CSE 574 Introduction to Machine Learning
Programming Assignment 1
Handwritten Digits Classification

Team Members :
Adithya Ramakrishnan - 5009 8106
Shiyamsundar Soundara Rajan – 5009 7590
Vivekanandh Vel Rathinam – 5009 8075

Objective:

The objective of the project is to implement two different methods of classification (K-Nearest Neighbors and Neural Networks) and compare their performance in classifying handwritten digits.

K-Nearest Neighbors :

The K-Nearest Neighbor algorithm is a method used for classification where a set of k closest training examples are identified in the feature space and the output is classified as one of the k closest samples.

In KNN classification the object is classified based on its neighbors, it is assigned to the class to which the majority of its neighbors belong to.

Procedure :

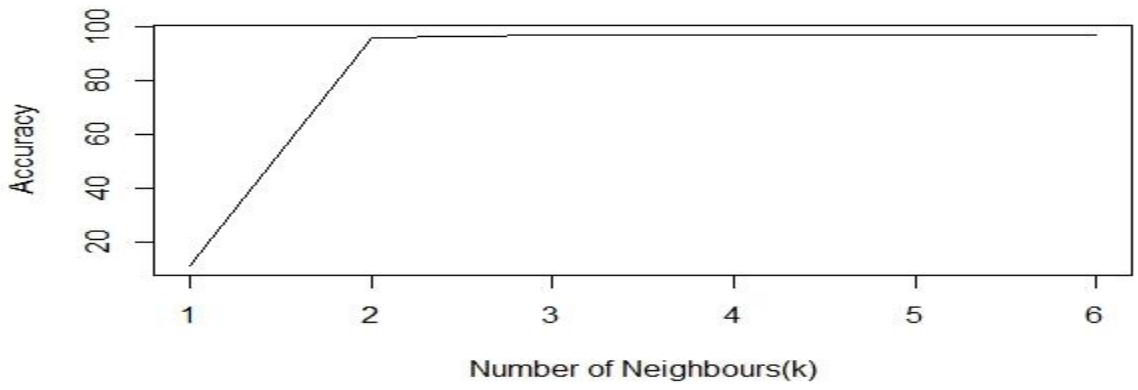
1. To perform the KNN classification we first call the pdist2() giving it the data and the K value which represents the number of neighbors to consider. This function returns the distances D and the Index values I associated with it.

2. We retrieve the corresponding labels based on the index value I and find the majority label from it.

Implementation :

```
train_data=double(train_data);
test_data=double(test_data);
[D,I] = pdist2(train_data,test_data,'euclidean','Smallest',k);
predicted_label = train_label(I,:);
predicted_label = reshape(predicted_label,size(I,1),size(I,2));
label = mode(predicted_label);
label = transpose(label);
```

Analysis :



1. As we can see from the graph the accuracy stays very consistent when the number of neighbors is more than 1.

2. The accuracy is very low when k=1 because the object is assigned the same class as it nearest neighbor. The probability that the object belongs to the same class as its nearest neighbor is very low. Hence the accuracy takes a hit when we consider just one neighbor.

3. The accuracy reached **96.85 for K=3** which is the highest accuracy achieved for the given testing set.

Advantages :

1. The KNN method does not need to train the data.

2. Effective if training data is large.

Disadvantages :

1. The KNN method takes a lot of running time during the prediction as it has to compute the distance between the input image and every other image present before it can classify the image.

2. KNN is very sensitive to irrelevant or redundant features.

Neural Networks :

Procedure :

Forward Propagation :

1. First we load the feature vectors into the input matrix combining all the given input data. Then we train the model using the given training data.

2. For training we create two matrices one from the weights of links between the input and the hidden units and another from the weight of links between the hidden units and the output.

The value of each output is found using the following formula :

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma \left(\sum_{j=0}^{M} w_{kj}^{(2)} h \left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \right) \right)$$

Backward Propogation :

1.In the backpropogation we compute the error in prediction that has occurred and transmit this backward and update the weights of the links accordingly.

2. To achieve this we find the derivative of the error function with respect to both the weight from ouput to hidden units and from the hidden units to the input.

The following formula is used to find the derivative of the error function with respect to weights from the output to hidden units :

$$\begin{aligned} \frac{\partial E_n}{\partial W_{kj}^{(2)}} &= \frac{\partial E_n}{\partial y_k} \frac{\partial y_k}{\partial b_k} \frac{\partial b_k}{\partial W_{kj}^{(2)}} \\ &= \delta_k z_j \end{aligned}$$

$$\delta_k = \frac{\partial E_n}{\partial y_k} \frac{\partial y_k}{\partial b_k} = -(\frac{t_k}{y_k} - \frac{1 - t_k}{1 - y_k})(1 - y_k) y_k = y_k - t_k$$

The formula for the derivative of error function for the weights from the hidden units to the input is as follows :

$$\begin{aligned}\frac{\partial E_n}{\partial W_{ji}^{(1)}} &= \sum_{k-1}^{K} \frac{\partial E_n}{\partial y_k}\frac{\partial y_k}{\partial b_k}\frac{\partial b_k}{\partial z_j}\frac{\partial z_j}{\partial a_j}\frac{\partial a_j}{\partial W_{ji}^{(1)}} \\ &= \sum_{k-1}^{K} \delta_k W_{kj}^{(2)}(1-z_j)z_j x_i \\ &= (1-z_j)z_j(\sum_{k-1}^{K}\delta_k W_{kj}^{(2)})x_i\end{aligned}$$

3. Once we find the derivative of the error function we find the gradient of error function and update the weights using the same

The equation is as follows :

Gradient error function:

$$\nabla E(W) = \frac{1}{N}\sum_{n-1}^{N}\nabla E_n(W)$$

Updating the weights:

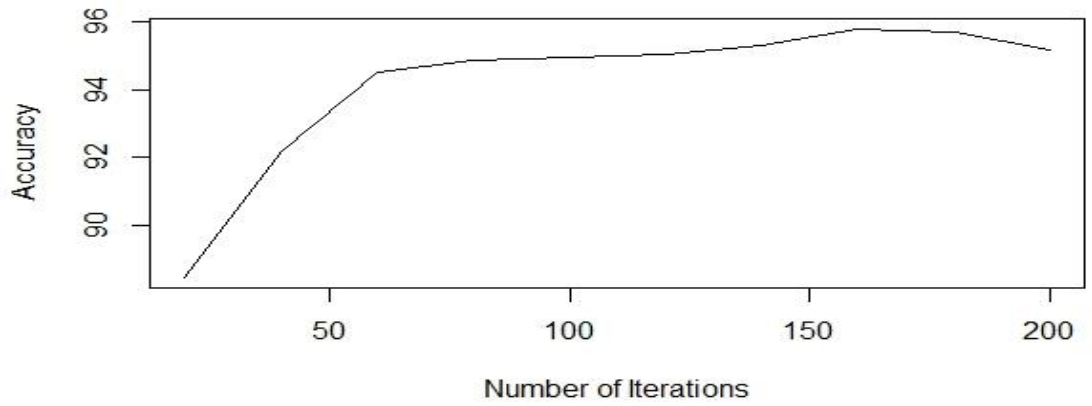
$$W^{new} = W^{old} - \gamma \nabla E(W^{old})$$

<u>Analysis</u> :

The analysis is performed by varying three parameters one by one in the following order :

<u>Number Of Iterations</u>

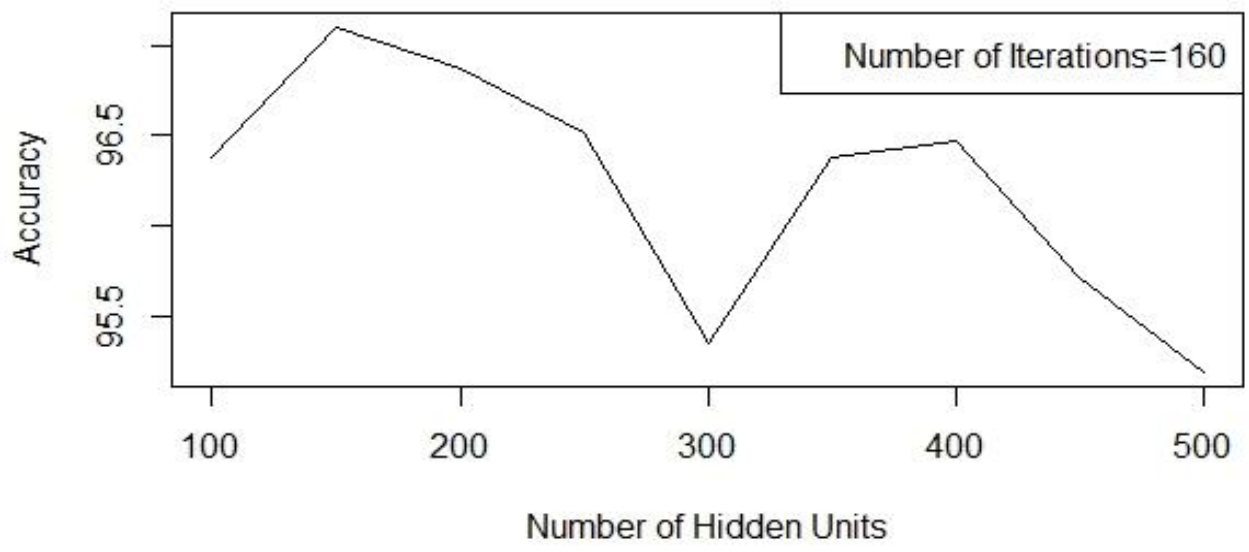
For the hidden units=50 and learning rate=0 we get the following graph varying the number of iterations.



As we can infer from the graph as the number of iterations increases the accuracy seems to increase and attains the highest value at **iteration=160** for which the testing set accuracy is **95.79**

<u>Number Of Hidden Units</u> :

Keeping the number of iterations at 160 we plot the graph by varying the number of hidden units

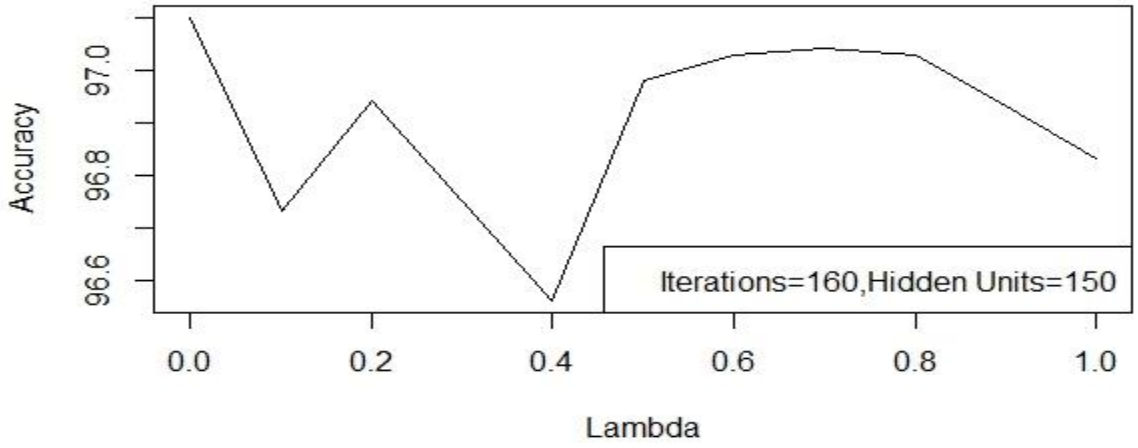


The hidden neuron can influence the error on the nodes to which their output is connected. The stability of neural network is estimated by error. The minimal error reflects better stability, and higher error reflects worst stability. The excessive hidden neurons will cause over fitting; that is, the neural networks have overestimated the complexity of the target problem. It greatly degrades the generalization capability to lead with significant deviation in prediction.

In accordance with the above statement the accuracy was unpredictable for varying number of hidden units. By trial and error we found that the accuracy attained an accuracy of **97.1 for number of hidden units =150.**

Varying Lambda Value :

We vary the Lambda value keeping the number of iterations and the number of hidden units constant at 160 and 150 respectively.



As we can see from the above graph the accuracy varies randomly as we increase the learning rate. A definite conclusion about the relation between the two cannot be drawn from the above graph. By trial and error we find that the accuracy reaches **97.1 for lambda=0.3** which is the highest. By doing this we avoid both underfitting and overfitting.

Advantages :

1. It can adapt to unknown situations
2. Ease of use, learns by example, and very little user domain-specific expertise needed.

Disadvantages :

1. Large complexity of the network structure.
2. It is not accurate.

Comparison Between The Two Classification Methods :

Classification Method	Accuracy	Time Taken
KNN	**96.85**	**1084.874097**
Neural Networks	**97.1**	**980.516253**

References :

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.