# Mody University of Science and Technology
## School of Engineering and Technology

### Mid Term Examination- I, Spring Semester 2024-2025

**Program Name:** B. Tech. – CSE | **Year:** III year

**Course Code:** CS 19.312 | **Max. Mark:** 20

**Course Name:** Soft Computing | **Weight:** 100%

**Total no. of Printed Pages:** 2 | **Time:** 60 Minutes

**Instructions to candidates:**
1. Question number 1 is compulsory; each short answer type question is of 1 mark.
2. Attempt any three questions out of four (Question number 2 to 5); each question is of 5 marks.

---

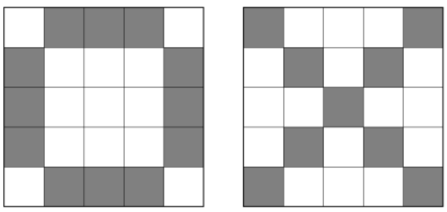
Q1. Write the equation for each activation function

- a). Identity activation function
- b). Bipolar activation function
- c). Rectified linear unit (ReLU) activation function
- d). Logistic activation function
- e). Hyperbolic tangent activation function

Q2. The given python code implements McCulloch-Pitts neural network model. Analyze the code and draw the explanatory diagram of the activation function used. Write the output of line no. 6 & 7 (Hint: Calculate and write the weighted sum ($Y_{in}$) and output (Y) of the neural network, based on input (X) and weight (W) values as given in line no. 9 and 10).

```
1    import tensorflow as tf
2    def mcculloh_pits_model(inputs, weights, threshold):
3           mul = tf.multiply(inputs,weights)
4           weighted_sum = tf.reduce_sum(mul)
5           output = tf.cond(weighted_sum >= threshold,lambda:1.0,lambda:0)
6           print("Y_in = ",weighted_sum)
7           print("Y = ", output)
8           return output
9    inputs = tf.constant([5,3,2],dtype=tf.float32)
10   weights = tf.constant([2,3,3],dtype=tf.float32)
11   threshold = tf.constant(25.5,dtype=tf.float32)
12   output = mcculloh_pits_model(inputs, weights, threshold)
```

Q3. For the given two input image patterns, perform binary classification using a single layer Hebb net (architecture + algorithm). Graphically illustrate Hebbian learning for the given problem. Write down the equations satisfying linear separation and final weight matrix after training of the network is complete.
Hint: Represent network as bipolar neuronal model (Both Input and output neurons)
Assumption: Dark pixels depict positive signal, white pixels represent lack of signal.

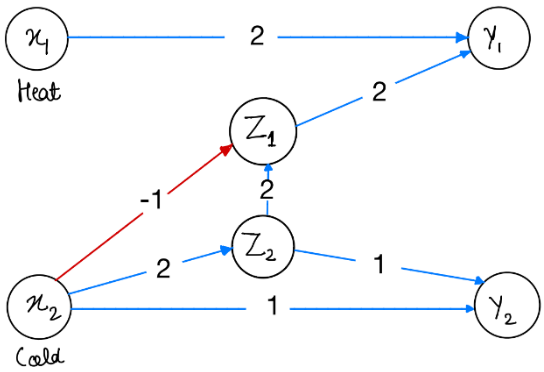


Q4. Our brain network has the ability to sense heat and cold touch through skin contact. It is true that when a cold stimulus is applied to a person's skin for a very short period, the subject perceives heat. Contrarily, if the cold touch is present for a longer period, the subject will sense cold sensation.

For the given neural network where, threshold for each neuron is 2, use the discrete time step feature of McCulloch-Pitts model to demonstrate the above phenomenon. Write the equations of the model to predict and prove (graphically/exemplify) heat/cold (0/1) sensation.

Hint:

- Use logic gate truth table
- Perform logical operations at different time points (i.e. t = 0, t = 1, ... )
- Heat stimulus can be represented as [1 0]
- Cold stimulus is better represented as [0 1]
- Elongated exposure can be modeled with two consecutive time points.




Q5. Briefly explain Hebb net. Draw its schematic representation of the architecture and graphically illustrate (timeline model) decision boundary and its application to solve OR gate classification problem.

Hint:

Use bipolar type input and output neurons to calculate the final weight vector **[ w1 w2 b ]** after training Hebb net with **OR** gate input vectors **[ x1 x2 1 ]** with corresponding targets is complete.

---

**Answer 1.**

a). $f(x) = x$

b).
$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x \leq 0 \end{cases}$$

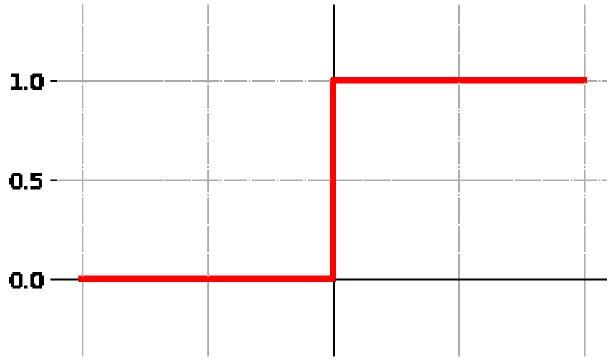
c). $f(x) = \max(0, x)$

d).
$$f(x) = \frac{1}{1 + e^{-x}}$$

e).
$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

**Answer 2.**

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$




```
Yin = 10 + 9 + 6 = 25
Threshold = 25.5
Yin < Threshold
Therefore Y = 0
```

---

**Answer 3.**

| | | | | |
|---|---|---|---|---|
| -1 | 1 | 1 | 1 | -1 |
| 1 | -1 | -1 | -1 | 1 |
| 1 | -1 | -1 | -1 | 1 |
| 1 | -1 | -1 | -1 | 1 |
| -1 | 1 | 1 | 1 | -1 |

For above X, Y = -1

| | | | | |
|---|---|---|---|---|
| 1 | -1 | -1 | -1 | 1 |
| -1 | 1 | -1 | 1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | 1 | -1 | 1 | -1 |
| 1 | -1 | -1 | -1 | 1 |

For above X, Y = 1

Where X = $[X_1\ X_2\ X_3\ .\ .\ .\ X_{25}]$
There are only two training data points, but multiple dimensions act as individual inputs giving robustness to neural network.

More information can be seen at online resource.
https://github.com/Depthgr8/Softy/blob/main/Hebb_Net_based_classification.ipynb

---

**Answer 4.**

Timeline model can be represented with multiple snapshots of architecture at various time points (t = 0, t = 1, t = 2).
Since the architecture is fixed throughout the experiment, a logic table can be drawn for the neural network
Short cold stimulus is when stimulus (1) is present for t = 0 only
Long cold stimulus is when stimulus (1) is present for t = 0 and t = 1 time point

A truth table based simulation method can be used to understand the expected behavior of the network

Logic gate equations for simulation of each neuron at t, time point

1. $Z_{2(t)} = X_{2(t-1)}$
2. $Z_{1(t)} = Z_{2(t-1)}$ AND-NOT $X_{2(t-1)}$
3. $Y_{1(t)} = Z_{1(t-1)}$ OR $X_{1(t-1)}$
4. $Y_{2(t)} = Z_{2(t-1)}$ AND $X_{2(t-1)}$

Experiment no. 1 – Short cold exposure (for one time point)

| Simulation table | $X_1$ | $X_2$ | $Z_1$ | $Z_2$ | $Y_1$ | $Y_2$ |
|---|---|---|---|---|---|---|
| Experiment 1 cold (t = 0) | 0 | 1 | | | | |
| Experiment 1 cold (t = 1) | 0 | 0 | 0 | 1 | | |
| Experiment 1 cold (t = 2) | | | 1 | 0 | 0 | 0 |
| Experiment 1 cold (t = 3) | | | | | 1 | 0 |

Experiment no. 2 – Elongated cold exposure (for two consecutive time points)

| Simulation table | $X_1$ | $X_2$ | $Z_1$ | $Z_2$ | $Y_1$ | $Y_2$ |
|---|---|---|---|---|---|---|
| Experiment 2 cold (t = 0) | 0 | 1 | | | | |
| Experiment 2 cold (t = 1) | 0 | 1 | 0 | 1 | | |
| Experiment 2 cold (t = 2) | | | 0 | 1 | 0 | 1 |

---

**Answer 5.**

Weights after Hebbian learning = `[2 2 2]`
Perfect answer is where decision boundary is drawn for training of each input pair (In timeline fashion). Lenient marking is when at least weights have been calculated for OR gate classification problem

More information can be seen at online resource
https://github.com/Depthgr8/Softy/blob/main/Hebb_Net_OR_Gate.ipynb