CSE 574

INTRODUCTION TO MACHINE LEARNING

PROGRAMMING ASSIGNMENT 2

HANDWRITTEN DIGITS CLASSIFICATION

ALEKYA KUMAR

(50249052)

TRUPTI JADHAV

(50249177)

Neural Networks:

NN are computing systems vaguely inspired by the biological neural networks that constitute animal brains.[1] Such systems "learn" tasks by considering examples, generally without task-specific programming.

Most machine learning algorithms involve “hyperparameters” which are variables set before actually optimizing the model's parameters. Neural networks can have many hyperparameters, including those which specify the structure of the network itself and those which determine how the network is trained. In particular, we are focusing on feed-forward neural nets trained with gradient descent.

Setting the values of hyperparameters can be seen as model selection.

There are two ways in which hyper-parameters are tuned:

- **Manual Tuning**: The modeler is responsible for searching in the hyper-parameter space to test different parameter combinations.
- **Automated Tuning**: The hyper-parameter search is automated and is made part of the training algorithm.

The various hyperparameters are:

Learning Rate :

The learning rate ϵ eta determines how quickly the gradient updates follow the gradient direction. If the learning rate is very small, the model will converge too slowly; if the learning rate is too large, the model will diverge.

Number of Training Iterations :

This can be a powerful way to prevent overfitting, to the extent that it may make the choice of other hyperparameters less important. The validation performance should be evaluated infrequently; for example every time the algorithm has seen several times more new examples than there are in the validation set.

Number of Hidden Units :

Large hidden layers can allow the neural network to fit the training data arbitrarily well, but because regularization is typically used, it’s mostly important to just use large hidden layers. Using the same size for all hidden layers generally works better or the same as using a decreasing or increasing size. In addition, using a first hidden layer which is larger than the input layer tends to work better.

Regularization :

Regularization is any modification we make to a learning algorithm to reduce its generalization error but not its training error. We use regularization in Neural Network to avoid overfitting problem

Feature Selection:

As required, the features which have exactly the same values for all the datapoints were removed at the preprocessing step of neural network(nnScript and facenn).

The actual variables are 784 and after featutre selection they are reduced to 717 and submitted with the code.

Part A: Implement Neural Network (forward pass and back propagation)

The dataset used for running the neural network is the MNIST dataset consists of a training set of 60000 examples and test set of 10000 examples. All digits have been size-normalized and centered in a fixed image of 28*28 size.
The nnScript file contains following functions:
preprocess(), sigmoid(), nnObjFunction(), nnPredict() and initialiseWeights(). The required functions are coded and updated.

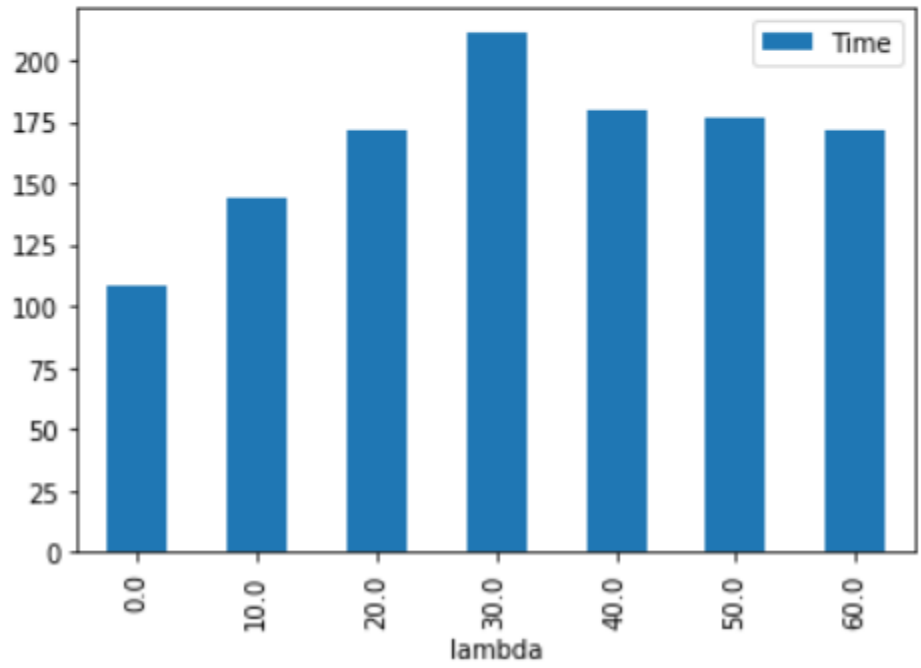
Part B: Incorporate regularization on the weights

We are varying lambda from 0 to 60 with step size 10.
The hidden units takes the values as 4, 8, 12, 16, 20.

Part C: Tune hyper-parameters for Neural Network

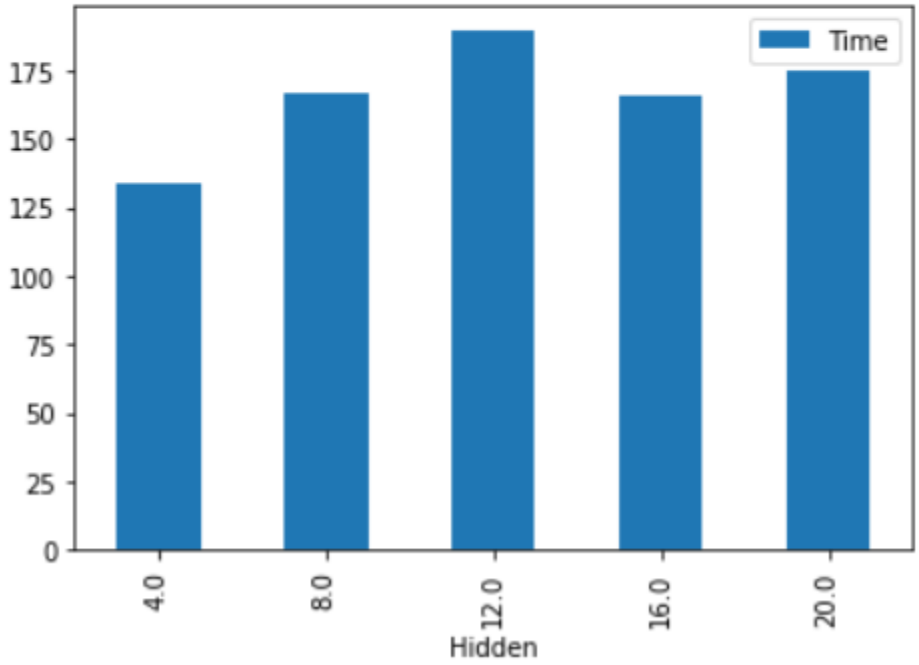
The following graph shows the average time the network takes for each of the lambda values. The time is indicated in seconds. For a regularization parameter of 30, the average time taken to run the network is 211 seconds.

lambda	Time
0.0	108.411342
10.0	144.547639
20.0	171.545574
30.0	211.007193
40.0	179.404666
50.0	176.987725
60.0	171.191414



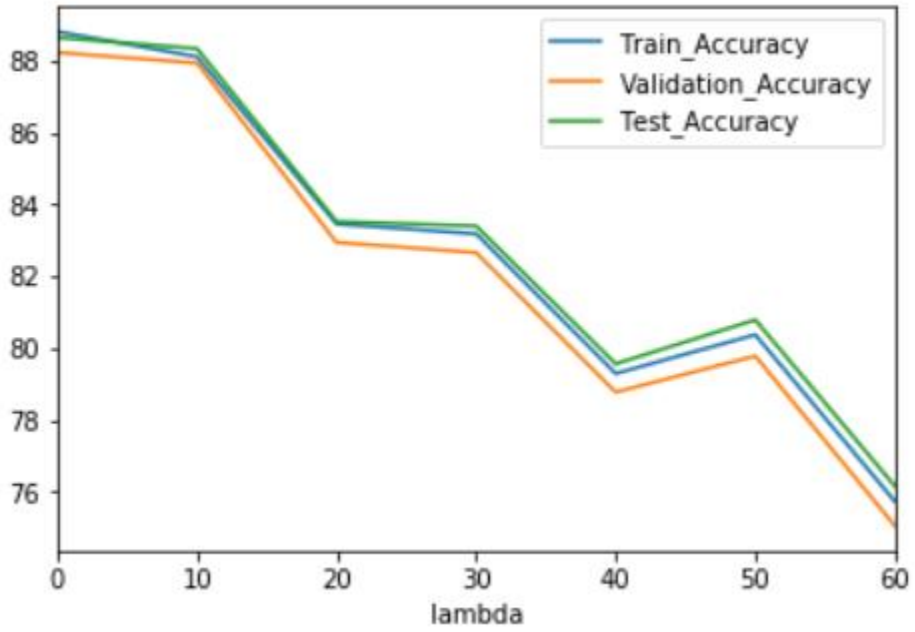
The following graph shows the average time each configuration of hidden units has taken while running the network. Surprisingly, the network with 12 hidden units has taken the maximum time.

Hidden	Time
4.0	133.941034
8.0	167.084114
12.0	189.180547
16.0	166.150243
20.0	174.426600



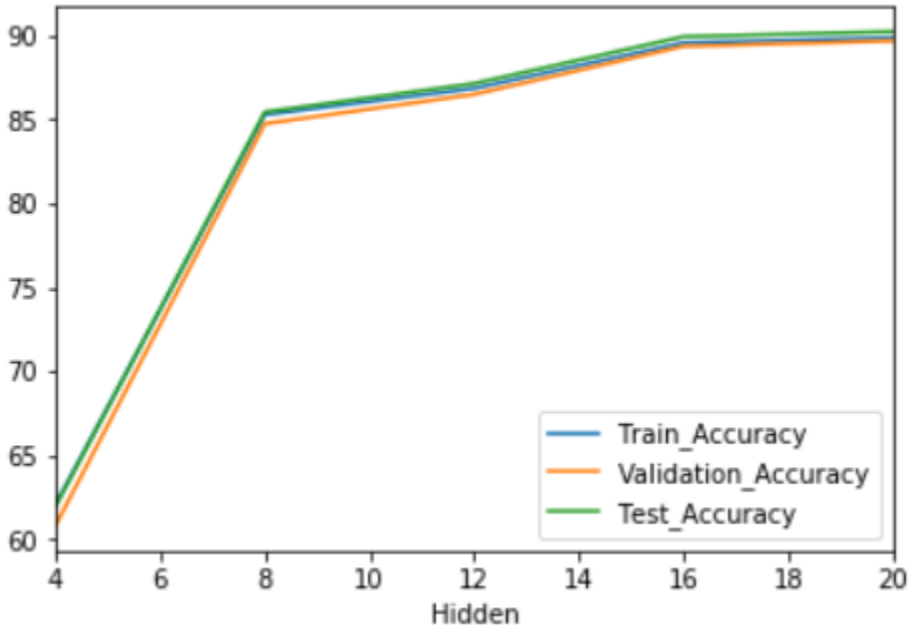
The table displayed below shows the Accuracy for each of the Data sets – Training data, Validation Data and Test Data. When the value of lambda increases, the performance keeps decreasing with all three networks showing almost a similar pattern. There seems to be an overfitting with lower values of lambda. As lambda increases, it moves towards models with lower complexity, thus giving us accurate results.

	Train_Accuracy	Validation_Accuracy	Test_Accuracy
lambda			
0.0	88.8124	88.232	88.648
10.0	88.1132	87.938	88.340
20.0	83.4568	82.944	83.524
30.0	83.1832	82.660	83.400
40.0	79.2916	78.774	79.562
50.0	80.3732	79.780	80.788
60.0	75.7340	75.054	76.160



The following table shows the average accuracy for increasing Hidden Units for Train Data, Validation Data and Test data. As we can see from the graph, the accuracy of the model increases when the number of hidden units increase. For a hidden node value of 20, the model gives the highest accuracy of 90.2%

	Train_Accuracy	Validation_Accuracy	Test_Accuracy
Hidden			
4.0	62.089143	60.820000	61.958571
8.0	85.254857	84.724286	85.420000
12.0	86.823143	86.465714	87.111429
16.0	89.537429	89.328571	89.900000
20.0	89.841429	89.648571	90.197143



Summary:

The optimal hyper parameters are as follows:

- Lambda Value = 60
  We are choosing a high lambda value as the higher ones give less model complexity thus avoiding overfitting.
- Hidden Unit = 20

  The higher the number of hidden units, the better the accuracy as we can see from the graph. The hidden unit of 20 nodes gives the hihest accuracy ion Test Data .

Part D: Comparing the results of deep neural network with normal neural network.

CelebFaces Attributes Dataset (CelebA) [3] is a large-scale face attributes dataset with more than 200K
celebrity images. The subset will consist of
data for 26407 face images, split into two classes. One class will be images in which the individual is wearing glasses and the other class will be images in which the individual is not wearing glasses.
The deepnn file has the following functions:
preprocess(), sigmoid(), nnObjFunction(), nnPredict(), create_multilayer_perceptron(), create_multilayer_perceptron_3(), create_multilayer_perceptron_5(), create_multilayer_perceptron_7() and Calculate().
The accuracy of the deep neural network for 3, 5 and 7 hidden layers are as follows:

Hidden layer	Accuracy	Time
2	0.8168054	335.06352186203003
3	0.79295987	336.35260105133057
5	0.71763813	358.0116457939148
7	0.7289932	452.0559723377228

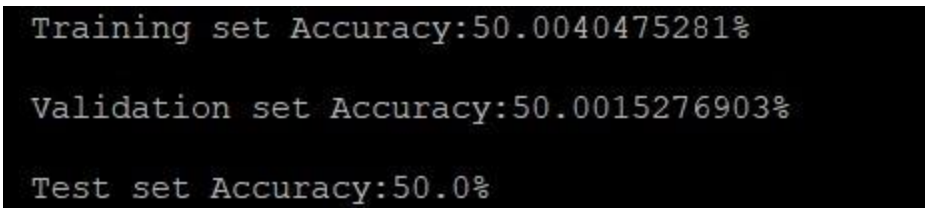
From the above results, we see the accuracy for the double hidden layer is the best and is highest. The accuracy of the network drops down further which is not a good indicator.

If the number of neurons and/or hidden layers are more the model will start to overfit. Hence if the neurons/hidden layers are more then the regularization value should also be more to compensate it.

Increasing the number of hidden layer is always not a wise decision. With increase in the number of hidden layers the complexity of the neural network increases.

We ran the CelebA dataset for both type of neural network i.e deep network(deepnn.py) and normal network(facenn.py).

With more layers, the accuracy of the network does not seem to improve anymore, (79-80%) and it performed a drastically poorer than a single layer network (50%) for the faces dataset.



The accuracies noted for both of them are as follows:

Normal neural network: Around 50%

Deep neural network: Around 70% and 80%

Part E: Extra Credit

Convolutional Neural Network:

CNN are category of neural networks that are widely used in areas such as image recognition and classification. They use a variation of multi layer perceptrons that require minimal preprocessing. CNNs use relatively little preprocessing compare to other image classification methods.

The dataset used for CNN is MNIST dataset.

We have to compare the working and performance of the normal neural network and CNN.

The normal neural networks are fully connected neural nets while CNN's are not. The weights are shared in convolutional neural network so ideally the performance of CNN should be more.

The accuracy for normal neural network is very high i.e 90% and for convolutional neural network it is 98.8% which was expected.

So the CNN performs NN.

Conclusion:

- The explanation on how to choose the hyper parameter for the analysis is given.
- For the MNIST dataset, the corresponding values for different model are recorded. In summary normal neural network performance and accuracy was pretty good (around 90%) for test set. While the CNN's performance was better about 99% which very good.
- For CelebA dataset, the deep neural networks and normal neural network modelling is performed. We saw that normal NN performed really bad (50%). While deep NN performance was as expected around 70%-80%

Accuracy of classification method on the handwritten digits test data