Homework 3: Neural Networks

CS412

Released: February 27th

Due: March 12th, 11:30pm on Gradescope

1 Neural Network with Backpropagation

The primary portion of this assignment will be coding a neural network that finds a solution using gradient descent and backpropagation. A .py file with class definition and some helpful functions stubs have been made available on piazza. **You will need to submit this as a .py file. No jupyter notebooks for this portion and scikit-learn will not be allowed**.

I have provided some basic structure, and will provide a video walk through of the 1 hidden layer 2-node hard coded perceptron. **Do not change any function headers** Additionally, your code will need to allow three additional pieces of functionality:

a) η (eta): Your function should perform gradient descent relative to a learning rate

b) MaxIterations: Your neural network should halt after it goes through the given number of iterations

c) numLayers,numNeurons: numLayers should be the number of hidden layers and numNeurons is the number of hidden neurons per layer. You may assume that numLayers and numNeurons will be at least one and that the total shape of the hidden nodes will be rectangular, i.e. all layers have the same number of nodes.

This code will be automatically tested by gradescope and you will get access to results and feedback from the code each time you submit it. There is no limit to the number of submission.

2 Neural Network with sklearn

In this section, run package neural networks on the digits data and on the new dataset. **HERE** is the python documentation for the neural network classifier. For all models tested, you should use `activation = "relu", epsilon=0.001,numIter=10000,alpha=0` and `solver = "adam"`. All other inputs can be default. For our exercise, all hidden layers will have the same number of neurons.

For the blood transfusion data, give the cross validation errors for the following. Define the number of number of hidden layers to be in $\{1, 2, 5\}$ and the number of nodes per layer to be in $\{2, 5, 10\}$. Give the 10-fold cross validation error for each of these. A table is sufficient here. Report the number of hidden layers and nodes per layer which provides the highest accuracy.

For the digits dataset, use your HW1 features and try to build a neural network. Define the number of hidden layers to be in $\{1, 2, 5, 10\}$ and the number of nodes per layer to be $\{2, 5, 10, 50, 100\}$. For each of these 20 models, report the 10-fold cross validation error and the runtime in milliseconds.
Short answer

a) How does runtime scale with the number of layers and the number of nodes per layer? Do each have a similar effect?

b) What is the number of layers and nodes per layer that gives an optimum result?

c) For your optimum model, try increasing and decreasing the learning rate from the default (0.001). Discuss the tradeoff here between runtime and accuracy?

d) Is the neural network finding the same solution everytime? Why or why not? Does this have an impact on the expected fit?

e) **Graduate Student Question**: Experiment with early stopping on neural networks that have a large number of internal nodes. Does this cause the model to under or over fit the data? Why?

Draw the 2D region for your optimal neural network. **Label this Figure 3.1**

Extra Credit

On the test set of your choice. Experiment with neural networks and bagging. Since bagging is a variance reduction strategy, you should make sure to compile all of the variances when you collect your cross-validation data. It may also be worthwhile to increase the number of folds for your cross-validation sets.

Questions to answer.
Does the optimum number of bagged neural networks change with the complexity of those networks?
Do the more complex models provide better solutions when applied in a bagging ensemble model?
What about neural networks makes them react well to bagging?
Would you choose a bagging model as your ”optimum” neural network model?

Making your report

You will need to submit your neuralnetwork.py file to the appropriate gradescope submission. As well, you will need to submit a pdf of your report from part 2 (which can be a jupyter notebook pdf). If you are not using jupyter notebook, please submit a zip of your code.