ECE4870/7870 CS 4770/7770 F'18 Computer Assignment 1 Part B Due 10/23/2018

Backpropagation Training of a MLP

This assignment will be given out, and graded, in two separate parts. The idea is that if you can't get the correct first part, it will guarantee wrong results later. So we will have a check point at the beginning.

Assuming that you now have a correctly working version of backpropagation training for a MLP, do the following.

Part B:

1) Using the cross data, continue to train the network until the change in sum-of-squared-errors is less than 0.001. Randomize the presentation order of the samples for each epoch after the first. Plot the sum-of-squared-errors per epoch. Then test your network in the following way. Sample the square containing the first 2 coordinates (roughly [-2.1, 2.1] x [-2.1, 2.1]) by increments of 0.01 (or 0.001 for finer resolution). Set the 3rd coordinate for all such vectors to some very small random number around 0.0. Create a visual display of the classification labels using colors and/or different symbols for all the points in the square to estimate the decision regions formed by the network on the cross dataset. To get a smoother boundary, you can display only the output from one of the output nodes as a continuous variable between 0 and 1, or the difference between out values. You can use a function to plot 0.5 (or 0.0) isocurves What would happen if you just eliminated the 3rd coordinate for all the training data? Does this dimension help in classification?

2) Retrain the network with 3 different choices of learning rate (0.01, 0.2, 0.9) and momentum of 0.3. Plot the sum-of-squared-errors per epoch for the 3 training sessions along with the original on the same graph. Be careful to label everything properly and use different line types/colors/symbols to easily distinguish the curves. How does learning rate affect convergence?

3) With a learning rate of 0.01, retrain with no momentum and momentum equal to 0.6. Again, plot the 2 curves on the same graph, with everything easily labeled.

4) Download the data set Two_Class_FourDGaussians500.txt from Canvas.

Here is its description:

Two 4-D Gaussians with 500 points

Input parameters:

```
R1=[2 0.5 0 0; 0.5 1 0 0; 0 0 3 0; 0 0 0 1];   % covariance matrix1
mu1=[0 1 0 -1]';          % mean1
```

R2=[2 0 -1 1; 0 2 1 0; -1 1 4 0; 1 0 0 1]; % covariance matrix2
mu2=[-3 -2 -1 -1]'; % mean2

Recovered Means and Covariances (for all 500 points)

Mu1: 0.0152 0.994 0.0229 -1.0568

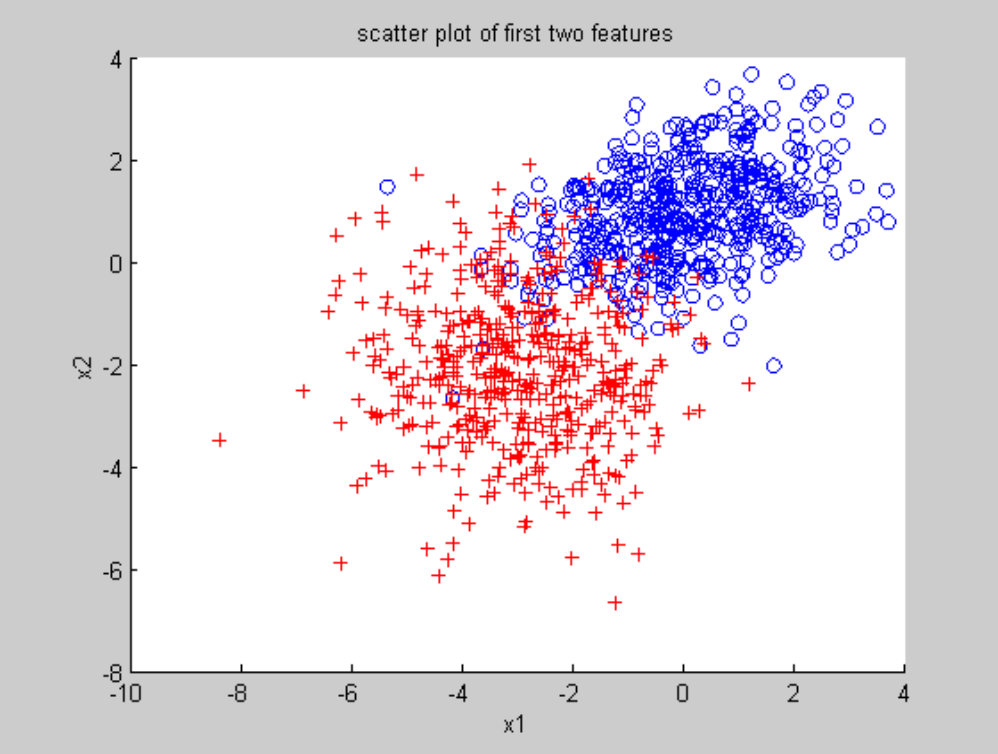
R1:

2.000	0.495	-0.0627	-0.0424
0.495	0.968	0.086	-0.0054
-0.0627	0.0861	3.321	0.0259
-0.0424	-0.0054	0.0259	1.0823

Mu2: -2.973 -2.062 -0.996 -0.963

R2:

2.0715	-0.103	-1.152	1.012
-0.103	2.091	1.093	-0.0382
-1.152	1.0931	4.431	0.0716
1.0122	-0.0382	0.0716	1.0218



Construct a MLP of structure 4:N:2, where N is the number of hidden units. You can pick the learning rate and momentum. Set up and execute a 5-fold cross validation experiment with this data. Recall that this means for each "fold", you leave off the corresponding 20% of each class, train of the remaining 80%, and test the left-out data, recording the classification accuracy. Your output should be "confusion matrices" for each fold, followed by the summary confusion matrix for all 5 folds. In this case, the confusion matrix is a 2 x 2 matrix where the rows represent the true class labels and the columns correspond to the label assigned by the network. An idea confusion matrix for a given fold would have 100 in each of the diagonal positions and 0 off diagonal. The off diagonal entries indicate classification error.

Try this for three values of N, start with N = 8. After the first experiment, you need only show the aggregated confusion matrices. How does N affect the training error and cross validation accuracy?

Your typed report should have a brief introduction to each part, the required graphs or confusion matrices, clearly delineated by what is being tested with each one, and a brief analysis of the results. Did the results match your expectations, i.e., what did you learn?

You need to turn in your modified code with the report.