CSCI 374 — Machine Learning and Data Mining
Oberlin College — Fall 2016

Homework #3: Neural Networks

Important Dates

Assigned: November 14

Final Due Date: November 28 (11:59 PM)

Assignment

In this assignment, you will practice:

1) working with APIs for existing machine learning libraries
2) experimenting with various parameter settings for neural networks,
3) analyzing the results of those experiments to evaluate the performance of the different learning algorithms with respect to different data sets, and
4) writing a technical report detailing (i) how your implementation works, (ii) your experimental setup, (iii) the results of your experiments, and (iv) any implications or lessons learned from your implementation and results.

In particular, you will conduct experiments analyzing the behavior of neural networks using an existing machine learning library of your choice (e.g., Weka in Java, scikit-learn or Tensorflow in Python). Through reusing existing implementations, you will gain practice with standard libraries you would be likely to use outside class in the real-world, as well as gain practice possibly useful for your final projects. By experimenting with changing the parameters of neural networks, you should also gain a better understanding of how these parameters affect the behavior of neural networks and how to find appropriate settings for a given data set.

Acceptable Programming Languages

You can use either the **Java** or **Python** programming languages to complete this assignment.

Program

In this assignment, your goal is to train and test neural networks on two of the data sets considered in Homeworks 1 and 2, then compare the results from Naïve Bayes and a decision tree algorithm of your choice (e.g., ID3, C4.5, CART).

Note: you are free to use either your implementations of Naïve Bayes and decision trees or from an existing machine learning library (including the library you choose for your neural network implementation).

Data Sets

1) **monks1.csv**: A data set describing two classes of robots using all nominal attributes and a binary label. This data set has a simple rule set for determining the label: `if head_shape = body_shape ∨ jacket_color = red, then` *`yes`*`, else` *`no`*. This data set is useful for debugging your implementations and verifying their correctness. Monks1 was one of the first machine learning challenge problems (http://www.mli.gmu.edu/papers/91-95/91-28.pdf). This data set comes from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/MONK%27s+Problems
2) **opticalDigit.csv**: A data set of optical character recognition of numeric digits from processed pixel data. Each instance represents a different 32x32 pixel image of a handwritten numeric digit (from 0 through 9). Each image was partitioned into 64 4x4 pixel segments and the number of pixels with non-background color were counted in each segment. These 64 counts (ranging from 0-16) are the 64 attributes in the data set, and the label is the number from 0-9 that is represented by the image. This data set is more complex than the Monks1 data set, but still contains only nominal attributes and a nominal label. This data set comes from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

The two data sets can still be downloaded from the "Course Content/Homework 1" folder on Blackboard. The file format for each of these data sets is described in the Homework 1 Assignment, in case you need to refer back to it.

Program Behavior

Your program should behave as follows, similar to your program for Homeworks 1 and 2:

1) Your program should be named neuralnets
2) It should take as input five parameters:
   a. The path to a file containing a data set (e.g., monks1.csv or monks1.arff)
   b. A random seed as an integer
   c. The proportion of data to use for training
   d. The maximum number of training iterations
   e. The number of hidden neurons to use
3) Next, the program should read in the data set as a set of instances (you are free to use existing libraries to represent the instances, and do not have to reuse your implementation from Homeworks 1 and 2)
4) The instances should be split into training and test sets (using the random seed input to the program)
5) The training set should be used to create a neural network to learn the weights in the network
6) The learned model should be evaluated using the test set created in Step 3.

7) The confusion matrix counted during Step 5 should be output as a file with its name following the pattern: results_<DataSet>_Neural_<TrainingProportion>train_<MaxIterations>iters_<NumNeurons>neurons_<Seed>.csv (e.g., results_monks1_Neural_0.6train_1000iters_10neurons_12345.csv).

The file format for your output file should be exactly the same as in Homework 1 and 2. Please refer back to the Homework 1 Assignment for details, if necessary.

Experiment #1: Algorithm Comparison

The goal of the first experiment is to investigate how different types of supervised learning algorithms perform on different data sets with different properties. That is, our goal is to compare Neural Networks with Bayesian Learning (Naïve Bayes) and Decision Trees (ID3, C4.5, or CART).

For Experiment #1, pick 30 random seeds (include them in your report), then calculate the average accuracy of Naïve Bayes, ID3, and C4.5 on the two data sets across the 30 runs (one run per seed). Afterwards, compare the average accuracies between the algorithms to evaluate their performances on the different data sets. In particular, evaluate:

1) How do the three approaches perform on Monks 1? What do you think might have caused any differences or similarities in their performances?
2) How do the three approaches perform on this data set on Optical Digit? What do you think might have caused any differences or similarities in their performances?
3) What inferences can we draw about the three types of algorithms from these two data sets?

For your results analysis in Experiment #1, use the same confidence interval calculations over the 30 seeds that you used in Homework #2 (instructions are still available on the Homework #2 Assignment on the course webpage).

Experiment #2: Fitting the Training Data

The goal of the second experiment is to investigate how neural networks behave with different amounts of training. In particular, you will vary both (1) the amount of training data to use for training (by selecting a proportion of the data set to use for training), as well as (2) the number of training iterations (i.e., the number of times the algorithm feeds the training set into the algorithm for learning). For this experiment, you can use either the Optical Digits data set, or a data set of your choosing (for example, you can choose one from the UCI Machine Learning Repository).

First, you should pick four values for the training data proportion (e.g., 0.5, 0.7, 0.8, 0.9), then compare the performance on the testing set for each proportion (holding the number of training iterations fixed, e.g., 1,000). Using a line curve, plot the average accuracy (over at least 10 runs) of the neural network on the test set as the training set proportion increased.

Second, you should pick a larger value for the number of training iterations (e.g., 2,000, 5,000, or 10,000) and hold the proportion of training data fixed (e.g., 0.7). Using a line curve, plot the average accuracy (over at least 10 runs) of the neural network on both the training and test sets as the number of iterations increases. For example, if you choose 2,000 maximum iterations, record the accuracy of the network on the training and test sets after 200, 400, etc. iterations to build your line curve.

For both parts of Experiment #2, you should pick at least 10 random seeds (include them in your report) to create your results. From these runs, you should evaluate:

1) How did the neural network perform as the amount of training data increased?
2) How did the neural network perform as the number of training iterations increased?
3) Did you find evidence of overfitting, where increased training (in the amount of instances or the number of iterations) caused the algorithm to do worse on the test set?

You do not need to calculate confidence intervals for Experiment #2.

Experiment #3: Choosing a Topology

The goal of the third and final experiment is to investigate how neural networks behave with different topologies. In particular, you will vary the number of neurons in the hidden layer to see how increasing the complexity of the network affects neural network performance. That is, you should choose four values for the number of hidden neurons parameter (e.g., 5, 10, 20, 50), then compare the performance on the testing set for each value. Using a line curve, plot the average accuracy (over at least 10 runs) of the neural network on the test set as the number of hidden neurons increased. For this experiment, you can use either the Optical Digits data set, or a data set of your choosing (for example, you can choose one from the UCI Machine Learning Repository).

For Experiment #3, you should pick at least 10 random seeds (include them in your report) to create your results. From these runs, you should evaluate:

1) How did the neural network perform as the number of hidden neurons increased?

You do not need to calculate confidence intervals for Experiment #3.

Final Handin

Before the assignment due date (Monday November 28 at 11:59 PM), you should turn in:

1) A ZIP file (named as your OCCS username) containing:
   a. Your source code
   b. A README file
2) Your technical report as a PDF file, named the same as your ZIP file.

Your README file should describe the different source code files used by your program, as well as instructions for compiling your program, running your program, and finding its output file(s).

Your technical report should contain:

- An introduction describing the assignment and the contents of the report (provide the reader with the background needed to understand the rest of the report)
- A description of your implementation for the assignment (what did you create?)
- A description of your experimental setups for all three experiments (what did you run and for what purpose?)
- A discussion of the results from all three experiments (what did you find, why did you find that, and what are the implications?)
- A conclusion summarizing the report and assignment
- An estimate of the total time spent on this assignment

Grading

The homework will be graded as follows:

- Implementation Correctness and Documentation: 25%
- Report: 75%

Honor Code

Each student is to complete this assignment individually or with a single partner. Since the assignment is a mini-project in scope, students are encouraged to collaborate with one another to discuss the abstract design and processes of their implementations. For example, please feel free to discuss the pseudocode for each learning algorithm to help each other work through issues understanding exactly how the learning algorithms work. You might also want to discuss the processes used to generate the training and test sets from the read in data set. Or, you might need to discuss how to work with the input and output files.

At the same, since this is an individual or small group assignment, no code should be shared between small groups. Any violation of the above will be considered an Honor Code violation. However, you are permitted to discuss how the APIs for the machine learning library you chose works, and you can share online tutorials describing how to use the APIs.

If you have any questions about what is permissible and what is not, please discuss with the professor. Please also feel free to stop by office hours to discuss the homework assignment if you have any questions or concerns.