Report

Handwritten Digits Classification

CSE 574 Assignment 1

Submitted By:-

Name	**UBIT name**	**Person #**
Pratik Bhat	pratiksu	50096806
Anirudha Karwa	akarwa	50097742
Andreina Uzcategui	auzcateg	50056733

Q.1 Understanding of both the classification methods

Neural Network based Classification:

It is a type of learning which is more applicable to problems which contain noisy input data. In Neural Networks based classification, knowledge is represented in the patterns of interactions between network components or nodes. A neuron consists of multiple inputs and a single output. Each input is modified by a *weight*, which multiplies with the input value. The neuron will combine these weighted inputs and, with reference to a threshold value and activation function, use these to determine its output.

A multi layer Neural network generally comprises of 3 layers namely:-

Input layer: The feature vector of an image is fed as input to the neural network.

Hidden Layer: The layer is present between input and output layers. The number of hidden inputs represents the dimension of learned features in a neural network.

Output Layer: For our classification problem, each class of output (digits 0-9) represents the output node.

Phases in training a neural network:

Feed Forward Network: At every layer, each node calculates its activation value and provides input to the next layer. At the outermost layer, what we get is the probability of the image belonging to a specified output class.

Error value calculation: From the difference between the desired response and the actual response, the *error* is determined and a portion of it is propagated backward through the network. At each neuron in the network the error is used to adjust the weights and threshold values of the neuron, so that the next time, the error in the network response will be less for the same inputs.

Back propagation: Training data is fed through the network in a forward direction, producing results at the output layer. Error is calculated at every output node based on known target information, and changes to the weights that lead into the output layer are determined based upon this error calculation. The changes to the weights that lead to the preceding network layers are determined; changes are calculated, layer by layer, as a function of the errors determined for all subsequent layers, working backward toward the input layer until all necessary weight changes are calculated for the entire network.

Batch learning is employed in this implementation wherein weights not updated after learning every image but only after learning the entire training data set. Update of weights is handled by the function *fmincg* which calculates conjugate gradient descent and updates the weights after ach iteration of 50,000 images.

The training time for a neural network takes time as it involves a number of iterations to generate a set of weight matrices such that the error produced is the minimum. However predicting time for a neural network takes much less time as compared to KNN based classification.

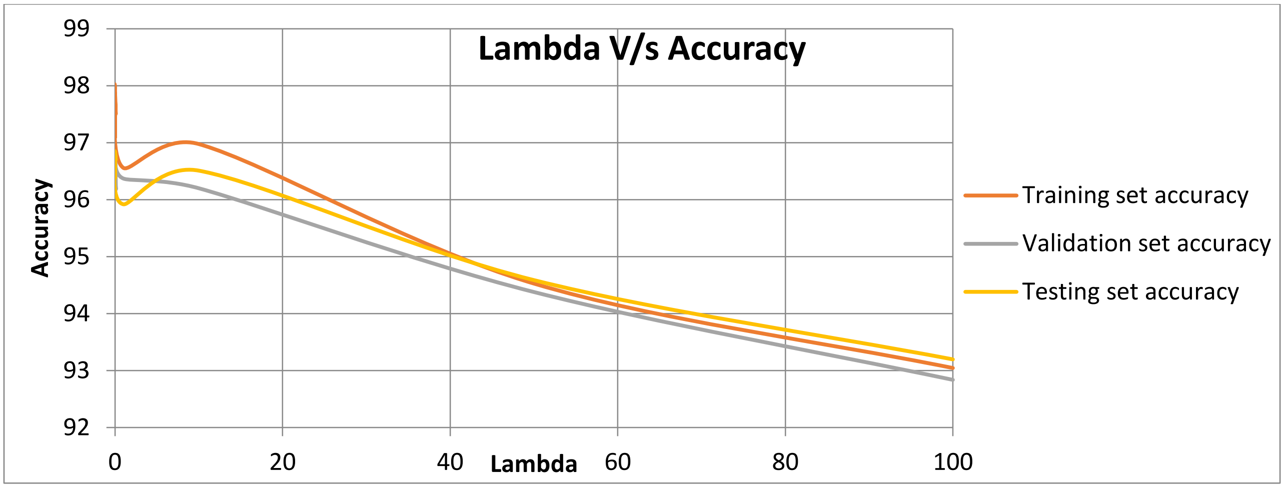
KNN classification:

The KNN classifier is an instance based learning algorithm that does not abstract any information from the training data during the learning phase. Learning is lazily done as there is no explicit training phase. KNN – makes decision based on the entire training data set hence the testing phase is time consuming. Given a image with its data points, KNN computes the distance of this image to all images in the training data set to find the k- closest points.

Given k, it finds k closest points based on the distance. In our case, we employ Euclidean distance. For these set of k points, it determines the label to which majority of the points belong. In case of a tie between 2 sets, the *knnclassify* makes use of nearest neighbor approach by default. The given image is assigned the label that corresponds to the majority of this set of k.

Q.2 Explanation of how to choose the hyper-parameters for Neural Network (number of hidden nodes, regularization term ?) and kNN (parameter k).

Lambda V/s Accuracy



Hidden units	lambda	Iterations	Training set accuracy	Validation set accuracy	Testing set accuracy
100	0.0001	100	97.722	96.730327	96.71
100	0.001	100	97.084	96.620338	96.37
100	0.01	100	97.852	96.830317	96.82
100	0.1	100	97.502	96.180382	96.7
100	0	100	98.004	96.780322	96.84
100	1	100	96.558	96.370363	95.92
100	10	100	96.976	96.20038	96.51
100	50	100	94.528	94.380562	94.59
100	100	100	93.046	92.840716	93.2

As we can see in the above graph, we notice how accuracy changes for different values of lambda (regularization factor).

When the value of lambda is 0 i.e. regularization factor is not included, we see that the difference between the accuracy of training data and that of validation/testing data is highest. Reason is over fitting of the training examples at the cost of decreasing generalization accuracy over the unseen data (validation and test).

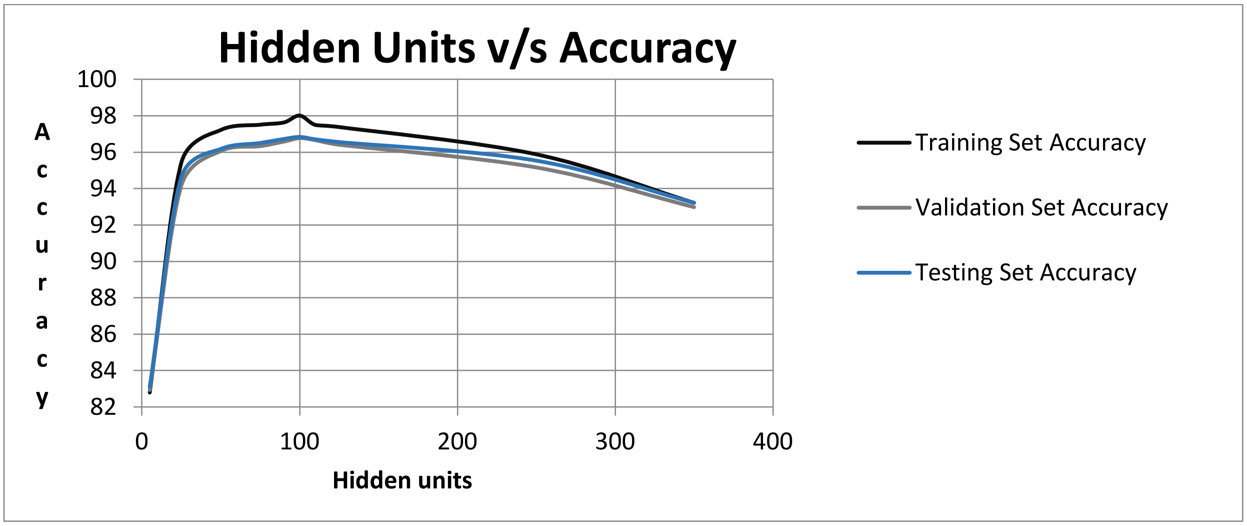
When the value of lambda is 100, we notice that the accuracy decreases significantly for all the data. The problem we face is of under fitting of the examples.

Clearly we need to find a suitable value of lambda to avoid over fitting and under fitting problems. For this purpose, we used the trial and error approach to train the neural network with different values of lambda.

1. For 50 < lambda < 60, the difference between the accuracies of all the three phases is least i.e. the network is trained in much generalized way. But the accuracy is not optimal.
2. For lambda = 0.01, the validation set accuracy is highest.

So the value of lambda depends upon the situation whether you want to generalize the network or have the highest accuracy for the examples to be classified.

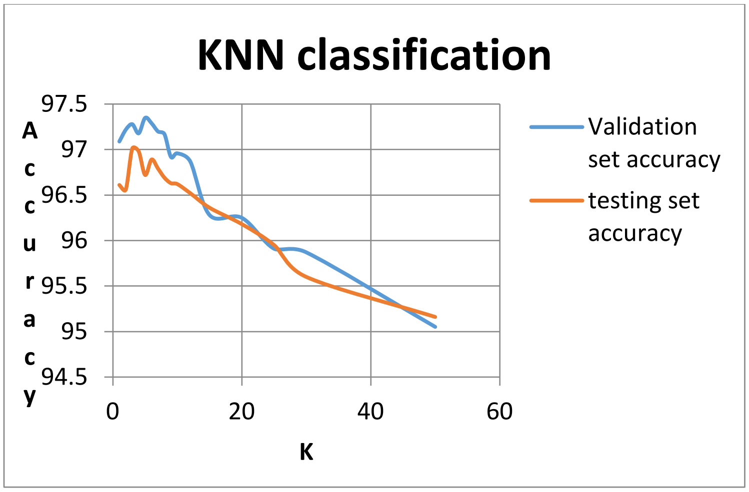
Hidden Unit V/s Accuracy



Hidden Units	Lambda	Iterations	Training Set Accuracy	Validation Set Accuracy	Testing Set Accuracy
5	0	100	82.796	82.941706	83.11
25	0	100	95.414	94.170583	94.63
50	0	100	97.21	96.070393	96.18
75	0	100	97.502	96.340366	96.51
90	0	100	97.626	96.590341	96.73
100	0	100	98.004	96.780322	96.84
110	0	100	97.488	96.670333	96.71
125	0	100	97.374	96.420358	96.56
250	0	100	95.876	95.170483	95.54
350	0	100	93.212	92.990701	93.22

We conform from the above graph that there is a direct relation between the number of hidden units and the accuracy. We trained the neural network with different number of hidden units.

1. With very less number (5) of hidden units the accuracy is least.
2. With the number of hidden units almost equal to half the number of input units (785), the accuracy is above 90% but not optimal.
3. For our network we found out that the accuracy is the maximum for 100 hidden units.



K	Validation set accuracy	Testing set accuracy
1	97.090291	96.61
2	97.220278	96.56
3	97.280272	97.01
4	97.180282	96.98
5	97.350265	96.72
6	97.290271	96.89
7	97.20028	96.79
8	97.170283	96.69
9	96.920308	96.63
10	96.960304	96.62
12	96.870313	96.52
15	96.280372	96.36
20	96.250375	96.18
25	95.910409	95.95
30	95.870413	95.6
50	95.050495	95.16

1. General strategy is to perform KNN classification using different values of k on validation data.

2. Use the value of k for which the prediction accuracy is highest for the validation data.

Q.3 Comparison of the Performance between 2 classification methods:

1. *Accuracy:*

During our experiments, we have found that both the methods are at par with regards to accuracy.

Accuracy of a neural network depends on parameters like lambda or regularization parameter and the number of hidden nodes as well as number of iterations. In our experiments, we have observed that the highest validation set accuracy is achieved when lambda is between 0.01 – 0.001.

Varying the number of hidden units also gave increased accuracy, the highest accuracy was achieved with hidden units = 100

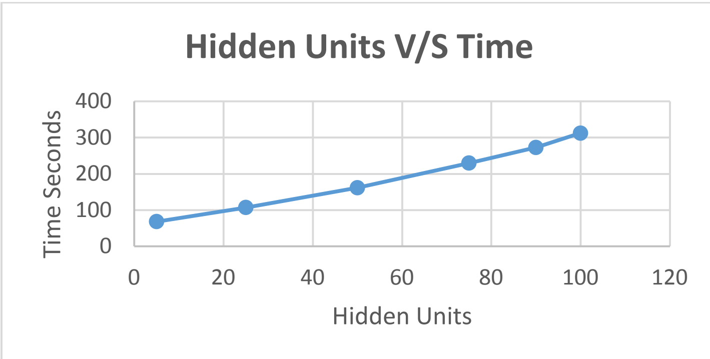
For KNN method, the accuracy depends on 'k'. Based on repeated readings and our observations for the given dataset, we have observed that accuracy reaches a maximum for k = 5 or 6 and then keeps on decreasing. Fluctuation in accuracy while tuning parameters is observed more with Neural Network (82 – 97) as compared to KNN (95 – 97).

Learning time:

- The learning time of a neural network depends on various factors like no. of hidden units, size of the neural network (no. of hidden layers) and no. of iterations. Learning time may also increase as the training set gets bigger and bigger.

Hidden Units V/s Time

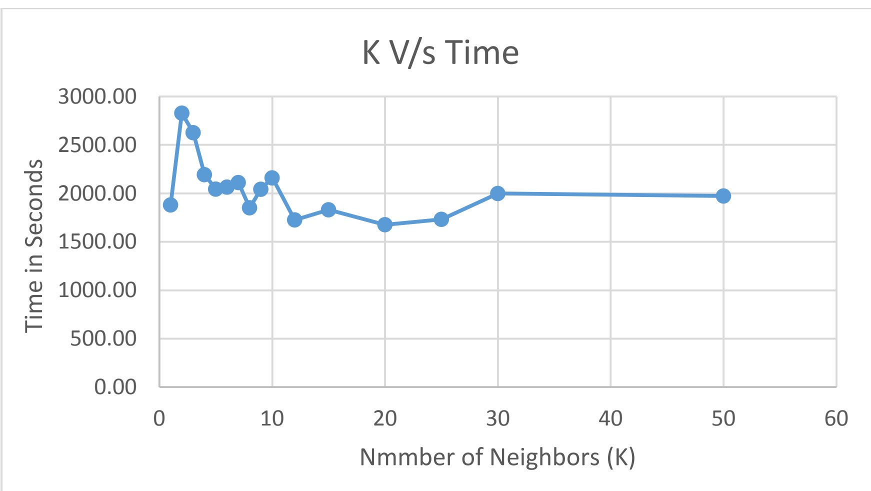
Following is the graph for hidden units against time for a constant lambda = 0 and Iterations = 100



Hidden Units	Time(Seconds)
5	68.3668
25	107.3136
50	161.7157000
75	229.923
90	272.7879
100	312.2722

We see that the time required to train the neural network is directly proportional to the number of hidden units. As number of hidden units increases, time required to train the network increases.

- In KNN based classification, there is no training phase. It assumes no prior information related to the data set. Given an image, it computes its distance with each image in the training data set. Hence the prediction phase takes a long time as compared to neural networks classification. As a result, learning time will grow as the training data increases.



K	Seconds
1	1880.50
2	2827.60
3	2626.40
4	2193.00
5	2043.90
6	2064.60
7	2111.40
8	1851.20
9	2042.40
10	2159.60
12	1726.20
15	1830.60
20	1676.60
25	1732.00
30	1998.60
50	1972.80

Based on our observations, we can say that the total time to produce an accurate reading taken by neural networks is very less when compared to KNN approach.

Q.4 Advantages and Disadvantages of both the classification methods

Neural Networks:

Advantages:

1. A trained neural network takes less time to predict a given test image accurately.
2. A neural network is resistant to noisy input data.
3. A neural network can capture many kinds of relationships and relatively easily model the phenomena which otherwise may have been very difficult or impossible to explain.
4. Once trained, it can be used multiple times provided classification features remain the same.

Disadvantages:

1. Takes a long time for training phase. These times depend on factors like number of hidden units, training data and even the size of the network.
2. Neural network suffers from over fitting and under fitting for training samples, avoiding these problems can be tricky or cumbersome.
3. The number of parameters to be tuned (no. of hidden units, lambda and weights) is more in neural network.

KNN:-

Advantages:

1. The main and most important advantage of k-NN method is its simplicity.
2. Good performance on large training sets.
3. Few parameters to tune (distance evaluation metric and k) as compared to lambda, hidden units, number of iterations, error findings, weights, etc. in neural networks.

Disadvantages:

1. Not robust to noisy data.
2. For large training sets, require more memory and is time consuming.
3. Very redundant as the algorithm needs to compute the distance from all the training examples in every prediction.
4. For some complicated tasks, KNN may not be the right algorithm.