卷積神經網路 Convolutional Neural Network
&
電腦視覺 Computer Vision
Part2

林彥宇 & 教研處

「版權聲明頁」

本投影片已經獲得作者授權台灣人工智慧學校得以使用於教學用途，如需取得重製權以及公開傳輸權需要透過台灣人工智慧學校取得著作人同意；如果需要修改本投影片著作，則需要取得改作權；另外，如果有需要以光碟或紙本等實體的方式傳播，則需要取得人工智慧學校散佈權。

–台灣人工智慧學校

本日課程內容

本日課程:

1. CNN in TensorFlow

延伸閱讀 (Optional):

1. Deep Learning Summarization

本次課程結束後你 (妳) 應該會什麼?

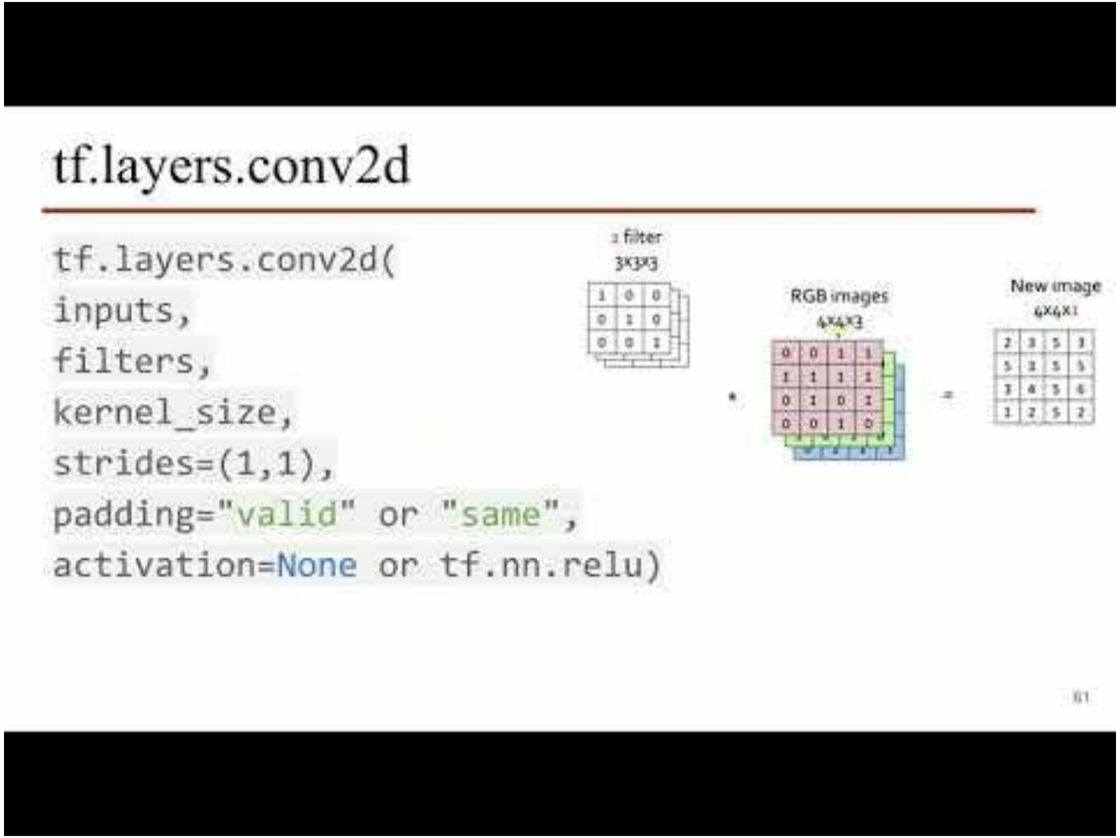
- **軟實力**
  - 了解 CNN在TensorFlow落實的原理以及其參數設定
- **硬底子**
  - 如何用 TensorFlow 寫出基本的 CNN
  - 完成 Computer Vision 中的經典資料集 cifar-10

Code / Data 放在 hub 中的 courses 內

- 為維護課程資料，courses 中的檔案皆為 read-only, 如需修改請 cp 至自身的環境中
- 打開 terminal, 輸入
    - cp -r courses-tpe/CVCNN/part2 <存放至本機的名稱>

CNN in TensorFlow

[實戰演練]CNN in TensorFlow:



tf.layers.conv2d

```
tf.layers.conv2d(
inputs,
filters,
kernel_size,
strides=(1,1),
padding="valid" or "same",
activation=None or tf.nn.relu)
```

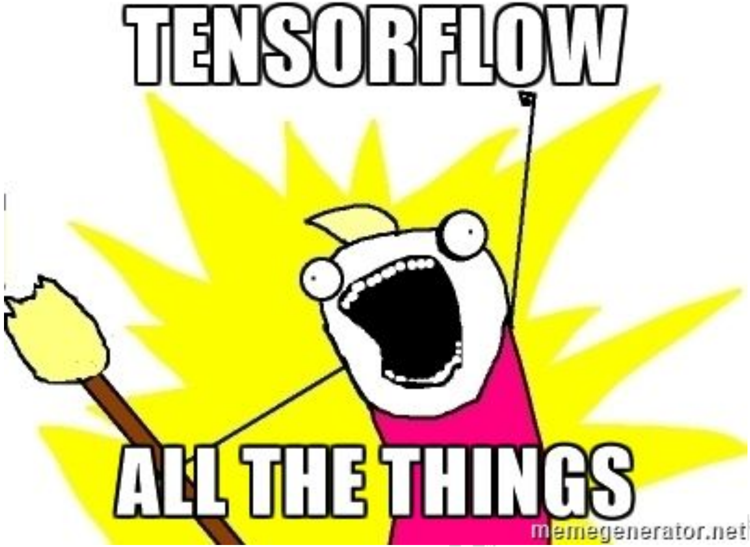
1	0	0
0	1	0
0	0	1

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

2	3	5	3
5	3	5	5
3	4	5	6
1	2	5	2

CNN in TensorFlow

- 前面課程提到的包含 convolution, pooling, padding, activation, flatten 等等，這些 TensorFlow 都幫你寫好了！只要學會如何使用以及了解裡面參數的意義就沒問題囉

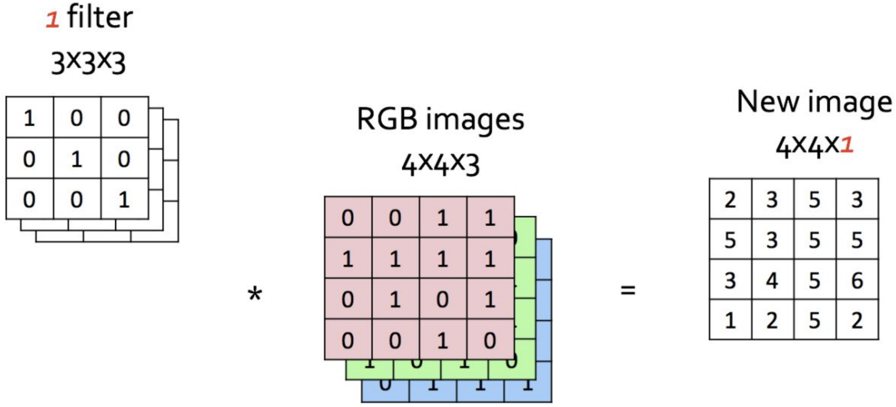


convolution

```
import tensorflow as tf

## assume we have a x_tensor with shape=(7,7,3)

n_filters = 32

conv_result = tf.layers.conv2d(x_tensor, n_filters)

## the shape of output will be (7,7,32) with same padding,
## strides=1, kernel_size=(3,3)
```

tf.layers.conv2d

```
tf.layers.conv2d(
inputs,
filters,
kernel_size,
strides=(1,1),
padding="valid" or "same",
activation=None or tf.nn.relu)
```



1	0	0
0	1	0
0	0	1

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

2	3	5	3
5	3	5	5
3	4	5	6
1	2	5	2

maxpooling

```
import tensorflow as tf

## assume we have a x_tensor with shape=(8,8,3)

conv_layer = tf.layers.max_pooling2d(x_tensor,
pool_size=(2,2))

## the shape of output will be (4,4,3) with no padding,
strides=2
```

tf.layers.max_pooling2d

```
tf.layers.max_pooling2d(
inputs,
pool_size,
strides=,
padding="valid" or "same")
```

CIFAR-10 Dataset

- 60,000 (50,000 training + 10,000 testing) samples, 32x32 color images in 10 classes
- 10 classes
  - airplane, automobile, ship, truck, bird, cat, deer, dog, frog, horse
- Official website
  - https://www.cs.toronto.edu/~kriz/cifar.html

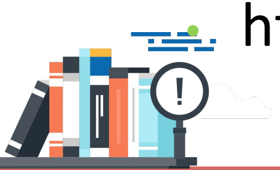
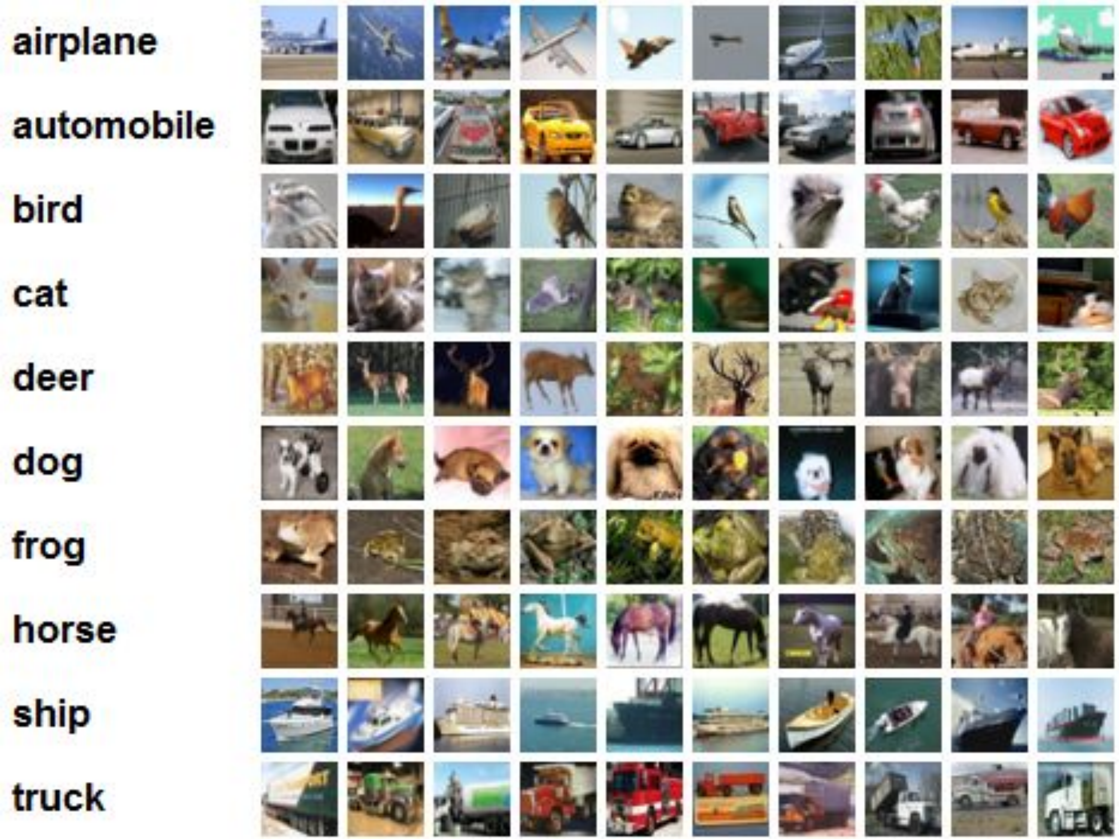
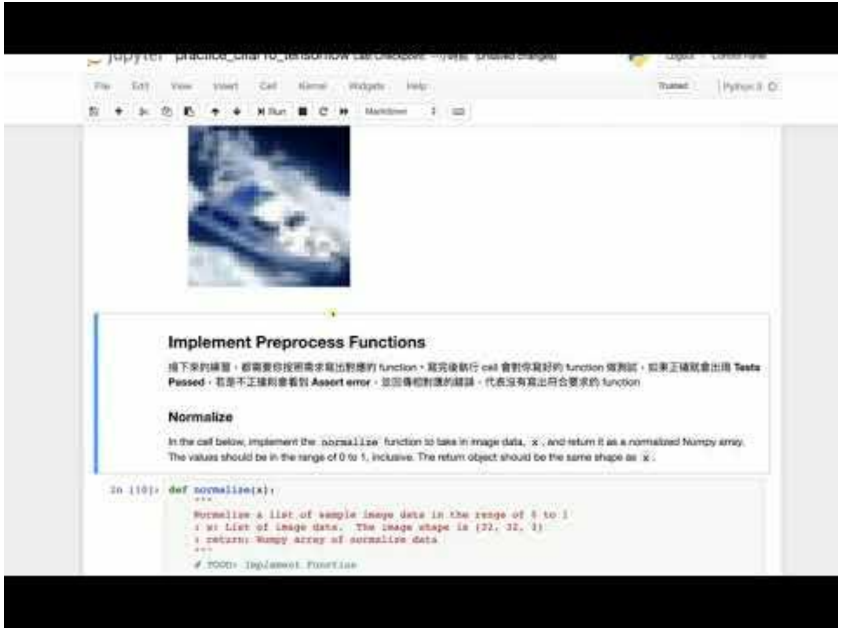


Image classification



練習時間

- 請參考 part2/01_cnn_in_tensorflow/practice_cifar10_tensorflow.ipynb，完成自己的第一個 CNN in TensorFlow!
- 有任何問題歡迎隨時詢問助教，已經完成的同學也歡迎在mattermost中分享自己的做法



Implement Preprocess Functions

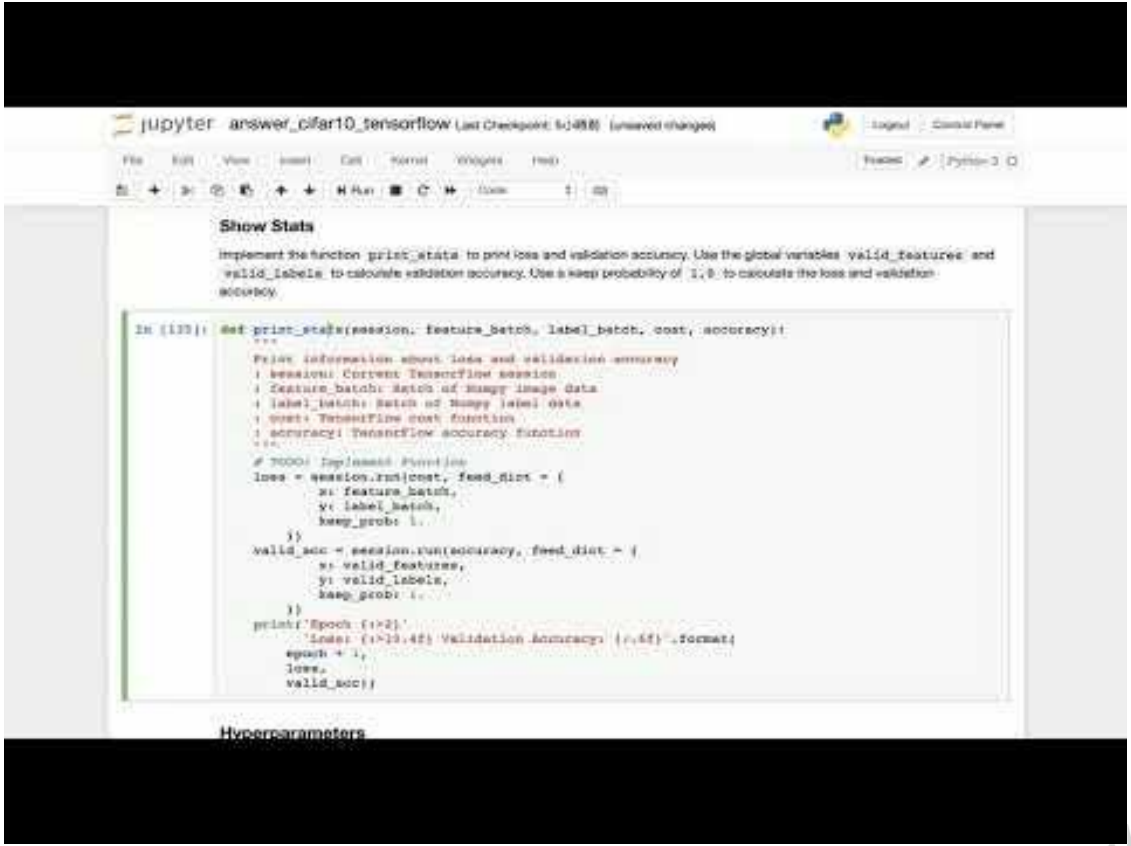
接下來的練習，都需要你按照需求寫出對應的 function，寫完後執行 cell 會對你寫好的 function 做測試，如果正確就會出現 **Tests Passed**，若是不正確則會看到 **Assert error**，並回傳相對應的錯誤，代表沒有寫出符合要求的 function

Normalize

In the cell below, implement the `normalize` function to take in image data, `x`, and return it as a normalized Numpy array. The values should be in the range of 0 to 1, inclusive. The return object should be the same shape as `x`.

```
In [10]: def normalize(x):
             """
             Normalize a list of sample image data in the range of 0 to 1
             : x: List of image data.  The image shape is (32, 32, 3)
             : return: Numpy array of normalize data
             """
             # TODO: Implement Function
```

練習題解答



jupyter answer_cifar10_tensorflow Last Checkpoint: [illegible] (unsaved changes)
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

Show Stats

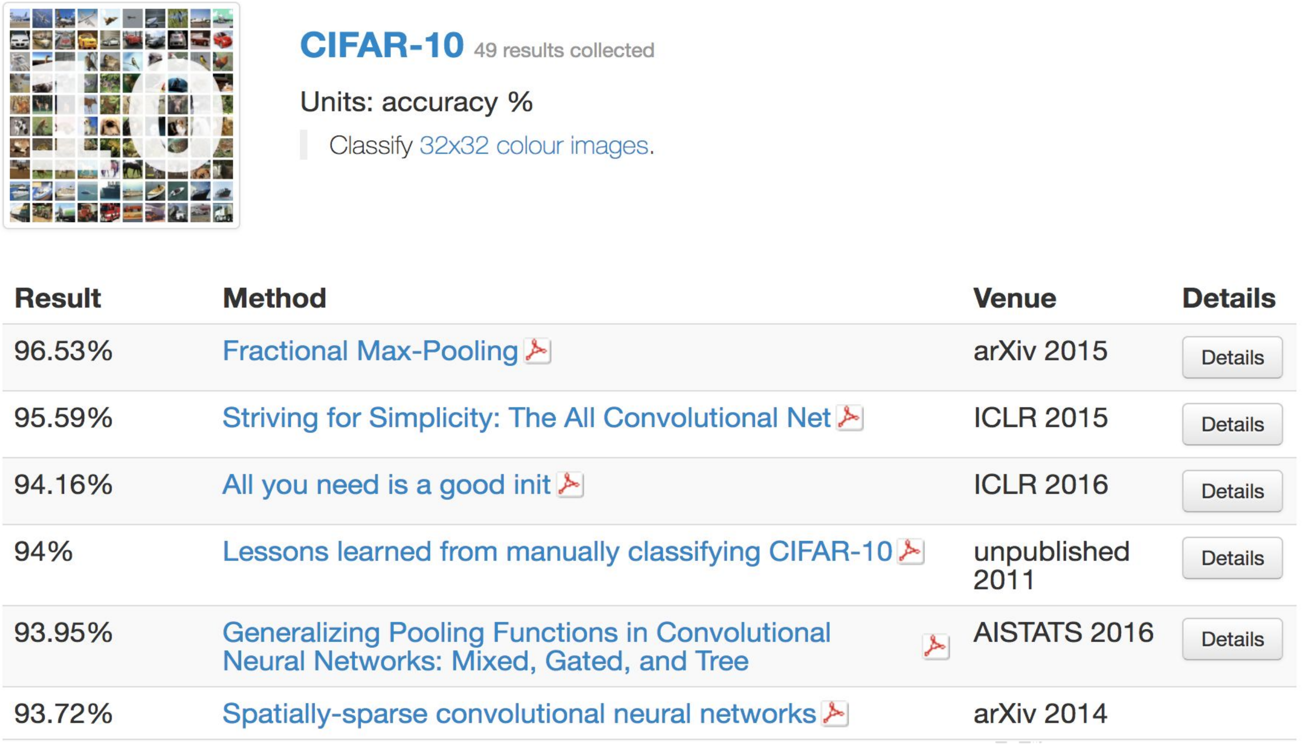
Implement the function `print_stats` to print loss and validation accuracy. Use the global variables `valid_features` and `valid_labels` to calculate validation accuracy. Use a keep probability of `1.0` to calculate the loss and validation accuracy.

```
In [133]: def print_stats(session, feature_batch, label_batch, cost, accuracy):
              """
              Print information about loss and validation accuracy
              : session: Current TensorFlow session
              : feature_batch: Batch of Numpy image data
              : label_batch: Batch of Numpy label data
              : cost: TensorFlow cost function
              : accuracy: TensorFlow accuracy function
              """
              # TODO: Implement Function
              loss = session.run(cost, feed_dict = {
                          x: feature_batch,
                          y: label_batch,
                          keep_prob: 1.
                  })
              valid_acc = session.run(accuracy, feed_dict = {
                          x: valid_features,
                          y: valid_labels,
                          keep_prob: 1.
                  })
              print('Epoch {:>2}'
                    'Loss: {:>10.4f} Validation Accuracy: {:.6f}'.format(
                  epoch + 1,
                  loss,
                  valid_acc))
```

Hyperparameters

Who is the best in CIFAR-10

- 此網頁紀錄世界各地高手在 CIFAR-10 跑出來的結果與 paper



CIFAR-10 49 results collected

Units: accuracy %

> Classify 32x32 colour images.

Result	Method	Venue	Details
96.53%	Fractional Max-Pooling	arXiv 2015	Details
95.59%	Striving for Simplicity: The All Convolutional Net	ICLR 2015	Details
94.16%	All you need is a good init	ICLR 2016	Details
94%	Lessons learned from manually classifying CIFAR-10	unpublished 2011	Details
93.95%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
93.72%	Spatially-sparse convolutional neural networks	arXiv 2014	

讀書會徵求

- CNN 發展至今，最廣泛使用的就是 ResNet，後續還有延伸的模型如 ResNext, Resnet-inception, DenseNet, DPN等等，太多模型無法一一分享
- 如果可以透過讀書會的方式進行，彼此認領一至兩篇最新的CNN paper，可以在最短時間內理解最新的模型架構

CNN 必須要 GPU 才能跑的快

- CNN 處理的參數量非常大，沒有強大的 GPU 做矩陣運算，速度會非常的慢

- 若希望在本機上使用 GPU，須注意 Nvidia 顯卡不能太舊，自行安裝 tensorflow-gpu, CUDA, CuDNN 後就可以使用 GPU 囉！參考教學

延伸閱讀-1

Deep learning summarization

Recap – Fundamentals

- Fundamentals of deep learning
  - A neural network = a function
  - Gradient descent
  - Stochastic gradient descent
  - Mini-batch
  - Guidelines to determine a network structure

Recap – Improvement on Training Set

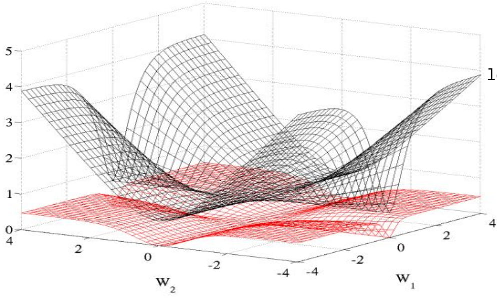
- How to improve performance on training dataset

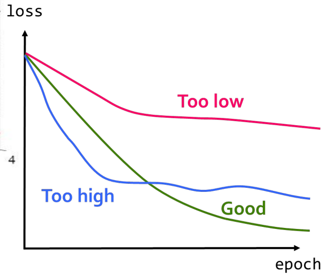
Loss Function

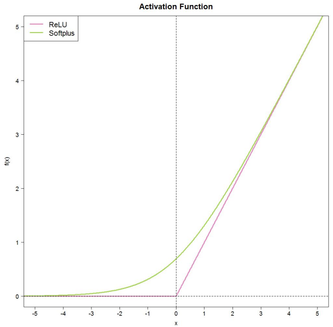
Learning Rate

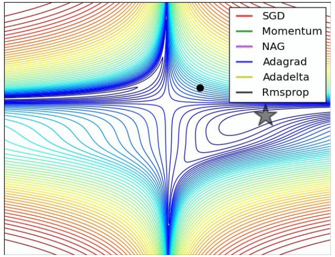
Activation Function

Optimizer









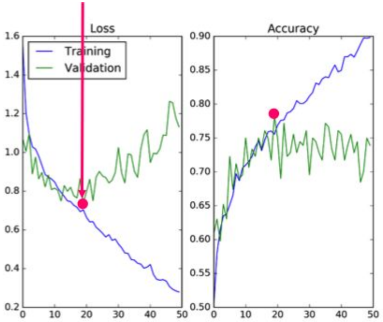
Recap – Improvement on Testing Set

- How to improve performance on testing dataset

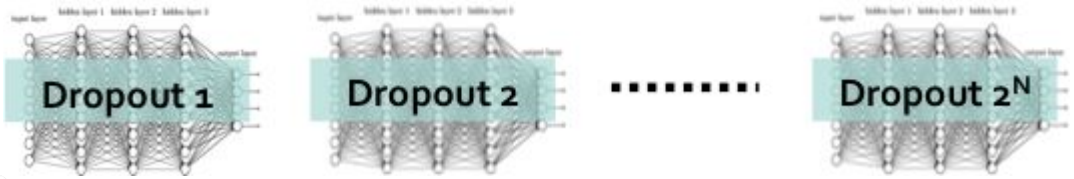
Regularization

Early Stopping

Dropout

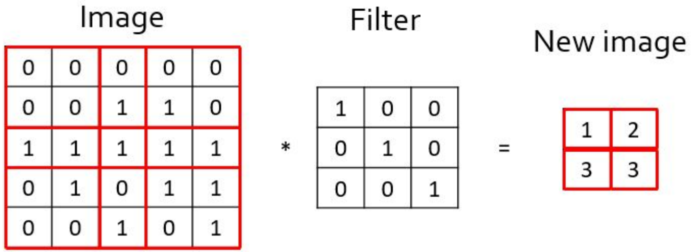


$$\mathrm{Loss}_{reg}=\sum(y-(b+\sum w_i x_i))+\alpha(\text{regularizer})$$



Recap – CNN

- Fundamentals of CNN
  - Concept of filters
  - Hyper-parameters
    - Filter size
    - Zero-padding
    - Stride
    - Depth (total number of filters)
  - Pooling layers
- How to train a CNN in Keras
  - CIFAR-10 dataset



$$\underset{\text{Image}}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}} * \underset{\text{Filter}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} = \underset{\text{New image}}{\begin{bmatrix} 1 & 2 \\ 3 & 3 \end{bmatrix}}$$