American University of Sharjah

ELE494-09

Deep Networks in Machine Learning

Homework 2

Nasir Mohammad Khalid

65082

March 21, 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, attened 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 Background to Answer

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

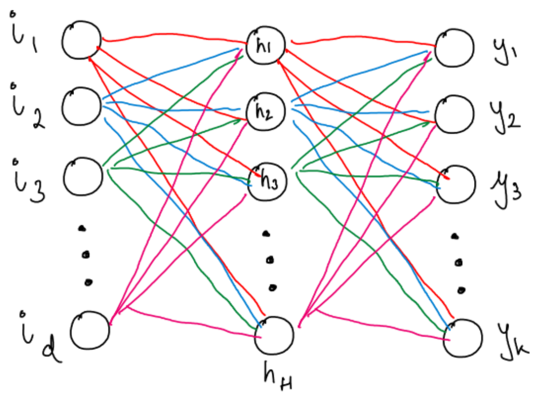

Figure 1: Representation of the neural network

The output of the node 'h' of hidden layer will be calculated using the sigmoid activation function and is given as follows:

$$Z_h = \sum_{n=1}^{d} i_n * w_{nh} + b_h \tag{1}$$

$$O_h = \frac{1}{1 + e^{Z_h}} \tag{2}$$

Z_h = Input to the sigmoid function of node 'h'
O_h = Output of the sigmoid function of the node 'h'
b_h = Bias of the input to node h
w_{nh} = Weight between input node n and the hidden node h

The output of the node 'o' of the final layer will be calculated using the softmax function and it will be given as follows:

$$Z_o = \sum_{n=1}^{h} O_n * w_{no} + b_o \tag{3}$$

$$Y_o = \frac{e^{Z_o}}{\sum_{j=1}^{k} e^{Z_j}} \tag{4}$$

Z_o = Input to the softmax function of node 'o'
Y_o = Output of the 'o'th output layer
b_o = Bias of the Output node 'o'
w_{no} = Weight between the 'n'th hidden node to the 'o'th output node

The output layer will give values between 0 and 1 only. The error function is the cross-entropy function. 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:

$$E = -\sum_{i=1}^{k} L_i * log(Y_i) \tag{5}$$

E = Error value from the cross-entropy function
L_k = Value of the label at the 'k'th index
Y_k = Output of output node 'k'

Now we want to get the derivative of the error with respect to the first set of weights between the hidden layer and output layer:

$$\frac{\partial E}{\partial w_{hk}} = \frac{\partial E}{\partial Z_k} * \frac{\partial Z_k}{\partial w_{hk}} \tag{6}$$

$\frac{\partial E}{\partial w_{hk}} = \partial$ of Error function w.r.t weights between hidden and output layer

$\frac{\partial E}{\partial Z_k} = \partial$ of the Error function with respect to the input to the softmax

$\frac{\partial Z_k}{\partial w_{hk}} = \partial$ of input to softmax w.r.t weights between hidden and output layer

The derivative of the softmax function Y_o with respect to the input of the softmax function Z_k is obtained through the quotient rule and is given by:

$$\frac{\partial Y_o}{\partial Z_k} = Y_k * (1 - Y_j) = Y_k * (1 - Y_k) \quad when \quad j = k \tag{7}$$

$$\frac{\partial Y_o}{\partial Z_k} = -Y_j * Y_k \quad when \quad j \neq k \tag{8}$$

The derivative of the error with respect to the input to the softmax can will consist of two parts. The first is when $i = k$ (in summation) and the second is when they are not the same:

$$\frac{\partial E}{\partial Z_k} = -\sum_{i=1}^{k} L_i * \frac{\partial log(Y_i)}{\partial Y_i} * \frac{\partial Y_i}{\partial Z_k} \tag{9}$$

$$\frac{\partial E}{\partial Z_k} = -L_k * \frac{\partial log(Y_k)}{\partial Y_k} * \frac{\partial Y_k}{\partial Z_k} = -L_k * (1 - Y_j) \quad when \quad i = k \tag{10}$$

$$\frac{\partial E}{\partial Z_k} = -\sum_{i \neq k} L_i * \frac{\partial log(Y_i)}{\partial Y_i} * \frac{\partial Y_i}{\partial Z_k} = \sum_{i \neq k} L_i * Y_j \quad when \quad i \neq k \tag{11}$$

We take the summation of both cases to get the final derivative of error with respect to the input to the softmax function:

$$\frac{\partial E}{\partial Z_k} = -L_k * (1 - Y_j) + \sum_{i \neq k} L_i * Y_j \tag{12}$$

$$\frac{\partial E}{\partial Z_k} = Y_j * (L_k + \sum_{i \neq k} L_i) - L_k \tag{13}$$

but $(L_k + \sum_{i \neq k} L_i)$ is equal to 1 as it is all the intended outputs added up. Due to the fact that the outputs are always probabilities their sum will give 1 and therefore we get the final expression:

$$\frac{\partial E}{\partial Z_k} = Y_j - L_k = Y_k - L_k \tag{14}$$

$\frac{\partial E}{\partial Z_k}$ $= \partial$ of Error function w.r.t input to the softmax function

Y_k $=$ It is the final output of the output neuron

L_k $=$ It is the intended output of the output neuron

Similarly we the derivative of the input to the softmax with respect to the weights. Here O_h is the output of the hidden layer:

$$\frac{\partial Z_k}{\partial w_{hk}} = O_h \tag{15}$$

1.3 A1: Gradient Update for hidden-to-output weights

The final answer is given by:

$$\frac{\partial E}{\partial w_{hk}} = \frac{\partial E}{\partial Z_k} * \frac{\partial Z_k}{\partial w_{hk}} = (Y_k - L_k) * O_h \tag{16}$$

$\frac{\partial E}{\partial w_{hk}}$ $= \partial$ of Error function w.r.t weights between hidden-output

Y_k $=$ It is the final output of the output neuron

L_k $=$ It is the intended output of the output neuron

O_h $=$ It is the final output of the hidden neuron

1.4 A1: Gradient Update for input-to-hidden weights

Here we further extend the partial derivatives until we get the change in error with respect to the input-hidden weights

$$\frac{\partial E}{\partial w_{ih}} = \frac{\partial E}{\partial Z_k} * \frac{\partial Z_k}{\partial O_h} * \frac{\partial O_h}{\partial Z_h} * \frac{\partial Z_h}{\partial w_{ih}} \tag{17}$$

$\frac{\partial E}{\partial w_{ih}} = \partial$ of Error function w.r.t weights between input-hidden

$\frac{\partial E}{\partial Z_k} = \partial$ of Error function w.r.t input to softmax

$\frac{\partial Z_k}{\partial O_h} = \partial$ of Input to softmax w.r.t output of hidden

$\frac{\partial O_h}{\partial Z_h} = \partial$ of Output of hidden w.r.t input to sigmoid

$\frac{\partial Z_h}{\partial w_{ih}} = \partial$ of Input to sigmoid w.r.t weight between input and hidden

We already obtained the first term $\frac{\partial E}{\partial Z_k}$ in section 1.4 and the rest are derived below. The first one (18) where $w_h k$ is the weight between hidden and output layer

$$\frac{\partial Z_k}{\partial O_h} = w_{hk} \tag{18}$$

(19) is derivative of the sigmoid function

$$\frac{\partial O_h}{\partial Z_h} = O_h * (1 - O_h) \tag{19}$$

Here i_n is the input to the network

$$\frac{\partial Z_h}{\partial Z_h} = i_n \tag{20}$$

Combining all of these together gives the final equation for the change in error with respect to the weights between input-hidden layer.

$$\frac{\partial E}{\partial w_{ih}} = (Y_k - L_k) * w_{hk} * Oh * (1 - O_h) * i_n \tag{21}$$

$\frac{\partial E}{\partial w_{ih}}$ = ∂ of Error function w.r.t weights between input-hidden

Y_k = Output of the final neuron

L_k = Intended output of final neuron

w_{hk} = Weights between hidden and output layer

O_h = Output of hidden neuron

i_n = Input of first neuron