American University of Sharjah

ELE494-09
Deep Networks in Machine Learning

Homework 2

Nasir Mohammad Khalid
65082

March 19, 2019

Submitted To: *Dr. Usman Tariq*

Contents

List of Figures

Listings

1 Task 1: Backpropagation

1.1 Q1

The goal of this part to develop a theoretical understanding of how do you get the expressions with backpropagation algorithm. Suppose you have a three layer fully connected neural network (input layer, hidden layer and output layer). Following are some further ”specifications”:

- The input feature vectors are d dimensional (you can think of these as MNIST images, flattened as vectors)
- There are N feature vectors in you training dataset
- There are H hidden nodes and K output nodes (for K mutually exclusive classes).
- This neural network is to be trained for classification under cross-entropy loss function

Derive the equations for batch gradient updates for the input-to-hidden unit weights and hidden-to-output unit weights. Is it wise to use a learning rate parameter? Why?

1.2 Answer

Based on the specifications given we can assume the architecture of the neural network to be as follows:

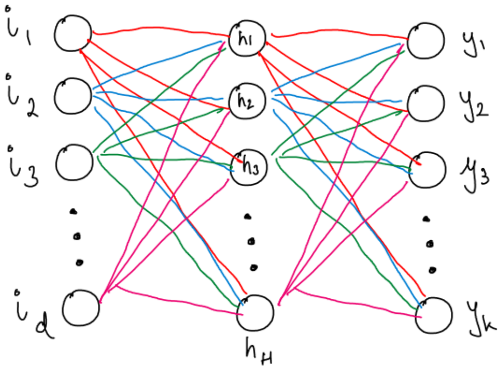

Figure 1: Representation of the neural network

The input to any one of the hidden layer nodes can be given as the sum of the bias and the product of weights and inputs. This will be as follows:

$$I_h = \sum_{n=1}^{d} i_n * w_{nh} \tag{1}$$

where:

I_h = Input to a node 'h' of hidden layer
i_n = Input to a node 'n' of the input layer
w_{nh} = Weight between input node n and the hidden node h

The input to any one of the output layer nodes can be given as the sum of the bias and the product of weights and inputs from the hidden layer. This will be as follows:

$$I_o = \sum_{j=1}^{k} i_j * w_{jk} \tag{2}$$

where:

I_o = Input to a node 'k' of output layer
i_j = Input to a node 'j' of the hidden layer
w_{jo} = Weight between hidden node j and the output node o

Since the output consists of K mutually exclusive classes then we can take the label for the feature vectors to be of a matrix of length K and it contains a 1 at the appropriate label. Then the cross entropy loss function is given as:

$$E = -\sum_{i=1}^{k} Label_k * log(y_k)$$

Since we are using a cross entropy loss function, we therefore assume that the outputs of the hidden layer are the result of a softmax function. Therefore they can be given as:

$$H_h = e^{\sum_{i=1}^{d}}$$