Machine Learning Course 2024 Spring: Homework 3

April 28, 2024

1 Problem 1

Consider the following multi-layer neural network (Figure 1) which includes an input layer, one hidden layer, and an output layer, containing d, n, q neurons respectively. The parameters between the input layer and the hidden layer are $\boldsymbol{W}_1 \in \mathbb{R}^{d\times n}$, $\boldsymbol{b}_1 \in \mathbb{R}^n$, and the parameters between the hidden layer and the output layer are $\boldsymbol{W}_2 \in \mathbb{R}^{n\times q}$, $\boldsymbol{b}_2 \in \mathbb{R}^q$. Where $\boldsymbol{W}_1$,$\boldsymbol{W}_2$ are the weight matrices and $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ are the bias vectors.

Let us first compute the forward propagation. Let $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]^\top \in \mathbb{R}^d$ be the input. The hidden layer is computed as follows:

$$\boldsymbol{h} = \boldsymbol{W}_1^\top \boldsymbol{x} + \boldsymbol{b}_1 \in \mathbb{R}^n \tag{1.1}$$

Then the ReLU activation function is applied to Eq.(1.1):

$$\boldsymbol{a} = \mathrm{ReLU}(\boldsymbol{h}) \in \mathbb{R}^n \tag{1.2}$$

The output layers' activation is obtained using the following transformation

$$\boldsymbol{z} = \boldsymbol{W}_2^\top \boldsymbol{a} + \boldsymbol{b}_2 \in \mathbb{R}^q \tag{1.3}$$

Finally, the soft-max function is applied to Eq.(1.3) to obtain the probability for each category.

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_q \end{bmatrix} = \mathrm{Softmax}(\boldsymbol{z}) = \begin{bmatrix} \mathrm{Softmax}\,(z_1) \\ \mathrm{Softmax}\,(z_2) \\ \vdots \\ \mathrm{Softmax}\,(z_q) \end{bmatrix} = \begin{bmatrix} \frac{\exp(z_1)}{\sum_{k=1}^q \exp(z_k)} \\ \frac{\exp(z_2)}{\sum_{k=1}^q \exp(z_k)} \\ \vdots \\ \frac{\exp(z_q)}{\sum_{k=1}^q \exp(z_k)} \end{bmatrix} \in \mathbb{R}^q \tag{1.4}$$

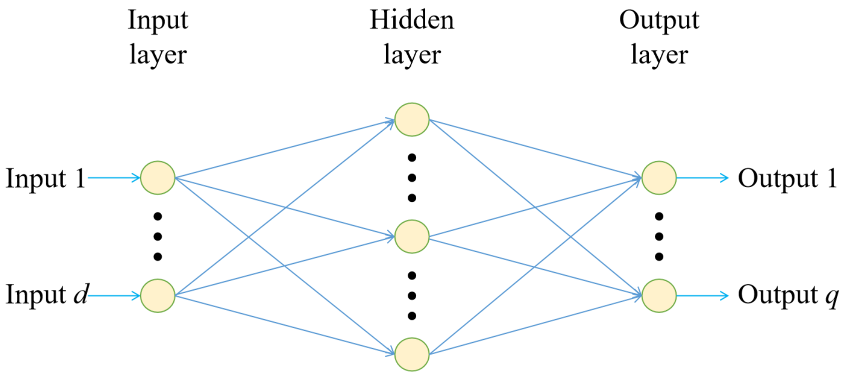

Figure 1: Neural Network

Here, $\hat{y}_i, z_i$ is the i-th element of $\hat{y}$, z, and $\hat{y}$ is the predicted output by the feed-forward neural network.

Your task is to compute the derivatives of the Cross-Entropy loss function given in Eq.(1.5) with respect to $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ by hand, i.e.,

$$\begin{aligned} &\frac{\partial \text{ Loss}}{\partial \boldsymbol{W}_1}, \frac{\partial \text{ Loss}}{\partial \boldsymbol{W}_2}, \frac{\partial \text{ Loss}}{\partial \boldsymbol{b}_1}, \frac{\partial \text{ Loss}}{\partial \boldsymbol{b}_2}. \\ \text{Loss} &= -\sum_{i=1}^{q} y_i^s \ln\left(\hat{y}_i\right). \end{aligned} \tag{1.5}$$

Here, $y_i^s = (1-\epsilon)y_i + \frac{\epsilon}{q}$ is the soft label via label smoothing, where ϵ is a small constant. $y_i \in \{0, 1\}$ is the ground-truth indicator, $\hat{y}_i$ is the i-th element of $\hat{\boldsymbol{y}}$.

Show all the intermediate derivative computation steps. You might benefit from making a rough schematic of the back-propagation process.

2 Problem 2

Consider a simple practical case with only one input instance: $\boldsymbol{x} = [0, 2]^\top \in \mathbb{R}^2, \boldsymbol{y} = [1, 0, 0]^\top \in \mathbb{R}^3$; $d = 2$, $n = 3$ and $q = 3$. The initial values of the parameters are listed below:

$$\boldsymbol{W}_1 = \begin{bmatrix} -5 & 2 & 2 \\ -5 & 2 & -1 \end{bmatrix} \in \mathbb{R}^{2\times 3}, \quad \boldsymbol{b}_1 = \begin{bmatrix} -3 \\ -2 \\ -3 \end{bmatrix} \in \mathbb{R}^3,$$

$$\boldsymbol{W}_2 = \begin{bmatrix} 3 & -4 & 1 \\ -2 & -4 & -2 \\ -4 & -2 & 3 \end{bmatrix} \in \mathbb{R}^{3\times 3}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 4 \\ 5 \\ 5 \end{bmatrix} \in \mathbb{R}^3. \tag{2.1}$$

Your task is to compute the updated value of the parameters $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, and $\boldsymbol{b}_2$ after one step of Gradient Descent ($\theta^{t+1} \leftarrow \theta^t - \eta \cdot \nabla_\theta \text{Loss}$) for $\epsilon = 0.3$ and $\eta = 0.1$.

Use the intermediate computation you derived from Problem 1. You might benefit from performing the feed-forward process first and then the back-propagation process. You can use a calculator or write code to do the calculations.