ECE 6604: Machine Learning for Communication Systems – Fall 2019

Project 2: Neural Networks
Due: Friday, November 1, 2019, 5:00pm

In this project, you will learn how to train neural networks using back propagation algorithm. You are given: i) a training dataset: **N-data.txt**, and a validation dataset: **N-valdata.txt**. As discussed in the class, the training dataset should be used to train your selected network architectures while the validation dataset should be used to select the most appropriate network architecture. For each dataset, there are two input variables, x_1 and x_2, and one output variable, y. In each dataset, the first two columns represent the inputs and the third column represents the output. Assume that the output variable is binary, i.e., $y \in \{0, 1\}$ and we consider a classification problem. You are going to train a neural network with a single hidden layer that contains C neurons while the output layer has a single neuron. Note that using the notations in the handout, we have $L = 1$ (the number of hidden layers) and $n(1) = C$ (the number of neurons in the first hidden layer). For this setup, answer the following questions.

1. In order to implement the backpropagation algorithm, you need to obtain the derivatives of the loss function with respect to all the weight parameters for this problem setup. This is the objective of this part. Using the notations in the handout, please show that the derivatives of the loss function (corresponding to the i^{th} training example) with respect to the weight parameters are given by

$$\frac{\partial L^{(i)}}{\partial \mathbf{w}^{(0)}} = \left(g\left(a_o\right) - y_i\right) \begin{bmatrix} \beta_{o1} g'\left(a_1^{(1)}\right) \mathbf{x} \\ \beta_{o2} g'\left(a_2^{(1)}\right) \mathbf{x} \\ \cdots \\ \beta_{oC} g'\left(a_C^{(1)}\right) \mathbf{x} \end{bmatrix}^T , \tag{1}$$

$$\frac{\partial L^{(i)}}{\partial \mathbf{w}^{(1)}} = \left(g\left(a_o\right) - y_i\right) \bar{\mathbf{h}}^{(1)^T} . \tag{2}$$

2. Write a function which performs the feedforward operation. In other words, the inputs to this function should be: i) the weight parameters of the neural network, and ii) a training example, while the outputs of this function are: i) the response vector at the hidden layer, and ii) the response variable at the output layer.

3. Write a function which evaluates the loss function value corresponding to a single training example and its derivatives with respect to the weight parameters. The inputs to this function are: i) the weight parameters of the neural network, and ii) a training example, while the outputs of this function are: i) the value of the loss function, and ii) the derivatives of the loss function with respect to the weight parameters.

4. Now you are going to implement the back propagation algorithm using the functions you created in parts 2 and 3. In this part, consider that $1 \leq C \leq 20$. For each network architecture, i.e., for each value of C, state the training accuracy. It is important to randomly initialize the weight parameters to small values for breaking symmetry. One way of doing so is to randomly select each weight parameter

in the l^{th} layer uniformly in the range $\left[-\psi^{(l)}, \psi^{(l)}\right]$[1], where $\psi^{(l)} = \sqrt{\frac{6}{n(l)+n(l+1)}}$. Notice that the value of $\psi^{(l)}$ depends on which layer the weight parameter belongs to. Clearly state your choices for the learning rate and the number of iterations in the gradient descent algorithm.

5. Use the validation dataset **N-valdata.txt** to select the most appropriate network architecture from amongst the ones that you trained in part 4. Justify your answer for full credit.

6. **Reading Assignment:** In the last part of this project, you are going to read the following paper: `https://arxiv.org/abs/1412.6980`. This paper introduced a new algorithm for stochastic optimization called Adam. Based on your reading, please summarize the key differences between Adam and the stochastic gradient descent (SGD) algorithm.

Submission instructions: Please submit a zip file containing a pdf of your report and a subfolder containing your Matlab codes. Name your zip file as "LastnameFirstname.zip".

You should treat your project report as if your boss were going to read it. The report should not be a simple listing of the results and steps performed. It should be written as a standard technical report, where you should provide some background and context for each result. Your report should also contain a separate subsection titled "Matlab Code", where you should briefly describe the Matlab files that you submitted. Just a few lines description for each file is sufficient. This is just to ensure that we can easily locate the source file that was used to generate a specific result (in case we need to verify something).

Contact: Please direct any specific questions related to this project to GTA Mohamed Abd-Elmagid (maelaziz@vt.edu). Please keep the instructors cc'ed on your emails. Please feel free to contact the instructors directly for any conceptual questions related to the course material.

[1]This initialization criteria is similar to the one used in the following paper: X. Glorot, and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, 2010.