Assignment 4 - Report

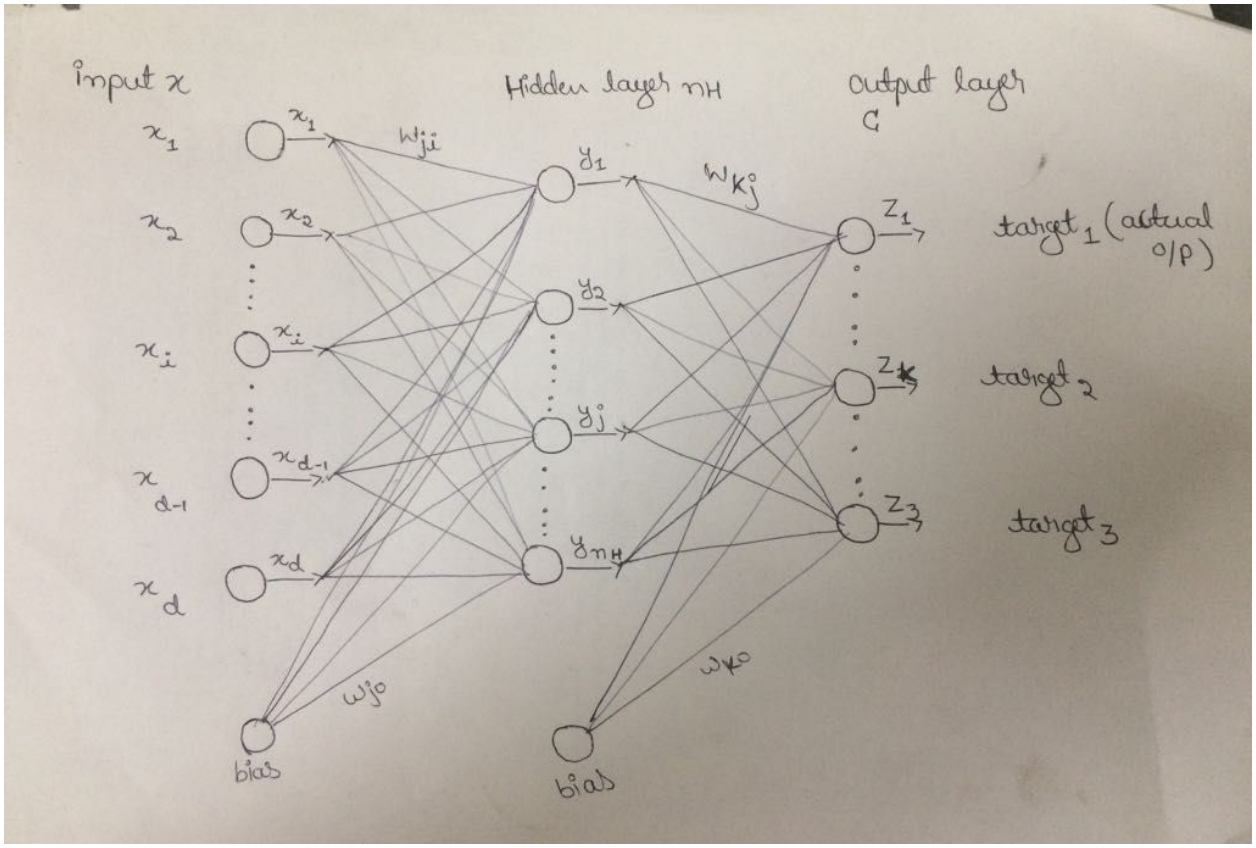
Name : Vatsal Nagda

Roll : 20162008

Subject : SMAI

Question 1 : Training 3-layer Neural Network

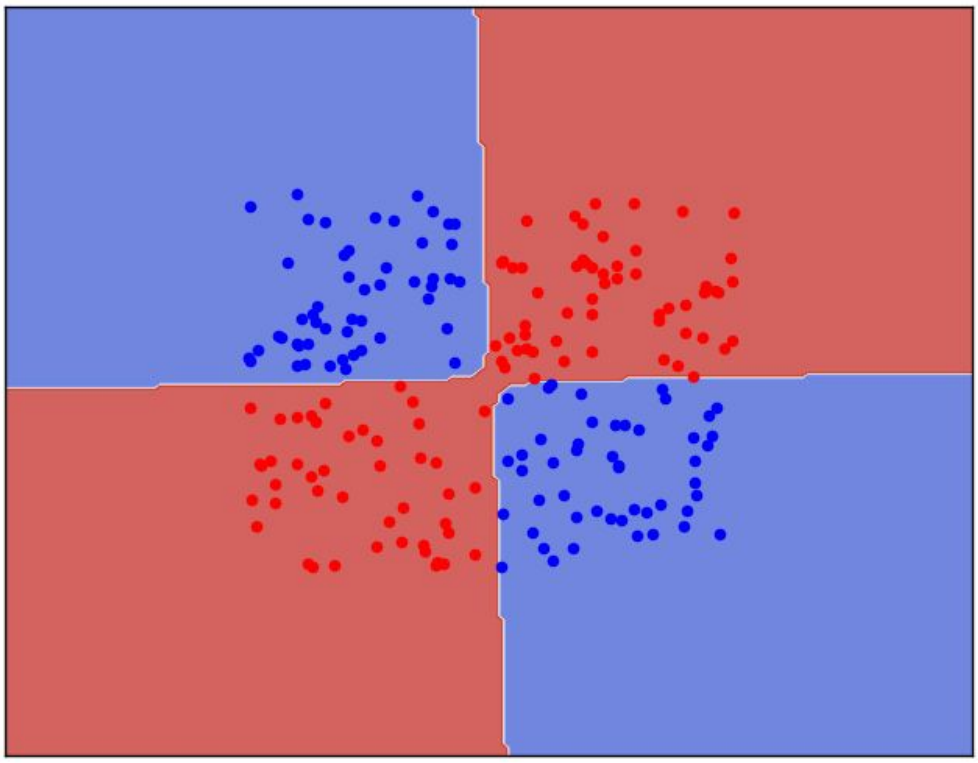
```
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  5
count :  347
Accuracy :  64.6182495345  % .
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  10
count :  526
Accuracy :  97.9515828678  % .
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  15
count :  526
Accuracy :  97.9515828678  % .
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  20
count :  521
Accuracy :  97.0204841713  % .
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  25
count :  534
Accuracy :  99.4413407821  % .
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  30
count :  530
Accuracy :  98.696461825  % .
vatsal Assignment-4 $ python test_v2.py
Hidden Layers:  35
count :  527
Accuracy :  98.1378026071  % .
```



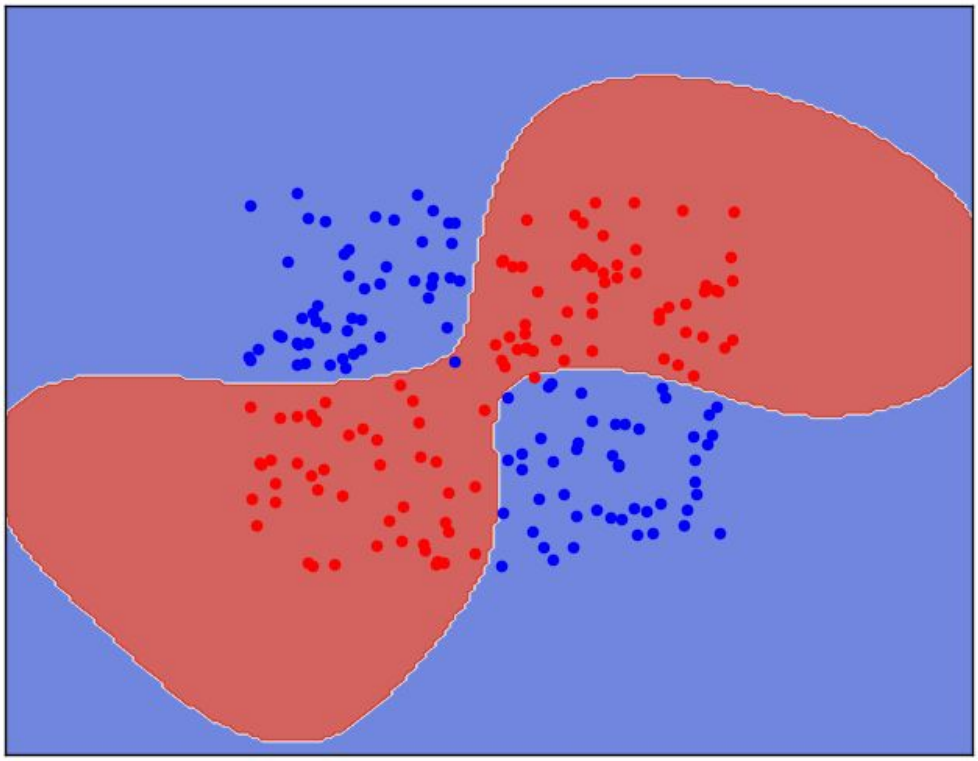
In the problem, we have considered a d-nH-c fully connected graph, where d is 64, nH is variable and c is 3.

For nH = 1 or 2 or 3, we get a low accuracy around 60-70%. But above 3 nodes it drastically increases to around 90% for 4 nodes and 5 and increases gradually till it reaches its peak at 99.44 % at nH = 25, which is like the sweet spot for the network, and after further increasing it beyond it decreases a little bit but is still above 90%. Learned weights and hidden layer outputs are too high to tabulate here. They can be seen by executing the code.

Problem 2 : Support Vector Machine



This is what we get when for Polynomial Kernel SVM, for C=2048 and degree = 2.



This is what we get when for RBF Kernel SVM, for C=8192 and Gamma = 0.5.

A linear kernel allows you to use linear functions, which are really impoverished. As you increase the order of the polynomial kernel, the size of the function class increases. An n-th order polynomial kernel gives you all analytic functions whose derivatives of order (n+1) are constant, and hence all derivatives of and above order (n+2) are zero.

If you use polynomial kernels, you're model is parametric. In a way nonparametric model like (rbf) means that the complexity of the model is potentially infinite, it's complexity can grow with the data. If you give it more and more data, it will be able to represent more and more complex relationships. In contrast a parametric model's size is fixed, so after a certain point your model will be saturated, and giving it more and more data won't help. So asymptotically, assuming you have unlimited data and very weak assumptions about the problem, a nonparametric method is always better.

Problem 3 : Bayes Theory

1. Summary of the data:-

The data given has approx 1.95 lacs lines of records. Prediction task is to determine the income level for the person represented by the record. Incomes have been binned at the $50K level to present a binary classification problem. Data has 42 columns, out of which the last columns represents the class label of each record. There are two types of data- Continuous and Nominal. For sake of simplicity, only Nominal data is being considered for classification.

2. Handling of Missing Entries:-

Missing entries in the data set is represented by '?'. If these fields are left as it is, it may not give optimum results. The missing entries are handled by replacing them with the most frequently occurring sample of the corresponding column having the same class label. Doing so improves the accuracy to some extent.

PROGRAM EXECUTION: