Loss Criteria for ASR

Cal Peyser

February 6, 2019

This document is about loss criteria for end-to-end neural models for automatic speech recognition. Simple loss criteria are hard to adapt to ASR, since an ASR model emits a distribution for every frame of input. Furthermore, the model tends to be evaluated not simply on the likelihood of the result transcript, but rather on some non-differentiable error computation, such as word error rate.

1 Connectionist Temporal Classification

In a CTC model, an RNN emits a distribution for each input frame over the set $A' = A \cup \{blank\}$, where A is the label alphabet. We denote the activation y_k^t as the probability of emitting the label k at time t. For a sequence π of output labels, we have

$$P(\pi|x) = \prod_{t=0}^{T} y_{\pi_t}^t$$

Here, we make the significant assumption that the y_t are conditionally independent given x. This is a simplifying assumption, since one can imagine the emission of a certain word affecting subsequent words.

Define F as a function that removes blank symbols from a sequence, and let $l \in A^{\leq T}$ be a sequence of non-blank labels. Then we can compute the probability of a sequence output:

$$P(l|x) = \sum_{\pi \in F^{-1}(l)} P(\pi|x)$$

1.1 Forward-Backward Algorithm

Computing $P(l|x)$ efficiently requires a dynamic programming algorithm. The algorithm aims to capture the probability that at a certain time t, the sequence of output emissions has correctly "gotten up to" the spot s in the ground truth. For example, if the ground truth is

$$A\ B\ C$$

And we've so far seen the emissions

$$b\ b\ A\ A\ A\ B\ b\ b$$

We can say that the emissions have correctly "gotten up to" the second label in the ground truth.

To this end, we define the sequence l' which is the same as l, but with blanks at the beginning and end and between every label. This represents the possible paths forward to correctly emitting the sequence l - we can start with an arbitrary number of blanks, followed by an arbitrary number of l_1, followed by an arbitrary number of blanks, ect.

We define $\alpha_t(s)$ as the probability that at time t we have correctly "gotten up to" label s in l'. More specifically:

$$\alpha_t(s) = \sum_{\pi|\pi_{1:t}=l'_{1:s}} \prod_{t'=1}^{t} y^{t'}_{\pi_{t'}}$$

We define $\beta_t(s)$ similarly, only with suffixes. So, $\beta_t(s)$ is the probability that the output suffix beginning at position t is correct and "gets up to" label s in l'. More specifically:

$$\beta_t(s) = \sum_{\pi|\pi_{t:T}=l'_{s:|l'|}} \prod_{t'=t}^{T} y^{t'}_{\pi_{t'}}$$

The trick for computing these properties is that $\alpha_t(s)$, for example, depends only on a handful of other probabilities. Suppose, for instance, that l'_s is blank. Then in order for $\pi_{1:t} = l'_{1:s}$, we would need either that $\pi_{t-1} = l'_{s-1}$ or that $\pi_{t-1} = blank$. In both these cases, the next label should be π_t. So, if l'_s is blank:

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1))y^t_{l'_s}$$

If l'_s is not blank, then we have three possibilities: either the previous label was l'_{s-1}, the previous label was blank, or the previous label was l'_s and this will be a duplicate. So, if l'_s is not blank:

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-2}(s-1)y^t_{l'_s}$$

Similar relationships can be defined for $\beta_t(s)$. We can use these relationships to compute α and β values recursively.

1.2 Computing CTC Loss and Gradient

We use α and β in order to determine the CTC loss. We observe that $\alpha_t(s)\beta_t(s)$ is equal to the probability that the outputs define a path that correctly reduces to l, and which also passes through label l'^s and time t, with an extra $y^t_{l'_s}$ term since that term is contained in both $\alpha_t(s)$ and $\beta_t(s)$. So we have:

$$P(l|x) = \sum_{s=1}^{|l'|} \frac{1}{y^t_{l'_s}} \alpha_t(s)\beta_t(s)$$

We define CTC loss as:

$$L_l = -\ln P(l|x)$$

In order to differentiate with respect to y^t_k we choose those values of s for which $l'_s = k$, and call it $lab(l,k)$. Then,

$$\frac{\partial P(l|x)}{\partial y^t_k} = \frac{1}{{y^t_k}^2} \sum_{s \in lab(l,k)} \alpha_t(s)\beta_t(s)$$