

資源區

講義、程式碼：<https://github.com/Shawnroom/keras>

回饋表單：

C1:使用Keras建構CNN模型

主講人：謝長潤、陳俊豪

2016-12-01

關於這堂課……

- 不是教學，是分享

今天沒有的東西

- 艱深的數學（微積分、線性代數）
- 困難的程式語法（I hope so…）

今天有的東西

- 簡單的比喻
- 有趣的應用

大綱

1. CNN能用來做什麼?
2. CNN運作原理簡介
3. CNN的參數調整
4. Keras實際演練

大綱

1. CNN能用來做什麼?
2. CNN運作原理簡介
3. CNN的參數調整
4. Keras實際演練

CNN能用來做什麼？

1. 即時辨識手寫數字：

Keras.js - <https://transcranial.github.io/keras-js/#/mnist-cnn>

Basic Convnet for MNIST

Interactive demo of a simple convnet trained on MNIST (see [Jupyter notebook](#)). All computation performed entirely in your browser. Toggling GPU on/off shouldn't reveal any significant speed differences, as this is a fairly small network. In the architecture diagram below, intermediate outputs at each layer are also visualized.

✕ CLOSE

Draw any digit (0-9) here ↴



✕ CLEAR

☒ use GPU



Google Quick , draw

2. 即時辨識手繪圖案：

Quick , draw - <https://quickdraw.withgoogle.com/>

Google 翻譯 - <http://ppt.cc/Ha6kZ>

You were asked to draw dragon

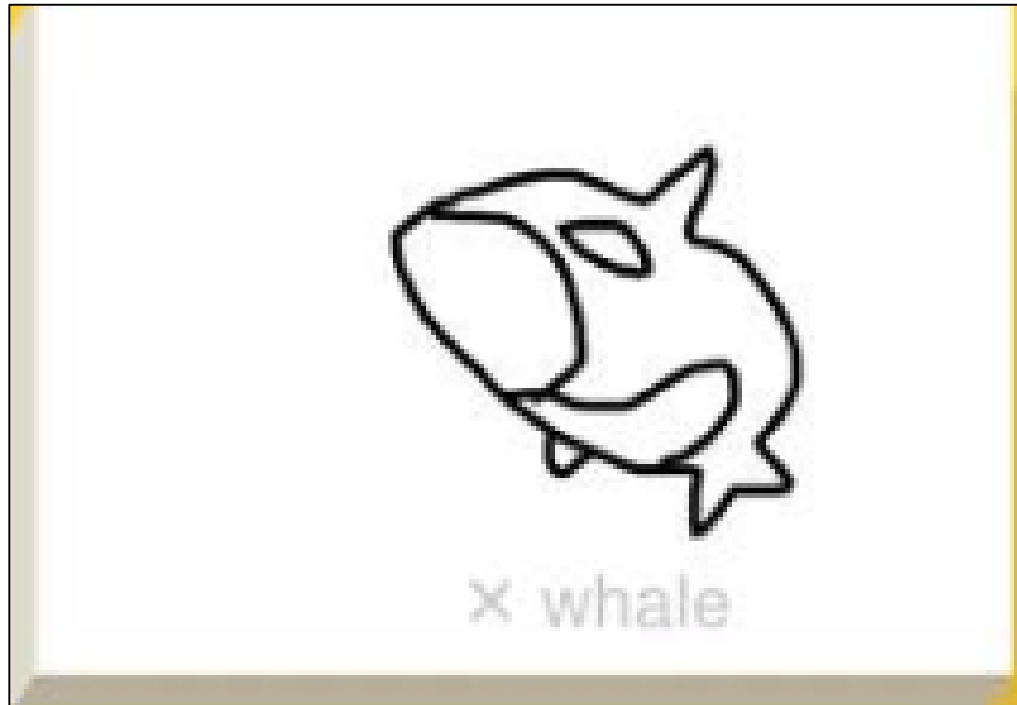
You drew this, and the neural net recognized it.



Google Quick, draw

玩完之後，你一定有一個疑問……

為什麼我畫得這麼好，Google小姐卻看不出來？（做壞了吧？！



我們請得分高的同學上台玩一次

我們是如何理解 / 辨認圖形？

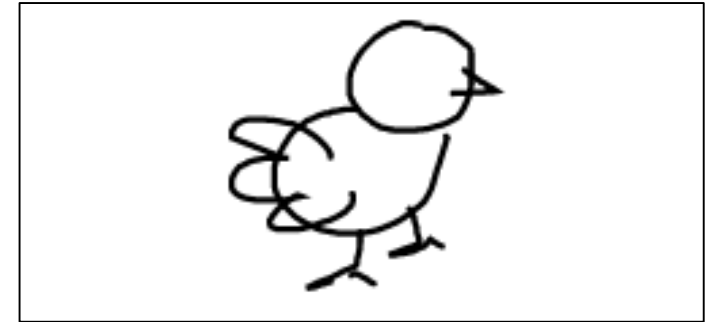
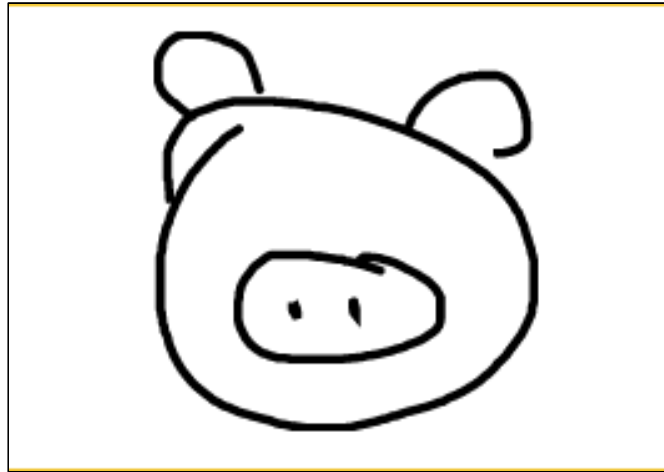
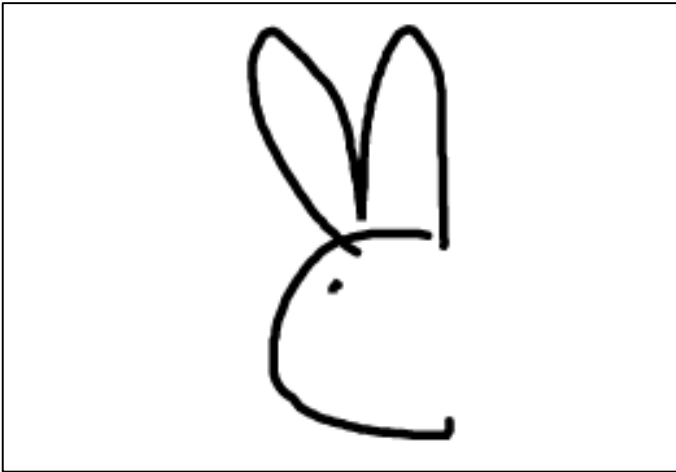
見微知著

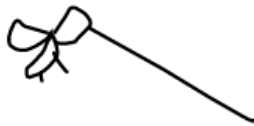
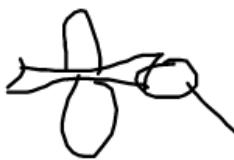
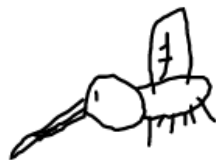
每件物品一定有其獨特的特徵。透過觀察圖片中不同局部擁有的特徵，幫助判斷可能是哪一個物品

Ex：家長教小孩

如果不給你答案，你看得出來其他人在畫什麼嗎？

Google Quick , draw





CNN是如何理解 / 辨認圖形？

見微知著

每件物品一定有其獨特的特徵。CNN透過觀察圖片中不同局部擁有的特徵，幫助判斷可能是哪一個物品

CNN又沒有眼睛

它是用什麼方式看的呢？

大綱

1. CNN能用來做什麼?
2. CNN運作原理簡介
3. CNN的參數調整
4. Keras實際演練

Neural network 學習流程

決定神經網絡架構

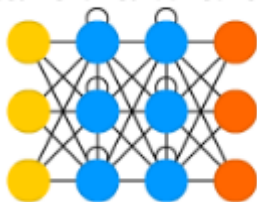


設定學習目標

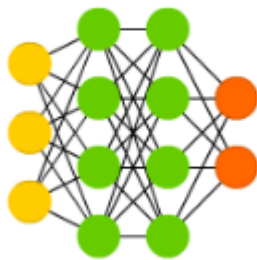


訓練，找到最佳解

Recurrent Neural Network (RNN)



Deep Feed Forward (DFF)



任務性質

分類？
預測數值？

什麼叫做好結果？

梯度下降方式
成本函數

調整模型參數

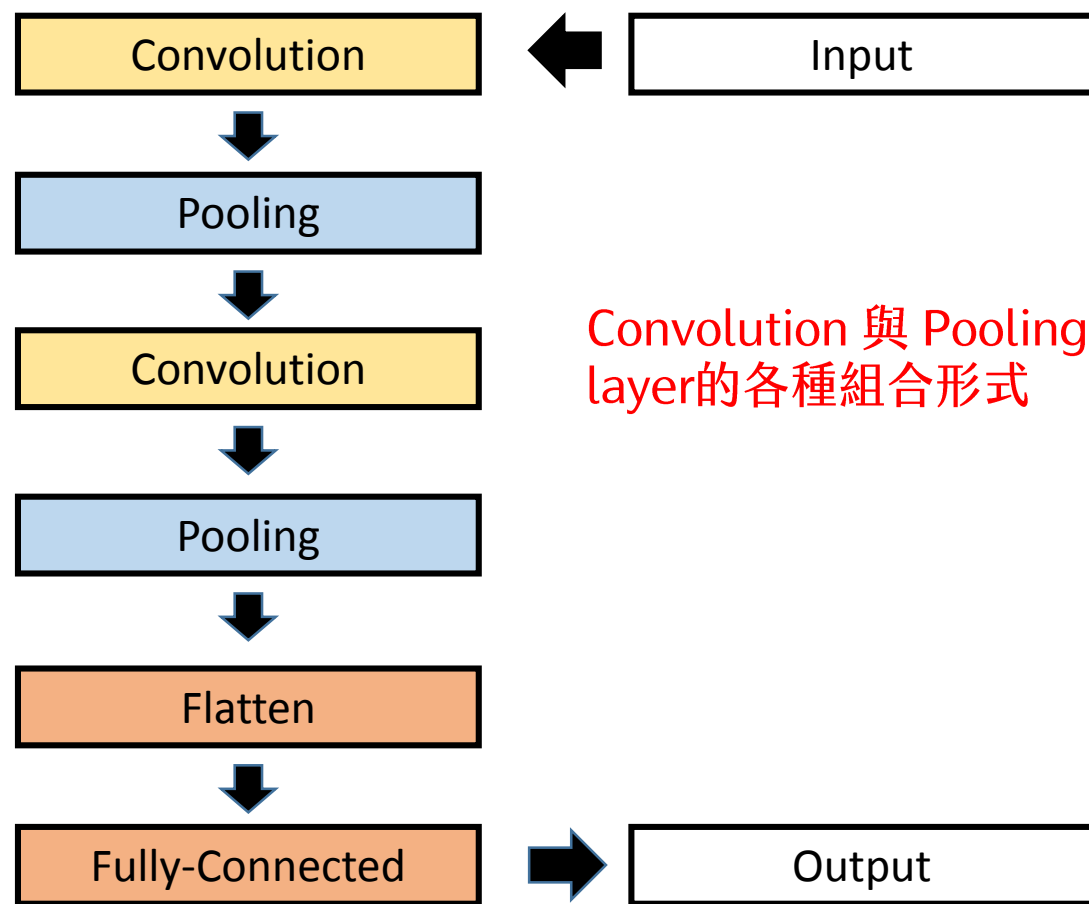
Activation function
訓練幾輪
其它CNN參數

CNN學習過程

決定神經網絡架構

Some famous CNN models:

LeNet
AlexNet
ResNet



Convolution

CNN模型的眼睛

圖片

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

Stride = 1

Filter - 特徵觀察器

局部觀察

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

-3	1	-2
-3	-2	0
3	-2	-1

Convolution

圖片

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

Stride = 1

Filter - 特徵觀察器

局部觀察

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

-3	1	-2
-3	-2	0
3	-2	-1

Convolution

圖片

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

Stride = 1

Filter - 特徵觀察器

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

-3	1	-2
-3	-2	0
3	-2	-1

局部觀察

Convolution

觀察結果

圖片

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

Filter - 特徵觀察器

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

-3	1	-2
-3	-2	0
3	-2	-1

圖形的左下角
有一條斜線

Convolution

跳著看

圖片

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

Stride = 2

Filter - 特徵觀察器

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

-3	-2
3	-1

Convolution

跳著看

圖片

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

Stride = 2

Filter - 特徵觀察器

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

-3	-2
3	-1

Convolution

Zero Padding : 圖片邊緣加入一層0，可幫助保留邊界訊息 看好看滿

圖片

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

Stride = 1

Filter - 特徵觀察器

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器



Feature map

0	0	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

Convolution

圖片

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

可以有很多個filter

Filter - 特徵觀察器

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器

1	1	1
0	0	0
0	0	0

橫線
觀察器

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

斑馬線
觀察器



產生出很多張Feature map

Feature map

-3	1	-2
-3	-2	0
3	-2	-1
3	2	2
1	1	2
1	0	0
2	-1	4
2	-1	2
0	0	0

圖形的左下方
有一條斜線

圖形的左上方
有一條橫線

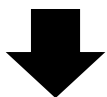
圖形的右上方
有斑馬線

Convolution

Convolution L1



Convolution L2



Convolution L3

直線、橫線、斜線、
斑馬線.....

三角形、正方形、
菱形、圓形.....

蹠、鳥嘴、眼睛、
嘴巴.....

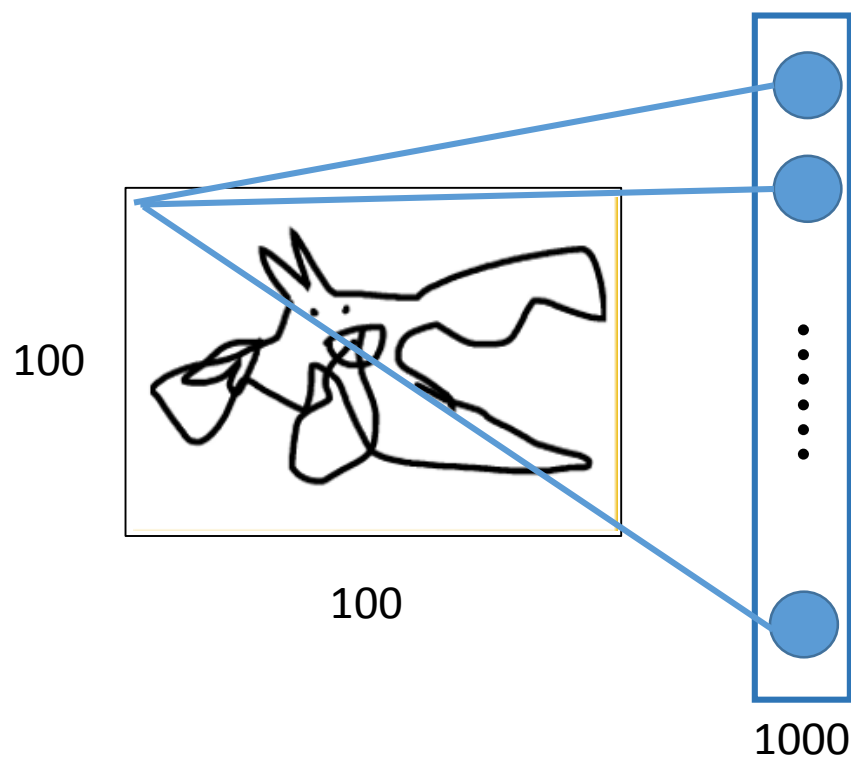
簡單



複雜

第二層的Convolution能利用第一層Convolution的特徵結果組合出更複雜的特徵進行filter。換句話說，越高層的Convolution，能夠觀察出越細微的特徵。

為什麼要局部觀察，一格一格觀察不是更詳細嗎？

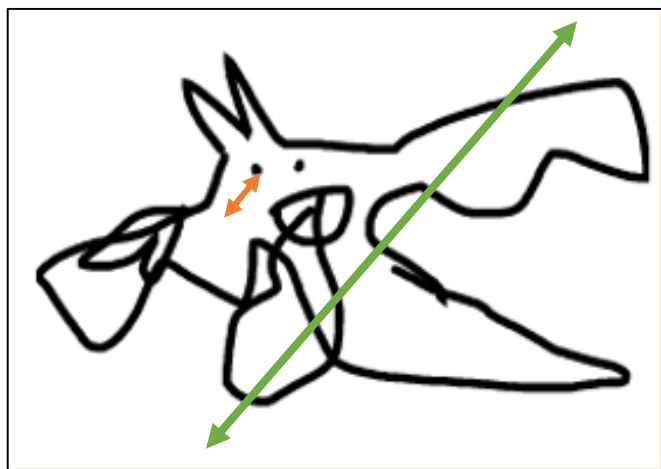


光是一張圖片的第一層，就有 $100 * 100 * 1000 = 10^7$ 個參數要訓練

如果是彩色圖片，還要乘3，等於有 $3 * 10^7$ 個參數要訓練

拳頭硬硬的！

局部觀察



圖片中相鄰Pixel的關連性高，
距離越遠的關聯性越低

有些局部特徵比圖片小很多，
不必觀察整張圖片

局部觀察並不會損失
太多關聯性高的資訊

見微知著

Parameter Sharing (參數共享)

Filter用同一個權重去掃描圖片中的不同區域，能有效的減少參數數量

圖片

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

Stride = 1

Filter - 特徵觀察器

局部觀察

1	-1	-1
-1	1	-1
-1	-1	1

斜線
觀察器

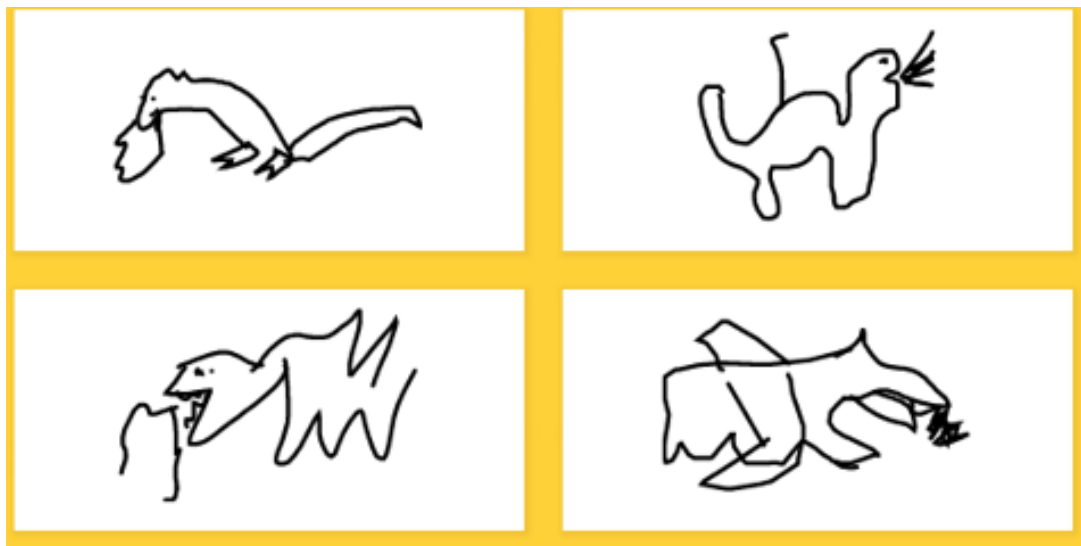


Feature map

2	2	2
0	1	1
3	0	1

Parameter Sharing (參數共享)

為什麼使用參數共享是個好主意？



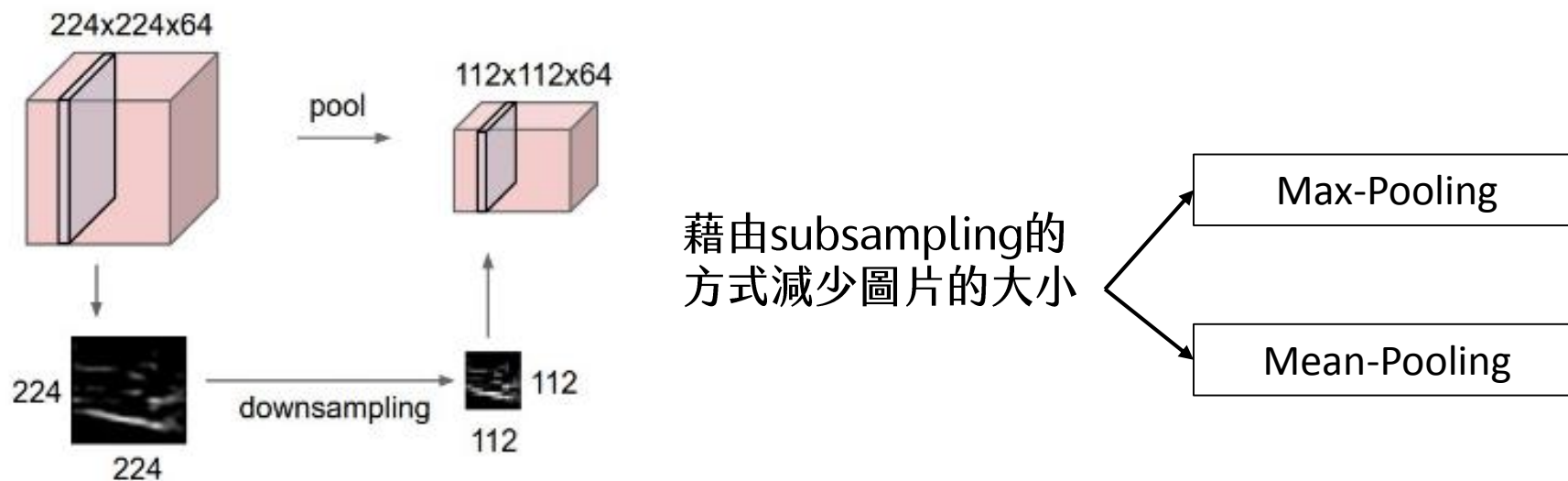
特徵的絕對位置不重要，
重要的是特徵有沒有出現

有些特徵會重複出現在圖
片中不同位置

牛牽到北京還是牛

Pooling

壓縮圖片的長寬 ➡ 減少後續layer的參數數量，防止overfitting



Pooling

Max-Pooling

6	8
7	5

較常使用

Feature map

1	2	3	4
5	6	7	8
7	6	5	4
3	2	1	0

Filter = 2x2
Stride = 2

Mean-Pooling

3.5	5.5
4.5	2.5

Pooling

可以不要Pooling嗎？
有人這樣做過

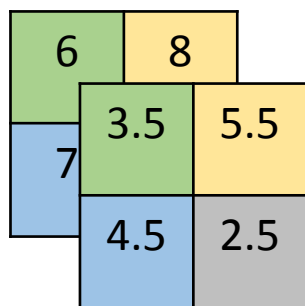
在Convolutional layer 中選擇較大的stride取代（VAEs, GANs）

<https://arxiv.org/abs/1412.6806>

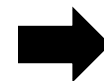
Flatten



Fully-Connected



Fully-Connected



Output

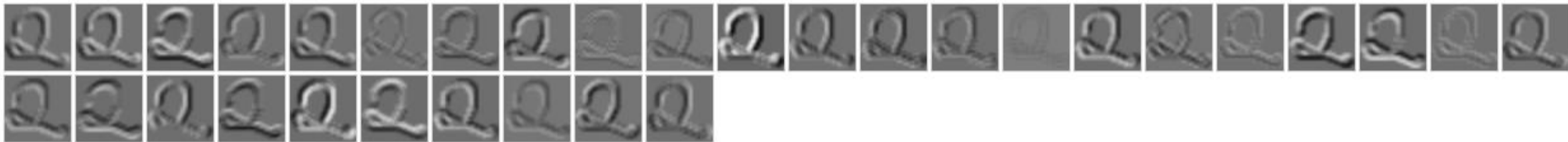
經過Pooling後的
feature map

Keras.js

Keras.js - <https://transcranial.github.io/keras-js/#/mnist-cnn>

Convolution2D

32 3x3 filters, border mode valid, 1x1 strides



Activation

ReLU



Back to Google Quick , draw

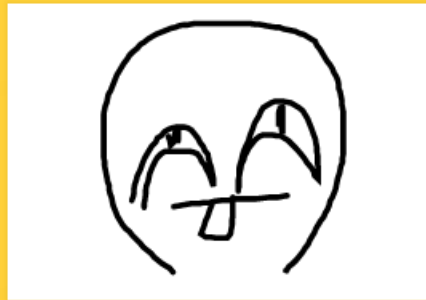
再來玩一次吧！！！！

Quick , draw - <https://quickdraw.withgoogle.com/>

Google 翻譯 - <http://ppt.cc/Ha6kZ>

You were asked to draw octopus

You drew this, and the neural net recognized it.



Back to Google Quick , draw

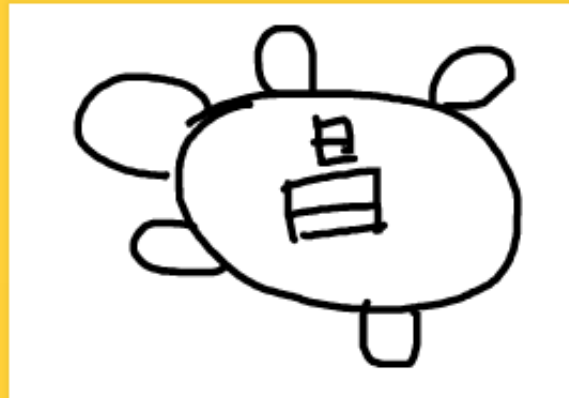
怎麼畫比較好？

大眾認為這個物品應該有的特徵
盡量別加不重要的特徵

Back to Google Quick, draw

You were asked to draw sea turtle

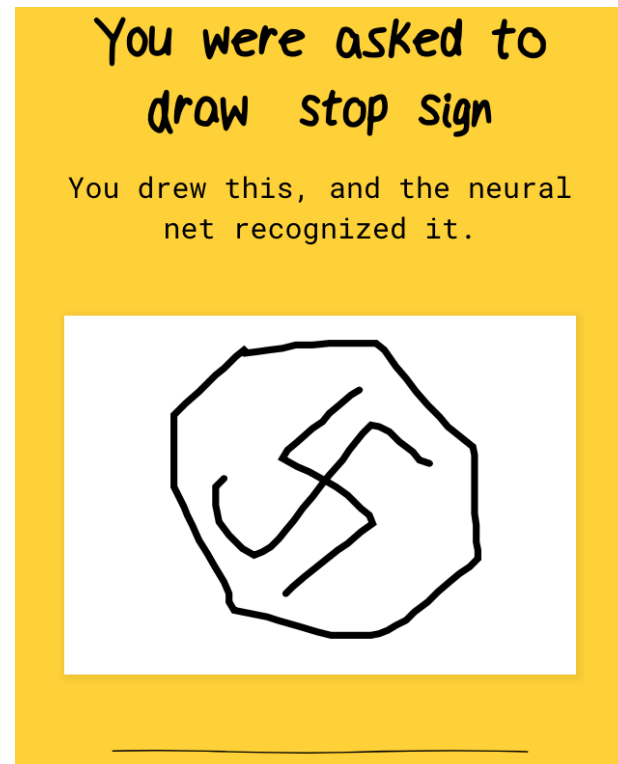
You drew this, and the neural net recognized it.



Back to Google Quick, draw

當Google小姐遇到4chan鄉民 - (警告：18禁)

[搜尋：\[NSFW\] Google's Quick Draw is easily fooled](#)



大綱

1. CNN能用來做什麼?
2. CNN運作原理簡介
3. CNN的參數調整
4. Keras實際演練

設定學習目標

以mnist數字辨識為例

任務性質



分類，0-9共10類

分類？
預測數值？

什麼叫做好結果？



怎樣梯度下降比較好？
該使用哪種成本函數？

梯度下降方式
成本函數

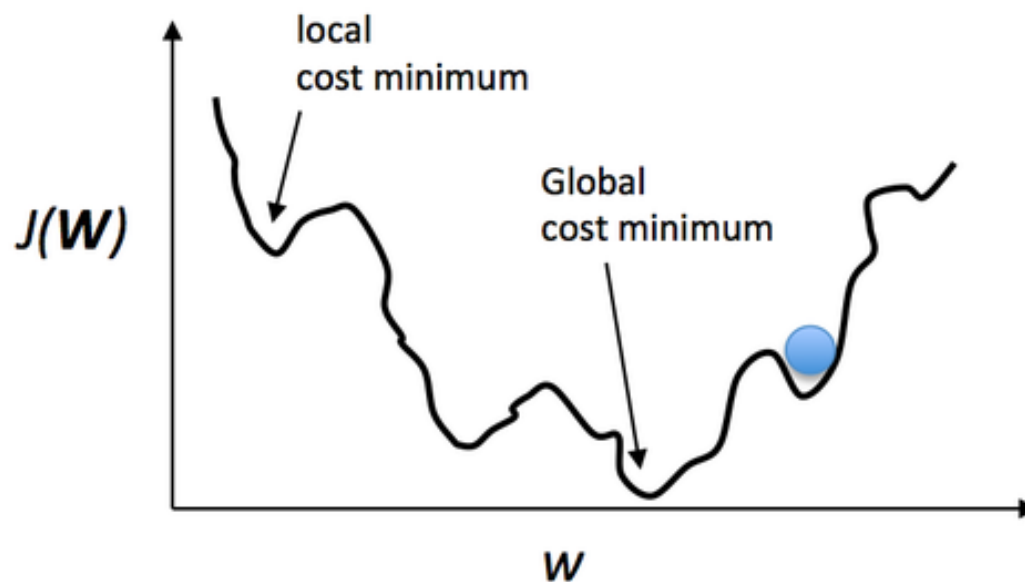
設定學習目標

任務性質

分類

什麼叫做好結果？

梯度下降方式
成本函數



設定學習目標

任務性質

分類

什麼叫做好結果？

梯度下降方式
成本函數

Stochastic Gradient Decent (SGD)

將資料分成 N 組 (batch)，每組訓練完成更新一次參數，固定的 lr

RMSprop

在梯度下降過程中採用adaptive lr (lr是待估參數)。

Adam 最熱門

RMSprop + Momentum(動能)，能幫助避免落入local minimum

```
model.compile(loss='categorical_crossentropy', optimizer=Adam(), metrics=["accuracy"])
```

<http://sebastianruder.com/optimizing-gradient-descent/index.html#momentum>

設定學習目標

任務性質

分類

什麼叫做好結果？

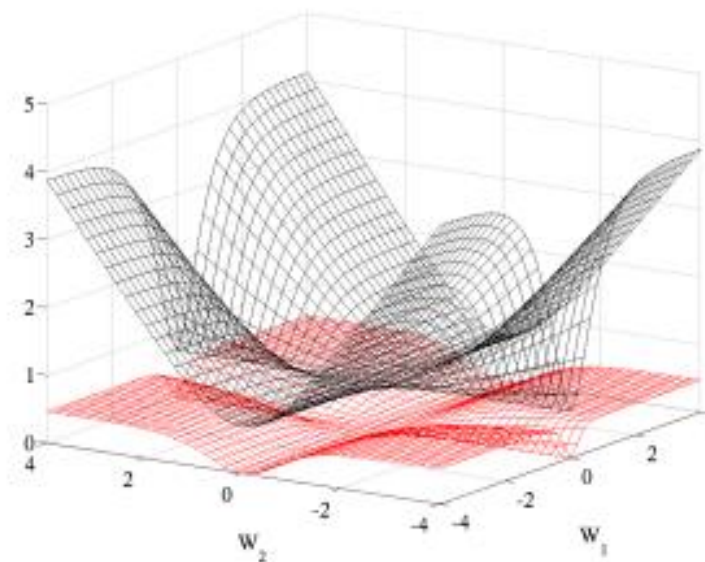
梯度下降方式
成本函數

Square Error $\sum (y_i - \hat{y}_i)^2 = 0$

```
model.compile(loss='mean_squared_error', optimizer=Adam(), metrics=['accuracy'])
```

Cross Entropy $-\sum \hat{y}_i \ln y_i = 0$

```
model.compile(loss='categorical_crossentropy', optimizer=Adam(), metrics=["accuracy"])
```



訓練，找到最佳解

調整模型參數

訓練幾輪

Activation function
其它CNN參數

```
model.fit(X_train, Y_train, batch_size=128, nb_epoch=10)
```

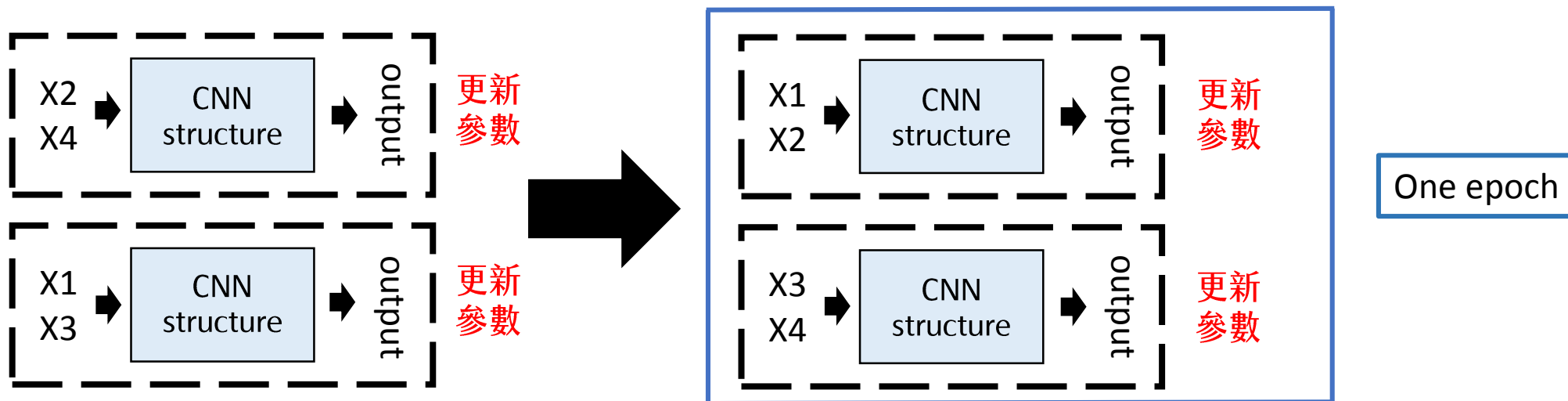
Batch size

Recall : Stochastic Gradient Decent (SGD)

將資料分成 **N 組** (batch)，每組訓練完成更新一次參數

Batch size 決定幾個樣本點為一組

EX : Input 有4張圖片，Batch size = 2，nb_epoch = 2



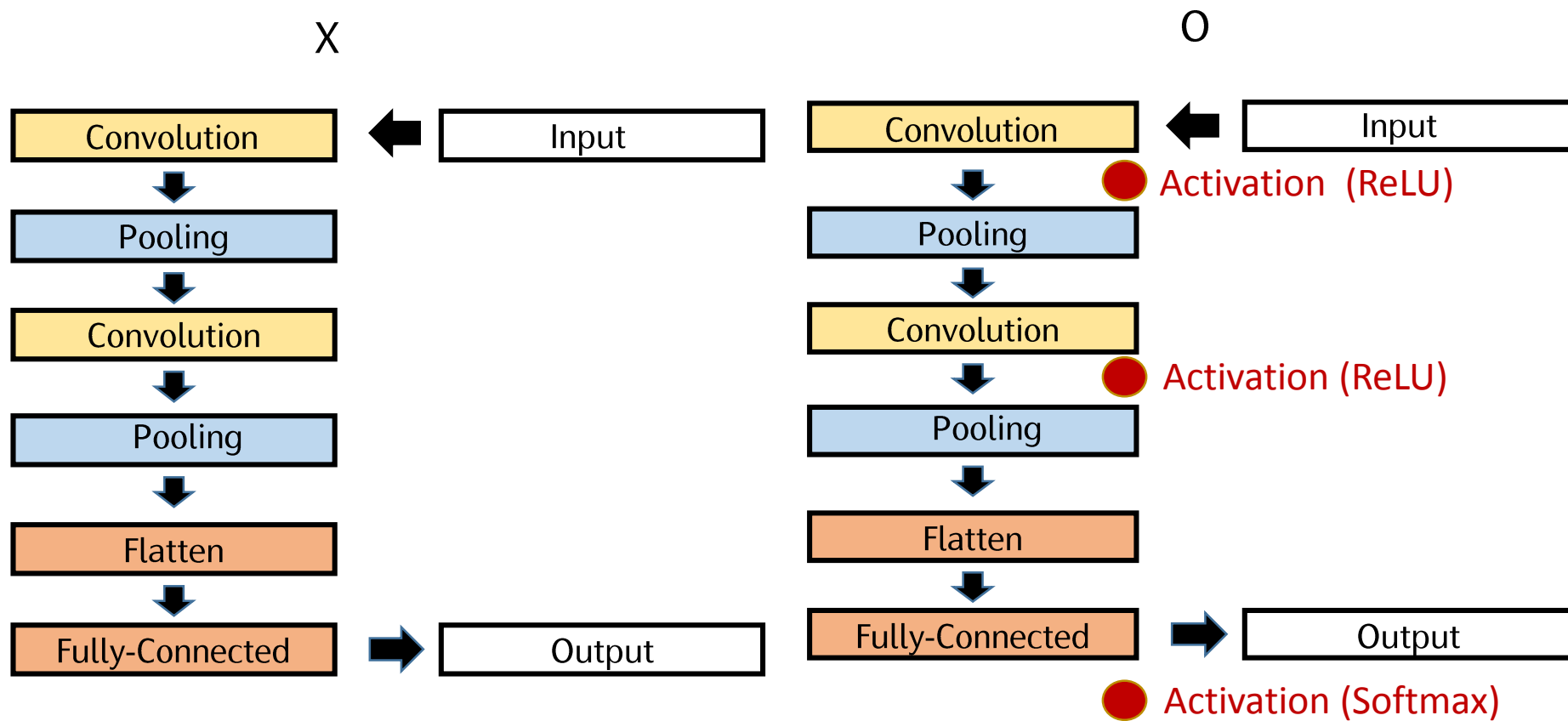
訓練，找到最佳解

調整模型參數

訓練幾輪

Activation function

其它CNN參數



訓練，找到最佳解

調整模型參數

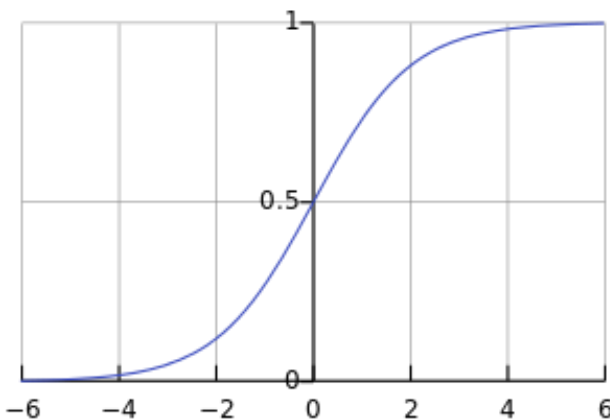
訓練幾輪

Activation function

其它CNN參數

想像成一個開關，將訊號跟刺激傳過去下一層layer
傳遞過程中，會對值進行壓縮

sigmoid



將值壓縮在 0~1 之間

由於數學上的特性，有些情況下會在
更新參數時出現問題，失去學習能力

時代的眼淚，幫QQ

訓練，找到最佳解

調整模型參數

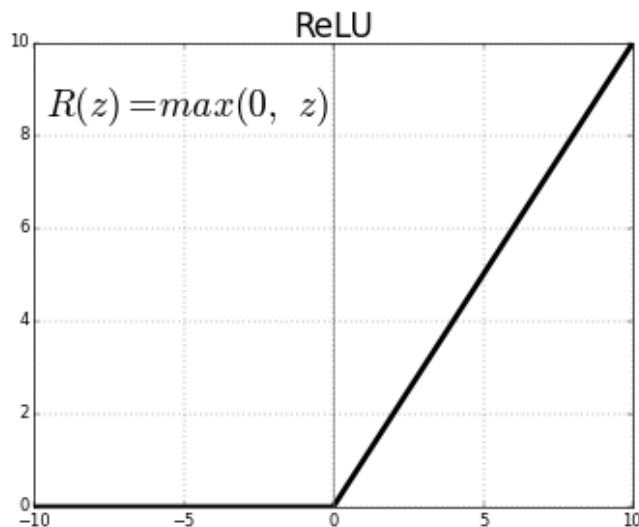
訓練幾輪

Activation function

其它CNN參數

想像成一個開關，將訊號跟刺激傳過去下一層layer
傳遞過程中，會對值進行壓縮

ReLU



速度比sigmoid快很多倍

左半邊會面臨到與sigmoid相同的問題

- ➡ 使用別人訓練好的CNN架構
- ➡ 修正過的ReLU (ex: PReLU)

當紅炸子雞

訓練，找到最佳解

調整模型參數

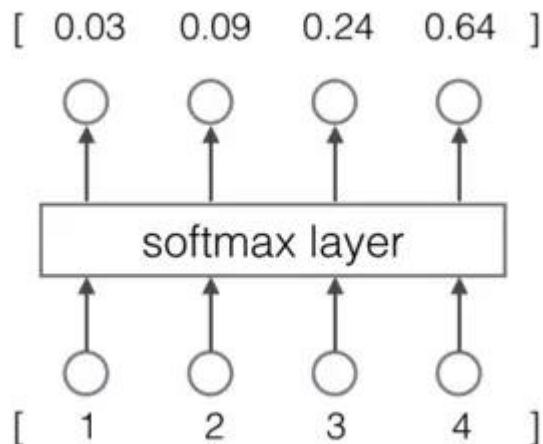
訓練幾輪

Activation function

其它CNN參數

想像成一個開關，將訊號跟刺激傳過去下一層layer
傳遞過程中，會對值進行壓縮

Softmax



通常只有在output layer才會使用
原因: <http://ppt.cc/vunLC>

將值壓縮在0~1之間
所有結點的總合為1
將數值轉化為機率的型式

訓練，找到最佳解

調整模型參數

訓練幾輪

Activation function

其它CNN參數

```
# 建造模型架構

model = Sequential()
model.add(Convolution2D(4, 5, 5, # 4個filter,每個filter大小為5*5
                        border_mode='valid', # 不使用zero-padding
                        subsample = (1,1), # stride預設為1
                        input_shape = input_shape))

model.add(Activation('relu')) # activation function

model.add(Convolution2D(8, 3, 3, # 8個filter,每個filter大小為3*3
                        border_mode='same')) # 使用zero-padding (須轉換backend為tensorflow,或安裝BLAS library)

model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(2, 2))) # filter大小為2*2

model.add(Convolution2D(16, 3, 3, border_mode='valid'))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128, init='normal')) #fully-connected layer
model.add(Activation('relu'))
model.add(Dense(10, init='normal'))
model.add(Activation('softmax'))
```

訓練，找到最佳解

調整模型參數

訓練幾輪

Activation function

其它CNN參數

定義什麼叫做學的好

```
sgd = SGD(lr=0.05,      # learning rate
          decay=1e-6,    # learning rate衰減的速度
          momentum=0.9)  # 慣性動能
```

```
model.compile(loss='categorical_crossentropy', # cost function
              optimizer='adam', # gradient decent method
              metrics=["accuracy"])
```

訓練 找出最佳解

```
model.fit(X_train, Y_train,
          batch_size=100, # 100張圖為一組
          nb_epoch=10,    # 訓練10輪
          shuffle=True, verbose=1,
          validation_split=0.2, # 5折交叉驗證
          show_accuracy=True)
```

大綱

1. CNN能用來做什麼?
2. CNN運作原理簡介
3. CNN的參數調整
4. Keras實際演練

工商時間

下週社課：教電腦玩瑪莉歐

<http://www.ithome.com.tw/news/93656>

共學讀書會