CSE-5368-001Neural Networks Fall 2020 Exam 1

Name	ID

Prob #	1	2	3	4	5
Points	12	22	22	22	22

Time: 80 Minutes

NOTES:

a. **Credit is only given to the correct numerical values.**
b. **All numerical values must be calculated with three digits of accuracy after the decimal point.**

Name	ID

Hinge (SVM)

Taylor Series

$$F(\mathbf{x}) = F(\mathbf{x}^*) + \nabla F(\mathbf{x})^T\Big|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x}-\mathbf{x}^*)$$

$$+\frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^T\nabla^2 F(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x}-\mathbf{x}^*) + \cdots$$

Name	ID

1. Consider a neural network used for classification. If the output of the network is:

$$\begin{bmatrix} class\ 0\ score \\ class\ 1\ score \\ class\ 2\ score \end{bmatrix} = \begin{bmatrix} 0.2 \\ 1 \\ -0.5 \end{bmatrix}$$

a. If the target class is **class 0**, what is numerical value of the SVM loss? Assume a margin of 1.

Numerical value of the loss is:

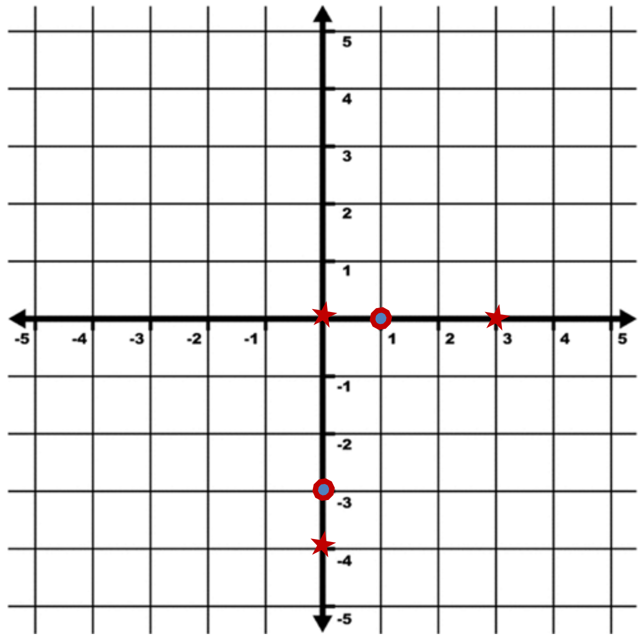
b. What would the loss be if the target class was **class 1**?

Numerical value of the loss is:

Name	ID

Name	ID

2. Design a two-layer Perceptron neural network which will correctly classify the two classes (circles and stars) as shown below. Assume the activation (transfer) function for all the nodes are hard-limit with the output of **0 (star)** and **1(Circle)**.



Show the weight matrices and biases for both layers. Biases should be included in the weight matrices in the first column.

Weight matrix for the first layer =

Weight matrix for the second layer =

Name	ID

Name	ID

3. Consider the expression: $\boldsymbol{f(x) = \frac{(x+y)}{y} * [min(y,z)]^2}$

Given the inputs $\boldsymbol{x = 2}$, $\boldsymbol{y = 4}$, $\boldsymbol{z = -3}$

Draw the computational graph.

Calculate the $\frac{\delta f(x,y,z)}{\delta x}$ ***and*** $\frac{\delta f(x,y,z)}{\delta y}$ ***and*** $\frac{\delta f(x,y,z)}{\delta z}$ and show all the numerical values of the backward pass.

You must show all the numerical values as they flow in the forward and backward path.

Name	ID

Name	ID

4. Consider the following performance surface:

$$\boldsymbol{F(X) = x_1^3 - 2x_2^2 + 5x_3^2 + 4x_1x_2}$$

Find the second order Taylor series expansion of this function around point $P = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

Note: second order means to include second order derivatives

Show both matrix version and expanded version

Name	ID

Name	ID

5. Complete the code for the following function (this is similar to function in assignment 01). Note that YA and YT are given

```
import numpy as np
def calculate_percent_error(YA,YT):
""" Given a batch of input, actual outputs, and desired outputs, this function
calculates percent error. For each sample, if the actual output vector is not
exactly the same as the desired output, it is considered one error. Percent
error is 100*(number_of_errors/ number_of_samples)
:param YA: Array of actual outputs [number_of_nodes, ,n_samples]. Assume that
each element of YA is either 0 or 1 (Result of hard-limit activation function).
param YT: Array of desired (target) outputs [number_of_nodes ,n_samples].
Assume that each element of YT is either 0 or 1
:return percent_error"""
```

Name	ID