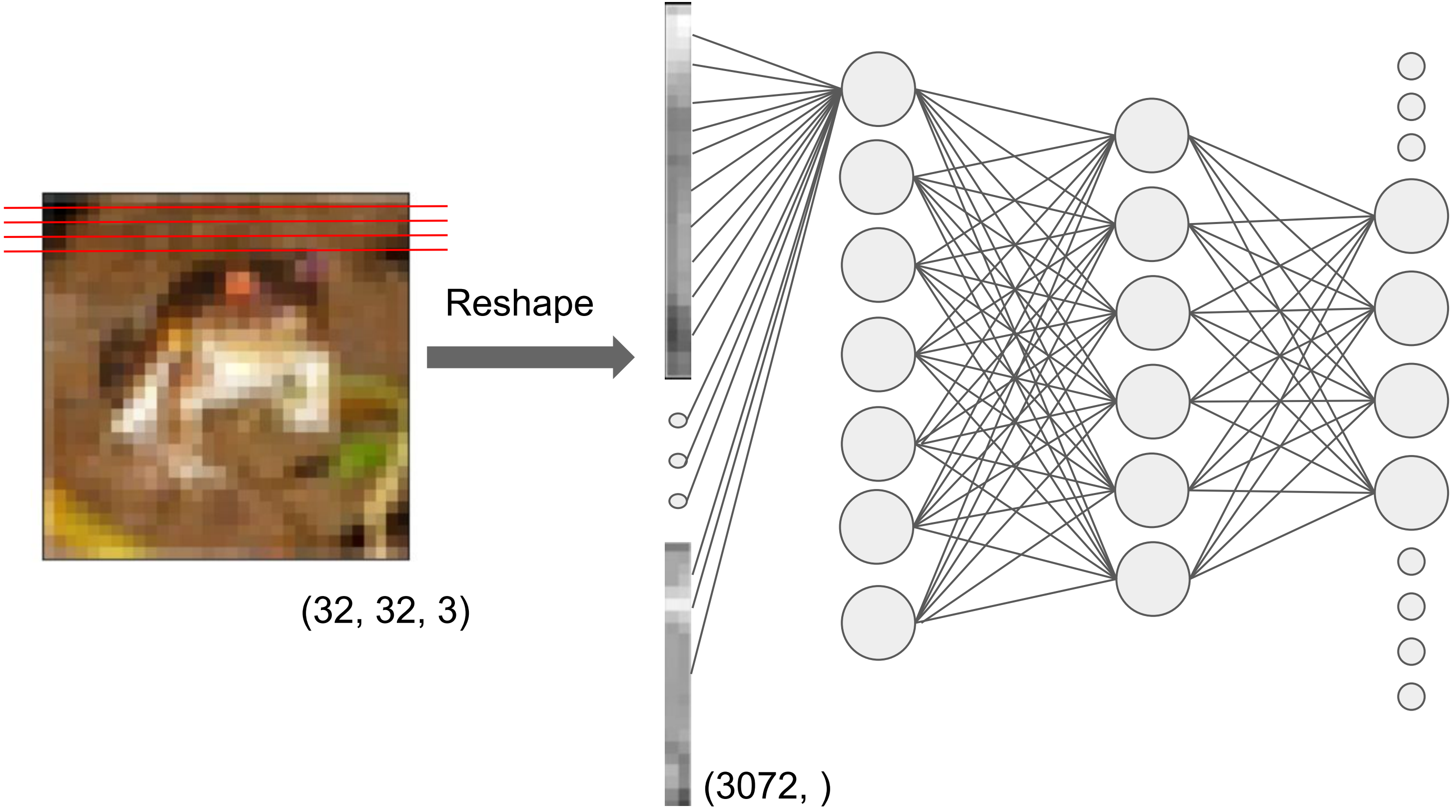
$$\cancel{\lim_{h \to 0} \frac{((\boldsymbol{W} + h)x + b) - (\boldsymbol{W}x + b)}{h}}$$

W^3_{11} ⬅️ W^3_{11} - lr x $\nabla_{W^3_{11}}$ L

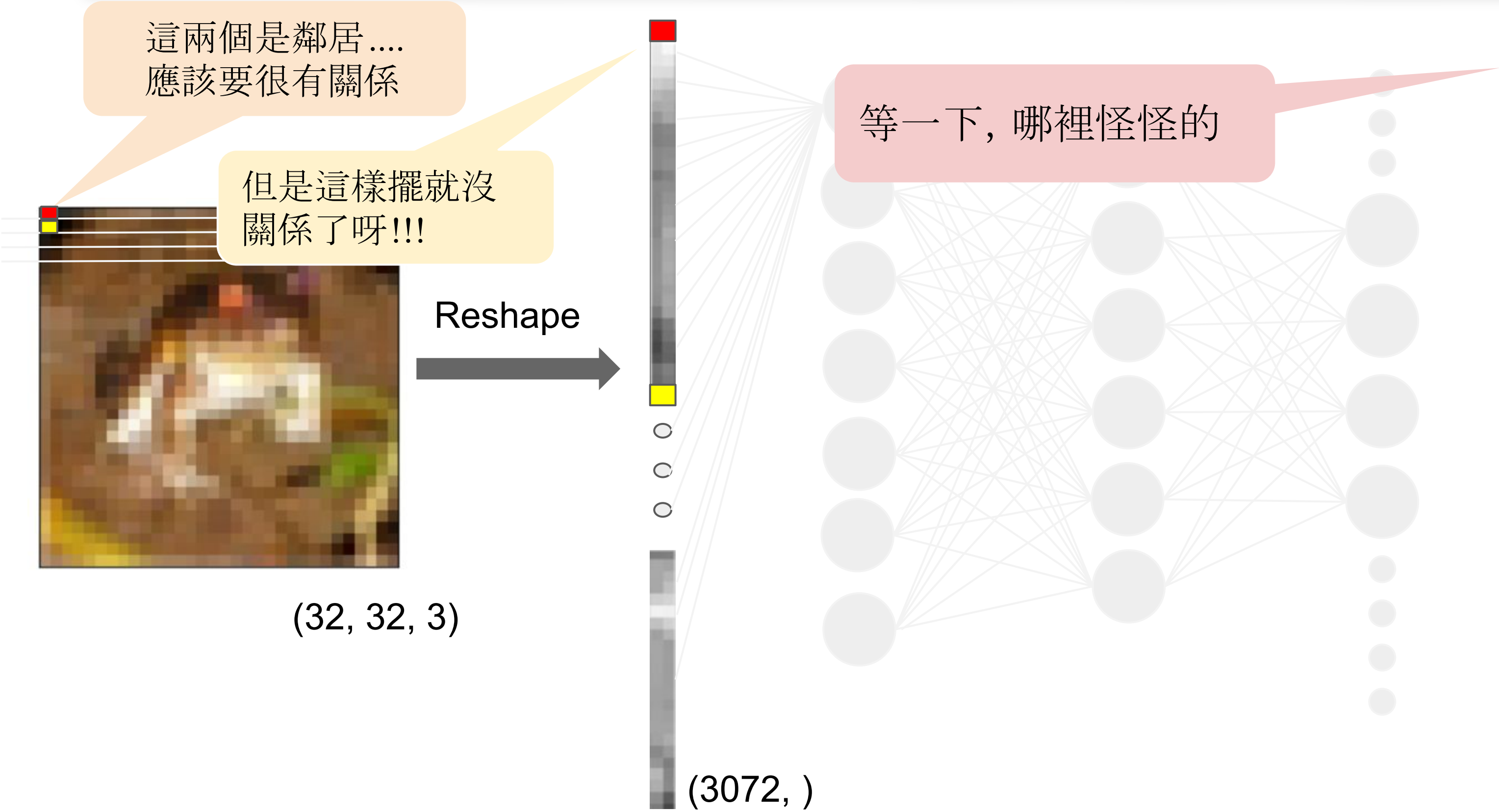
$\nabla_{W^3_{11}} L = \nabla_{\hat{y}} L \times \nabla_{z_{31}} \hat{y} \times \nabla_{W^3_{11}} z_{31}$

$= -2(y-\hat{y}) \times z_{31}(1-z_{31}) \times z_{21}$

回憶過去....cifar10



回憶過去....cifar10



Convolution Operation - 讓每個鄰居都很有關係

What is convolution?

Image

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

*

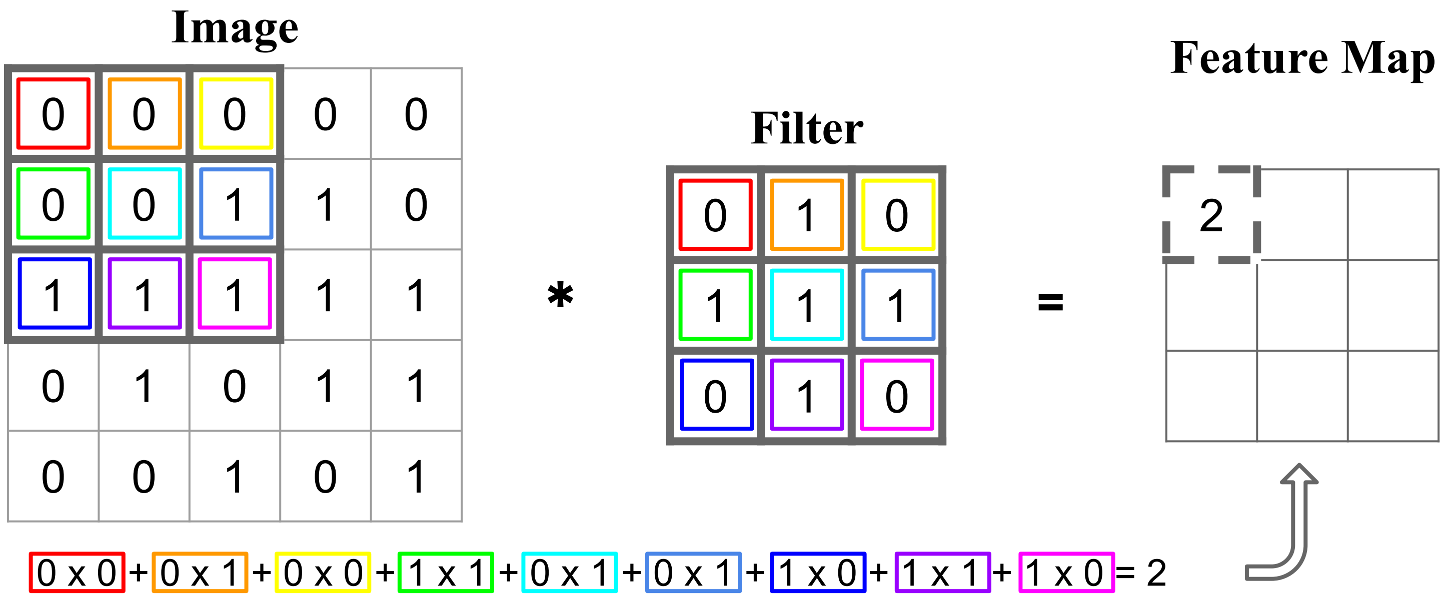
Filter

0	1	0
1	1	1
0	1	0

= ?

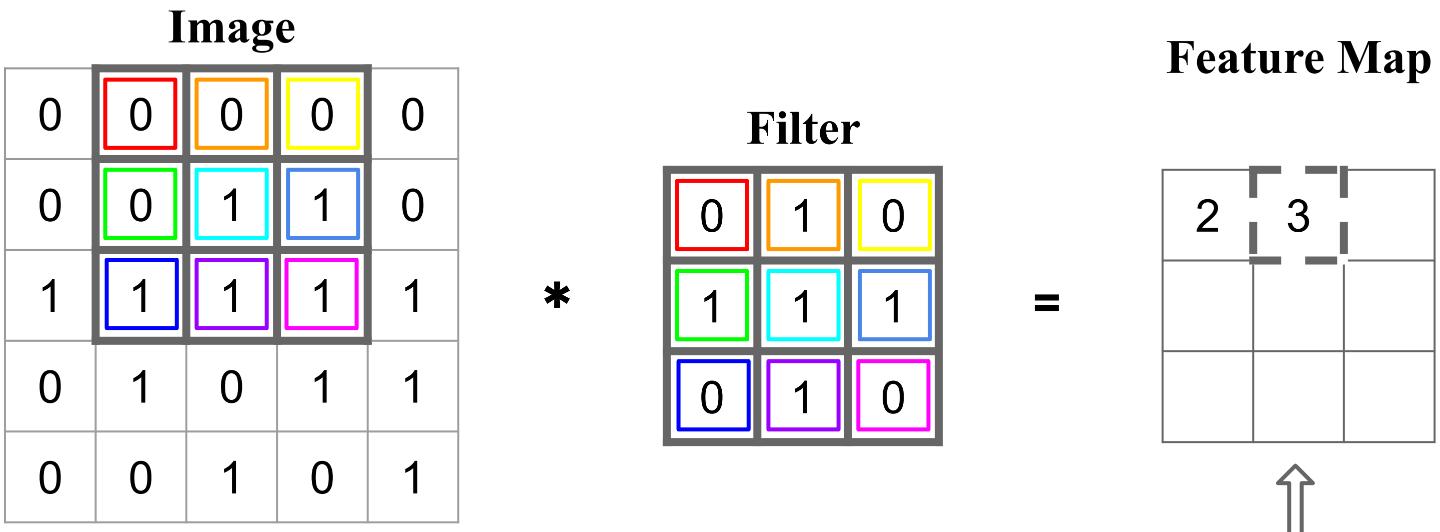
Convolution Operation - 讓每個鄰居都很有關係

What is convolution?



Convolution Operation - 讓每個鄰居都很有關係

What is convolution?



0 x 0 + 0 x 1 + 0 x 0 + 0 x 1 + 1 x 1 + 1 x 1 + 1 x 0 + 1 x 1 + 1 x 0 = 3

Convolution 卷積

Functional API

Conv2D

- input_channel = 1
- filters = 1
- kernel_size = (3, 3)
- strides = (1, 1)

動手算算看....

看看你的學習有沒有 Overfitting??

Image

0	0	0	0	0	0
0	0	3	3	3	0
0	0	1	1	3	0
0	0	1	1	3	0
0	0	0	0	0	0
0	0	0	0	0	0

*

Sobel Filter

1	0	-1
1	0	-1
1	0	-1

1

=   ? $\xrightarrow{\text{relu}}$

2

?

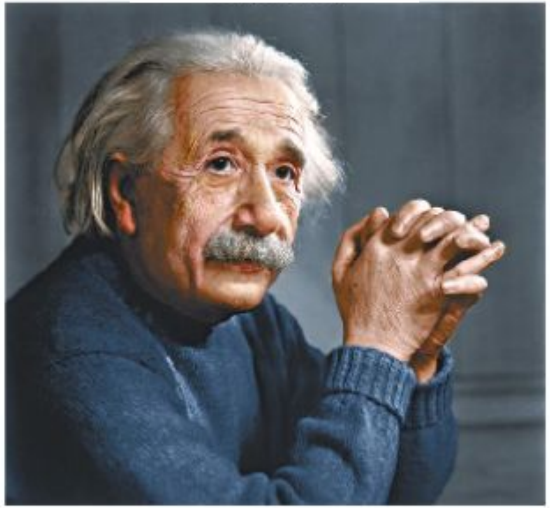
Filter 的古老密技

Smoothing
平滑化

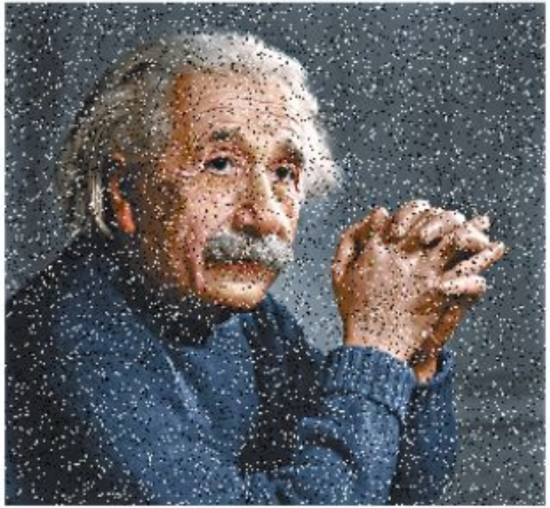
Sharpening
銳利化

說說 Smoothing - 說說哪裡不一樣

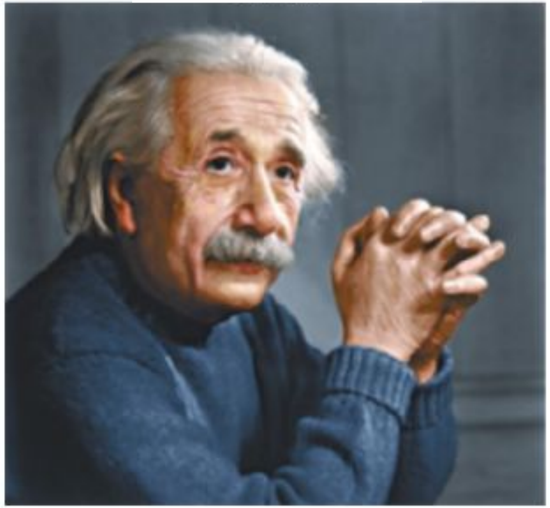
Original Image



Oops!

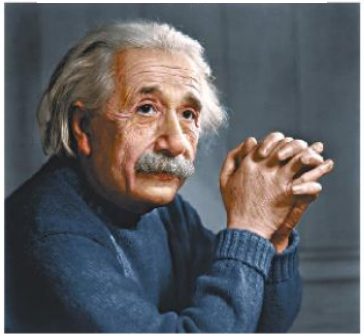


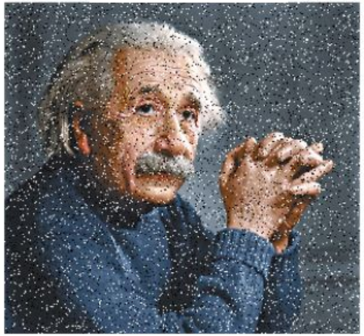
Average Filter



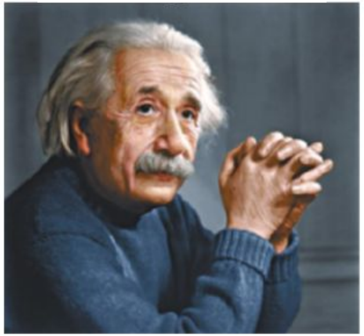
說說 Smoothing - 去去雜訊走

Original Image

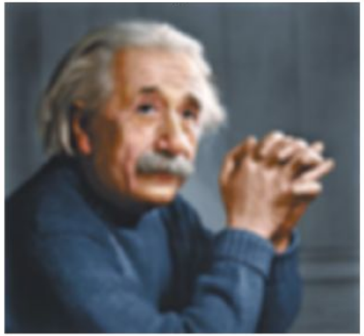




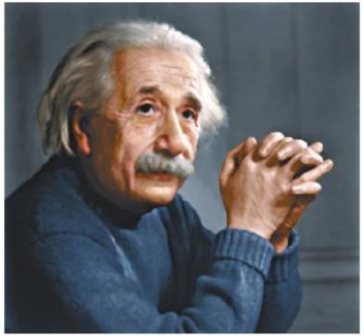
Average Filter



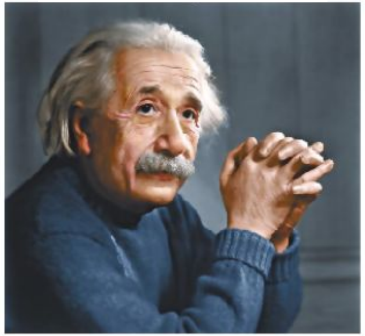
Gaussian Filter



Median Filter



Bilteral Filter



Average Filter

$\frac{1}{9}$ X

1	1	1
1	1	1
1	1	1

Gaussian Filter

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

0.045	0.122	0.045
0.122	0.332	0.122
0.045	0.122	0.045

說說 Sharpening - 就是能摸得著邊

Original

Sobel Filter_H

1	2	1
0	0	0
-1	-2	-1

Sobel Filter_v

1	0	-1
2	0	-2
1	0	-1

Laplacian Filter

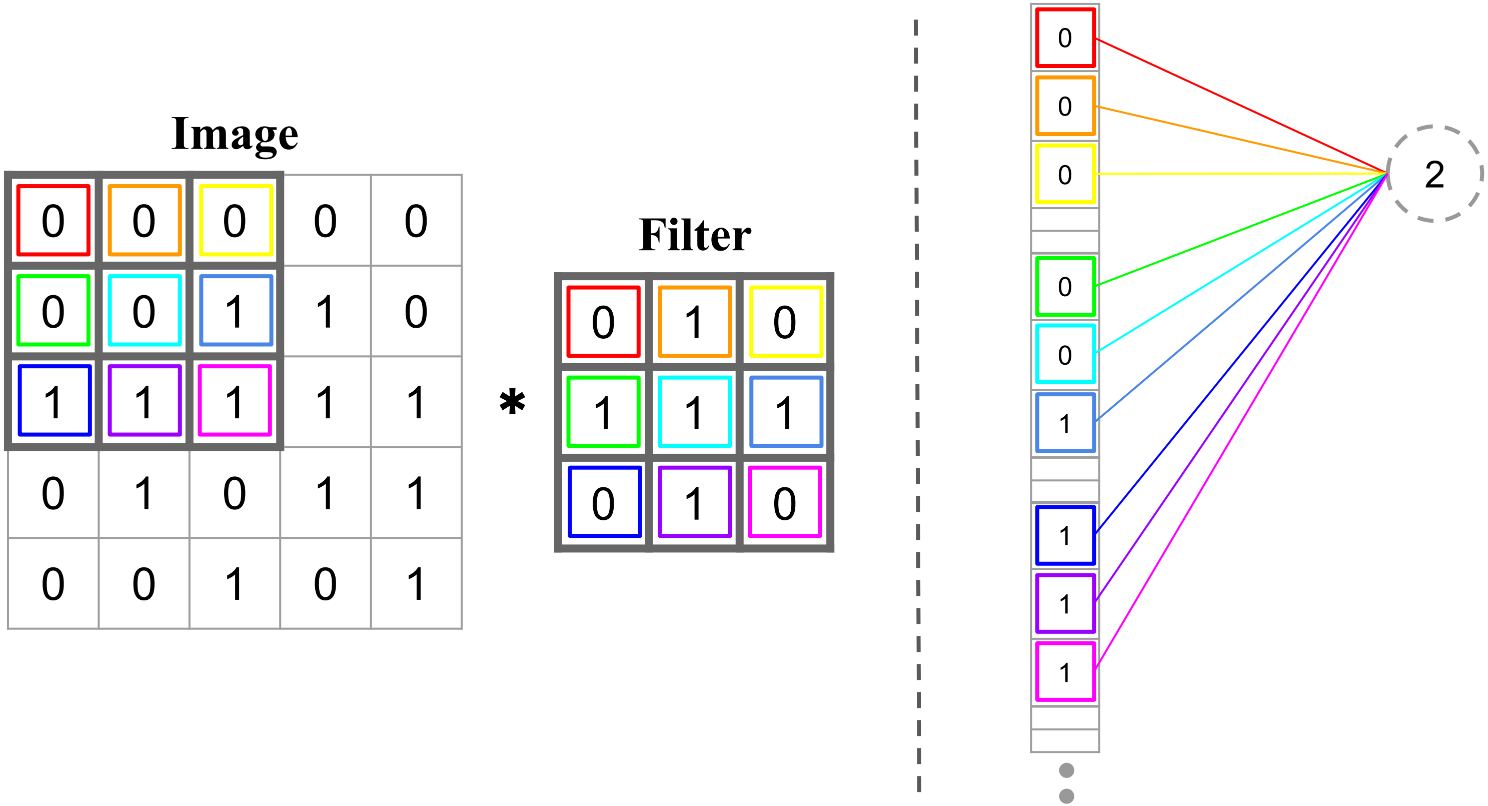
-1	-1	-1
-1	8	-1
-1	-1	-1

Horizantol

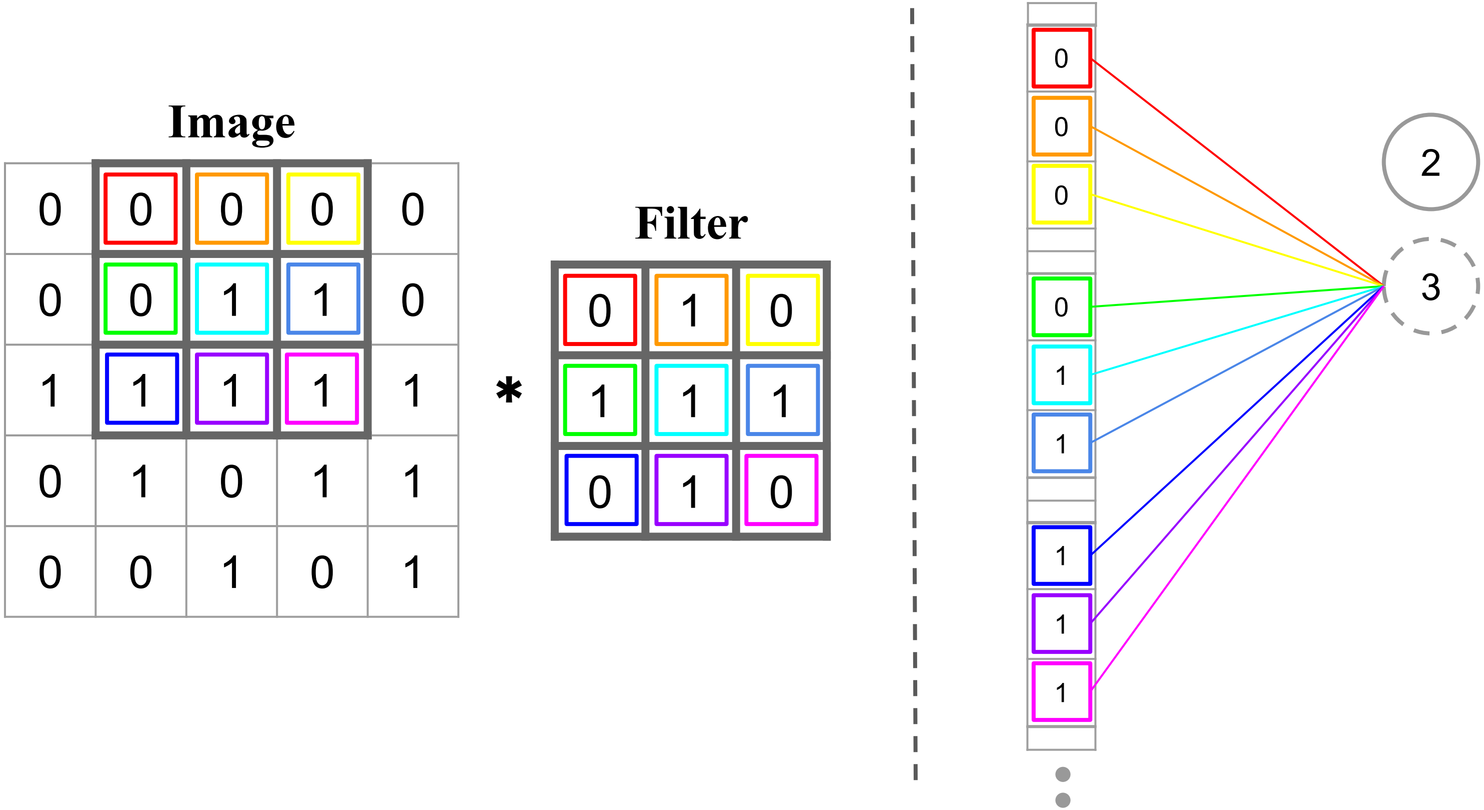
Vertical

Laplacian

CNN - convolution 和 NN 有什麼關係



CNN - convolution 和 NN 有什麼關係



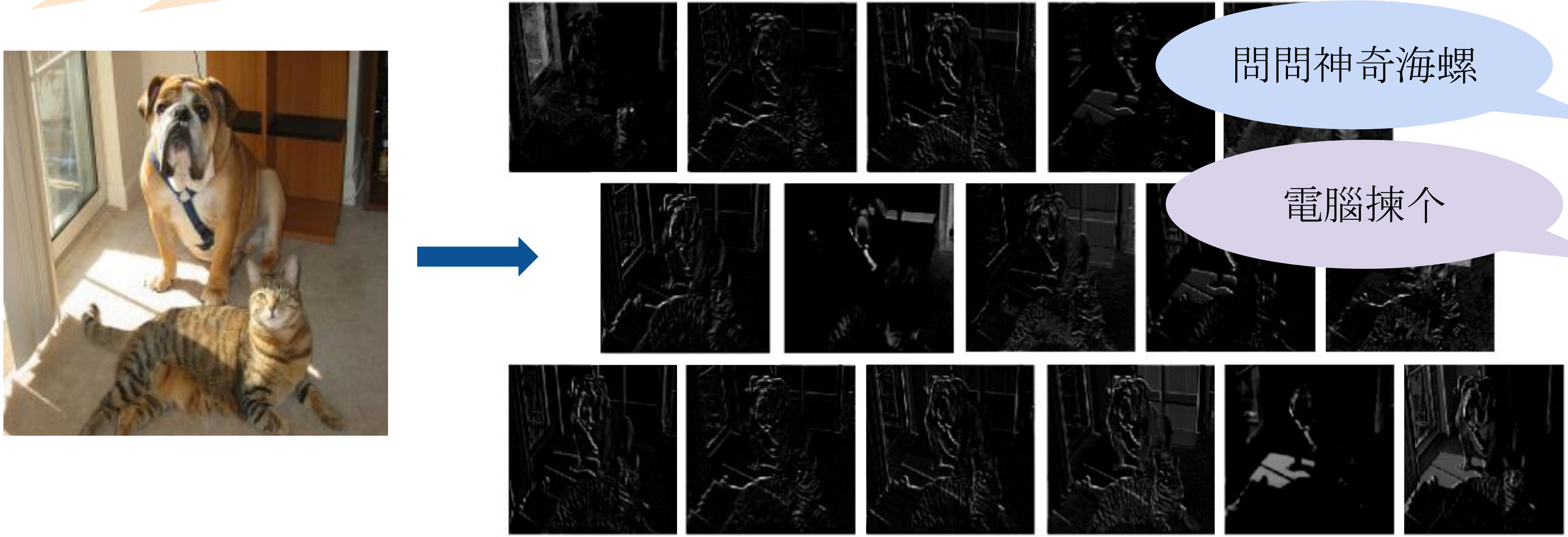
Convolutional Neuron

很多個 Neuron
決定很多個影像的特徵

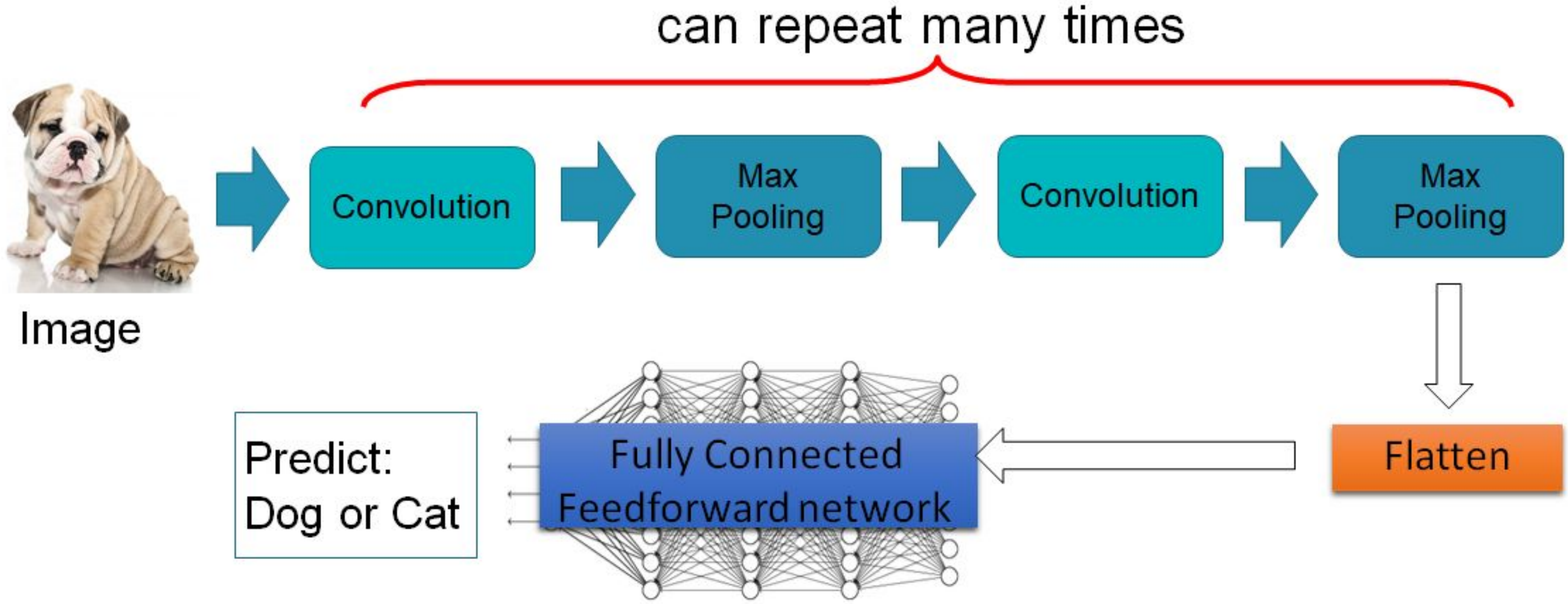
怎麼樣決定特徵的？

問問神奇海螺

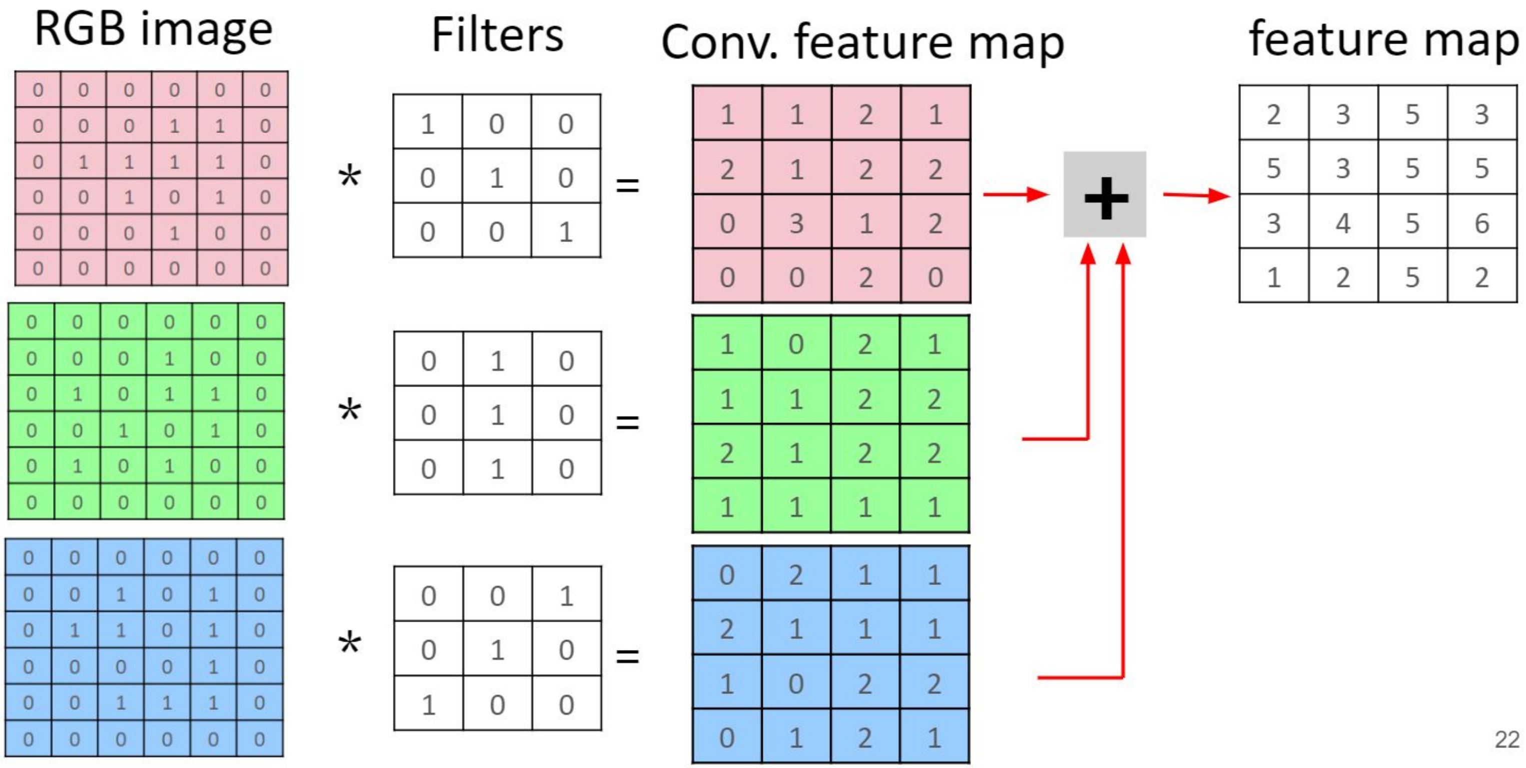
電腦揀个



A Convolutional Neural Network



Convolution with Multi-Channel



Convolution with Multi-Channel

Functional API

Conv2D

- input_channel = 3
- filters = 1
- kernel_size = (3, 3)
- strides = (1, 1)

MaxPooling

Functional API

MaxPooling

- window_size = (2, 2)
- strides = (2, 2)

Flatten

動手刻一個
CNN Model

Data Preprocessing

資料前處理

不急 train 一發
要先做前處理....

為什麼要做前處理？

影像大小不同

我是柴柴

我變秋田了

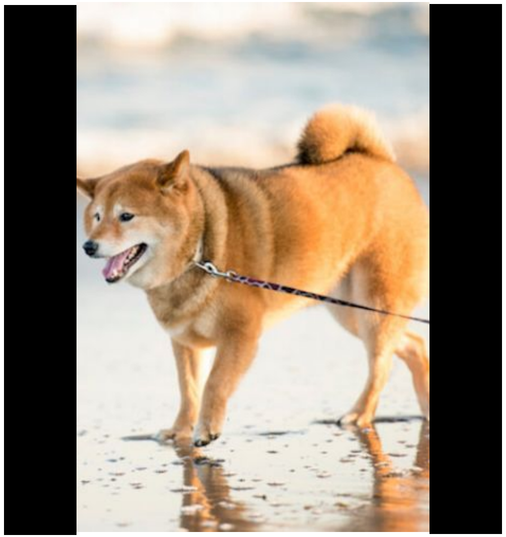
我變成科基了

方式之一....

我是柴柴

我也是柴柴

我還是柴柴



Data Augmentation

資料增強

ImageGenerator

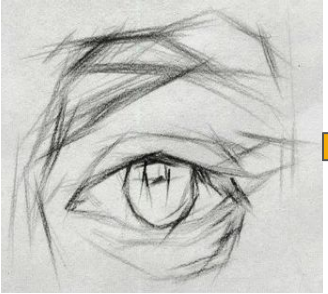
```
keras.preprocessing.image.ImageDataGenerator(featurewise_center=False,
                                             samplewise_center=False,
                                             featurewise_std_normalization=False,
                                             samplewise_std_normalization=False,
                                             zca_whitening=False,
                                             zca_epsilon=1e-06,
                                             rotation_range=0,
                                             width_shift_range=0.0,
                                             height_shift_range=0.0,
                                             brightness_range=None,
                                             shear_range=0.0,
                                             zoom_range=0.0,
                                             channel_shift_range=0.0,
                                             fill_mode='nearest',
                                             cval=0.0,
                                             horizontal_flip=False,
                                             vertical_flip=False,
                                             rescale=None,
                                             preprocessing_function=None,
                                             data_format=None,
                                             validation_split=0.0,
                                             dtype=None)
```

Transfer Learning

遷移式學習

一張影像的點線面

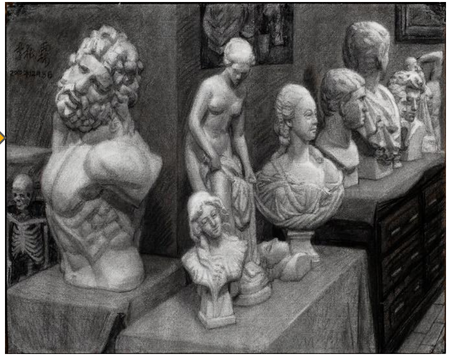
Line Segment



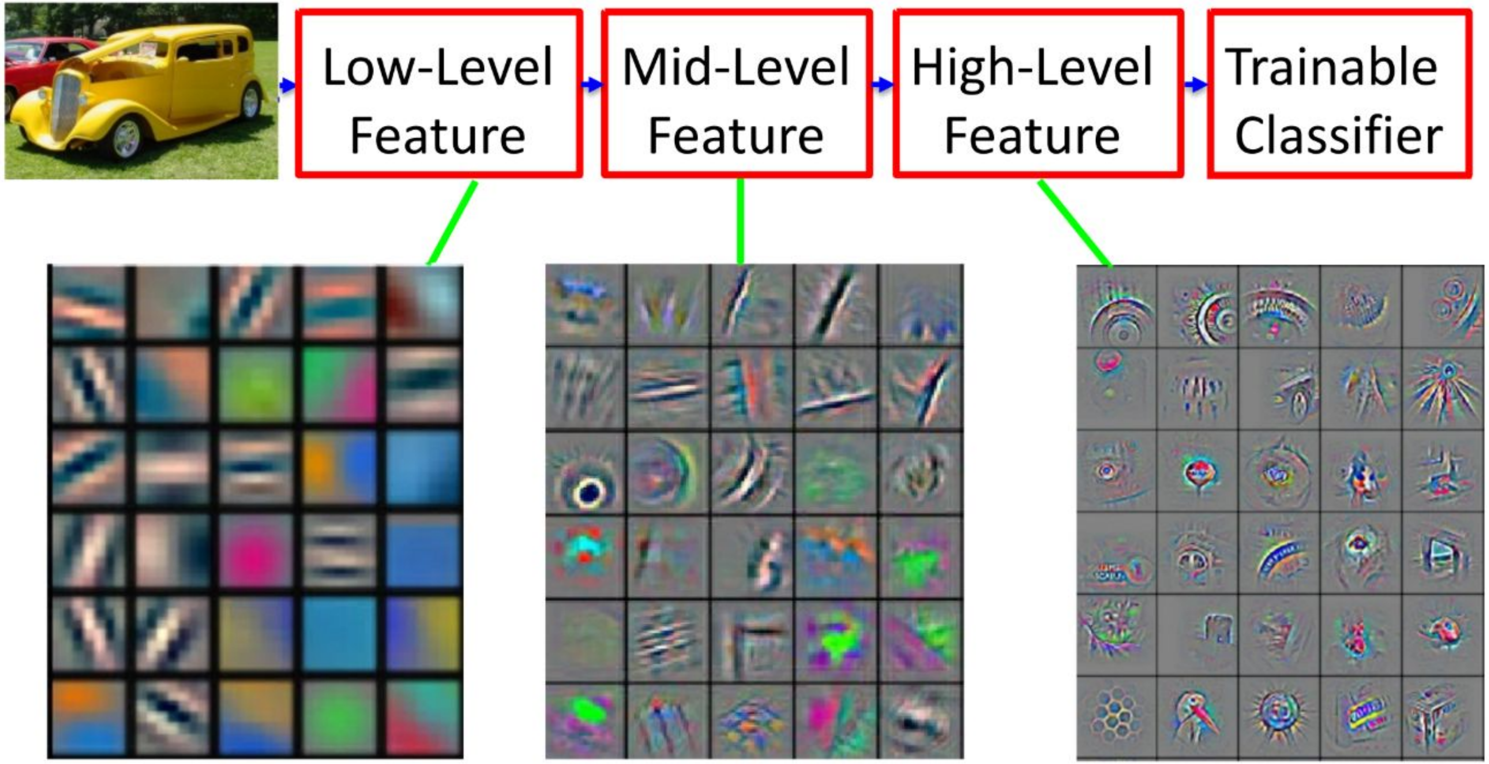
Pattern

Object

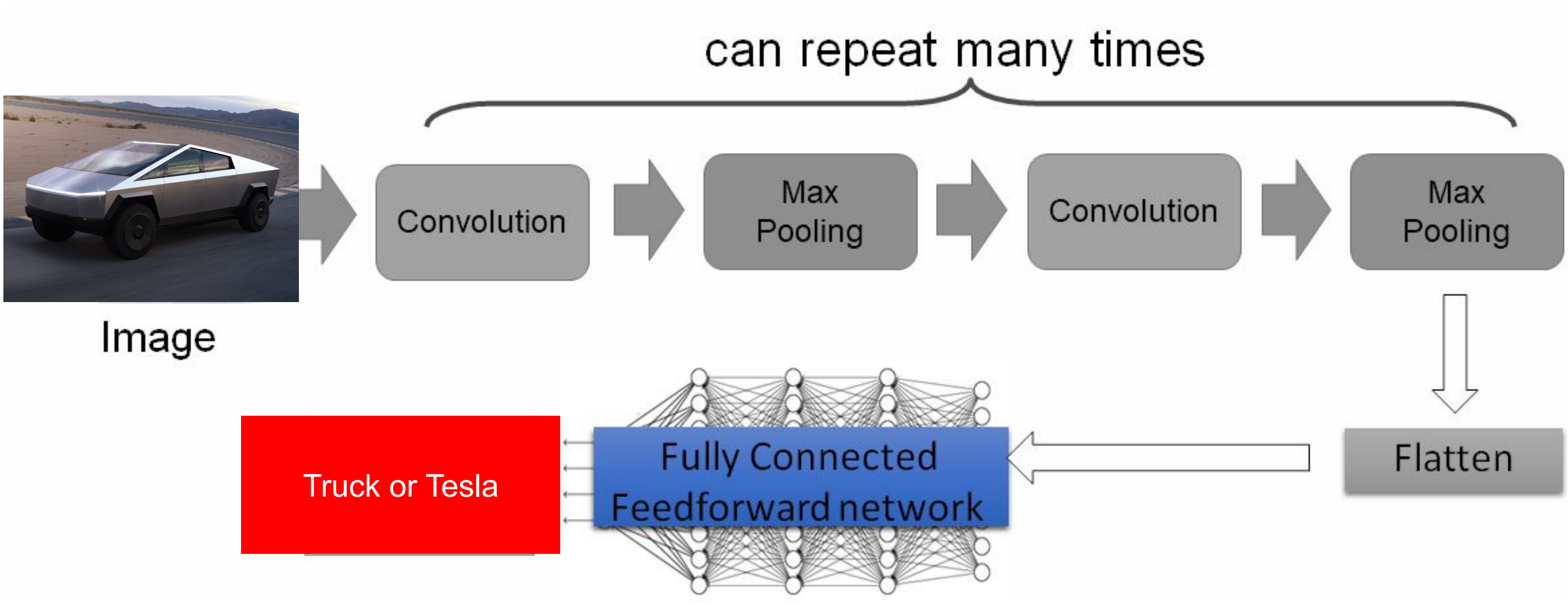
Scene



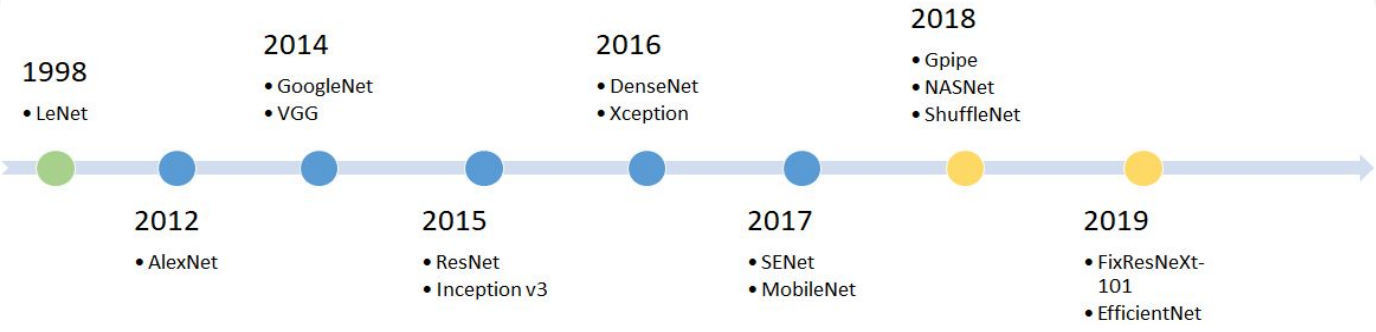
Transfer Learning 的概念



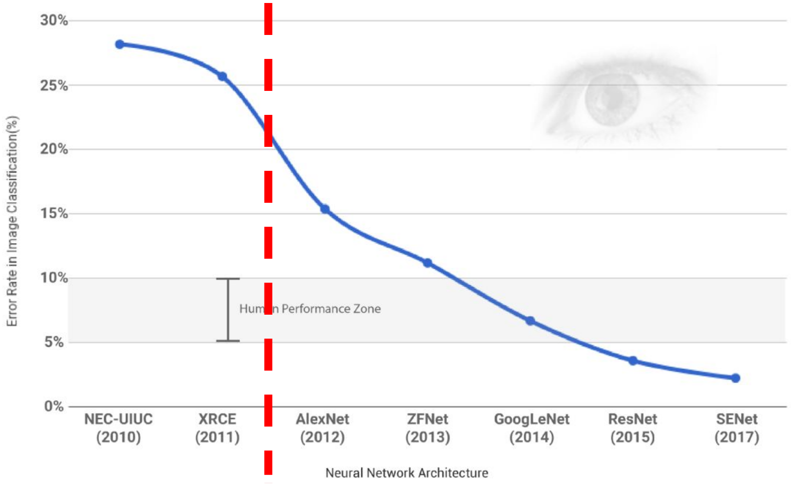
唯一不同的是....


can repeat many times
Image
Convolution
Max Pooling
Convolution
Max Pooling
Flatten
Fully Connected Feedforward network
Truck or Tesla

The MileStone of the ILSVRC



- 李飛飛 Li Fei Fei
- 2010~2017
- 15,000,000 Label Images



Available models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-
EfficientNetB0	29 MB	-	-	5,330,571	-
EfficientNetB1	31 MB	-	-	7,856,239	-
EfficientNetB2	36 MB	-	-	9,177,569	-
EfficientNetB3	48 MB	-	-	12,320,535	-
EfficientNetB4	75 MB	-	-	19,466,823	-
EfficientNetB5	118 MB	-	-	30,562,527	-
EfficientNetB6	166 MB	-	-	43,265,143	-
EfficientNetB7	256 MB	-	-	66,658,687	-

GlobalAveragePooling

Functional API

GlobalAveragePooling

Appendix A

optimizer

Optimizer 優化器

- Gradient Descent (GD)

  - 模型看完一輪所有資料，更新一次權重

  - 優點
    - 根據所有資料的訊息，更新模型
  - 缺點
    - 結果好壞很依賴起始點
    - 很容易掉進去 Local Minimum 就限制住找 Global Minimum 的可能

- Stochastic Gradient Descent (SGD)

  - 模型看完一小堆(batch)資料，更新權重一次

  - 優點
    - 收斂速度上，比 GD 快
  - 缺點
    - 可能會抽到極端的擾動值資料，偏離了 Global Minimum 的收斂方向

Optimizer 優化器

- RMSProp

  - 模型看完一小堆(batch)資料，更新權重一次

  - 優點
    - 不僅有依照當次計算的梯度做更新，還有參考過去的梯度做更新依據
  - 缺點
    - 還是有掉進去 Local Minimum 的可能

$$w^1 \leftarrow w^0 - \frac{\eta}{\sigma^0} g^0 \qquad \sigma^0 = g^0$$

$$w^2 \leftarrow w^1 - \frac{\eta}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\alpha(\sigma^0)^2 + (1-\alpha)(g^1)^2}$$

$$w^3 \leftarrow w^2 - \frac{\eta}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\alpha(\sigma^1)^2 + (1-\alpha)(g^2)^2}$$

$$\vdots$$

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\alpha(\sigma^{t-1})^2 + (1-\alpha)(g^t)^2}$$

- ADAM

  - 模型看完一小堆(batch)資料，更新權重一次
  - 加了 Momentum 的 RMSProp

  - 優點
    - 給予一點點小小的擾動 (Momentum)，協助翻出 Local Minimum 的小峽谷，找到 Global Minimum 的可能
    - 適合大部分的優化狀況使用

Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t/(1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t/(1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t/(\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)