

(a)

題目要求要選擇非線性之 model 去解二元分類問題，我選擇了 SVM 及 ANN，並比較 SVM 不同的 kernel 下之 performance。

SVM :

此處 SVM 的 model 使用 sklearn 中的套件 SVC 作為演算法，其中 kernel function 共有三種選擇 poly, sigmoid 及 rbf。我分別將其作實驗並加入 linear SVM 做比較。此處使用 Cross Validation 的方法做為評估一個 model 好壞的依據，fold 的數量為 5，亦即每次都拿數據總量之 20%作為 validation data。

ANN :

此處使用之人工神經網絡為較簡單的全連接層搭配 relu，以下為模型之 hyperparameters。

model:

Input(118) -> FC(64) -> ReLU -> FC(64) -> ReLU -> FC(1) -> Sigmoid

Loss_function : binary_crossentropy

Optimizer : Adam(lr=0.01)

EarlyStopping(monitor='val_loss', patience=3)

因為二元分類，故選擇使用 binary_crossentropy 而非 categorical_crossentropy，使其對於分錯的正樣本與分錯的負樣本都具有較優良的更新梯度。

以下即為(a)之實驗結果:

```
# class_number = 2
```

```
poly(3) = 0.7211440245148111
```

```
sigmoid = 0.7708631256384065
```

```
rbf = 0.8342951991828397
```

```
linear = 0.7800306435137896
```

```
ANN = 0.8321246027946472
```

```
# kernel 為 poly 時，degree 為 3 具有較好之 performance，故此處 degree 選擇 3
```

```
#使用之 metrics 為 validation data 之分類準確度
```

實驗結果發現使用 rbf 的 kernel 在此 case 上相較於其他的 kernel 具有較好之 performance。而 ANN 雖乍看之下具有也相當不錯的準確度，但在訓練時其準確度在每次訓練時均有約 5%左右的浮動，較為不穩定，即便將 learning rate 調

小以及更換 optimizer 也無法使其較為穩定。推測原因為在此 case 中之 loss function 形狀較為崎嶇，故其收斂之結果較容易掉進 local minimum。

故此處最後選用於 test data 上之 model 即為 SVM(kernel = 'rbf'):

Acc = 0.7948717948717948

(b)(c)

題目要求在 input 及 output data 中加入 Noise 並訓練模型後觀察結果，此處加入噪聲的方式 input 及 output 不一樣，原因為 output data 需為整數。

故詳細方法如下:

InputData_noise: 對 input 中的每一項加入噪聲，其噪聲大小取決於 SNR 之大小

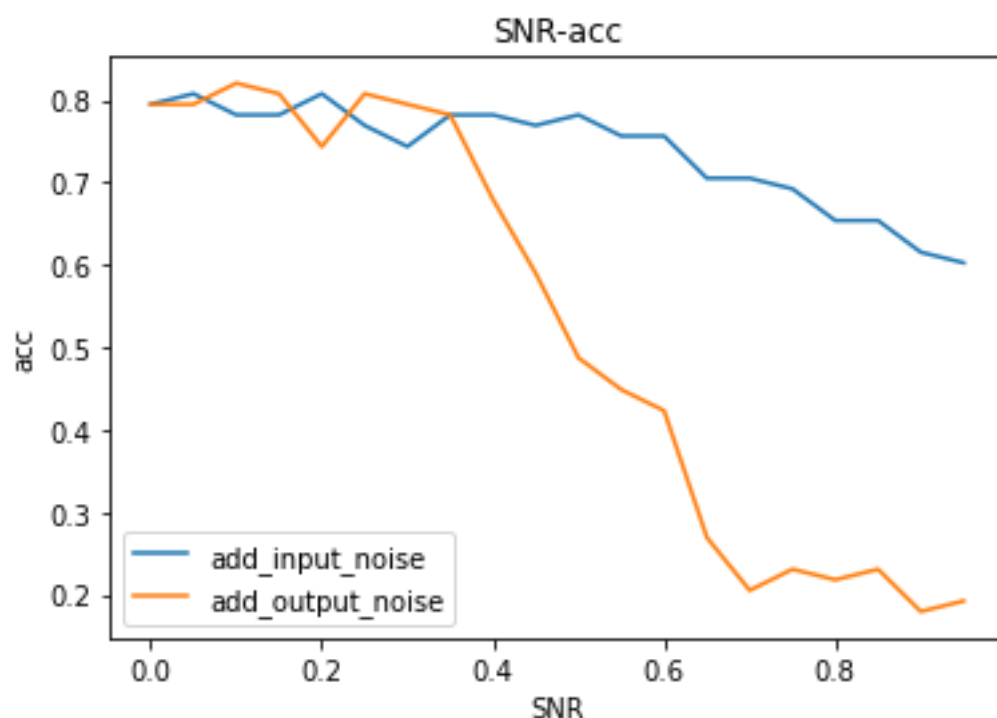
OutputData_noise: 其噪聲為錯誤的 label，即當 SNR=0.5 時，更改 50% 的 training data 之 label，將其改為錯誤之 label

在此定義下，比較 SNR 由小至大對於 validation acc 的影響，此處

Cross_validation 使用之 fold 數一樣為 5。實驗結果如下圖:

#add_input_noise: 單純加入 input_noise

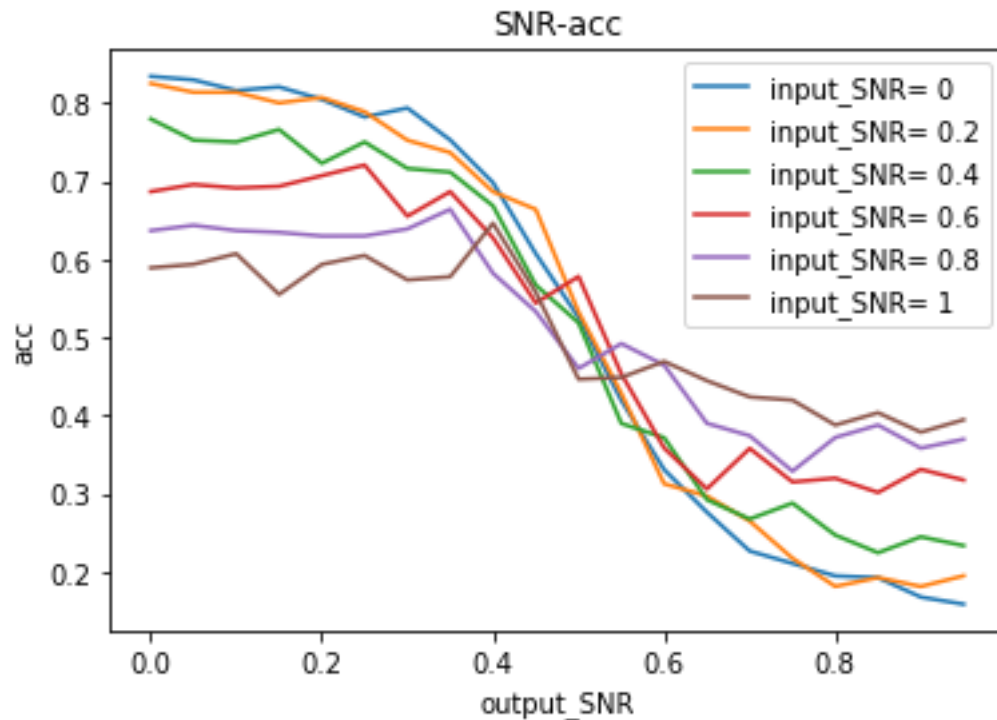
#add_output_noise: 單純加入 output_noise



因為 input 與 output 加入噪聲之方式不同，並不能直接的將兩種噪聲的 SNR 視為相同的變量。當錯誤的 label 占比小於 40% 時，模型之 acc 並沒有受到太大的影響，模型具有一定程度的抗噪能力。在當錯誤的 label 占比大於 40% 之後，模型的預測準確度急遽滑落。而兩者的趨勢也均說明了，輸入資料的品質很大程

度的決定了模型的準確度。

除此之外亦做了另外一個實驗，測試了在不同的 `input_SNR` 的情況下，改變 `output_SNR` 對於 `acc` 的趨勢。實驗結果如下：



基本上其結果大致上相同，即使是在不同的 `input` 噪聲下，模型對於 `ouput` 噪聲的抗噪能力並沒有太大的區別。

(d)

此處使用之模型與 a 小題幾乎相同，唯一不同處為 ANN 之 `output` 層變為 4，事先做 `one hot encoding`，並且將 `Activation` 變為 `categorical_crossentropy`，使其適用於神經網路架構。實驗結果如下：

```
# class_number = 4
poly(3) = 0.5827630234933605
sigmoid = 0.5803370786516855
rbf      = 0.666394279877426
linear   = 0.5646067415730337
ANN      = 0.6538461446762085
```

在 4class 的分類問題上，同樣是 `rbf` 略勝一籌，因此同樣選用 `SVM(kernel = 'rbf')` 作為最終選用的 `model`。

`Acc = 0.6410256410256411`