Artificial Neural Networks
Programming Assignment 2

DUE DATE: DEC 30, 2019

OBJECT OF THE ASSIGNMENT:

To understand how the **Backpropagation algorithm** learns the weight and bias values for ***multilayer networks***. And to realize how different numbers of hidden nodes and different learning rates change the performance of the backpropagation algorithm.

PROBLEM:

Implement the **Stochastic Backpropagation** to classify the popular UCI Iris dataset. Note that you should implement the algorithm from scratch without using any other existing software.

DATASETS:

Your TA will split the Iris dataset into two sub-datasets, 120 examples of the training dataset and 30 examples of the testing dataset. Both datasets have the following characteristics:

- Four numeric attributes: sepal length, sepal width, petal length, and petal width
- Three class labels: Iris-setosa, Iris-versicolor, and Iris-virginica

INPUT OF THE PROBLEM:

Training dataset/testing dataset

OUTPUT OF THE PROBLEM:

Display both of the training and testing accuracies and the number of epochs when the program stops.

EXPERIMENTS:

(a) Build a two-layer neural network with four components in the input layer and three neurons in the output layer. Find experimentally a good number of hidden neurons, start with the number of hidden neurons = 1.

(b) Rerun the experiment for different learning rates.

DISCUSSION:

Write a brief report:

(a) Report which neural network architecture (i.e., different number of hidden nodes) obtained the best performance. Analyze and explain your findings.

(b) Compare the performance of different learning rates.

You may summarize your experimental results using a table or a figure to discuss and analyze your results.

EXTRA CREDIT PORTION (+20 POINTS)

Derive the gradient descent rule for the Softmax activation using the cross-entropy loss and show the derivation in the report. Implement the **Stochastic Backpropagation** and rerun the experiments. Is the performance better than the Sigmoid activation using the squared-error loss? Discuss and explain your findings.

REMARKS

1. Stopping criteria usually includes:
   (a) Stop when a maximum number of epochs has been exceeded.
   (b) Stop when the root mean squared error (RMSE) on the training set D is small enough (other error measures such as the mean absolute error (MAE) can also be used).
      - $E_{RMSE} = \sum_{d \in D} \sqrt{\frac{(t_d - o_d)^2}{|D|}}$ < a given τ_1
      - $E_{MAE} = \sum_{d \in D} |t_d - o_d| / |D|$ < a given τ_2

      where $|D|$: the number of training examples

      t_d: the target of the training example d

      o_d: the neuron output of the training example d
2. One obvious choice of the target vectors for three class labels are: $[1\ 0\ 0]^T$, $[0\ 1\ 0]^T$, and $[0\ 0\ 1]^T$. Instead of 0 and 1 values, use values of 0.1 and 0.9, so that the target vectors are $[0.9\ 0.1\ 0.1]^T$, $[0.1\ 0.9\ 0.1]^T$, and $[0.1\ 0.1\ 0.9]^T$. The reason for avoiding target values of 0 and 1 is that sigmoid units cannot produce these output values given finite weights.