Optimization and Machine Learning, Spring 2020
Homework 5

(Due Tuesday, June 2 at 11:59pm (CST))

1. Show that weighted Euclidean distance in $\mathbb{R}^p$,

$$d_e^{(w)}(x_i, x_{i'}) = \frac{\sum_{l=1}^{p} w_l (x_{il} - x_{i'l})^2}{\sum_{l=1}^{p} w_l},$$

satisfies

$$d_e^{(w)}(x_i, x_{i'}) = d_e(z_i, z_{i'}) = \sum_{l=1}^{p} (z_{il} - z_{i'l})^2,$$

where

$$z_{il} = x_{il} \left(\frac{w_l}{\sum_{l=1}^{p} w_l} \right)^{1/2}.$$

Thus weighted Euclidean distance based on x is equivalent to unweighted Euclidean distance based on z. (15 points)

2. Consider a dataset of n observations $\mathbf{X} \in \mathbb{R}^{n \times d}$, and our goal is to project the data onto a subspace having dimensionality p, $p < d$. Prove that PCA based on projected variance maximization is equivalent to PCA based on projected error (Euclidean error) minimization. (20 points)

3. Show that the conventional linear PCA algorithm is recovered as a special case of kernel PCA if we choose the linear kernel function given by $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$. (15 points)

4. Let $S = \{x^{(1)}, \cdots, x^{(n)}\}$ be a dataset of n samples with 2 features, i.e $x^{(i)} \in \mathbb{R}^2$. The samples are classified into 2 categories with labels $y^{(i)} \in \{0, 1\}$. A scatter plot of the dataset is shown in Figure 1.

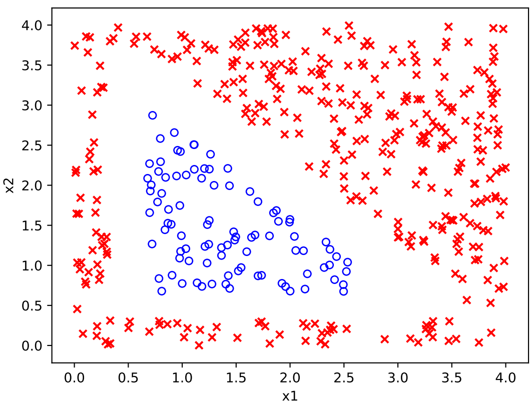


Figure 1: Plot of dataset S.

The examples in class 1 are marked as as "$\times$" and examples in class 0 are marked as "$\circ$". We want to perform binary classification using a simple neural network with the architecture shown in Figure 2.

Denote the two features x_1 and x_2, the three neurons in the hidden layer h_1, h_2, and h_3, and the output neuron as o. Let the weight from x_i to h_j be $w_{i,j}^{[1]}$ for $i \in \{1, 2\}, j \in \{1, 2, 3\}$, and the weight from h_j to o be $w_j^{[2]}$. Finally, denote the intercept weight for h_j as $w_{0,j}^{[1]}$, and the intercept weight for o as $w_0^{[2]}$. For the loss

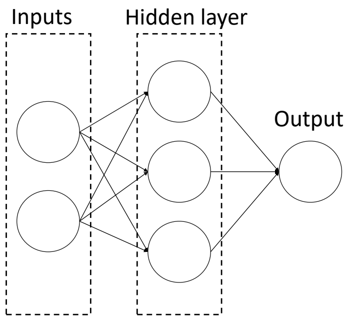


Figure 2: Architecture for our simple neural network.

function, we'll use average squared loss instead of the usual negative log-likelihood:

$$l = \frac{1}{n}\sum_{i=1}^{n}(o^{(i)} - y^{(i)})^2,$$

where $o^{(i)}$ is the result of the output neuron for example i.

(a) Suppose we use the sigmoid function as the activation function for h_1, h_2, h_3 and o. What is the gradient descent update to $w_{2,1}^{[1]}$, assuming we use a learning rate of α? Your answer should be written in terms of $x^{(i)}$, $o^{(i)}$, $y^{(i)}$, and the weights. (10 points)

(b) Now, suppose instead of using the sigmoid function for the activation function for h_1, h_2, h_3 and o, we instead used the step function $f(x)$, defined as

$$f(x) = \begin{cases} 1, x \geq 0, \\ 0, x < 0. \end{cases}$$

Is it possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy? If it is possible, please provide a set of weights that enable 100% accuracy and explain your reasoning for those weights in your PDF. If it is not possible, please explain your reasoning in your PDF. (10 points) (Hint: There are three sides to a triangle, and there are three neurons in the hidden layer.)

(c) Let the activation functions for h_1, h_2, h_3 be the linear function $f(x) = x$ and the activation function for o be the same step function as before. Is it possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy? If it is possible, please provide a set of weights that enable 100% accuracy and explain your reasoning for those weights in your PDF. If it is not possible, please explain your reasoning in your PDF. (10 points)

5. Convolutional neural networks targets the processing of 2-D features instead of the 1-D ones in multi-layer perceptron (MLP), the structure of which is depicted in Fig. 3.

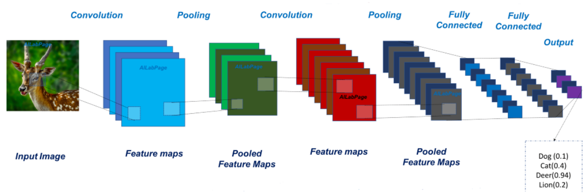


Figure 3: `https://mc.ai/how-does-convolutional-neural-network-work/`

(a) Kernel convolution is process where we take a kernel (or filter), we pass it over our image and transform it based on the values from filter. Now, you are given the following formula

$$G(m,n) = (f * h)(m,n) = \sum_j \sum_k h(j,k) f(m-j, n-k),$$

where the input image is denoted by f and the kernel by h. The indexes of rows and columns of the result matrix are marked with m and n respectively. Please calculate the feature maps, if you are given the following 3×3 image matrix and 2×2 kernel matrix. (5 points)

1	2	3
4	5	6
7	8	9

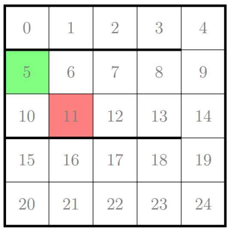
Table 1: 3×3-Image Matrix.

1	0
0	1

Table 2: 2×2-Kernel Matrix.

(b) Assume the input with the size of (width = 28, height = 28) and a filter with the size of (width =5, height = 5) and the convolutional layer parameters are $S = 1$ (the stride), $P = 0$ (the amount of zero padding). What is the exact size of the convolution output? (5 points)

6. Consider the following grid environment. Starting from any unshaded square, you can move up, down, left, or right. Actions are deterministic and always succeed (e.g. going left from state 1 goes to state 0) unless they will cause the agent to run into a wall. The thicker edges indicate walls, and attempting to move in the direction of a wall results in staying in the same square. Taking any action from the green target square (no. 5) earns a reward of +5 and ends the episode. Taking any action from the red square of death (no. 11) earns a reward of -5 and ends the episode. Otherwise, each move is associated with some reward $r \in \{-1, 0, +1\}$. Assume the discount factor $\gamma = 1$ unless otherwise specified.



(a) Define the reward r for all states (except state 5 and state 11 whose rewards are specified above) that would cause the optimal policy to return the shortest path to the green target square (no. 5). (3 points)

(b) Using r from part (a), find the optimal value function for each square. (5 points)

(c) Does setting $\gamma = 0.8$ change the optimal policy? Why or why not? (2 points)