# Fall 2020: CSCI 4/5588 Homework #1

# Name: Tharani Maaneeivaannan ID:2575198

# Task 1:

Wrote the below code in MATLAB to produce the Figure-1

>> x=-6.0:0.1:6.0;

>> y=(1./(1+exp(-x)));

>> plot(x,y,'Linewidth',3)

>> hold on;

>> x=-6.0:0.1:6.0;

>> y=((exp(x)-exp(-x))./((exp(x)+exp(-x))));

>> plot(x,y,'Linewidth',3)

>> grid on

>> grid minor

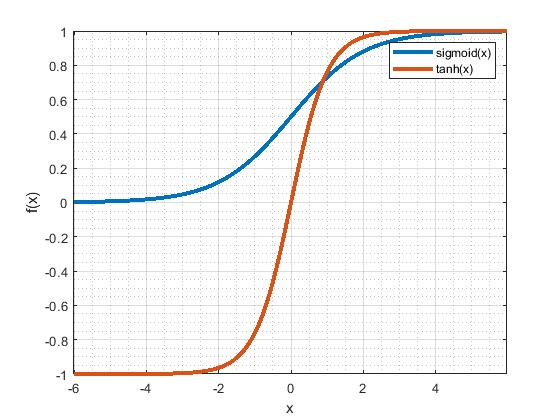


Figure 1 Sigmoid and Hyperbolic tangent graph

# Task 2

Let us define the output error,

Here, layer l =1, 2, …,L and L = Output Layer. And, is the target/original output and is the predicted/network output.

The back-propagation learning rule is based on gradient descent. This is because we want

to minimize the error by changing the weights. The weights are initialized with random

values, and then they are changed in a direction that will reduce the error:

where or and is the learning rate. The overall idea of the error back-propagation will be to compute the discrepancy of the targeted output versus the predicted output from output node(s) first and then to propagate backward to compute the discrepancy for each of the nodes in the hidden layer except the input layer – because we do not want to change the input layer. Then we will update the weight or the parameter for the network to minimize the error. First, we will compute the rate of changes of error with respect to the weight connected to the output node(s): Differentiating E( with respect to

Using chain rule,

Substituting

Differentiating with respect to ,

Note:

Using chain rule,

Therefore,

Assume the error term for the output layer,

Finally, for output layer we have,

where

Let us now compute the rate of changes of error with respect to the weight for the hidden

layer(s) similarly:

Substituting

Using chain rule,

Using chain rule,

Note:

As we have seen before in

Finally, for the hidden layer we have,

Or we can write it as

where

Involvements of the Bias term:

If we use Equation (iii) for the bias term at layer (L-1), we can rewrite equation (iii) as:

Similarly, for the other hidden layers following equation (iv) and the idea of equation (v),

we can write:

Now, we have generated the required equation to describe the back-propagation algorithm. We will explain the backpropagation algorithm next, and then we will illustrate the idea in further detail:

## Algorithm 1: Error Backpropagation

BEGIN

1. From a data point (xi, yi), apply an input vector xi to the network and forward propagate through the network (as we computed the logistic regression).
2. For each of the output node/unit compute the error term
3. For each of the hidden layer node /unit compute the error term

And Compute because it is the input layer and we do not want to change the original input data.

1. Update the weights ( ) of the network [See Equations: i, iii, iv, v, vi]

where, i=1, 2, …, (L-1)

END