Description:

Implementing the backpropagation algorithm for Neural Networks and testing it on various real world datasets.

1. Pre-processing:

Pre-processing involves checking the dataset for null or missing values, cleansing the dataset of any wrong values, standardizing the features and converting any nominal (or categorical) variables to numerical form. This step is essential before running neural net algorithm, as they can only accept numeric data and work best with scaled data.

The arguments to this part will be:

- complete input path of the raw dataset
- complete output path of the pre-processed dataset

The pre-processing code will read in a dataset specified using the first command line argument and first check for any null or missing values. Any data points (i.e. rows) that have missing or incomplete features will be removed.

Then, the following was performed for each of the features (independent variables):

- If the value is numeric, it needs to be standardized, which means subtracting the mean from each of the values and dividing by the standard deviation.

See here for more details: https://en.wikipedia.org/wiki/Feature_scaling#Standardization

- If the value is categorical or nominal, it needs to be converted to numerical values.

For example, if the attribute is gender and is encoded as "male" or "female", it needs to be converted to 0 or 1. You are free to figure out specifics of your encoding strategy, but be sure to mention it in the report.

2. Training a Neural Net:

The processed dataset will be used to build a neural net. The input parameters to the neural net are as follows:

- input dataset – complete path of the post-processed input dataset
- training percent – percentage of the dataset to be used for training
- maximum_iterations – Maximum number of iterations that your algorithm will run. This parameter is used so that your program terminates in a reasonable time.
- number of hidden layers
- number of neurons in each hidden layer

For example, input parameters could be:

ds1 80 200 2 4 2

The above would imply that the dataset is ds1, the percent of the dataset to be used for training is 80%, the maximum number of iterations is 200, and there are 2 hidden layers with (4, 2) neurons. The program will initialize the weights randomly.

The following assumptions were made while coding the Neural Net:

- the activation function will be sigmoid
- The backpropagation algorithm was used
- the training data will be randomly sampled from the dataset. The remaining will form the test dataset
- The mean square error was used as the error metric
- one iteration involves a forward and backward pass of the back propagation algorithm

- The algorithm will terminate when either the error becomes 0 or the max number of iterations is reached.

After building the model, the output of the model parameters is as below:

Layer 0 (Input Layer):
Neuron1 weights:
Neuron 2 weights:
..
Layer 1 (1st hidden layer):
Neuron1 weights:
Neuron 2 weights:
..
Layer n (Last hidden layer):
Neuron1 weights:
Neuron 2 weights:
..
Total training error = ….
The model is also applied on the test data and report the test error:
Total test error = ….

Testing your program

The pre-processing part and then the model creation and evaluation will be tested on the following datasets:

1. Car Evaluation Dataset
https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
2. Iris dataset
https://archive.ics.uci.edu/ml/datasets/Iris
3. Adult Census Income dataset
https://archive.ics.uci.edu/ml/datasets/Census+Income