Optimization and Machine Learning
Assignment 2

Due Date: 29 March 2023, 5 pm

Instructions

- Please read each question carefully
- Note that the last question is only for groups with four members
- After completing your assignment compress all your files into a **single** zip file and name the file with your group's name (*e.g.*, `OverAchievers.zip`)

Points for The Questions

	3-member groups	**4-member groups**
Question 1	5 pts.	2.5 pts.
Question 2	5 pts.	2.5 pts.
Question 3	10 pts.	10 pts.
Question 3	-	5 pts.

Question 1

In our third lecture, we discussed a simple binary classification example with neural networks. Figure 1 shows this network with selection of activation functions. Using the lecture notes, write down the network function $E(\beta) = \sum_{i=1}^{n} E_i(\beta)$ for this example and derive its partial derivative with respect to $\beta_{12}^{(1)}$ for sample i; *i.e.*, $\partial E_i(\beta)/\partial \beta_{12}^{(1)}$.

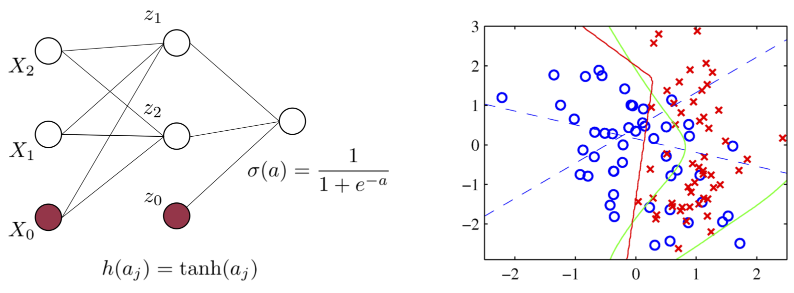

Figure 1: Simple binary classification example.

What will you turn in?

A PDF file showing the steps that you followed to obtain the solution. The name of the file should be your group's name appended with "Q1" (*e.g.*, `OverAchieversQ1.pdf`). Add the renamed file into the final compressed file that you will submit.

Evaluation. This question will be evaluated in terms of correctness and clarity of your steps.

Question 2

Suppose that we train a single hidden layer feedforward neural network for a multinomial classification problem with K classes, using a training sample of n observations. Let x_i denote the p-dimensional input vector, and $y_{ik} = 1$ if observation i belongs to class k, and $y_{ik} = 0$ otherwise (with $i = 1, \dots, n$ and $k = 1, \dots, K$). Assume that we use the softmax function to transform the outputs from the hidden layer, that is,

$$\hat{y}_{ik} = \frac{e^{a_{ik}}}{\sum_{\ell=1}^{K} e^{a_{i\ell}}},$$

where $a_{ik} = \beta_{0k} + \beta_k^\intercal Z_i$ with $Z_i = (Z_{i1}, \dots, Z_{iM})$ and $Z_{im} = \sigma(\alpha_{0m} + \alpha_m^\intercal x_i)$ for some activation function σ. Suppose we use the cross-entropy to train the neural network, that is, we estimate the vector β by minimizing

$$E(\beta) = \sum_{i=1}^{n} E_i(\beta) = -\sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \ln \hat{y}_{ik}.$$

Show that $\partial E_i(\beta)/\partial a_{ij} = \hat{y}_{ij} - y_{ij}$

What will you turn in?

A PDF file showing the steps that you followed to obtain the solution. The name of the file should be your group's name appended with "Q2" (*e.g.*, `OverAchieversQ2.pdf`). Add the renamed file into the final compressed file that you will submit.

Evaluation. This question will be evaluated in terms of correctness and clarity of your steps.

Question 3

Let us consider a dataset consisting of data points $i \in \{1, \dots, n\}$ with features $x_i \in \mathbb{R}^p$ and labels $y_i \in \{-1, +1\}$. Recall that SVM is a powerful approach for binary classification of n labeled data. In a nutshell, SVM is based on solving the following problem:

$$\min_{\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p} \left\{ \frac{1}{2} \|\beta\|_2^2 + C \sum_{i=1}^{n} \max\left\{0, 1 - y_i(\beta^\intercal x_i + \beta_0)\right\} \right\}. \tag{1}$$

Overall this problem has $p + 1$ decision variables ($\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p$). In this question, you will be solving this problem approximately. In all your experiments:

- As initial iterate use standard normally distributed vector.
- Set the maximum number of iterations to 10000.

The objective function of problem (1) is nondifferentiable due to the max operator. An approach is to replace the function $h(z) = \max\{0, 1 - z\}$ with the following differentiable function

$$h_\xi(z) = \begin{cases} 0, & \text{if } z > 1 + \xi; \\ \frac{1}{4\xi}(1 + \xi - z)^2, & \text{if } |1 - z| \leq \xi; \\ 1 - z, & \text{if } z < 1 - \xi. \end{cases} \tag{2}$$

As $\xi \downarrow 0$, the approximation function h_ξ approaches to function h. Set $\xi = 10^{-4}$ and rewrite the objective function of (1) using (2). The resulting objective function will be differentiable and can therefore be minimized using optimization methods that make use of the first order (gradient) information.

1. Write the Python function that solves problem (1) with mini-batch SGD. You can set a fixed learning rate.
2. Compare the generalization performance of your Python function against the generalization performances of `sklearn` packages on SVM with linear, polynomial, and RBF kernels.

What will you turn in?

Add your function to the designated place in the `test_SVMSGD_.py` template script and rename the file after appending your group's name as a **single word** (*e.g.*, `test_SVMSGD_OverAchievers.py`). Add the renamed file into the final compressed file that you will submit.

Evaluation. The results of your implementation will be tested against the `sklearn` SVM packages. The template script `test_SVMSGD_.py` also shows our testing setup as well as the datasets. The clarity and correctness of your implementation along with the comments will determine your score.

Question 4 - Only for groups with four members

On page 43 of our lecture notes on neural networks, we have presented the algorithms for SGD with momentum and Adam. You can try different values for the parameters α, γ, η, γ_1, and γ_2. Implement these methods for the same problem in Question 3. The lecture notes had a typo that has just been fixed. Therefore, please use the most recent version on the shared folder. The subscript i in Adam algorithm refers to the ith component of the corresponding vector; *e.g*, $w_{k+1} = (w_{k+1,1}, \dots w_{k+1,i}, \dots, w_{k+1,p})^\intercal$.

What will you turn in?

Add your function to the designated place in your `test_SVMSGD_OverAchiever.py` file that you prepared for Question 3. Please submit just one file for both Question 3 and Question 4.

Evaluation. The results of your implementation will be tested against the `sklearn` SVM packages and our SGD implementation. The clarity and correctness of your implementation along with the comments will determine your score.