Assignment 2

Due date: Oct 11, 11:59pm

Assignment 2 consists of two parts – writing and coding. Students can write computer programs for both parts to double-check the solution. However, the writing part should list all algorithm steps with intermediate values for full credit. The same type of problems can be expected on the midterm exam. The programming part can be submitted either as IPython Notebooks (recommended) or as stand-alone scripts. Python interpreter and imported libraries should be compatible with the latest Anaconda distribution (hCps://www.anaconda.com/).

Written part (50 points)

1. **(9 points)** You roll two dice. What is the probability of rolling a number greater than eight, if you know the following:
    a) No prior information is given
    b) One dice displays the number 5? The other dice can take any value, *including* number 5.
    c) One dice displays a number larger than 2?

2. **(11 points)** A doctor thinks that a patient has one of the three cancer types: Hodgkin lymphoma (c1), Diffuse Large B-cell lymphoma (c2), or T-cell lymphoma (c3). Before the tests are conducted, they assume an equal probability for each lymphoma subtype. He carries out a test that will be positive with a probability of 0.85 if the patient is afflicted by c1, 0.4 if he has disease c2, and 0.3 if he has cancer c3. The test comes out positive. What are the probabilities for the patient to have a particular disease type? Provide answers for c1, c2, and c3.
3. **(15 points)** Conduct two iterations of the gradient descent algorithm for the linear regression model with intercept (bias column) over the following dataset:

X	Y
3	5
4	6
5	8
6	8

with the initial weights 0.1, 0.1. Assume the learning rate is 0.5.

Report weights for each step and three MSE values – for the initial weights and after each iteration. Show each step of your work for the full credit.

4. **(15 points)** Consider a multi-label classification problem model. You have a training dataset with the following labels:

Y
Green
Blue
Blue
Red
Green
Green
Blue
Green
Green
Green

There is a model that outputs the following logits:

$\widehat{Y}_{Green}$	$\widehat{Y}_{Blue}$	$\widehat{Y}_{Red}$
0.6	0.3	0.1
0.4	0.5	0.1
0.5	0.3	0.2
0.3	0.3	0.4
0.3	0.5	0.2
0.1	0.8	0.1
0.2	0.7	0.1
0.3	0.2	0.5
0.4	0.3	0.3
0.2	0.5	0.3

Compute the cross-entropy of this model. Use logarithm with the base 2 ($\log_2 x$) for calculations. What information can you derive from it? Can we tell whether the model overfits or underfits?

Build a confusion matrix for this problem. Calculate accuracy and F1 score. Are those metrics informative?

Programming part (50 points total)

On the second day of your job, you are asked to investigate a small dataset of the company's products and corresponding quality levels, with *Green* being good, *Red* denoting faulty, and *Orange* being a borderline case. Implement and evaluate simple machine learning classification models – logistic regression and softmax regression. For the logistic regression (binary classification problem) consider only the following split – *Orange* instances against the rest of the dataset.

X_1	X_2	Y
1	0.5	Green
4	2	Orange
2.5	4	Orange
5	0.8	Red
0	1.5	Green
2.3	4	Green
5	2	Orange
1.4	3	Green
1.2	5.2	Green
3.9	4.6	Green

(20 points total) Implement logistic regression. You will have two classes – *Orange* and *Green+Red.* Use binary cross-entropy (BCE) for a loss function.

1. For this problem:

   Set initial weights to [0.1, 0.1, 0.1]
   For optimization, use the following algorithms:
   a. (**5 points**) Regular gradient descent (GD). Use the entire dataset to conduct an optimization step.
   b. (**5 points**) Stochastic gradient descent (SGD). Use the sequential batches of the size two to conduct each optimization step.

   You have to implement *both* approaches.

   (**5 points**) Run 1000 iterations of each optimization algorithm. What is the time difference? Record loss function for each step. Record the F1 score for each step. Report confusion matrix for the last step.

   (**5 points**) Plot loss function over each step for GD and SGD. What do you see?

(20 points total) Implement softmax regression. You will have to apply one-hot encoding to the label vector. For ground truth values, you should construct a binary matrix in the same shape as in Q4 of the writing part. Use binary cross-entropy (BCE) for a loss function.

Set initial weights to [0.1, 0.1, 0.1]
For optimization, use the following algorithms:

a. (**5 points**) Regular gradient descent (GD). Use the entire dataset to conduct an optimization step.
b. (**5 points**) Stochastic gradient descent (SGD). Use the sequential batches of the size two to conduct each optimization step.

You have to implement *both* approaches.

(**5 points**) Run 1000 iterations of each optimization algorithm. What is the time difference? Record loss function for each step. Record the F1 score for each step. Report confusion matrix for the last step.

(**5 points**) Plot loss function over each step for GD and SGD. What do you see?

(10 points) Comparison

Compare performance between logistic regression and softmax regression. What differences between loss function plots do you see? How about F1 score?