JMLR: Workshop and Conference Proceedings 1:1–4, 2016 Submitted 11/20/2016; Published 11/20/2016

Introduction to Machine Learning - EN 605.449.81 - Lab 6

Max Henry MHENRY22@JHU.EDU

Abstract

In this lab, the radial basis function neural network and a feed-forward neural network are implemented and compared using five classification datasets. Using a random 10% subset of the data for the centers of the Gaussian centers, the radial basis network is outperformed by the feed-forward network on each of the five datasets.

Keywords: radial basis function neural network, feed-forward neural network, backpropagation

1. Problem Statement

In this lab, the radial basis function neural network and feed-forward neural network are implemented and compared on multiple datasets. The centers of the radial basis network are chosen from a random 10% of the training data. To learn the hidden layer and output layer weights of the feed-forward neural network backpropagation is used. I hypothesize the feed-forward network with two hidden layers will perform best of every dataset.

2. Algorithms

2.1. Radial Basis Network Neural Network

The radial basis function network is a simple three layer neural network. The first layer is the feature layer, one node for each feature. The second layer is comprised of n nodes whose outputs are the outputs of radial basis functions and n is the number of training examples (in most cases some k less than n is used instead), called hidden nodes. The implementer has a choice of using different radial basis functions for each hidden node. In addition, each radial basis function needs to define a center to anchor the function and a spread parameter, which defines how wide a particular basis functions influence is. In these experiments, a Gaussian radial basis function is used for each node but the spreads are allowed to vary. The spreads were determined by testing values between -1 to 1 in 0.1 increments. To determine the center of these basis functions, a random subset of 10% of the training data is used. Finally, the last layer of the network is the output layer. The output of the layer is determined by the dot product of a vector of weights, one for each hidden node, by the vector outputs from the hidden nodes. To learn these weights, batch update gradient descent is used. An error function of squared error is used in these experiments. In the case of classification, an activation function of $\frac{1}{1+e^{-u}}$ is used where u is the dot product of the weight vector and the hidden node outputs.

2.2. Feed-Forward Neural Network Trained with Backpropagation

A feed-forward neural network is a network made of neurons organized into three groups: an input layer, zero or more hidden layers, and an output layer. The connections between the neurons cannot found a cycle, so information only progresses from the input layer ot the output layer in a linear fashion, hence feed-forward. The input layer in practice is just the data instance that is fed into the network. Each hidden node in each hidden layer is completely connected to all the nodes in the next layer of the network. To determine a prediction for a data instance, the data instance is directly fed into each node of the first hidden layer. The output of these hidden nodes is the dot product between the data instance and the weights connecting the first hidden layer to the next layer after it is passed through an activation function. The activation function used in these experiments is the logistic function. In the case of multiclass classification, the output layer consists of one node per class and the class with the highest output is chosen as the prediction.

To train the weights of the feed-forward network, backpropagation is used. During backpropagation, the gradient of sum of squared errors at the output layer is fed back through the network and is used to update the weights of each of the connections at each layer. Since it uses the chain rule to compute the gradients for each layer, backpropagation requires the network to use an activation function that is differentiable.

3. Experimental Approach

3.1. Data Cleaning

Due to the unprocessed nature of the data, each continuous-valued dataset was normalized between 0 and 1. In order to be fair to both algorithms, discretized attributes were translated with a one-hot encoder. In addition, any missing values were replaced with a random number draw from the distribution of instances with the same class label.

3.2. Experiments

Each algorithm was run against each of the five datasets using 5-fold cross-validation. The success rate, calculated as the number of correctly labeled instances out of the total number of instances in the test set, was averaged over every fold. This number was used to compare the algorithms.

Three variants were tested for the feed-forward network: 1) without a hidden layer 2) with one hidden layer and 3) with two hidden layers. To tune the number of hidden nodes per layer, 5-fold cross validation was used. The best performing number of nodes in the single hidden layer case were used to tune the two hidden layer case.

4. Results

Table 1: Accuracy Rates

	Radial Basis Network	Feed-Forward		
		No Hidden Layer	One Hidden Layer	Two Hidden Layer
Cancer	65.47%	**95.68%**	94.68%	88.63%
Glass	34.76%	55.24%	**60.47%**	28.09%
Iris	45.33%	90.67%	**93.33%**	82.00%
Soybean	35.56%	17.78%	17.78%	**37.78%**
Vote	51.49%	51.26%	**54.94%**	42.99%

5. Conclusions

The radial basis network was outperformed on each dataset by the feed-forward network. It offered comparable performance for soybean and voting datasets but its performance on the other datasets were, on average, 34.64% off the best performance. I claim the variability in the radial basis function performance is due to the variability of the quality of the random centers used for the Gaussian basis functions. To limit this variability, these experiments could be rerun with k-means clustering to determine the Gaussian centers.

The feed-forward network with one hidden layer performed best on three of the five datasets, partially disproving my hypothesis. The fact that the single hidden layer feed-forward network outperformed the network without a hidden layer suggests that glass, iris and vote datasets have some degree of non-linearity. The fact that the feed-forward with two hidden layers underperforms on most datasets suggests that their isn't enough non-linearity in most of the datasets to justify that amount of hidden layers. This may be due to the vanishing gradient problem.

6. Summary

In these experiments, results show that a feed-forward single hidden layer network performs best on the datasets tested. Further work includes using k-means clustering to better assign Gaussian centers in the radial basis network.

7. References

[1] UCI machine learning repository: Breast cancer Wisconsin (diagnostic) data set. (1995, November 1). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wi

[2] UCI machine learning repository: Iris Data set. (1988, July 1). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Iris

[3] Index of /ml/machine-learning-databases/soybean. Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/machine-learning-databases/soybean/

[4] UCI machine learning repository: Congressional voting records data set. (1987, April 27). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

[5] UCI machine learning repository: Glass identification data set. (1987, September 1). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Glass+Identification

[6] Hecht-Nielsen, Robert. "Theory of the backpropagation neural network." Neural Networks, 1989. IJCNN., International Joint Conference on. IEEE, 1989.

[7] Park, Jooyoung, and Irwin W. Sandberg. "Universal approximation using radial-basis-function networks." Neural computation 3.2 (1991): 246-257.