

CSE-5368-001 Neural Networks

Fall 2021 Exam 1

Last Name	First Name	ID

Prob #	1	2	3	4	5
Points	10	20	20	25	25

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.

1. Consider a convolutional neural network.

Note: **Do NOT consider Biases.**

Input layer:

Input to this CNN are color images of size **64x52x3** with the batch size = 128

Note: Input image has **different horizontal and vertical** resolution.

Next layer is Conv2D layer:

number of filters: 30, filter size: 11x11 ; stride: 3x3 ; padding: 2x2

What is the shape of the weight matrix for this layer? _____

What is the shape of the output (tensor) of this layer? _____

Next layer is Flatten layer:

What is the shape of the output (tensor) for this layer? _____

Next layer is Dense layer:

number of nodes: 100

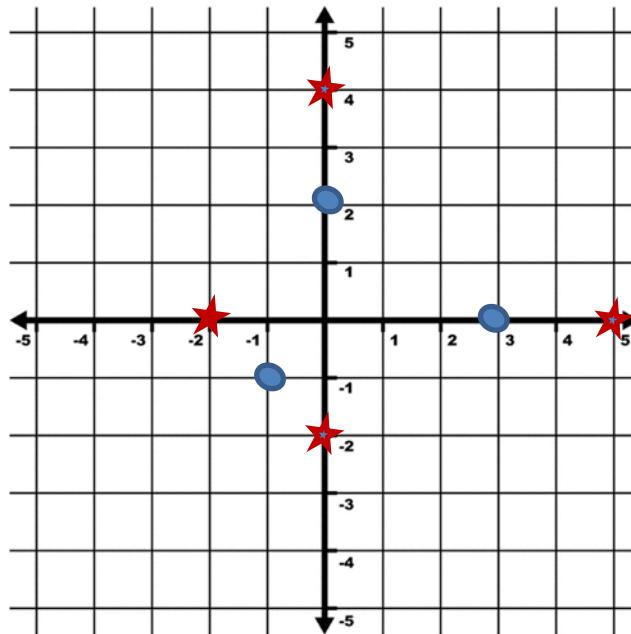
What is the shape of the weight matrix (tensor) for this layer? _____

What is the shape of the output (tensor) for this layer? _____

CSE-5368-001 Neural Networks

Fall 2021 Exam 1

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 =

CSE-5368-001 Neural Networks
Fall 2021 Exam 1

3. Consider the expression:

$$f(x, y) = \frac{xy}{\min(xy, y^2) - 6}$$

given the inputs: $x = 5$, $y = 4$

Draw the computational graph for this expression.

- You must show all the forward values and backward **partial derivatives for each node** in the graph and compute numerical values of the partial derivatives with respect to x and y

$\frac{\delta f(x, y)}{\delta x} =$
$\frac{\delta f(x, y)}{\delta y} =$

CSE-5368-001 Neural Networks

Fall 2021 Exam 1

- Complete the code for the following method/function (this is the same function in assignment 04).
Assume that the object that you are working with, i.e. self, is the same as the object that you used in your assignment 04. For example, if you used a Keras model in your assignment 04 then assume your object (self) is a Keras model.

Hint: you may use `layers[]` and `get_layer()` and `get_weights()` property and methods

```
import numpy as np

def get_biases(self,biases,layer_number=None,layer_name=""):
    """ This function gets the biases for the layer specified by layer_number or by
    layer_name.
    :param layer_number: Layer number starting from layer 0.
    :param layer_name: Layer name (if both layer_number and layer_name are
    specified, layer number takes precedence).
    :return: biases for the given layer (If the given layer does not have bias then
    None should be returned) """
```

5. Complete the following code using Tensorflow. The code should initialize the weights and biases and compute the output, loss, and gradients, and perform a single weight update of a neural network. Assume the dimension of each input sample is 400. There are 100 nodes with sigmoid activation in the first layer, and 10 output nodes with linear activation. Loss is MSE. Assume you have a batch of inputs called X that has dimensions [20, 400], and a batch of targets called y that has dimensions [20, 10]

```
import tensorflow as tf
import numpy as np
# Create random weights and biases
W_1 = tf.Variable(np.random.randn(          ), trainable=True)
b_1 = tf.Variable(np.random.randn(          ), trainable=True)
W_2 = tf.Variable(np.random.randn(          ), trainable=True)
b_2 = tf.Variable(np.random.randn(          ), trainable=True)
with tf.GradientTape(persistent=True) as tape:
# Calculate output

output=
# Calculate loss
```

CSE-5368-001 Neural Networks
Fall 2021 Exam 1

Loss=

Calculate gradients

dW_1, db_1 =

dW_2, db_2 =

Calculate new values of weights and biases
