Machine Learning (DS4023) Assignment 4

Deadline: Nov. 25, 2024.

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:

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

Given the input vector $\boldsymbol{x} \in \mathbb{R}^4$ and target output $\boldsymbol{y} \in \mathbb{R}^2$. 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:

1. 1) Derive the equations for the forward pass through the network, including both the hidden and output layers. (3 pts)

   2) Calculate the outputs $\boldsymbol{Z}_1, \boldsymbol{H}, \boldsymbol{Z}_2$, and $\hat{\boldsymbol{y}}$ explicitly for a given input $\boldsymbol{x} = [1, -1, 0.5, 2]^T$ and the following initial weights and biases:

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

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

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

2. Derive the gradient of the loss with respect to each parameter $(\boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2)$ 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 $\boldsymbol{\delta}_2$ and $\boldsymbol{\delta}_1$ 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.

Solution (Full marks can be given if it is correct for the first three decimal points.)

1. The net input to the hidden layer, $\boldsymbol{Z}_1$, is calculated as:

$$\boldsymbol{Z}_1 = \boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1 \quad \textbf{(1 pt)}$$

Expanding the matrix-vector multiplication and addition:

$$\boldsymbol{Z}_1 = \begin{pmatrix} 0.1 & -0.2 & 0.3 & 0.4 \\ 0.5 & -0.3 & 0.1 & -0.2 \\ 0.4 & 0.2 & -0.5 & 0.3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0.5 \\ 2 \end{pmatrix} + \begin{pmatrix} 0.1 \\ -0.1 \\ 0.05 \end{pmatrix},$$

which is

$$\boldsymbol{Z}_1 = \begin{pmatrix} 1.25 \\ 0.45 \\ 0.55 \end{pmatrix} + \begin{pmatrix} 0.1 \\ -0.1 \\ 0.05 \end{pmatrix} = \begin{pmatrix} 1.35 \\ 0.35 \\ 0.6 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

The activation of the hidden layer $\boldsymbol{H}$ is computed by applying the custom activation function $f(x) = \mathrm{ReLU}(x) + \sin(x)$ to each element of $\boldsymbol{Z}_1$.

$$\boldsymbol{H} = f(\boldsymbol{Z}_1) = \begin{pmatrix} \text{ReLU}(1.35) + \sin(1.35) \\ \text{ReLU}(0.35) + \sin(0.35) \\ \text{ReLU}(0.6) + \sin(0.6) \end{pmatrix} = \begin{pmatrix} 2.3257 \\ 0.6929 \\ 1.1646 \end{pmatrix}. \quad \textbf{(1 pt)}$$

The net input to the output layer $\boldsymbol{Z}_2$ is calculated as:

$$\boldsymbol{Z}_2 = \boldsymbol{W}_2\boldsymbol{H} + \boldsymbol{b}_2 \quad \textbf{(1 pt)}$$

Expanding the matrix-vector multiplication and addition:

$$\boldsymbol{Z}_2 = \begin{pmatrix} -0.3 & 0.2 & 0.1 \\ 0.4 & -0.5 & 0.3 \end{pmatrix} \begin{pmatrix} 2.3257 \\ 0.6929 \\ 1.1646 \end{pmatrix} + \begin{pmatrix} 0.05 \\ -0.05 \end{pmatrix} = \begin{pmatrix} -0.3927 \\ 0.8832 \end{pmatrix} \quad \textbf{(0.5 pt)}$$

Applying the softmax function to $\boldsymbol{Z}_2$:

$$\hat{y}_i = \frac{e^{\boldsymbol{Z}_{2,i}}}{\sum_{j=1}^{2} e^{\boldsymbol{Z}_{2,j}}}, \quad \textbf{(1 pt)}$$

We have

$$\hat{\boldsymbol{y}}_1 = \frac{e^{-0.3927}}{e^{-0.3927} + e^{0.8832}} = \frac{0.6752}{0.6752 + 2.4182} = \frac{0.6752}{3.0934} \approx 0.2182,$$
$$\hat{\boldsymbol{y}}_2 = \frac{e^{0.8832}}{e^{-0.3927} + e^{0.8832}} = \frac{2.4182}{3.0934} \approx 0.7818.$$

Thus:

$$\hat{\boldsymbol{y}} = \begin{pmatrix} 0.2182 \\ 0.7818 \end{pmatrix} \quad \textbf{(1 pt)}$$

2. Gradient calculation

We will compute gradients with respect to $\boldsymbol{W}_2$, $\boldsymbol{b}_2$, $\boldsymbol{W}_1$, and $\boldsymbol{b}_1$ by applying the chain rule in backpropagation.

Gradient of Loss with Respect to $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$

Since we are using the softmax activation with cross-entropy loss, we have:

$$\boldsymbol{\delta}_2 = \hat{\boldsymbol{y}} - \boldsymbol{y}, \quad \textbf{(1 pt)}$$

where $\boldsymbol{\delta}_2$ represents the error at the output layer.

The gradient of the loss with respect to $\boldsymbol{W}_2$ is:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_2} = \boldsymbol{\delta}_2 H^T \quad \textbf{(1 pt)}$$

where H^T is the transpose of the hidden layer output.

The gradient of the loss with respect to $\boldsymbol{b}_2$ is:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{b}_2} = \boldsymbol{\delta}_2 \quad \textbf{(1 pt)}$$

These expressions are derived from the chain rule applied to the softmax and cross-entropy, making the derivations for $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$ correct.

Gradient of Loss with Respect to $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$

To find the gradients with respect to $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$, we need the hidden layer error term $\boldsymbol{\delta}_1$, which backpropagates from the output layer through the hidden layer.

Calculation of $\boldsymbol{\delta}_1$

The error term $\boldsymbol{\delta}_1$ for the hidden layer is influenced by the output layer error term $\boldsymbol{\delta}_2$ backpropagated through $\boldsymbol{W}_2$:

$$\boldsymbol{\delta}_1 = (\boldsymbol{W}_2^T \boldsymbol{\delta}_2) \odot f'(\boldsymbol{Z}_1) \quad \textbf{(1 pt)}$$

where $f'(\boldsymbol{Z}_1)$ represents the element-wise derivative of the activation function applied to $\boldsymbol{Z}_1$.

For the activation function $f(x) = \text{ReLU}(x) + \sin(x)$, the derivative $f'(x)$ is:

$$f'(x) = \begin{cases} 1 + \cos(x) & \text{if } x > 0 \\ \cos(x) & \text{if } x \leq 0 \end{cases} \quad \textbf{(1 pt)}$$

Thus, $f'(\boldsymbol{Z}_1)$ is applied element-wise to each component of $\boldsymbol{Z}_1$ to calculate $\boldsymbol{\delta}_1$.

Gradients with Respect to $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$

Using $\boldsymbol{\delta}_1$, the gradients for $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ are as follows:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_1} = \boldsymbol{\delta}_1 \boldsymbol{x}^T \quad \textbf{(1 pt)}$$

where $\boldsymbol{x}^T$ is the transpose of the input vector $\boldsymbol{x}$.

The gradient of the loss with respect to $\boldsymbol{b}_1$ is simply:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{b}_1} = \boldsymbol{\delta}_1 \quad \textbf{(1 pt)}$$

These gradients are derived by applying the chain rule from the hidden layer output $\boldsymbol{H}$ back to $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ through $\boldsymbol{Z}_1$, taking into account the influence of the custom activation function's derivative.

Summary of Gradient Expressions

Gradients for Output Layer:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_2} = \boldsymbol{\delta}_2 H^T$$

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{b}_2} = \boldsymbol{\delta}_2$$

Gradients for Hidden Layer:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_1} = \boldsymbol{\delta}_1 \boldsymbol{x}^T$$

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{b}_1} = \boldsymbol{\delta}_1$$

This completes the derivation process for the gradients of each parameter in the network, considering the custom activation function $f(x) = \text{ReLU}(x) + \sin(x)$ in the hidden layer.

Substitution:

Output layer error term $\boldsymbol{\delta}_2$:

$$\boldsymbol{\delta}_2 = \hat{\boldsymbol{y}} - \boldsymbol{y} = \begin{pmatrix} 0.2182 \\ 0.7818 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -0.7818 \\ 0.7818 \end{pmatrix}.$$

To compute $\boldsymbol{\delta}_1$, we need $\boldsymbol{W}_2^T \boldsymbol{\delta}_2$ and the derivative of the activation function, $F'(\boldsymbol{Z}_1)$.

$$\boldsymbol{W}_2^T = \begin{pmatrix} -0.3 & 0.4 \\ 0.2 & -0.5 \\ 0.1 & 0.3 \end{pmatrix}$$

$$\boldsymbol{W}_2^T \boldsymbol{\delta}_2 = \begin{pmatrix} -0.3 & 0.4 \\ 0.2 & -0.5 \\ 0.1 & 0.3 \end{pmatrix} \begin{pmatrix} -0.7818 \\ 0.7818 \end{pmatrix} = \begin{pmatrix} 0.5473 \\ -0.5473 \\ 0.1716 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

For $F'(\boldsymbol{Z}_1)$:

$$F'(\boldsymbol{Z}_1) = \begin{pmatrix} 1 + \cos(1.35) \\ 1 + \cos(0.35) \\ 1 + \cos(0.6) \end{pmatrix} \approx \begin{pmatrix} 1.219 \\ 1.9394 \\ 1.8253 \end{pmatrix}.$$

Thus:

$$\boldsymbol{\delta}_1 = (\boldsymbol{W}_2^T \boldsymbol{\delta}_2) \odot F'(\boldsymbol{Z}_1) = \begin{pmatrix} 0.5473 \\ -0.5473 \\ 0.1546 \end{pmatrix} \odot \begin{pmatrix} 1.219 \\ 1.9394 \\ 1.8253 \end{pmatrix} = \begin{pmatrix} 0.6672 \\ -1.0614 \\ 0.2855 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

The gradient of the loss with respect to $\boldsymbol{W}_2$ is:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_2} = \delta_2 H^T = \begin{pmatrix} -0.7818 \\ 0.7818 \end{pmatrix} \begin{pmatrix} 2.3257 & 0.6929 & 1.1646 \end{pmatrix}.$$

Calculating each term:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_2} = \begin{pmatrix} -1.8182 & -0.5417 & -0.9105 \\ 1.8182 & 0.5417 & 0.9105 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

The gradient of the loss with respect to $\boldsymbol{b}_2$ is:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{b}_2} = \boldsymbol{\delta}_2 = \begin{pmatrix} -0.7818 \\ 0.7818 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

The gradient of the loss with respect to $\boldsymbol{W}_1$ is:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_1} = \boldsymbol{\delta}_1 \boldsymbol{x}^T = \begin{pmatrix} 0.6672 \\ -1.0614 \\ 0.2855 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0.5 & 2 \end{pmatrix}.$$

Calculating each term:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{W}_1} = \begin{pmatrix} 0.6672 & -0.6672 & 0.3336 & 1.3344 \\ -1.0614 & 1.0614 & -0.5307 & -2.1228 \\ 0.2855 & -0.2855 & 0.1428 & 0.5710 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

The gradient of the loss with respect to $\boldsymbol{b}_1$ is:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{b}_1} = \boldsymbol{\delta}_1 = \begin{pmatrix} 0.6672 \\ -1.0614 \\ 0.2855 \end{pmatrix}. \quad \textbf{(0.5 pt)}$$

3. Using the update rule:

$$\text{New parameter} = \text{Old parameter} - \alpha \times \text{Gradient}.$$

Updated $\boldsymbol{W}_2$:

$$\boldsymbol{W}_2' = \boldsymbol{W}_2 - 0.001 \times \frac{\partial \text{Loss}}{\partial \boldsymbol{W}_2}$$

$$\boldsymbol{W}_2' \approx \begin{pmatrix} -0.2982 & 0.2005 & 0.1009 \\ 0.3982 & -0.5005 & 0.2991 \end{pmatrix}. \quad \textbf{(1 pt)}$$

Updated $\boldsymbol{b}_2$:

$$\boldsymbol{b}_2' \approx \begin{pmatrix} 0.0508 \\ -0.0508 \end{pmatrix}. \quad \textbf{(1 pt)}$$

Updated $\boldsymbol{W}_1$:

$$\boldsymbol{W}_1' \approx \begin{pmatrix} 0.0993 & -0.1993 & 0.2997 & 0.3987 \\ 0.5011 & -0.3011 & 0.1005 & -0.1979 \\ 0.3997 & 0.2003 & -0.5001 & 0.2994 \end{pmatrix}. \quad \textbf{(1 pt)}$$

Updated $\boldsymbol{b}_1$:

$$\boldsymbol{b}_1' \approx \begin{pmatrix} 0.0993 \\ -0.0989 \\ 0.0497 \end{pmatrix}. \quad \textbf{(1 pt)}$$