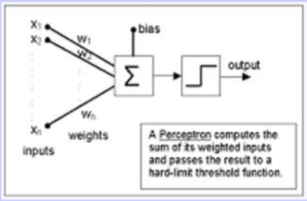
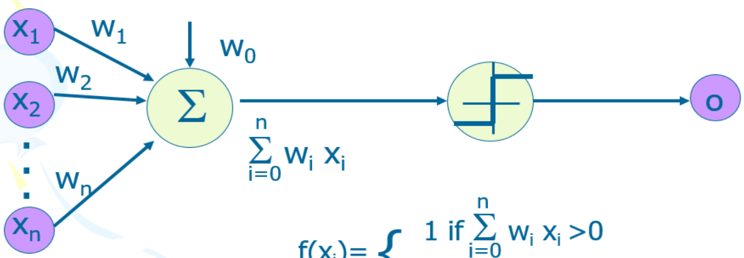
MSc CS Soft Computing Lecture (04 feb. 2021 10:30 am to 12 pm)

PERCEPTRON NETWORKS

- One type of NN system is based on the "perceptron".
- A perceptron computes a sum of weighted combination of its inputs, if the sum is greater than a certain threshsold (bias), then it ouputs a "1", else a "-1".



Perceptron Network:



$$f(x_i) = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$$

Epoch : Presentation of the entire training set to the neural network.
In the case of the AND function, an epoch consists of four sets of inputs being presented to the network (i.e. [0,0], [0,1], [1,0], [1,1]).

Error: The error value is the amount by which the value output by the network differs from the target value. For example, if we required the network to output 0 and it outputs 1, then Error = -1.

LEARNING ALGORITHM

Target Value, T : When we are training a network we not only present it with the input but also with a value that we require the network to produce. For example, if we present the network with [1,1] for the AND function, the training value will be 1.

Output , O : The output value from the neuron.

Ij : Inputs being presented to the neuron.

Wj : Weight from input neuron (I_j) to the output neuron.

LR : The learning rate. This dictates how quickly the network converges. It is set by a matter of experimentation. It is typically 0.1.

MSc CS Soft Computing Lecture (04 feb. 2021 10:30 am to 12 pm)

TRAINING ALGORITHM

- Adjust neural network weights to map inputs to outputs.
- Use a set of sample patterns where the desired output (given the inputs presented) is known.
- The purpose is to learn to
  - Recognize features which are common to good and bad exemplars

Single Layer Perceptron Training Algorithm (single output classes)

Step 0: Initialize the weights and the bias (for easy calculation they can be set to zero). Also initialize the learning rate $\alpha (0 < \alpha \leq 1)$. For simplicity α is set to 1.

Step 1: Perform Steps 2–6 until the final stopping condition is false.

Step 2: Perform Steps 3–5 for each training pair indicated by $s:t$.

Step 3: The input layer containing input units is applied with identity activation functions:

$$x_i = s_i$$

Step 4: Calculate the output of the network. To do so, first obtain the net input:

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

where "n" is the number of input neurons in the input layer. Then apply activations over the net input calculated to obtain the output:

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \leq y_{in} \leq \theta \\ -1 & \text{if } y_{in} < -\theta \end{cases}$$

Step 5: *Weight and bias adjustment:* Compare the value of the actual (calculated) output and desired (target) output.

If $y \neq t$, then

$$w_i(\text{new}) = w_i(\text{old}) + \alpha t x_i$$
$$b(\text{new}) = b(\text{old}) + \alpha t$$

else, we have

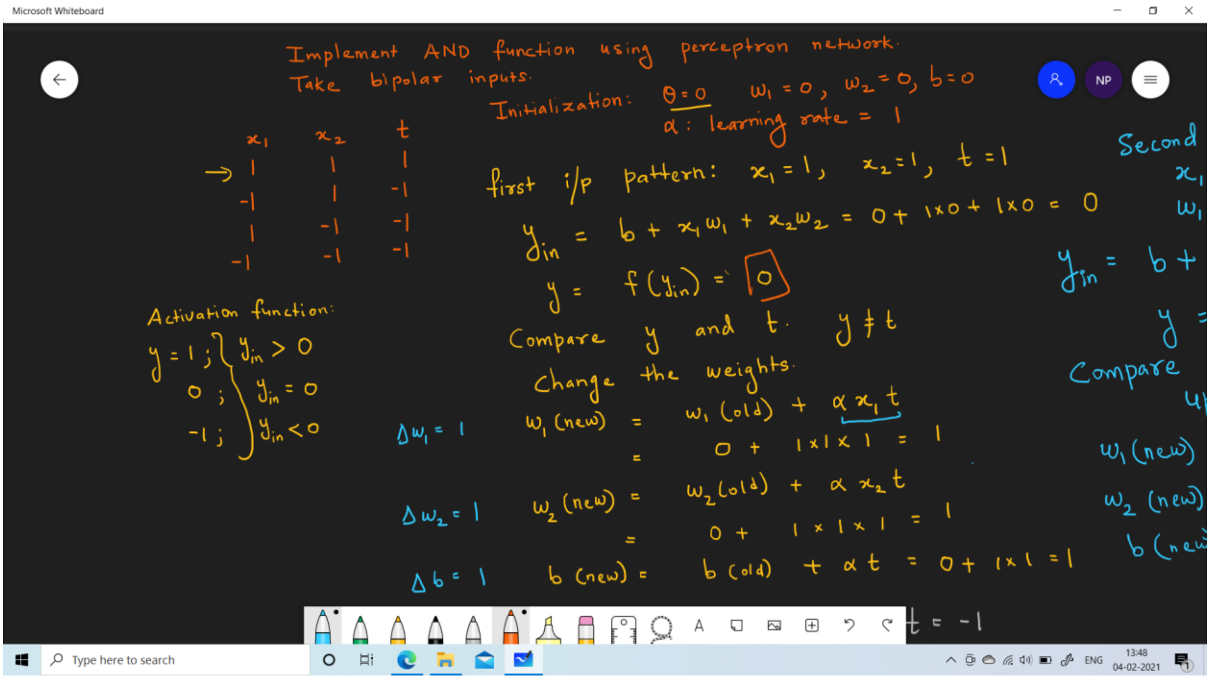
$$w_i(\text{new}) = w_i(\text{old})$$
$$b(\text{new}) = b(\text{old})$$

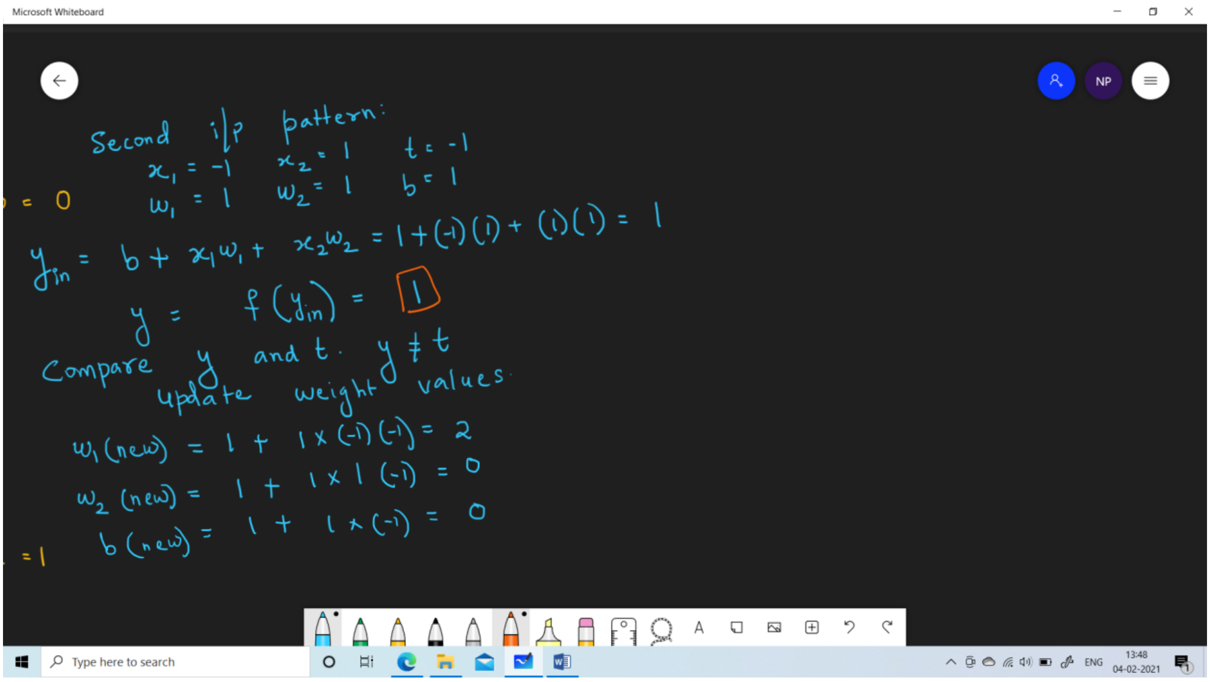
Step 6: Train the network until there is no weight change. This is the stopping condition for the network. If this condition is not met, then start again from Step 2.

MSc CS Soft Computing Lecture (04 feb. 2021 10:30 am to 12 pm)

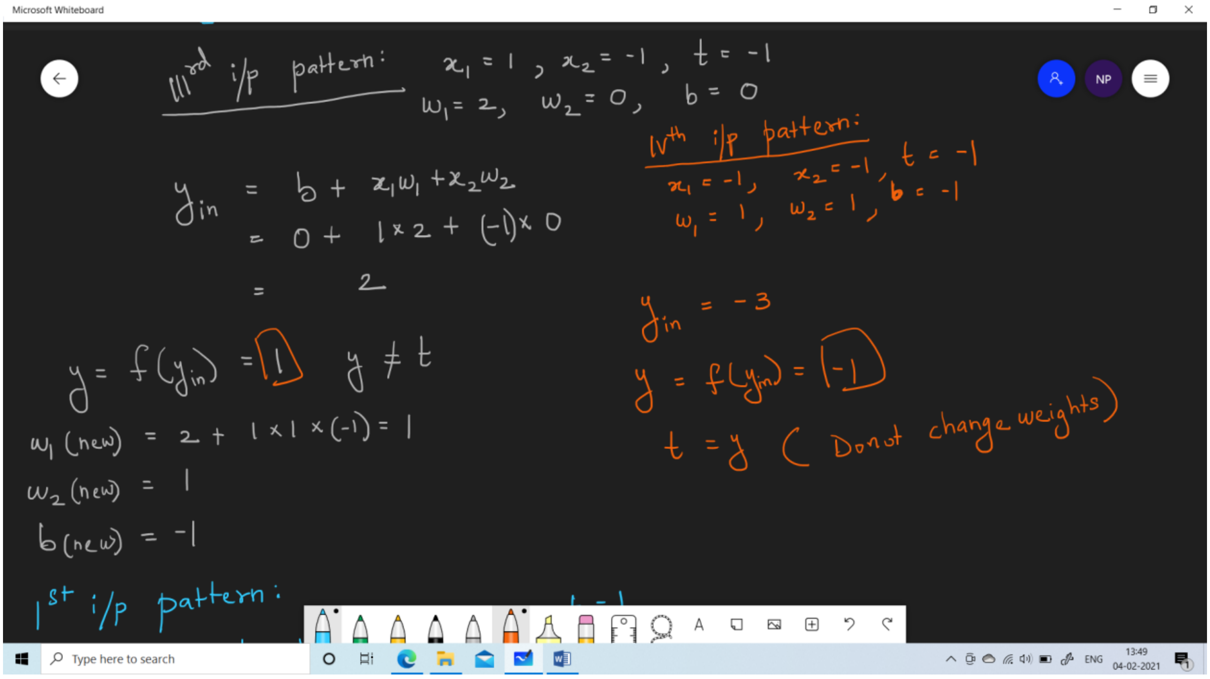
Example

Epoch 1:





MSc CS Soft Computing Lecture (04 feb. 2021 10:30 am to 12 pm)



Epoch 2:

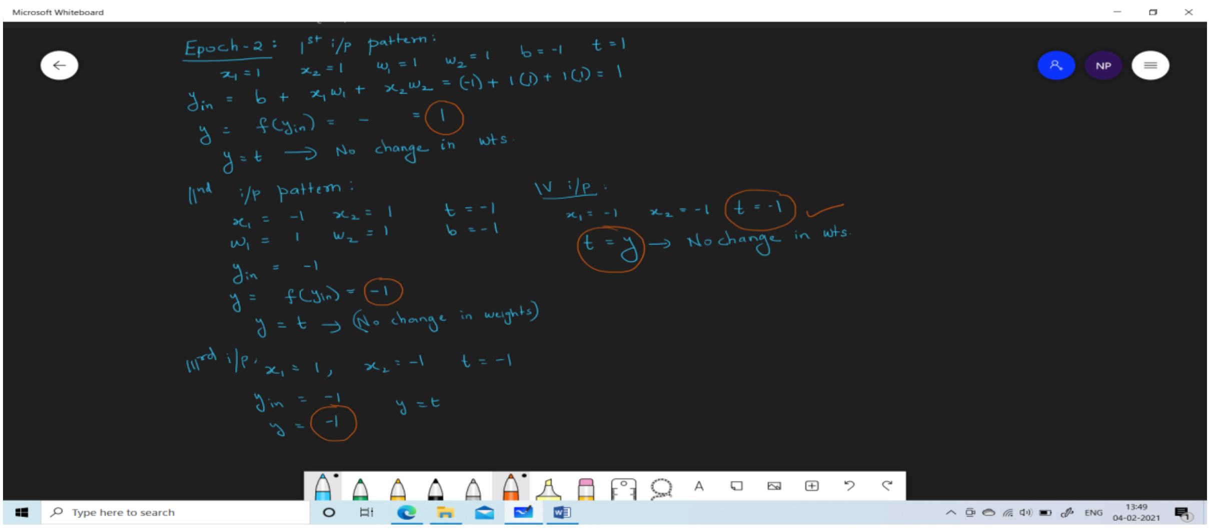


Table 2

Input			Target	Net input	Calculated output	Weight changes			Weights		
x_1	x_2	1	(t)	(y_{in})	(y)	Δw_1	Δw_2	Δb	w_1 (0	w_2 0	b 0)
EPOCH-1											
1	1	1	1	0	0	1	1	1	1	1	1
1	−1	1	−1	1	1	−1	1	−1	0	2	0
−1	1	1	−1	2	1	+1	−1	−1	1	1	−1
−1	−1	1	−1	−3	−1	0	0	0	1	1	−1
EPOCH-2											
1	1	1	1	1	1	0	0	0	1	1	−1
1	−1	1	−1	−1	−1	0	0	0	1	1	−1
−1	1	1	−1	−1	−1	0	0	0	1	1	−1
−1	−1	1	−1	−3	−1	0	0	0	1	1	−1

Practice Problem no. 2, 3 and 4 from reference book chapter no. 03.