Logistic

Zhiliang Tian

2016.05.01

1 logistic

1.1 introduction

Logistic Function , (also called sigmoid function), is a non-linear function. We usually use it to transform a scores range($-\infty$, $+\infty$) to another score range(0, 1). Binary classification task usually need logistic to normalized.

Taking logistic regression(called LR) as an example, (Note that logistic regression is not a regression model, but classification model). we feed model score in logistic function, treat the output of logistic as probability. Why using Logistic? Logistic has some exclusive features as follow:

- monotonicity
- output range is from 0 to 1 (the necessary condition of acting as a probability)
- non-linear transform
- easy to get derivative (even if working with loss function)

1.2 formula

$$P = \frac{1}{1 + \exp(-S)} \tag{1}$$

In LR, S_k is logistic input, we usually treat S as the model score, it also can be called non-normalized probability. P and $1 - P$ the probability of the two classes.

1.3 logistic regression model

In logistic regression(Equation 2), we feed the result of $W_i * x_i$ in logistic. x_i is the i-th dimension of input feature vector $\mathbf{x}$, W is weight vector, W_i is the connection between i-th dimension of input feature and the output.

$$P = \frac{1}{1 + \exp(-\sum_i^{dim} W_i * x_i)} \tag{2}$$

1.4 the calculation of gradient

Our target: $\frac{\alpha(P)}{\alpha(S)}$.

$$\frac{\alpha(P)}{\alpha(S)} = \frac{\exp(-S)}{(1 + \exp(-S))^2} = P(1 - P) \tag{3}$$

1.5 more about logistic

1. Activation funciton

   Usually acting as a activation function(non-linear function) in neural network(called NN). In NN, we use activation function to find a active values from a series of input value, just like a neural cell in biology.)

2. Different with softmax

   - Output is a value, so usually is used in binary classification. If being used in multi-class classification, we need K logistic node to do binary classification K times(K is class number).
   -

2 logistic with cross entropy loss

2.1 introduction

Using in a real task, What effect will logistic take? How can we estimate the unknown parameter in real model. We take LR as an example. To solving this problem, we must estimate the parameter W in LR. So, we use maximum likelihood estimation.

2.2 formula

Maximum likelihood estimation:

$L = \prod_m^{dataset} P_m = \prod_m^{dataset} \prod_{k=0}^{K} ((P_{mk})^{y(mk)}) = \prod_m^{dataset} ((P_m)^{y(m)} * (1 - P_m)^{(1-y(m))})$

$$y(m) = \begin{cases} 1 \\ 0 \end{cases} \tag{4}$$

$y(m)$ is m-th sample's label.

Using log maximum likelihood for convenient.

$$\log(L) = \sum_m^{dataset} (y(m) \log(P_m) + (1 - y(m)) * \log(1 - P_m)) \tag{5}$$

We only introduce sgd-based algorithm. Using sgd, we need the loss on one sample.

$$J_m = \log(L_m) = y(m) \log(P_m) + (1 - y(m)) * \log(1 - P_m)) \tag{6}$$

From Equation 6, it can be concluded that "using log maximum likelihood to estimate unknown parameter on logistic" is equal to "using cross entropy loss to optimize model on logsitic".

2.3 the calculation of gradient logistic with cross entropy loss

2.3.1 the gradient of corss entropy loss

We can get it easily:

$$\frac{\alpha(J)}{\alpha(P)} = \frac{y - P}{P(1 - P)} \tag{7}$$

2.3.2 the gradient of logistic with cross entropy loss

According to Equation 3

$$\frac{\alpha(J)}{\alpha(P)} = y - P \tag{8}$$