Midterm Exam I
Introduction to Deep Learning
ECE 685D Fall 2022

Instructor: Vahid Tarokh
ECE Department, Duke University
19 Oct. 2022
10:15-11:30 AM (exam duration **75 minutes**)

You are not allowed to communicate with others.

Name	
Duke ID	

The total number of points on the exam is 110 (10 of which are **bonus** points)

Problem 1 (30)		Problem 2 (40)			Problem 3 (40)		
1(15)	2(15)	1(20)	2(5)	3(15)	1(15)	2(10)	3(15)
Total:		Total:			Total:		
Grand Total:							

Problem 1

Consider an RGB image given by the following matrix:

$$X[:,:,0] =$$

10	0	0	0	0	0	20	0	0
0	0	10	0	0	0	0	0	10
0	0	0	0	10	0	20	0	20
20	0	10	0	20	0	0	0	0
0	0	0	0	10	0	20	0	10
10	0	0	0	0	0	10	0	10
0	0	20	0	10	0	10	0	10
10	0	0	0	20	0	20	0	20
20	0	0	0	0	0	0	0	0

$$X[:,:,1] =$$

0	0	0	20	0	0	0	10	0
0	0	0	0	0	20	0	0	0
0	20	0	10	0	20	0	0	0
0	0	0	20	0	0	0	20	0
0	10	0	0	0	0	0	0	0
0	0	0	20	0	0	0	0	0
0	10	0	0	0	0	0	0	0
0	10	0	0	0	0	0	20	0
0	0	0	20	0	10	0	10	0

$$X[:,:,2] = X[:,:,1]$$

The filter k_1 is given by the following $3 \times 3 \times 3 \times 1$ tensor:

$$k_1[0,:,:,0] = k_1[1,:,:,0] = k_1[2,:,:,0] = \begin{bmatrix} 0 & 0.1 & 0 \\ -0.1 & 0 & -0.1 \\ 0.1 & 0 & 0 \end{bmatrix}$$

The filter k_2 is given by a $1 \times 1 \times 1 \times 2$ tensor:

$$k_2[0,:,:,0] = \begin{bmatrix} 0.5 \end{bmatrix}$$
$$k_2[0,:,:,1] = \begin{bmatrix} -0.5 \end{bmatrix}$$

The output of the first convolutional layer is given by $Y_1 = \text{ReLU}(X * k_1 + \mathbb{1}_{1\times4\times4})$, where $*$ is convolution with **stride being 2 in both directions**, and $\mathbb{1}_{1\times4\times4}$ is a $1 \times 4 \times 4$ matrix with all ones.

The output of the second convolutional layer $Z_1 * k_2$ is performed with the **stride being 1**.

1. (15 pts) Compute the output of the first convolutional layer Y_1. Recall from the lecture notes the convolution between a 4D filter k and a 3D input w is defined as $(w * k)_{fij} = \sum_c \sum_{p,q} w_{c,i+p,j+q} \cdot k_{c,p,q,f}$, where c is the channel index, p, q are the location indices, f is the output channel index.

2. (15 pts) Apply maxpooling on non-overlapping 2×2 sub-matrices of the filtered image and compute the output Z_1 as $Z_1 = \text{maxpool}(Y_1)$. Then calculate $Y_2 = \text{ReLU}(Z_1 * k_2 + 0.5 \times \mathbb{1}_{2\times2\times2})$. Lastly, calculate the output $Z_2 = \text{maxpool}(Y_2)$, with the maxpool applied on non-overlapping 2×2 sub-matrices.

Solution:

This page is intentionally left blank

This page is intentionally left blank

Problem 2

(40 pts) Consider a binary logistic regression problem as follows

$$p_i = p(y_i = 1|\mathbf{x_i}) = \sigma(\mathbf{w}^T \mathbf{x}_i + b),\ \forall i \in \{1, \ldots, 4\}$$

where $y \in \{1, 0\}$, $\mathbf{x} \in \mathbb{R}^{2\times 1}$, $\mathbf{w} \in \mathbb{R}^{2\times 1}$, $b \in \mathbb{R}$, and $\sigma(\cdot)$ is the sigmoid function, given as:

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

We are given a dataset with four data points $\{\mathbf{x}_1, y_1\} = \{(1, 1)^T, 1\}$, $\{\mathbf{x}_2, y_2\} = \{(1, 2)^T, 1\}$, $\{\mathbf{x}_3, y_3\} = \{(-2, -1)^T, 0\}$, $\{\mathbf{x}_4, y_4\} = \{(-3, -3)^T, 0\}$.

The loss function is

$$\mathcal{L} = \sum_{i=1}^{4} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

The initial value of $\mathbf{w}$ is $\mathbf{w}_1 = (1, 1)^T$.

The initial value of b is $b_1 = 0.5$.

Perform 2 steps of Nesterov's accelerated gradient descent method with $\beta = 0.9$ on $\mathbf{w}$ and b using the dataset formed with four data points $\{\mathbf{x}_i, y_i\}_{i=1}^{4}$. **Use the definition of Nesterov's Accelerated Gradient Descent in the lecture notes, as shown below.**

Algorithm 1 Nesterov's Accelerated Gradient Descent

First define the following sequences: $\lambda_0 = 0,\ \lambda_k = (1 + \sqrt{1 + 4\lambda_{k-1}^2})/2,\ \gamma_k = (1 - \lambda_k)/\lambda_{k+1}$
for $k = 1, 2, \ldots$ **do**
  $\mathbf{t_{k+1}} = \mathbf{w_k} - \nabla\mathcal{L}(\mathbf{w_k})/\beta$
  $\mathbf{w_{k+1}} = (1 - \gamma_k)\mathbf{t_{k+1}} + \gamma_k \mathbf{t_k}$
end for

1. (20 pts) Write out the gradient of the loss function $\mathcal{L}$ with respect to the weight $\mathbf{w}$ and bias b explicitly.
2. (5 pts) Write out $\lambda_0,\ \lambda_1,\ \lambda_2,\ \lambda_3$ and $\gamma_1,\ \gamma_2$ explicitly.
3. (15 pts) Calculate $t_2,\ t_3$ and $\mathbf{w}_2,\ \mathbf{w}_3$, b_2, and b_3.

Solution:

This page is intentionally left blank

This page is intentionally left blank

Problem 3

Figure 2 depicts a simple, fully connected multi-layer perceptron network with one hidden layer. The inputs to the network are x_1, x_2, the output is y. The activation functions of the neurons in the hidden layer are given as $h_1(z) = z^2$, $h_2(z) = z^2$, and the output unit activation function is $h_3(z) = \sigma(z)$, where $\sigma(\cdot)$ is the sigmoid function and $\sigma(z) = \frac{1}{1+e^{-z}}$. The bias b is added to the output of the hidden layer before passing it to the output layer. Let $\mathbf{w} = (W_{1,1}^{(1)}, W_{1,2}^{(1)}, W_{2,1}^{(1)}, W_{2,2}^{(1)}, W_1^{(2)}, W_2^{(2)})$ denote the vector of the network parameters.

1. (15 pts) Let $\mathcal{D} = \{(x_{1,i}, x_{2,i}), y_i\}_{i=1}^N$ denote a training dataset of N points where $y_n \in \{0, 1\}$ are the labels of the corresponding data points. We want to estimate the network parameters $\mathbf{w}$ using $\mathcal{D}$ by minimizing the Mean Square Error $E(\mathbf{w})$.
   Compute the gradient with respect to the network parameters $\mathbf{w}$, you must use **the backpropagation algorithm**. Specifically, write out explicit formulas for $\frac{\partial y}{\partial W_{1,1}^{(1)}}, \frac{\partial y}{\partial W_{1,2}^{(1)}}, \frac{\partial y}{\partial W_{2,1}^{(1)}}, \frac{\partial y}{\partial W_{2,2}^{(1)}}, \frac{\partial y}{\partial W_1^{(2)}}, \frac{\partial y}{\partial W_2^{(2)}}$.
2. (10 pts) Calculate the gradient for a initial weight vector of $\mathbf{w} = (W_{1,1}^{(1)} = 1, W_{1,2}^{(1)} = 1, W_{2,1}^{(1)} = 2, W_{2,2}^{(1)} = 1, W_1^{(2)} = 0.1, W_2^{(2)} = 3)$, and with the data point $\{(x_1, x_2), y\} = \{(1, 2), 1\}$.
3. (15 pts) Update $\mathbf{w}$ once using the stochastic gradient descent algorithm, the gradient in the last question, the data point $\{(x_1, x_2), y\} = \{(1, 2), 1\}$, and a learning rate of $\eta = 0.001$.

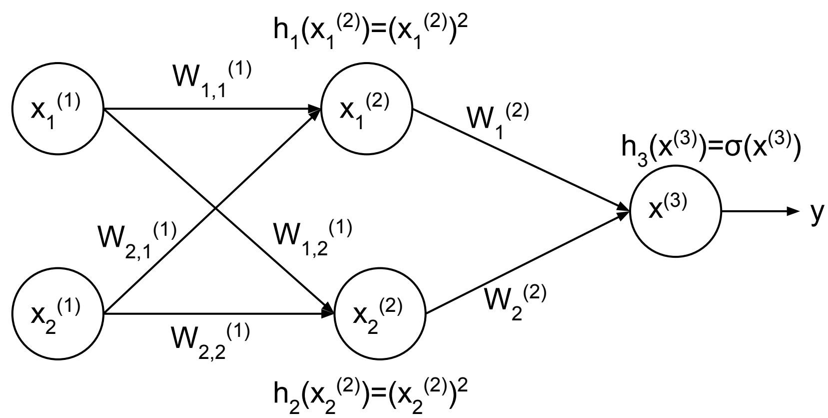

Figure 1: Schematic of the MLP network.

Solution:

This page is intentionally left blank

This page is intentionally left blank