Homework 1

Spring 22, CS 442: Trustworthy Machine Learning
Due Friday Feb. 25th at 23:59 CT

Instructor: Han Zhao

Instructions for submission All the homework submissions should be typeset in LaTeX. For all the questions, please clearly justify each step in your derivations or proofs.

1 Matrix Calculus [10pts]

Given a dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$ are the labeled sample used for training a logistic regression model, we would like to derive the gradient with respect to the model parameter $w \in \mathbb{R}^d$. Recall that the loss function of logistic regression is given as follows:

$$\min_w \quad \mathcal{L}(w) := -\sum_{i=1}^n y_i \log \sigma(w \cdot x_i) + (1 - y_i) \log \sigma(-w \cdot x_i),$$

where $\sigma(t) = 1/(1 + \exp(-t))$ is the sigmoid function. Derivate the gradient of the loss function $\mathcal{L}(w)$ with respect to w, $\nabla \mathcal{L}(w)$. Could you simplify the gradient using matrix notation?

2 Feed-forward Neural Networks [30 pts]

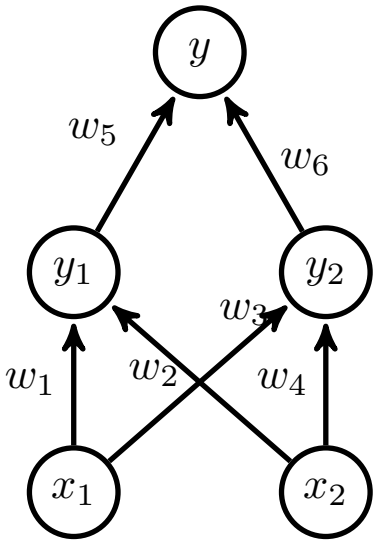
Consider a feed-forward neural network (FNN) with one input layer, one hidden layer and one output layer shown in Fig. 1a. In this problem we will use the FNN to classify the XOR pattern shown in Fig. 1b. Suppose that the activation function at every unit in the FNN are linear, i.e., $y_1 = w_1 x_1 + w_2 x_2$, $y_2 = w_3 x_1 + w_4 x_2$ and $y = w_5 y_1 + w_6 y_2 + w_0$. Classify the input instance (x_1, x_2) as $+1$ if $y(x_1, x_2) \geq 0$ otherwise -1.

2.1 [10pts]

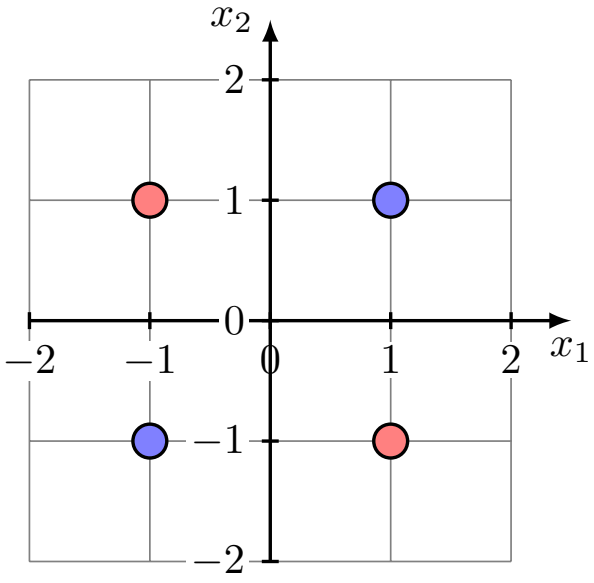
For any weight configuration $(w_0, w_1, w_2, w_3, w_4, w_5, w_6)$ of the FNN shown above, can you find another FNN with only two layers, i.e., one input layer with two inputs x_1, x_2 and one output layer with one output unit y that computes the same function? If yes, describe a such two-layer FNN and give the weights as functions of $(w_0, w_1, w_2, w_3, w_4, w_5, w_6)$. If no, briefly describe your reasoning.

2.2 [10pts]

Show that there is no weight configuration $(w_0, w_1, w_2, w_3, w_4, w_5, w_6)$ under which the FNN shown in Fig. 1a can classify the XOR pattern with no error.



(a) A three-layer feed-forward neural network with two inputs.



(b) XOR pattern. Instances in blue have label $+1$ while instances in red have label -1.

2.3 [10pts]

For the FNN shown in Fig. 1a, will changing the activation functions at y_1 and y_2 to be nonlinear help classify the XOR pattern? If yes, construct a nonlinear activation function $f(\cdot)$ to be applied only at y_1 and y_2, i.e., $y_1(x_1, x_2) = f(w_1x_1 + w_2x_2)$, $y_2(x_1, x_2) = f(w_3x_1 + w_4x_2)$, such that the new FNN can perfectly classify the XOR pattern. If no, briefly describe your reasoning on why it is not possible.

3 The Price of Statistical Parity [30pts]

Consider a binary classification problem with two groups. Let (X, A, Y) be the tuple drawn from an underlying distribution μ. For simplicity, in this problem we assume all the variables are binary, i.e., $X, A, Y \in \{0, 1\}$. Recall from the lecture, in this example we use A to denote the group membership of an instance, i.e., $A = 0$ means the majority group whereas $A = 1$ means the minority group. More concretely, let $p \geq 1/2$ be the marginal probability of $A = 0$: $\Pr_\mu(A = 0) = p$. To complete the specification of the joint distribution μ over (X, A, Y), the group-wise distributions over the pair (X, Y) are given as follows:

- For each group $A = a \in \{0, 1\}$, the conditional probability $\Pr_\mu(X = 1 \mid A = a) = 1/2$ is uniform, i.e., with equal probabilities, X can take either 0 or 1.
- For each group $A = a \in \{0, 1\}$, $\Pr_\mu(Y = a \mid A = a) = 1$, i.e., with probability 1 the value of the target Y equals a.

3.1 [10pts]

Show that there exists a deterministic classifier $h : \{0,1\} \times \{0,1\} \to \{0,1\}$, taking the pair (x, a) as input and predicts the corresponding label, is simultaneously perfect on both groups. Construct such a deterministic classifier. Note: we say a classifier to be perfect if it achieves 0 classification error on the corresponding distribution.

3.2 [10pts]

Now we consider a randomized classifier h as follows. Upon receiving the input pair (x, a), a randomized classifier $h(x, a)$ will flip a fair coin. Depending on the outcome of the coin, the randomized classifier $h(x, a)$ will make the following prediction:

$$h(x,a) = \begin{cases} 0 & \text{If the outcome of the fair coin is H} \\ 1 & \text{If the outcome of the fair coin is T.} \end{cases}$$

3.2.1 [5pts]

What is the classification error rate of this randomized classifier on the joint distribution μ? i.e., what is $\mathbb{E}_{(X,A,Y)\sim\mu}[h(X, A) \neq Y]$?

3.2.2 [5pts]

Show that despite taking the group membership A explicitly as its input, the above randomized classifier satisfies statistical parity.

3.3 [10pts]

3.3.1 [8pts]

Prove that, for any deterministic classifier $h(X, A)$[1], if $h(X, A)$ satisfies statistical parity, then

$$\mathbb{E}_\mu[h(X, A) \neq Y \mid A = 0] + \mathbb{E}_\mu[h(X, A) \neq Y \mid A = 1] = 1.$$

3.3.2 [2pts]

Using the result above, show that the overall error rate of h over μ is at least $1 - p$, i.e., $\mathbb{E}_\mu[h(X, A) \neq Y] \geq 1 - p$.

4 The Implicit Bias of Gradient Descent in Linear Regression [30 pts]

Consider a dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ is the i-th input and $y_i \in \mathbb{R}^d$ is the i-th label. The loss function for linear regression on this dataset is given by

$$\min_{w\in\mathbb{R}^d} \quad \mathcal{L}(w) := \frac{1}{2}\sum_{i=1}^{n}(w \cdot x_i - y_i)^2, \tag{1}$$

[1]This actually also applies to randomized classifiers as well, but in this problem you only need to show this for deterministic classifiers.

where $w \in \mathbb{R}^d$ is the model parameter of interest. In the course we find the optimal solution w^* via solving the *normal equation*. In this problem, instead, we will use the gradient descent method to numerically find the optimal solution instead. In particular, in order to solve (1), we apply the following algorithm:

Procedure 1 Gradient Descent for Linear Regression

Input: Initial model parameter w_0

1: **for** $t = 1, 2, \ldots$ until convergence **do**
2: $\quad w_t \leftarrow w_{t-1} - \frac{1}{t}\nabla_w \mathcal{L}(w_{t-1})$
3: **end for**
4: **return** w^*

In this problem, let's assume that the above algorithm will converge, and we use w^* to denote the convergent point.

4.1 [10pts]

Prove that $w^* - w_0 \in \text{span}\{x_1, \ldots, x_n\}$.

4.2 [10pts]

Let $X \in \mathbb{R}^{n \times d}$ be the data matrix where the i-th row of this matrix is given by $x_i^\top$, and let $y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ be the label vector.

4.2.1 [5pts]

Show that w^* satisfies the normal equation, i.e., $X^\top X w^* = X^T y$.

4.2.2 [5pts]

For any vector $\widehat{w} \in \mathbb{R}^d$ that is the optimal solution of (1), show that $w^* - \widehat{w} \in \text{Ker}(X)$, where $\text{Ker}(\cdot)$ denotes the kernel of a matrix.

4.3 [10pts]

Prove that among all the model parameters that optimize (1), w^* has the minimum distance to w_0. Formally, let $W := \{\widehat{w} \in \mathbb{R}^d : \widehat{w} \text{ is an optimal solution to (1)}\}$. Prove that $w^* = \arg\min_{w \in W} \|w - w_0\|_2$. Note: this means that gradient descent finds the optimal solution that is closest to the initial point.