

# Machine Learning For Robotics (CS 4090)

Date: November 3, 2025

**Course Instructor(s)**

Dr. Mirza Mubasher Baig

Sessional-II

Total Time (Hrs): 1  
Total Marks: 30  
Total Questions: 3

221-0804

Roll No.

BCC-20

Section

四百九

---

**Student Signature**

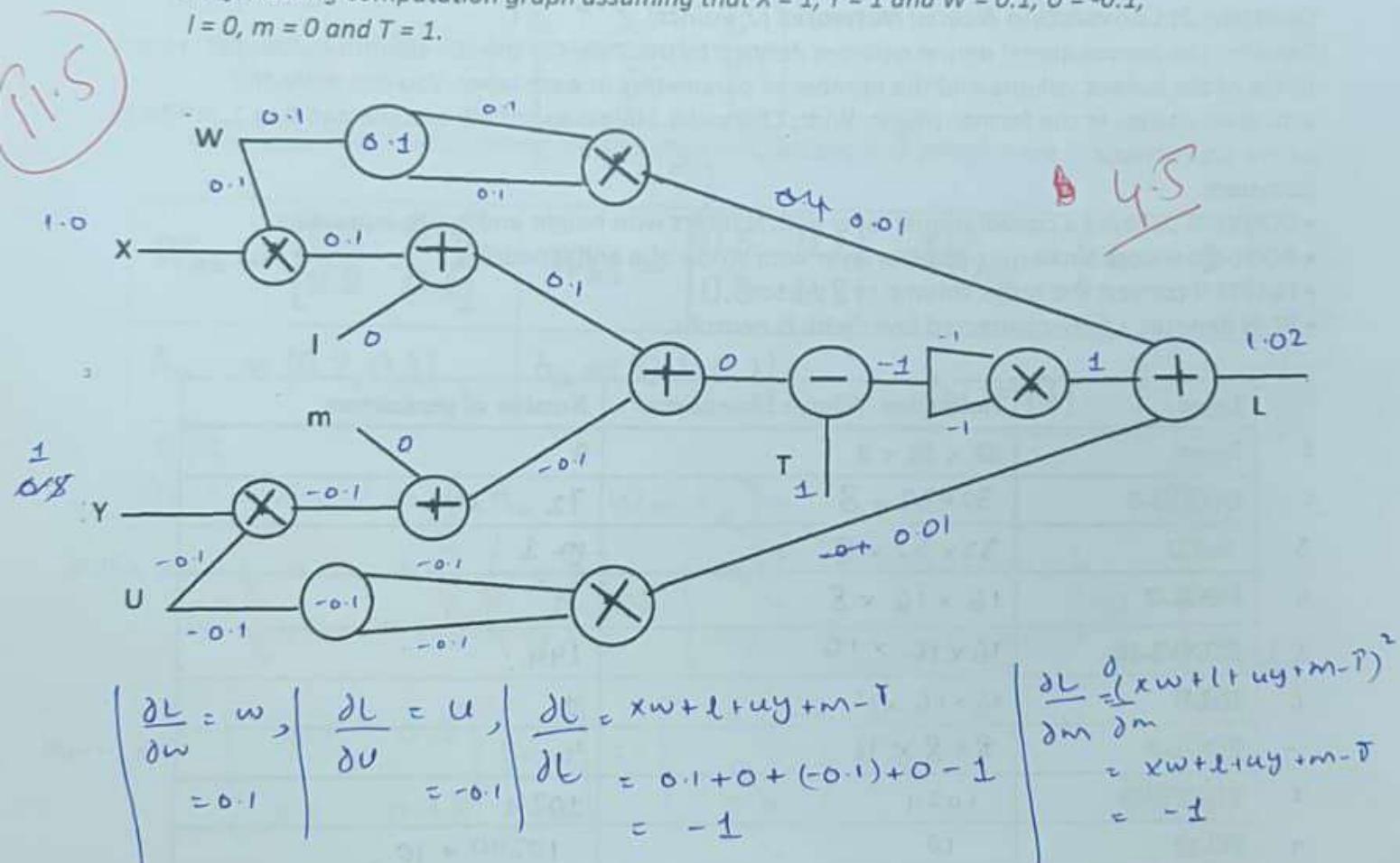
**Instructions:** Attempt all questions.

**Write your answers in the space provided for answer**

Attach extra sheets if absolutely necessary.

#### **Question 1: Feed-Forward Neural Networks [8 + 2 + 5 Points]**

- i) For the input  $x = 1.0$  and  $y = 0.5$ , compute the gradient of  $L$  with respect to  $W$ ,  $U$ ,  $I$  and  $m$  using the following computation graph assuming that  $X = 1$ ,  $Y = 1$  and  $W = 0.1$ ,  $U = -0.1$ ,  $I = 0$ ,  $m = 0$  and  $T = 1$ .



$$\begin{array}{l}
 \text{ii) Compute the updated values of } W, U, l \text{ and } m \text{ using learning rate of 0.1} \\
 \begin{array}{c|c|c|c}
 W = w - \eta \frac{\partial L}{\partial w} & U = U - \eta \left( \frac{\partial L}{\partial U} \right) & l = l - \eta \left( \frac{\partial L}{\partial l} \right) & m = m - \eta \left( \frac{\partial L}{\partial m} \right) \\
 = 0.1 - (0.1)(0.1) & = -0.1 - (0.1)(-0.1) & = 0 - (0.1)(-1) & = 0 - (0.1)(-1) \\
 W = 0.09 & = -0.09 & = 0.1 & = 0.1
 \end{array}
 \end{array}$$

- iii) In an assignment a student used a neural network with two hidden layers and one output layer to classify an image as a digit belonging to one of the digits 0, 1, ..., 9. He used 67 neurons in the first hidden layer and 82 neurons in the second hidden layer. Compute the size of his matrices used in the computation

5

Input layer -  $x$

$$\begin{array}{l}
 w_1 = (67)(x) + 67 \Rightarrow 67x + 67 \\
 w_2 = (67)(x)(82)(82)(67) + 82 \Rightarrow 5494 + 82 \\
 w_3 = (67)(x)(82)(10)(82) + 10 \Rightarrow 820 + 10 \\
 w_1 = [x \times 67], b_1 = [67], w_2 = [82 \times 67], b_2 = [82] \\
 w_3 = [10 \times 82], b_3 = [10]
 \end{array}$$

**Question 2: Convolution Neural Networks [7 Points]**  $w_3 = [10 \times 82], b_3 = [10]$

Consider the convolutional neural network defined by the layers in the left column below. Fill in the shape of the output volume and the number of parameters at each layer. You can write the activation shapes in the format (Height; Width; Channels). Unless specified, assume padding 1, stride 1 where appropriate.

Notation:

- CONVx-N denotes a convolutional layer with N filters with height and width equal to x.
- POOL-n denotes a nxn max-pooling layer with stride of n and 0 padding.
- FLATTEN convert the input volume to a vector
- FC-N denotes a fully-connected layer with N neurons

(5)

Layer	Activation Volume Dimensions	Number of parameters
1 Input	$32 \times 32 \times 3$	0
2 CONV3-8	$32 \times 32 \times 8$ ✓	72 ✓
3 ReLU	$32 \times 32 \times 8$ ✓	✓
4 POOL-2	$16 \times 16 \times 8$ ✓	4 ✓
5 CONV3-16	$16 \times 16 \times 16$ ✓	144 ✓
6 ReLU	$16 \times 16 \times 16$ ✓	1 ✓
7 POOL-2	$8 \times 8 \times 16$ ✓	4 ✓
8 FLATTEN	1024 ✓	1024 ✓
9 FC-10	10 ✓	$10240 + 10$ ✓

## Question 3: Recurrent Neural Networks [2 + 6 Points]

- i) What would the phrase "RNN is Turing complete" means to a programmer? Explain it using a single sentence

RNN can be used to solve problem of any kind whether linear, non-linear or any other requiring memory.

1

- ii) Given the a RNN using the following expressions during computation

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

Compute the output corresponding to the input sequence

Input Sequence		
X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>
1	1	0
1	0	1

assuming that the following weight matrices, biases and initial state is used

$W_{xh} = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix}$	$W_{hh} = \begin{bmatrix} 0.5 & 0.1 \\ 0.3 & 0.2 \end{bmatrix}$	$W_{hy} = \begin{bmatrix} 1 & 3 \end{bmatrix}$
$h_0 = [0.2, 0.1]$	$b_h = [0.1, 0.1]$	$b_y = [0]$

$t=1$   
 $h_1 = \tanh(W_{hh}h_0 + W_{xh}x_1)$

$$W_{hh}h_0 = \begin{bmatrix} 0.5 & 0.1 \\ 0.3 & 0.2 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix} \Rightarrow \begin{bmatrix} (0.5)(0.2) + (0.1)(0.1) \\ (0.3)(0.2) + (0.2)(0.1) \end{bmatrix} \Rightarrow \begin{bmatrix} 0.11 \\ 0.08 \end{bmatrix}$$

$$W_{xh}x_1 = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$$

6