Statistical Methods in AI	Date:	*26/02/2019*
Instructor: *Dr. Ravi Kiran*	Submitted By:	Tarun Mohandas (2018201008)

Assignment 5: Neural Networks

1 Model

In this assignment, we explore the basics of neural network with various activation functions. The apparel data set is used which contains greyscale images of various apparels that are each categorized into 10 categories. We train it with a relatively small model. The following is the specification of the model:

1. **Input**: 60000 pictures of apparel of each 28x28 pixels, flattened to 784 columns of data. Therefore input shape is 60000 x 784
2. **Output**: One of the ten categories 0-9
3. **Activation function**: Sigmoid/Tanh/Relu
4. **Loss Function**: Categorical cross entropy function
5. **Layers**:
   (a) **Input Layer**: 784 input neurons
   (b) **Dense Layer 1**: 100 activation neurons
   (c) **Dense Layer 2**: 100 activation neurons
   (d) **Dense Layer 3**: 100 activation neurons
   (e) **Output Layer**: 10 output neurons
6. **Output function**: Softmax function
7. **Parameters**: W_0,W_1,W_2,W_3,b_0,b_1,b_2,b_3
8. **Optimizer**: Batch Gradient Descent

2 The functions

In this section we are going to look at different functions that has been used in the model.

Note that for all functions below, $z = W_i X + b_i$ for every layer i

1. Sigmoid function
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

2. Tanh function
$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

3. ReLU function
$$ReLU(z) = max(0, z)$$

4. Softmax function
$$softmax(z) = \frac{e^{z_i}}{\sum_n e^{z_n}} \forall i \in n$$

5. Categorical cross entropy
$$L(\hat{y}, y) = \sum y log\hat{y}$$

We have also used the derivative of these functions for gradient descent update. They are as follows:

1. Derivative of sigmoid function
$$\sigma^{'}(z) = \sigma(z)(1 - \sigma(z))$$

2. Derivative of tanh function
$$tanh^{'}(z) = 1 - (tanh(z))^2$$

3. Derivative of ReLU function
$$ReLU^{'}(z) = \begin{cases} 0 & z < 0 \\ 1 & z > 0 \end{cases}$$

3 Gradient Descent

This model uses Gradient Descent Optimizer. The parameters W and b are to be updated in each epoch. The following is the update rules.

$$W_i = W_i - \alpha \frac{dL}{dW}$$
$$b_i = b_i - \alpha \frac{dL}{db}$$

where α is learning rate

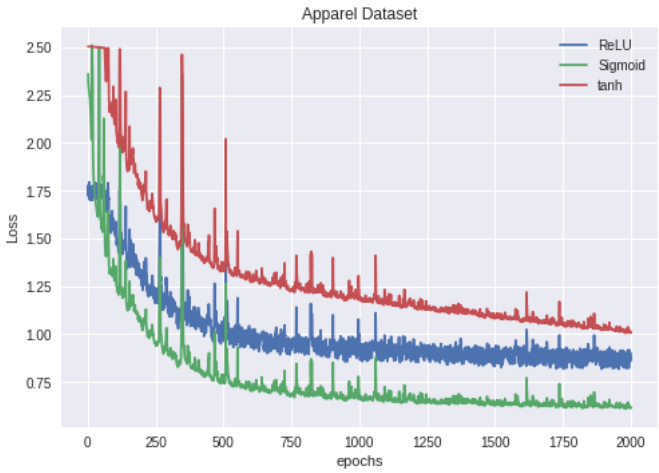

Figure 1: Effect of activation functions

4 Analysis

4.1 Effect of different activation functions

The effect of different activation functions in the neural network is evident by the graph show in Figure 1. As we can see the ReLU, Sigmoid and tanh function seems to be reaching the same loss but are taking longer. tanh is the slowest and sigmoid appears to be the fastest to reach the optimal loss.

4.2 Effect of number of layers

The effect of number of layers will be evident as we increase the layers. As the number of layers increases, the loss function becomes much flatter and it will overfit the data. It also causes slow descent into the optimal loss.

5 Question 2

5.1 The question

Consider the House Price Prediction dataset . Suppose you need to predict the Sale Price of a house and for the task you want to use a neural network with 3 hidden layers. Write a report on how you would modify your above neural network for such task with proper reasoning.

5.2 Answer

The number of layers and number of neurons in each layer, and the input layer can remain the same, including all the activation functions. Although, we need to change the output function (because it is a regression problem), Loss function (because it

does not involve probability distribution), Output Layer(because it need to contain that many neurons. Just one is enough). The following are the details.

1. **Output Function**: Linear function $Wx + b$
2. **Loss Function**: Mean square error