INFR10086 Machine Learning (MLG) Semester 2, 2024/5

Practice Exam

1. In this question, we will look at the hinge loss for binary classification. Recall that a linear classifier has the form

$$f(x) = \begin{cases} +1 & \text{if } w^\top x \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

The hinge loss for binary classification with linear classifier is defined as

$$L_{\text{hinge}}(x, y; w) = \max(1 - yw^\top x, 0), \tag{2}$$

where $x \in \mathbb{R}^d$ and $y \in \{+1, -1\}$.

(a) Show that the hinge loss is an upper bound of the zero-one loss

$$L_{01}(x, y; w) = \mathbb{1}_{yw^\top x < 0}. \tag{3}$$

In other words, show that

$$L_{01}(x, y; w) \leq L_{\text{hinge}}(x, y; w) \tag{4}$$

for all $x \in \mathbb{R}^d$, $y \in \{+1, -1\}$, and $w \in \mathbb{R}^d$.

> First note that $L_{\text{hinge}} \geq 0$ for all w, x, and y. When $yw^\top x \geq 0$, $L_{01} = 0$ and $L_{\text{hinge}} \geq 0 = L_{01}$. When $yw^\top x < 0$, $L_{01} = 1$ and $L_{\text{hinge}} = 1 - yw^\top x \geq 1 = L_{01}$.

(b) In the following three steps, we will look at the convexity of hinge loss.

(i) Show that

$$\max(a + b, c + d) \leq \max(a, c) + \max(b, d) \tag{5}$$

for any $a, b, c, d \in \mathbb{R}$.

$$\begin{aligned} \max(a + b, c + d) &\leq \max(\max(a, c) + b, c + d) \\ &\leq \max(\max(a, c) + b, \max(a, c) + d) \\ &\leq \max(\max(a, c) + \max(b, d), \max(a, c) + d) \\ &\leq \max(\max(a, c) + \max(b, d), \max(a, c) + \max(b, d)) \\ &= \max(a, c) + \max(b, d) \end{aligned}$$

(ii) Let

$$h(x) = \max(f(x), g(x)) \tag{6}$$

for any two convex functions f and g. Use (b) and show that h is convex in x.

For any $0 \leq \alpha \leq 1$,

$$\begin{aligned} h(\alpha x + (1-\alpha)y) &= \max(f(\alpha x + (1-\alpha)y), g(\alpha x + (1-\alpha)y)) \\ &\leq \max(\alpha f(x) + (1-\alpha)f(y), \alpha g(x) + (1-\alpha)g(y)) \\ &\leq \max(\alpha f(x), \alpha g(x)) + \max((1-\alpha)f(y), (1-\alpha)g(y)) \\ &= \alpha \max(f(x), g(x)) + (1-\alpha)\max(f(y), g(y)) \\ &= \alpha h(x) + (1-\alpha)h(y) \end{aligned}$$

By definition, $h(x)$ is convex in x.

(iii) Use (c) and show that the hinge loss L_{hinge} is convex in w for any $x \in \mathbb{R}^d$ and $y \in \{+1, -1\}$.

A constant function is convex, and $1 - yw^\top x$ is an affine function of w, hence, also convex in w. The hinge loss is a max of two convex functions, hence convex.

(c) If we happen to find a linear classifier that achieves a hinge loss of 0 on a data set, what does that tell us about the optimal value of log loss on that particular data set?

Suppose there are n data points in the data set. Since the hinge loss is an upper bound of the zero-one loss, when the hinge loss is 0, we know that the zero-one loss must also be 0, The data is hence separable, and there exists a w^* such that $y_i {w^*}^\top x_i \geq 0$ for all data points $i = 1, \ldots, n$. Since $y_i {w^*}^\top x_i \geq 0$, we have $y_i (aw^*)^\top x_i \geq 0$ for any $a > 0$. For log loss $L(w) = \sum_{i=1}^n \log(1 + \exp(-y_i w^\top x_i))$, we can plug in aw^* and let $a \to \infty$. The term $\exp(-y_i (aw^*)^\top x_i) \to 0$ when $a \to \infty$, and the log loss goes to 0. In other words, when the data is separable, we can achieve a log loss of 0.

2. In this question, we are going to implement a layer called layer normalization in a neural network library. Formally, layer normalization is a function

$$f(x) = \begin{bmatrix} \frac{x_1 - \mu}{\sigma} \\ \frac{x_2 - \mu}{\sigma} \\ \vdots \\ \frac{x_d - \mu}{\sigma} \end{bmatrix} \tag{7}$$

where

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i \qquad \sigma^2 = \frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2 \tag{8}$$

(a) Show that

$$\sigma^2 = \frac{1}{d}\sum_{i=1}^{d} x_i^2 - \mu^2. \tag{9}$$

$$\begin{aligned}\sigma^2 &= \frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2 = \frac{1}{d}\sum_{i=1}^{d}(x_i^2 - 2\mu x_i + \mu^2) \\ &= \frac{1}{d}\sum_{i=1}^{d} x_i^2 - 2\mu\frac{1}{d}\sum_{i=1}^{d} x_i + \frac{1}{d}\sum_{i=1}^{d}\mu^2 \\ &= \frac{1}{d}\sum_{i=1}^{d} x_i^2 - 2\mu^2 + \mu^2 = \frac{1}{d}\sum_{i=1}^{d} x_i^2 - \mu^2\end{aligned}$$

(b) The forward function is as defined, and is straightforward to implement. The backward function (as part of the backpropagation) is more involved. Given the forward computation, the backward computation can be worked out using the total derivative

$$\frac{\partial L}{\partial x_j} = \sum_{i=1}^{d}\frac{\partial L}{\partial f_i}\frac{\partial f_i}{\partial x_j} + \frac{\partial L}{\partial \mu}\frac{\partial \mu}{\partial x_j} + \frac{\partial L}{\partial \sigma}\frac{\partial \sigma}{\partial x_j}, \tag{10}$$

where f_i is a shorthand for the i-th coordinate of $f(x)$ and L is the loss function. Note that $\partial L/\partial f_i$ will be given during backpropagation. Our goal is the derive the rest of the terms.

i. Show that

$$\frac{\partial \mu}{\partial x_j} = \frac{1}{d}. \tag{11}$$

$$\frac{\partial \mu}{\partial x_j} = \frac{1}{d}\sum_{i=1}^{d}\frac{\partial}{\partial x_j}x_i = \frac{1}{d}\sum_{i=1}^{d}\mathbb{1}_{i=j} = \frac{1}{d}$$

ii. Show that

$$\frac{\partial f_i}{\partial x_j} = \frac{1}{\sigma}\mathbb{1}_{i=j}, \tag{12}$$

where $\mathbb{1}_c$ is 1 when c is true and 0 otherwise.

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial}{\partial x_j}\frac{x_i - \mu}{\sigma} = \frac{\mathbb{1}_{i=j} - 0}{\sigma} = \frac{1}{\sigma}\mathbb{1}_{i=j}$$

iii. Show that

$$\frac{\partial \sigma}{\partial x_j} = \frac{1}{\sigma d} x_j \tag{13}$$

$$\frac{\partial \sigma}{\partial x_j} = \frac{1}{2}\left(\frac{1}{d}\sum_{i=1}^{d} x_i^2 - \mu^2\right)^{-1/2}\left(\frac{2}{d}x_j\right) = \frac{1}{2}\frac{1}{\sigma}\left(\frac{2}{d}x_j\right) = \frac{1}{\sigma d}x_j$$

iv. Show that

$$\frac{\partial L}{\partial \sigma} = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\left(-\frac{x_i - \mu}{\sigma^2}\right). \tag{14}$$

$$\frac{\partial L}{\partial \sigma} = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\frac{\partial f_i}{\partial \sigma} = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}(x_i - \mu)\frac{-1}{\sigma^2} = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\left(-\frac{x_i - \mu}{\sigma^2}\right)$$

v. Show that

$$\frac{\partial L}{\partial \mu} = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\left(\frac{-1}{\sigma}\right) + \frac{\partial L}{\partial \sigma}\left(-\frac{\mu}{\sigma}\right). \tag{15}$$

$$\begin{aligned}\frac{\partial L}{\partial \mu} &= \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\frac{\partial f_i}{\partial \mu} + \frac{\partial L}{\partial \sigma}\frac{\partial \sigma}{\partial \mu} = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\left(\frac{0-1}{\sigma}\right) + \frac{\partial L}{\partial \sigma}\frac{1}{2}\left(\frac{1}{d}\sum_{i=1}^{d} x_i^d - \mu^2\right)^{-1/2}(-2\mu) \\ &= \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\left(\frac{-1}{\sigma}\right) + \frac{\partial L}{\partial \sigma}\frac{1}{2\sigma}(-2\mu) = \sum_{i=1}^{d} \frac{\partial L}{\partial f_i}\left(\frac{-1}{\sigma}\right) + \frac{\partial L}{\partial \sigma}\left(-\frac{\mu}{\sigma}\right)\end{aligned}$$

3. Suppose we have a data set organized as a matrix X where each row vector is a sample point.

We know that the first principal component of X is a vector w_1 such that

$$w_1 = \underset{w}{\text{argmax}} \frac{w^\top X^\top X w}{w^\top w} \tag{16}$$

(a) Show that if w_1 is the optimal solution for $\max_w \frac{w^\top X^\top X w}{w^\top w}$, then aw_1 is also an optimal solution for any $a \neq 0$.

Since

$$\frac{(aw_1)^\top X^\top X (aw_1)}{(aw_1)^\top (aw_1)} = \frac{a^2 w_1^\top X^\top X w_1}{a^2 w_1^\top w_1} = \frac{w_1^\top X^\top X w_1}{w_1^\top w_1}$$

we conclude that aw_1 attains the same value as w_1; hence optimal.

(b) Suppose we rotate the entire data set by a rotation matrix R, where $RR^\top = \mathrm{I}$. Show that if w_1 is the first principal component of X, then $R^\top w_1$ is the first principal component of the rotated data set XR.

Since

$$\frac{(R^\top w)^\top (XR)^\top (XR)(R^\top w)}{(R^\top w)^\top (R^\top w)} = \frac{w^\top R R^\top X^\top X R R^\top w}{w^\top R R^\top w} = \frac{w^\top X^\top X w}{w^\top w}$$

for any w, the variance does not change after rotation. If w_1 is the optimal solution when the data matrix is X, then, $R^\top w_1$ is the optimal solution when the data matrix is XR.