

Machine Learning Final Report

Small Data Training for Medical Images

隊名：B05901101QQAQQ

組員：B05901101 陳泓廷 B05901004 蔡維禎 B05901043 莊鎧爾

I. Introduction & Motivation

醫學疾病的照片用在 machine learning 上，常見的問題是有 label 的資料遠遠小於沒有 label 的資料，也就是說，若是能善用沒有 label 的資料進行 unsupervised learning，可以有效的提高 model 準確率，在此 project 中，我們有 10002 張有 label 的胸腔 X 光影像，還有 68466 張沒有 label 的影像，希望能藉由這些資料 train 出一個可以判別胸腔 X 光影像的 model，此 model 可以判別受試者患有 14 種肺部疾病的機率。

II. Data Preprocessing/Feature Engineering

我們用 Pillow 的 Image Module 讀入影像，接下來我們一共嘗試了幾種 data preprocessing 的方法，如下：

Supervised Model :

針對 supervised model，我們使用 torchvision 的 transforms function 來進行影像轉換，主要有以下幾個步驟：

(1)轉換成 RGB：

因為 dataset 中的圖片有 3 個 channel 也有 1 個 channel 的圖片，我們將其全部轉換成 3 個 channel，效果比灰階要好。

(2)Random Resized Crop：

這次處理的圖片大概都是 2000*2000 左右的大小，及使用 data loader 每次都只讀進那個 batch 的影像，記憶體還是不夠用。因此我們使用 RandomResizedCrop 將其隨機取圖片中的一塊，再 resize 成我們要的大小。這麼做的好處是，因為每次取到的都會是圖片不太一樣的部分，有類似 data augmentaion 的效果。

(3)Data Augmentation：

讓 training data 的多元性更高，也如同有更多的訓練資料，補足原本資料量少的不足。另外，testing 時為了穩定性，Training 和 Testing 使用不一樣的 transform function。Training 時使用了旋轉、鏡射、以及調整不同的亮度及對比，testing 時則是取先 resize 到一定大小之後取中間一塊作為 reference 的依據。

(4) Normalization :

為了讓 **activation function** 可以更有效的在範圍內運作，正規化是必要的步驟。我們原本試驗了三種方法：

- (a)將每個 pixel 除以 255，將 0-255 的值 cast 成 0-1 之間。
- (b)計算整個 dataset 的 mean 跟 variance，以此為基礎做 normalization。
- (c)使用 pytorch ImageNet training 所計算出的 mean 跟 variance，以此做 normalization。

最後我們選擇在 validation 以及 public dataset 效果都較好的方法(c)。

最後，針對 Supervised Model 所做的所有 feature transformation 如下所示：

Training data:



Testing data:



Unsupervised Model

我們希望可以利用大量 unlabeled data 訓練一個 feature extractor，可以很準確地將胸腔 X-ray 圖片中的特徵抽取出來，並將實際訓練時將 labeled data 通過此 extractor，以更好的得到圖片的 feature，並以此 feature 做 supervised training。我們使用了以下兩種方法做 feature extraction(詳細 model 描述在下一個部分)：

1. 對 unsupervised data 進行旋轉，轉0°, 90°, 180°, 270°，接著 train 用一個可以判斷胸腔照片的旋轉方向的 model，之後將 labeled data 通過 CNN 的 feature map 提取出來做 supervised training。
2. 我們 train 一個 unsupervised data 的 autoencoder，之後把前幾層提出來對 supervised data 進行 preprocessing。

III. Model Description

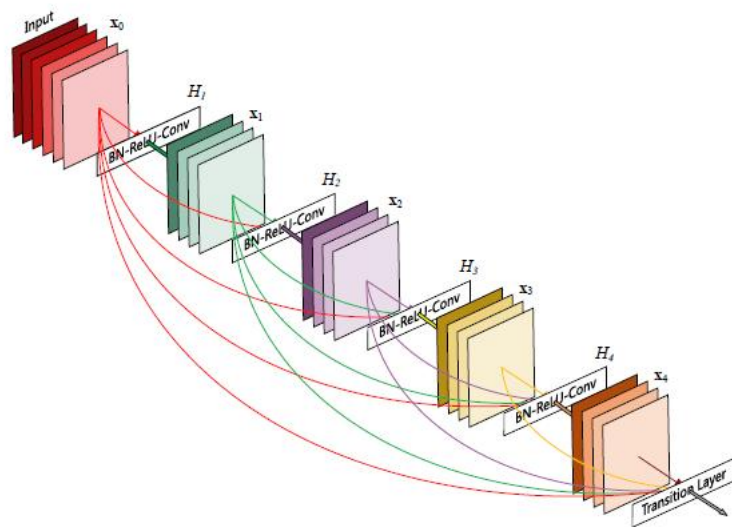
1. CNN model (採用 AlexNet 的結構)

結構：

Layer	Channel 數	內容
Convolution	96	11x11 conv, stride 4, padding 2
Pooling	96	2 x 2 max pool, stride 2
Convolution	256	5x5 conv, padding 1
Pooling	256	2 x 2 max pool, stride 2
Convolution	384	3x3 conv, padding 1
Convolution	384	3x3 conv, padding 1
Convolution	256	3x3 conv, padding 1
Pooling	256	2 x 2 max pool, stride 2
		256*6*6D fully-connected
		2048D fully-connected
		512D fully-connected

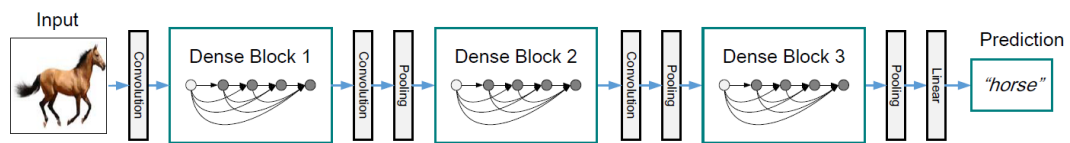
2. Densenet :

為了提升當層數加深時，前端 layer 學習效率下降的問題，**densenet** 將前端 layer 直接連接到各個距離遙遠的後端 layer，讓 **gradient** 可以更有效的傳遞，不會出現 **gradient vanishing** 的問題，如下圖所示：



為了讓相互連接的各層有同樣大小，但又有用到 CNN 中重要的 **pooling** 性質，讓圖片有濃縮的效果，**densenet** 把其整個網絡分成數個 **denseblock**，每一個

denseblock 內部形成如上述的相互連接，denseblock 彼此之間的 transition layer 則用普通 CNN 的單層 convolution + average pooling 串接，如下圖所示：



而在一 denseblock 內第 i 層 channel 數 $= \theta(k_0 + (i - 1)k)$ ，其中 k_0 為 input channel 數、 k 為參數 Growth rate、 θ 為 compression 參數。

我們利用 imageNet pretrain 過的 densenet121，做了兩個模型，其結構如下：

i. densenet121v1 (data preprocessing 用 244*224 的圖片)

Layer	Output 大小	內容
Convolution	112x112	7 x 7 conv, stride 2
Pooling	56x56	3 x 3 max pool, stride 2
Denseblock1	56x56	$\left(\begin{matrix} 1x1 \text{ conv} \\ 3x3 \text{ conv, padding 1} \end{matrix} \right)_{x6}$
Tansition1	56x56	1 x 1 conv
	28x28	2 x 2 average pool, stride 2
Denseblock2	28x28	$\left(\begin{matrix} 1x1 \text{ conv} \\ 3x3 \text{ conv, padding 1} \end{matrix} \right)_{x12}$
Tansition2	28x28	1 x 1 conv
	14x14	2 x 2 average pool, stride 2
Denseblock3	14x14	$\left(\begin{matrix} 1x1 \text{ conv} \\ 3x3 \text{ conv, padding 1} \end{matrix} \right)_{x24}$
Tansition3	14x14	1 x 1 conv
	7x7	2 x 2 average pool, stride 2
Denseblock4	7x7	$\left(\begin{matrix} 1x1 \text{ conv} \\ 3x3 \text{ conv, padding 1} \end{matrix} \right)_{x16}$
Classification	1000	1000D fully-connected, softmax
	14	1000D fully-connected, softmax

其中， $k=32$ 、 $k_0=64$ 、 $\theta = 1$ 。

ii. densenet121v2 (data preprocessing 用 600*600 的圖片)

結構與 i. 同，但 input 大小改為 600*600。

3. unsupervised learning – direction encoder

用上述的 AlexNet 以及 densenet121 model 來訓練一個 feature extractor，這兩個 model 架構就不再重述。這個 model 的目的是用來預測圖片的旋轉，作法就是針對每個圖片生成 4 個 one-hot encoding 的 label，在圖片輸入之後根據 label 做圖片的旋轉(前處理)，之後再將旋轉後的圖片丟進 model 裡訓練，以期能藉由預測圖片的旋轉來學習圖片的特徵，抽取到準確的 feature。

4. unsupervised learning – autoencoder

CNN for autoencoder

Layer type	Kernel size	stride	padding	Output channel	Activation	
Conv2D	11	4	2	64	Relu	Max Pooling
Conv2D	7	1	1	128	Relu	Max Pooling
Conv2D	7	1	1	256	Relu	-----
Conv2D	7	1	1	512	Relu	-----
Dense	-----	-----	-----	1024	Relu	-----
Dense	-----	-----	-----	2048	Relu	-----
Dense	-----	-----	-----	3*244*244	Relu	-----

用沒有 label 的 data，儘量把輸出和輸入的(224, 224, 3)的 image 一致。

之後把前四層的 Conv2D layer 取出，後面接 densenet121，用 label data train 14 種疾病的機率估計 model。

5. ensemble model

最後將上面幾種表現比較好的 model，直接用 weighted sum 的方式進行 ensemble 即為最後的成果。

IV. Experiment and Discussion

實作 project 的過程中，我們試驗了幾種不同的 model，並測試其表現。另外我們也針對表現最好的 densenet 121 做了許多參數的調整，以下列出我們所做的實驗結果以及討論。

一、首先針對不同 model 架構進行測試與分析：

1. CNN model：Kaggle public score 0.57 ~ 0.67

- CNN model 的優點是可以很快的收斂
- 缺點是無法得到很好的結果，成績無法過 simple baseline

2. densenet121v1：Kaggle public score 0.7 ~ 0.74

- 因為 CNN model 不好，所以改用 densenet121
- 成績有明顯提升，並且過 simple baseline

3. densenet121v2 : Kaggle public score 0.75 ~ 0.78

- i. 發現 training data 中很多 dirty data，所以在 data preprocessing 時加入更多的規則，對 data 進行進一步的轉換。因為圖片中肺部胸腔大小和位置不一，所以做 random resized crop。因為圖片光暗度不一，所以做 Color Jitter。圖片可能不從面拍的或是後面拍的，所以做 random horizontal flip。因為圖片不是拍正的，所以我們還做 random rotation。
- ii. 我們覺得 224*224 的圖片會讓很多疾病細節被忽略掉，所以在 preprocessing 時把圖片調成 600*600。
- iii. 在 testing data 的 preprocessing 中，因為不需要大多的 random，所以直接把圖片縮到 685*685，之後再 center crop 到 600*600。
- iv. 經過這些處理後，public score 可以得到顯著的提升。

4. unsupervised learning – direction encoder :

- i. 我們想要有效的利用 unsupervised data，所以 train 一個可以判別圖片方向的 model，希望可以藉此從圖片中抽 feature
- ii. 但是胸腔影片的方向特徵太過明顯，因為胸腔輪廓都相似，太容易判別，所以難以從中抽 feature。
- iii. 無法有效的抽出 feature 得到好的結果。

5. unsupervised learning – autoencoder : validation set score 0.70 ~ 0.72

- i. 用 unsupervised data，所以 train 一個可以判別圖片的 autoencoder，希望可以藉此從圖片中抽 feature
- ii. 圖片太小疾病特徵就消失了，而圖片太大 autoencoder 的 input 和 output 太多維度，所以 autoencoder 維度很難選擇。
- iii. Train 出的 autoencoder 只能學到胸腔的骨架輪廓，無法得到疾病細節，所以無法有效的抽出疾病的 feature。
- iv. 難以知道 autoencoder 是否真的抽出疾病的 feature，單單看 loss 無法判定 autoencoder 的好壞。
- v. 無法有效的抽出 feature 得到好的結果。

6. ensemble : Kaggle public score 0.78 ~ 0.79

- i. 把之前比較得的 densenet121v1 和 densenet121v2 最 ensemble
- ii. 一開始嘗試將結果做平均，可以過 strong baseline
- iii. 之後手動調整權重，把好的 model 的權重調大，之後最加權平均，可以得到目前最好的 model。

7.總結：

Supervised model 還是比 unsupervised model 的效果好，而使用較大的圖片會使得訓練的效果較好。以下列出各個 model 最好的結果：

	AlexNet	Densenet v1	Densenet v2	Direction encoder	Autoencoder	Ensemble
Kaggle public	0.56609	0.74823	0.76997	0.72077	0.70141	0.79251
Kaggle private	0.58467	0.74730	0.76973	0.72014	0.68361	0.78367

二、針對 densenet121 的架構，我們使用了不同的 input size，以及不同的 data augmentation 方法，並調整 model 的 hyperparameter。以下是我們的實驗結果：

1. Data Augmentation：

因為我們將圖片 size 調整成 600 * 600 之後，就基本上都有做 data augmentation，所以以下得比較表格是 224 * 224 的圖片上做出來的結果。可以明顯看出有 data augmentation 的效果較好，因此後來也都沿用，只是調整不同的 augmentation 方法。最後依照 validation set 上面的分數來決定要使用的 augmentation 參數。

	With data augmentation	Without data augmentation
Kaggle Public	0.74823	0.74240
Kaggle Private	0.74730	0.73862

2. Size of Input Image：

由下表可以看出，將圖片的大小調大之後，效果就提升滿多的，可以想見其實將圖片切成較小的一塊可能會導致圖片中的一些重要特徵流失掉，導致辨識準確率下降。

	224 * 224	600 * 600
Kaggle Public	0.74823	0.76997
Kaggle Private	0.74730	0.76973

3. Dropout of linear layer：

因為 overfitting 的問題，我們有常識在後面的 linear classifier 加上 dropout，而由結果可以看出，雖然在 dropout 0.4 的 model 在 public 上表現較差，但其實它在 validation set 以及 private set 上面的表現都較好。這邊我們就有點過度 fit public set，而忽略了 validation set 分數的重要性。

Dropout	0.2	0.4
Kaggle Public	0.77238	0.76997
Kaggle Private	0.76741	0.76973

4. Droprate of dense block :

同樣的，我們也嘗試在 densenet 中的每個 dense layer 加上 dropout，明顯可以看出有加 dropout 的效果是會進步的，如果以平均來看的話，是 dropout = 0.5 的表現最佳。

Droprate	0	0.2	0.4	0.5
Kaggle Public	0.76947	0.77266	0.77434	0.77238
Kaggle Private	0.76537	0.76414	0.76330	0.76741

V. Conclusion

1. 利用 densenet 取代傳統的 CNN 可以在預測結果得到顯著的提升。
2. 使用 pytorch 官方的 densenet121 pre-train model 可以讓 model 表現顯著提升。
3. 有效的處理 training data 可以大幅度提升 model 表現。
4. 好的 ensemble 可以結合不同 model 的優點，predict 的結果有顯著提升。
5. 一些 unsupervised 的方法目前沒有比 supervised 表現好，但若可以解決現有問題，有機會讓 model 表現再次提升。

VI. Reference

1. Densenet pretrained model :
<https://github.com/pytorch/vision/blob/master/torchvision/models/densenet.py>
2. Pytorch ImageNet Training
<https://github.com/pytorch/examples/blob/master/imagenet/main.py>
3. Densenet : <https://arxiv.org/abs/1608.06993>
4. ChexNet : <https://arxiv.org/abs/1711.05225>
5. Unsupervised Training with Rotation : <https://arxiv.org/abs/1803.07728>
6. Densenet : <https://arxiv.org/pdf/1608.06993.pdf>