Deep Learning:
Assignment 1: Shallow Neural Networks

Instructor: Jia Xu

Fall 2020

Due: October 28th 2020

Build a Neural Network on Wine Dataset

You should upload a zip file in the Canvas that contains *everything* to reproduce your systems and plots, including but not limited to codes, comments, data sets (or command to load data), library calls, running commands, output accuracies, tune values, output plots, and a readme file.

step 1: download and install Matlab. (You are allowed to use Python instead of Matlab)

step 2: follow
"https://www.mathworks.com/videos/get-started-with-deep-learning-using-matlab-1564487611332.html"
and
"https://www.mathworks.com/videos/wine-classification-with-neural-pattern-recognition-tool-68798.html"
in "https://www.mathworks.com/discovery/neural-network.html"
to build a neural network.

step 3: test the neural network and output its accuracy. [2 points]

Questions:

1. tune the learning rate and batch size on the validation set and plot the parameter values in the x-axes and the accuracy in the y-axes. output your optimal learning rate and batch size of this neural network. [2 points]

2. use 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the total training dataset to train the neural work and test on it, respectively. You build 9 systems. Plot a graph where the x-axes indicate the percentage of the training set you used to train a system, and y-axes indicate the accuracy on the test set of each system. [2 points]

3. use 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of randomly selected features to train the neural work and test on it, respectively. You build 9 systems. Plot a graph where the x-axes indicate the percentage of features you used to train a system, and y-axes indicate the accuracy on the test set of each system. [2 points]

Hard-Coding a Network

In this problem, you need to find a set of weights and biases for a multilayer perceptron which determines if a list of length 4 is in sorted order. More specifically, you receive four inputs $x_1, ..., x_4$, where $x_i \in R$, and the network must output 1 if $x_1 < x_2 < x_3 < x_4$, and 0 otherwise. You will use the following architecture: All of the hidden units and the output unit use a hard threshold activation function:

$$\phi(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

1. Please give a set of weights and biases for the network which correctly implements this function (including cases where some of the inputs are equal). Your answer should include:

- A 3×4 weight matrix $W^{(1)}$ for the hidden layer [0.5 point]
- A 3-dimensional vector of biases $b^{(1)}$ for the hidden layer [0.5 point]
- A 3-dimensional weight vector $w^{(2)}$ for the output layer [0.5 point]
- A scalar bias $b^{(2)}$ for the output layer [0.5 point]

2. Backprop. Consider a neural network with N input units, N output units, and K hidden units. The activations are computed as follows:

$$z = W^{(1)}x + b^{(1)} \tag{1}$$

$$h = \delta(z) \tag{2}$$

$$y = x + W^{(2)}h + b^{(2)} \tag{3}$$

The cost will involve both h and y:

$$E = R + S \tag{4}$$

$$R = r^T h \tag{5}$$

$$S = \frac{1}{2}||y - s||^2 \tag{6}$$

for given vectors r and s.

- Draw the computation graph relating x, z, h, y, R, S, and E. [1 point]
- Derive the backprop equations for computing $x = \frac{dE}{dx}$. [1 point]