CS 523 – Deep Learning

Exam

Summer 2021

Instructions:

1- Share your video and audio
2- Set an alarm for 2:45pm EST. You have 1 hour and 40 minutes to solve the exam
3- Solve the exam using paper and pen
4- Print your name and BU ID clearly on the top right of the first page (the page that will have the solution to Problem 1)
5- Start a new page for each question
6- Stop solving at 2:45pm EST
7- Take photos of your solutions in order using your phone
8- Convert the photos into a single pdf
9- Submit your pdf file on GradeScope
10- You will receive a confirmation message from us that we received your submission. **<u>It is your responsibility to make sure the file contains solutions to all the problems you solved.</u>**

Notes:

** There are six questions. The last page has some common formulas*
** If you have any inquiries during the exam please message us in the zoom chat in 40mins after raising your hand or using a zoom reaction.*
** You may turn the speakers off, but please keep your chat open*
** We are only able to grade what we can read*
** Total points: 64*

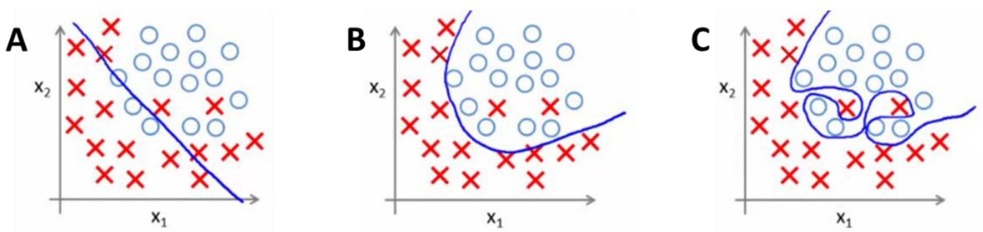
Good luck ☺

Q1. [8 points] Overfitting and Regularization

Suppose you want to fit a Logistic Regression model to predict whether an email is spam ($y = 1$) or not spam ($y = 0$) based on the frequency of the words "buy" (feature x_1) and "click" (feature x_2). You decide to use polynomial basis functions to represent the input features and to apply regularization. You have fit three models by minimizing the regularized Logistic Regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log\left(h_\theta(x^{(i)})\right) - (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=2}^{n}\theta_j^2$$

for $\lambda = 10^{-2}, 10^{0}, 10^{2}$. The following are sketches of the resulting decision boundaries.



a) [3 points] Which value of λ goes with each of the plots?

A: **B:** **C:**

b) [3 points] You plot model complexity *M* (number of polynomials) versus the cost function, computed on the test data, and a similar curve for the training data. Explain how you can use these curves to detect when the model is overfitting and draw and example to illustrate.

c) [2 points] Name the regularization technique specifically designed for Deep Neural networks and briefly describe how it achieves such regularization.

Q2. [12 points] Training Strategies

a) [2 points] Alice decides to use Principal Component Analysis to implement image compression. Briefly explain how she should train the algorithm and how she should use it to compress a new image. What parameter controls the amount of compression?

b) [2 points] Suppose you have trained an anomaly detection system for intruder detection in a security camera, and your system flags anomalies when p(x) is less than ε. You find on the cross-validation set that it is missing many intruder events. What should you do?

c) [2 points] Describe K-fold cross-validation.

d) [2 points] Describe the goal of an autoencoder, incorporating how that goal is achieved through the design of the loss function.

e) [2 points] What are the benefits of fine-tuning a pre-trained network?

f) [2 points] What is a minibatch? List an advantage of using a minibatch.

Q3. [10 points] Design Questions

a) [4 points] For the tasks of de-noising/inpainting an input image using an auto-encoder:

i) What is the input?

ii) What is the ground-truth?

iii) What is the loss function?

iv) Sketch the entire pipeline

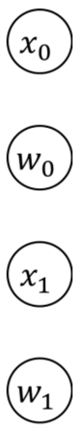
b) [3 points] How would you design a deep learning model that computes the temporal regions where an advertisement makes audience laugh/smile while watching it online, and how can you contrast that with regions where an advertisement is meant to be funny?

c) [3 points] Sketch a pipeline for unsupervised domain adaptation, where the source data has class labels but the target data does not, and formulate and explain the associated loss/cost function.

Q4. [11 points] Backpropagation

Suppose we want to compute the gradients for the function $h(x) = q(w_0x_0 + w_1x_1)$, where $x = [x_0\ x_1]^T$ is the input vector, $w = [w_0\ w_1]^T$ is the parameter vector, and q is the *tanh* function: $\mathrm{q(u)} = \tanh(u) = (e^u - e^{-u})/(e^u + e^{-u})$. The *tanh* function is also plotted in the appendix.

a) [2 points] Complete the computational graph for this function below by adding two nodes $f_1(u, c) = c * u$ (multiplication), one node $f_2(u, c) = u + c$ (addition), and one node $f_3(u) = q(u)$. Label the nodes clearly with f_1, f_2 or f_3 and leave plenty of space between them.



b) [3 points] Write down the gradient $\frac{\partial f}{\partial u}$ for functions f_1, f_2, f_3. *Hint: note that* $tanh'(u) = 1 - tanh(u)^2$.

$\frac{\partial f_1}{\partial u} =$

$\frac{\partial f_2}{\partial u} =$

$\frac{\partial f_3}{\partial u} =$

c) [2 points] Perform a forward pass for $x = [5 \quad 5]^T$ and $w = [1 \quad -1]^T$, **writing values on top of the arrows** in your computational graph. What is the output of the forward pass, *i.e.* $h(x)$?

d) [4 points] Perform a backward pass for the example in (c), **writing values below the arrows** in the graph. What are the gradients of h with respect to x_0, x_1, w_0, w_1?

$\frac{\partial h}{\partial x_0} =$ $\qquad$ $\frac{\partial h}{\partial x_1} =$ Type equation here.

$\frac{\partial h}{\partial w_0} =$ $\qquad$ $\frac{\partial h}{\partial w_1} =$

Q5. [14 points] DL Architectures

Answer the following questions in brief one or two sentence answers.

a) [2 points] Is it recommended to do feature engineering first and then apply deep learning? Contrast deep learning with other machine learning algorithms in terms of feature engineering.

b) [2 points] Suppose we have a neural network with ReLU activation function. Let's say, we replace ReLu activations by linear activations. Would this new neural network be able to approximate a non-linear function? And why?

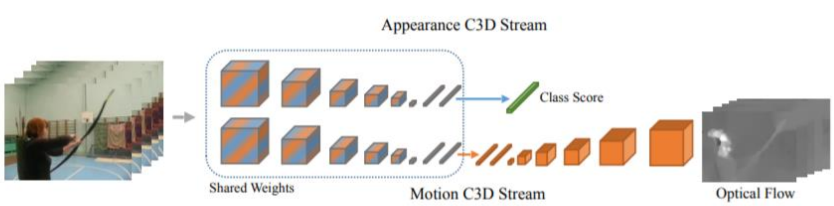
c) [2 points] Name an example of a data augmentation approach and explain how it helps improve model generalization.

d) [2 points] What is the main difference between a fully-connected and a convolutional network?

e) [2 points] List two ways to downsize feature maps in convolutional neural networks.

f) [2 points] Describe the benefit of using soft over hard targets in knowledge distillation.

g) [2 points] If the following is an action classification model, discuss the auxiliary task presented and how it can help the classification task.



Q6. [9 points] Training Strategies

a) [3 points] Suppose you decide to use the loss $L(z) = exp(-2z - 1)$, where z is a function of the weights w. Write down the gradient descent algorithm for minimizing this loss on a set of training examples $\{x_i, y_i\}, i = 1, \dots, m$, using a squared L-2 norm regularizer $R(w) = \|w\|^2$ on the parameter vector.

i) What is the cost function?

ii) Should it be minimized or maximized?

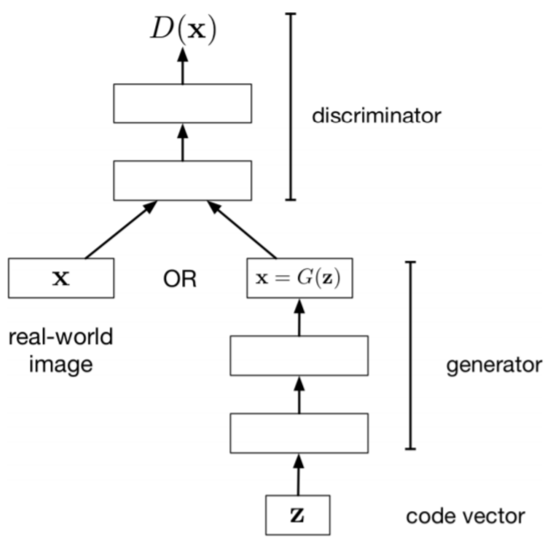
iii) What is the corresponding gradient descent update step for w?

b) [3 points] Explain the terms in the following equation and why they are needed, then list one application that would benefit from using this setup.

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i = \gamma^t r_t + \gamma^{t+1} r_{t+1} \dots + \gamma^{t+n} r_{t+n} + \cdots$$

γ: discount factor; $0 < \gamma < 1$

c) [3 points] Describe how the discriminator and generator modules of this GAN are typically trained, *i.e.* how their parameters are updated.



Appendix: Common Formulas

Chain Rule

If z = f(y) and y = g(x), then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} = f'\big(g(x)\big) * g'(x)$

Hyperbolic Tangent Function (tanh)