# Written Assignment 2

Deadline: February 10, 2024

**Instruction:** You may discuss these problems with classmates, but please complete the write-ups individually. Remember the collaboration guidelines set forth in class: you may meet to discuss problems with classmates, but you may not take any written notes (or electronic notes, or photos, etc.) away from the meeting. Your answers must be **typewritten**, except for figures or diagrams, which may be hand-drawn. Please submit your answers (pdf format only) on **Canvas**.

## Q1. Kernels (35 points)

In this question, we are going to use Perceptron with a kernel $K(x, z) = (x \cdot z)^2$. Also note that we are not using a bias weight $w_0$ for simplicity). Imagine that we have $x^{(i)} \in \mathbb{R}^2$, and we encounter the following data points:

| $i$ | $x_1^{(i)}$ | $x_2^{(i)}$ | $y^{(i)}$ |
|---|---|---|---|
| 1 | 0 | -2 | -1 |
| 2 | 2 | 0 | -1 |
| 3 | 0 | 1 | 1 |
| 4 | -1 | 0 | 1 |
| 5 | -2 | 0 | -1 |
| 6 | 1 | 0 | 1 |

- Provide values of the weights $\{a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)}, a^{(5)}, a^{(6)}\}$ after training
- We have a new data point $x^{\text{test}} = [\sqrt{2} \quad \sqrt{2}]^T$. What is the predicted label $\hat{y}^{\text{test}}$ for $x^{\text{test}}$?

**Note.** For this question, you can compute each step manually or write a python program for it. If you write a python program, you will need to submit the program together with your answers.

## Q2. Computational Graph + Back-propagation (40 points)

Let's consider a regression problem. We have each data point $x = [x_1, x_2] \in \mathbb{R}^2$. We build a simple neural net with one hidden layer to produce a prediction as follows: (i) The hidden layer: $z = w \cdot x$ where $z \in \mathbb{R}$ and the weight vector $w = [w_1, w_2] \in \mathbb{R}^2$; and (ii) The output layer produces a prediction $\hat{y} = \text{sigmoid}(z)$. We use the error $(\hat{y} - y)^2$ to measure the prediction loss where $y$ is the true label. Assume that we have an input $x = [1, 0]$ with the true label $y = 1$. The initial weight

vector $w = [-1, 2]$. Draw the computational graph for this problem and run the forward/backward pass to compute the gradient of the prediction loss with respect to the weights, i.e., $\frac{\partial \text{loss}}{\partial w_1}$ and $\frac{\partial \text{loss}}{\partial w_2}$.

**Note** For this question, you can compute each step manually or write a python program for it. If you write a python program, you will need to submit the program together with you answers. In addition, there are different computational graphs that you can build for this problem. You just need to provide one computational graph for your answer.

## Q3. Convolutional Neural Network (25 points)

Let's consider the following image of size $7 \times 7$.

- We apply the convolution operation on this image using a $3 \times 3$ filter as shown below with stride $= 2$ and zero-padding $= 1$. Provide the resulting output.
- Now, instead of applying convolution, we are going to apply the max pool with a $4 \times 4$ filter and stride $= 1$ to the original image. Provide the resulting output.

**Note** For this question, you can compute each step manually or write a python program for it. If you write a python program, you will need to submit the program together with you answers.

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 4 | 2 | 5 | 6 | 2 | 3 |
| 1 | 7 | 2 | 3 | 3 | 6 | 7 |
| 2 | 5 | 3 | 6 | 5 | 5 | 6 |
| 2 | 4 | 4 | 7 | 6 | 7 | 5 |
| 3 | 6 | 3 | 6 | 7 | 8 | 6 |
| 4 | 5 | 5 | 5 | 7 | 6 | 7 |
| 6 | 4 | 3 | 6 | 6 | 7 | 7 |

7x7 image

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

3x3 filter