CS 6375
ASSIGNMENT -2
(Neural Networks)

Names of students in your group:

Siddhant Suresh Medar (ssm200002)
Adithya Sundararajan Iyer (asi200000)

Number of free late days used: ___________1___________

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

1 Theoretical Part (40 points)

1.1 Gradient Descent

Derive a gradient descent training rule for a single unit neuron with output o, defined as:

$$o = w_0 + w_1(x_1 + x_1^2) + \cdot\cdot\cdot + w_n(x_n + x_n^2)$$

where $x_1, x_2, \ldots, x_n$ are the inputs, $w_1, w_2, \ldots, w_n$ are the corresponding weights, and w_0 is the bias weight. Show all steps of your derivation and the final result for weight update. You can assume a learning rate of η.

Gradient descent is an iterative first-order optimization algorithm that is used to discover the local minimum/maximum of a function. In other words, it tries to progressively reduce error or minimize loss by updating the weights at each step.

The perceptron training rule is:

$$w_i \leftarrow w_i + \Delta w_i$$

where:

$$\Delta w_i = \eta(t - o)x_i = -\eta \frac{\partial E}{\partial w_i}$$

Now output: $o = \sum_{i=0}^{n} w_i\left(x_i + x_i^2\right)$

Take activation function: $f(x) = x \quad f'(x) = 1$

According to the gradient descent rule:

$$O = w \cdot x$$

$$E(w) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

η – learning rate
E – error function
D – set of training samples
t_d – target output for the *d* training sample
o_d – predicted output for the *d* training sample

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i}\frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 = \frac{1}{2}\sum_{d\in D} 2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d)$$

$$\frac{\partial E}{\partial w_i} = \sum_{d\in D}(t_d - o_d)\frac{\partial}{\partial w_i}\left(t_d - w_i(x_i + x_i^2)\right)$$

$$\frac{\partial E}{\partial w_i} = \sum_{d\in D}(t_d - o_d)\left(-x_i - x_i^2\right)$$

The value of Δw_i can thus be obtained as:

$\Delta w_i = -\eta \sum_{d\in D}(t_d - o_d)\left(-x_{id} - x_{id}^2\right)$

Δw_i – represents the result of the weight update for gradient descent
x_{id} – denotes the single input component x_i for the training sample *d*

$$\therefore\ \Delta w_i = \eta \sum_{d\in D}(t_d - o_d)\left(x_{id} + x_{id}^2\right)$$

1.2 Comparing Activation Function

Consider a neural net with 2 input layer neurons, one hidden layer with 2 neurons, and 1 output layer neuron as shown in Figure 1. Assume that the input layer uses the identity activation function i.e. f(x) = x, and each of the hidden layers and output layer use an activation function h(x). The weights of each of the connections are marked in the figure.

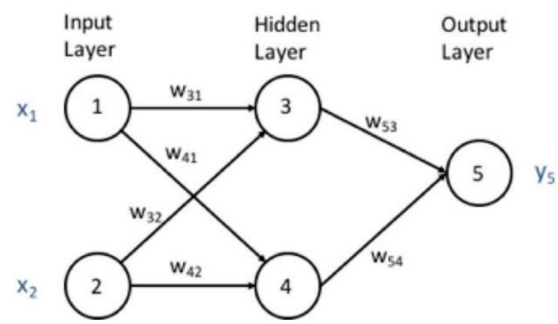

Figure 1: A neural net with 1 hidden layer having 2 neurons

a. Write down the output of the neural net y_5 in terms of weights, inputs, and a general activation function h(x).

Output of the hidden layer neurons:

$$y_3 = h(w_{31}x_1 + w_{32}x_2)$$

$$y_4 = h(w_{41}x_1 + w_{42}x_2)$$

Output of the neural net:

$$y_5 = h(w_{53}y_3 + w_{54}y_4)$$

$$\therefore y_5 = h\big(w_{53} \cdot h(w_{31}x_1 + w_{32}x_2) + w_{54} \cdot h(w_{41}x_1 + w_{42}x_2)\big)$$

b. Now suppose we use vector notation, with symbols defined as below:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$W^{(1)} = \begin{pmatrix} w_{3,1} & w_{3,2} \\ w_{4,1} & w_{4,2} \end{pmatrix}$$

$$W^{(2)} = \begin{pmatrix} w_{5,3} & w_{5,4} \end{pmatrix}$$

Write down the output of the neural net in vector format using above vectors.

$$\text{Output of the hidden layer} = \mathrm{h}\big(W^{(1)} \cdot X\big)$$

$$\text{Output of the neural net} = \mathrm{h}\Big(W^{(2)} \cdot \mathrm{h}\big(W^{(1)} \cdot X\big)\Big)$$

c. Now suppose that you have two choices for activation function h(x), as shown below:

Sigmoid: $h_s(x) = \frac{1}{1+e^{-x}}$

Tanh: $h_t(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Show that neural nets created using the above two activation functions can generate the same function.

$$h_s(x) = \frac{1}{1+e^{-x}} = \frac{1}{1+\frac{1}{e^x}} = \frac{e^x}{e^x+1}$$

$$h_t(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^x - \frac{1}{e^x}}{e^x + \frac{1}{e^x}} = \frac{e^{2x}-1}{e^{2x}+1} = \frac{e^{2x} + (e^{2x} - e^{2x}) - 1}{e^{2x}+1}$$

$$h_t(x) = \frac{2e^{2x} - e^{2x} - 1}{e^{2x}+1} = 2\frac{e^{2x}}{e^{2x}+1} - \frac{e^{2x}+1}{e^{2x}+1}$$

$$h_t(x) = 2\frac{1}{1+e^{-2x}} - 1$$

But $2\frac{1}{1+e^{-2x}} = h_s(2x)$

$$\therefore \boldsymbol{h_t(x) = 2h_s(2x) - 1}$$

Here we see that the two activation functions sigmoid and tanh have a linear relationship, i.e., *$h_t(x) = A \cdot h_s(x) + B$*

The *sigmoid* and the *tanh* functions can be derived from one another by performing some linear transformations and adding/subtracting a constant term.

As a result, it is proven that *sigmoid* and *tanh* can generate the same function.