

Midterm Exam I
ECE 685D—Introduction to Deep Learning
Fall 2024

Instructor: Prof. Vahid Tarokh
ECE Department, Duke University

Oct 9th 2024
10:05 AM - 11:20 AM

Name: _____
Duke ID: _____

This exam contains 10 pages and 10 questions. This is a closed-book exam. No exam aids are permitted except for a one-sided letter-sized cheat sheet. Communication with others is strictly prohibited.

Distribution of Marks

Question	Points	Score
1	3	
2	3	
3	3	
4	3	
5	3	
6	3	
7	5	
8	25	
9	35	
10	17	
Total:	100	

Definition of Leaky ReLU function. Given the non-negative coefficient α (i.e., $\alpha < 1$) and the input x , the leaky ReLU is defined as:

$$\text{LeakyReLU}(x, \alpha) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}$$

For the multiple-choice questions, circle ALL of the correct answers. These questions can have more than one correct answer.

1. (3 points) After training a convolutional neural network, you observe a large gap between the training accuracy (e.g., 99%) and the test accuracy (e.g., 50%). Which of the following methods is commonly used to resolve this issue?

- (a) Dropout
- (b) Leaky ReLU
- (c) Sigmoid activation

Answer: (a)

2. (3 points) Which of the following activation functions can lead to vanishing gradients?

- (a) Sigmoid
- (b) Leaky ReLU
- (c) Tanh

Answer: (a)(c)

3. (3 points) Which of the following is true about Batchnorm

- (a) Batchnorm is another way of performing dropout.
- (b) Batchnorm makes the training process converge faster.
- (c) Batchnorm is a non-linear transformation to center the dataset around the origin.

Answer: (b)

4. (3 points) If the size of the input is $64 \times 64 \times 16$ (i.e., number of channels = 16), how many parameters are there, including bias, in a convolution filter with kernel size=1, stride=1, padding=0?

- (a) 1
- (b) 2
- (c) 17

(d) 4097

Answer: (c)

5. (3 points) Consider an input of dimensions (n_h, n_w, n_c) , where n_c is the number of channels. A convolutional layer with kernel size=1, stride=1, padding=0 is applied to this input. After the convolution, a standard max-pooling layer (e.g., stride=2) is applied. Which of the following statements about the output after both operations is correct?

- (a) The convolution can help reduce the dimension of n_h, n_w , but not n_c .
- (b) The convolution can help reduce the dimension of n_c , but not n_h, n_w .
- (c) The standard maxpooling can help reduce the dimension of n_h, n_w , but not n_c .
- (d) The standard maxpooling can help reduce the dimension of n_c , but not n_h, n_w .

Answer: (b)(c)

6. (3 points) Recall that the L_p norm of $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ is defined as:

$$\|\mathbf{w}\|_p = \left(\sum_i^n |w_i|^p \right)^{1/p}.$$

Which of the following regularization metrics is convex and most likely leads to weight sparsity?

- (a) $L_{1/2}$
- (b) L_1
- (c) L_2
- (d) L_∞

Answer: (b)

7. (5 points) What are the advantages of using convolutional layers instead of fully connected layers for visual tasks (e.g., image classification, object detection)? Explain your answer.

Answer: The advantages of convolutional layers: shared parameters, fewer number of parameters, reduced computation, spatial information, translation invariance

8. Consider an image X with 2 channels. The image is represented as a $4 \times 4 \times 2$ matrix where the last dimension correspond to the number of channel of this image. We denote a kernel W with dimension $3 \times 3 \times 2$. The values of the image pixels and weights of the kernel are given below.

$$X[:, :, 0] = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 3 \end{bmatrix} \quad W[:, :, 0] = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$$

$$X[:, :, 1] = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix} \quad W[:, :, 1] = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

The output of the convolutional filter is given as:

$$Y = \text{LeakyReLU}(X' * W - \mathbf{1}_{4 \times 4}, 0.1), \quad (1)$$

where Y is the output image, X' is the input image after applying zero-padding around the edges (i.e. each channel is converted to a 6×6 matrix such that a row of zeros is added to the top and bottom and a column of zeros is added to the left and right.), $X' * W$ is the convolution between X' and W with stride size=1, and $\mathbf{1}_{4 \times 4}$ is a 4×4 matrix with elements equal to one. Here, α is the coefficient of the LeakyReLU function, where the leaky ReLU function is defined as:

$$\text{LeakyReLU}(x, \alpha) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha x & \text{if } x < 0. \end{cases}$$

- (a) (15 points) Compute the output Y of the image X .
- (b) (5 points) Apply max pooling on non-overlapping 2×2 sub-matrices of the output image and compute the output.
- (c) (5 points) Apply average pooling on non-overlapping 2×2 sub-matrices of the output image and compute the output.

Answer: (a)

$$X'[:, :, 0] * W[:, :, 0] = \begin{bmatrix} -1 & 2 & -2 & -2 \\ 5 & -3 & 2 & 0 \\ 2 & 3 & 2 & 0 \\ 0 & 0 & 1 & -3 \end{bmatrix} \quad X'[:, :, 1] * W[:, :, 1] = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & -1 \end{bmatrix}$$

$$X' * W - \mathbf{1}_{4 \times 4} = \begin{bmatrix} -1 & 1 & -1 & -2 \\ 4 & -5 & 1 & -1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 2 & -5 \end{bmatrix} \quad Y = \begin{bmatrix} -0.1 & 1 & -0.1 & -0.2 \\ 4 & -0.5 & 1 & -0.1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 2 & -0.5 \end{bmatrix}$$

(b)

$$\text{maxpool}(Y) = \begin{bmatrix} 4 & 1 \\ 2 & 2 \end{bmatrix}$$

(c)

$$\text{avgpool}(Y) = \begin{bmatrix} 1.1 & 0.15 \\ 1 & 1.125 \end{bmatrix}$$

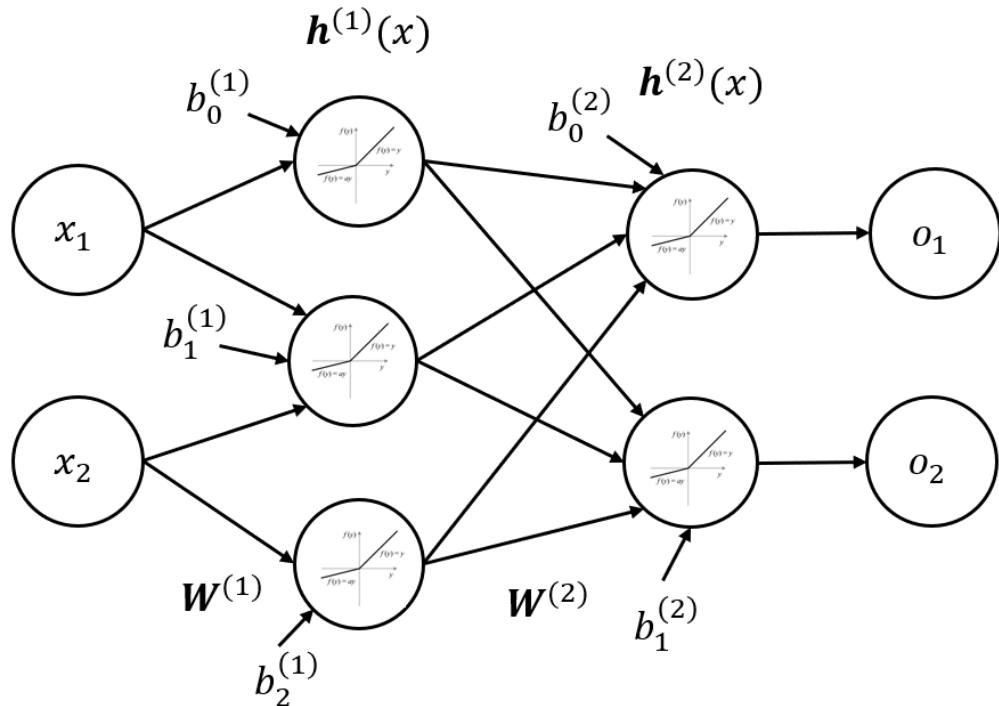
9. Given the neural network with $L = 2$ hidden layers, the input units $\mathbf{x} = [x_1, x_2]^T$, and LeakyReLU activations (as shown below). The weights and bias of hidden units are denoted \mathbf{W} and \mathbf{b} . The hidden layers are defined as below:

$$\mathbf{h}^{(0)}(\mathbf{x}) = \mathbf{x} = [x_1, x_2]^T$$

$$\mathbf{h}^{(i)}(\mathbf{x}) = \sigma(\mathbf{b}^{(i)} + \mathbf{W}^{(i)}\mathbf{h}^{(i-1)}(\mathbf{x})), \text{ for } i = \{1, 2\}$$

where σ is the LeakyReLU activation with the coefficient $\alpha = 0.5$. The output $\mathbf{o} = [o_1, o_2]^T$ is defined as:

$$\mathbf{o}(\mathbf{x}) = \mathbf{h}^{(2)}(\mathbf{x})$$



The values of the inputs are $\mathbf{x} = [1, 2]^T$. The ground truth values of the output are $\mathbf{t} = [4, 1]^T$. The weights of the network are given as follows:

$$\mathbf{W}^{(1)} = \begin{bmatrix} -0.5 & 0 \\ 0.5 & 1 \\ 0 & 1 \end{bmatrix} \quad \mathbf{W}^{(2)} = \begin{bmatrix} 0.5 & 2 & 0.5 \\ 1 & -0.5 & -2 \end{bmatrix}$$

$$\mathbf{b}^{(1)} = [2.5 \quad -0.5 \quad 1]^T \quad \mathbf{b}^{(2)} = [-5 \quad -1]^T$$

- (a) (10 points) Compute the output $[o_1, o_2]^T$ of the input $[x_1, x_2]^T$ using the network parameters as specified above. Write down all calculations of the intermediate layers.

- (b) (5 points) Compute the mean squared error (MSE) between the output $[o_1, o_2]^T$ and the target ground truth $[t_1, t_2]^T$.
- (c) (5 points) Using the calculated MSE, update the weight $W_{0,1}^{(2)}$ (i.e., entry $\{0, 1\}$ of matrix $\mathbf{W}^{(2)}$) using gradient descent and the backpropagation algorithm with the learning rate of 0.1. You do not need to simplify your answer.
- (d) (5 points) Similarly, update the weight $W_{1,1}^{(2)}$ (i.e., entry $\{1, 1\}$ of matrix $\mathbf{W}^{(2)}$) using gradient descent and the backpropagation algorithm with the learning rate of 0.1. You do not need to simplify your answer.
- (e) (10 points) Update the weight $W_{1,0}^{(1)}$ (i.e., entry $\{1, 0\}$ of matrix $\mathbf{W}^{(1)}$) using gradient descent and the backpropagation algorithm with a learning rate of 0.1. Write down the details of the backward pass. You do not need to simplify your answer.

Answer: (a)

$$h^{(1)}(x) = \sigma(b^{(1)} + W^{(1)}h^{(0)}(x)) = [2, 2, 3]^T \quad (2)$$

$$o(x) = h^{(2)}(x) = \sigma(b^{(2)} + W^{(2)}h^{(1)}(x)) = [1.5, -3]^T \quad (3)$$

(b)

$$L_{MSE} = \frac{1}{2} \sum_{i=1}^2 2(t_i - o_i)^2 = 0.5(2.5^2 + 4^2) = 11.125 \quad (4)$$

(c)

$$\frac{\partial L_{MSE}}{\partial W_{0,1}^{(2)}} = \frac{\partial L_{MSE}}{\partial o_1} \frac{\partial o_1}{\partial W^{(2)_0}} \frac{\partial W^{(2)_0}}{\partial W_{0,1}^{(2)}} = (o_1 - t_1) * 1 * h_1^{(1)} = -5 \quad (5)$$

$$w_{0,1}^{(2)} = w_{0,1}^{(2)} - 0.1 * (-5) = 2.5 \quad (6)$$

(d)

$$\frac{\partial L_{MSE}}{\partial W_{1,1}^{(2)}} = \frac{\partial L_{MSE}}{\partial o_2} \frac{\partial o_2}{\partial W^{(2)_1}} \frac{\partial W^{(2)_1}}{\partial W_{1,1}^{(2)}} = (o_2 - t_2) * 0.5 * h_1^{(1)} = -4 \quad (7)$$

$$w_{1,1}^{(2)} = w_{1,1}^{(2)} - 0.1 * (-4) = -0.1 \quad (8)$$

(d)

$$\frac{\partial L_{MSE}}{\partial W_{1,0}^{(1)}} = \frac{\partial L_{MSE}}{\partial h_1} \frac{\partial h_1}{\partial W_{1,0}^{(1)}} = \left(\frac{\partial L_{MSE}}{\partial w_0^{(2)}} \frac{\partial w_0^{(2)}}{\partial h_1^{(1)}} + \frac{\partial L_{MSE}}{\partial w_1^{(2)}} \frac{\partial w_1^{(2)}}{\partial h_1^{(1)}} \right) \frac{\partial h_1}{\partial W_{1,0}^{(1)}} = -4 \quad (9)$$

$$w_{1,0}^{(1)} = w_{1,0}^{(1)} - 0.1 * (-4) = 0.9 \quad (10)$$

10. The L_1 regularization of the model parameter $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ is defined as:

$$\|\mathbf{w}\|_1 = \sum_i^n |w_i|.$$

Below, we apply L_1 regularization to a neural network model, of which the original objective function is defined as $J(\mathbf{w}; X, y)$, where \mathbf{w} is the model weights, X is the input, and y is the ground truth labels. The regularized objective function after adding the L_1 normalization term becomes:

$$J_R(\mathbf{w}; X, y) = J(\mathbf{w}; X, y) + \alpha \|\mathbf{w}\|_1.$$

- (a) (2 points) Write the gradient $\partial J_R / \partial \mathbf{w}$ in terms of $\partial J / \partial \mathbf{w}$.
- (b) (15 points) Finding the closed form solution for the root $\mathbf{w}_R = [w_{R1}, w_{R2}, \dots, w_{Rn}]^T$ of $\partial J_R / \partial \mathbf{w} = 0$ is difficult. Hence, we apply Taylor expansion to approximate $J_R(\mathbf{w}; X, y)$, and discard the high-order terms:

$$\hat{J}(\mathbf{w}; X, y) \approx J(\mathbf{w}^*) + \frac{1}{2} (\mathbf{w} - \mathbf{w}^*)^T H (\mathbf{w} - \mathbf{w}^*),$$

where H is the Hessian matrix, and \mathbf{w}^* is the solution of $\partial J / \partial \mathbf{w} = 0$ (i.e., the optimal parameter for the objective function J without the regularization term). Suppose that it is known that H is a diagonal matrix with $H = \text{diag}([H_{1,1}, \dots, H_{n,n}])$, where $H_{i,i} > 0, \forall i \in n$. Write down a closed form expression for the root of \mathbf{w}_R .

Answer: (a)

$$\partial J_R / \partial w = \partial J / \partial w + \alpha sign(w) \quad (11)$$

(b)

$$J_R(w) = J(w^*) + 0.5(w - w^*)^T H(w - w^*) + \alpha \|w\|_1 \quad (12)$$

$$\partial J_R / \partial w = \partial J(w^*) / \partial w + 0.5 \partial(w - w^*)^T H(w - w^*) / \partial w + \alpha sign(w) \quad (13)$$

$$w_i = sign(w_i^*) max(|w_i^* - \alpha/H_{i,i}|, 0) \quad (14)$$

Hence,

$$w_i = \begin{cases} w_i^* + \alpha/H_{i,i} & \text{for } w_i^* \leq -\alpha/H_{i,i} \\ 0 & \text{for } \alpha/H_{i,i} > w_i^* > -\alpha/H_{i,i} \\ w_i^* - \alpha/H_{i,i} & \text{for } w_i^* \geq -\alpha/H_{i,i} \end{cases} \quad (15)$$