IV. Lab Four - 31298516

A. Wide MLPs on MNIST

I designed this model with a very flexible hidden layer, and subsequently I observed the visualisation results of the model validation accuracy while namely increasing the number of nodes in only one hidden layer and the number of layers while their number of nodes is fixed. The common parameters are: $optimiser = optim.SGD, loss_func = F.cross_entropy, epochs = 50, lr = 0.05, momentum = 0.9$. In the first work, the number of nodes in the hidden layer increases according to [10, 20, 40, 60, 100, 150, 200, 300, 400, 500, 600, 800, 1000, 1200, 1500] in each run; this is omitted due to the page limitation.

And in the second work, for saving the computational costs and also due to our model can perform well while there are only ten nodes in one hidden layer; I set 10 nodes in each layer and increase the number of layers from 0(no hidden layer) to 10(10 hidden layers); this is shown in the second graph above.

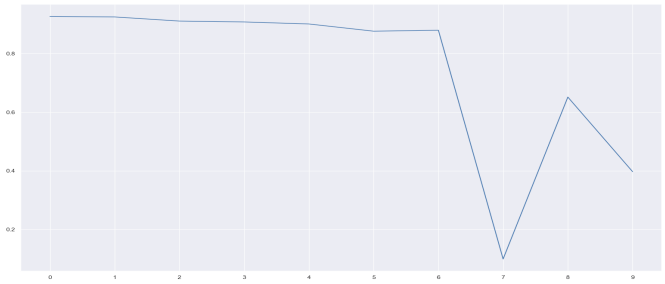

Figure 8: Curve while adding hidden layers while the number of nodes in each layer and all other common parameters are fixed.

The detailed results are [0.928, 0.9291, 0.9237, 0.9291, 0.8206, 0.9279, 0.9291, 0.9291, 0.9298, 0.929, 0.9276, 0.9289, 0.9291, 0.9286, 0.9291] of the first work, and there is a very noteworthy and counter-intuitive phenomenon that the model performance only apparently decreases while the value is 100. In other words, all the values perform well, our model just hates this number. In 'Figure 8', the detailed results are [0.9263, 0.9249, 0.9108, 0.9078, 0.9009, 0.8763, 0.8796, 0.0991, 0.6515, 0.3971], and we can tell that the overall tendency of accuracy is decreasing while increasing the number of layers.

For the second phenomenon, I think it is because the model is overly complicated for the MNIST dataset while the number of layers exceeds 7, then the model gets overfitting in training; there are not too many other latent possibilities, and I think this can be attributed to the low complexity of our task.

As to the first phenomenon, I also run it for another time with a larger range of the number of nodes in the hidden layer. And I found it decays in the different points with no clues could be seen in their corresponding loss curves (two graphs are shown below). Different from adding extra hidden layers, the average performance while changing this is very similar for most of the situations, it will rarely cause our model overfitting — no matter how many nodes you put in a single hidden layer, its structure is relatively simple. The hyper-parameter tuning has been being considerable trouble in this area, and though you can introduce heuristics into your model, it still cost a lot especially while the computational ability is limited.

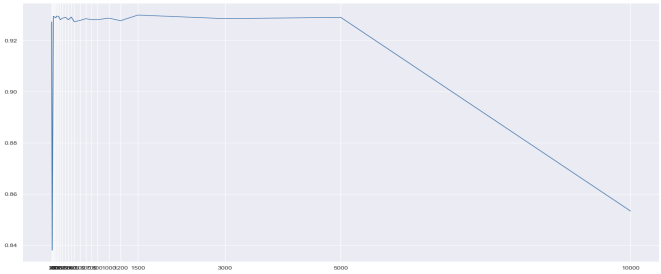

Figure 9: Curve while changing the number of nodes of the hidden layer while there is only one hidden layer and all other common parameters are fixed.

Figure 10: Loss functions correspond to all the test points above, as the labels at the upper right corner of this graph.

The test points in this time are [10, 20, 40, 60, 80, 100, 130, 160, 200, 250, 300, 350, 400, 500, 600, 700, 800, 1000, 1200, 1500, 3000, 5000, 10000]. And make a comparison with the previous work, we could only find that the model will finally have a worse convergence performance while the nodes of hidden layer are over 5k, as to the plunge while the value is 20, I think it is because of a bad initialisation by chance.

Even the simple structure 3*3 MLP can perform pretty good on the MNIST dataset. However, it doesn't mean it is vain to increase the model complexity for this task because it is unsure about whether or not they will be still good while the testing data get closer to the real-world scenarios. So the generalisation ability of a model is also an important factor which we haven't mentioned in this work. In conclusion, it is more useful of increasing the network structure than increasing the amount of the node in a layer if we aim to strengthen the model complexity; however, we need to achieve this under an appropriate range.