CS-460 Machine Learning
Assignment 2 – Due on Wednesday November 28 at midnight

Design of a Classifier that Recognizes Handwritten Digits

In this assignment you will use the BACKPROPAGATION algorithm we learned in class to build and train an Artificial Neural Network (ANN) classifier that recognizes images of hand-written digits (0-9).

Use the training data (file mnist_train.csv) to build the ANN. Then use the test data (file mnist_test.csv) to test the accuracy of your ANN model.

Training and Test Data:

The training and test data are from the MNIST Database. They are provided to you in .csv format in 2 files:

- **mnist_train.csv**: training data consisting of 60,000 28x28 pixels grayscale images of handwritten digits. The format of this file is:
    - Number of rows = 60,000 (1 row per image).
    - Number of columns = 785 (1 + (28 x 28)).
    - Column 0 is the class label: a digit from 0 to 9
    - Columns 1 to 784 are the pixels (each image is 28 x 28 = 784). Pixel values are from 0 to 255.
- **mnist_test.csv**: testing data consisting of 10,000 28x28 pixels grayscale images of handwritten digits. The format of this file is identical to that of mnist_train.csv. The only difference is that it has 10,000 instead of 60,000 rows.

What You Need To Do:

1. Write a program (preferably in Java, C#, or Python) that uses the BACKPROPAGATION algorithm to build the classifier. You must code the steps of BACKPROPAGATION from scratch. In other words, you cannot use a ready-made Machine Learning package such TensorFlow.

2. You can use 1 hidden layer only. Research shows that the use of 2 hidden layers does not necessarily always improve accuracy. In some cases where more than 1 hidden layer were used, accuracy deteriorated. See the table here which lists the accuracy of different ANN configurations as well as other algorithms such as SVM and *K*-Nearest Neighbor.

3. Vary the number of training images you use to train your model. For instance, try training it against 15,000, 25,000, 40,000, 50,000, and 60,000 images. Record how the accuracy varies

with the number of training samples.

4. Your program should accept a single argument: a path to a .csv file that contains a number of images – one image per row. The file will have a few rows where each row consists of 784 columns (the pixels of the 28x28 image). Your program should output the classification it predicts for each image.

   For example: assume your program is called ANNDigitsClassifier. Then, I will execute your program like this:

   ```
   ANNDigitsClassifier C:\Temp\test.csv
   ```

   If the `test.csv` I pass to you has say 5 images (i.e., 5 rows), then your program should output:

   image 1: x
   image 2: x
   image 3: x
   image 4: x
   image 5: x

   Where x is a digit (from 0 to 9) that your ANN model predicted.

What You Need To Turn In:

1. Your source code (.java or .py or Visual Studio solution folder if using C#).
2. One to two pages summary (in PDF) that describes:

   a. What you did.

   b. The different configurations and parameters you experimented with (e.g., number of computational units in hidden layer, learning rate, number of training samples, etc.). And the accuracy you observed with the different configurations when testing against the test data (mnist_test.csv).

   c. Include a plot of the Learning Curve (Accuracy with respect to sample size). What do you observe?

      Below is an example Accuracy Curve for 2 different algorithms:

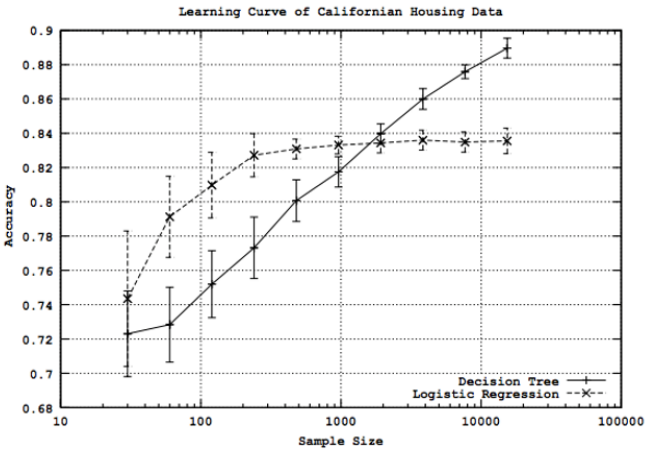

Figure from Perlich et al. *Journal of Machine Learning Research*, 2003

3. Zip all your submission materials (e.g., source code + summary write up) in a zip file. The name of your zip file should be first_last_hw2 (example: "john_smith_hw2.zip") and upload via the course website on Canvas. The deadline is Wednesday November 28 at 11:59 PM.