Softmax With Loss(for Multi Label Learning)

Zhiliang Tian

2016.07.02

1 Introduction

1.1 Thought

Based on original softmax function(single-label), multi-label softmax regression loss fit for multi-labeling learning. Softmax with cross entropy loss minimize the KL-divergence between prediction and ground-truth probabilities.

2 Feed Forward

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

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.

3 Loss

3.1 Cross entropy loss

$$Loss = \sum_{i}^{dataset} Loss_i = \sum_{i}^{dataset} \sum_{k}^{K} P_{ik}^{g} * \log(P_{ik}^{p}) \tag{2}$$

For i-th sample in dataset, K is the category size, k is one class in K. P_{ik}^{g} stands for the ground-truth of i-th sample k-th class, P_{ik}^{p} stands for the probability of i-th sample k-th class.

3.2 Suit for Multi-label Task

The difference between single-label and multi-label is the ground-truth, in this loss function, P^g_{ik} is different.

Single-label:

$$P^g_{ik} = \begin{cases} 1 & \mathbf{Y}_{ik} = 1 \\ 0 & \mathbf{Y}_{ik} = 0 \end{cases} \tag{3}$$

$\mathbf{Y}_{ik}$ is the label of i-th sample's k-th class. We can obtain the P^g_{ik} of multi-label by normalizing label $\mathbf{Y}_i$ as $\mathbf{Y}/||\mathbf{Y}||_1$

Multi-label:

$$P^g_{ik} = \begin{cases} 1/|\mathbf{Y}_i| & \mathbf{Y}_{ik} = 1 \\ 0 & \mathbf{Y}_{ik} = 0 \end{cases} \tag{4}$$

For i-th sample, $|\mathbf{Y}_i|$ is the number of positive label.

3.3 Cross entropy loss by normalization(for multi-label)

$$Loss = \sum_{i}^{dataset} Loss_i \tag{5}$$

$$Loss_i = \sum_{i}^{dataset} \sum_{k}^{K} \frac{\mathbf{Y}_{ik}}{|\mathbf{Y}_i|} * \log(P^p_{ik}) = \sum_{i}^{dataset} \sum_{k}^{K} \theta_{(ik)} \frac{1}{|\mathbf{Y}_i|} * \log(P^p_{ik}) \tag{6}$$

$$\theta_{(ik)} = \begin{cases} 1 & \mathbf{Y}_{ik} = 1 \\ 0 & \mathbf{Y}_{ik} = 0 \end{cases} \tag{7}$$

$$P^p_k = \frac{\exp(S_k)}{\sum_{k=0}^{K} \exp(S_k)} \tag{8}$$

4 Back Propegatation

We calculate the gradient three steps: get $\frac{\alpha(P)}{\alpha(S_k)}$;get $\frac{\alpha(Loss)}{\alpha(P_j)}$, get $\frac{\alpha(Loss)}{\alpha(S_k)}$.

4.1 the Gradient of $\frac{\alpha(P)}{\alpha(S_k)}$

We have some ideas:

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

$$\frac{\alpha(P_i)}{\alpha(S_k)} = \frac{\alpha(\frac{\exp(S_i)}{\sum_{j=0}^{K} \exp(S_j)})}{\alpha(S_k)} \tag{9}$$

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

$$\sum_{j=0}^{K} \exp(S_j) = C_k + \exp(S_k) \tag{10}$$

C_k is a constant variable when S_k is independent variable.

$$\frac{\alpha(P_i)}{\alpha(S_k)} = \frac{\alpha(\frac{\exp(S_i)}{C_k+\exp(S_k)})}{\alpha(S_k)} \tag{11}$$

Then, we can easily calculate the derivative.

$$\frac{\alpha(P_i)}{\alpha(S_k)} = \begin{cases} \frac{C_k * \exp(S_k)}{(C_k+\exp(S_k))^2} = P_k * (1 - P_k) & i = k \\ -\frac{\exp(S_i) * \exp(S_k)}{(C_k+\exp(S_k))^2} = -P_i * P_k & i \neq k \end{cases} \tag{12}$$

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

We can also write like this:

$$\begin{aligned} \frac{\alpha(P_i)}{\alpha(S_k)} &= P_i * (\delta_{ik} - P_k) \\ \delta_{ik} &= \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} \end{aligned} \tag{13}$$

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

4.2 the Gradient of $\frac{\alpha(Loss)}{\alpha(P_j)}$

According to equation 6, (for each sample)

$$\frac{\alpha(Loss)}{\alpha(P_j)} = \frac{\mathbf{Y}_j}{|\mathbf{Y}|} * \frac{1}{P_j^p} \tag{14}$$

$$\frac{\alpha(Loss)}{\alpha(P_j)} = \begin{cases} 0 & \mathbf{Y}_j = 0 \\ \frac{\log(P_j^p)}{|\mathbf{Y}|} & \mathbf{Y}_j = 1 \end{cases} \tag{15}$$

4.3 the Gradient of $\frac{\alpha(Loss)}{\alpha(S_k)}$

$$\frac{\alpha(Loss)}{\alpha(S_k)} = \sum_i^K \frac{\alpha(Loss)}{\alpha(P_i)} * \frac{\alpha(P_i)}{\alpha(S_k)} = \sum_{i \neq k} \frac{\mathbf{Y}_i}{|\mathbf{Y}| * P_i} * (-P_i * P_k) + \frac{\mathbf{Y}_k}{|\mathbf{Y}| * P_k} * (P_k * (1 - P_k))$$
$$= \sum_{i \neq k}(-\frac{\mathbf{Y}_i}{|\mathbf{Y}|} * P_k) + \frac{\mathbf{Y}_k}{|\mathbf{Y}|} * (1 - P_k) = \frac{\mathbf{Y}_k}{|\mathbf{Y}|} - P_k * \sum_i^K \frac{\mathbf{Y}_i}{|\mathbf{Y}|}$$
$$= \frac{\mathbf{Y}_k}{|\mathbf{Y}|} - P_k \tag{16}$$

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 $P_k - \frac{\mathbf{Y}_k}{|\mathbf{Y}|}$

4.4 More about This Gradient

1. Understand the Gradient

   Similar to Softmax with loss note, for a sample:

   - If hitting the ground-truth, gradient of score S_k is $\frac{\mathbf{Y}_k}{|\mathbf{Y}|} - P_k$. if $\frac{\mathbf{Y}_k}{|\mathbf{Y}|} > P_k$, gradient ranged $[0, \frac{\mathbf{Y}_k}{|\mathbf{Y}|}]$,the gradient lead the score to be higher, the less P_k, the more higher.
   - If hitting the ground-truth, but $\frac{\mathbf{Y}_k}{|\mathbf{Y}|} < P_k$, the gradient ranged $[1 - \frac{\mathbf{Y}_k}{|\mathbf{Y}|}, 0]$, lead the score to be lower, the less P_k, the less lower. This is the biggest difference from single-label softmax, multi-label remand that every label should be predict as positive during do not knowing which is important, according to maximum entropy thought, we set the every label's target: probability are all both equally and maximized, so P should be $\frac{\mathbf{Y}_k}{|\mathbf{Y}|}$.
   - 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.

5 Reference

Deep Convolutional Ranking for Multilabel Image Annotation 2014 Arxiv
Tagprop: Discriminative metric learn- ing in nearest neighbor models for image auto-annotation. ICCV, 2009