BME/ECE 695 Deep Learning
Midterm II Solution
April 21, 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: _____

Q2 Convolution versus Correlation (20 Points)

Let w_n and y_n be two discrete time functions for $n \in \{\cdots, -1, 0, 1, \cdots\}$. Then denote the 1D convolution as

$$z_n = w_n * y_n$$

and denote the 1D cross-correlation as

$$x_n = w_n \circ y_n \ .$$

Q2.1 (5 Points)

Write out an explicit expression for the convolution z_n in terms of w_n and y_n.

Q2.2 (5 Points)

Write out an explicit expression for the cross-correlation x_n in terms of w_n and y_n.

Q2.3 (5 Points)

Is convolution commutative? Prove that it is or is not.

Q2.4 (5 Points)

Is cross-correlation commutative? Prove that it is or is not.

Solution:

Q2.1

$$z_n = \sum_{k=-\infty}^{\infty} w_k y_{n-k}$$

Q2.2

$$x_n = \sum_{k=-\infty}^{\infty} w_k y_{n+k}$$

Q2.3

Convolution is commutative.

Proof: For any discrete time functions w_n and y_n, we have that

$$\begin{aligned} w_n * y_n &= \sum_{k=-\infty}^{\infty} w_k y_{n-k} \\ &= \sum_{m=-\infty}^{\infty} w_{n-m} y_m \\ &= \sum_{m=-\infty}^{\infty} y_m w_{n-m} \\ &= y_n * w_n, \end{aligned}$$

where at the second equality we substitute the dummy variable k with $m = n - k$.

Q2.4

Correlation is **not** commutative in general.

Counter example: let $w_n = \delta_n$ and $y_n = \delta_{n-1}$. Then we have that

$$\begin{aligned} w_n \circ y_n &= \sum_{k=-\infty}^{\infty} \delta_k \delta_{n+k-1} = \delta_{n-1} \\ y_n \circ w_n &= \sum_{k=-\infty}^{\infty} \delta_{k-1} \delta_{n+k} = \delta_{n+1} \\ w_n \circ y_n &\neq y_n \circ w_n \end{aligned}$$

Q3 ML and Regularized Estimation (35 Points)
Consider a machine learning algorithm with input/output training pairs of (X_k, Y_k) that are i.i.d. for $k = 0, \cdots K-1$ and parameter vector $\theta \in \Re^p$. Furthermore, assume that

$$X_k = f_\theta(Y_k) + W_k \ ,$$

and where $W_k \sim N(0, \sigma_w^2 I)$. Also assume that $Y_k \sim p(y)$ for some fixed density.

In the Frequentist approach, we will assume that θ is deterministic but unknown.

Alternatively, in the Bayesian approach, we will assume that θ is random with distribution

$$p(\theta) = \frac{1}{z_\sigma} \exp\left\{ -\frac{1}{\sigma} \|\theta\|_1 \right\} \ ,$$

where $\| \cdot \|_1$ denotes the L_1 norm.

Q3.1 (5 Points)

Write out an expression for $p_k(x|y)$, the conditional distribution of X_k given Y_k.

Q3.2 (5 Points)

Write out an expression for $p_k(x, y)$, the joint distribution of X_k and Y_k.

Q3.3 (5 Points)

Write out an expression for

$$L(\theta) = -\log p_\theta(x, y) \ ,$$

the negative log likelihood of (X, Y) where $X = (X_0, \cdots, X_{K-1})$ and $Y = (Y_0, \cdots, Y_{K-1})$.

Q3.4 (5 Points)

Describe how you would compute the (approximate) maximum likelihood estimate for θ given the training data $\{X_k, Y_k\}_{k=0}^{K-1}$ using standard deep learning software?
(Hint: How would you set up your loss function? What loss function would you use?)

Q3.5 (5 Points)

Assuming a Bayesian framework, write out an expression for $p(x, y, \theta)$, the joint distribution of X, Y, and θ.

Q3.6 (5 Points)

Assuming a Bayesian framework, write out an expression for MAP estimate of θ given X, Y.

Q3.7 (5 Points)

Describe how you would compute the (approximate) MAP estimate for θ given the training data $\{X_k, Y_k\}_{k=0}^{K-1}$ using standard deep learning software?
(Hint: How would you set up your loss function? What loss function would you use? What options would you select?)

Solution:

Q3.1 Let $p_w(w) \sim N(0, \sigma^2 I)$, then

$$\begin{aligned} p_k(x|y) &= p_w(x - f_\theta(y)) \\ &= \frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left\{-\frac{1}{2\sigma^2}\|x - f_\theta(y)\|^2\right\} \ . \end{aligned}$$

Q3.2

$$\begin{aligned} p_k(x, y) &= p_k(x|y) \cdot p_k(y) \\ &= \frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left\{-\frac{1}{2\sigma^2}\|x - f_\theta(y)\|^2\right\} \cdot p(y) \ , \end{aligned}$$

(Correction: Problem 3.2 should have stated, "the joint distribution of X_k **and** Y_k.")
Q3.3

$$\begin{aligned} -\log p_\theta(x, y) &= \sum_{k=0}^{K-1} -\log p_k(x_k, y_k) \\ &= \sum_{k=0}^{K-1} -\log p_k(x_k|y_k) - \log p(y_k) \\ &= \frac{Kp}{2}\log(2\pi\sigma^2) + \sum_{k=0}^{K-1}\left\{\frac{1}{2\sigma^2}\|x_k - f_\theta(y_k)\|^2 - \log p(y_k)\right\} \end{aligned}$$

Q3.4
I would train my deep neural network with MSE loss $\frac{1}{K}\sum_{k=0}^{K-1}\|x_k - f_\theta(y_k)\|^2$, because MSE loss corresponds to the minimization of negative log likelihood $-\log p_\theta(x, y)$ w.r.t. parameter θ.
Q3.5

$$\begin{aligned} p(x, y, \theta) &= p(x|y, \theta) \cdot p(\theta|y) \cdot p(y) \\ &= p(x|y, \theta) \cdot p(\theta) \cdot p(y) \\ &= p(\theta) \prod_{k=0}^{K-1} \{p(x_k|y_k, \theta)p(y_k)\} \\ &= \frac{1}{z_\sigma} \exp\left\{-\frac{1}{\sigma}\|\theta\|_1\right\} \prod_{k=0}^{K-1}\left\{\frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left\{-\frac{1}{2\sigma_w^2}\|x_k - f_\theta(y_k)\|^2\right\} \cdot p(y_k)\right\} \ , \end{aligned}$$

(Correction: Problem 3.5 should have used two distinct variables σ_w and σ.)

Q3.6

$$\begin{aligned}\hat{\theta}_{MAP} &= \arg\max_{\theta} \{p(x, y, \theta)\} \\ &= \arg\min_{\theta} \{-\log p(x, y, \theta)\} \\ &= \arg\min_{\theta} \left\{ \frac{1}{2\sigma_w^2} \sum_{k=0}^{K-1} \|x - f_\theta(y)\|^2 + \frac{1}{\sigma}\|\theta\|_1 \right\}\end{aligned}$$

Q3.7

Given training pairs $(X_k = x_k, Y_k = y_k)$ for $K = 0, ..., K-1$, and the forward model parameter θ, we define our loss function as follows:

$$\text{loss}(\theta) = \frac{1}{2\sigma_w^2} \sum_{k=0}^{K-1} \|x_k - f_\theta(y_k)\|^2 + \frac{1}{\sigma}\|\theta\|_1$$

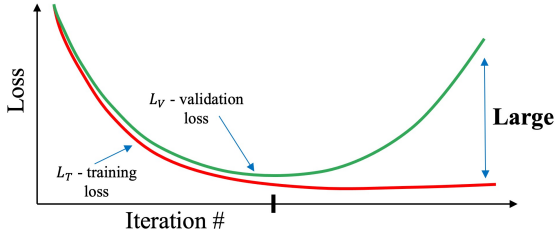
which is equivalent to minimizing the loss function

$$\text{loss}(\theta) = \sum_{k=0}^{K-1} \|x_k - f_\theta(y_k)\|^2 + \beta\|\theta\|_1 \ ,$$

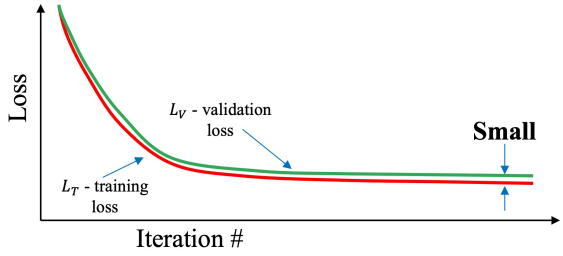
where $\beta = \frac{\sigma_w^2}{\sigma}$ is a hyper-parameter that can be chosen to control the regularization level enforced by the prior distribution of θ.
This can be implemented by using the MSE loss + the L1 norm of the parameters.

Q4 Training, Validation, and Testing (40 Points)

In order to train a deep neural network, you first partition your training data into three disjoint subsets denoted by training, validation, and testing.
For each iteration or epoch of the SGD algorithm, the training and validation loss are plotted, and in two separate cases the following behaviors are observed:
Case 1)



Case 2)



Q4.1 (5 Points)
Which of the three subsets of data are used for SGD optimization?
Q4.2 (5 Points)
What does case 1 suggest about your DL model architecture?
Q4.3 (5 Points)
If $\hat{\theta}_n$ denotes the value of the estimated parameter for case 1 after n iterations or epochs. Which value of n is likely to result in the best model?
Q4.4 (5 Points)
Suggest two techniques that might be used to improve the performance of the DL model in case 1.
Q4.5 (5 Points)
What does case 2 suggest about your DL model architecture?
Q4.6 (5 Points)
If $\hat{\theta}_n$ denotes the value of the estimated parameter for case 2 after n iterations or epochs. Which value of n is likely to result in the best model?
Q4.7 (5 Points)
Suggest an approach that might be used to improve the performance of the DL model used in case 2.
Q4.8 (5 Points)
Explain why the testing data set is also needed in addition to the training and validation data?

Solution:

Q4.1
The SGD optimization is done only with the training subset.
Q4.2
Case 1 suggests that the model order is too high given the available amount of training data.
Q4.3
Typically the best value n^* is chosen so that

$$n^* = \arg\min_n \text{loss}_V(\theta_n) \ ,$$

where $\text{loss}_V(\theta)$ is the loss computed on the validation subset.
So in other words, n^* is selected so that it is the minimum of the validation curve.
Q4.4
Some techniques that can be used to improve the result for Case 1 include:

- Reduce the model order or capacity of the DNN
- Get more training data
- Use L1 or L2 regularization of the model parameters.
- Use dropouts

Q4.5
Case 2 suggests that the model capacity is too low.
Q4.6
In this case, the best value n^* is the final iteration since the validation loss continues to drop to the end.
Q4.7
For Case 2, the model order or capacity of the DNN must be increased.
Q4.8
The test data is used to make a final determination of the inference accuracy.
This is needed because after a series of tuning steps that use the inference data, the inference loss function may not give an accurate (i.e., unbiased) estimate of the loss when used on new data.

Q5 Bayesian Discriminator (10 Points)
You have designed a generative adversarial network (GAN). In order to design the GAN, you are given K i.i.d. random vectors, $Y_k \sim p_r(y)$, with reference distribution $p_r(y)$.
The generator then used to generate K i.i.d. random vectors $\tilde{Y}_k \sim p_{\theta_g}(y)$ where θ_g is the generator parameter vector.
Finally, you flip a fair coin to produce a random variable $C \in \{R, F\}$, and depending on the outcome, you either select a random vector Y from the "reference" distribution $p_r(y)$, or from the "fake" distribution $p_{\theta_g}(y)$.

Q5.1 (5 Points)
Write an expression for $p(c, y)$, the joint probability of C and Y.
Q5.2 (5 Points)
Use Bayes' rule to write an expression for $f(y) = P\{C = R|Y = y\}$, i.e., the conditional probability of that y is real.

Solution:

Q5.1

$$\begin{aligned} p(c,y) &= p(y|c)p(c) \\ &= \frac{1}{2}p_r(y)\delta(c=R) + \frac{1}{2}p_{\theta_g}(y)\delta(c=F) \end{aligned}$$

Q5.2

$$\begin{aligned} f(y) &= P\{C=R|Y=y\} \\ &= \frac{P\{Y=y|C=R\}P\{C=R\}}{P\{Y=y\}} \\ &= \frac{P\{Y=y|C=R\}P\{C=R\}}{P\{Y=y|C=R\}P\{C=R\} + P\{Y=y|C=F\}P\{C=F\}} \\ &= \frac{\frac{1}{2}p_r(y)}{\frac{1}{2}p_r(y) + \frac{1}{2}p_{\theta_g}(y)} \\ &= \frac{p_r(y)}{p_r(y) + p_{\theta_g}(y)} \\ &= \frac{1}{1 + \frac{p_{\theta_g}(y)}{p_r(y)}} \end{aligned}$$