Quiz-2 - DeepLearning

Question 1: Backpropagation by Hand (MLP) (20 marks)

Consider a simple feedforward neural network with the following architecture:

Input (2 units) $\rightarrow$ Hidden Layer (2 units, sigmoid) $\rightarrow$ Output Layer (1 unit, sigmoid).

You are given:

- Input vector:
$$x = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$$
- Target output:
$$y = 0.6$$
- Input-to-Hidden weight matrix and bias:
$$W^{(1)} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}, \quad b^{(1)} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$
- Hidden-to-Output weight matrix and bias:
$$W^{(2)} = \begin{bmatrix} 0.5 & 0.6 \end{bmatrix}, \quad b^{(2)} = 0.0$$
- Activation function:
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$
- Loss function:
$$E = \tfrac{1}{2}(y - \hat{y})^2$$

Tasks:

1. Perform a complete forward pass and compute the predicted output $\hat{y}$.
2. Compute the loss E.
3. Using backpropagation, compute the gradients $\nabla W^{(2)}, \nabla b^{(2)}, \nabla W^{(1)}, \nabla b^{(1)}$.
4. If the learning rate is $\eta = 0.5$, update all weights and biases.
5. Update all the weights for $\eta = 0.1$ and compare and comment on which learning rate is more suitable

—

Question 2: CNN Forward Pass (Programming Question) (20 Marks)

Consider a simple Convolutional Neural Network with the following configuration:

- Input: a 4×4 grayscale image

$$X = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

- One convolutional filter of size 3×3 (no bias):

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

- Convolution stride = 1, padding = 0 (valid convolution).
- Nonlinearity after convolution: ReLU

$$\text{ReLU}(z) = \max(0, z)$$

- Flatten the feature map into a vector.
- Fully connected (dense) layer with 2 outputs:

$$W^{(fc)} = \begin{bmatrix} 0.1 & -0.2 & 0.3 & 0.4 \\ -0.5 & 0.2 & 0.1 & -0.3 \end{bmatrix}, \quad b^{(fc)} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

  (Assume the flattened feature map is 4 dimensional after convolution for simplicity.)

- Softmax activation at output:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Tasks: Write a Python code to

1. Perform the 3×3 convolution of X with the filter K and write the resulting 2×2 feature map.
2. Apply ReLU activation.
3. Flatten the ReLU output into a vector.
4. Compute the output of the fully connected layer.
5. Apply the softmax function to obtain the predicted class probabilities.
6. How do you change your code if the input is RGB ?

Question 3: RNN (10 Marks)

Consider the first-order linear difference equation:

$$y_t = 0.5\, y_{t-1} + x_t, \quad t = 1, 2, 3, \ldots$$

where x_t is the input sequence and y_t is the output sequence.

1. Explain how this RNN can be used to model the difference equation above. (Hint: think about how the hidden state h_t can represent the role of y_{t-1}.)
2. Briefly explain how the hidden state h_t provides the "memory" needed to capture the recursive relation.