# CS 615 Deep Learning
# Spring 2020
# Exam

Rules and Regulations

1. The following exam is open book in that you **MAY** use your textbook, class notes, course slides and your prior assignments.
2. You are **NOT** allowed to discuss solutions with other students or use web resources.
3. **Since this is a "take home" exam, honor and honesty extremely important. Any evidence of violating this with be reported to the integrity review board and will result in an automatic failure for the course.**
4. You **May** use a language like Matlab or Python to do computations for you, including basic matrix operations. As a rule of thumb, don't just use some function that defeats the spirit of the problem. Use your judgment, and when in doubt, show more work!
5. **Your solutions MUST be hand-written, scanned (or photographed), and then submitted via Blackboard**
    a. I have provided what I think is adequate space for your solution to each question. But if you prefer to work on other paper just make sure to have everything well labeled and/or transfer your solutions back onto the exam paper.
6. **You have 2.5 contiguous hours from when you start the exam until when you must submit it (2hrs for working on the exam + ½ hr for preparing submission).**
7. Since I will not be available for clarification during the exam, if you are unsure of something explain your uncertainty and attempt the problem(s) to the best of your ability.

| **Part** | **Potential** | **Obtained** |
|---|---|---|
| Part I: Open Ended | 40 | |
| Part II: Forward Propagation | 35 | |
| Part III: Derivations | 25 | |
| | | |
| **Total** | **100** | |

*Good Luck!*

## Part I: Basic Theory/Open-Ended (40pts)

*For each of the following answer with a number, word, or sentence. Keep things short (the more concise, the better)*

1. For each objective function, what is the optimal value (2pts each)
    a. Log-likelihood?

    b. Cross entropy?

    c. Squared error?

2. Why does it often make more sense to use a softmax activation function at the output layer than a logistic one if we are doing multi-class classification? (5pts)

3. If we notice that our objective function does well for our training data but not our testing data, what might this indicate? (5pts)

4. What might we do to attempt to overcome the problem mentioned in the previous question? (5pts)

5. If we notice that our objective function is oscillating during our training process, what might this indicate? (5pts)

6. What might we do to attempt to overcome the problem mentioned in the previous question? (5pts)

7. Are we **more** or **less** likely to overfit as we increase the number of nodes in a hidden layer? (3pts)

8. Given $\hat{y} = [0.1, 0.3, 0{,}6]$ and $y = [1, 0\ 0]$, what is the value of (2ea):
   a. Log likelihood

   b. Cross entropy

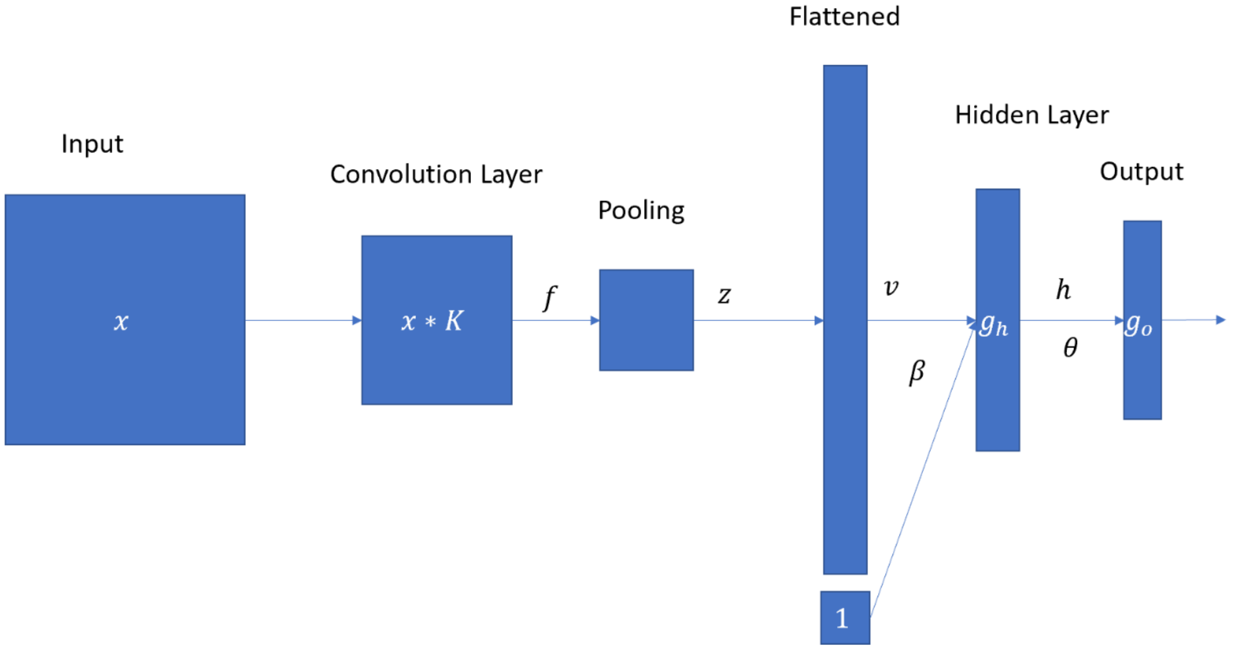
   c. Squared error

## Part II: Forward Propagation (35pts)

*For questions in this section you should use a tool like Matlab or Python to do math/matrix operations (including convolution). But show as much work as you feel is necessary to follow along with your thought process.*

9. Compute forward propagation, given the following architecture. Show the input and output of each stage (15pts)

*Architecture*



Input/Parameters

- Input: $x = \begin{bmatrix} 4 & 6 & 2 & 10 & 10 & 8 & 7 & 1 & 6 \\ 3 & 6 & 2 & 4 & 6 & 7 & 8 & 1 & 9 \\ 9 & 2 & 3 & 2 & 1 & 5 & 1 & 1 & 8 \\ 5 & 9 & 5 & 8 & 3 & 6 & 10 & 1 & 7 \\ 10 & 7 & 1 & 4 & 4 & 3 & 8 & 1 & 4 \\ 2 & 4 & 10 & 3 & 9 & 8 & 5 & 1 & 9 \\ 3 & 6 & 10 & 5 & 1 & 2 & 5 & 1 & 6 \\ 2 & 5 & 5 & 1 & 1 & 7 & 5 & 1 & 4 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
- Convolution Kernel: $K = \begin{bmatrix} 3 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}$. Only do **valid** convolution, stride of 1.
- Pooling: $3 \times 3$ max pool with stride of 2 (no padding).
- Bias node from flattened layer $V$ to hidden layer
- Weights from flattened layer to hidden layer (plus bias node): $\beta = \begin{bmatrix} -1 & 2 & 5 \\ 0 & 0 & 1 \\ -5 & 10 & 3 \\ 1 & 1 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
- Linear activation function at hidden layer, $g_h$
- Weights from hidden layer to output layer: $\theta = \begin{bmatrix} \frac{1}{100} & \frac{2}{100} \\ -\frac{1}{100} & \frac{2}{100} \\ \frac{0}{100} & \frac{4}{100} \end{bmatrix}$
- Softmax activation function at output layer, $g_o$

10. An air conditioning unit can either be on or off. In your apartment you find that you are either too hot, too cold, or just right. (20 points total)

To analyze what is going on you took notice of how you felt several times throughout the day, checked on the HVAC system, an observed the following probabilities:

| | AC On | AC Off |
|---|---|---|
| Too Hot | 0.1 | 0.6 |
| Too Cold | 0.5 | 0.3 |
| Just Right | 0.4 | 0.1 |

Additionally, you noticed that at any given moment the likelihood of the system staying in its current state is 75% and that whenever you start your observations the initial state of the AC is purely random (50/50).

You decide this is a Hidden Markov Model problem and want to use this information to determine some things.

Let us start off by establishing information about the system:

a. (2pts) What are the hidden states?

b. (5pts) What is the initial state probability vector?

c. (5pts) What is the state transition matrix?

d. (8pt) Today you noticed that you were Too hot, then Just Right, then Just Right again. What is the most probable state sequence given those observations? **You MUST show your computations.**

Part III: Derivations (25pts)

*For the next few questions, we will describe the architecture of a "made up" network.*

11. Given an objective function $J = \sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^3$ what is partial derivative of this objective function with regards to the predicted outputs? (4pts)?

12. What are the partial derivatives for each of the following activation functions with regards to the data that is coming into them $(z)$? (2pts each)

    a. $g(z) = 2z$

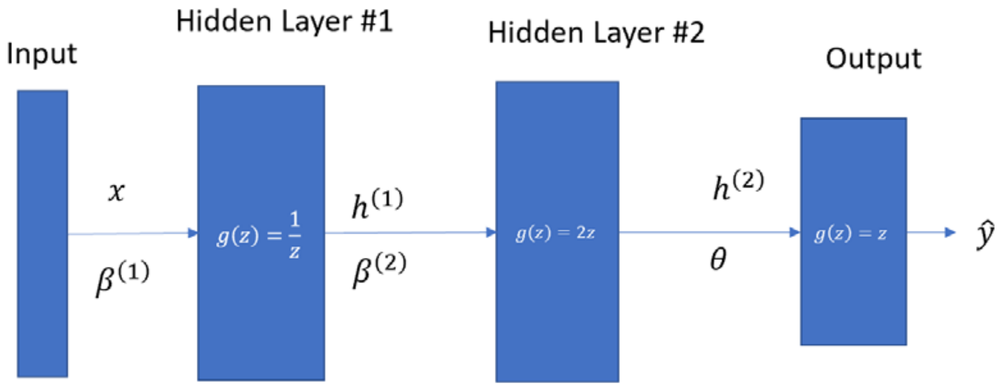
    b. $g(z) = z^2 - 3z$

    c. $g(z) = \frac{1}{z}$

13. Given the following architecture, determine the gradient rules necessary to update all the parameters: $\frac{\partial J}{\partial \theta}, \frac{\partial J}{\partial \beta^{(2)}}, \frac{\partial J}{\partial \beta^{(1)}}$. Note that you did many of the partials in the previous questions (15pts).

    In addition:

    - We'll do online learning, so our objective function for a single observation is
    $$J = (y - \hat{y})^3$$
    - There are no bias nodes.
    - $\beta^{(1)}$ are the weights going from the input layer to the first hidden layer.
    - $\beta^{(2)}$ are the weights going from the output of the first hidden layer to the second hidden layer.
    - $\theta$ are the weights going from the output of the second hidden layer to the output layer.
    - $h^{(1)}$ and $h^{(2)}$ are the outputs of the first and second hidden layers, respectively.
    - Note the activation functions shown within each layer.
    - $\hat{y}$ is a single value output.


Input
Hidden Layer #1
Hidden Layer #2
Output
$x$
$\beta^{(1)}$
$g(z) = \frac{1}{z}$
$h^{(1)}$
$\beta^{(2)}$
$g(z) = 2z$
$h^{(2)}$
$\theta$
$g(z) = z$
$\hat{y}$