Machine Learning Final Project

NTU_r07921078_4000 孟

張廷維 r07921078/李沂倫 r05621110

介紹與動機

競賽標題:

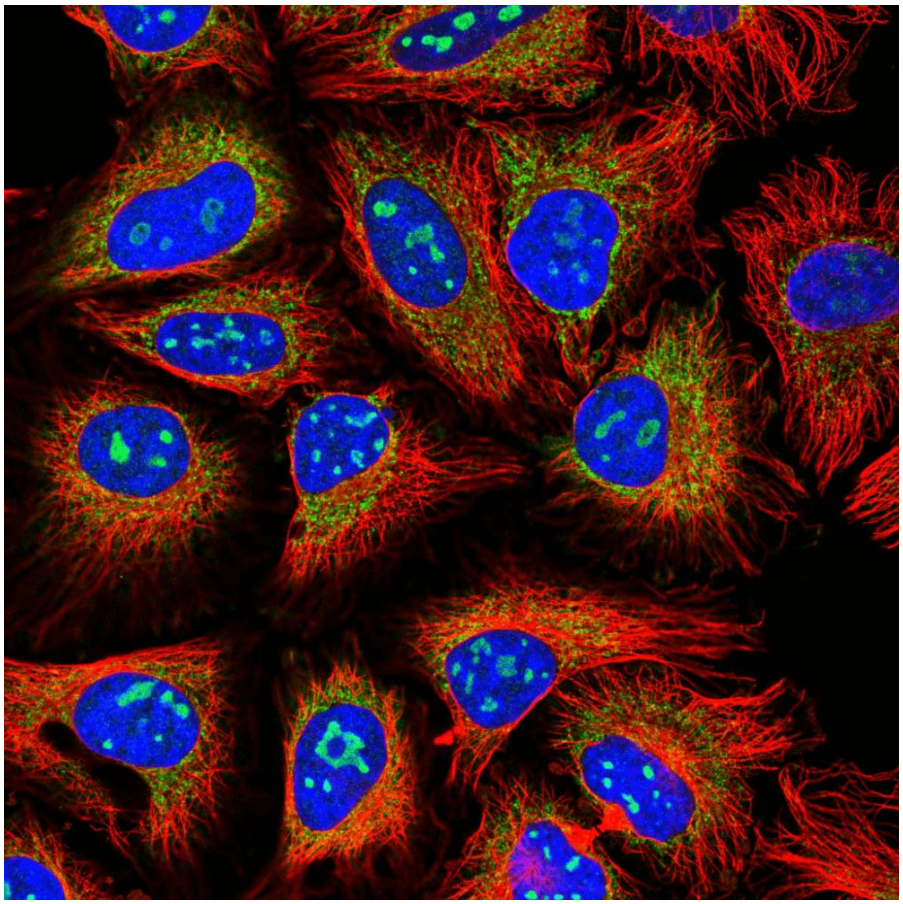
Human Protein Atlas Image Classification

Classify subcellular protein patterns in human cells

介紹:

本次競賽主要是針對細胞電顯圖片做分類，有 28 個類別，分別代表蛋白質(螢光綠) 在細胞的位置，資料主要由 Human Protein Atlas 提供，希望可以利用程序化的方式快速隊蛋白質在細胞的分布進行鑑定。

總共有 31072 個 Training dataset 與 11702 個 Testing dataset，每個樣本由四個代表不同顏色(RGBY)的 PNG 組成，大小為 512*512 個像素點，與先前不同的是，每個樣本可能屬於多個類別。



動機:

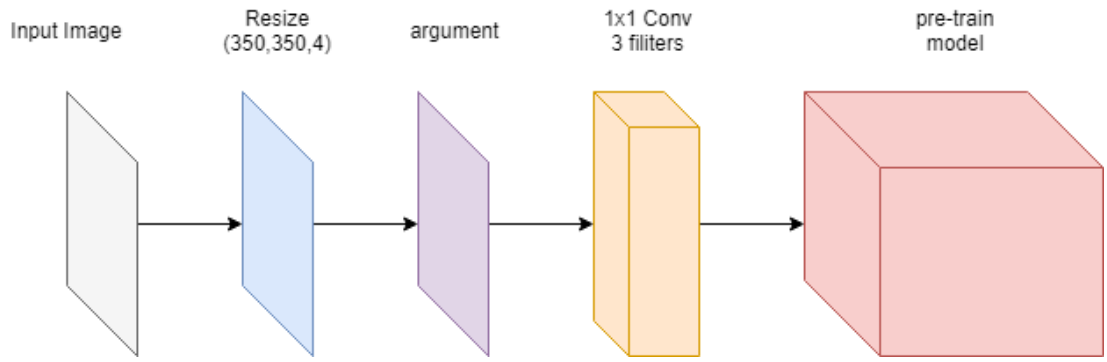
在本次組隊當中，隊員組成分別為電資背景以及生物統計背景，希望透過這次競賽可以互相了解學習對方的領域知識；另一方面由於兩人皆為初學者，希望可以透過 kaggle 上的 discussion/kernel 學習前輩高手們的作法，因此選擇此題目作為 final project。

資料前處理與特徵工程

在開始 train model 之前就遇到第一個問題，那就是在 final project 的資料量比以往的 homework 都還要大上許多，無法像先前的作業一樣一次把所以資料都讀進來，會面臨 out of memory 的問題。因此，會需要 data generator 來處理 training process 批次讀取 data，首先讀取 train.csv 將 filename 以及 file path 記錄下來用於提供 data generator 讀取資料使用，另外，透過 data argumentation 來增加資料量。

解決 out of memory 之後，在 feature engineering 時考量到自己 train 的 model 效果不及 pre-train model，因為決定使用常見的 ResNet / Inception 等等知名 model，然而在此次比賽提供的資料是 4 張 images 為一組 input data，而 pre-train model 多數 input channel 都是 3 維，如此面臨了一個問題就是無法直接將 input channel 4 維的 data 當作 input 直接接到 pre-train model，在這方面的 data preprocessing 我們決定在 pre-train model 前面新增一層 convolution layer 來將 4 維的 input 轉乘 3 維，然後再接到 pre-train model。

(以下為示意圖)



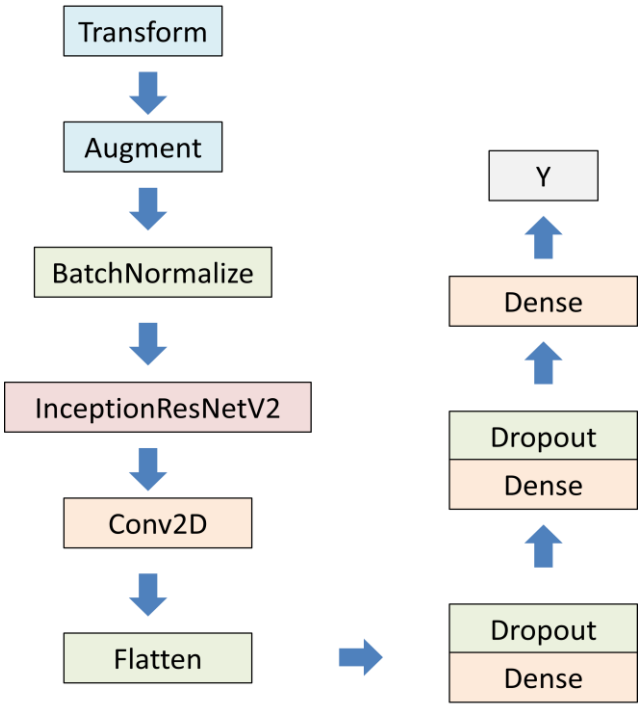
圖中 350 為可調整變數，而 pre-train model 可以是 ResNet or Inception

Augmentation: rotation(0, 90, 180, 270), flip(上下, 左右)

模組描述

Model1

InceptionResNetV2 + Convolution layer + Fully Connected layer



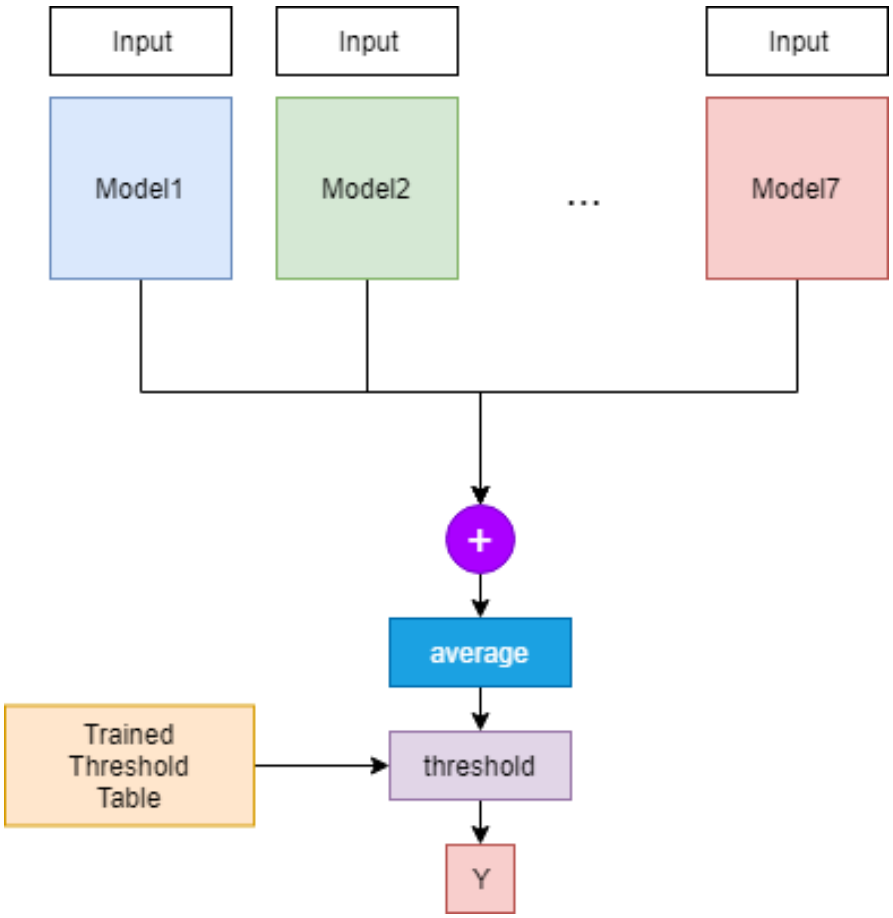
Model2

ResNet50 + Convolution layer + Fully Connected layer

Model3

在 model3 之前，像 model2 這種小更動的 model 多達 7 個，經過幾次調整 model 始終無法提升 kaggle 分數後，決定將這些多個 models 透過 ensemble 結合起來。過程中曾嘗試使用 fast.ai 使用 ResNet34 作為 pre-train model，但後來由於對 fast.ai 不太熟悉，因此最後還是決定改回使用 keras。

(以下為示意圖)



在 model3 中，每一個 model 最後一層都是通過 sigmoid，因此 28 維的 output 剛好就對應到每一個 label 的機率值，ensemble 的作法採用做基本的方法，將所有 score 加總起來取平均當作是整個 model 最後的 output score，在通過 threshold 後決定真正的 Y。

實驗與討論

1. 首先是第一個 model，使用 pre-train model IncptionResNetV2，抽出 feature 後通過一層 convolution layer 跟三層 Fully connected layers，最後同樣是需要經過 threshold，一開始設定為 0.5，得出來的 kaggle 分數為 0.29，這個 model 也是一開始通過 simple baseline 的 model。

2. 由於各個經過最後一層後，每個 class 會有其分數，然而不同 class 分數的分布不同，固定門檻值可能效果會不太好(0.29)，因此利用最後每個 class 的 training 分數(還沒變成 0,1) 以及真實的 y 訓練出較佳的門檻值，可以明顯的提升 kaggle 分數 (0.32)。雖然 kaggle 的討論有提到，這樣可能會造成 overfitting，不過似乎對我們的模型利大於弊。

3. 第二個 model 使用 resnet50，同樣通過幾層 Fully connected layers，最後 kaggle 分數為 0.36。

4. 然後在 Ensemble 之前做過的好幾個 models，基本上大同小異，以下為整理的表格

Model Name	Pre-train	NN layer	Class Weight	Resize	Model weight
Model1	ResNet50	3	Matrix	350x350	0.36
Model2	ResNet50	3	Vector	299x299	0.37
Model3	ResNet50	2	Vector	299x299	0.38
Model4	Inception	2	Vector	299x299	0.35
Model5	Inception	3	None	299x299	0.39
Model6	Inception	2	None	299x299	0.35
Model7	Inception	3	Matrix	350x350	0.395

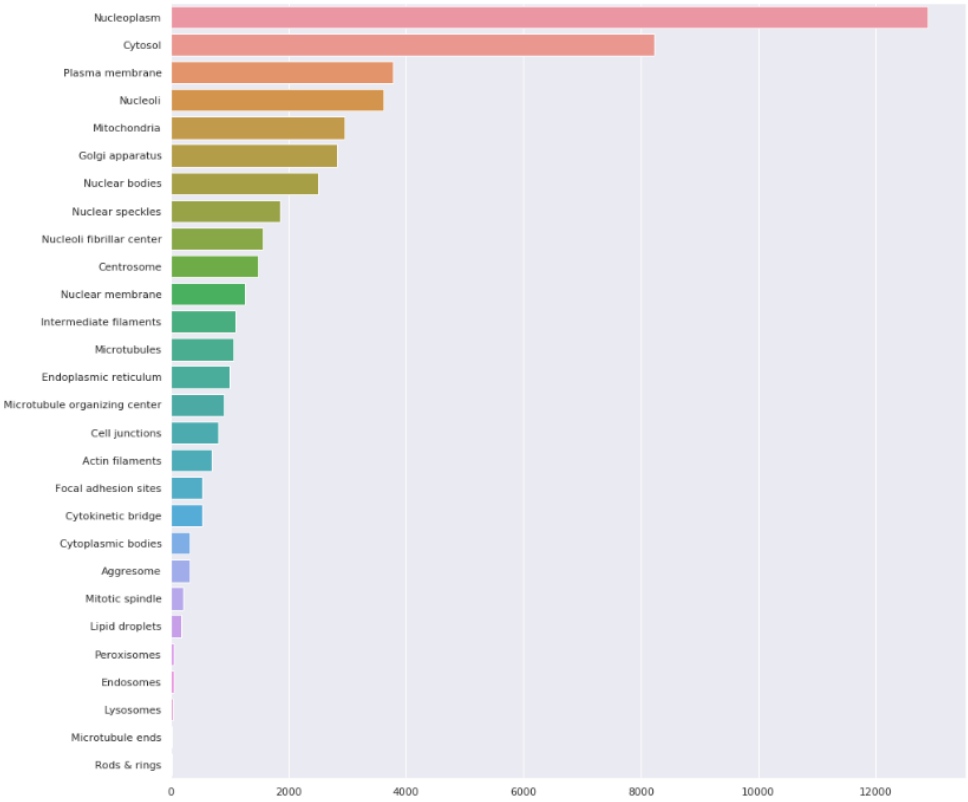
以上 7 個 models 為最後 ensemble 的 models，從以上幾個 models 可以發現

1. Model1 v.s. Model7: 可以發現差別只有 pre-train model，在我們的 case 中 Inception 的 performance 是比 ResNet50 來的好的。
2. Model2 v.s. Model3: 在這兩個 models 只有 NN layers 數量的差別，差了一層，就實驗結果可以推斷，通過 pre-train model 之後疊上去的作為 classifier 的 NN layer 應該不需要太多層，太多層可能會讓原本已經 pre-train 好的 model 又動到，這部分也是我們實驗時沒注意到的地方，沒有將 pre-train 的 fine turn 跟 classifier 分開 train，在聽完 Top3 的同學們分享才注意到這個細節。
3. Model5 v.s. Model7: 在這兩個 models 中，可以發現當 input size 大一點確

實可以有比較好的效果，我們認為是因為當 resize 越來越小，會有越來越多的 information 會被壓縮，導致在好的 neural network 都學不到應該學到的 feature。

4. 為了處理 imbalance 資料的問題，自訂義 loss function 使其能接受各個 class 的權重。這裡 class 權重的計算及賦予方式可以分成兩類:
    1. weight = -1*np.log2(class_num/n_samples) weight 爲一個長度為 28 的 Vector (表中 Vector 的意義)
    2. 利用 sklearn 的 compute_class_weight 計算每個 class 中 用 balance 法計算 1 跟 0 的 weight，weight 最後爲一個 28＊2 的 Matrix 我有對每一項取 log2 並將小於 1 的值變爲 1，weight[,0] 爲 y_true=0 時 為 false positive 處罰，weight[,1] 爲 y_true=1 時為 false negative 處罰。(對應表中 Matrix 的意義)

    雖然對單一 model 而言這個加入 class weight 對 f1 的提升幫助並不大，不過相信其在 ensemble 時有發揮不錯的作用。過程中雖然也有嘗試過 focal loss 但效果並不是很顯著。



5. 最後，將這 7 個 models 都 Ensemble 起來，依照各個 model 的權重進行加權平均後，針對不同的 class 進行門檻值的訓練，最後 kaggle 分數達到 0.439，完全出乎預料，沒想到一堆 0.3x 連 0.4 都不到的 models ensemble 竟然有 0.04 的漲幅。

結論

在本次 kaggle 比賽中才真正感受到碰到真正 data 時會遇到的種種問題絕不像是 homework 這麼單純，如同這次比賽遇到 data imbalance 以及 multi-label 的問題都是現實生活中真的會碰到的，當面臨不同問題時可能還會有不同的 issue 要處理。在此次 final project 學到了很多東西，尤其是發現原來 ensemble 這麼強大，明明就是一堆 0.3x 的 model 全部集合起來就衝到 0.43 了，直接驗證了老師上課所說的”三個臭皮匠勝過一個諸葛亮”。

參考

https://www.kaggle.com/byrachonok/pretrained-inceptionresnetv2-base-classifier
https://www.kaggle.com/rejpalcz/best-loss-function-for-f1-score-metric
https://www.kaggle.com/iafoss/pretrained-resnet34-with-rgby-0-460-public-lb
https://www.kaggle.com/allunia/protein-atlas-exploration-and-baseline