EECS 440: Machine Learning Fall 2015 Write-up for Project 2, assigned 9/25. Max points: 20.

Yingcheng Sun ID: yxs489

1. Answer the following questions:
   (a) What is the area under ROC of the ANN with no hidden units on each dataset? Set the weight decay coefficient γ=0, and train to convergence. This approximates the perceptron, which uses a step function instead of a sigmoid. How does this compare to the decision stump/tree results in the previous assignment?

   Answer: Below is the data from the 5 experiments, run with no hidden nodes and weight decay coefficient=0. The data sets were run with 5-fold cross validation and all results (except AROC) averaged.

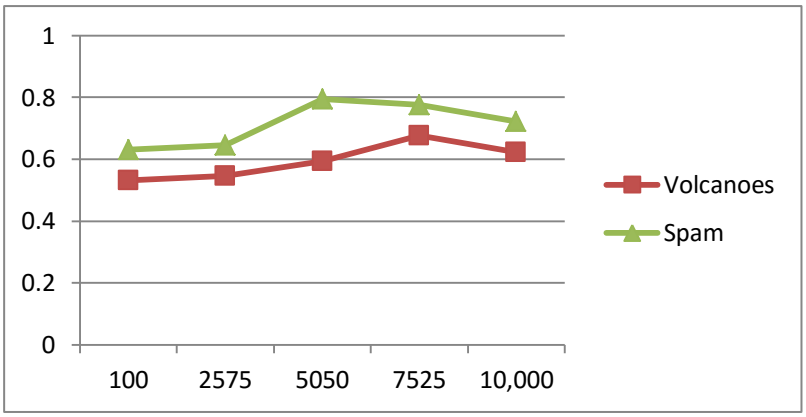
	Voting	Spam	Volcanoes
AROC	0.723	0.660	0.514

   Compared to the decision stump analysis on the same datasets, the ANN was more accurate for almost all the datasets. For the ye data, the decision stump was more accurate. For the ab data, the stumps were considerably more accurate.

   (b) For volcanoes and spam, and explore how the AROC changes as learning iterations are increased. Fix the number of hidden units to 5 and γ=0.01 for these experiments. Plot AROC results for at least three values of learning iterations evenly spaced between 100 and 10,000. Compare your results to the "perceptron" in part (a). Does the introduction of hidden units lead to improved accuracy? How many iterations does this ANN need to converge compared to the "perceptron"?

   Answer: the values of AROC of Volcanoes and Spam with different iterations are listed below.

Iterations	Volcanoes	Spam
100	0.531	0.691
2575	0.546	0.646
5050	0.594	0.794
7525	0.677	0.777
10,000	0.623	0.723



The values of accuracy of Volcanoes and Spam with different iterations are listed below.

Iterations	Volcanoes	Spam
100	0.633	0.625
2575	0.701	1.000
5050	0.738	0.701
7525	0.700	0.599
10,000	0.677	0.741

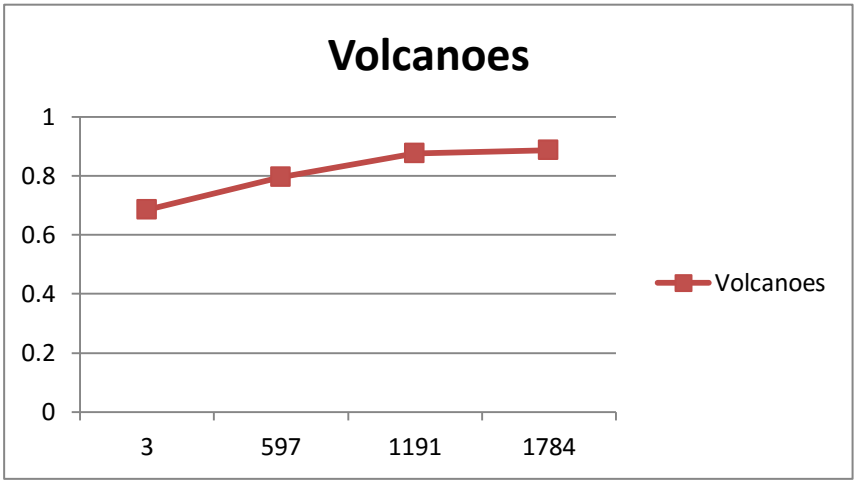
Compared to the perceptron, the aroc of Volcanoes and Spam were increased, even just training for 100 iterations. However it took much longer to converge to a steady answer. Additionally, the aroc started to decrease as the iterations went up, but very slowly. This was accompanied by a fall in accuracy, indicating that overfitting was starting to occur. This ANN need about 6000 iterations to converge.

(c) Explore how the AROC changes as the number of hidden units are varied on volcanoes and voting. Plot AROC results for at least three values of hidden units evenly spaced between 3 and f, where f is the number of input units. Set γ=0.01 and the learning iterations to 100f , with a minimum of 10,000. Compare the ROC and training times to the results in parts (a) and (b). Warning: This experiment may take a long time to run

Answer: the values of AROC of Volcanoes with different hidden units and iterations are listed below.

f: 1784 learning iterations=100f= 178,400

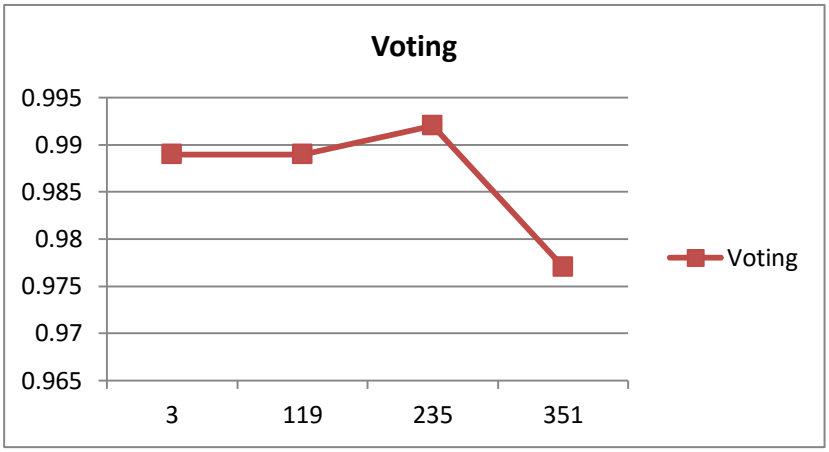
No. of hidden units	Volcanoes
3	0.686
597	0.795
1191	0.875
1784	0.886



The values of AROC of Voting with different hidden units and iterations are listed below.
f: 351 learning iterations=100f=35,100

No. of hidden units	Voting
3	0.989
119	0.989
235	0.992
351	0.977

The corresponded plot is:



The training times for each experiment varied, even between folds. For the most part these results agree well with expectations. The AROC is much higher than the values found for the perceptron, indicating a better fit. However, increasing the number of nodes to a large number does not seem to have a strong impact on over fitting. Maybe there is some noise in the data.

2. Write down any further insights or observations you made while implementing and running the algorithms, such as time and memory requirements, the complexity of the code, etc. You can also

test your code with classification problems from your research.

Answer:

(a) The results from all of these experiments seem to indicate that averaging model results over a n-fold cross validation, when combined with the weight decay modification for training, provides a reasonable defense against overfitting.

(b) Stochastic gradient descent is obviously faster than regular gradient descent to do get convergence in this project. It takes about ten times more time to use regular gradient descent method than stochastic gradient descent method.

(c) Besides doing five-fold stratified cross validation to preprocess data, I shuffled the data in each fold, and it showed better results than not shuffled. I also standardized data before training, it also showed better performance than not standardized.