Assignment 4

EE 468: Selected Topics on Communications and Signal Processing
Neural Networks and Deep Learning
Due on: May 20$^{\text{th}}$, 2022

Theoretical Part

Problem 1: Backpropagation

We have learned in Lecture 5 how to perform backpropagation to compute the gradients of any MLP network weight, and we derived the gradient for two weight matrices in different layers. In this problem, you will be task with deriving the gradient for two bias vectors in the same network example we have in Lecture 5—and depicted in Figure.1 for convenience.

More specifically, let

$$\mathbf{b}^{(2)} = \begin{bmatrix} b_1^{(2)} \\ b_2^{(2)} \end{bmatrix} \tag{1}$$

$$\mathbf{b}^{(1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \end{bmatrix} \tag{2}$$

be the bias vectors of layers 2 and 1, respectively. Assume that thee activation functions of the first layer are ReLU, and the activation functions of the second layer are linear (i.e., no activation). Further assume that the loss function is the MSE loss give as:

$$\mathcal{L} = ||\mathbf{t} - \hat{\mathbf{y}}||_2^2, \tag{3}$$

where $\hat{\mathbf{y}} = [y_1, y_2]^T$ and $\mathbf{t} = [t_1, t_2]^T$ is the desired response (i.e., groundtruth) vector. Use the chain rule of derivatives to:

1. derive the gradient of $\mathcal{L}$ with respect to each vector, i.e., $\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(1)}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(2)}}$. You derivation should be in matrix multiplication format (your final equation should look something like that in Equation 1.7 or 1.14 in Lecture 5).

2. knowing the loss function and the activations, differentiate each factor in (1) above and multiply them to get the formula for the gradient vectors $\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(1)}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(2)}}$. You may use the following derivative for the ReLU activation

$$\frac{\partial}{\partial x}\text{ReLU}(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}, \quad \text{for } x \in \mathbb{R} \tag{4}$$

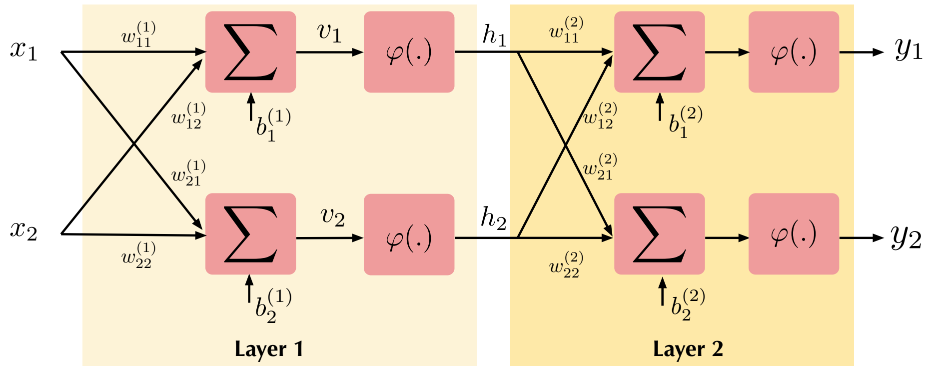

Figure 1: *Schematic for the MPL network for which bias gradient should be computed*

Coding Part

In Assignment 3, you got to code an MLP network from scratch and load pre-trained parameters to test it. A natural extension to that is to develop a full neural network, i.e., construct, train, and test the network. That means we need to code a backward pass for every layer with its weight update rule, and code the back-propagation algorithm. This is quite challenging, especially for complex networks such as CNNs. Instead, we will take advantage of highly-optimized, well-coded, and open-source deep learning software frameworks to develop neural networks. In this assignment, you learn a bout one of the most popular frameworks that is PyTorch, and you will do so by developing a CNN for an image classification problem using the popular CIFAR-10 dataset.

I Network Architecture

You are required to develop a CNN with two convolutional blocks and two dense blocks. The choice of number of channels and kernel size are left to you to design. Just make sure the convolutional layers do not spatially downsample the input (i.e., use appropriate padding), and only downsample using max-pooling layers. The equation that governs the height of the output is

$$H_{out} = \left\lfloor \frac{H_{in} - (k-1) + 2p - 1}{s} + 1 \right\rfloor, \tag{5}$$

where k is the kernel size, p is the padding, s is the stride, and they are all along the height dimension. For the width, the equation is the same except that all parameters, k, p, and s, are taken along the width.

II Dataset

We will rely on the CIFAR-10 dataset, which has 50K images for training and 10K images for testing. The 10 in the name refers to the total number of classes in the dataset. One good thing about the dataset is that it provide low-resolution images, 32×32, and hence, it does not require a lot of storage space or processing time.

III What Should You Do?

Visit the course GitHub website at: `https://github.com/ModernMLCourse`. Go to the repo "Assignment_4," and download it. You should get the following

- Network development script files: (1) main.py, (2) build_net.py, (3) algo_train.py, and (4) data_feed.py
- Dataset preparation script files: (1) get_data.py and (2) utils.py.
- A "README.md" file with some instructions.

The dataset is not provided here, for you are required to download it from the CIFAR website and prepare it using the preparation scripts yourself. Please, read the README.md file carefully to understand what you are expected to do, and make sure you review all the notes and comments provided in all script files.

> **REMARK:** you are required to understand the whole script and not only those lines you complete; moving forward, I will assume you are familiar with everything you have seen in an assignment.