SIV895: Special Module on Intelligent Information Processing
Project Report
Title: Classification of Iris flower species: Analysis using Neural Network.

Submitted By: **Date: 20-Apr-2016**
RAJESH KEDIA
2014CSZ8383

Scope: To train a neural network for the 3 varieties of Iris flower species (Iris-setosa, Iris-versicolor, Iris-virginica) and use the network to classify the inputs. Observe and understand the variations due to the below factors:

1. Change in the training data size
2. Change the number of parameters used for training and testing
3. Change in the number of iterations for training the neural network

Implementation details:
A back propagation neural network (BPNN) code in python was picked up from web. The data set of 150 flowers (50 of each type) was taken and modified in the format that the BPNN could accept. The neural network output is the weightages of each of the type, the code was modified to assign the class for which the maximum weight was observed in output. Then it was compared with the original type and accordingly a match/mismatch was determined. Total number of mismatches were counted.
The neural network was configured to have input size varying from 1, 2, 3, 4 while the output size was fixed to 3. The hidden layer was chosen to be 4 neurons. tanh function was used for error fitting and learning.

There was a provision to vary the iteration count for the neural network during training. This was exploited to study the impact of iteration count on the error count.

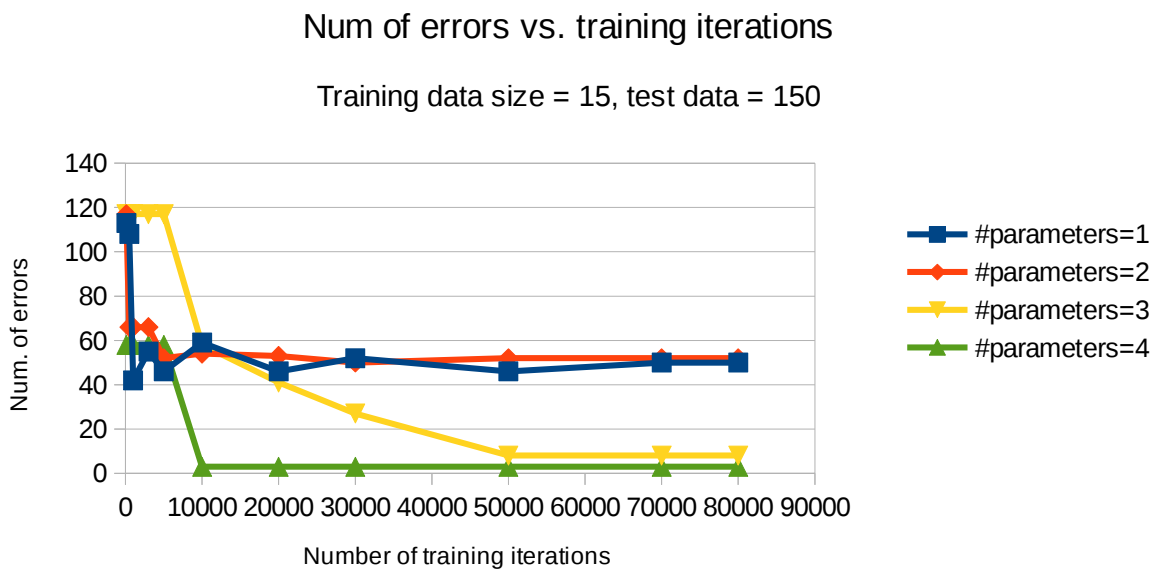
Results and Observations - I:
The next set of 4 graphs have the below conditions:

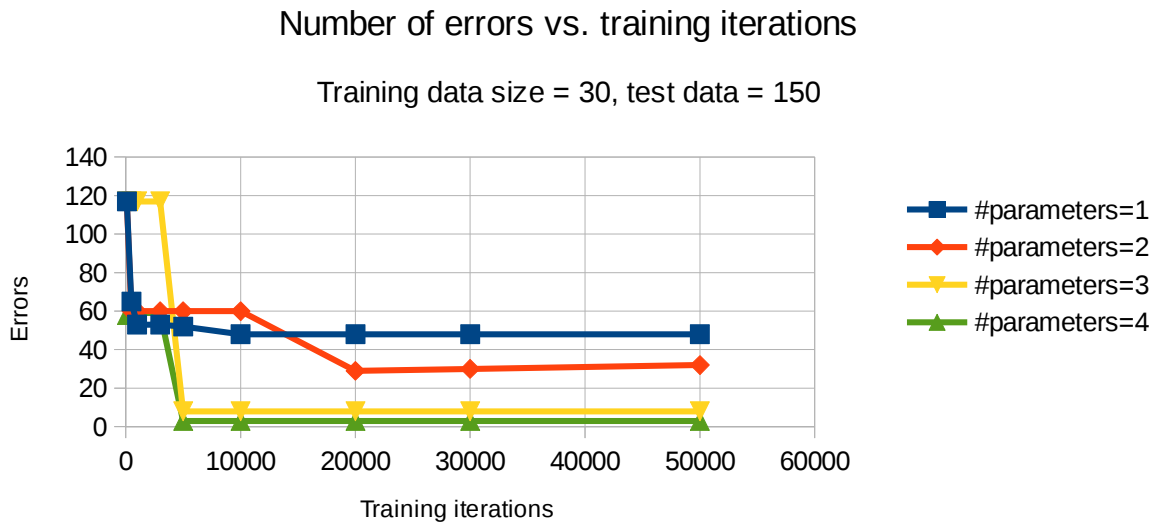
1. Each of them is for a fixed number of training data size. 4 values of data size is chosen: 15, 30, 50 and 150. The test data size is same as 150 in all cases.
2. Each graph consists of 4 plots, each corresponding to how many parameters of the flowers were used during training and testing. There were 4 parameters for flowers and 1, 2, 3 and 4 parameters were used during experiments.

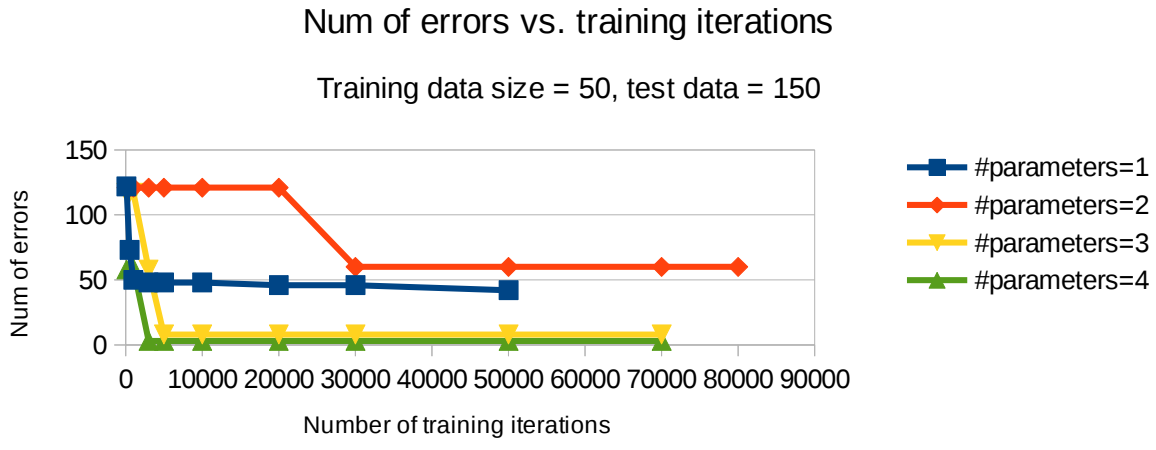
Common Observations/Reasoning:

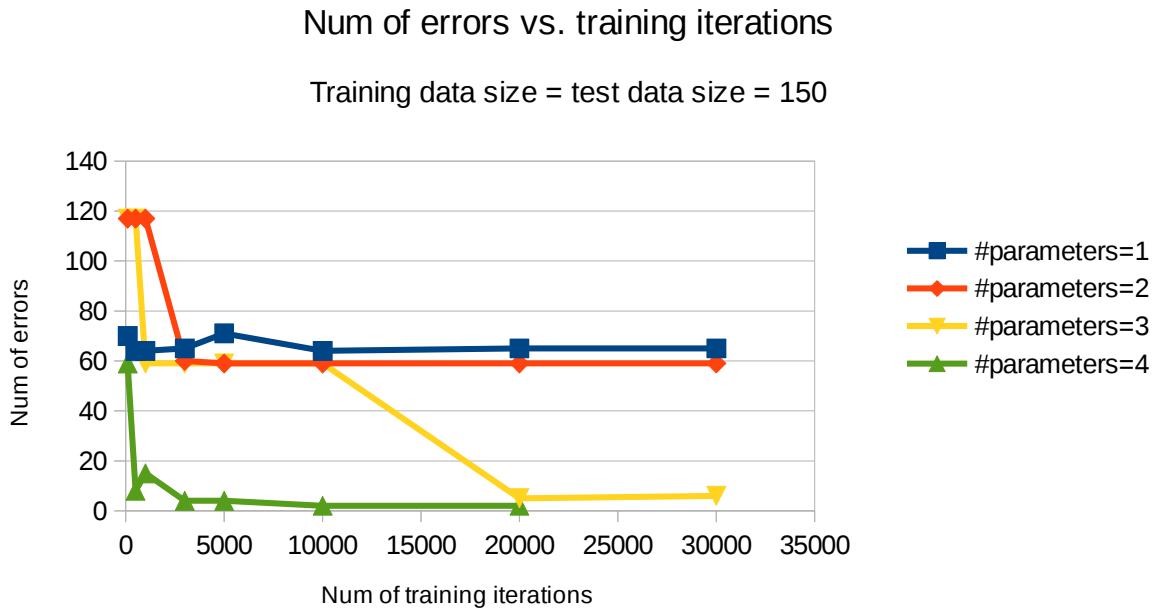
- The saturating values of errors for each of the case is lower when using more number of parameters for training/testing.
- While it is seen that increasing the number of iterations reduce the number of errors in most of the cases, there are exceptions where this is seen as reverse with slight increase in errors. This increase is a natural property of the way training occurs in the network.
- Also, in most of the cases, increasing the training data size leads to a better accuracy. But in a few cases, this is seen as reverse. This is because of the over-fitting due to larger number of training samples which the network is not able to train itself.
- It is also seen that the number of iterations needed to achieve the same error rate increases with the increase in training samples due to larger data size, the iterations for training also increases.
- Since the overall data size is itself very small, probably there are some fluctuations seen in

the results obtained. A more smoother curve would be expected for a larger training data size.









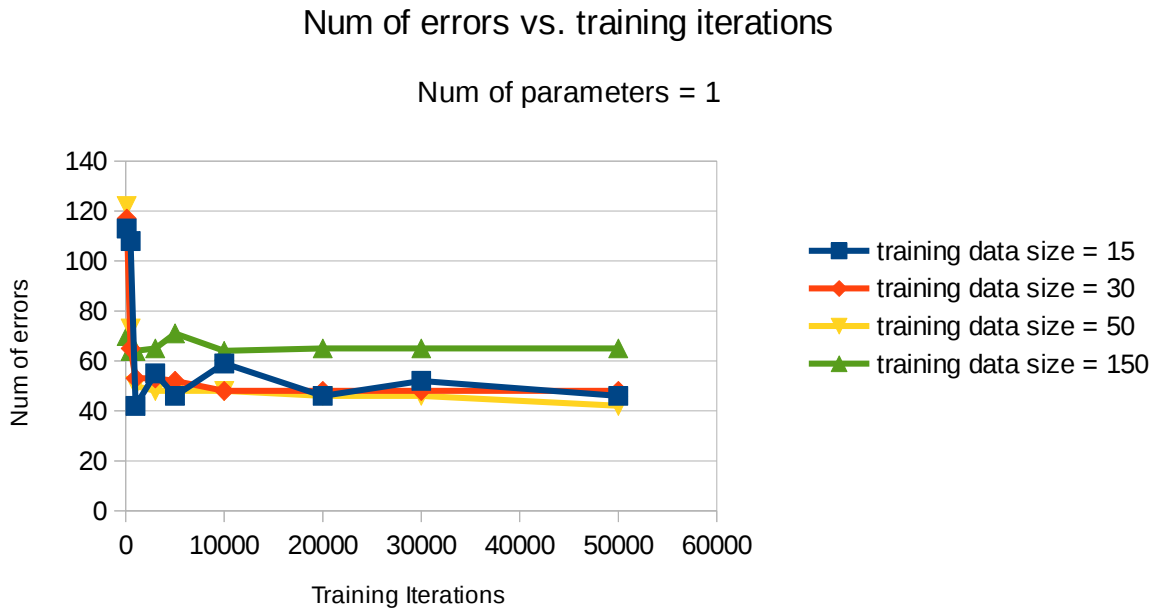
Results and Observations - II:

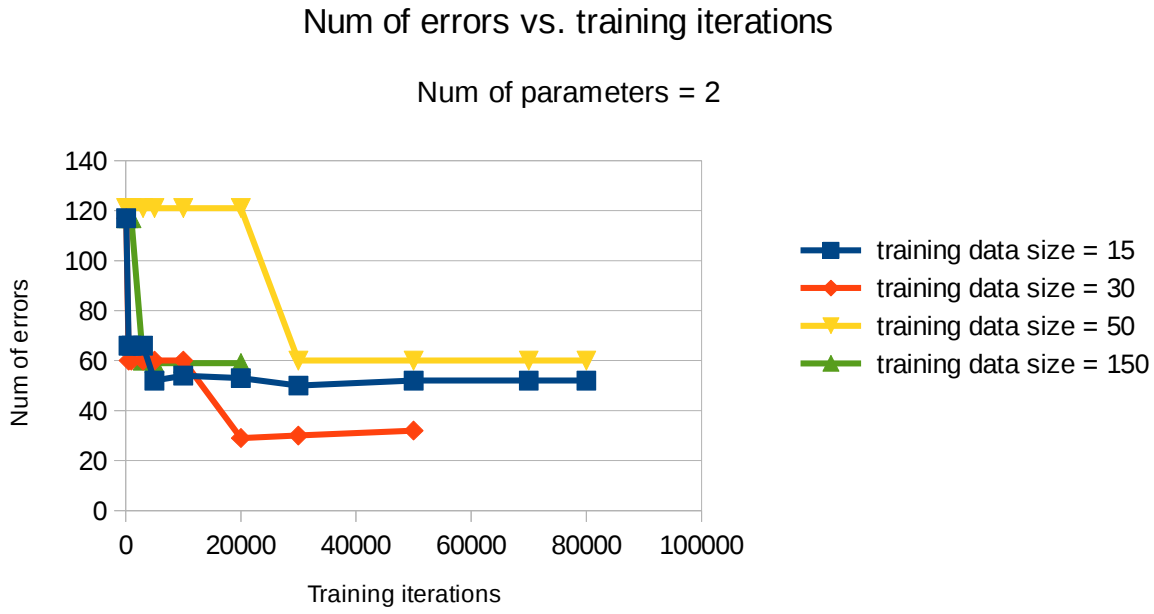
The next set of 4 graphs have the below conditions:

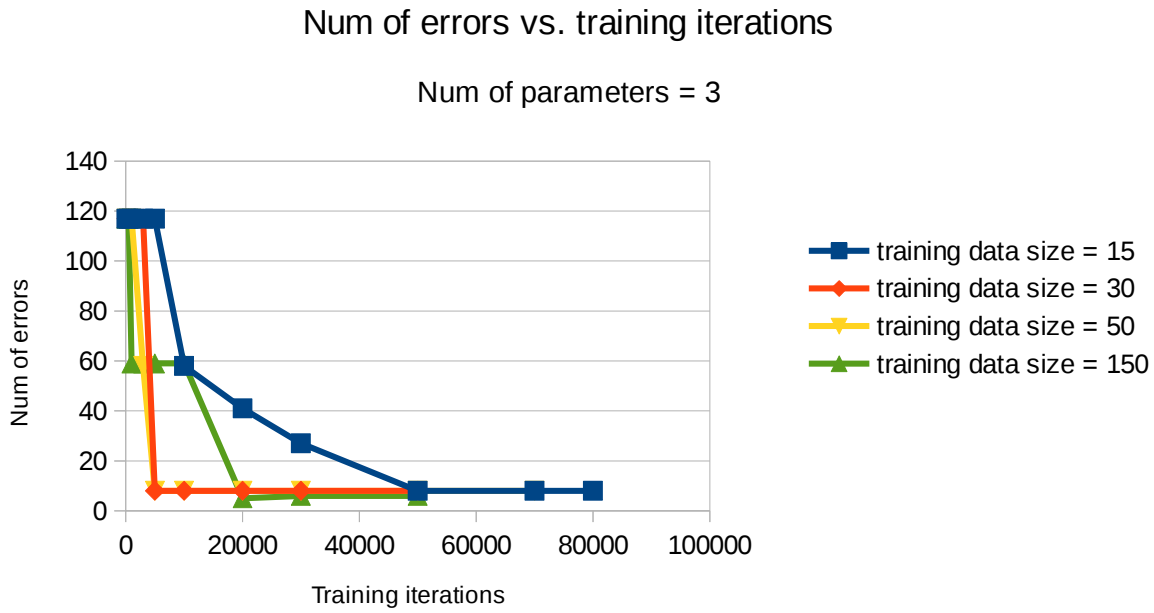
1. Each of them is for a fixed number of parameters of flower being used for training and testing. 4 cases are used: 1, 2, 3, 4 parameters.
2. Each graph consists of 4 plots, each corresponding to how many samples were used for training. There were 4 cases considered: 15, 30, 50 and 150.
3. The test data size is same as 150 in all cases.

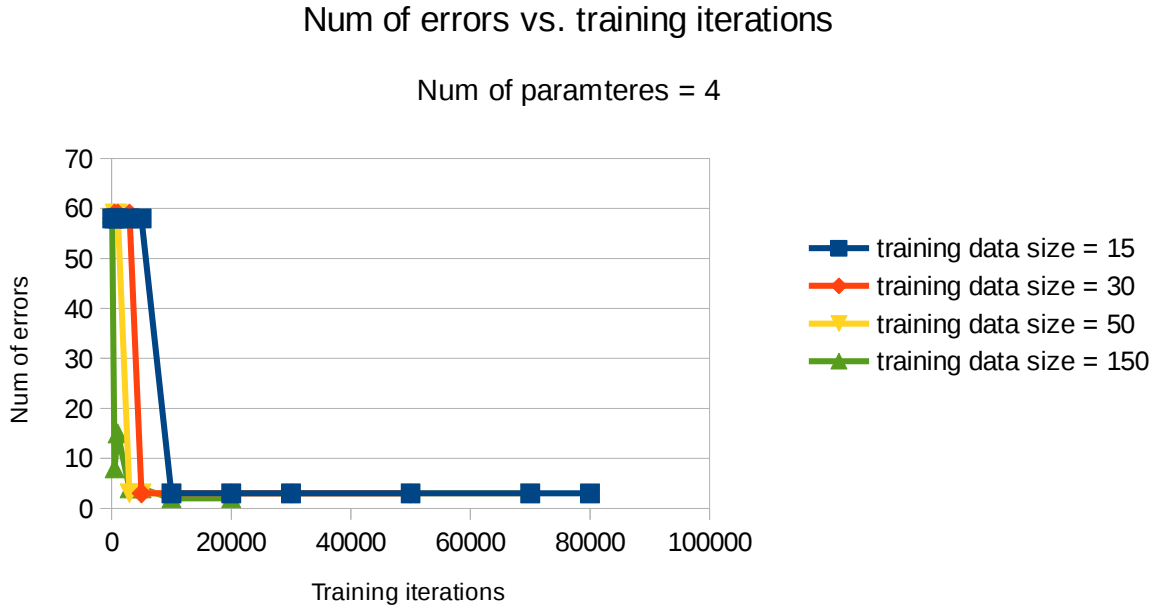
Common Observations/Reasoning:

- Using more samples for training doesn't necessarily result in a better accuracy. As seen in graph with 1 parameter, using 150 samples for training results in a higher errors than others. This is consistently seen for large number of training iterations.
- There are more perturbations in the curve for 1-parameter compared to more.
- The slope of the curve is steeper for larger number of parameters compared to smaller. This indicates that higher number of parameters lead to a faster convergence of the error rate.
- The overall number of errors seen in case of 4 parameters is mostly same for any size of training data but varies for cases with lesser parameters or takes more iteration to converge, if same.
- Since the overall data size is itself very small, probably there are some fluctuations seen in the results obtained. A more smoother curve would be expected for a larger training data size.


Num of errors vs. training iterations
Num of parameters = 1
Num of errors
140
120
100
80
60
40
20
0
0
10000
20000
30000
40000
50000
60000
Training Iterations
training data size = 15
training data size = 30
training data size = 50
training data size = 150


Num of errors vs. training iterations
Num of parameters = 2
Num of errors
140
120
100
80
60
40
20
0
0
20000
40000
60000
80000
100000
Training iterations
training data size = 15
training data size = 30
training data size = 50
training data size = 150





Further Results: The same data set was classified using MLE (Maximum Likelihood Estimation) by assuming Gaussian distribution of the feature values. The result obtained are compared with the same from BPNN as below (training data size = 30, test data size = 120):

Sl. No.	Num. of parameters used	Errors with MLE	Errors with BPNN
1	1	42	43
2	4	4	3