Deep Learning

Vazgen Mikayelyan

YSU, Krisp

September 19, 2019

Outline

Linear Regression

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}$ be our training data.

Linear Regression

Let $(x_i, y_i)_{i=1}^{n}$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}$ be our training data. Consider the function

$$f(x) = f\left(x^1, x^2, \ldots, x^k\right) = w^1x^1 + w^2x^2 + \ldots + w^kx^k + b = w^T x + b.$$

Linear Regression

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}$ be our training data. Consider the function

$$f(x) = f\left(x^1, x^2, \ldots, x^k\right) = w^1x^1 + w^2x^2 + \ldots + w^kx^k + b = w^Tx + b.$$

Our aim is to find parameters $b, w^1, w^2, \ldots, w^k$ such that

$$f(x_i) \approx y_i, i = 1, \ldots, n.$$

Linear Regression

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}$ be our training data. Consider the function

$$f(x) = f\left(x^1, x^2, \dots, x^k\right) = w^1x^1 + w^2x^2 + \ldots + w^kx^k + b = w^Tx + b.$$

Our aim is to find parameters $b, w^1, w^2, \dots, w^k$ such that

$$f(x_i) \approx y_i, i = 1, \dots, n.$$

We choose L^2 distance as our loss function:

$$\frac{1}{n}\sum_{l=1}^{n}\left(f(x_l) - y_l\right)^2.$$

Questions

1. Should we minimize the loss function using gradient descent?

Questions

1. Should we minimize the loss function using gradient descent?
2. Can you represent this model as a neural network?

Logistic Regression

Let $(x_i, y_i)_{i=1}^{n}$, $x_i \in \mathbb{R}^k$, $y_i \in \{0, 1\}$ be our training data.

Logistic Regression

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \{0, 1\}$ be our training data. Consider the function

$$\begin{aligned} f(x) = f\left(x^1, x^2, \ldots, x^k\right) &= \sigma\left(w^1 x^1 + w^2 x^2 + \ldots + w^k x^k + b\right) \\ &= \sigma(w^T x + b). \end{aligned}$$

Logistic Regression

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \{0, 1\}$ be our training data. Consider the function

$$f(x) = f\left(x^1, x^2, \ldots, x^k\right) = \sigma\left(w^1x^1 + w^2x^2 + \ldots + w^kx^k + b\right)$$
$$= \sigma(w^T x + b).$$

Our aim is to find parameters $b, w^1, w^2, \ldots, w^k$ such that

$$f(x_i) \approx y_i, i = 1, \ldots, n.$$

Let $(x_i, y_i)_{i=1}^{n}$, $x_i \in \mathbb{R}^k$, $y_i \in \{0,1\}$ be our training data. Consider the function

$$f(x) = f\left(x^1, x^2, \ldots, x^k\right) = \sigma\left(w^1 x^1 + w^2 x^2 + \ldots + w^k x^k + b\right)$$

$$= \sigma(w^T x + b).$$

Our aim is to find parameters $b, w^1, w^2, \ldots, w^k$ such that

$$f(x_i) \approx y_i, i = 1, \ldots, n.$$

We choose cross entropy distance as our loss function:

$$\frac{1}{n} \sum_{l=1}^{n} \left(-y_l \log f(x_l) - (1 - y_l) \log (1 - f(x_l))\right).$$

Questions

1. Can you represent this model as a neural network?

Questions

1. Can you represent this model as a neural network?
2. Why do we use the function sigmoid in this case?

Questions

1. Can you represent this model as a neural network?
2. Why do we use the function sigmoid in this case?
3. Why don't we use L^2 distance in this case?

Questions

1. Can you represent this model as a neural network?
2. Why do we use the function sigmoid in this case?
3. Why don't we use L^2 distance in this case?
4. Can we do logistic regression when number of classes is greater than 2?

L1 and L2 Regularizations

L1 and L2 Regularizations

In linear regression instead of L^2 loss we use one from this two:

$$\frac{1}{n}\left(\sum_{l=1}^{n}\left(f\left(x_l\right)-y_l\right)^2+\lambda\sum_{l=1}^{k}\left|w_l\right|\right),$$

$$\frac{1}{n}\left(\sum_{l=1}^{n}\left(f\left(x_l\right)-y_l\right)^2+\lambda\sum_{l=1}^{n}w_l^2\right).$$

Outline

Softmax Classifier

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}^m$ (one-hot vectors) be our training data.

Softmax Classifier

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}^m$ (one-hot vectors) be our training data. Consider the function

$$f(x) = \left(\frac{e^{w_1^T x + b_1}}{\sum\limits_{i=1}^{m} e^{w_i^T x + b_i}}, \ldots, \frac{e^{w_m^T x + b_m}}{\sum\limits_{i=1}^{m} e^{w_i^T x + b_i}} \right)$$

Softmax Classifier

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}^m$ (one-hot vectors) be our training data. Consider the function

$$f(x) = \left(\frac{e^{w_1^T x + b_1}}{\sum_{i=1}^{m} e^{w_i^T x + b_i}}, \ldots, \frac{e^{w_m^T x + b_m}}{\sum_{i=1}^{m} e^{w_i^T x + b_i}} \right)$$

Our aim is to find parameters $(b_i, w_i)_{i=1}^m$ such that

$$f(x_i) \approx y_i, i = 1, \ldots, n.$$

Softmax Classifier

Let $(x_i, y_i)_{i=1}^n$, $x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}^m$ (one-hot vectors) be our training data. Consider the function

$$f(x) = \left(\frac{e^{w_1^T x + b_1}}{\sum_{i=1}^{m} e^{w_i^T x + b_i}}, \dots, \frac{e^{w_m^T x + b_m}}{\sum_{i=1}^{m} e^{w_i^T x + b_i}} \right)$$

Our aim is to find parameters $(b_i, w_i)_{i=1}^m$ such that

$$f(x_i) \approx y_i, i = 1, \dots, n.$$

We choose cross entropy distance as our loss function:

$$\frac{1}{n} \sum_{l=1}^{n} \left(-y_i^T \log f(x_i) \right).$$

Questions

1. Can you represent this model as a neural network?

Questions

1. Can you represent this model as a neural network?
2. Can we use the function sigmoid in this case?

Questions

1. Can you represent this model as a neural network?
2. Can we use the function sigmoid in this case?
3. What to do in the case of multi-label classification?