

Gradients of logistic regression with squared error and binary cross-entropy loss functions

Pawel Wocjan

February 17, 2019

Abstract

We derive the gradients for logistic regression with (a) mean squared error and (b) binary cross-entropy as loss functions. You will need these results for the first homework.

1 Logistic regression

Let

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \in \mathbb{R}^n \quad \text{and} \quad b \in \mathbb{R} \quad (1)$$

be the weight vector and the bias of a neuron, respectively. Let $a : \mathbb{R} \rightarrow \mathbb{R}$ denote its activation function a . The neuron outputs

$$\hat{y} = a \left(\sum_{j=1}^n w_j x_j + b \right) \quad (2)$$

when given a feature vector $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ as input. We use z to denote

$$z = \sum_{j=1}^n w_j x_j + b. \quad (3)$$

The function implemented by the neuron consists of two steps:

$$\mathbf{x} \mapsto z = \mathbf{w}^T \mathbf{x} + b = \sum_{j=1}^n w_j x_j + b \mapsto a = a(z). \quad (4)$$

For logistic regression the activation function $a(z)$ is equal to the sigmoid function $\sigma(z)$, which is defined by

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \quad (5)$$

The sigmoid function σ maps \mathbb{R} to the open interval $(0, 1)$. It satisfies the following properties:

$$\sigma(0) = \frac{1}{2} \quad (6)$$

$$\sigma(-z) = 1 - \sigma(z) \quad (7)$$

$$\lim_{z \rightarrow -\infty} \sigma(z) = 0 \quad (8)$$

$$\lim_{z \rightarrow \infty} \sigma(z) = 1. \quad (9)$$

Its derivative $\sigma'(z)$ is obtained by applying the chain rule and simple algebraic manipulations:

$$\sigma'(z) = -\frac{1}{(1 + e^{-z})^2} \cdot e^{-z} \cdot (-1) \quad (10)$$

$$= \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}} \quad (11)$$

$$= \frac{1}{1 + e^{-z}} \cdot \frac{1 + e^{-z} - 1}{1 + e^{-z}} \quad (12)$$

$$= \sigma(z) \cdot (1 - \sigma(z)) \quad (13)$$

Logistic regression can be used for binary classification, which is the task of classifying the feature vectors into two classes, denoted by 0 and 1. The activation $\sigma(z)$ can be interpreted as the probability that the neuron assigns to the class 1. Using this probability, we make a prediction as follows:

$$\text{class 0 if } \sigma(z) < \frac{1}{2} \quad (14)$$

$$\text{class 1 if } \sigma(z) \geq \frac{1}{2} \quad (15)$$

2 Squared error and binary entropy loss functions

We consider two loss functions for logistic regression: squared error and binary cross-entropy. Let $\mathbf{x} \in \mathbb{R}^n$ be a feature vector and $y \in \{0, 1\}$ its correct label.

- The squared error loss \mathcal{L}_{se} is defined by

$$\mathcal{L}_{\text{se}} = \frac{1}{2}(a - y)^2. \quad (16)$$

Its derivative with respect to a is equal to

$$\frac{d\mathcal{L}_{\text{se}}}{da} = a - y. \quad (17)$$

- The (binary) cross-entropy loss is defined by

$$\mathcal{L}_{ce} = -y \log a - (1 - y) \log(1 - a). \quad (18)$$

Its derivative with respect with a is equal to

$$\frac{d\mathcal{L}_{ce}}{da} = -\frac{y}{a} + \frac{1 - y}{1 - a}. \quad (19)$$

I will explain the intuition behind the cross-entropy loss in class in detail. For the analysis, consider the following two cases:

- If the true label is $y = 1$, then the loss is equal to $-\log a$, which is a strictly decreasing function on the open interval $(0, 1)$. The loss tends to ∞ as $a \rightarrow 0$ and to 0 as $a \rightarrow 1$, respectively.
- If the true label is $y = 0$, then the loss is equal to $-\log(1 - a)$, which is a strictly increasing function of the open interval $(0, 1)$. The loss tends to ∞ as $a \rightarrow 1$ and to 0 as $a \rightarrow 0$, respectively.

3 Gradient of squared error and binary cross-entropy loss functions

We have compute the partial derivatives of the loss functions with respect to w_j and b to be able to apply stochastic gradient descent. This is done by applying the chain rule multiple times. The derivative of the activation function a is $a' = (1 - a)$ because the sigmoid function is used as the activation for logistic regression.

- The partial derivatives of the squared error loss \mathcal{L}_{se} are derived as follows:

$$\frac{\partial \mathcal{L}_{se}}{\partial w_j} = \frac{d\mathcal{L}_{se}}{da} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_j} = (a - y) \cdot a' \cdot x_j \quad (20)$$

$$\frac{\partial \mathcal{L}_{se}}{\partial b} = \frac{\partial \mathcal{L}_{se}}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial b} = (a - y) \cdot a' \quad (21)$$

Note that $a' \approx 0$ whenever z is either very small or very large. In either case, we say that the neuron is saturated. The problem is that a saturated neuron learns slowly when a is very far from y . This can be improved by using the binary cross entropy loss function.

- The partial derivatives of the cross entropy loss \mathcal{L}_{ce} are derived as follows:

$$\frac{\partial \mathcal{L}_{ce}}{\partial w_j} = \frac{d\mathcal{L}_{ce}}{da} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_j} \quad (22)$$

$$= \left(-\frac{y}{a} + \frac{1-y}{1-a} \right) \cdot a' \cdot x_j \quad (23)$$

$$= \left(-\frac{y}{a} + \frac{1-y}{1-a} \right) \cdot a \cdot (1-a) \cdot x_j \quad (24)$$

$$= \left(-y \cdot (1-a) + (1-y) \cdot a \right) \cdot x_j \quad (25)$$

$$= (a - y) \cdot x_j \quad (26)$$

$$\frac{\partial \mathcal{L}_{be}}{\partial b} = \frac{d\mathcal{L}_{ce}}{da} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial b} = a - y. \quad (27)$$

Here it is essential that we expand a' as $a \cdot (1-a)$ to cancel a and $1-a$ is the numerators.