Solutions to the Midterm Exam
Introduction to Deep Learning
ECE 685D Fall 2021

Instructor: Vahid Tarokh
ECE Department, Duke University
11 Oct. 2021
10:15-11:30 am (exam duration **75 minutes**)

You are not allowed to communicate with others.

Name	
Duke ID	

Problem 1 (35)		Problem 2 (30)		Problem 3 (35)		
1.1 (20)	1.2 (15)	2.1 (15)	2.2 (25)	3.1 (5)	3.2 (20)	3.3 (10)
Total:		Total:		Total:		
Grand Total:						

Problem 1

Consider an image $\mathbf{X}$ with three channels. The image is represented as a $4 \times 4 \times 3$ tensor, where the last dimension corresponds the channels of the image. Let $\mathbf{W}$ denote a filter of size $3 \times 3 \times 3$. The image and the filter are given as follows:

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

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

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

Consider the following convolutional layer:

$$\mathbf{Y} = \mathrm{ReLU}\left(\mathbf{1}_{4\times 4} + \sum_{i=0}^{2} \tilde{\mathbf{X}}[:,:,i] * \mathbf{W}[:,:,i]\right),$$

where $\mathbf{Y}$ is the output image, $\tilde{\mathbf{X}}$ 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.), $\tilde{\mathbf{X}}[:,:,i] * \mathbf{W}[:,:,i]$ is the **convolution** of the i-th channel of $\tilde{\mathbf{X}}$ with the the i-th channel of $\mathbf{W}$, and $\mathbf{1}_{4\times 4}$ is a 4×4 matrix with all ones.

1. (20) Compute the output image.
2. (15) Apply max pooling on non-overlapping 2×2 sub-matrices of the output image and compute the output.

Solution:

1. Compute the output image.

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

$$\tilde{\mathbf{X}}[:,:,1] * \mathbf{W}[:,:,1] = \begin{bmatrix} 2 & 0 & 0 & 1 \\ -1 & -1 & -6 & -3 \\ -2 & 1 & 0 & -2 \\ 0 & 0 & 0 & -3 \end{bmatrix}$$

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

$$\sum_{i=0}^{2} \tilde{\mathbf{X}}[:,:,i] * \mathbf{W}[:,:,i] = \begin{bmatrix} 6 & 1 & 0 & 2 \\ -2 & -1 & -4 & -3 \\ -1 & 5 & -1 & -3 \\ 2 & 0 & -1 & -3 \end{bmatrix}$$

$$\mathbf{1}_{4\times 4} + \sum_{i=0}^{2} \tilde{\mathbf{X}}[:,:,i] * \mathbf{W}[:,:,i] = \begin{bmatrix} 7 & 2 & 1 & 3 \\ -1 & 0 & -3 & -2 \\ 0 & 6 & 0 & -2 \\ 3 & 1 & 0 & -2 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 7 & 2 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{bmatrix}$$

2. Apply max pooling on non-overlapping 2×2 sub-matrices of the output image and compute the output. The output is

$$\begin{bmatrix} 7 & 3 \\ 6 & 0 \end{bmatrix}$$

Problem 2

Let $x \in \mathbb{R}$ denote a random variable with the following cumulative distribution function (CDF):

$$F\left(x\right) = \exp\left(-\exp\left(-\frac{x-\mu}{\beta}\right)\right),$$

where μ and $\beta > 0$ denote the location and scale parameters, respectively. Let $\mathcal{D} = \{x_1, \ldots, x_n\}$ be a set of n independent and identically distributed (i.i.d.) observations of x.

1. (15) Write an equation for a cost function $L(\mu, \beta|\mathcal{D})$ whose minimization gives the maximum likelihood estimates for μ and β.
2. (15) Compute the derivatives of $L(\mu, \beta|\mathcal{D})$ with respect to μ and β and write a system of equations whose solution gives the maximum likelihood estimates of μ and β.

Solution:

1. By definition, the likelihood function is given by the joint probability density function (PDF) of $x_1, \ldots, x_N$:

$$\ell(\mu, \beta|\mathcal{D}) = f(x_1, \ldots, x_N; \mu, \beta) = \prod_{i=1}^{N} f(x_i; \mu, \beta), \tag{1}$$

where the product is due to the independence of the observations. The PDF $f(x; \mu, \beta)$ is given as

$$f(x; \mu, \beta) = \frac{dF(x)}{dx} = \frac{1}{\beta} \exp\left\{-\left(\frac{x-\mu}{\beta} + \exp\left(-\frac{x-\mu}{\beta}\right)\right)\right\}. \tag{2}$$

Replacing (2) in (1), yields

$$\begin{aligned}\ell(\mu, \beta|\mathcal{D}) &= \prod_{i=1}^{N} \frac{1}{\beta} \exp\left\{-\left(\frac{x-\mu}{\beta} + \exp\left(-\frac{x-\mu}{\beta}\right)\right)\right\} \\ &= \frac{1}{\beta^N} \exp\left\{-\left(\frac{\sum_{i=1}^{N} x_i - N\mu}{\beta} + \sum_{i=1}^{N} \exp\left(-\frac{x_i-\mu}{\beta}\right)\right)\right\}. \end{aligned} \tag{3}$$

Instead of maximizing the likelihood function (3) with respect to μ and β directly, it is easier to minimize the negative log-likelihood, defined as

$$L(\mu, \beta|\mathcal{D}) = -\ln \ell(\mu, \beta|\mathcal{D}) = N \ln \beta + \frac{\sum_{i=1}^{N} x_i - N\mu}{\beta} + \sum_{i=1}^{N} \exp\left(-\frac{x_i-\mu}{\beta}\right). \tag{4}$$

2. The derivatives of $\mathcal{L}(\mu, \beta|\mathcal{D})$ with respect to μ and β can be computed as follows:

$$\frac{\partial \mathcal{L}(\mu, \beta|\mathcal{D})}{\partial \mu} = -\frac{N}{\beta} + \frac{1}{\beta} \sum_{i=1}^{N} \exp\left(-\frac{x_i-\mu}{\beta}\right), \tag{5}$$

$$\frac{\partial \mathcal{L}(\mu, \beta|\mathcal{D})}{\partial \beta} = \frac{N}{\beta} - \frac{\sum_{i=1}^{N} x_i - N\mu}{\beta^2} + \frac{1}{\beta^2} \sum_{i=1}^{N} (x_i - \mu) \exp\left(-\frac{x_i-\mu}{\beta}\right). \tag{6}$$

The maximum likelihood estimates of μ and β are defined as the solution to the following system of equations:

$$\frac{\partial \mathcal{L}(\mu, \beta|\mathcal{D})}{\partial \mu} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}(\mu, \beta|\mathcal{D})}{\partial \beta} = 0.$$

Hence, the system of equations whose solution gives the maximum likelihood estimates of μ and β can be written as:

$$\begin{aligned}&\sum_{i=1}^{N} \exp\left(-\frac{x_i-\mu}{\beta}\right) - N = 0, \\ &N(\beta + \mu) - \sum_{i=1}^{N} x_i + \sum_{i=1}^{N} (x_i - \mu) \exp\left(-\frac{x_i-\mu}{\beta}\right) = 0.\end{aligned}$$

Problem 3

Fig. 1 depicts a simple neural network with one hidden layer. The inputs to the network are denoted by x_1, x_2 and x_3 and the output is denoted by y. The activation functions of the neurons in the hidden layer are given by $h_1(z) = \sigma(z)$, $h_2(z) = \tanh(z)$ and the output unit activation function is $g(z) = z$, where $\sigma(z) = \frac{1}{1+\exp(-z)}$ and $\tanh(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$ are the logistic sigmoid and hyperbolic tangent, respectively. The biases b_1 and b_2 are added to the inputs of the neurons in the hidden layer before passing them through the activation functions. Let $\mathbf{w} = (b_1, b_2, w_{1,1}^{(1)}, w_{1,2}^{(1)}, w_{2,1}^{(1)}, w_{3,1}^{(1)}, w_{3,2}^{(1)}, w_1^{(2)}, w_2^{(2)})$ denote the vector of the network parameters.

1. (5) Write the input-output relation $y = f(x_1, x_2, x_3; \mathbf{w})$ in explicit form.
2. (20) Let $\mathcal{D} = \{(x_{1,n}, x_{2,n}, x_{3,n}), y_n\}, n = 1, \ldots, N$ denote a training data set of N points where $y_n \in \mathbb{R}$ are the labels of the corresponding data points. We want to estimate the network parameters $\mathbf{w}$ using $\mathcal{D}$ by minimizing the mean squared error loss:
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(f(x_{1,n}, x_{2,n}, x_{3,n}; \mathbf{w}) - y_n\right)^2.$$
Compute the gradient of $E(\mathbf{w})$ with respect to the network parameters $\mathbf{w}$.
3. (10) Write a pseudo-code for **one iteration only** for minimizing $E(\mathbf{w})$ with respect to the network parameters $\mathbf{w}$ using stochastic gradient descent with a learning rate $\eta > 0$.

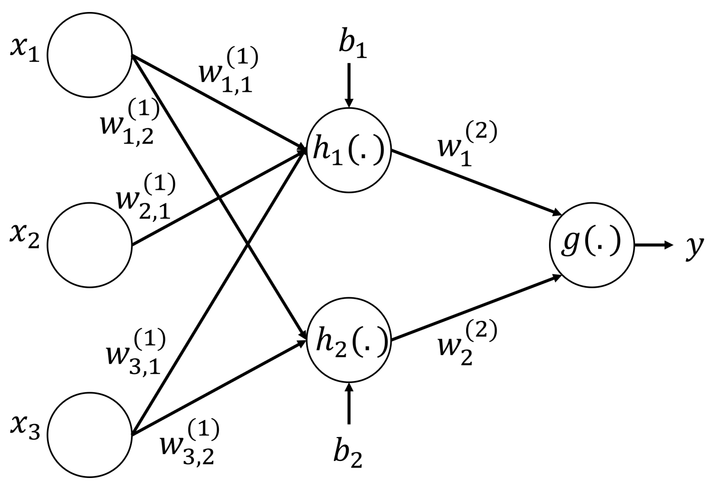

Figure 1: Neural network with one hidden layer

Solution:

1. The inputs to the neurons in the hidden layer are

$$\text{first neuron: } a_1^{(1)} = w_{1,1}^{(1)}x_1 + w_{2,1}^{(1)}x_2 + w_{3,1}^{(1)}x_3 + b_1,$$
$$\text{second neuron: } a_2^{(1)} = w_{1,2}^{(1)}x_1 + w_{3,2}^{(1)}x_3 + b_2.$$

The input to the neuron in the output layer is

$$\begin{aligned} a^{(2)} &= w_1^{(2)}h_1(a_1^{(1)}) + w_2^{(2)}h_2(a_2^{(1)}) \\ &= w_1^{(2)}\sigma(w_{1,1}^{(1)}x_1 + w_{2,1}^{(1)}x_2 + w_{3,1}^{(1)}x_3 + b_1) + w_2^{(2)}\tanh(w_{1,2}^{(1)}x_1 + w_{3,2}^{(1)}x_3 + b_2) \end{aligned}$$

Finally, the output of the neural neural network can be written as

$$\begin{aligned} y &= f(x_1, x_2, x_3; \mathbf{w}) = g(a^{(2)}) = a^{(2)} \\ &= w_1^{(2)}\sigma(a_1^{(1)}) + w_2^{(2)}\tanh(a_2^{(1)}) \\ &= w_1^{(2)}\sigma(w_{1,1}^{(1)}x_1 + w_{2,1}^{(1)}x_2 + w_{3,1}^{(1)}x_3 + b_1) + w_2^{(2)}\tanh(w_{1,2}^{(1)}x_1 + w_{3,2}^{(1)}x_3 + b_2). \end{aligned}$$

2. The gradient of $E(\mathbf{w})$ with respect to the network parameters can be simply written as

$$\begin{aligned} \nabla_{\mathbf{w}} E(\mathbf{w}) &= \nabla_{\mathbf{w}} \left(\frac{1}{2} \sum_{n=1}^{N} \left(f(x_{1,n}, x_{2,n}, x_{3,n}; \mathbf{w}) - y_n\right)^2 \right) \\ &= \frac{1}{2} \sum_{n=1}^{N} \nabla_{\mathbf{w}} \left(f(x_{1,n}, x_{2,n}, x_{3,n}; \mathbf{w}) - y_n\right)^2 \\ &= \sum_{n=1}^{N} (f_n - y_n) \nabla_{\mathbf{w}} f_n, \end{aligned}$$

where we denoted $f(x_{1,n}, x_{2,n}, x_{3,n}; \mathbf{w}) = f_n$ for brevity. Considering that

$$\nabla_{\mathbf{w}} f_n = \nabla_{\mathbf{w}} f|_{x_1 = x_{1,n}, x_2 = x_{2,n}, x_3 = x_{3,n}},$$

as well as

$$\nabla_{\mathbf{w}} f = \left(\frac{\partial f}{\partial b_1}, \frac{\partial f}{\partial b_2}, \frac{\partial f}{\partial w_{1,1}^{(1)}}, \frac{\partial f}{\partial w_{1,2}^{(1)}}, \frac{\partial f}{\partial w_{2,1}^{(1)}}, \frac{\partial f}{\partial w_{3,1}^{(1)}}, \frac{\partial f}{\partial w_{3,2}^{(1)}}, \frac{\partial f}{\partial w_1^{(2)}}, \frac{\partial f}{\partial w_2^{(2)}} \right)^\top,$$

it suffices to derive the partial derivatives of f with respect to the network parameters. This is done through successive application of the chain rule, yielding the following results:

$$\frac{\partial f}{\partial b_1} = w_1^{(2)}\sigma(a_1^{(1)})(1-\sigma(a_1^{(1)})),$$
$$\frac{\partial f}{\partial b_2} = w_2^{(2)}(1-\tanh^2(a_2^{(1)})),$$
$$\frac{\partial f}{\partial w_{1,1}^{(1)}} = w_1^{(2)}\sigma(a_1^{(1)})(1-\sigma(a_1^{(1)}))x_1,$$
$$\frac{\partial f}{\partial w_{1,2}^{(1)}} = w_2^{(2)}(1-\tanh^2(a_2^{(1)}))x_1,$$
$$\frac{\partial f}{\partial w_{2,1}^{(1)}} = w_1^{(2)}\sigma(a_1^{(1)})(1-\sigma(a_1^{(1)}))x_2,$$
$$\frac{\partial f}{\partial w_{3,1}^{(1)}} = w_1^{(2)}\sigma(a_1^{(1)})(1-\sigma(a_1^{(1)}))x_3,$$
$$\frac{\partial f}{\partial w_{3,2}^{(1)}} = w_2^{(2)}(1-\tanh^2(a_2^{(1)}))x_3,$$
$$\frac{\partial f}{\partial w_1^{(2)}} = \sigma(a_1^{(1)}),$$
$$\frac{\partial f}{\partial w_2^{(2)}} = \tanh(a_2^{(1)}).$$

3. Let $\mathbf{w}^{(0)}$ denote the initialization of the network parameters $\mathbf{w}$. The iterative update rule for minimizing $E(\mathbf{w})$ with respect to $\mathbf{w}$ using stochastic gradient descent with step-size η is

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta(f_{i^*} - y_{i^*})\nabla_{\mathbf{w}} f_{i^*},$$

where i^* denotes the index of a randomly chosen data point from $\mathcal{D}$ at step τ.