Exercise 1: Hand-Crafted Network

Exercise 1: Hand-Crafted Network (15 Points)

In this exercise, we aim to construct neural networks that classify an arbitrary training set with zero training error. As preparation, we design single neurons (i.e., specify their weights, bias, and activation function) for the following tasks:

1. Logical OR Neuron

To design a neuron that performs the logical OR operation on a binary input vector $\mathbf{z} \in \{0,1\}^D$:

- **Weights:** $w_j = 1$ for all j.
- **Bias:** $b = 0$.
- **Activation Function:** Use a threshold function, which outputs 1 if the weighted sum of inputs is greater than or equal to 1, otherwise 0.

$$f(\tilde{\mathbf{z}}) = \begin{cases} 1 & \text{if } \sum_j \tilde{z}_j \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

2. Masked Logical OR Neuron

To design a neuron for the masked logical OR operation for an arbitrary but fixed binary vector $\mathbf{c} \in \{0,1\}^D$:

- **Weights:** $w_j = c_j$ for all j.
- **Bias:** $b = 0$.
- **Activation Function:** Use a threshold function, which outputs 1 if the weighted sum of inputs is greater than or equal to 1, otherwise 0.).

$$f(\tilde{\mathbf{z}}) = \begin{cases} 1 & \text{if } \sum_j \tilde{z}_j \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

3. Perfect Match Neuron

To design a neuron for the perfect match operation for an arbitrary but fixed binary vector $\mathbf{c} \in \{0,1\}^D$:

- **Weights:** $w_j = 1$ for all j.
- **Bias:** $b = -\sum_j c_j$.
- **Activation Function:** Use a threshold function, which outputs 1 if the sum of z_j and c_j are equal to each other.

$$f(\tilde{\mathbf{z}}) = \begin{cases} 1 & \text{if } \sum_j \tilde{z}_j = 0 \\ 0 & \text{otherwise} \end{cases}$$

2. Linear Activation Function (5 Points)

For a feed-forward network, the output of each layer l is calculated iteratively by:

$$Z_0 = X \tag{1}$$

$$\tilde{Z}_l = Z_{l-1} \cdot B_l + b_l \quad \text{(multiply with weights and add bias vector)} \tag{2}$$

$$Z_l = \phi_l(\tilde{Z}_l) \quad \text{(apply the activation function element-wise to pre-activations)} \tag{3}$$

Where X and the Z_l and b_l are understood as row vectors.

Prove that if ϕ_l is the identity function, then any network (with depth $L > 1$) is equivalent to a 1-layer neural network.

Proof

Given:

$$Z_0 = X \tag{1}$$

$$\tilde{Z}_l = Z_{l-1} \cdot B_l + b_l \tag{2}$$

$$Z_l = \phi_l(\tilde{Z}_l) \tag{3}$$

To Prove:

If ϕ_l is the identity function, then any network (with depth $L > 1$) is equivalent to a 1-layer neural network.

Step-by-Step Proof:

Initialization:

$$Z_0 = X$$

Layer 1:

Using Equations (2) and (3) for layer 1:

$$\tilde{Z}_1 = Z_0 \cdot B_1 + b_1$$

$$Z_1 = \phi_1(\tilde{Z}_1)$$

Since ϕ_1 is the identity function:

$$Z_1 = \tilde{Z}_1 = Z_0 \cdot B_1 + b_1 = X \cdot B_1 + b_1$$

Layer 2:

Using Equations (2) and (3) for layer 2:

$$\tilde{Z}_2 = Z_1 \cdot B_2 + b_2$$

$$Z_2 = \phi_2(\tilde{Z}_2)$$

Since ϕ_2 is the identity function:

$$Z_2 = \tilde{Z}_2 = Z_1 \cdot B_2 + b_2$$

Substitute Z_1 from the previous step:

$$Z_2 = (X \cdot B_1 + b_1) \cdot B_2 + b_2$$

Generalization to Layer l:

For any layer l:

$$\tilde{Z}_l = Z_{l-1} \cdot B_l + b_l$$
$$Z_l = \phi_l(\tilde{Z}_l)$$

Since ϕ_l is the identity function:

$$Z_l = \tilde{Z}_l = Z_{l-1} \cdot B_l + b_l$$

By induction, assume this holds for layer $l-1$:

$$Z_{l-1} = X \cdot (B_1 \cdot B_2 \cdot \ldots \cdot B_{l-1}) + (b_1 \cdot B_2 \cdot \ldots \cdot B_{l-1}) + \ldots + b_{l-1}$$

Then for layer l:

$$Z_l = (X \cdot (B_1 \cdot B_2 \cdot \ldots \cdot B_{l-1}) + (b_1 \cdot B_2 \cdot \ldots \cdot B_{l-1}) + \ldots + b_{l-1}) \cdot B_l + b_l$$

Simplifying the Final Output:

The final output Z_L can be simplified to:

$$Z_L = X \cdot (B_1 \cdot B_2 \cdot \ldots \cdot B_L) + b_{1,2,\ldots,L}$$

where $b_{1,2,\ldots,L}$ is a combined bias term:

$$b_{1,2,\ldots,L} = (b_1 \cdot B_2 \cdot \ldots \cdot B_L) + (b_2 \cdot B_3 \cdot \ldots \cdot B_L) + \ldots + b_L$$

This combined equation shows that the network can be represented as a single-layer network with:

- **Weights:** $W = B_1 \cdot B_2 \cdot \ldots \cdot B_L$
- **Bias:** $b = b_{1,2,\ldots,L}$

Conclusion:

If the activation functions ϕ_l are identity functions, then any deep network (with depth $L > 1$) can be represented as a single-layer network. This proves that a multi-layer network with identity activation functions is equivalent to a one-layer neural network.