Perceptron and Neural Networks (50 Points)

In this homework, you will implement the Perceptron algorithm and compare it with WEKA implementation of Neural networks. You will also compare it with either your own or WEKA implementations of Logistic Regression and Naive Bayes.

TASK 1: Download the datasets available on the class webpage or elearning. As in homework 2, the classification task is spam/ham.

40 Points: Implement the perceptron algorithm (use the perceptron training rule and not the gradient descent rule). Your task here is to experiment with different values of number of iterations and the learning rate. Report the accuracy for 20 suitable combinations of number of iterations and the learning rate. Repeat your experiment by filtering out the stop words. Compare the accuracy of your perceptron implementation with that of Naive Bayes and Logistic Regression (implemented in Homework 2). If you did not implement Naive Bayes (NB) and/or Logistic regression (LR) in homework 2 or you are not sure that your implementation is correct, please use the LR and NB implementations available in WEKA.

10 points Neural networks in WEKA.
Download WEKA http://www.cs.waikato.ac.nz/ml/weka/.
Convert the spam/ham dataset into the ARFF format used by WEKA.

Using the Neural networks implementation in WEKA (called MultiLayered Perceptron), report the accuracy on the test set. Experiment with different number of hidden layers and units. Report on how the number of hidden layers and units as well as other options such as momentum, number of iterations, and learning rate affect the accuracy.

K-means clustering on images [50 points]

In this problem, you will use K-means clustering for image compression. We have provided you with two images.

- Display the images after data compression using K-means clustering for different values of K (2, 5, 10, 15, 20).
- What are the compression ratios for different values of K? Note that you have to repeat the experiment multiple times with different initializations and report the average as well as variance in the compression ratio.
- Is there a tradeoff between image quality and degree of compression? What would be a good value of K for each of the two images?

We have provided you java template KMeans.java which implements various image input/output operations. You have to implement the function k-means in the template. See the file for more details. Note that your program must compile and we should be able to replicate your results. Otherwise no credit will be given.

What to turn in:

1. Your code and datasets (in ARFF format)
2. A README for your compiling/using your code
3. A report (pdf or doc file) containing answers to the questions posed.