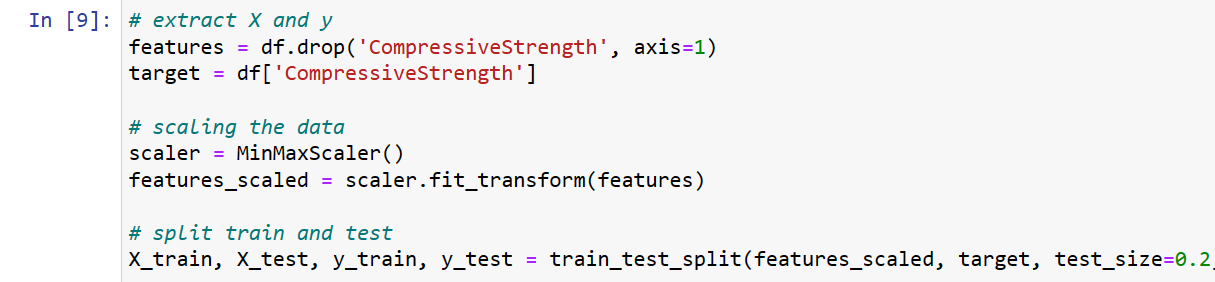
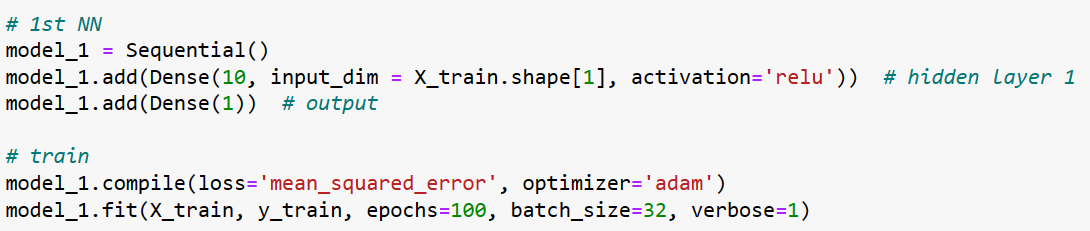
數據科學與決策科技 HW3 CNN

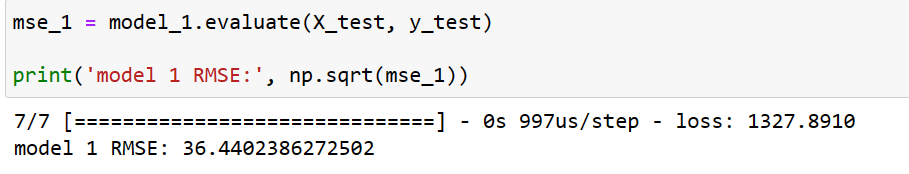
B08504052 化工四 賴正鎧

1. Concrete data – 1 hidden layer

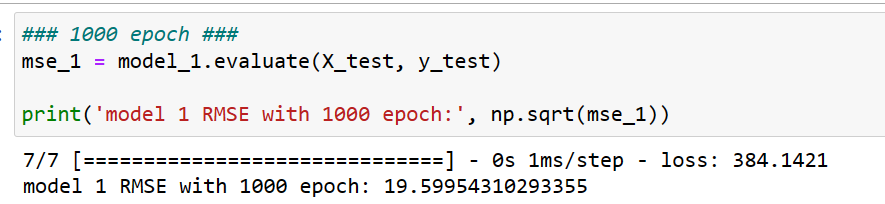
When dealing this dataset, I first scale the data since the scale of each columns are quite different. For example, the **CoarseAggregate** is up to 1000 while the **Superplasticizer** is only 2.5. Then, I split the data into train and test as usual.







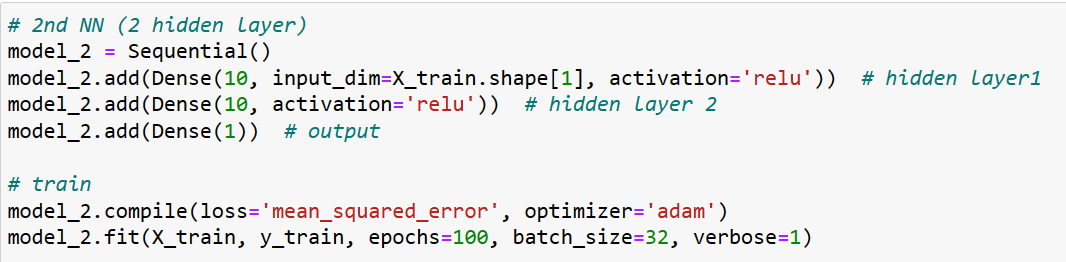
This is the result of 100 epoch and batch size 32. And I also try the version of 1000 epoch. Below is the result:

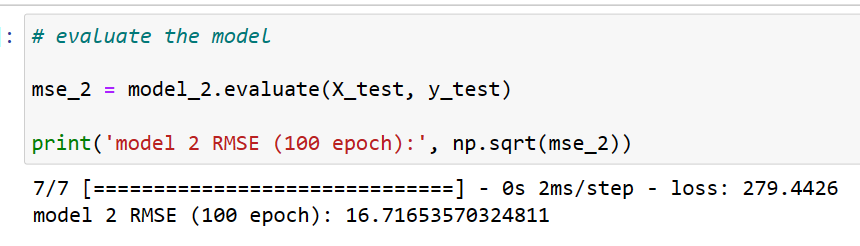
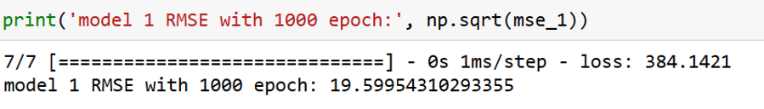


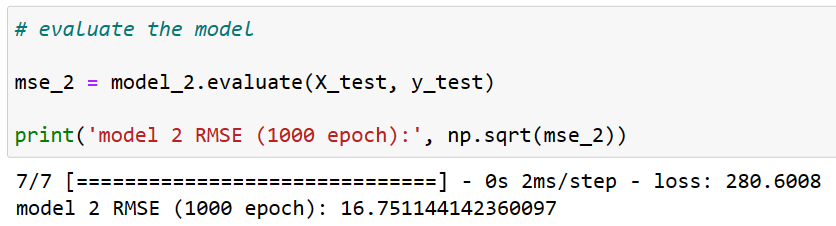
So I found that even 1000 epoch won’t cause overfit since the rmse of test set is decrease quite a lot!

1. Concrete data – 2 hidden layer

By construct one more hidden layer, the total rmse decrease even better than 1000 epoch’s 1 hidden layer neural network. Such a nice thing!



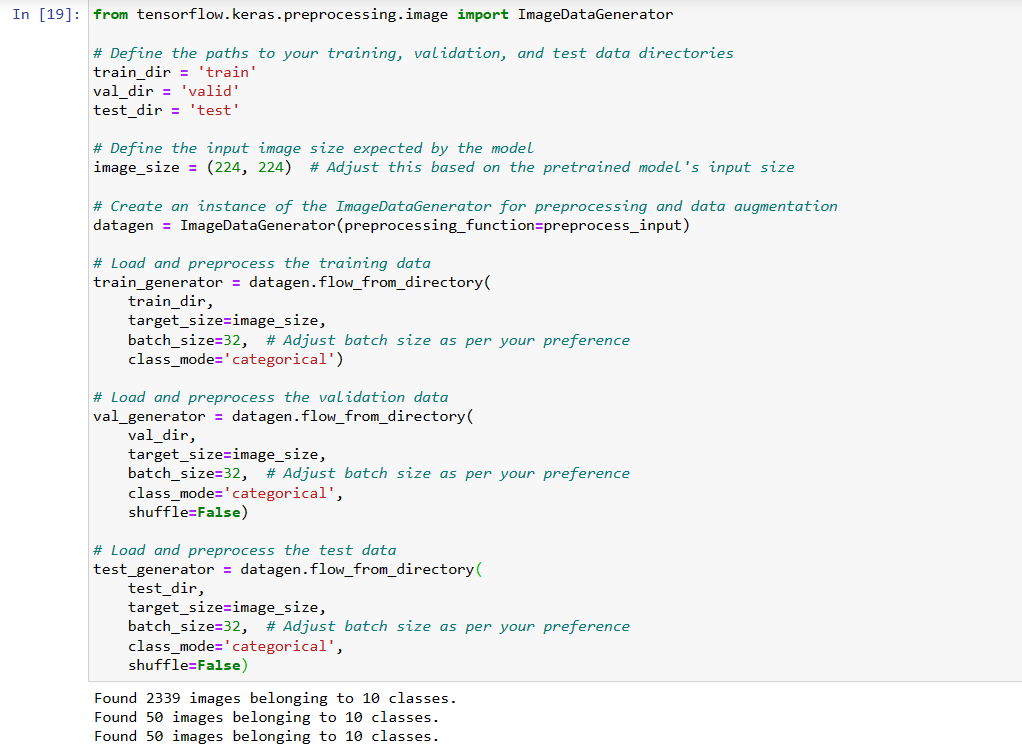


Compare to the previous rmse, the loss is better in a more efficient way! But I still want to know what will happen if trying 1000 epochs, and the result is below :

It turned out that the 100 epoch one performed better than the 1000 epochs one in 2 hidden layer situation! This showed that 1000 epochs might be overfitting senario.

1. Transfer learning

In the CNN part, I try to used the pretrained MobileNetV2 model. However, I have to read the figure into python so I can finally start traing. First, I do some preprocess about the figure,



After preprocessed, here is my network building script:



First I apply the MobileNetV2, then the data will go through 1 pooling layer and 3 fully connected layers, and finally go to the softmax function. I also freezed the basic layers to prevent it from changing accidently. Although I only try 5 epochs, the performance is quite well.

