|  |
| --- |
| **Deep Learning for Perception(CS4045)** |
| Date: March 1st 2025 |
| **Course Instructor(s)** |
| Mr. Syed Irtaza Muzaffar |

|  |  |
| --- | --- |
| **Sessional-I Exam** | |
| **Total Time (Hrs):** | **1** |
| **Total Marks:** | **30** |
| **Total Questions:** | **3** |

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Roll No Section Student Signature |

**Attempt all the questions.**

***CLO2 #: Understand and design the structure of deep neural networks..***

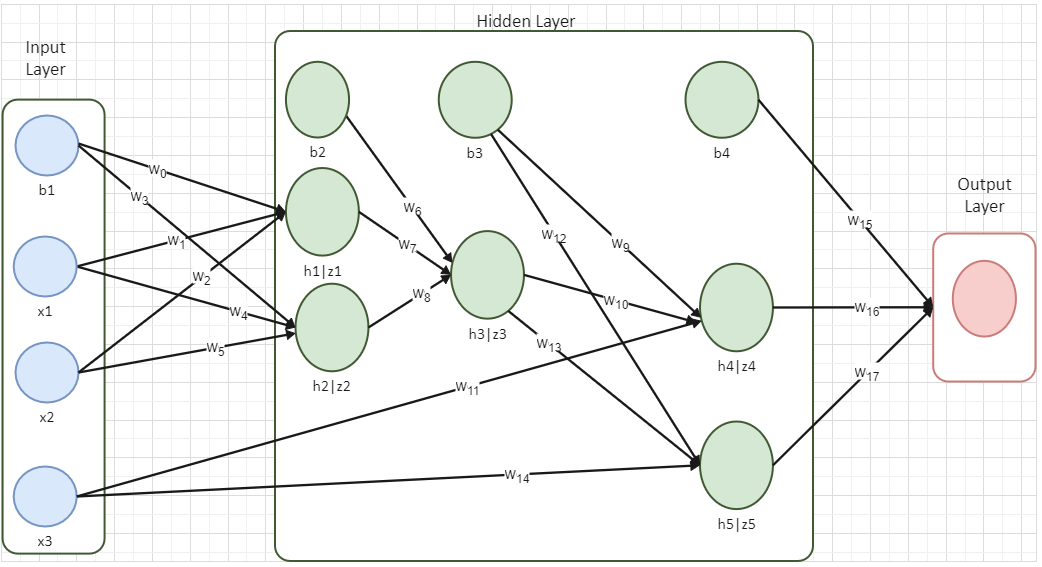
**Question 1 [10 marks]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Find out all the weight values (w0, w1, w2, ……., w17) for the given XOR neural network.

|  |  |  |  |
| --- | --- | --- | --- |
| x1 | x2 | x3 | Output |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

* hi represents the pre-activation value of a neuron. Example: h1=w1 . x1+ w2 . x2+w0 . b1
* zi represents the output of a neuron after applying the step activation function, defined as:
* z1 and z4 represent the NAND function, z2 and z5 represent the OR function, z3 and z6 (Output layer) represent the AND function.

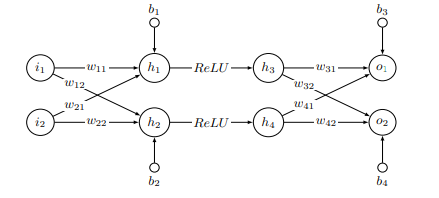


***CLO3 #: Understand the different layers and their operations.***

**Question 2 [3+2+5 marks]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

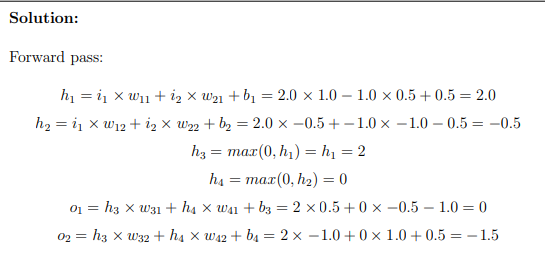
Given the following neural network with fully connection layer and ReLU activations, including two input units (i1, i2), four hidden units (h1, h2) and (h3, h4). The output units are indicated as (o1, o2) and their targets are indicated as (t1, t2). The weights and bias of fully connected layer are called w and b with specific sub-descriptors.



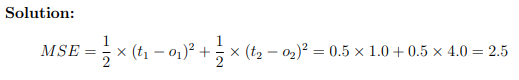
The values of variables are given in the following table:



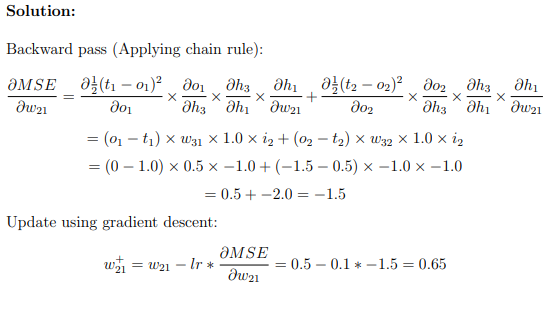
1. Compute the output (o1, o2) with the input (i1, i2) and network paramters as specified above. Write down all calculations, including intermediate layer results.



1. Compute the mean squared error of the output (o1, o2) calculated above and the target (t1, t2).



1. Update the weight w21 using gradient descent with learning rate 0.1 as well as the loss computed previously. (Please write down all your computations.)



***CLO1 #: Understand the theoretical foundations of deep learning, including neural networks and***

***optimization techniques.***

***CLO3 #: Understand the different layers and their operations.***

**Question 3:** Answer the following questions **[3+3+4 marks]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. A data scientist designs a neural network for a binary classification task to predict whether an email is spam or not. The network uses ReLU activation in all hidden layers and sigmoid activation in the output layer. After training, what potential issues could arise?

Dying ReLU Problem: ReLU outputs zero for negative inputs, making some neurons inactive. If many neurons "die" during training, the model loses learning capacity.

Vanishing Gradient Problem: The sigmoid function in the output layer produces small gradients for extreme values (near 0 or 1). This weakens backpropagation, slowing down learning or preventing convergence.

1. A machine learning engineer is training a deep neural network to classify handwritten digits. However, during training, the loss remains constant, and the weights do not update after each epoch. What could be the possible reason, and how can it be resolved?

**Answer:**

**Possible Reasons**

* 1. **Learning rate is set to 0**, preventing any weight updates.
  2. **Vanishing Gradient Problem**

**Resolution:**

1. Set an appropriate **learning rate**
2. Use **ReLU or Leaky ReLU** instead of sigmoid/tanh to prevent vanishing gradients.
3. Apply **Batch Normalization** to stabilize gradients and improve learning speed.
4. In a neural network, layer U sends outputs to layer V, with the weights W on the connections between U and V. Layer U employs a hyperbolic tangent (tanh) activation function, while layer V uses a sigmoid. Given that ui is the output of the ith neuron of U, vk is the output of the kth neuron of V, ui = 0.8, vk = 0.5, and the weight between those neurons is wi,k = 0.3:
5. What is the value of ∂vk /∂wi,k ?

0.8 \* 0.5 \* (1 - 0.5) = 0.2

1. What is the value of ∂vk /∂ui ?

0.3 \* 0.5 \* (1 - 0.5) = 0.075