AWD-LSTM-Notes

潘宜城

2019 年 10 月 17 日

目录

1 SplitCrossEntropy **2**

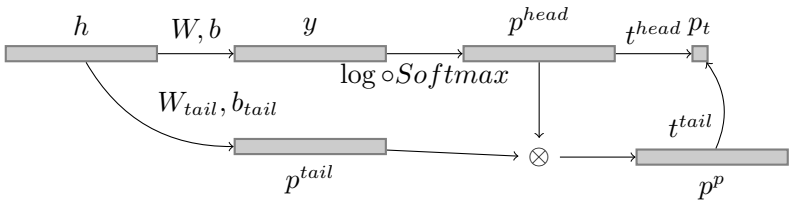
1.1 相关符号 2

1.2 Forward process 2

1.3 Log probability 4

1 SplitCrossEntropy

该Loss对不同区间的单词赋予不同的权重，然后再进行联合概率的计算。简单地看，对一个输出词向量，损失的计算如图所示。



1.1 相关符号

n_h隐藏层大小，也是循环网络输出的维度。

n_v词库的大小，即词的总数量，也是下面的划分的右边界。

$S = \{s_0, s_1, ..., s_{m-1}, s_m\}$用于将词分为不同区间的分割点，以词的字典索引为准。划分区间个数为m. 为了包含整个词库，其中s_0为0,s_m为词库大小。

$W_{tail} \in R^{(m-1)\times n_h}, b_{tail} \in R^{m-1}$ is the weight and bias that used to calculate the probability of the tombstones.

1.2 Forward process

This process gives total cross entropy loss according to the hidden output H, real targets t and the decoder weights $W \in R^{n_v \times n_h}, b \in R^{n_v}$.

First reshape H to a size of $R^{n \times n_h}$, and then split the input tensor according to targets t position in the splits with the split parameter $S = \{s_0, s_1, ..., s_{m-1}, s_m\}$. The results are denoted as $\tilde{H} = \{\tilde{H}_1, ..., \tilde{H}_m\}, \tilde{t} = \{\tilde{t}^1, ..., \tilde{t}^m\}$. For clarity, $\tilde{t}^i$ includes all the target word indices that lie in $[s_{i-1}, s_i)$. And $\tilde{H}$ is split the same way. The size of every split is denoted as n_i

Second, perform soft max on the head split $[s_0, s_1$) and the tombstones.

Construct the weight for it as

$$W_{head} = \left(\begin{array}{c} W[s_0,:] \\ W[s_0+1,:] \\ \vdots \\ W[s_1-1,:] \\ \hline W_{tail} \end{array}\right), b_{head} = \left(b_{s_0} b_{s_0+1} \cdots b_{s_1-1} \middle| b_{tail}\right)$$

And concatenate $\tilde{H}$ in the row dimension, we get

$$\tilde{H}' = \begin{pmatrix} \tilde{H}_1 \\ \tilde{H}_2 \\ \vdots \\ \tilde{H}_m \end{pmatrix}$$

Then apply a linear function of the above parameters, the result is

$$Y^{head} = \tilde{H}' \times W_{head}^T + b_{head}$$

After that we apply a log soft max along the column of Y to get the logarithm of the possibility of head words and tombstones.

$$P^{head} = \log\left(Softmax(Y^{head})\right)$$

Finally we can calculate the cross entropy of the words.

For the first split segment, also noted as head words, the entropy is just the original form, as

$$E_{head} = -\sum_{i \in [0,\, n_1)} P^{head}_{i,\tilde{t}_i^1}$$

For the words in the remaining segments, the entropy will include the possibility of tombstones, assume we are calculating the entropy for segment $p (p \in [2, m])$. The range of the index of the words in this segment is $[s_{p-1}, s_p)$. We use the decoder weight within this range $W[s_{p-1} : s_p, :], b[s_{p-1} : s_p]$ to get the soft max vector.

$$\begin{aligned} P^{p\prime} &= \log\left(Prob(Tomb_p) \times Prob(Words|Tomb_p)\right) \\ &= \left(\log\left(Prob(Tomb_p)\right) + \log\left(Prob(Words|Tomb_p)\right)\right) \\ &= P^{head}[s_{p-1} : s_p, -(m+1-p)] + P^p \end{aligned}$$

For calculating some variables above, we use Y^p to denote the linear output of hidden state in segment p. So we can get the log soft max value as follows.

$$Y^p = \tilde{H}_p \times W[s_{p-1} : s_p, :]^T + b[s_{p-1} : s_p]$$
$$P^p = \log\Big(Softmax(Y^p)\Big)$$

And the final entropy of segment p is

$$E^p = -\sum_{i \in [0, n_p)} P^{p\prime}_{i, \tilde{t}^p_i}$$

So the total entropy of the generation H is $\frac{1}{n} \times \big(E^{head} + \sum_p E^p\big)$.

1.3 Log probability

This process can calculate the modified soft max probability of any given hidden state. It is the same like the above forward process except that no targets are taken to sum over samples. So for an input of size $(k \times n_h)$, the output will be of size $(k \times n_v)$, every row is a log of the probability distribution of the words.