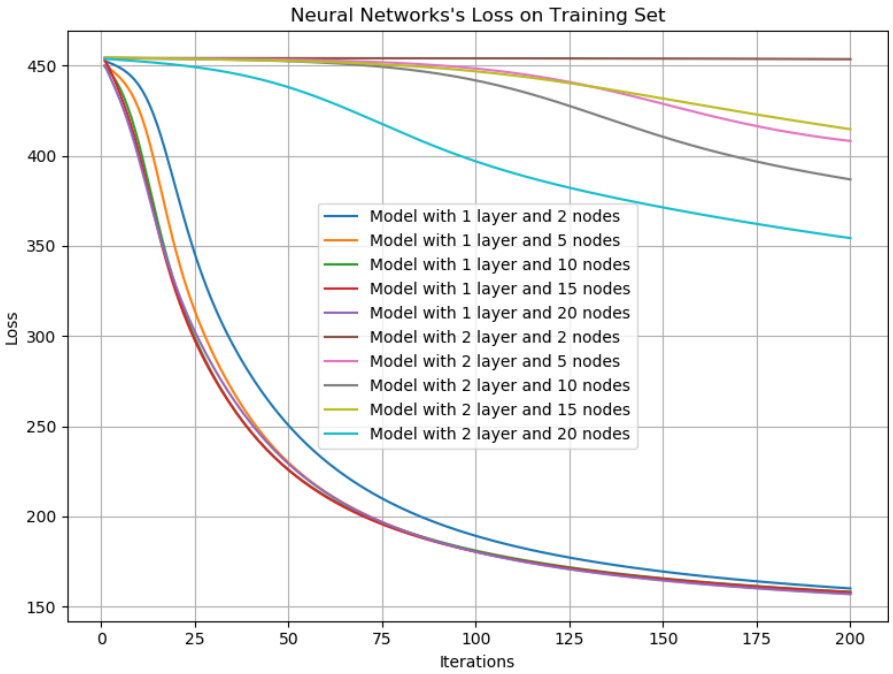
# Neural Network

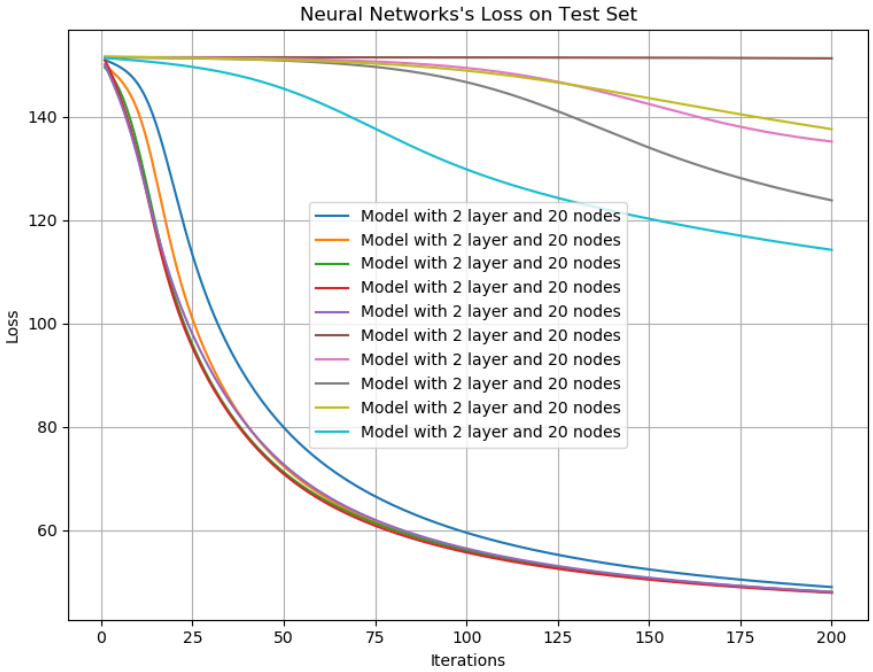
## 1 Training the Neural Networks

All the raw program output can be found in the *Resut-NeuralNetwords.txt* file.

### 1.1 Training Set Loss




### 1.2 Test Set Loss



### 1.3 Analyze

Because we use the following Loss function, the loss is related with the dataset size. This is the reason we can see the loss form **Training Set** is larger than **Test Set**.

$$E(\vec{w}) = \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2$$

From the 2 plot we can also find out that:

- More layer doesn't mean better result
- Because we initialize the *Neural Networks* with random weights, every time the result can be a little bit different.

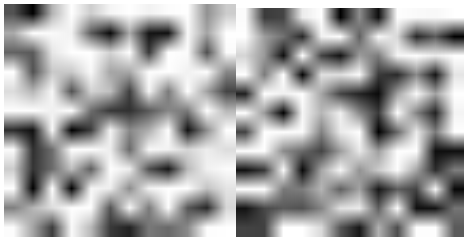
### 1.4 Best Model

From the program output we can find that the *Neural Network Model* can get at best of 90.5941% accuracy and error bound [88.9506% 92.2375%] on **Test Set**, when using:

- Hidden Layer 1
- Nodes for Each Layer 20

## 2 Hidden Layer Weights

The 2 Visualizations of the weights are:

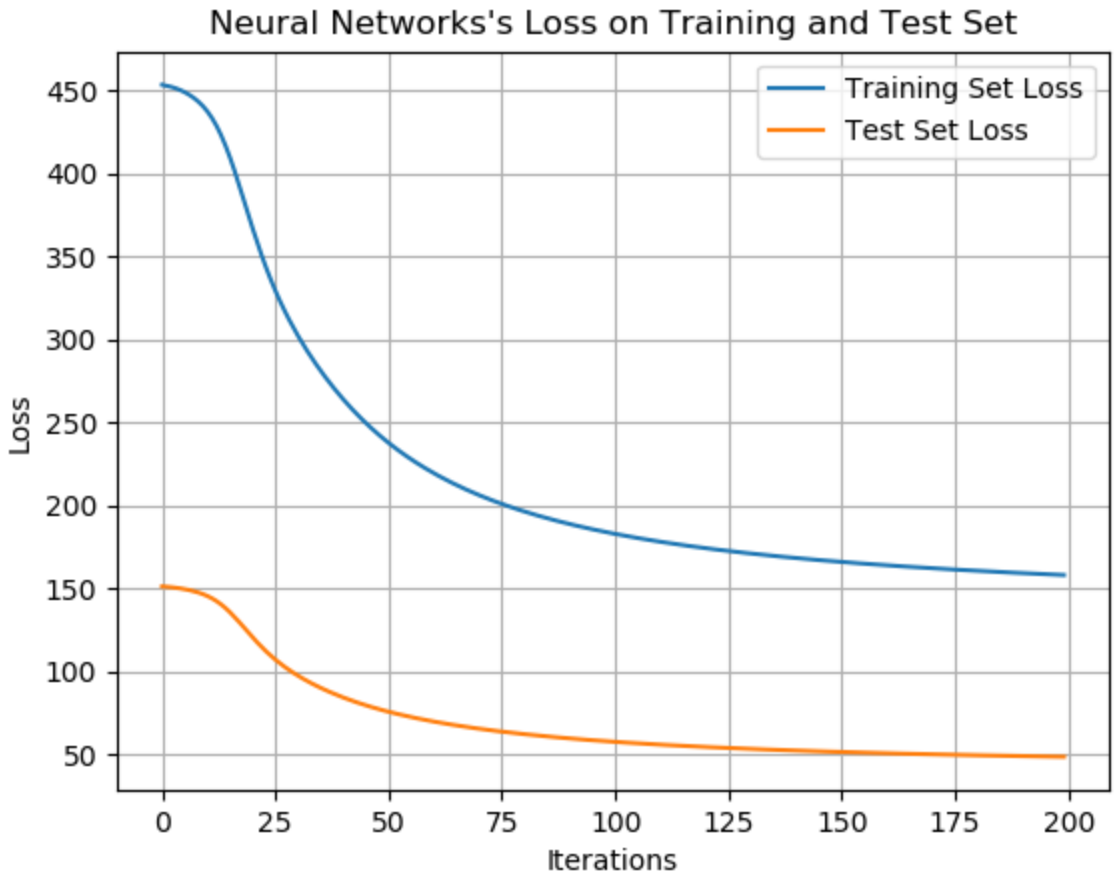


Because we used the absolute value of the weight to compute the pixel intensity, these 2 visualizations mean the importance of features on the 2 nodes in the hidden layer. Such as:

- The place that marked as black, will means these parts of the image is provide many information (no matter is positive or negative) to the final result.
- The place that marked as gray, will means these parts of the image is provide some information to the final result.
- The place that marked as white simply means for this node, these features don't provide and contribute to then end result.

## 3 Underfitting and Overfitting

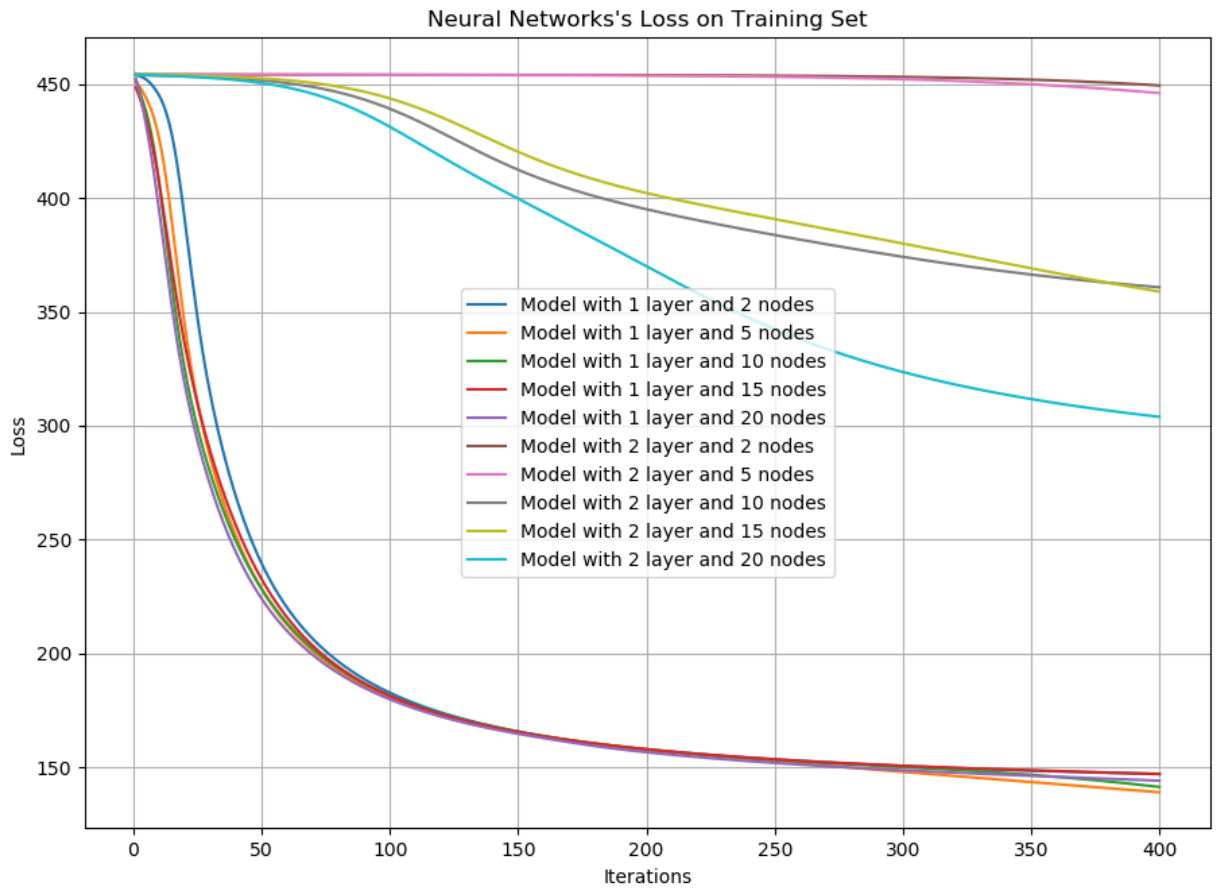
To check the underfitting and overfitting of the model (1 layer and 2 nodes) we run the training iteration again and plot the predict loss on **Training Set** and **Test Set** from each iteration. By checking this graph, we can find that:




- When the iteration number is less than about 5, both loss on **Training Set** and **Test Set** are both high and not start decreasing, this means the model now is underfitting that it cannot performance good on both **Training Set** and **Test Set**.
- When the iteration greater than about 125-150, the decrease speed of loss on **Test Set** are become slow than **Training Set**. This should mean the model is starting overfitting the **Training Set**. But since the test accuracy still becomes better, this might because of the data set for Training and Test is not uniform distribution.

And by checking different models we can find that the *Neural Networks* with 2 hidden layers will performance worse than the *Neural Networks* with just 1 hidden layer. This may mean that:

- More layer doesn't mean better result, it may underfitting on the **Training Set** for a long time. Such as, for *Neural Networks* with 2 layers and 2 nodes, the total loss on both **Training Set** and **Test Set** didn't decrease. But when we re-run the models with 400 iteration, then we can find the loss just start decreasing.


Neural Networks's Loss on Training Set
Loss
Iterations
450
400
350
300
250
200
150
0
50
100
150
200
250
300
350
400
Model with 1 layer and 2 nodes
Model with 1 layer and 5 nodes
Model with 1 layer and 10 nodes
Model with 1 layer and 15 nodes
Model with 1 layer and 20 nodes
Model with 2 layer and 2 nodes
Model with 2 layer and 5 nodes
Model with 2 layer and 10 nodes
Model with 2 layer and 15 nodes
Model with 2 layer and 20 nodes