

tags: Machine Learning

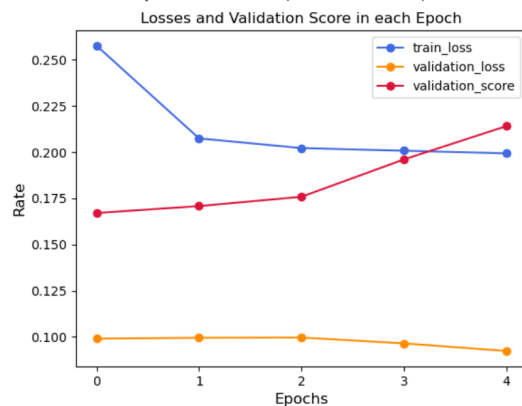
## Assignment 5

408410042 林靖紳

更改model的block數，比較調整前、調整後的模型準確率。

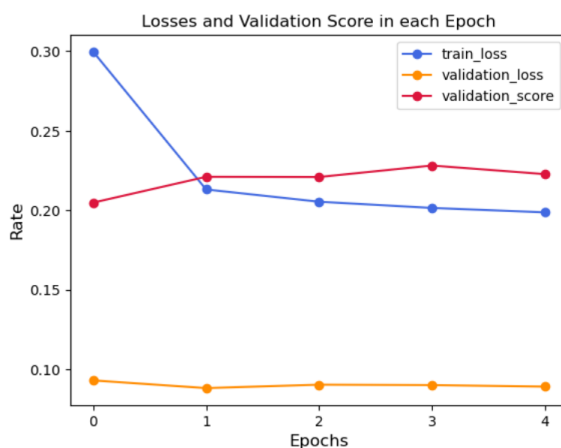
- 初始 nblock = 1
  - train loss = 0.199
  - validation loss = 0.092
  - validation score = 0.214

```
train loss: [0.2573769495925673, 0.20730993192251554, 0.20207530534428553, 0.20063702957126758, 0.19924253976030132]  
validation loss: [0.09908567768211293, 0.09957387965535933, 0.0997009845941005, 0.09653665967336726, 0.09242124371757436]  
validation score: [0.16698533194882432, 0.170730231002904, 0.17574211027713016, 0.1959090508291698, 0.2138982684137556]
```



- 將 nblock = 3
  - train\_loss = 0.198
  - validation loss = 0.089
  - validation score = 0.222
- 可以看到結果比 nblock = 1 時，還要好一點

```
train loss: [0.29968334012116515, 0.2131032024540912, 0.20540098274691584, 0.20149404882847616, 0.19873271968895456]  
validation loss: [0.09306811278997006, 0.08822911519541646, 0.09037853644102534, 0.09012218692092969, 0.0891750156115599]  
validation score: [0.20484254695545231, 0.22107027041061564, 0.22093784772775735, 0.22815288539416903, 0.222754934192678]
```

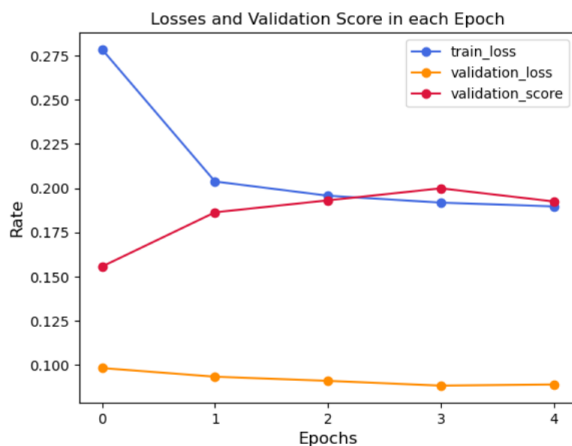


- nblocks = 5
  - train loss = 0.190
  - validation loss = 0.088
  - validation score = 0.192
- 可以看到效果比 nblock = 1 時不好
- 可以看到驗證集的 validation\_score 變小，但同時 validation\_loss 也變小，可能表示模型在訓練過程中出現了過度擬合（overfitting）的現象。

```

train loss: [0.27850895118522956, 0.20374711924721758, 0.19571421030979774, 0.1917895777484762, 0.18965186133825118]
validation loss: [0.09815059985005853, 0.0932508746940399, 0.09090923356468782, 0.08819597779869838, 0.0888576004344358]
validation score: [0.155626570151531, 0.18629935611038806, 0.1930775560888129, 0.19987251638248235, 0.1924547466314299]

```

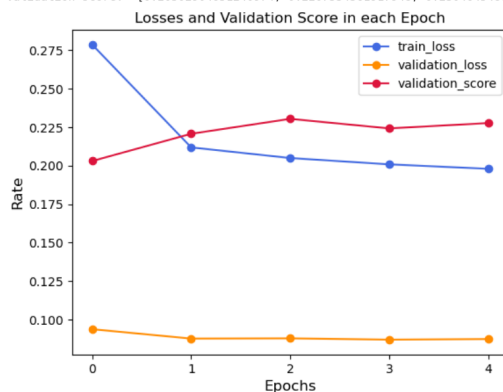


- nblock = 10
  - train loss = 0.198
  - validation loss = 0.087
  - validation score = 0.228
- 可以看到，相較於上面的測試，當 nblock = 10 時，有較好的成果

```

train loss: [0.27874122107465954, 0.2118900571199267, 0.20498433083558262, 0.2008742813281097, 0.19796549300421903]
validation loss: [0.09363268938901029, 0.08752646525814209, 0.08767027170921544, 0.08682978975026948, 0.08721593724526029]
validation score: [0.20302904651240974, 0.2207554361917948, 0.23046454866595387, 0.22425412852794904, 0.227712908809143]

```



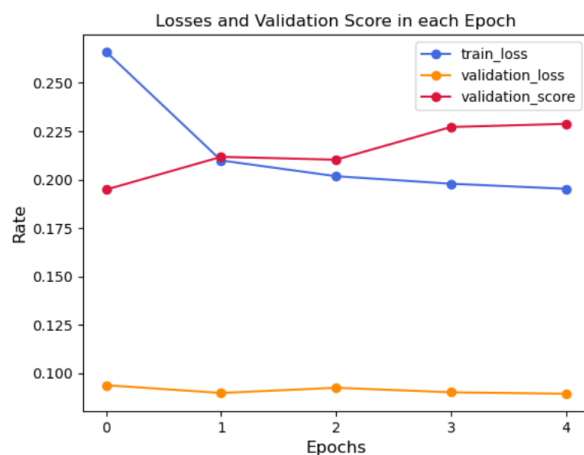
更改model內block的層數，試著增加或減少nn.Linear或nn.BatchNorm1d

因為上面的測試中，nblock = 10 為最好的成果，因此以下實驗的 nblock 都為 10

- linear 3 次
  - train loss = 0.195
  - validation loss = 0.089
  - validation score = 0.229
- 相較於原本 nblock =10 時的結果，發現 train loss 下降，但是validation loss 提高，常意味著模型出現了過度擬合（overfitting）的現象。

```
def _block(in_features, out_features, drop_rate):  
    return nn.Sequential(  
        nn.Linear(in_features, out_features),           #全連接層  
        nn.Linear(in_features, out_features),  
        nn.Linear(in_features, out_features),  
        nn.BatchNorm1d(out_features),  
        nn.ReLU(),  
        nn.Dropout(drop_rate)  
    )
```

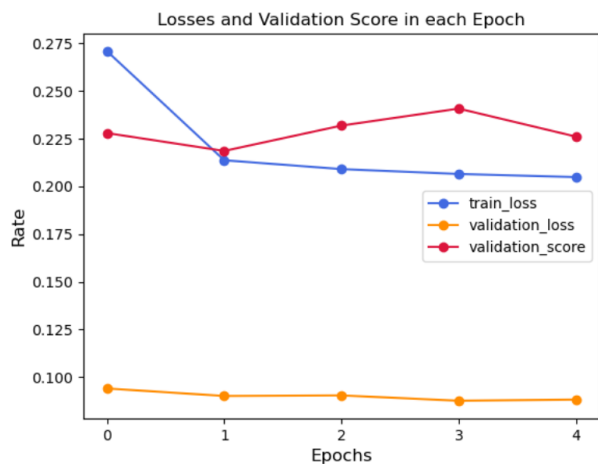
```
train loss: [0.2660200221968703, 0.20990609936530585, 0.20175111332853723, 0.19787075699669598, 0.  
19523771664121223]  
validation loss: [0.0938160691493573, 0.08980100047688043, 0.09249126036898134, 0.09015989602638  
497, 0.08940413899929396]  
validation score: [0.1948620522874908, 0.21174641811644565, 0.21028436825228333, 0.2271570814383  
1478, 0.22880090955488805]
```



- batchNorm1d 3 次
  - train loss = 0.205
  - validation loss = 0.088
  - validation score = 0.226
- 可以看出此結果，並沒有比原本來的好

```
def _block(in_features, out_features, drop_rate):
    return nn.Sequential(
        nn.Linear(in_features, out_features),           #全連接層
        #nn.Linear(in_features, out_features),
        #nn.Linear(in_features, out_features),
        nn.BatchNorm1d(out_features),
        nn.BatchNorm1d(out_features),
        nn.BatchNorm1d(out_features),
        nn.ReLU(),
        nn.Dropout(drop_rate)
    )
```

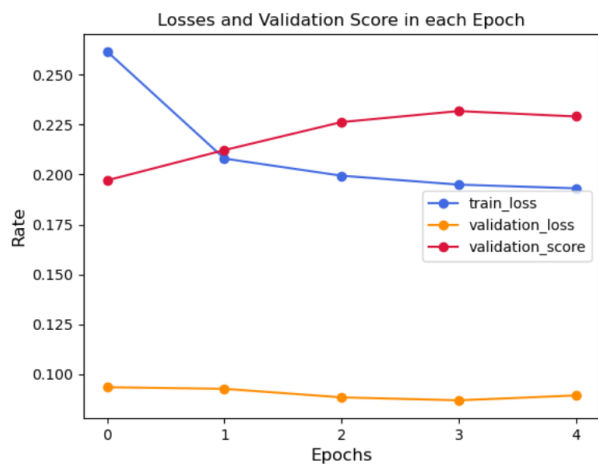
train loss: [0.27071836748231565, 0.21360902672245588, 0.2089809745385367, 0.20643683511801195, 0.20479439662209842]  
validation loss: [0.09411397429459385, 0.09018280831560295, 0.09048139796731573, 0.08768843880561308, 0.08830508841972184]  
validation score: [0.22784814447403, 0.21848489516601266, 0.23176499457134866, 0.24066753059175752, 0.22601086591481614]



- linear 2 次；batchNorm1d 2 次
  - train loss = 0.193
  - validation loss = 0.089
  - validation score = 0.229
- 相較於原本 nblock =10 時的結果，發現 train loss 下降，但是validation loss 提高，常意味著模型出現了過度擬合（overfitting）的現象。

```
def _block(in_features, out_features, drop_rate):
    return nn.Sequential(
        nn.Linear(in_features, out_features),           #全連接層
        nn.Linear(in_features, out_features),
        #nn.Linear(in_features, out_features),
        nn.BatchNorm1d(out_features),
        nn.BatchNorm1d(out_features),
        #nn.BatchNorm1d(out_features),
        nn.ReLU(),
        nn.Dropout(drop_rate)
    )
```

```
train loss: [0.26160826158833606, 0.20800846282903127, 0.1994204596707305, 0.19495178213769873, 0.19309158405941723]
validation loss: [0.09347409232653105, 0.09267830071708862, 0.08841005067864564, 0.0869197262858889, 0.08935493456114917]
validation score: [0.1971113347282023, 0.2122259472794784, 0.22627005086916094, 0.2317955714122072, 0.22908893567517666]
```



## Conclusion

對一個大型資料集的模型去做優化，找到適合的參數是一件非常困難的事情，要考慮的綜合因素非常多，每次要測試的時間也很久，不知不覺一天就過去了。

在本次的作業中，我學習到了一個專案的機器學習程式架構，還有對於資料前處理有了進一步的認識。

還有 notebook 形式的程式碼真的很好用 OWO

## Discussion

做完這次的作業，會比較想要去思考該怎麼有效率的去尋找參數，現在做起來比較偏向以 binary search 的方式去做，但是感覺還是有點沒效率，還有參數有可能會互相影響，該如何去衡量影響的權重對於結果造成的誤差。