CSE-5368 Neural Networks Fall 2024 Exam_01

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

Prob #	**1**	**2**	**3**	**4**	**Total**
Points	**25**	**25**	**25**	**25**	

Time: 80 Minutes

,

CSE-5368 Neural Networks Fall 2024 Exam_01

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

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

$$\frac{\mathbf{p}^T\nabla F(\mathbf{x})}{\|\mathbf{p}\|} \qquad \frac{\mathbf{p}^T\nabla^2 F(\mathbf{x})\mathbf{p}}{\|\mathbf{p}\|^2} \qquad \alpha_k = -\frac{\mathbf{g}_k^T\mathbf{p}_k}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\mathbf{g}_k \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$$

$$L_i = \sum_{j\neq i} max(0, y_j - y_i + \Delta)$$

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

$$H(p,q) = -\sum_x p(x)log(q(x))$$

$$L_i = -log(\frac{e^{y_i}}{\sum_j e^{y_j}})$$

CSE-5368 Neural Networks Fall 2024 Exam_01

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

1. Consider the following performance surface

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

Given the initial point $\begin{bmatrix} \mathbf{-1.5} \\ \mathbf{-2} \end{bmatrix}$, take **two steps** of the **steepest descent algorithm**, minimizing along a line **at each step**.

You must show your steps and the resulting positions after each step.

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

Problem 1 Continued

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

2. Consider a convolutional neural network.
   Note: **Do NOT consider Biases**.

Input layer:

Input to this CNN are color images of size **100x70x3** with the **batch size = 30**

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

Next layer is Conv2D layer:

Number of filters: **15**, filter size: **9x9**; stride: **3x3**; padding: **4x4**

Q1: What is the shape of the weight matrix for this layer? **Q1:** _____

Q2: What is the shape of the output (tensor) of this layer? **Q2:** _____

Next layer is Conv2D layer:

Number of filters: **10**, filter size: **4x4**; stride: **2x2**; padding: **1x1**

Q3: What is the shape of the weight matrix for this layer? **Q3:** _____

Q4: What is the shape of the output (tensor) of this layer? **Q4:** _____

Next layer is Flatten layer:

Q5: What is the shape of the output (tensor) for this layer? **Q5:** _____

Next layer is Dense layer:

Number of nodes: **50**

Q6: What is the shape of the weight matrix for this layer? **Q6:** _____

Q7: What is the shape of the output (tensor) for this layer? **Q7:** _____

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

Problem 2 Continued

CSE-5368 Neural Networks Fall 2024 Exam_01

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

3. Consider the expression:

$$f(x,y) = \frac{xy}{35 - min(xy, x^2)}$$

given the inputs: $x = 5,\ \ y = 6$

Draw the computational graph and calculate the $\frac{\delta f(x,y)}{\delta x}$ and $\frac{\delta f(x,y)}{\delta y}$

For proper credit, you MUST SHOW all the numerical values **for each node** as they flow in the forward and backward path in the computational graph.

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

Problem 3 Continued

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

4. Complete the code for the following function.

This function is the same as the **initialization part of the Assignment_01**.

```
import numpy as np
def multi_layer_nn(X_train,Y_train,layers):
# This function Initializes all the weights and returns weight
# matrices in a list
# bias should be included in the weight matrix.
# X_train: Array of input for training [input_dimension,nof_train_samples]
# Y_train: Array of desired outputs [output_dimension,nof_train_samples]
# layers: array of integers representing number of nodes in each layer
# return: This function should return a list of weight matrices.
# Each element of the list should be a numpy array corresponds to the
# weight matrix of the corresponding layer.
# Initialize all the weights to zero
```

Last Name	First Name	Student ID Number
«Last_Name»	«First_Name»	«UTA_ID»

Problem 4 Continued