

Convolutional Neural Network for Image Recognition

前處理

圖片大小	128 x 128
正則化	0~255 → 0~1
Training 筆數	10000
Testing 筆數	4000

前處理方面，因為有 10000 張 128x128 的圖片，如果全部載入進來，RAM 必然會爆滿，所以必需分批進入。

1. 圖片大小平均是 100~200，所以取 128x128，一來參數比較方便計算，二來這個大小不會令較小的圖片失真，令 NN 能更好學習。
2. glob 套件載入全部圖片的路徑，並安排好圖片對應的 Labels
3. 撰寫了一個 Iterator，每次 iterate 一個 batch size 大小的圖片集 (Shape = (batch_size, 128, 128, 3))
4. Training iterator 需要 shuffle 而 Test Iterator 則不用。
5. 圖片先用 Image 套件硬轉成 RGB 制式，因為有些 PNG 是 RGBA 制式，即(128, 128, 4)。
6. Pytorch 的 Conv2d input 格式為(batch_size, channels, h, w)，所以要用 swapaxes 把 channels 數放到前面

網路參數設定

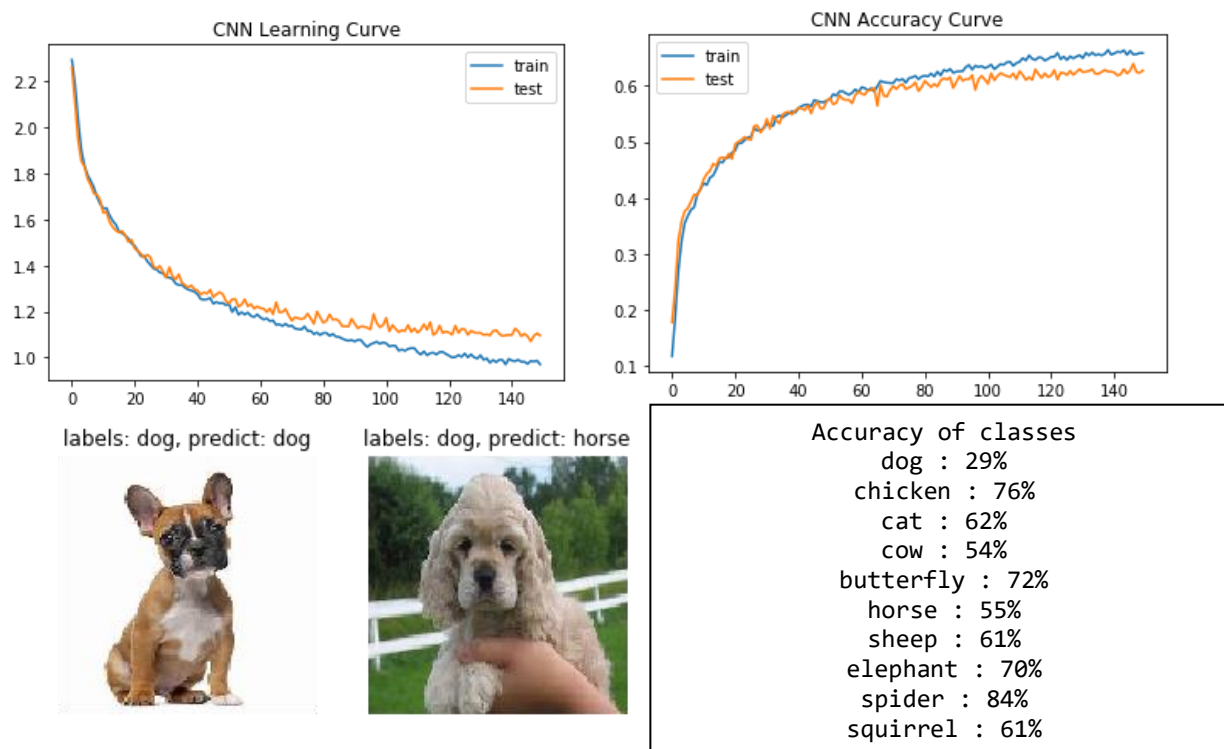
Loss	Cross entropy
Optimizer	Adam with Learning rate 0.001
Epochs	150
Batch size	200

第一個網路架構

類型	Activation	Neurons
Conv2D	ReLU	64 * 3x3 filter size
MaxPool2D	/	2x2 Pooling
Dropout	/	0.25 dropout rate
Conv2D	ReLU	32 * 3x3 filter size
MaxPool2D	/	2x2 Pooling
Dropout	/	0.25 dropout rate
Conv2D	ReLU	16 * 3x3 filter size
MaxPool2D	/	2x2 Pooling
Dropout	/	0.25 dropout rate
Conv2D	ReLU	8 * 3x3 filter size
MaxPool2D	/	2x2 Pooling
Dropout	/	0.25 dropout rate
Flatten	/	/
DNN	Softmax	10

第一個 CNN 由四個 Convolution layer 及一層輸出 DNN 組成，而每層 CNN 加入 Max Pooling（強化特徵）及 Dropout（防止 Overfitting）。而每層 CNN 的 Filter 數會遞減至上一層的一半，因為越後層的 CNN 能讀到 Feature map 已經很多信息，毋需太多 Filter，否則會嚴重 Overfitting。

結果



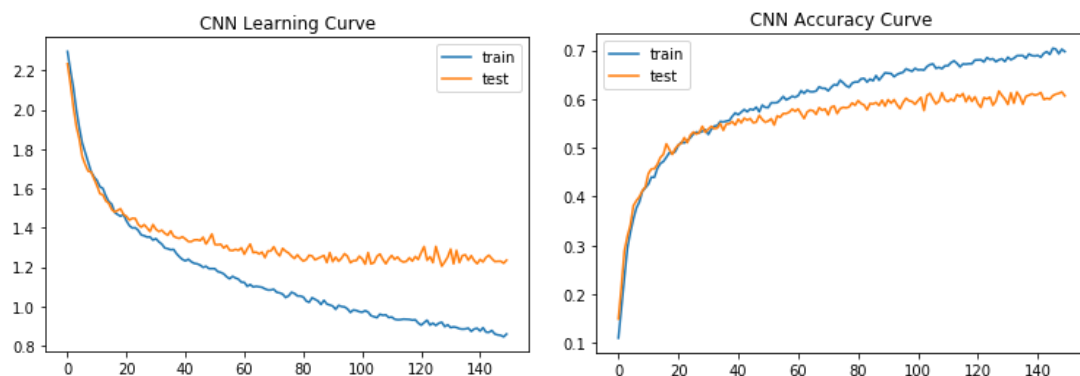
尚未出現 Overfitting，但更多 Epochs 的話相信會出現。Test accuracy 大概停留於 0.6 左右徘徊。第一個架構除了在猜 dog 的類別比平均差外，其他皆在平均附近。

第二個網路架構

類型	Activation	Neurons
Conv2D	ReLU	64 * 4x4 filter size Stride = 2
Dropout	/	0.25 dropout rate
Conv2D	ReLU	32 * 4x4 filter size Stride = 2
Dropout	/	0.25 dropout rate
Conv2D	ReLU	16 * 4x4 filter size Stride = 2
Dropout	/	0.25 dropout rate
Conv2D	ReLU	8 * 4x4 filter size Stride = 2
Dropout	/	0.25 dropout rate
Flatten	/	/
DNN	Softmax	10

沿用第一種架構，但 Filter size 改成 4x4 的 filter，Stride 改成 2 而且棄用 MaxPool2d，因為 MaxPool2d 其實有機會棄用掉一些圖片上有用的資訊，所以改成 Stride 2 去代替 MaxPool2d 可以解決 MaxPool 造成的稀疏性問題

結果



labels: dog, predict: chicken



labels: dog, predict: dog



Accuracy of classes

- dog : 41%
- chicken : 67%
- cat : 70%
- cow : 51%
- butterfly : 67%
- horse : 54%
- sheep : 56%
- elephant : 65%
- spider : 73%
- squirrel : 58%

收斂比第一個架構快 (Train loss 下降更多)，而 Test accuracy 亦在 0.6 左右徘徊
第二個架構對於每個類別都有較平均的命中率，尤其是 dog 的類別由 29% 上升至 41%

Recurrent Neural Network for Prediction of Paper Acceptance

前處理

字典長度	10
字典字數	2539 (不包含 0 及 Other)
Training 筆數	1250
Testing 筆數	85

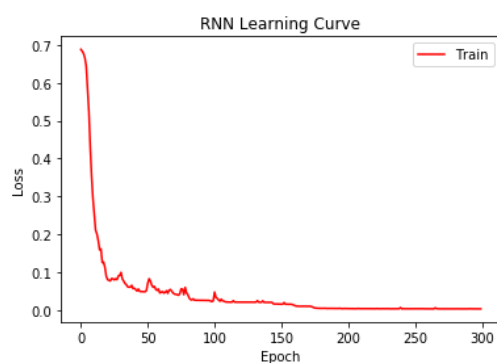
網路架構

類型	Activation	Neurons
Embedding	Linear	30
RNN/LSTM	Tanh	3 layers, 30 nodes each
DNN	Tanh	16
DNN	Tanh	8
DNN	Softmax	2

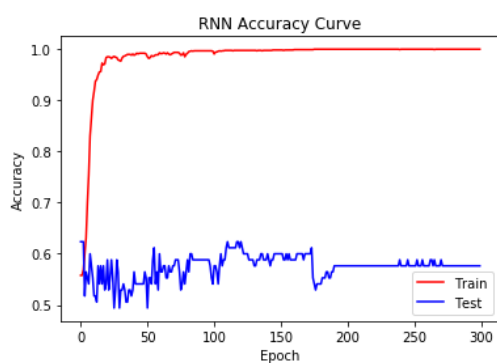
網路參數設定 RNN 及 LSTM

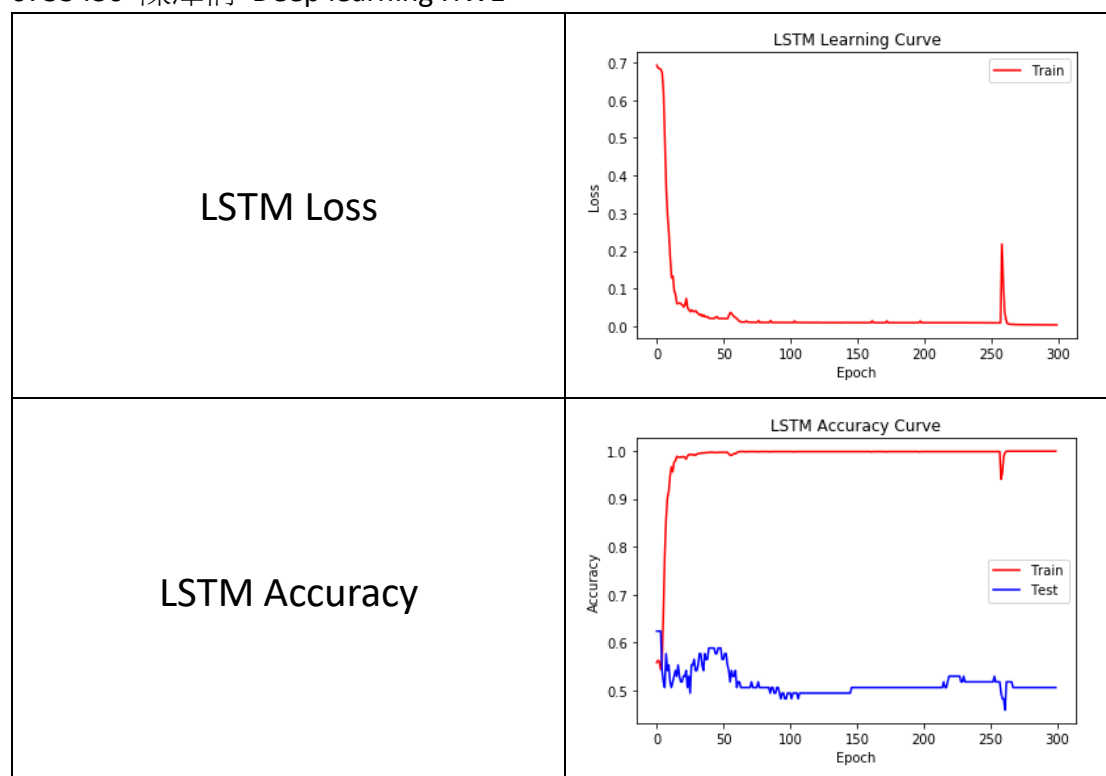
Loss	Cross entropy
Optimizer	Adam with Learning rate 0.001
Epochs	300
Batch size	32

RNN Loss



RNN Accuracy





基本上 RNN 和 LSTM 有著差不多的效果，甚至 RNN 最後訓練準度比 LSTM 還高，而 LSTM 則在訓練上比較穩定，Learning Curve 較少出現震盪。收斂方面，兩者都很快可以收斂到接近 100%準度，但有點輕微 Overfitting，建議可以加入 Dropout 或降低模型參數量