# Assignment 1 - Neural Networks

CSE 474 Group 22

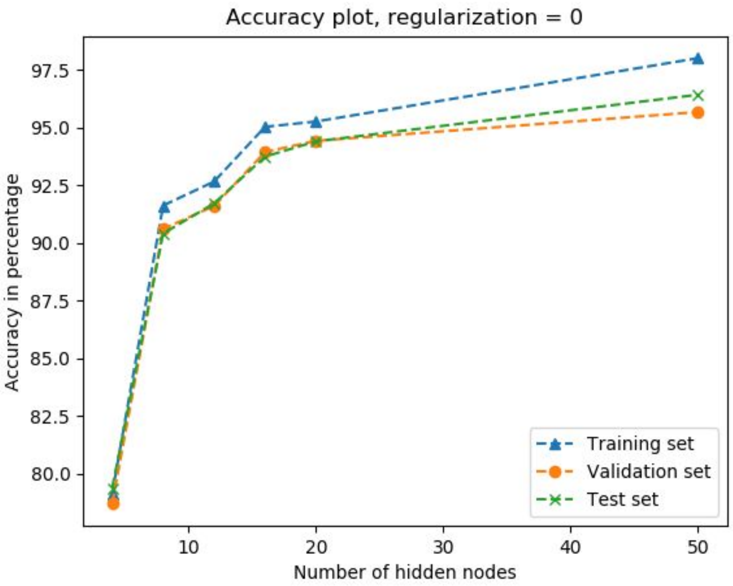
David Olsen, Der Shen Tan, Hoan Duc Tran

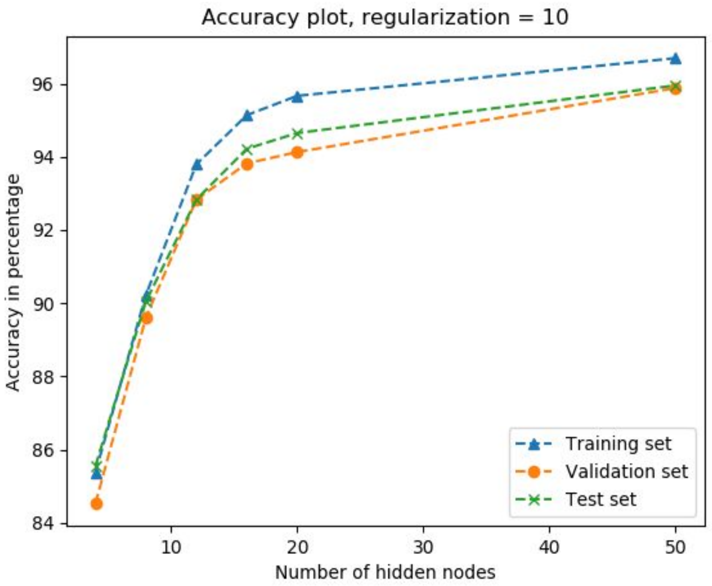
1– Explanation with supporting figures of how to choose the hyper-parameter for Neural Network: 30 points
2 – Accuracy of classification method on the handwritten digits test data: 10 points
3– Accuracy of classification method on the Ai Quick Draw data set: 10 points
4– Compare the accuracy and training time of deep neural network (using TensorFlow) with different number of layers : 10 points

## I. Number of hidden nodes with respect to accuracy and time.

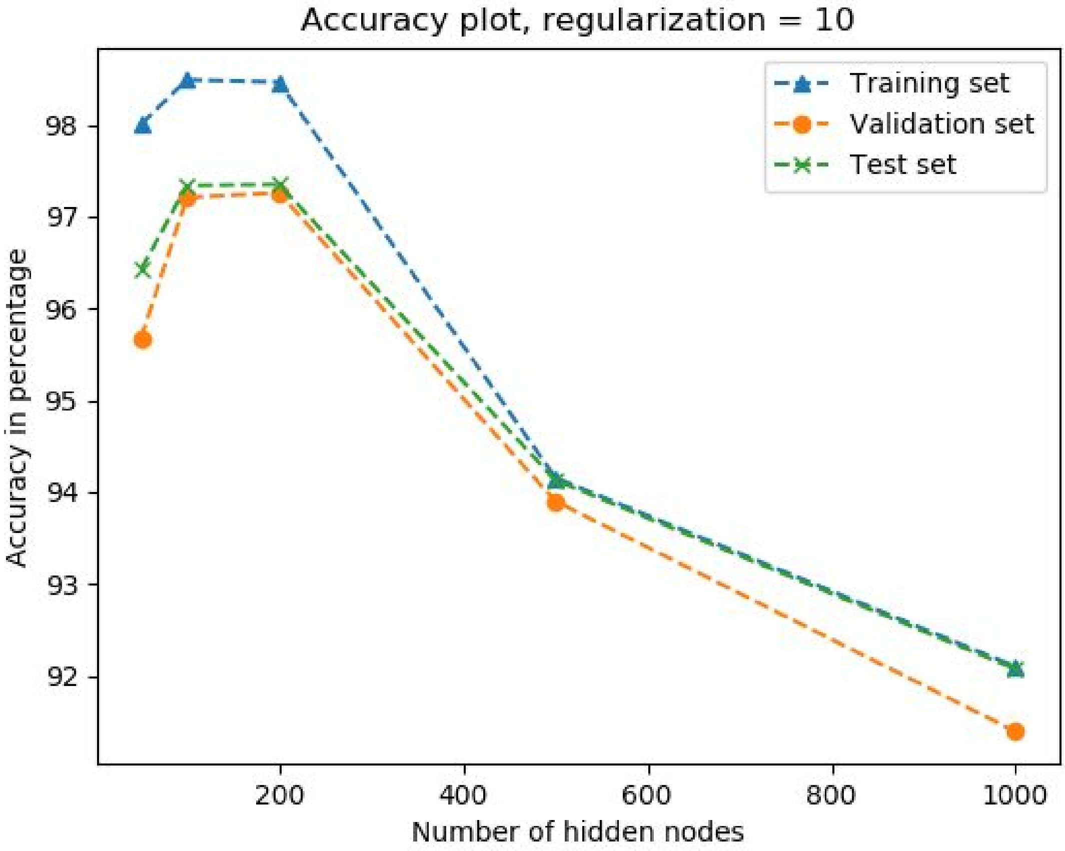
In the project, we set the number of nodes follow the recommended value, which is 4 to 20 with an increment of 4. We have also included 50 in our test, since this is the default value. Throughout the training part, we notice that when limiting the max number of iteration to 50, the objective loss function is still quite high, and the accuracy (although above 90%) could be improved. Therefore, throughout the experiment, unless stated otherwise, the max_iter value is set to be 100.

Below are the two figures demonstrating the accuracy when increasing the number of nodes from 4 to 50, with regularization coefficients at 0 and 10 respectively.



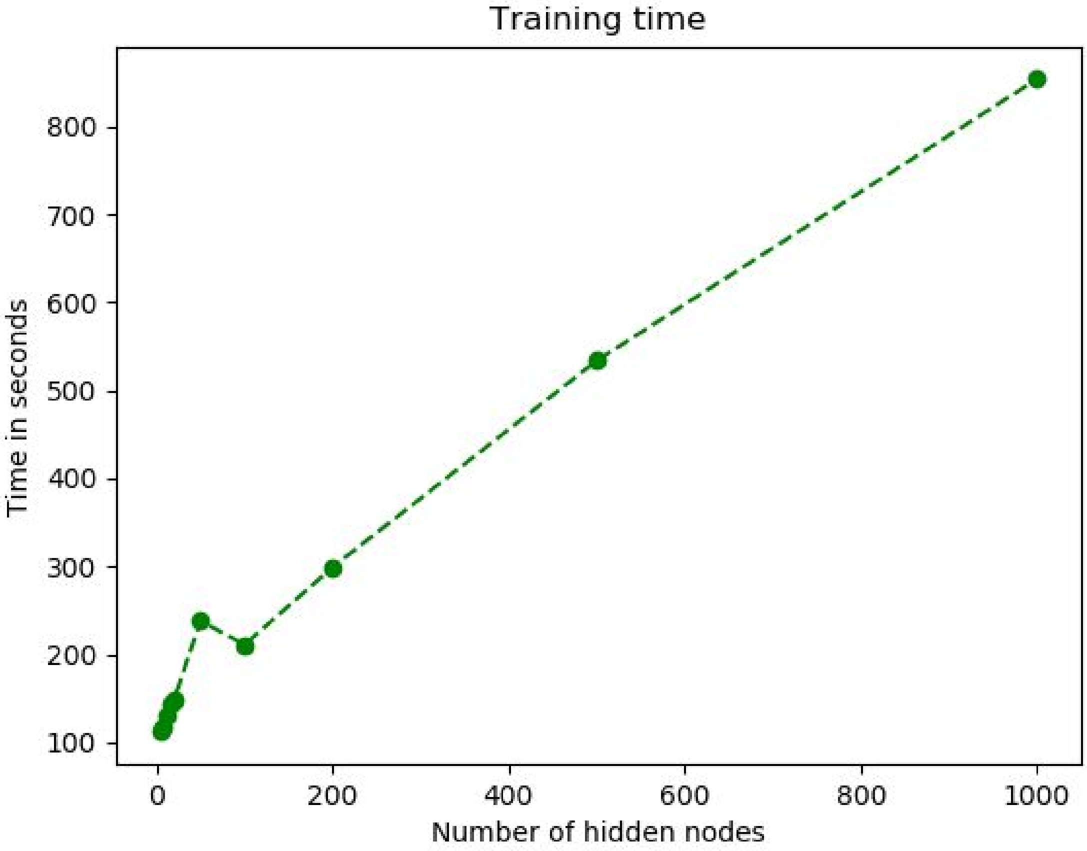


As the figures illustrated, the accuracy increased sharply as the number of nodes increase. This is to be expected, as more nodes mean that our model can perform more complex task. With this in mind, we decided to crank the setting up to 1000 hidden nodes.



Not what we expected at first. The accuracy rise up to 97 and 98 percent for testing and training respectively, then drop down to less than 92, 93. The reason for this is explainable: More nodes leads to more complex model, which needs more time and iteration to effectively train, which we have limited by the option max_iter = 100. More complex model may also leads to more generalization, or stucking at a non-optimal minima.

Regarding training time, it is expected that the time to train will be high.



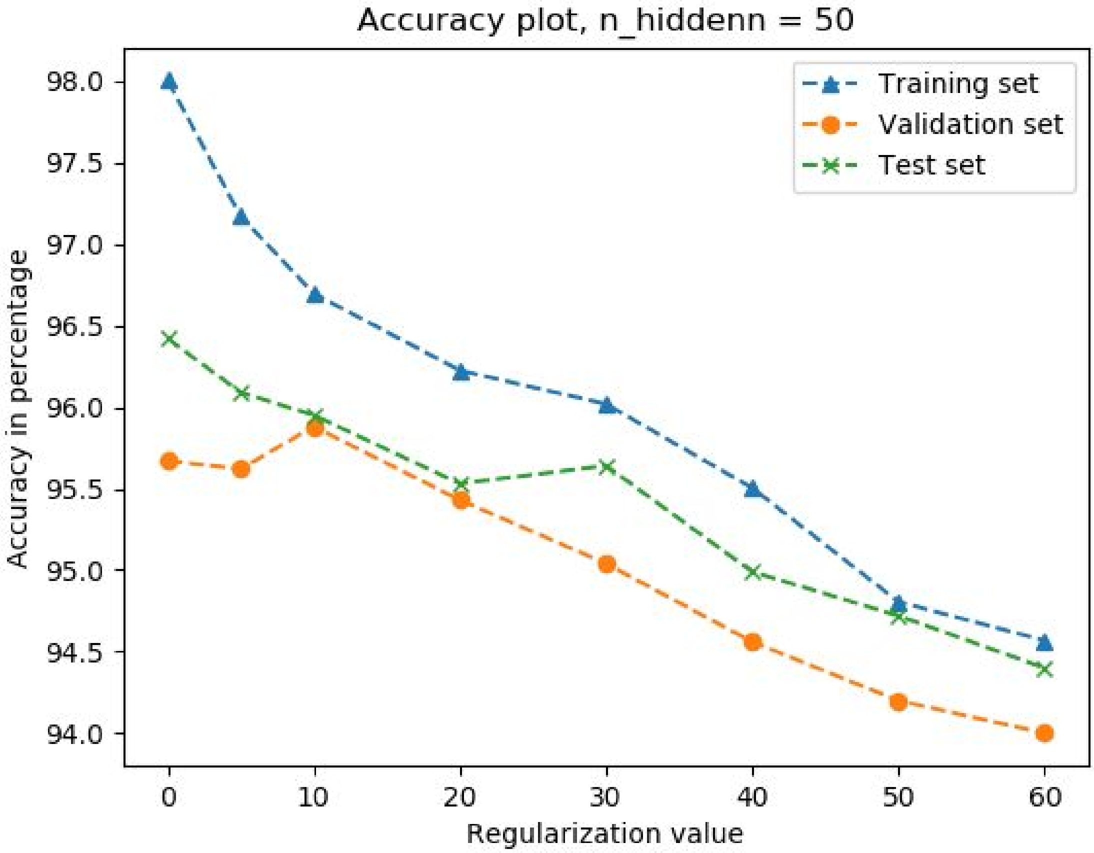
For low n_hidden nodes from 4-20, training time is less than 150 seconds. For 100 hidden nodes, the training time is a little bit higher than when hidden nodes is 150, (238 vs 211 seconds). This is due to various effect when running on personal computers (we did not use UB server, explanation below). For higher hidden nodes, the increase in time is linear, up to more than 800 seconds when there is 1000 nodes. The result were obtained with lambdaval = 0, but it is expected to have similar effect for higher lambdaval.

One weird thing to note is that the metallica server failed to converge for hidden nodes larger than 500. It would stop while the objective function is still has a high value, and the accuracy is less than 10%. However, the same code put on an average, personal computer will have decent accuracy. Therefore, all tests were conducted on personal computer.

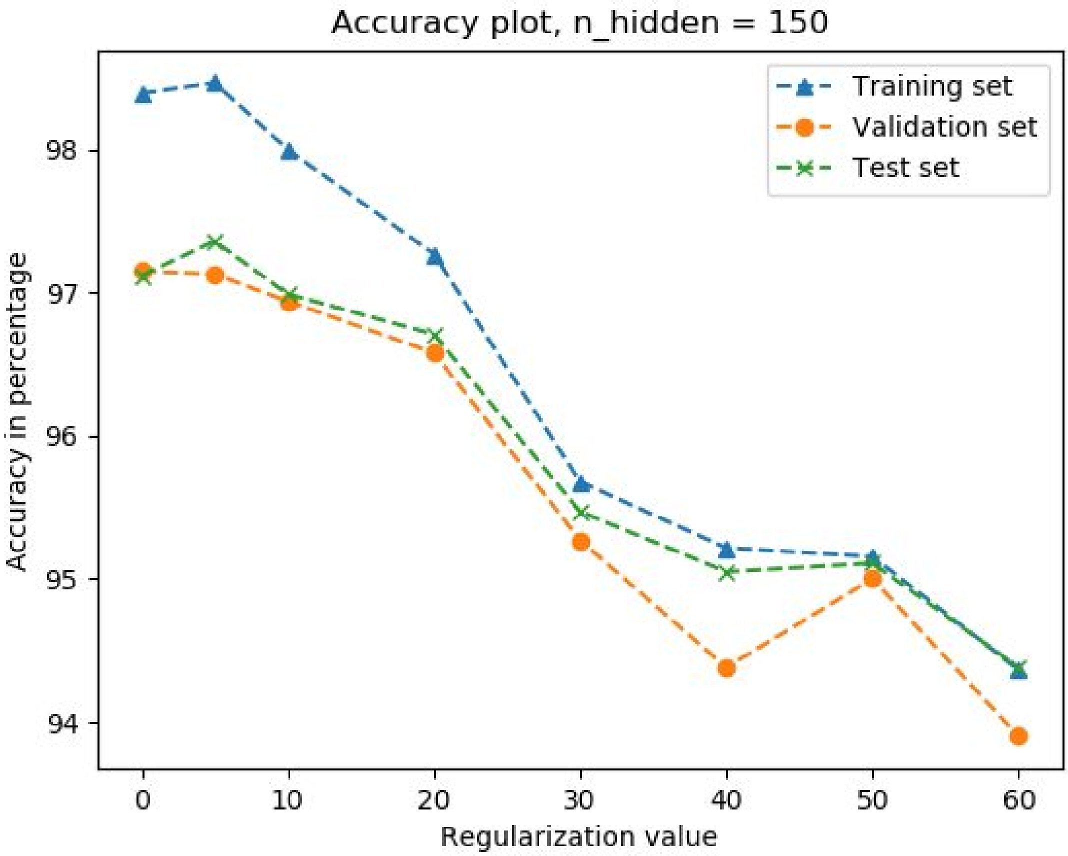
Overall speaking, **n_hidden = 150** seems to be the optimal value. It got high accuracy, decent training time (around 4 minute). To get the same accuracy with 50 nodes, one would have to increase the max iteration option, which would result in same training time eventually (we have not test this).

## II. Regularization with respect to accuracy

It is to be expected that with higher regularization value, the training accuracy (while keeping the hidden nodes) is lowered for reducing overfitting.



As expected, accuracy for training set reduced. But surprisingly, testing accuracy is also reduced alongside. The reason for this might be because the test set and training set is similar. But it could be because of this n_hidden value is not the best value. So we decided to go and test the value n_hidden = 150. The result is below.



You can see a similar trend. This suggest that increase in lambdaval does not increase the testing and validation accuracy **for this dataset**. You can still see that, with n_hidden = 150, lambdaval = 5 does in fact increase accuracy.

In conclusion, **regularization = 0** seems to be the most optimal value in this **dataset**. Setting **regularization = 5** does increase accuracy in certain cases. So 0, and 5 are the two, most optimal value in this case.

## III. Test with the Quick Draw data set

With n_hidden = 150, and regularization = 0 and regularization = 5 (as from previous section), we obtained the following result.

With regularization = 0, the training accuracy is 80.57%, validation accuracy is 78.66% and testing accuracy is 79.072%.

With regularization = 5, the training accuracy is 78.42%, validation accuracy is 77.3% and testing accuracy is 77.4488%.

## IV. Deep neural network using Tensorflow

We were not able to test with 7 layers. But for 3 layers, the training time is 160 minutes (almost 3 hours). And for 5 layers, it is 226 minutes (almost 4 hours). It is expected that 7 layers would take at least 6 hours or more. The accuracy also dropped from 3 layers to 5 layers. With 3 layers, the accuracy is 0.574. But with 5 layers, the result dropped to a varying 0.44.