Homework #1

Due: 24th April 2024, Wednesday, before 11:59 pm

Problem 1 (Perceptron)

Suppose we have a training set with 8 samples, each sample has feature vector in $\mathbb{R}^2$:

#	1	2	3	4	5	6	7	8
X	[4,0]	[1,1]	[0,1]	[-2,-2]	[-2,1]	[1,0]	[5,2]	[3,0]
y	1	-1	-1	1	-1	1	-1	-1

We are going to implement the perceptron algorithm to train a linear classifier with 2 dimensional weight vector $\mathbf{w} \in \mathbb{R}^2$ (no bias term). We start with initial weight vector as the first sample in our dataset, i.e. $\mathbf{w}_1 = \mathbf{x}_1$. Note that: when $\mathbf{w}^\top x = 0$, the algorithm predicts $+1$.

To simplify the calculation, you only need to test and possibly update each sample once in the given sequence. You can either implement the algorithm by hand or programming.

(a) Is the data linearly separable? Will our algorithm converge if we run it several times over the same sequence? Explain.

(b) Regardless of whether the dataset is linearly separable or not, calculate the updates of the weight vector on this sequence for one round over the entire dataset. Follow the order of the index for the samples and show your calculations.

(c) Provide closed-form functions for the perceptron, Voted perceptron, and Average perceptron, using the weight vector(s) derived in part (b).

(d) Using the functions derived in part (c), compare the errors between the perceptron, the Voted perceptron predictor, and the Average perceptron predictor across the entire dataset. For each point in the dataset, find the label assigned by each classifier and report the error over the dataset.

Problem 2 (Modified Logistic Regression with alternative labels)

In class, we have seen the logistic regression when labels are {0,1}. In this question, you will derive the logistic regression when labels are instead {-1,1}.

In class, we considered the dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with n samples where $\mathbf{x}_i \in \mathbb{R}^d$ and labels $y_i \in \{0, 1\}$ for all $i \in [n]$. The prediction function $h_{\mathbf{w}}(\mathbf{x})$ studied in class is given by

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}} = \frac{e^{\mathbf{w}^\top \mathbf{x}}}{1+e^{\mathbf{w}^\top \mathbf{x}}}. \tag{1}$$

Moreover, the objective studied in class to minimize for logistic regression was:

$$\text{P2.1}: \qquad \min_{\mathbf{w}} -\sum_{i=1}^{n} \left[y_i \log(h_{\mathbf{w}}(\mathbf{x}_i)) + (1-y_i)\log(1-h_{\mathbf{w}}(\mathbf{x}_i))\right] \tag{2}$$

We want to modify this objective function so that $\tanh_{\mathbf{w}}(\mathbf{x}) = \frac{e^{\mathbf{w}^\top \mathbf{x}} - e^{-\mathbf{w}^\top \mathbf{x}}}{e^{\mathbf{w}^\top \mathbf{x}} + e^{-\mathbf{w}^\top \mathbf{x}}} = \frac{e^{2\mathbf{w}^\top \mathbf{x}} - 1}{e^{2\mathbf{w}^\top \mathbf{x}} + 1}$ is the activation function and $\tilde{y}_i \in \{-1, 1\}$ the labels.

(a) Show that

$$\tanh_{\mathbf{w}}(\mathbf{x}) = 2h_{\mathbf{w}'}(\mathbf{x}) - 1, \ \ \mathbf{w}' = 2\mathbf{w}.$$

(b) What are the asymptotic values of the function $\tanh_{\mathbf{w}}(\mathbf{x})$ as $\mathbf{w}^\top \mathbf{x} \to \infty$ and $\mathbf{w}^\top \mathbf{x} \to -\infty$? Roughly draw the graph of this function with respect to $\mathbf{w}^\top \mathbf{x}$. What is the decision criterion you can choose for predicting labels as -1 or 1?

(c) Using your answer in part (b), argue that we cannot directly replace $h_{\mathbf{w}}(\mathbf{x}_i)$ with $\tanh_{\mathbf{w}}(\mathbf{x})$ in the optimization problem P2.1 (2).

(d) When labels are $\tilde{y}_i \in \{-1, 1\}$ show, using your answer in part (a), that the optimization problem in P2.1 is equivalent to:

$$\text{P2.2}: \qquad \min_{\mathbf{w}} -\sum_{i=1}^{n} \left[\frac{1+\tilde{y}_i}{2} \log\left(\frac{1+\tanh_{\mathbf{w}}(\mathbf{x}_i)}{2}\right) + \frac{1-\tilde{y}_i}{2} \log\left(\frac{1-\tanh_{\mathbf{w}}(\mathbf{x}_i)}{2}\right)\right] \tag{3}$$

(e) Compute the gradient of the loss function in P2.2 (3) for a single sample $\mathbf{x}_i$. Consider the two cases $\tilde{y}_i = 1$ and $\tilde{y}_i = -1$ separately.

Problem 3 (True/False)

In each of these problems, give a clear explanation for your choice.

(a) **Perceptrons:**

i. When the perceptron algorithm encounters an incorrectly classified sample $(\mathbf{x}_i, y_i)$ with weights $\mathbf{w}$, it updates the weights and obtains $\mathbf{w}'$ after the update. $\mathbf{w}'$ correctly classifies $(\mathbf{x}_i, y_i)$.
TRUE / FALSE.

ii. If the data is linearly seperable, then the Rosenblatt perceptron algorithm converges to a solution that makes no errors on the training data.
TRUE / FALSE.

iii. For a perceptron classifier with weights $\mathbf{w}$, the prediction made on a feature $\mathbf{x}$ is $\text{sign}(\mathbf{w}^\top \mathbf{x})$. If we multiply the weights obtained from perceptron by a negative scalar, this flips all predictions made by the perceptron algorithm.
TRUE / FALSE.

(b) **Logistic Regression:**

i. The prediction of a logistic regression classifier is given by $y = \sigma(\mathbf{w}^\top \mathbf{x}) \in [0, 1]$. Moreover, the decision boundary of this classifier is $\mathbf{w}^\top \mathbf{x} \geq 0$.
TRUE / FALSE.

ii. The stochastic gradient descent optimization is more computationally efficient than batch gradient descent per iteration.
TRUE / FALSE.

iii. Let $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$, $(\mathbf{x}, y)$ a sample, and $\sigma(a) = \frac{1}{1+e^{-a}}$. When $y = 0$, it is true that

$$-y \log\left(\sigma(h_{\mathbf{w}}(\mathbf{x}))\right) - (1-y)\log\left(1 - \sigma(h_{\mathbf{w}}(\mathbf{x}))\right) = \log\left(1 + e^{h_{\mathbf{w}}(\mathbf{x})}\right) - 1_{y=1} h_{\mathbf{w}}(\mathbf{x})$$

TRUE / FALSE.

iv. Let $\sigma(a) = \frac{1}{1+e^{-a}}$, then the derivative is $\frac{d\sigma(a)}{da} = \frac{-1+e^{-a}}{(1+e^{-a})^2}$.
TRUE / FALSE.

(c) **Linear regression:**

i. A closed form solution for linear regression is only possible if $\boldsymbol{X}^\top \boldsymbol{X}$ is positive definite. Note that a matrix $\boldsymbol{A}$ is positive definite if $\mathbf{z}^\top \boldsymbol{A} \mathbf{z} > 0$ for all vectors $\mathbf{z} \neq \mathbf{0}$
TRUE / FALSE.

ii. The loss function of ℓ_2 regularized linear regression is $\|\boldsymbol{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$. The gradient of the loss function with respect to $\mathbf{w}$ is given by $2\boldsymbol{X}^\top(\boldsymbol{X}\mathbf{w} - \mathbf{y}) + 2\lambda\mathbf{w}$.
TRUE / FALSE.

Problem 4 (Programming Exercise: Binary Classification)

In this exercise, you will work through a family of binary classifications. Our data consists of inputs $x_n \in \mathbb{R}^{1\times d}$ and labels $y_n \in \{-1, 1\}$ for $n \in \{1, \ldots, N\}$. We will work on a subset of the

Fashion-MNIST dataset which focuses on classifying whether the image is for a *Dress* $(y = 1)$ or a *Shirt* $(y = -1)$. Your goal is to learn a classifier based on linear predictor $h_{\mathbf{w}}(x) = \mathbf{w}^T x$. Let

$$\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{N \times d}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \in \{1, -1\}^N \tag{4}$$

The main file is the Notebook Jupyter notebook.

(a) (**Visualization**): Visualize a sample of the training data. What is the dimensions of X_{train}, and X_{test}.

(b) (**Perceptron**): Implement Perceptron Algorithm to classify your training data. Let the maximum number of iterations of the Algorithm $num_{iter} = N$ (number of training samples). At each iteration, compute the percentage of misclassified points in the training dataset, and save it into a `Loss_hist` array. Plot the history of the loss function (`Loss_hist`). What is the final value of the loss function and the squared ℓ_2 norm value of the weight $||\mathbf{w}||_2^2$? Looking at the loss function, can you comment on whether the Perceptron algorithm converges?

(c) (**Perceptron test error**): Compute the percentage of misclassified points in the test data for the trained Perceptron.

(d) (**Logistic Regression**): In this part, we will implement the logistic regression for binary classification. Recall that logistic regression attempts to minimize the objective function

$$J(\mathbf{w}) = \frac{1}{N}\left(\sum_{n=1}^{N} \log\left(1 + e^{h_{\mathbf{w}}(\mathbf{x}_n)}\right) - \sum_{n=1}^{N} \mathbf{1}_{y_n=1} h_{\mathbf{w}}(\mathbf{x}_n)\right) \tag{5}$$

where $\mathbf{x}_n = (1, x_n)$, and $\mathbf{1}_A = 1$ if A is true and 0 otherwise. Moreover, $h_{\mathbf{w}}(\mathbf{x}_n) = \mathbf{w}^T \mathbf{x}_n$. First, we will add an additional *feature* to each instance and set it to one. This is equivalent to adding an additional first column to $\mathbf{X}$ and setting it to all ones.
Modify the `get_features()` in Logistic.py file to create a matrix $\mathbf{X}$ for logistic regression model.

(e) Complete `predict()` in Logistic.py file to predict $\mathbf{y}$ from $\mathbf{X}$ and $\mathbf{w}$.

(f) Complete the function `loss_and_grad()` to compute the loss function and the gradient of the loss function with respect to $\mathbf{w}$ for a data set $\mathbf{X}$ and labels $\mathbf{y}$ at given weights $\mathbf{w}$. Test your results by running the code in the main file Notebook.ipynb. If you implement everything correctly, you should get the loss function within 0.7 and squared ℓ_2 norm of the gradient around 1.8×10^5.

(g) Complete the function `train_LR()` to train the logistic regression model for given learning rate $\eta = 10^{-6}$, `batch_size` $= 100$, and number of iterations $num_{iters} = 5000$. Plot the history of the loss function (Loss_hist). What is the final value of the loss function and the squared ℓ_2 norm value of the weight $||\mathbf{w}||_2^2$?

(h) (**Logistic Regression test error**): Compute the percentage of misclassified points in the test data for the trained Logistic Regression.