Project Assignment 1

CZ4042: Neural Networks

Deadline: 7th October, 2016

- ✓ The project is to be done individually.
- ✓ Complete both parts 1 and 2. Data files for both parts can be found under Assignments.
- ✓ Submit one project report for both parts including sections on
  - Introduction
  - Methods describing the architectures and learning algorithms of the neural networks implemented
  - Results including parameters used for training, plots of convergence of learning, training and test error rates, supporting tables and graphs
  - Discussion on the results and challenges

  in a PDF file named: your_name_P1_report.pdf

- ✓ Submit all the source codes in a zip file named: your_name_P1_codes.zip
- ✓ Grading of the project is based on execution (60%), report (30%), and bonus (10%).
- ✓ Submit both your report and source code on line via NTU Learn before the deadline.
- ✓ TA Mr. Sukrit Gupta (SUKRIT001@e.ntu.edu.sg) is in charge of the course projects. Please see him at the Biomedical Informatics Graduate Lab (NS4-04-33) during his office hours: Friday 3:30 P.M. – 5:30 P.M., in case you face issues.

Part 1: Classification Problem

This part of the assignment is to give you some exposure to the use of neural networks for classification problems.

1. Download the Spam database, 'spambase.data'. This database contains attributes obtained from spam emails and normal emails together with their corresponding label: "spam" or "not spam", which are coded as 1 and 0.

2. Initially, divide the data into test and train set. Use various strategies that are available for dividing the data set, i.e. dividing the data randomly, block wise and in an interleaved fashion. Do not use the data in the test dataset during training. It is reserved for the final performance measure. Think of it as unseen data during all of your work. Use all the remaining data for testing.

3. Take your training set and normalize real-valued attribute to zero mean and unit standard deviation. Use the mean and standard deviation derived from the training dataset to preprocess the test dataset. Note that you are not allowed to calculate these parameters from the testing dataset. Do not recalculate them from the test dataset. This would be completely wrong.

4. Next, create and train a neural network to identify spam on email automatically, based on the attributes obtained from the emails.
   a. Experiment with the following different neural network configurations:
      - Learning rate,
      - Different number of hidden layers,
      - Different number of neurons in each layer,
      - Different stopping criteria for your algorithm.
   b. Repeat previous step for different sets of three-way data split (0.7:0.15:0.15): different datasets for learning, for validation and for testing.
   c. Finally, identify a suitable ANN architecture and training parameters, justify your choice.

Hint: Refer to sample codes: spam_preprocess.m, 'spam_train.m'. Please refer to the Neural Network Toolbox on Matlab for this exercise. You may also use python.

Part 2: Approximation Problem

This assignment aims to provide you with some exposure to the use of neural networks for regression/approximation problems.

1. Download the California Housing database, 'california_housing.data'. This database contains attributes of housing complexes in California such as location, dimensions, etc, together with their corresponding price.

2. Initially, divide the data into test and train set. Use various strategies that are available for dividing the data set, i.e. dividing the data randomly, block wise and in an interleaved fashion. Do not use the data in the test dataset during training. It is reserved for the final performance measure. Think of it as unseen data during all of your work. Use all the remaining data for testing.

3. Preprocess your data as in previous part. An example Matlab script for training a feed-forward neural net with the California data provided:, 'california_preprocess.m' script and california_train.m

4. Next, create and train a neural network to predict the prices of the houses automatically, based on the attributes obtained from the housing data.

a. Using the three-way data splits method, experiment with the following different neural network configurations:
   - Learning rate,
   - Hidden layers,
   - Different neuron size,
   - Different stopping criteria for your algorithm,

b. Repeat Step I for different sets of three-way data split (0.7:0.15:0.15), i.e., different learning, validation and testing datasets.

c. Identify a suitable ANN architecture and training parameters, justify your choice.

5. (Bonus) Next, train a Radial Basis Function (RBF) neural network to predict the house price automatically. Measure the performance of your trained network on the test dataset and discuss how it compares to the results found in point 4.