ECE421: Introduction to Machine Learning — Fall 2024

Worksheet 2 Solution: Gradient, Logistic Regression, and Non-linear Transformation

Q1 (Gradient Computation) For a scalar-valued function $f : \mathbb{R}^d \to \mathbb{R}$, the gradient evaluated at $\underline{w} \in \mathbb{R}^d$ is

$$\nabla f(\underline{w}) = \begin{bmatrix} \frac{\partial f(\underline{w})}{\partial w_1} & \cdots & \frac{\partial f(\underline{w})}{\partial w_d} \end{bmatrix}^\top \in \mathbb{R}^d.$$

Using this definition, compute the gradients of the following functions, where $A \in \mathbb{R}^{d\times d}$ is not necessarily a symmetric matrix.

1.a $f(\underline{w}) = \underline{w}^\top A\underline{v} + \underline{w}^\top A^\top \underline{v} + \underline{v}^\top A\underline{w} + \underline{v}^\top A^\top \underline{w}$, where $\underline{v} \in \mathbb{R}^d$.

Answer. $\nabla f(\underline{w}) = A\underline{v} + A^\top \underline{v} + A^\top \underline{v} + A\underline{v} = 2(A + A^\top)\underline{v}$.

1.b $f(\underline{w}) = \sum_{i=1}^d \log(1 + \exp(w_i))$

Answer. $\frac{\partial}{\partial w_j} f(\underline{w}) = \sum_{i=1}^d \frac{\partial}{\partial w_j} \log(1 + \exp(w_i)) = \frac{\partial}{\partial w_j} \log(1 + \exp(w_j)) = \frac{\exp(w_j)}{1+\exp(w_j)} = \frac{1}{1+\exp(-w_j)}$. Thus,

$$\nabla f(\underline{w}) = \begin{bmatrix} \frac{\partial}{\partial w_1} f(\underline{w}) \\ \vdots \\ \frac{\partial}{\partial w_d} f(\underline{w}) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+\exp(-w_1)} \\ \vdots \\ \frac{1}{1+\exp(-w_d)} \end{bmatrix}.$$

1.c $f(\underline{w}) = \sqrt{1 + \|\underline{w}\|_2^2}$

Answer.

$$\nabla f(\underline{w}) = \frac{\nabla(1 + \|\underline{w}\|_2^2)}{2\sqrt{1 + \|\underline{w}\|_2^2}} = \frac{\nabla(\|\underline{w}\|_2^2)}{2\sqrt{1 + \|\underline{w}\|_2^2}} = \frac{\nabla(\underline{w}^\top \underline{w})}{2\sqrt{1 + \|\underline{w}\|_2^2}} = \frac{\nabla(\underline{w}^\top I\underline{w})}{2\sqrt{1 + \|\underline{w}\|_2^2}} = \frac{2I\underline{w}}{2\sqrt{1 + \|\underline{w}\|_2^2}} = \frac{\underline{w}}{\sqrt{1 + \|\underline{w}\|_2^2}}$$

Q2 (Logistic Regression) You are given a dataset $\mathcal{D} = \{(\underline{x}_n, y_n)\}_{n=1}^N$, where $\underline{x}_n \in \mathbb{R}^{d+1}, d \geq 1$, and $y_n \in \{+1, -1\}$. For $\underline{w} \in \mathbb{R}^{d+1}$ and $\underline{x} \in \mathbb{R}^{d+1}$, we wish to train a logistic regression model $h(\underline{x}) = \theta(\underline{w}^\top \underline{x})$, where $\theta(.)$ is the logistic function defined as $\theta(z) = \frac{e^z}{1+e^z}$, for $z \in \mathbb{R}$. Following the arguments on page 91 of LFD, the in-sample error can be written as $E_{\text{in}}(\underline{w}) = \frac{1}{N}\sum_{n=1}^N \log\left[\frac{1}{P_{\underline{w}}(y_n|\underline{x}_n)}\right]$ where

$$P_{\underline{w}}(y \mid \underline{x}) = \begin{cases} h(\underline{x}), & \text{if } y = +1, \\ 1 - h(\underline{x}), & \text{if } y = -1. \end{cases}$$

2.a Show that $E_{\text{in}}(\underline{w})$ can be expressed as

$$E_{\text{in}}(\underline{w}) = \frac{1}{N}\left(\sum_{n=1}^N \mathbb{I}(y_n = +1)\log\left[\frac{1}{h(\underline{x}_n)}\right] + \mathbb{I}(y_n = -1)\log\left[\frac{1}{1 - h(\underline{x}_n)}\right]\right),$$

where $\mathbb{I}$(argument) evaluates to 1 if the argument is true and 0 if it is false.

Answer. Observe that for any $n \in \{1, 2, \ldots, N\}$,

$$\begin{aligned} \mathbb{I}(y_n = +1)\log\left[\frac{1}{h(\underline{x}_n)}\right] + \mathbb{I}(y_n = -1)\log\left[\frac{1}{1 - h(\underline{x}_n)}\right] &= \begin{cases} \log\left[\frac{1}{h(\underline{x}_n)}\right], & \text{if } y_n = +1, \\ \log\left[\frac{1}{1-h(\underline{x}_n)}\right], & \text{if } y_n = -1, \end{cases} \\ &= \begin{cases} \log\left[\frac{1}{P_{\underline{w}}(+1|\underline{x})}\right], & \text{if } y_n = +1, \\ \log\left[\frac{1}{P_{\underline{w}}(-1|\underline{x})}\right], & \text{if } y_n = -1, \end{cases} \\ &= \log\left[\frac{1}{P_{\underline{w}}(y_n \mid \underline{x})}\right] \end{aligned}$$

Thus,

$$E_{\text{in}}(\underline{w}) = \frac{1}{N}\left(\sum_{n=1}^{N} \log\left[\frac{1}{P_{\underline{w}}(y_n \mid \underline{x})}\right]\right) = \frac{1}{N}\left(\sum_{n=1}^{N} \mathbb{I}(y_n = +1)\log\left[\frac{1}{h(\underline{x}_n)}\right] + \mathbb{I}(y_n = -1)\log\left[\frac{1}{1-h(\underline{x}_n)}\right]\right).$$

2.b Show that $E_{\text{in}}(\underline{w})$ can also be expressed as

$$E_{\text{in}}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} \log(1 + \exp(-y_n \underline{w}^\top \underline{x}_n)).$$

Answer. Observe that $P_{\underline{w}}(y \mid \underline{x}) = \begin{cases} h(\underline{x}), & \text{if } y = +1, \\ 1 - h(\underline{x}), & \text{if } y = -1 \end{cases}$. Substituting $h(\underline{x})$,

$$P_{\underline{w}}(y \mid \underline{x}) = \begin{cases} \theta(\underline{w}^\top \underline{x}), & \text{if } y = +1, \\ 1 - \theta(\underline{w}^\top \underline{x}), & \text{if } y = -1 \end{cases} = \begin{cases} \frac{e^{\underline{w}^\top \underline{x}}}{1+e^{\underline{w}^\top \underline{x}}}, & \text{if } y = +1, \\ 1 - \frac{e^{\underline{w}^\top \underline{x}}}{1+e^{\underline{w}^\top \underline{x}}}, & \text{if } y = -1 \end{cases} = \begin{cases} \frac{1}{1+e^{-\underline{w}^\top \underline{x}}}, & \text{if } y = +1, \\ \frac{1}{1+e^{\underline{w}^\top \underline{x}}}, & \text{if } y = -1 \end{cases}$$
$$= \frac{1}{1 + e^{-y\underline{w}^\top \underline{x}}}.$$

Thus, $E_{\text{in}}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} \log\left[\frac{1}{P_{\underline{w}}(y_n|\underline{x}_n)}\right] = \frac{1}{N}\sum_{n=1}^{N} \log(1 + e^{-y\underline{w}^\top \underline{x}})$.

2.c Use **2.b** to show that

$$\nabla E_{\text{in}}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} -y_n \underline{x}_n \theta(-y_n \underline{w}^\top \underline{x}_n),$$

and argue that a "misclassified" example contributes more to the gradient than a correctly classified one.

Answer.

$$\nabla E_{\text{in}}(\underline{w}) = \nabla \frac{1}{N}\sum_{n=1}^{N} \log(1 + e^{-y_n \underline{w}^\top \underline{x}_n}) = \frac{1}{N}\sum_{n=1}^{N} \nabla \log(1 + e^{-y_n \underline{w}^\top \underline{x}_n})$$
$$= \frac{1}{N}\sum_{n=1}^{N} \frac{\nabla e^{-y_n \underline{w}^\top \underline{x}_n}}{1 + e^{-y_n \underline{w}^\top \underline{x}_n}} = \frac{1}{N}\sum_{n=1}^{N} \frac{e^{-y_n \underline{w}^\top \underline{x}_n}(-y_n \underline{x}_n)}{1 + e^{-y\underline{w}^\top \underline{x}}} = \frac{1}{N}\sum_{n=1}^{N} -y_n \underline{x}_n \theta(-y_n \underline{w}^\top \underline{x}_n). \quad (1)$$

Consider two arbitrary points $(\underline{x}_n, y_n)$ and $(\underline{x}_n, y_{n'})$. To have fair comparison between the contribution of the two points to the gradient, let's assume that $\|x_n\| = \|x_{n'}\|$. Assume that $\underline{x}_n$ is correctly classified by the weight parameters $\underline{w}$ but $\underline{x}_{n'}$ is misclassified. Therefore, $y_n \underline{w}^\top \underline{x}_n > 0$ and $y_{n'} \underline{w}^\top \underline{x}_{n'} \leq 0$. Consequently, $\theta(-y_n \underline{w}^\top \underline{x}_n) < \theta(-y_{n'} \underline{w}^\top \underline{x}_{n'})$. By the derivative derived in (1), $\theta(-y_n \underline{w}^\top \underline{x}_n)$ and $\theta(-y_{n'} \underline{w}^\top \underline{x}_{n'})$ represent the contribution of $(\underline{x}_n, y_n)$ and $(\underline{x}_{n'}, y_{n'})$ to the gradient, respectively. Note that $\theta(-y_n \underline{w}^\top \underline{x}_n) < \theta(-y_{n'} \underline{w}^\top \underline{x}_{n'})$, which implies that the misclassified point, *i.e.* $(\underline{x}_{n'}, y_{n'})$, contributes more than the correctly classified point to the gradient.

Q3 **(Problem 4, Midterm 2017)** Consider the logistic regression setup as in the previous question. Suppose we are given a dataset $D = \{(\underline{x}_1, y_1), (\underline{x}_2, y_2)\}$ with

$$\underline{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad y_1 = 1, \quad \underline{x}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad y_2 = -1.$$

For any $\underline{w} \in \mathbb{R}^2$, we consider the ℓ_2-regularized error as

$$E_{\text{in}}(\underline{w}) = -\sum_{n=1}^{N} \log\left[P_{\underline{w}}(y_n \mid \underline{x}_n)\right] + \lambda \|\underline{w}\|_2^2, \quad \lambda > 0,$$

where

$$P_{\underline{w}}(y \mid \underline{x}) = \begin{cases} h(\underline{x}) & y = +1 \\ 1 - h(\underline{x}) & y = -1 \end{cases},$$

and

$$h(x) = \frac{e^{\underline{w}^\top \underline{x}}}{1 + e^{\underline{w}^\top \underline{x}}} = \frac{1}{1 + e^{-\underline{w}^\top \underline{x}}}.$$

3.a For $\lambda = 0$, find the optimal $\underline{w}$ that minimizes $E_{\text{in}}(\underline{w})$ and the minimum value of $E_{\text{in}}(\underline{w})$. (Hint: you are given $(\underline{x}_n, y_n)$, so plug those values into the expression of the in-sample error).

Answer.
Approach 1: Given the dataset,

$$E_{\text{in}}(\underline{w}) = \log(1 + e^{-\underline{w}^\top \underline{x}_1}) + \log(1 + e^{\underline{w}^\top \underline{x}_2}) = \log(1 + e^{-w_0 - w_1}) + \log(1 + e^{w_0}).$$

Note that $E_{\text{in}}(\underline{w})$ is non-negative and it can be minimized by setting $w_0 = -\alpha$ and $w_1 = 2\alpha$ and making $\alpha \to \infty$, as both $\log(1 + e^{w_0})$ and $\log(1 + e^{-w_0 - w_1})$ will converge to zero with such $\underline{w}$. Thus, the minimum value of $E_{\text{in}}(\underline{w})$ is 0.

Approach 2: We can find the optimum point by finding the solution to $\nabla E_{\text{in}}(\underline{w}) = \underline{0}$, as $E_{\text{in}}(\underline{w})$ is a convex function. Given the dataset,

$$\begin{aligned} \nabla E_{\text{in}}(\underline{w}) &= \frac{1}{2}\left(-y_1 \underline{x}_1 \theta(-y_1 \underline{w}^\top \underline{x}_1) - y_2 \underline{x}_2 \theta(-y_2 \underline{w}^\top \underline{x}_2)\right) \\ &= \frac{1}{2}\left(-\frac{y_1 \underline{x}_1}{1 + e^{y_1 \underline{w}^\top \underline{x}_1}} - \frac{y_2 \underline{x}_2}{1 + e^{y_2 \underline{w}^\top \underline{x}_2}}\right) \\ &= \frac{1}{2}\left(-\frac{y_1 \underline{x}_1}{1 + e^{w_0 + w_1}} - \frac{y_2 \underline{x}_2}{1 + e^{-w_0}}\right) \\ &= \begin{bmatrix} \frac{-1}{1+e^{w_0+w_1}} + \frac{1}{1+e^{-w_0}} \\ \frac{-1}{1+e^{w_0+w_1}} \end{bmatrix} \end{aligned}$$

Observe that $\nabla E_{\text{in}}(\underline{w}) = \begin{bmatrix} \frac{-1}{1+e^{w_0+w_1}} + \frac{1}{1+e^{-w_0}} \\ \frac{-1}{1+e^{w_0+w_1}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ can be achieved by setting $w_0 = -\alpha$ and $w_1 = 2\alpha$ and making $\alpha \to \infty$. It is easy to check that with such $\underline{w}$, $\nabla E_{\text{in}}(\underline{w})$ converges to 0.

3.b Suppose λ is a very large constant such that it suffices to consider weights that satisfy $\|\underline{w}\|_2 \ll 1$. Since $\underline{w}$ has a small magnitude, we may use the Taylor series approximation

$$\log(1 + \exp(-y_n \underline{w}^\top \underline{x}_n)) \approx \log(2) - \frac{1}{2} y_n \underline{w}^\top \underline{x}_n.$$

Assuming the above approximation is exact, find $\underline{w}$ that minimizes $E_{\text{in}}(\underline{w})$ (it should be expressed in terms of λ).

Answer. Under such assumptions,

$$E_{\text{in}}(\underline{w}) = \sum_{n=1}^{N} \log(1 + \exp(-y_n \underline{w}^\top \underline{x}_n)) + \lambda \|\underline{w}\|_2^2 = 2\log(2) + \lambda \underline{w}^\top \underline{w} - \frac{1}{2}\sum_{n=1}^{N} y_n \underline{w}^\top \underline{x}_n.$$

Thus, $\nabla E_{\text{in}}(\underline{w}) = 2\lambda \underline{w} - \frac{1}{2}\sum_{n=1}^{N} y_n \underline{x}_n$. Given the dataset, $\nabla E_{\text{in}}(\underline{w}) = 2\lambda \underline{w} - \frac{1}{2}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. To find the optimal point of $E_{\text{in}}(\underline{w})$, it suffice to solve $2\lambda \underline{w}^\star - \frac{1}{2}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \underline{0}$, which results in $\underline{w}^\star = \begin{bmatrix} 0 \\ \frac{1}{4\lambda} \end{bmatrix}$

Q4 (Hinge Loss) Here are two reviews of "Perfect Blue", from *Rotten Tomatoes*:

> **Panos Kotzathanasis** (Asian Movie Plus): "Perfect Blue" is an artistic and technical masterpiece; however, what is of outmost importance is the fact that Satoshi Kon never deteriorate from the high standards he set here, in the first project that was entirely his own.
>
> **Derek Smith** (Cinematic Reflections): [An] nime thriller [that] often plays as an examination of identity and celebrity, but ultimately gets so lost in its own complex structure that it doesn't end up saying much at all.

Rotten Tomatoes has classified these reviews as "positive" and "negative," respectively.

In this assignment, you will create a simple text classification system that can perform this task automatically. We'll warm up with the following set of four mini-reviews, each labeled positive $(+1)$ or negative (-1):

1. $\underline{x}_1$: not good; label: (-1)
2. $\underline{x}_2$: pretty bad; label: (-1)
3. $\underline{x}_3$: good plot; label: $(+1)$
4. $\underline{x}_4$: pretty scenery; label: $(+1)$

Each review $\underline{x}$ is mapped onto a feature vector $\phi(\underline{x})$, which maps each word to the number of occurrences of that word in the review and adds a 1 to account for bias. For example, the second review maps to the (sparse) feature vector $\phi(\underline{x}_2) = \{\textbf{extra added 1} : 1, \textbf{pretty} : 1, \textbf{bad} : 1\}$. The hinge loss for a single datapoint is defined as

$$L_{\text{hinge}}(\underline{x}, y, \underline{w}) = \max\{0, 1 - y\underline{w}^\top \phi(\underline{x})\},$$

where $\underline{x}$ is the review text, y is the correct label, $\underline{w}$ is the weight vector.

4.a (Linearly Inseparable) Given the following dataset of reviews:

1. $\underline{x}_1$: bad; label: (-1)
2. $\underline{x}_2$: good; label: $(+1)$
3. $\underline{x}_3$: not bad; label: $(+1)$
4. $\underline{x}_4$: not good; label: (-1)

Prove that no linear classifier using word features (*i.e.*, word count) can get zero error on this dataset. Remember that this is a question about classifiers, not optimization algorithms; your proof should be true for any linear classifier of the form $f_{\underline{w}}(\underline{x}) = \text{sign}(\underline{w}^\top \phi(\underline{x}))$, regardless of how the weights are learned.

Propose a single additional feature for your dataset that we could augment the feature vector with that to fix this problem.

Answer. Assume by contradiction that there exists a weight vector $\hat{\mathbf{w}}$ that gets zero error on this dataset, *i.e.*,

$$\hat{\mathbf{w}}^\top \phi(\underline{x}_1) < 0, \tag{2}$$
$$\hat{\mathbf{w}}^\top \phi(\underline{x}_2) > 0, \tag{3}$$
$$\hat{\mathbf{w}}^\top \phi(\underline{x}_3) > 0, \tag{4}$$
$$\hat{\mathbf{w}}^\top \phi(\underline{x}_4) < 0. \tag{5}$$

Observe that $\phi(\underline{x}_1) + \phi(\underline{x}_4) = \phi(\underline{x}_2) + \phi(\underline{x}_3)$. Thus, $\hat{\mathbf{w}}^\top \phi(\underline{x}_1) + \hat{\mathbf{w}}^\top \phi(\underline{x}_4) = \hat{\mathbf{w}}^\top \phi(\underline{x}_2) + \hat{\mathbf{w}}^\top \phi(\underline{x}_3)$. However, by (2) and (5), $\hat{\mathbf{w}}^\top \phi(\underline{x}_1) + \hat{\mathbf{w}}^\top \phi(\underline{x}_4) < 0$, and by (3) and (4), $\hat{\mathbf{w}}^\top \phi(\underline{x}_2) + \hat{\mathbf{w}}^\top \phi(\underline{x}_3) > 0$, which is a contradiction.

To make the datapoints linearly separable, it suffices to introduce the new feature which is the multiple of "good" count and "no" count. After augmenting with this feature, the feature vectors will be as follows.

$\phi(\underline{x}_i)$	extra added 1	good	bad	not	not × good	y_i
$\phi(\underline{x}_1)$	1	0	1	0	0	-1
$\phi(\underline{x}_2)$	1	1	0	0	0	1
$\phi(\underline{x}_3)$	1	0	1	1	0	1
$\phi(\underline{x}_4)$	1	1	0	1	1	-1

It is easy to check that the weight vector $\underline{w} = [0, 1, -1, 2, -4]$ has zero loss on this augmented dataset.

Q5 **(Squared Loss)** Suppose that we are now interested in predicting a numeric rating for movie reviews. We will use a non-linear predictor that takes a movie review $\underline{x}$ and returns $\sigma(\underline{w}^\top \phi(\underline{x}))$, where $\sigma(z) = (1 + e^{-z})^{-1}$ is the logistic function that squashes a real number to the range $(0, 1)$. For this problem, assume that the movie rating y is a real-valued variable in the range $[0, 1]$.

5.a Suppose that we wish to use squared loss. Write out the expression of the loss $L(\underline{x}, y, \underline{w})$ for a single datapoint $(\underline{x}, y)$.

Answer.

$$L(\underline{x}, y, \underline{w}) = \frac{1}{2}\left(y - \frac{1}{1 + e^{-\underline{w}^\top \phi(\underline{x})}}\right)^2.$$

5.b Given $L(\underline{x}, y, \underline{w})$ from the previous part, compute the gradient of the loss with respect to $\underline{w}$, $\nabla_{\underline{w}} L(\underline{x}, y, \underline{w})$. Write the answer in terms of the predicted value $p = \sigma(\underline{w}^\top \phi(\underline{x}))$.

Answer.

$$\begin{aligned}
\nabla_{\underline{w}} L(\underline{x}, y, \underline{w}) &= \frac{1}{2}\nabla_{\underline{w}}\left(y - \frac{1}{1 + e^{-\underline{w}^\top \phi(\underline{x})}}\right)^2 = \left(y - \frac{1}{1 + e^{-\underline{w}^\top \phi(\underline{x})}}\right)\nabla_{\underline{w}}\left(y - \frac{1}{1 + e^{-\underline{w}^\top \phi(\underline{x})}}\right) \\
&= -(y - p)\nabla_{\underline{w}}\left(\frac{1}{1 + e^{-\underline{w}^\top \phi(\underline{x})}}\right) = \frac{y - p}{\left(1 + e^{-\underline{w}^\top \phi(\underline{x})}\right)^2}\nabla_{\underline{w}}\left(1 + e^{-\underline{w}^\top \phi(\underline{x})}\right) \\
&= \frac{y - p}{\left(1 + e^{-\underline{w}^\top \phi(\underline{x})}\right)^2}\nabla_{\underline{w}}\left(e^{-\underline{w}^\top \phi(\underline{x})}\right) \\
&= \frac{y - p}{\left(1 + e^{-\underline{w}^\top \phi(\underline{x})}\right)^2} e^{-\underline{w}^\top \phi(\underline{x})}\nabla_{\underline{w}}\left(-\underline{w}^\top \phi(\underline{x})\right) \\
&= -\frac{y - p}{\left(1 + e^{-\underline{w}^\top \phi(\underline{x})}\right)^2} e^{-\underline{w}^\top \phi(\underline{x})}\phi(\underline{x}) \\
&= -\frac{y - p}{1 + e^{-\underline{w}^\top \phi(\underline{x})}} \frac{e^{-\underline{w}^\top \phi(\underline{x})}}{1 + e^{-\underline{w}^\top \phi(\underline{x})}}\phi(\underline{x}) \\
&= (p - y)p(1 - p)\phi(\underline{x})
\end{aligned}$$

5.c Suppose there is one datapoint $(\underline{x}, y)$ with some arbitrary non-zero $\phi(\underline{x})$ and $y = 1$. Specify conditions for $\underline{w}$ to make the magnitude of the gradient of the loss with respect to $\underline{w}$ arbitrarily small (*i.e.*, minimize the magnitude of the gradient). Can the magnitude of the gradient with respect to $\underline{w}$ ever be exactly zero? You are allowed to make the magnitude of $\underline{w}$ arbitrarily large but not infinity.

Why does it matter? the reason why we're interested in the magnitude of the gradients is because it governs how far gradient descent will step. For example, if the gradient is close to zero when $\underline{w}$ is very far from the optimum, then it could take a long time for gradient descent to reach the optimum (if at all). This is known as the vanishing gradient problem when training neural networks.

Answer. Let $y = 1$. Then the gradient would be $\nabla_{\underline{w}} L(\underline{x}, y, \underline{w}) = -p(1 - p)^2\phi(\underline{x})$. To make the gradient arbitrary small, either $p \to 0$ or $p \to 1$, *i.e.*, $e^{-\underline{w}^\top \phi(\underline{x})} \to +\infty$ or $e^{-\underline{w}^\top \phi(\underline{x})} \to 0$. Hence, we must have $\underline{w}^\top \phi(\underline{x}) \to -\infty$ or $\underline{w}^\top \phi(\underline{x}) \to +\infty$. This can be achieved for any arbitrary non-zero $\phi(\underline{x})$, by making the magnitude of the elements in $\underline{w}$ arbitrary large.

The gradient cannot be exactly zero for any arbitrary $\phi(\underline{x})$ since $0 < p < 1$.