

HW4

Basic

1. Describe how you design your own model architecture

先利用 rescaling function 把原本為 uint8 的圖片轉換成 0-1 之間，在使用 3 層 conv2D 和 maxpooling2D，filter 分別為 16、32、64，kernel size 皆為 3，之後再進入 flatten layer，最後做兩個 dense layer 得到 output 的預測

2. Describe how you choose hyperparameters (eg. optimizer, learning rate).

這一個 model 中我使用 adam optimizer，會使得參數的更新比較平穩。使用 mean square error 作為 loss function，對使得對 outlier 有比較大的懲罰。

3. Describe difficulties you encountered.

在 basic part 基本上沒有遇到什麼問題，照著 tensorflow 的教學一步一步地做，就成功用這個 model 將 valid data train 到 0.7 了

4. Summarize your implementation.

在一開始的時候，先將 training data、validation data、test data 切好。再分別轉換成 np array，reshape 成(shape[0], 128, 128, 1)的大小，type 為 float32。將 X_train 和 df_train["Edema"]一起丟入 training_dataset 中，將 X_val 和 df_val["Edema"]一起丟入 validation_dataset 中。接著使用之前建好的 model 做 training。使用 trained model 對 X_test 做預測，並且將 output > 0.5 的設為 1，output < 0.5 的設為 0。最後將預測結果放入 CSV 中。

Advanced

1. Describe how you design or choose your own model architecture.

與 basic 相同，先利用 rescaling function 把原本為 uint8 的圖片轉換成 0-1 之間，在使用 3 層 conv2D 和 maxpooling2D，filter 分別為 16、32、64，kernel size 皆為 3，之後再進入 flatten layer，最後做兩個 dense layer 得到 output 的預測，與上方不同的地方是 dense layer 中，最後的 layer 為 Dense(7)，activation function 為 sigmoid。

2. Describe how you choose loss function and optimizer.

與上一題不同的是，在 loss function 是使用 binary_crossentropy，做 binary classification。

3. Describe difficulties you encountered.

在 advanced part 中，我遇到比較大的問題是在決定 loss function 的時候。我原本以為要使用 Categorical cross-entropy，導致每一次的 output 都非常的低且數字都一樣。後來我仔細看了使用 categorical cross-entropy 的時機，發現他是在一個問題有多個答案時才使用，而不是在多個 class 時使用。改完 binary cross-entropy 後，就很順利將 f1-score train 到 0.6

4. Summarize your implementation.

在 advanced part 中在資料處理的方面與 basic part 類似，先將 drop df 中的"age"、"gender"，再將 X_train 和 df_train 一起丟入 training_dataset 中，將 X_val 和 df_val 一起丟入 validation_dataset 中，接著使用之前建好的 model 做 training。使用 trained model 對 X_test 做預測，並且將 output > 0.5 的設為 1，output < 0.5 的設為 0。最後將預測結果放入 CSV 中。