

機器學習作業 3

B06502028

機械三

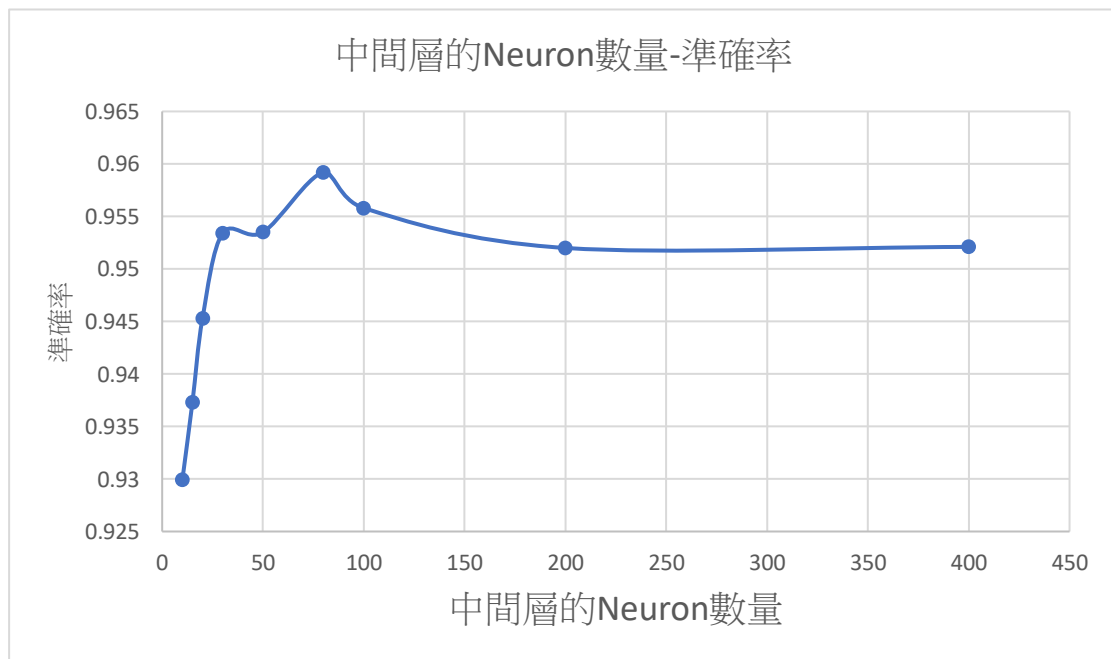
莊立楷

這次作業使用 fully connected 的 neural network，

於程式中可以選擇：

1. 最佳化方法使用 gradient descent 或 adagrad。
2. 可調整 learning rate 大小。
3. 可調整運行 epoch 數量，以及 mini_batch 的大小。
4. Input 的 data 可選擇是否做 normalization。若選擇使用，則會額外花非常多時間。
5. 可選擇是否做 training data 的 cross validation。
6. 可選擇中間 hidden layer 的 neuron 數量。

今天我使用不同的 neuron 數量，在 testing data 上實驗準確率，得到下圖：



可以發現神奇的現象：增加 neuron 數量會使準確率逐漸收斂至 0.95 左右，然而當 neuron 數量為 80 時，可以有最高的準確率。

另外，我嘗試使用 ReLU，可以得到較好的準確率，使用 sigmoid 的話準確率最多到 0.96 左右。

結論與討論

檢討過後，由於在 training set 上的準確率也不是太高，所以初步推論不是 overfit 問題，故若要再更 fit 一點，我覺得可以透過以下方法：

1. 增加 hidden layer。
2. 做 normalization，儘管很花時間。
3. Activation function 使用 ReLU。

當在 traing set 上做出較高準確率後，可再進一步：

1. 增加 Drop out。
2. 使用 cross validation。
3. 增加 regularization。

參考文獻

1. <https://medium.com/@intheblackworld/deep-learning-tutorial-%E5%BF%83%E5%BE%97-b1f7f84a497d>
2. https://github.com/imosafi/DNN_FC
3. 不用框架使用 numpy 从零搭建深度神经网络 (DNN)
<https://github.com/649453932/Build-a-deep-neural-network-DNN-using-Numpy>
4. Let's code a Neural Network in plain NumPy
<https://towardsdatascience.com/lets-code-a-neural-network-in-plain-numpy-ae7e74410795>