

Machine Learning (DS4023) Assignment 4

Problem 1: Neural Networks (20 pts)

Consider a 3-layer fully connected neural network with the following architecture:

- **Input layer:** $n = 4$ neurons.
- **Hidden layer:** $m = 3$ neurons using a custom activation function $f(x) = \text{ReLU}(x) + \sin(x)$.
- **Output layer:** $k = 2$ neurons using the softmax activation function $\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$.

The network parameters (**weights and biases**) are given as:

- $\mathbf{W}_1 \in \mathbb{R}^{3 \times 4}$ and $\mathbf{b}_1 \in \mathbb{R}^3$ for the hidden layer.
- $\mathbf{W}_2 \in \mathbb{R}^{2 \times 3}$ and $\mathbf{b}_2 \in \mathbb{R}^2$ for the output layer.

Given the input vector $\mathbf{x} \in \mathbb{R}^4$ and target output $\mathbf{y} \in \mathbb{R}^2$. Define the loss function as cross-entropy loss:

$$\text{Loss} = - \sum_{i=1}^k y_i \log(\hat{y}_i),$$

where \hat{y} is the output after the softmax activation.

Your tasks (rounding to 4 decimal points):

- 1) Derive the equations for the forward pass through the network, including both the hidden and output layers. (3 pts)
- 2) Calculate the outputs \mathbf{Z}_1 , \mathbf{H} , \mathbf{Z}_2 , and $\hat{\mathbf{y}}$ explicitly for a given input $\mathbf{x} = [1, -1, 0.5, 2]^T$ and the following initial weights and biases:

$$\mathbf{W}_1 = \begin{pmatrix} 0.1 & -0.2 & 0.3 & 0.4 \\ 0.5 & -0.3 & 0.1 & -0.2 \\ 0.4 & 0.2 & -0.5 & 0.3 \end{pmatrix}, \quad \mathbf{b}_1 = \begin{pmatrix} 0.1 \\ -0.1 \\ 0.05 \end{pmatrix}$$

$$\mathbf{W}_2 = \begin{pmatrix} -0.3 & 0.2 & 0.1 \\ 0.4 & -0.5 & 0.3 \end{pmatrix}, \quad \mathbf{b}_2 = \begin{pmatrix} 0.05 \\ -0.05 \end{pmatrix}.$$

Note that \mathbf{Z}_1 is the net input to the hidden layer, \mathbf{H} is the activation output of the hidden layer, and \mathbf{Z}_2 is the net input to the output layer. (3 pts)

2. Derive the gradient of the loss with respect to each parameter ($\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2$) in the network and obtain the gradient values using results from the first question. Use matrix calculus to express the gradients. Hint: You can first calculate the error terms δ_2 and δ_1 for each layer and use them to express the gradients. (10 pts)
3. Suppose the learning rate $\alpha = 0.001$. Please calculate the updated parameter values after one back propagation process. (4 pts)

Problem 2: Programming (80 pts)

Complete the jupyter notebook attached on programming for CNN and RNN. Submit the completed file.

To submit:

1. A file containing the written answer to the Problems 1.
2. The Jupyter notebook with solutions to Problem 2.