BME/ECE 695 Deep Learning
Midterm I Solution
March 4, Spring 2022

Q1.
2 Points

Rules: I understand that this is an open book exam that shall be done within the allotted time of 120 minutes. I can use my notes, and web resources. However, I will not communicate with any other person other than the official exam proctors during the exam, and I will not seek or accept help from any other persons other than the official proctors.

Upload a scan of your signature here:

Name: (2 pt) ____________________________________

Q2 Convex Minimization
(30 Points)

Consider the two convolutional neural networks given by

$$\hat{x} = f_w(y) = w * y \ ,$$

and

$$\hat{x} = g_w(y) = f_w(f_w(y)) \ ,$$

where $y \in \Re^{N \times N}$ is a 2D image input, and $w * y$ denotes 2D convolution with the kernel $w \in \Re^{p \times p}$ filter kernel with $p << N$ using a **valid** boundary condition.
The associated MSE loss functions for a single training pair (x, y) are then given by

$$\text{loss}_f(w) = \|x - f_w(y)\|^2$$

and

$$\text{loss}_g(w) = \|x - g_w(y)\|^2 \ .$$

Q2.1
Is $f_w(y)$ a linear function of w? Justify your answer, i.e., prove the result or give a counter example.
Q2.2
Is $\text{loss}_f(w)$ a convex function of w? Justify your answer, i.e., prove the result or give a counter example.
Q2.3
Will gradient descent optimization of $\text{loss}_f(w)$ converge to a global minimum? Justify your answer.
Q2.4
Is $g_w(y)$ always a linear function of w? Justify your answer, i.e., prove the result or give a counter example.
Q2.5
Is $\text{loss}_g(w)$ always a convex function of w? Justify your answer, i.e., prove the result or give a counter example.
Q2.6
Will gradient descent optimization of $\text{loss}_g(w)$ always converge to a global minimum? Justify your answer.

Solution: Q2.1

$f_w(y)$ is a linear function of w because convolution is linear. More specifically, let $a, b \in \Re$, $w_1, w_2 \in \Re^2$ be two different convolutional kernels, then

$$\begin{aligned} f_{aw_1+bw_2}(y)_{j_1,j_2} &= \sum_{k_1}\sum_{k_2}(aw_1+bw_2)_{k_1,k_2} y_{j_1-k_1,j_2-k_2} \\ &= a\sum_{k_1}\sum_{k_2} w_{1k_1,k_2} y_{j_1-k_1,j_2-k_2} + b\sum_{k_1}\sum_{k_2} w_{2k_1,k_2} y_{j_1-k_1,j_2-k_2} \\ &= af_{w_1}(y)_{j_1,j_2} + bf_{w_2}(y)_{j_1,j_2} \end{aligned}$$

Q2.2

$\text{loss}_f(w)$ is a convex function of w.

Justification: Let $\hat{x} = Aw$ be a linear function of w, and $d(\hat{x})$ be a convex function of $\hat{x}$. then we know from the properties of convex functions taught in class that $d(Aw)$ is a convex function of w.

Therefore, since $\hat{x} = f_w(y)$ is a linear function of w, and the MSE loss function, $\|x - \hat{x}\|^2$, is a convex function of $\hat{x}$, then we know that $\text{loss}_f(w) = \|x - f_w(y)\|^2$ is a convex function of w.

Q2.3

Gradient descent optimization of $\text{loss}_f(w)$ will converge to a global minimum, because gradient descent will converge to a local minimum, and every local minimum of a convex function is a global minimum.

Q2.4

The function $g_w(y)$ is **not** always a linear function of w.

Justification: In order to prove that this is not always true, we only need to find a counter example. Consider the case where $w_{j_1,j_2} = w_0\delta_{j_1,j_2}$. Then

$$g_w(y) = w_0^2\ .y$$

is not a linear function of the one parameter w_0.

Q2.5

$\text{loss}_g(w)$ is **not** necessarily a convex function of w.

Justification: Since $g_w(y) = fw(fw(y))$ is not generally a linear function of w, then $\text{loss}_g(w) = \|x - g_w(y)\|^2$ is not generally a convex function of w.

We can prove that it is not always convex by giving a counter example. Let $x = 1, y = 1$ and $w_{j_1,j_2} = w_0\delta_{j_1,j_2}$, then we have that

$$\text{loss}_g(w) = (1 - w_0^2)^2 = (1 - w_o)^2(1 + w_0)^2\ .$$

This is clearly not a convex function with minimum at both 1 and -1.

Q2.6

Gradient descent optimization of $\text{loss}_g(w)$ will not necessarily converge to a global minimum because $\text{loss}_g(w)$ is not in general a convex function of w.

Q3 Maximum Likelihood Estimation
(30 Points)

Consider the machine learning algorithm with inference function f_θ and parameters $\theta \in \Re^p$ so that

$$\hat{p} = f_\theta(y) \ ,$$

where $y \in \Re^N$ with distribution $p(y)$, and $\hat{p} \in S_M$ where S_M is the M-D simplex set defined by

$$S_M = \{p \in [0,1]^M : p_m \geq 0 \text{ and } \sum_{m=0}^{M-1} p_m = 1\} \ .$$

Intuitively, our goal is to train the ML function, f_θ, so the it accurately estimates, $\hat{p}_m = [f_\theta(y)]_m$, the probability that y has class m.
In order to do this, we would like to compute the maximum likelihood estimate (MLE) of the parameter θ given the single training pair (x, y) where $x \in \{0,1\}^M$ and $y \in \Re^N$, i.e., x uses 1-hot encoding.

Q3.1
Write out an expression for the joint probability $p_\theta(x, y)$.

Q3.2
Write out an expression for the negative log likelihood, $L(\theta) = -\log p_\theta(x, y)$.

Q3.3
Assume that you now have K independent training samples, $(x_k, y_k)_{k=0}^{K-1}$, then write out an expression for the joint probability $p_\theta(x, y)$.

Q3.4
Assume that you now have K independent training samples, then write out an expression for the negative log likelihood, $L(\theta) = -\log p_\theta(x, y)$.

Q3.5
If you would like to compute the MLE estimate of θ using the K training samples, then what loss function should you use? Justify your answer.

Q3.6
Give at least one **advantage** and one **disadvantage** of the MLE estimate.

Solution:
Q3.1
Let $p_\theta(x|y)$ denote the conditional probability of $x \in S_M$ given $y \in \Re^N$ where x uses 1-hot encoding. So only one element $x_m^* = 1$, and the rest are 0.
First notice that since x has one-hot encoding,

$$p_\theta(x|y) = \hat{p}_{m^*} = \prod_{m=0}^{M-1} (\hat{p}_m)^{x_m} = \prod_{m=0}^{M-1} ([f_\theta(y)]_m)^{x_m}$$

So then we have that

$$\begin{aligned} p_\theta(x,y) &= p(x|y)p(y) \\ &= p(y) \prod_{m=0}^{M-1} ([f_\theta(y)]_m)^{x_m} \quad . \end{aligned}$$

Q3.2

$$\begin{aligned} L(\theta) &= -\log p_\theta(x,y) \\ &= -\log p(y) - \sum_{m=0}^{M-1} x_m \log\left([f_\theta(y)]_m\right) \end{aligned}$$

Q3.3

$$\begin{aligned} p_\theta(x,y) &= \prod_{k=0}^{K-1} p_\theta(x_k, y_k) \\ &= \prod_{k=0}^{K-1} \left\{ p(y_k) \prod_{m=0}^{M-1} ([f_\theta(y_k)]_m)^{x_{k,m}} \right\} \end{aligned}$$

where $x_{k,m}$ denotes the m^{th} element of the k^{th} training sample.
Q3.4

$$\begin{aligned} L(\theta) &= -\sum_{k=0}^{K-1} \left\{ \log p(y_k) + \sum_{m=0}^{M-1} x_{k,m} \log\left([f_\theta(y_k)]_m\right) \right\} \\ &= -\sum_{k=0}^{K-1} \log p(y_k) - \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} x_{k,m} \log\left([f_\theta(y_k)]_m\right) \end{aligned}$$

Q3.5
In order to compute the maximum likelihood estimate for this problem, one should use the cross-entropy loss function. This is because in the answer to question Q3.4 above, the negative log likelihood is equal to the cross-entropy loss plus a constant.
Q3.6
Advantage:

1. Good choice when there's plenty of training data, or the prior distribution is unknown or hard to model.

2. Mostly unbiased.

Disadvantage:

1. Needs lots of training data to get good results.

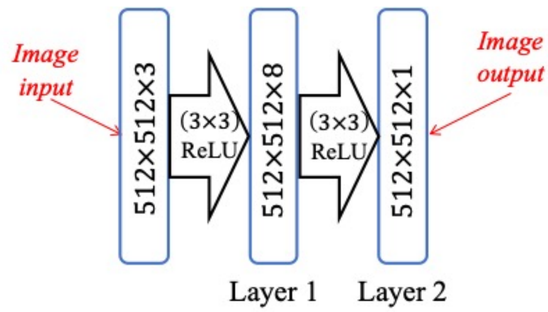
2. Overfits when there is no sufficient training data.

3. Does not exploit the knowledge of prior distribution of the data.

Q4 Convolutional Neural Networks

(10 Points)

Consider the following convolutional neural network pictured below with a color image as input, and a gray-scale image as output. Each layer uses a ReLu activation function, and denote the convolution kernel by w and the offsets by b.



Q4.1

For layer 1, what are the shapes of the tensors w and b?

Q4.2

For layer 2, what are the shapes of the tensors w and b?

Solution:

Q4.1

w: $3 \times 3 \times 3 \times 8$, b: 8

Q4.2

w: $3 \times 3 \times 8 \times 1$, b: 1

Q5 Adjoint Gradients
(10 Points)
Consider a single layer 1D convolutional neural network of the form

$$\hat{x} = f_w(y) = w * y \ ,$$

where $y \in \Re^5$ is a five component rank 1 vector, $w = [w_0, w_1, w_2]$, and $w * y$ denotes **true** convolution using a **valid** boundary condition.
Assume the CNN is trained using a single training pair (x, y) and that the MSE loss function is given by

$$\text{loss}(w) = \|x - f_w(y)\|^2 \ .$$

Q5.1
Determine the matrix A so that

$$Ay = [\nabla_y f_w(y)] y \ .$$

Q5.2
Determine an expression for the multiplication by the adjoint gradient given by

$$g_y = [\nabla_y f_w(y)]^t \, \epsilon \ .$$

Q5.3
Express g_y using the convolution operator and specify the required boundary condition.
Q5.4
Determine the matrix B so that

$$Bw = [\nabla_w f_w(y)] w \ .$$

Q5.5
Determine an expressions for the matrix C and the row vector ϵ so that

$$g_w = C\epsilon = \nabla_w \text{loss}(w) \ .$$

Q5.6
Express g_w using the convolution operator and specify the required boundary condition.

Solution:
Q5.1
We know that $\hat{x}_i = w_i * y_i$, so we have that

$$A_{i,j} = w_{2+i-j} \ ,$$

where the offset of $2 = 3 - 1$ is used due to the valid boundary condition. Then we have that

$$A = \begin{bmatrix} w_2 & w_1 & w_0 & 0 & 0 \\ 0 & w_2 & w_1 & w_0 & 0 \\ 0 & 0 & w_2 & w_1 & w_0 \end{bmatrix}$$

Then we have that

$$\begin{bmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} w_2 & w_1 & w_0 & 0 & 0 \\ 0 & w_2 & w_1 & w_0 & 0 \\ 0 & 0 & w_2 & w_1 & w_0 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Q5.2

$$g_y = \begin{bmatrix} w_2 & 0 & 0 \\ w_1 & w_2 & 0 \\ w_0 & w_1 & w_2 \\ 0 & w_0 & w_1 \\ 0 & 0 & w_0 \end{bmatrix} \epsilon$$

Notice that $\epsilon \in \Re^3$.
Q5.3
From Problem Q5.2, we can write

$$g_y = \begin{bmatrix} w_2 & 0 & 0 \\ w_1 & w_2 & 0 \\ w_0 & w_1 & w_2 \\ 0 & w_0 & w_1 \\ 0 & 0 & w_0 \end{bmatrix} \epsilon = \begin{bmatrix} w_1 & w_2 & 0 & 0 & 0 \\ w_0 & w_1 & w_2 & 0 & 0 \\ 0 & w_0 & w_1 & w_2 & 0 \\ 0 & 0 & w_0 & w_1 & w_2 \\ 0 & 0 & 0 & w_0 & w_1 \end{bmatrix} \begin{bmatrix} 0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 0 \end{bmatrix}$$

From the form of the matrix A^t in problem Q5.2, we see that

$$[g_y]_k = w_{-k} * \epsilon_k$$

with the "same" boundary condition and ϵ padded with zeros.
Q5.4
Since convolution is communitive, we also know that $\hat{x}_i = y_i * w_i$ with a "valid" boundary condition. So we have that

$$B_{i,j} = y_{2+i-j} \ ,$$

where again the offset of $2 = 3 - 1$ is due to the valid boundary condition. So then we have that

$$B = \begin{bmatrix} y_2 & y_1 & y_0 \\ y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \end{bmatrix}$$

Then we have that

$$\begin{bmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} y_2 & y_1 & y_0 \\ y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

Q5.5

$$\begin{aligned} \epsilon &= -2(x - f_w(y)) \\ C &= B^t \end{aligned}$$

Q5.6
We have that

$$\begin{aligned} g_w &= -2B^t\epsilon \\ &= -2y_{-i} * \epsilon_i \end{aligned}$$

using a "valid" boundary condition. In this case, we have that

$$\begin{bmatrix} g_{w,0} \\ g_{w,1} \\ g_{w,2} \end{bmatrix} = \begin{bmatrix} y_2 & y_3 & y_4 \\ y_1 & y_2 & y_3 \\ y_0 & y_1 & y_2 \end{bmatrix} \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \end{bmatrix}$$