



深度學習與電腦視覺 學習馬拉松

cupay 陪跑專家：楊哲寧





深度學習理論與實作

訓練一個CNN分類器： Cifar10為例

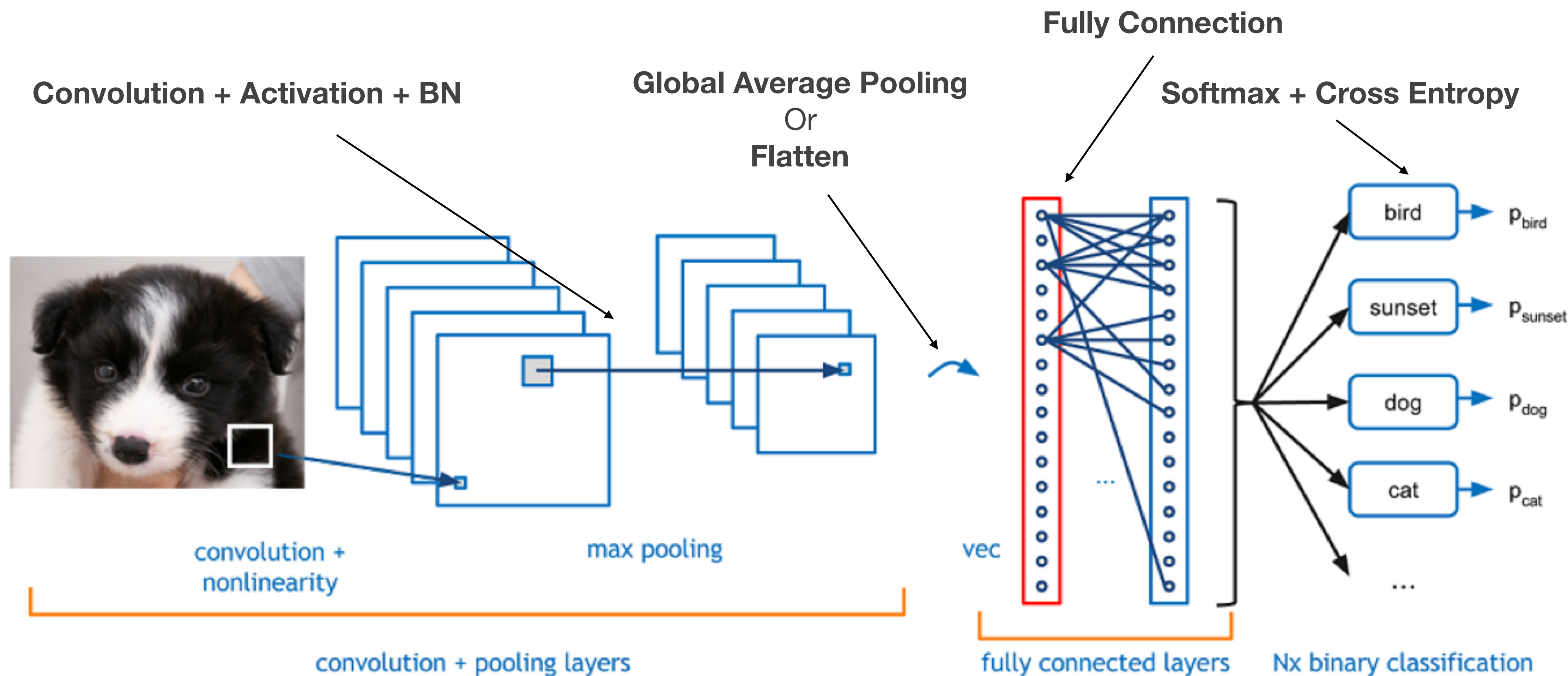
重要知識點



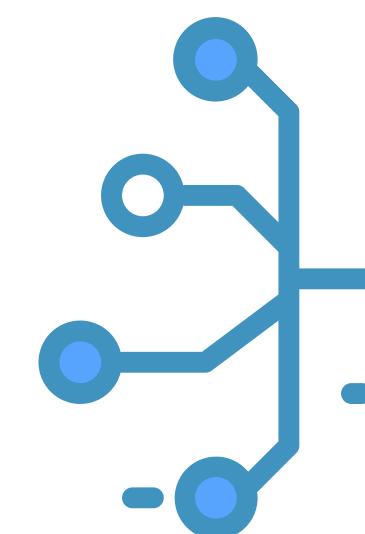
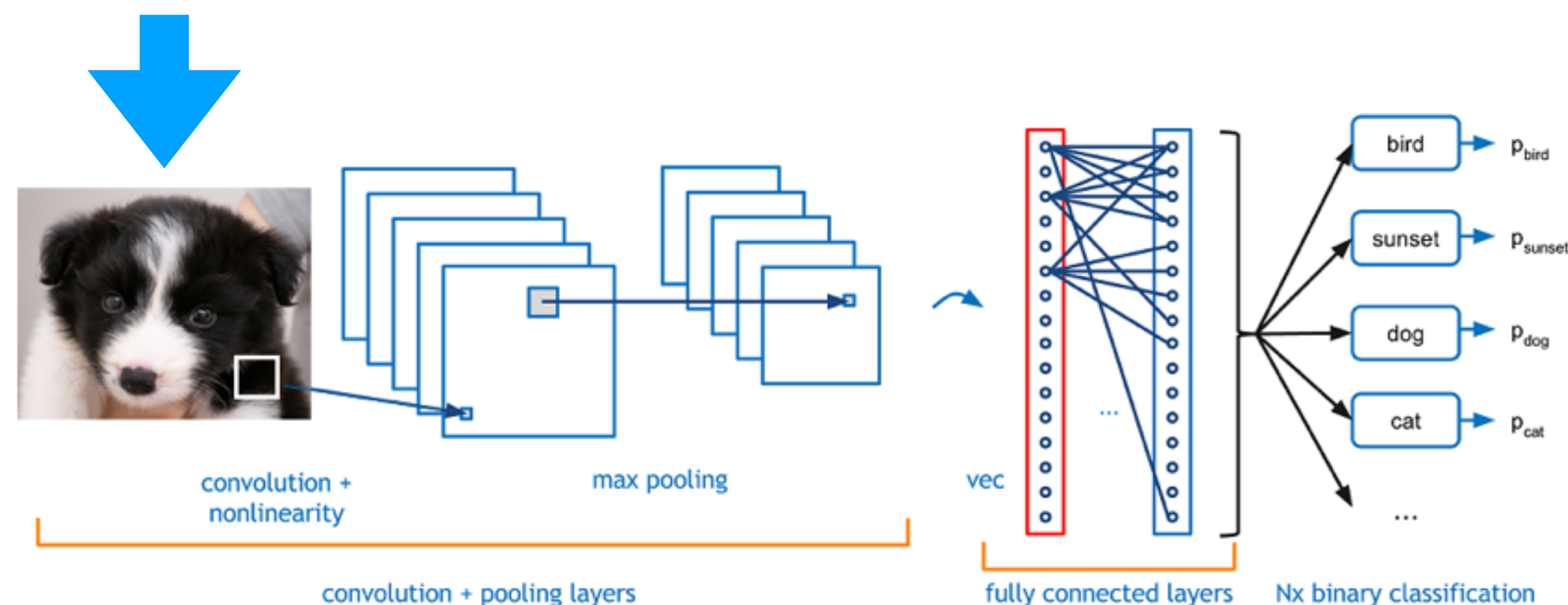
- 如何運用前幾章概念？
- 如何建造一個CNN分類器？

CNN分類器

我們先來看看下圖，一個基本的CNN圖像分類：



- 在 **Tensorflow** 與 **Keras** 中，輸入的影像整理為以下格式：
(Batch Size , Height , Width , Channels)
- 在 **Pytorch** 中，輸入的影像整理為以下格式：
(Batch Size , Channels , height , width)
- 其中 Channels 在 RGB 影像中為 3，灰階照片為 1

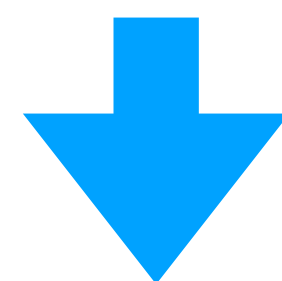




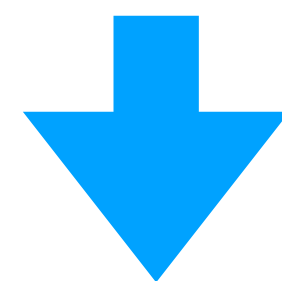
卷積層

一般來說卷積層包含：

Convolution



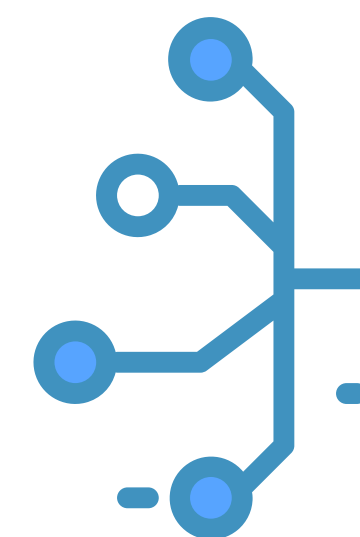
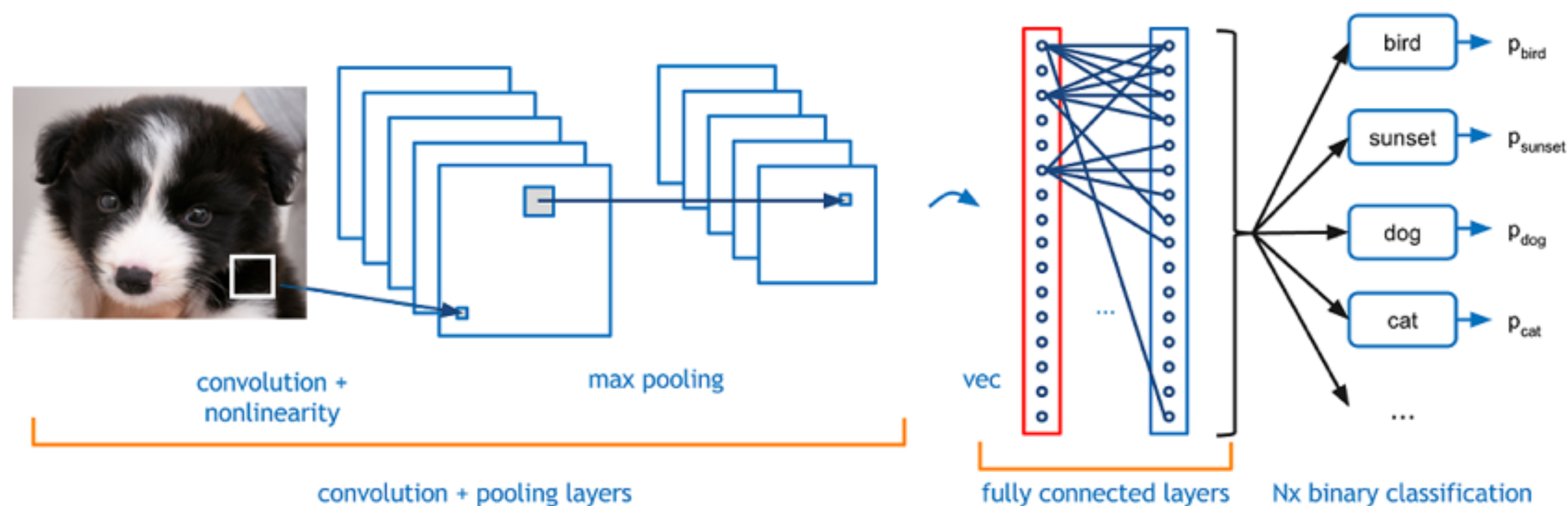
Batch Normalization

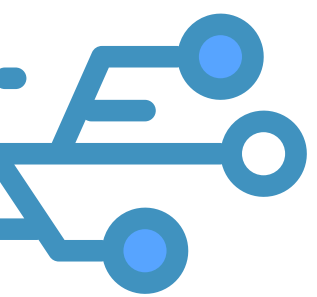


Activation Function

順序沒有一定，BN原文是加在 Convolution 後面，但隨著AF漸漸都使用ReLU，近期論文也許多是先加AF再用BN，可參考這篇論文：

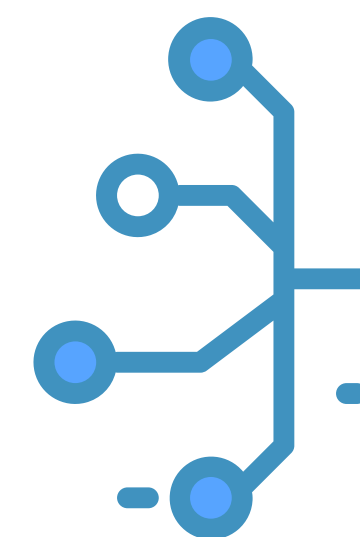
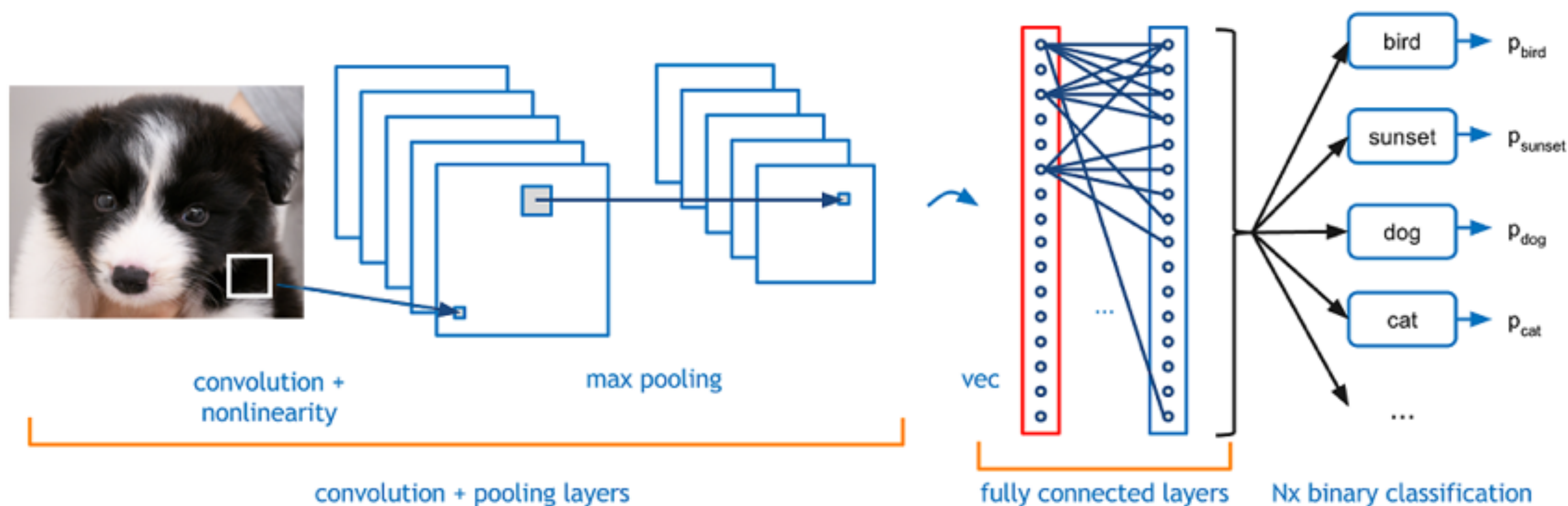
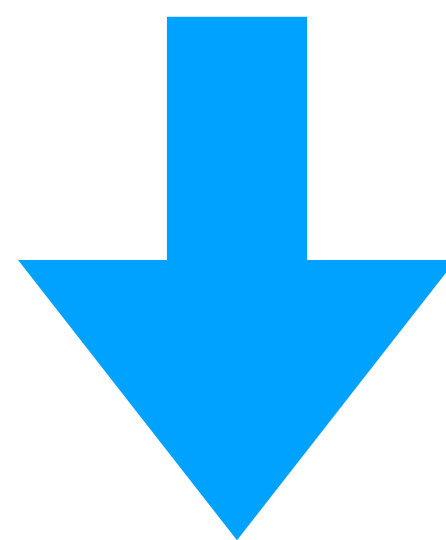
[Identity Mappings in Deep Residual Networks](#)

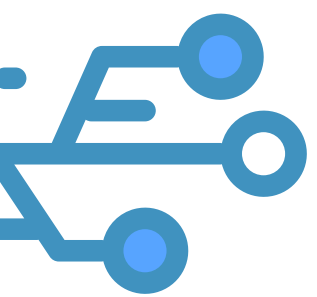




MaxPooling

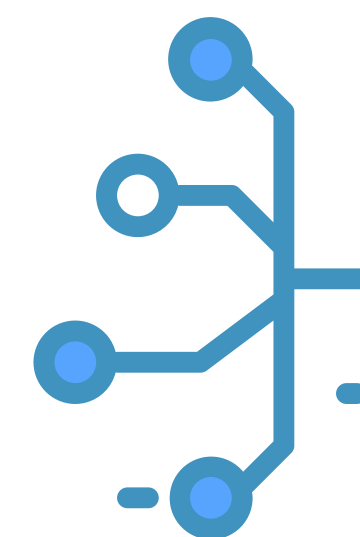
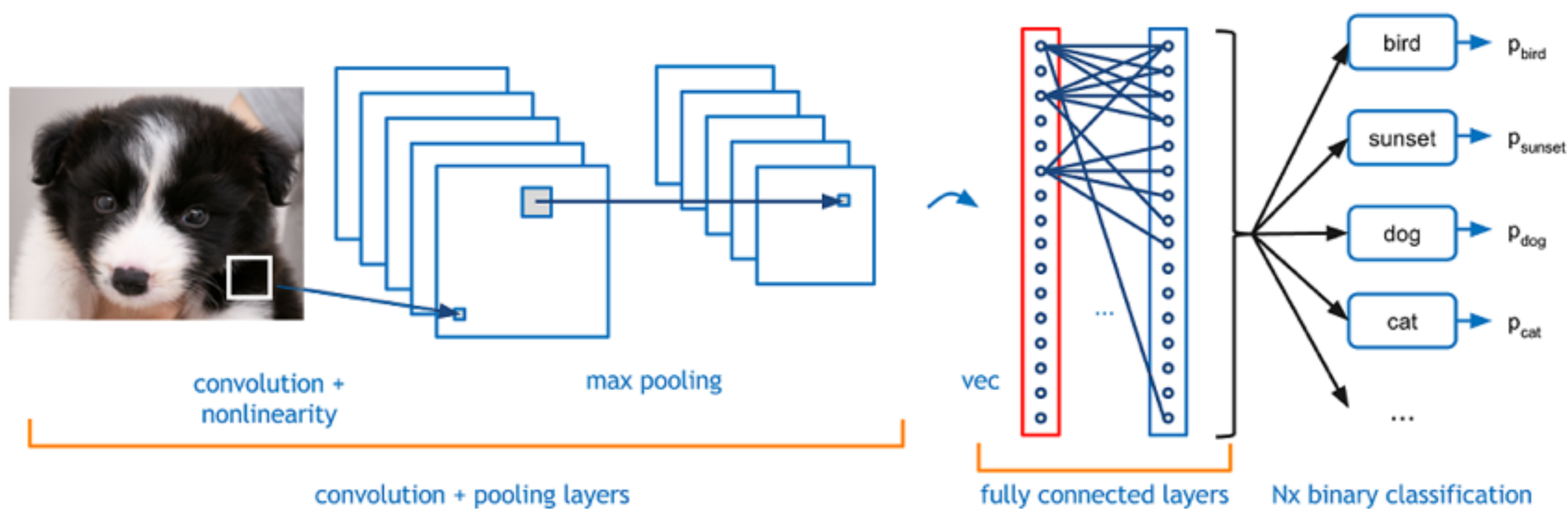
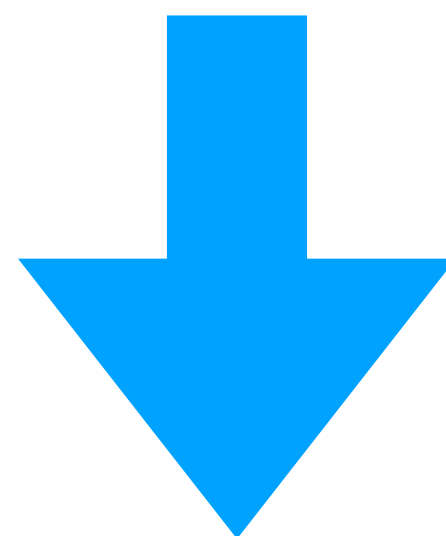
MaxPooling 通常會在幾層 Convolution 後，用來降低 **Feature Maps** 的尺度以及強化特徵。



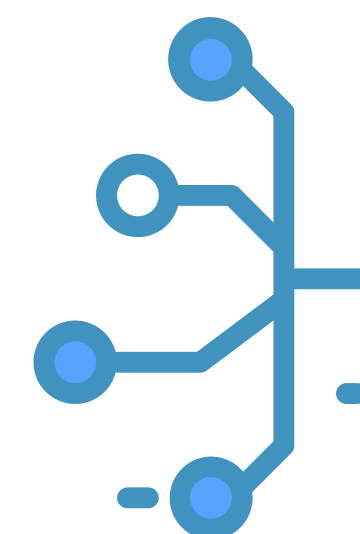
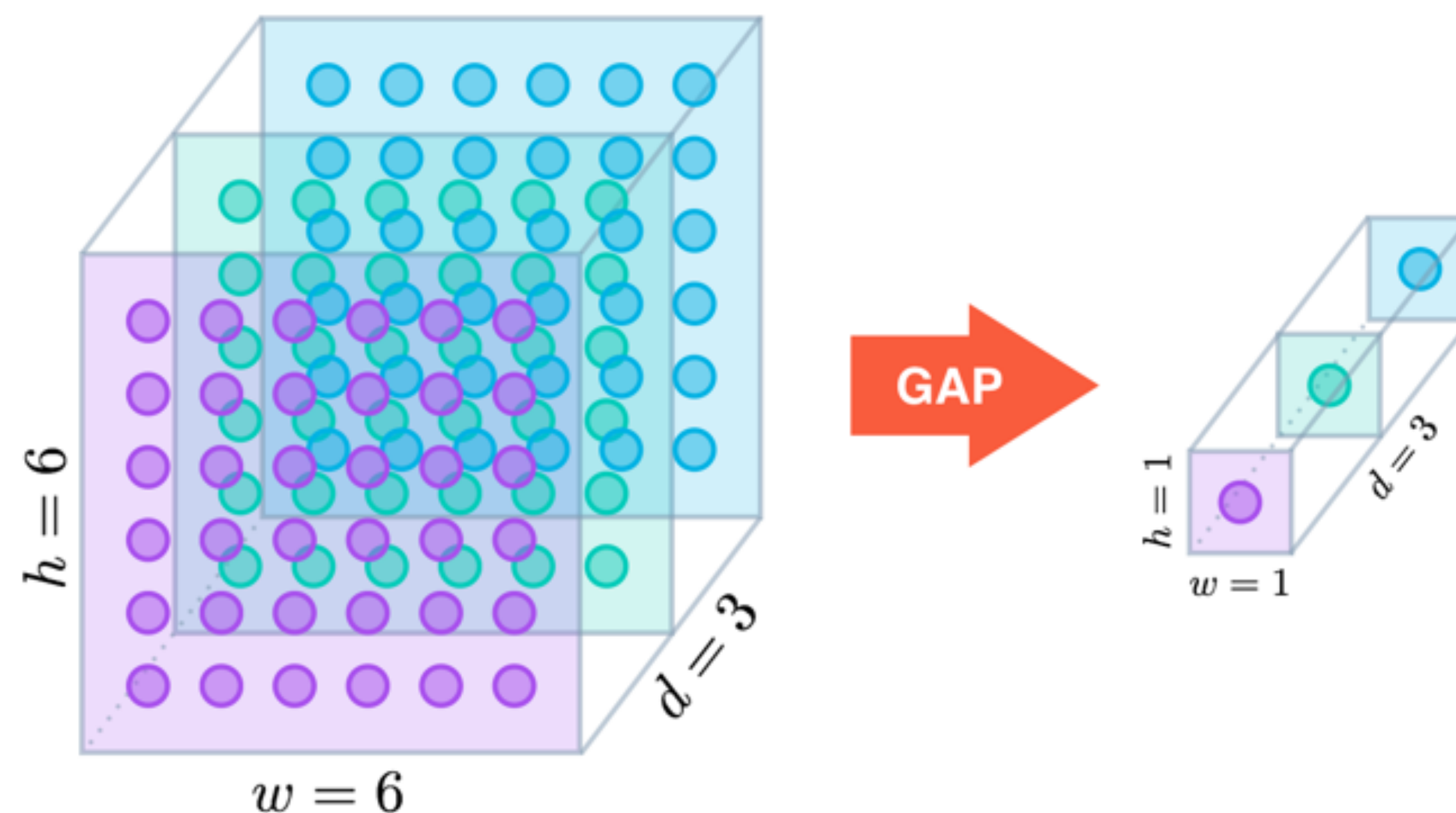


Flatten

當要將 CNN 的特徵輸入 FC 時，我們透過 Flatten 或是 Global Average Pooling。



Global Average Pooling (GAP) 就是將每張Feature Map上的資訊以平均的方式壓為一個值。



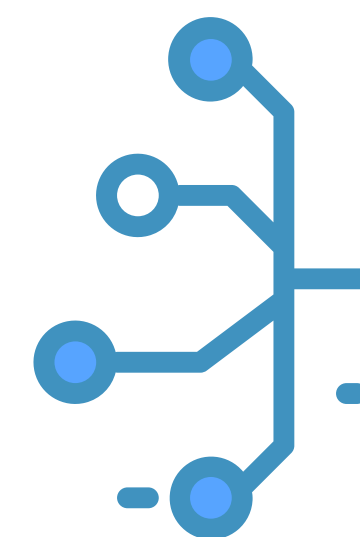
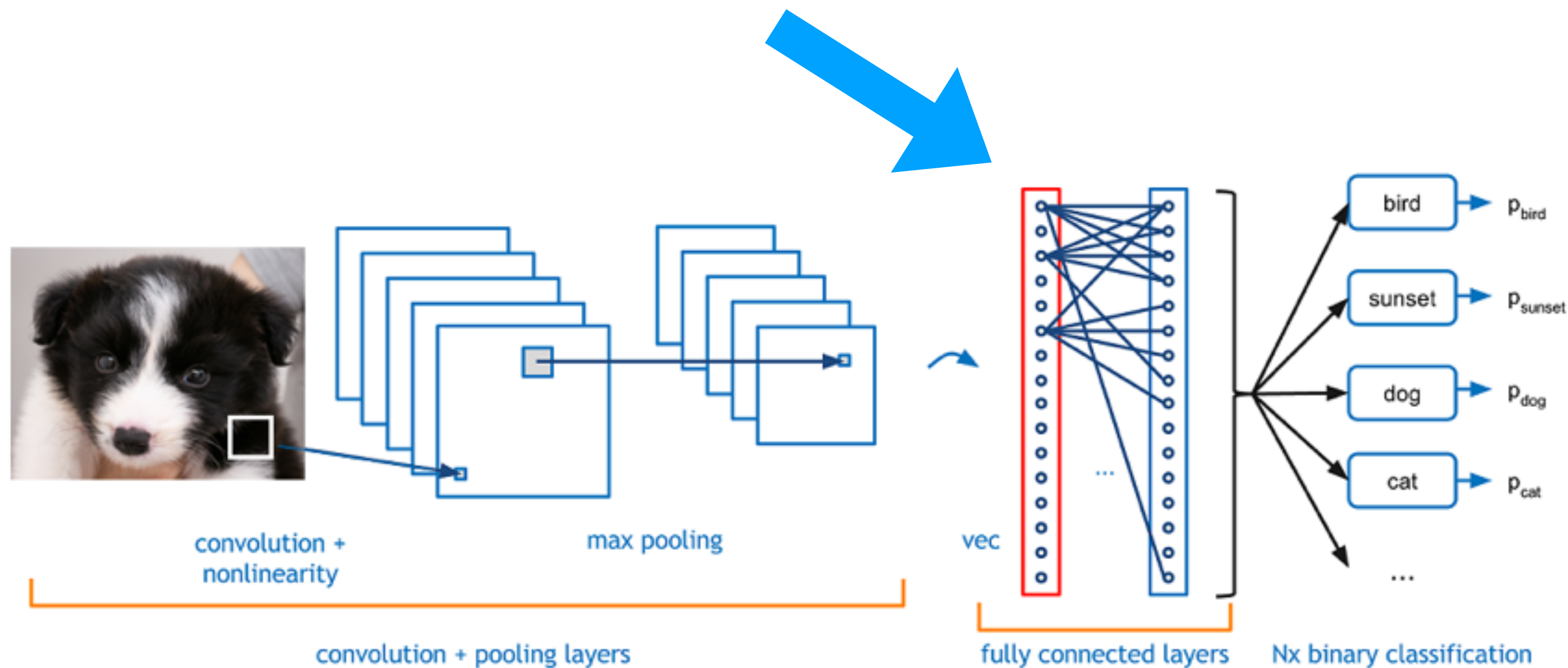


Fully Connected Layers

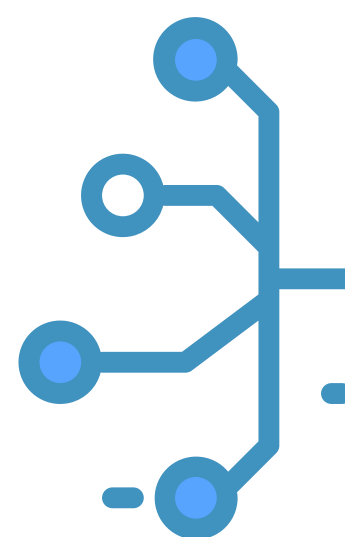
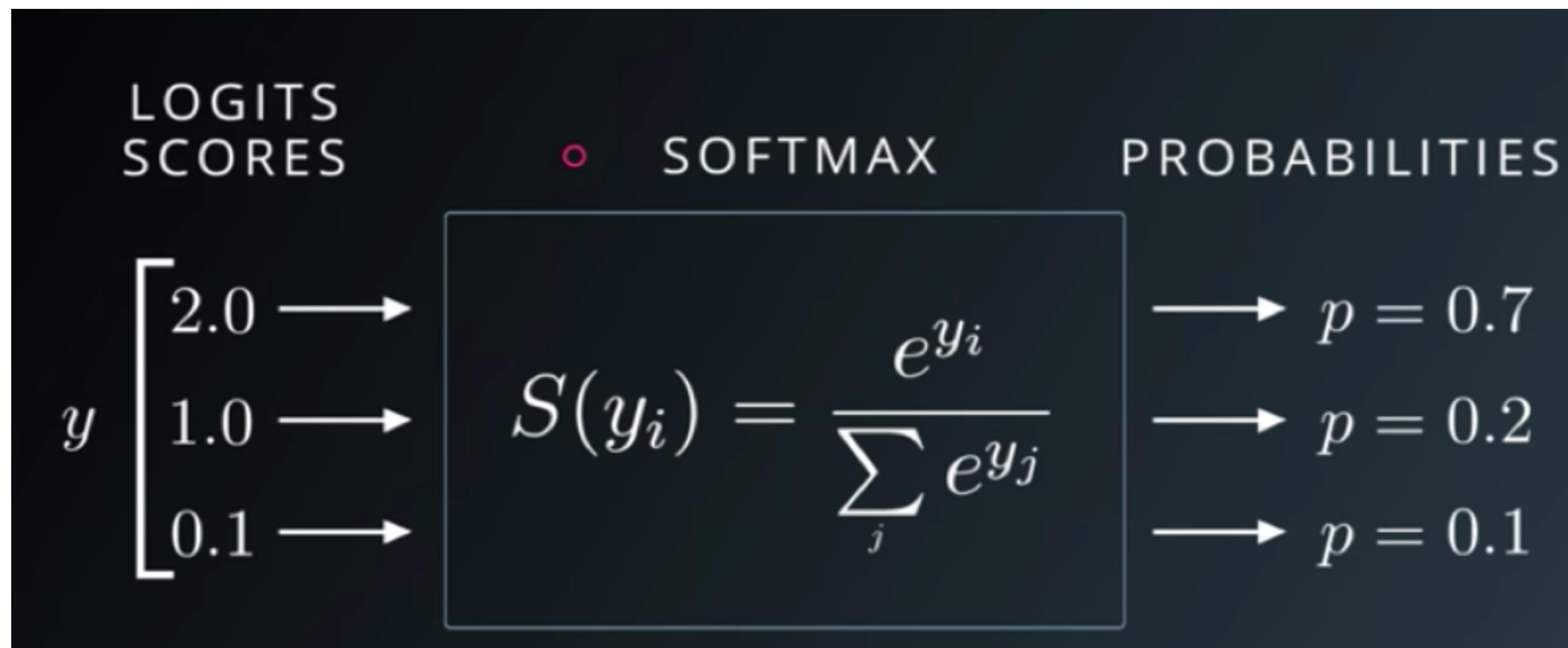


CUPOY

近期文獻偏好使用越少FC層越好，主要是由於FC層容易產生大量參數，FC層的最後一層要使用與預分類類別一樣多的神經元，當作各個類別的輸出特徵值，並透過Softmax轉換成機率值。



Softmax函數能將輸出值總合統整成1，轉換成機率的型態，通常用於多類的分類器輸出。





推薦延伸閱讀



K Keras Documentation

Search docs

Home

Why use Keras

GETTING STARTED

Guide to the Sequential model

Guide to the Functional API

FAQ

MODELS

About Keras models

Sequential

Model (functional API)

LAYERS

About Keras layers

Core Layers

Convolutional Layers

Pooling Layers

Locally-connected Layers

Recurrent Layers

Embedding Layers

Merge Layers

Advanced Activations Layers

Normalization Layers

Noise layers

GitHub

« Previous

Next »

Docs » Examples » CIFAR-10 CNN

Edit on GitHub

Train a simple deep CNN on the CIFAR10 small images dataset.

It gets to 75% validation accuracy in 25 epochs, and 79% after 50 epochs. (it's still underfitting at that point, though).

```
from __future__ import print_function
import keras
from keras.datasets import cifar10
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D
import os

batch_size = 32
num_classes = 10
epochs = 100
data_augmentation = True
num_predictions = 20
save_dir = os.path.join(os.getcwd(), 'saved_models')
model_name = 'keras_cifar10_trained_model.h5'

# The data, split between train and test sets:
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

# Convert class vectors to binary class matrices.
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=x_train.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
```

Keras教學：

Train a simple deep CNN on the CIFAR10 small images dataset.

連結

解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題