CS5242 Assignment 1

Name: Zhu Jin

NUS NETID: E0047338

Question 1

For the first network with two hidden layers, let h_1, h_2, h_3 be the output of each layer from the second layer to the last layer. We have:

$$\boldsymbol{h_1} = W_{0,1}\boldsymbol{x} + \boldsymbol{b_1}$$
$$\boldsymbol{h_2} = W_{1,2}\boldsymbol{h_1} + \boldsymbol{b_2} = W_{1,2}(W_1\boldsymbol{x} + \boldsymbol{b_1}) + \boldsymbol{b_2}$$
$$\boldsymbol{h_3} = W_{2,3}\boldsymbol{h_2} + \boldsymbol{b_3}$$
$$= W_{2,3}\left(W_{1,2}\boldsymbol{h_1} + \boldsymbol{b_2}\right) + \boldsymbol{b_3}$$
$$= W_{2,3}\left(W_{1,2}(W_{0,1}\boldsymbol{x} + \boldsymbol{b_1}) + \boldsymbol{b_2}\right) + \boldsymbol{b_3}$$
$$= W_{2,3}W_{1,2}W_{0,1}\boldsymbol{x} + W_{2,3}W_{1,2}\boldsymbol{b_1} + W_{2,3}\boldsymbol{b_2} + \boldsymbol{b_3}$$

For the second network with no hidden layers, let $\tilde{h}$ be the output. We have:

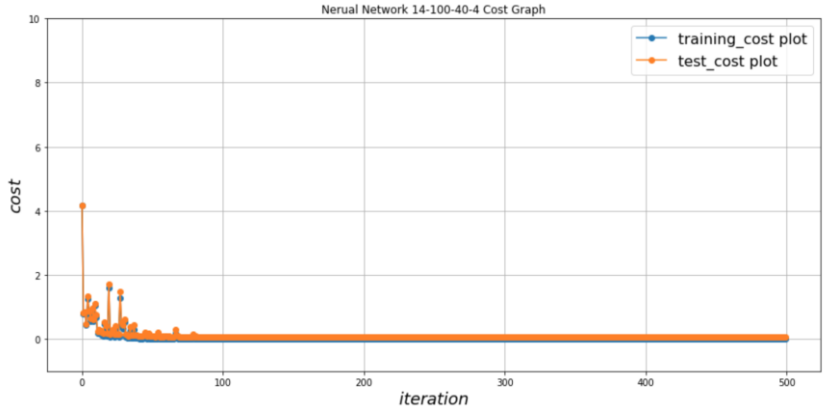
$$\tilde{\boldsymbol{h}} = \widetilde{W}\boldsymbol{x} + \tilde{\boldsymbol{b}}$$

In order to have $\boldsymbol{h_3} = \tilde{\boldsymbol{h}}$:

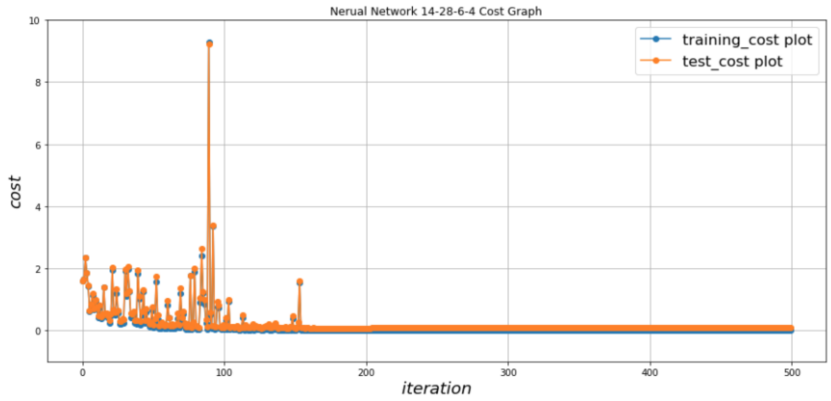
$$\widetilde{W} = W_{2,3}W_{1,2}W_{0,1}$$
$$\tilde{\boldsymbol{b}} = W_{2,3}W_{1,2}\boldsymbol{b_1} + W_{2,3}\boldsymbol{b_2} + \boldsymbol{b_3}$$

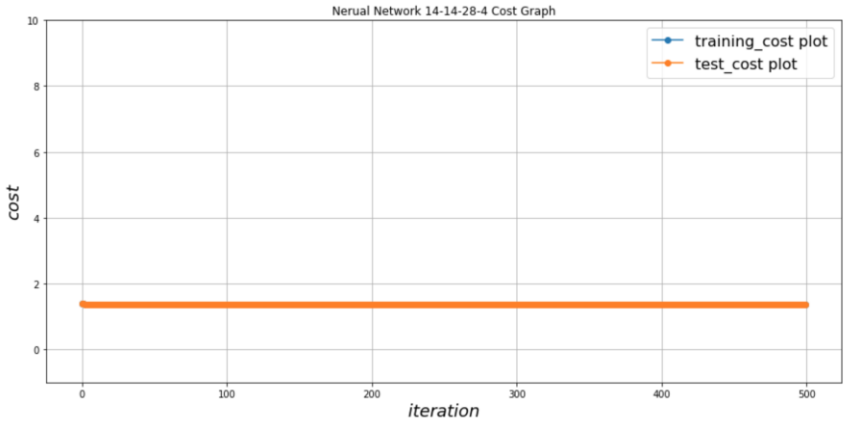
Question 2

1) & 2) Training & Testing costs for three networks

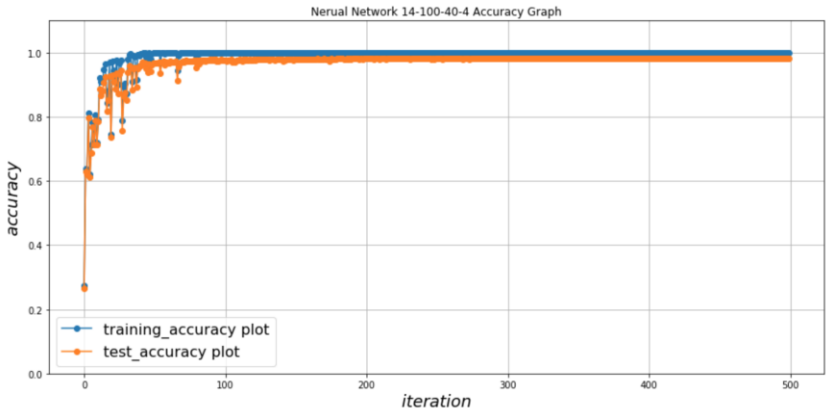


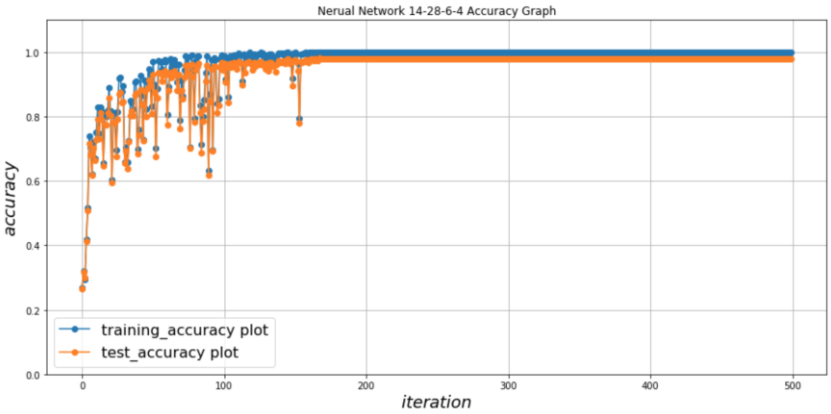
Tr

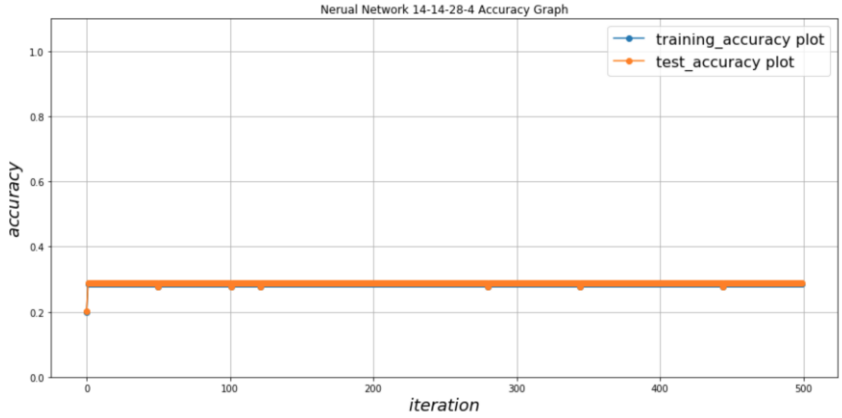




3) Training & Test accuracy for three networks







All three networks use uniform distribution between -0.5 to 0.5 to initialize weights and biases. They also use the same batch size 100 and same learning rate 0.1 for stochastic gradient descent. Based on the graphs, we see network 14-100-40-4 has the best performance as it converges much faster than the other two networks and achieves good accuracy. In the meanwhile, network 14-14*28-4 has the worst performance as its cost and accuracy stops improving immediately after first few iterations. This can be caused by the use of ReLu activation function and the choice of learning rate. Since the derivative of ReLu is zero when $x < 0$, it's possible that during gradient descent (especially with a large learning rate), the weights are updated in a way that the gradients of ReLu units become zero forever and the network stops learning. For network 14-28*6-4, there are some oscillations before it converges to optimum. Its learning rate could possibly be improved by using optimization methods such as Adam optimization which considers past gradients and reinforce gradients in the direction where they all agree.