**Machine Learning CS6140**

Kartikeya Shukla

NUID - 0017277475

**Question 1C.**

Decision Tree: [167.0, 13.0, 24.0, 256.0]

Linear Regression: [26.0, 1.0, 165.0, 268.0]

Logistic Regression: [95.0, 8.0, 96.0, 261.0]

**Question 1D.**

**Question 3B.**

The purpose of this training algorithm is to train the network such that the weights between the input layer and the hidden layer and, the hidden and output layer are such that, when a binary input is given in the input layer, the hidden layer is able to encode and decode it, and provide the same output in the corresponding output

node.

**Question 3C.**

No, this encoder-decoder scheme will not work for networks having less than 3 hidden units in the hidden layer.

This is because in order to correctly predict the output (with a layer with 8 inputs) we would need at least 3 bits. i.e. 2^3 = 8 possible outputs.

If the outputs/inputs were more than 8 say **X**, we would need at least **Z** hidden inputs where 2^**Z** = **X**.

**Question 4.**

Consider a three-layer network having linear transfer functions, an input layer **x**, a hidden vector **y** and a output vector **z.**

For this network we have two sets of weights matrices W1 and W2, one between the input-hidden layer and another between the hidden and output layer. They can be represented by the equations,

y = W1x

z = W2y,

i.e. z = W2W1x

z = W3x

(a new weight vector W3)

This appears to be similar to the equation for two-layer network but with weight vector W3.

Suppose we represent a non-linearly separable with a three layer network, from the above prove, this would mean we can represent it as a two-layer network as well. Having done this, it would mean that the problem is in fact linearly separable , which is a contradiction.

**Question 5.**

*Advice from applying Machine Learning – Andrew Ng*

In this lecture Andrew Ng talks about three things:

1. Debugging/Diagnostics of learning algorithms

Suppose we try our error function and are not satisfied with the result, we can try and improve the algorithm in a number of ways which are listed down as follows and we explore the intuitions behind each of them

– Try getting more training examples.

– Try a smaller set of features.

– Try a larger set of features.

* Try changing the features: Email header vs. email body features.

The above four are likely to be due to high variance and high bias, if the training error is high and is lower than the testing error.

– Run gradient descent for more iterations.

– Try Newton’s method.

– Use a different value for λ.

* Try using an SVM.

Another problems with learning algorithms are that they require mathematical optimizations to be implemented. We could optimize the objective function or the algorithm itself.

Even if the algorithm is running correctly we should run diagnostic on it,. It is important to know that the algorithm works but it is also required to know why it works.

1. Error analyses

Error analysis tries to explain the difference between current performance and prefect performance and Ablative analysis tries to explain the difference between some baseline and the current . Removing different components from the learning system and checking the error each time gives an in-depth view of how these features are affecting the system and what needs to be stream-lined in order to get an acceptable or even a perfect learning system.

1. And how to get started with machine learning problems

There are two approaches to building to learning system Careful design and Build and fix

The first, where we spend a long term designing exactly the right features, collecting the right dataset,

and designing the right algorithmic architecture. Implement and hope it works. And the second implement something quick and dirty, run error analysis and diagnostics to see if it works. Each have their own advantages and disadvantages.

*‘A few useful things to know about Machine leaning’*

Machine learning system automatically learns from data and thus nullifying the need to re-implement solutions. It is used in several applications like Spam Filter, Web Search, Recommender Systems, Ad Placement, Credit Scoring, Fraud Detection, Drug Design and many others.

The paper also notes how most machine learning projects take time to wind up and producing less than ideal results over time. This is partly becomes the author believes people who start these machine learning project are not aware of how to approach them. He uses the problem of a most common learning algorithm – classifier to explain the approach.

A machine learning problem mainly consists of three important parts:

* Representation

A problem/solution has to be expressed in a formal language understood by the computer.

* Evaluation

An evaluation function (objective or scoring function) is needed to distinguish good classifiers from bad ones. Internal evaluation function is different from the external evaluation function, which we want the classifier to optimize.

* Optimization

A method to search among the classifiers, in the language for the highest-scoring one. The choice of optimization technique is key to the efficiency of the learner.

Main Points to keep in mind are:

*Generalization:*

We should make sure that data is not overfitted such that the out of sample error is reduced and we get a more generalized model.

*Intuition fails in high dimensions:*

We see in three dimensions, therefore multiple features needs us to see data in multiple dimension which we can not image and visualize and therefore makes it harder to code.

*Theoretical guarantees are not sufficient –*

Machine learning requires practical results since, application will be used in real life, theoretical proofs are not enough.

*More data beats a clever algorithm-*

As mentioned Machine problem runs on data, therefore it is simple to say that the more data we have the better we can get the model trained and the better performance it will give. Rather than having a algorithm that manipulate very well a small data set, since, in real life there will always be more data coming in.

*Learn multiple Models-*

Machine learning requires different models to be learnt, as data changes, the model need to be able to adapt and change. Also, having multiple models gives the opportunity to engage in boosting and bagging, which involves training data with different models and choosing the best performing one.

*Simplicity does not imply accuracy -*

Occam’s razor famously states that entities should not be multiplied beyond necessity. The same is consider in machine learning far too often. One should not be too eager to accept a simple straightforward answer/model.

*Representable doesn’t mean Learnable-*

And finally, the very fact that a problem is representable does not mean the problem is learnable. We need to do enough research before hand such, it can be determined that the question that we are asking the machine to learn is possible to do so.