

Homework 1

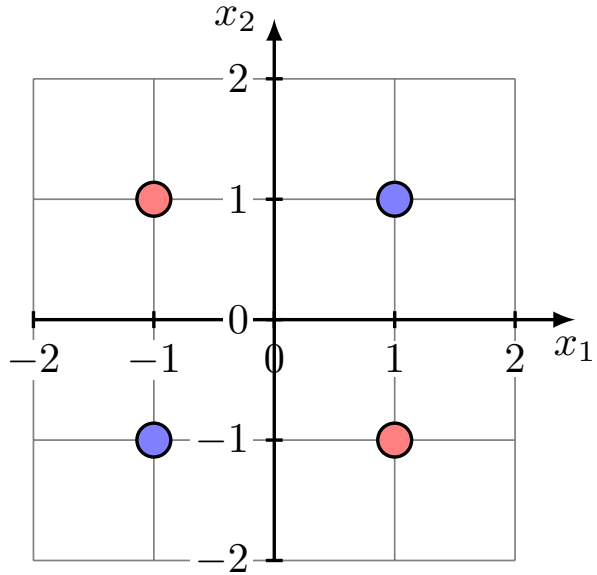
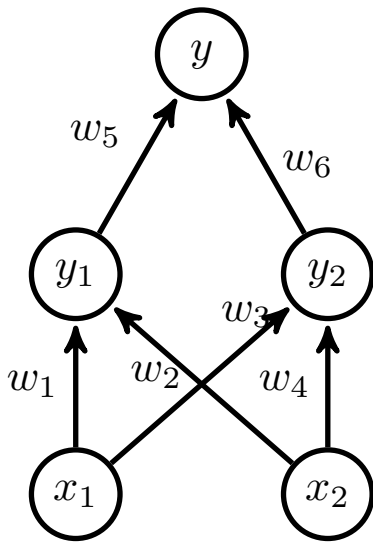
Spring 25, CS 442: Trustworthy Machine Learning
Due Friday Feb. 21st at 23:59 CT

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 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., $y_1 = w_1x_1 + w_2x_2$, $y_2 = w_3x_1 + w_4x_2$ and $y = w_5y_1 + w_6y_2 + w_0$. Classify the input instance (x_1, x_2) as $+1$ if $y(x_1, x_2) \geq 0$ otherwise -1 .



(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 .

1.1 [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.

1.2 [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.

1.3 [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.

2 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 .

2.1 [10pts]

Show that there exists a deterministic classifier $h : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$, 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.

2.2 [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:

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

2.2.1 [5pts]

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

2.2.2 [5pts]

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

2.3 [10pts]

2.3.1 [8pts]

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

$$\Pr_{\mu}(h(X, A) \neq Y \mid A = 0) + \Pr_{\mu}(h(X, A) \neq Y \mid A = 1) = 1.$$

2.3.2 [2pts]

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

3 Matrix Calculus [10pts]

A classic problem in unsupervised machine learning is known as *symmetric matrix factorization*. Given a matrix $P \in \mathbb{R}^{n \times n}$, the goal is to find a matrix $X \in \mathbb{R}^{n \times k}$ such that $P \approx XX^{\top}$. In this problem, we will derive the gradient of the following objective function with respect to X :

$$\min_{X \in \mathbb{R}^{n \times k}} \mathcal{L}(X) := \frac{1}{2} \|P - XX^{\top}\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, i.e., $\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n A_{ij}^2$ for $A \in \mathbb{R}^{m \times n}$. One typical application of the above problem is to find an embedding of n objects into a k -dimensional space, where P_{ij} denotes the similarity between the i -th and j -th objects. The above objective function encourages the similarity between the i -th and j -th objects to be approximated by the inner product between the i -th and j -th rows of X .

Please derive the gradient of $\mathcal{L}(X)$ with respect to X . Note: you need to show your derivation step by step, and you need to express the gradient in matrix notation in terms of X and P only.

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

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 $y_i \in \mathbb{R}$ is the i -th label. The loss function for linear regression on this dataset is given by

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

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** $t = 1, 2, \dots$ until convergence **do**
 - 2: $w_t \leftarrow w_{t-1} - \frac{1}{t} \nabla_w \mathcal{L}(w_{t-1})$
 - 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 $w^* - w_0 \in \text{span}\{x_1, \dots, x_n\}$.

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 $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ be the label vector.

4.2.1 [5pts]

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

4.2.2 [5pts]

For any vector $\hat{w} \in \mathbb{R}^d$ that is the optimal solution of (1), show that $w^* - \hat{w} \in \text{Ker}(X)$, 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 $W := \{\hat{w} \in \mathbb{R}^d : \hat{w} \text{ is an optimal solution to (1)}\}$. 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.