CPSC 340 Assignment 6 (due Friday April 7 at 11:59pm)

Neural Networks

Instructions

Rubric: {mechanics:3}

The above points are allocated for following the general homework instructions on GitHub.

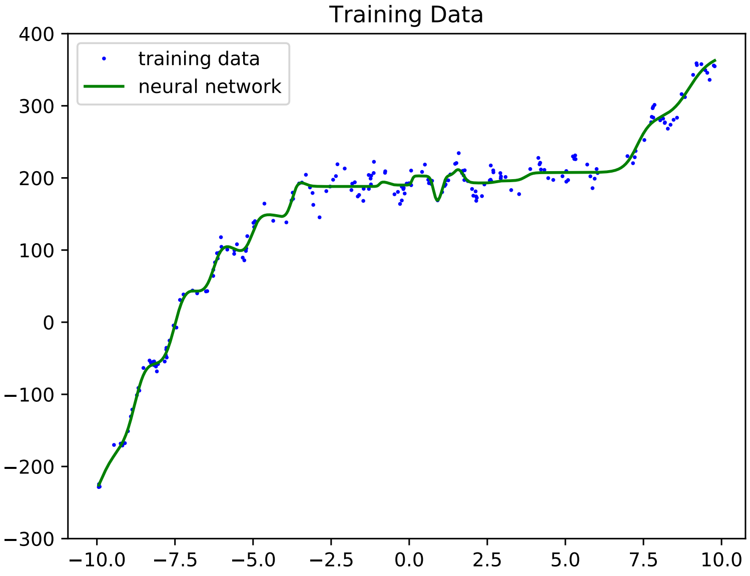
Note: this assignment is a bit more open-ended than the previous ones. There aren't really "correct answers". Try to have fun with it and learn something along the way. We'll mark generously... no need to go overboard!

1 Neural Networks

Rubric: {reasoning:5}

If you run `python main.py -q 1` it will train a neural network on the *basisData.pkl* data from Assignment 3. However, in its current form it uses no hidden layers and therefore is just performing linear regression. In fact, if you take a look at the figure that is generated, you should see that it matches the figure from Assignment 3 Question 2.1. Try to improve the performance of the neural network by adding hidden layers and tuning the other hyperparemeters. Hand in your plot after changing the code to have better performance, and list the changes you made. Write a couple sentences explaining why you think your changes improved the performance. When appropriate, refer to concepts from the course like overfitting or optimization.

To see the list of hyperparameters and their definitions, see the scikit-learn MLPRegressor documentation: `http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html`.
Note: "MLP" stands for Multi-Layer Perceptron, which is another name for artificial neural network.
Also: we didn't discuss L-BFGS in class, but it may help to set `solver` to `lbfgs` since the data set is tiny.



I changed the Regressor to 100 neurons in the 1st hidden layer, and added an alpha of 0.01 to regularize the error, and optimize the hyper-parameters, which in turn reduces overfitting. Came to the number 100 after trying various numbers between 10-1000 and tried adding layers as well, but came to the conclusion only one hidden layer was present.

2 Your Own Data

Rubric: {reasoning:5}

Try a neural network on the data set of your choosing. You can use a data set that's built into scikit-learn (`http://scikit-learn.org/stable/datasets/`), or you can use a data set from a previous homework assignment, or something from an area that interests you. If you are doing classification rather than regression, use scikit-learn's MLPClassifier rather than the MLPRegressor that we've provided in the assignment code: `http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`

Try fiddling with the hyperparameters. Write a few sentences about what you tried, what train/validation error you achieved, and whether you think neural networks are a good model for your dataset. Optionally (for glory rather than marks), install Keras/TensorFlow and compare it with the scikit-learn neural net.

For the Iris dataset in scikit, on using the MLPClassifier with no hidden layers gives Training error = 0.0 and Validation error = 0.6, while increasing the number of layers doesn't really decrease the validation error, on repeated runs, with and without hidden layers give very similar errors.