Softmax

Zhiliang Tian

2016.05.01

1 softmax and maximum entropy

1.1 introduction

Softmax Function is a non-linear function, normalization function. We usually use it to transform a series of non-normalized model scores to a probability distribution. Classify task usually need softmax to normalized. Why using Softmax? Softmax has some exclusive features as follow:

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

1.2 formula

$$P_k = \frac{\exp(S_k)}{\sum_{k=0}^{K} \exp(S_k)} \tag{1}$$

S_k is softmax input, we usually treat S_k as the model score of k-th class, it also can be called non-normalized probability. P_k is the probability of k-th class.

1.3 maximum entropy model

In maximum entropy model(Equation 2), we feed the result of $W_{ik} * x_i$ in softmax. x_i is the i-th dimension of input feature vector $\mathbf{x}$, W is weight matrix, W_{ik} is the connection between i-th dimension of input feature and k-th class.

$$P_k = \frac{\exp(\sum_{i=0}^{dim} W_{ik} * x_i)}{\sum_{k=0}^{K} \exp(\sum_{i=0}^{dim} W_{ik} * x_i))} \tag{2}$$

1.4 the calculation of gradient

Our target: $\frac{\alpha(P)}{\alpha(S_j)}$. we have some ideas:

1. From the standard softmax formula, we can see P_i is not independent with S_j (even if $i \neq j$). So the calculation of gradient is complex, we need discuss this solution in different cases, ($i \neq j$ and $i = j$).

$$\frac{\alpha(P_i)}{\alpha(S_j)} = \frac{\alpha(\frac{\exp(S_i)}{\sum_{k=0}^{K}\exp(S_k)})}{\alpha(S_j)} \tag{3}$$

2. $\sum_{k=0}^{K}\exp(S_k)$ is also complex. For convenience, we divide constant variable and volatile variable. When we take partial derivative respect to $\exp(S_j)$, $\sum_{k=0,k!=j}^{K}\exp(S_k)$ is a constant variable.

$$\sum_{k=0}^{K}\exp(S_k) = C_j + \exp(S_j) \tag{4}$$

C_j is a constant variable when S_j is independent variable.

$$\frac{\alpha(P_i)}{\alpha(S_j)} = \frac{\alpha(\frac{\exp(S_i)}{C_j+\exp(S_j)})}{\alpha(S_j)} \tag{5}$$

Then, we can easily calculate the derivative.

$$\frac{\alpha(P_i)}{\alpha(S_j)} = \begin{cases} \frac{C_j*\exp(S_j)}{(C_j+\exp(S_j))^2} = P_j*(1-P_j) & i=j \\ -\frac{\exp(S_i)*\exp(S_j)}{(C_j+\exp(S_j))^2} = -P_i*P_j & i \neq j \end{cases} \tag{6}$$

Note that if i=j this derivative is similar to the derivative of the logistic function.

We can also write like this:

$$\begin{aligned} \tfrac{\alpha(P_i)}{\alpha(S_j)} &= P_i*(\delta_{ij}-P_j) \\ \delta_{ij} &= \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \end{aligned} \tag{7}$$

δ_{ij} can be see as the expectation on observation from dataset.

2 softmax with cross entropy loss

2.1 introduction

Using in a real task, what effect will softmax take? How can we estimate the unknown parameter in real model. As Softmax usually being used in Classify task, we take an example of classify task—-maximum entropy model(called maxent). To solving this problem, we must estimate the parameter W in maxent. So, we use maximum likelihood estimation.

2.2 formula

Maximum likelihood estimation:

$$\begin{aligned} & L = \prod_m^{dataset} P_m = \prod_m^{dataset} \prod_{k=0}^{K}((P_{mk})^{y(mk)}) \\ & y(mk) = \begin{cases} 1 & k = label \\ 0 & k \neq label \end{cases} \end{aligned} \tag{8}$$

$y(mk)$ is m-th sample's label.
Using log maximum likelihood for convenient.

$$\log(L) = \sum_m^{dataset} \sum_{k=0}^{K} (y(mk) \log(P_{mk})) \tag{9}$$

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

$$J_m = \log(L_m) = y(k) \log(P_k)) \tag{10}$$

From Equation 10, it can be concluded that ”using log maximum likelihood to estimate unknown parameter on softmax” is equal to ”using cross entropy loss to optimize model on softmax”. Notice that the cross entropy on logistic(sigmoid) is: $L = y(k)\log(P_k)) + (1 - y(k))\log(1 - P_k))$ it may seems different with that on softmax. But indeed they are all same: $\sum_{k=0}^{K}(y(k)\log(P_k))$. In logistic model on 2-class classify, we use one node to predict the probability: 1-th class is P, and 2-th class is $1 - P$. but softmax is k-nodes and P_k.

2.3 the calculation of gradient softmax with cross entropy loss

2.3.1 the gradient of corss entropy loss

We can get it easily:

$$\frac{\alpha(J)}{\alpha(P_k)} = \frac{y(k)}{P_k} \tag{11}$$

2.3.2 the gradient of softmax with cross entropy loss

According to Equation 6

$$\begin{aligned}
&\tfrac{\alpha(J)}{\alpha(S_k)} = \textstyle\sum_i^K \tfrac{\alpha(L)}{\alpha(P_i)} * \tfrac{\alpha(P_i)}{\alpha(S_k)} = \tfrac{y(k)}{P_k} * (P_k * (1 - P_k)) + \textstyle\sum_{i \neq k} \tfrac{y(i)}{P_i} * (-P_i * P_k) \\
&= y(k) * (1 - P_k) + \textstyle\sum_{i \neq k} (-y(i) * P_k) = y(k) - P_k * \textstyle\sum_i y(i) \\
&= y(k) - P_k
\end{aligned} \tag{12}$$

Note that we already derived $\frac{\alpha(P_i)}{\alpha(S_k)}$ for $i = j$ and $i \neq j$ above. Notice that if minimizing the loss, the gradient should be $P_k - y(k)$

2.3.3 more about this gradient

1. Understand the Gradient

   For a sample, If hitting the ground-truth, gradient of score S_k is $y(k) - P_k$, ranged [0, 1]. The gradient lead the score to be higher, the less P_k, the more higher. If not hitting the ground-truth, gradient of score S_k is $-P_k$, ranged [-1, 0]. The gradient lead the score to be lower, the more P_k, the more lower.

2. with Logistic

   From equation, we can see softmax with cross entropy loss is similar to logistic with cross entropy loss.

3. Expectation of Observation and Estimation

   Some scholar find a interesting thing. If calculating the negative Maximum likelihood estimation(just the opposite number of MLE), final gradient(Equation 12) will be $P_k - y(k)$. and we should minimize it.

   $$\frac{\alpha(-J_m)}{\alpha(P_{mk})} = P_{mk} - y(mk) = E(P_{est_m}) - E(P_{obs_m}) = E_{est} - E_{obs} \tag{13}$$

P_{est}, the probability of model estimation. P_{obs}, the probability of real observation data. E_{est}, the expectation of model estimation. E_{obs}, the expectation of observation data.

It said that, we should by minimize the difference between E_{est} and E_{obs} during training, the ideal state is E_{est} equal to E_{obs}.This interesting thing will also happen to logistic similarly.