Lecture 21: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, part 2

Mark Hasegawa-Johnson

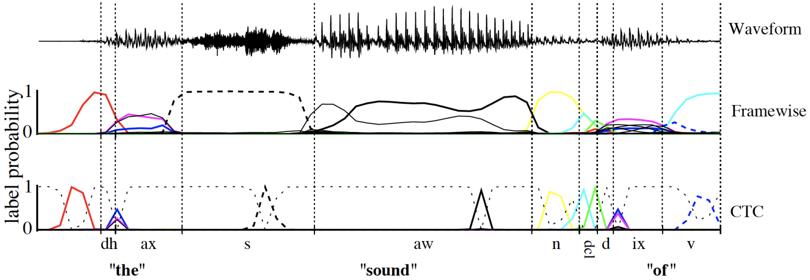
ECE 537, Fall 2022

1. Review: CTC in Testing Mode
2. Maximum Likelihood Training of a CTC Network
3. Summary

Outline

1. Review: CTC in Testing Mode
2. Maximum Likelihood Training of a CTC Network
3. Summary

Temporal Classification Example: Speech



Temporal classification maps from a sequence of speech frames (top) to a sequence of phoneme or character labels (bottom).

Graves et al., 2006, Figure 1. (c) ICML

Variables in CTC

- $\mathbf{x} = [x_1, \ldots, x_T]$ is the input. It is a sequence of vectors, $x_t = [x_1^t, \ldots, x_m^t]$.
- $\mathbf{y} = [y_1, \ldots, y_T]$ is the network output. It is a sequence of probability vectors, $y_u = [y_1^u, \ldots, y_{|L|+1}^u]$.
- $\pi = [\pi_1, \ldots, \pi_T]$ is the path. It is a sequence of characters,

$$y_k^t = p(\pi_t = k|\mathbf{x})$$

- $\mathbf{l} = [l_1, \ldots, l_U] = \mathcal{B}(\pi)$ is the label sequence, $U \leq T$, which should be compared to the correct label sequence, $\mathbf{z}$.

CTC Forward Algorithm: The Modified Label Sequence

In order to express the CTC forward algorithm, we need to define a modified label sequence, $\mathbf{l}'$. $\mathbf{l}'$ is equal to $\mathbf{l}$ with blanks inserted between every pair of letters. Thus if

$$\mathbf{l} = [\mathrm{f}, \mathrm{e}, \mathrm{d}],$$

then

$$\mathbf{l}' = [-, \mathrm{f}, -, \mathrm{e}, -, \mathrm{d}, -].$$

If the length of $\mathbf{l}$ is $|\mathbf{l}|$, then the length of $\mathbf{l}'$ is $2|\mathbf{l}| + 1$.

CTC Forward Algorithm: Partial Sequences

We also need to define the following partial sequences:

$$
\begin{aligned}
\mathbf{x}_{1:t} &= [x_1, \ldots, x_t] \\
\pi_{1:t} &= [\pi_1, \ldots, \pi_t] \\
\mathbf{l}'_{1:s} &= [l'_1, \ldots, l'_s] \\
&= \begin{cases} [-, l_1, -, l_2, \ldots, l_{s/2}] & s \text{ even} \\ [-, l_1, -, l_2, \ldots, l_{(s-1)/2}, -] & s \text{ odd} \end{cases}
\end{aligned}
$$

The CTC Forward Algorithm

Definition:

$$\alpha_t(\mathbf{l}'_{1:s}) \equiv p(\mathbf{l}'_{1:s}|\mathbf{x}_{1:t})$$

The CTC Forward Algorithm

1. **Initialize:**

$$\alpha_t([-]) = y_-^1$$
$$\alpha_t([-, l_1]) = y_{l_1}^1$$

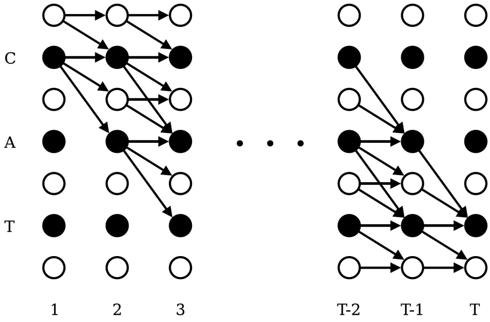
2. **Iterate:**

$$\alpha_t(\mathbf{l}'_{1:s}) = \begin{cases} (\alpha_{t-1}(\mathbf{l}'_{1:s}) + \alpha_{t-1}(\mathbf{l}'_{1:s-1})) \times y_{l'_s}^t \\ \ldots\ldots\ldots\ldots\text{if } l'_s = - \text{ or } l'_s = l'_{s-2} \\ (\alpha_{t-1}(\mathbf{l}'_{1:s}) + \alpha_{t-1}(\mathbf{l}'_{1:s-1}) + \alpha_{t-1}(\mathbf{l}'_{1:s-2})) \times y_{l'_s}^t \\ \ldots\ldots\ldots\ldots\text{otherwise} \end{cases}$$

3. **Terminate:**

$$p(\mathbf{l}_{1:U}|\mathbf{x}) = \alpha_T(\mathbf{l}'_{1:2U}) + \alpha_T(\mathbf{l}'_{1:2U+1})$$

The CTC Forward Algorithm



Graves et al., 2006, Fig. 3. (c) ICML

Outline

1. Review: CTC in Testing Mode
2. Maximum Likelihood Training of a CTC Network
3. Summary

The CTC Loss

The CTC loss function is the negative log probability of the correct label sequence given the waveform:

$$\mathcal{L}_{\text{CTC}} = -\ln p(\mathbf{z}|\mathbf{x})$$

This is similar to cross entropy, but differentiating it is more complicated, since the correct label sequence is $\mathbf{z} = [z_1, \ldots, z_U]$, while the speech sequence is $\mathbf{x} = [x_1, \ldots, x_T]$

Probability of Labels Given Speech

- We want to train the network to maximize the probability of the correct labeling, $p(\mathbf{z}|\mathbf{x})$.
- The most computationally efficient way to calculate $p(\mathbf{z}|\mathbf{x})$ is the forward algorithm:

$$p(\mathbf{z}|\mathbf{x}) = \alpha_T(\mathbf{z}'_{1:2U}) + \alpha_T(\mathbf{z}'_{1:2U+1}),$$

  …but that form is not easy to differentiate.
- The expansion over all possible paths is not a computationally efficient way to calculate $p(\mathbf{z}|\mathbf{x})$, but it's easier to differentiate:

$$p(\mathbf{z}|\mathbf{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{z})} p(\pi|\mathbf{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{z})} \prod_{t=1}^{T} y_{\pi_t}^t$$

Differentiating the CTC Loss

Remember that the basic principle of back-propagation is the chain rule. If we want $\frac{d\mathcal{L}}{dw}$, we can find it as

$$\frac{d\mathcal{L}}{dw} = \sum_{\tau=1}^{T} \sum_{k=1}^{|L|+1} \left(\frac{d\mathcal{L}}{dy_k^\tau}\right) \left(\frac{\partial y_k^\tau}{\partial w}\right),$$

- $\frac{\partial y_k^\tau}{\partial w}$ is the same as for any other RNN, so it's uninteresting.
- $\frac{d\mathcal{L}}{dy_k^\tau}$ is unique to CTC. Let's derive it.

Differentiating the CTC Loss

$$\mathcal{L}_{\text{CTC}} = -\ln p(\mathbf{z}|\mathbf{x}) = -\ln\left(\sum_{\pi\in\mathcal{B}^{-1}(\mathbf{z})}\prod_{t=1}^{T} y_{\pi_t}^t\right)$$

Therefore

$$\frac{d\mathcal{L}}{dy_k^\tau} = \left(\frac{-1}{p(\mathbf{z}|\mathbf{x})}\right)\left(\frac{dp(\mathbf{z}|\mathbf{x})}{dy_k^\tau}\right) = \left(\frac{-1}{p(\mathbf{z}|\mathbf{x})}\right)\left(\frac{1}{y_k^\tau}\sum_{\pi\in\mathcal{B}^{-1}(\mathbf{z}),\pi_\tau=k}\prod_{t=1}^{T} y_{\pi_t}^t\right)$$

- The sum in the last line is over all paths that are valid expansions of the correct transcription, and for which $\pi_\tau = k$.
- The $\frac{1}{y_k^\tau}$ comes from the derivative of the product:

$$\frac{d}{dy}xyz = xz = \frac{1}{y}xyz$$

Differentiating the CTC Loss

$$\frac{d\mathcal{L}}{dy_k^\tau} = \left(\frac{-1}{p(\mathbf{z}|\mathbf{x})}\right)\left(\frac{1}{y_k^\tau}\sum_{\pi\in\mathcal{B}^{-1}(\mathbf{z}),\pi_\tau=k}\prod_{t=1}^{T} y_{\pi_t}^t\right)$$

The sum in the last line is over all paths that are valid expansions of the correct transcription, and for which $\pi_\tau = k$. This has a nice Bayesian interpretation:

$$\begin{aligned}\frac{d\mathcal{L}}{dy_k^\tau} &= \left(\frac{-1}{p(\mathbf{z}|\mathbf{x})}\right)\left(\frac{1}{y_k^\tau}p(\mathbf{z},\pi_\tau = k|\mathbf{x})\right)\\ &= \frac{-1}{y_k^\tau}p(\pi_\tau = k|\mathbf{z},\mathbf{x})\end{aligned}$$

The CTC Gamma Probability

Just as for any other HMM, let's define a gamma probability, $\gamma_\tau(k) = p(\pi_\tau = k, \mathbf{z}|\mathbf{x})$. Then

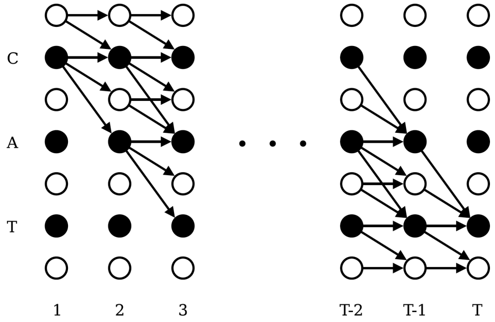
$$\frac{d\mathcal{L}}{dy_k^\tau} = -\frac{\gamma_\tau(k)}{y_k^\tau},$$

where

$$\gamma_\tau(k) = p(\pi_\tau = k, \mathbf{z}|\mathbf{x}) = \frac{1}{y_k^\tau} \sum_{s:z'_s=k} \alpha_\tau(\mathbf{z}'_{1:s})\beta_\tau(\mathbf{z}'_{s:(2U+1)}) \qquad (1)$$

- $\beta_t(\mathbf{z}'_{s:2U+1}) = p(\mathbf{z}'_{s:(2U+1)}|\mathbf{x}_{t:T})$
- Notice that $\alpha_\tau(\mathbf{z}'_{1:s})$ and $\beta_\tau(\mathbf{z}'_{s:(2U+1)})$ both include the fact that the network is producing $z'_s = k$ at time τ. To compensate for that duplication, Eq. (1) has a $\frac{1}{y_k^\tau}$ factor.

The CTC Forward-Backward Algorithm

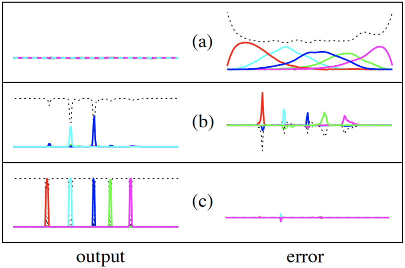


Graves et al., 2006, Fig. 3. (c) ICML

$\alpha_t(\mathbf{z}'_{1:s})$ is the probability of the best path up to (and including) state z'_s at time t. $\beta_t(\mathbf{z}'_{s:(2U+1)})$ is the probability of the best path starting from state z'_s at time t.

- At the start of training (a), $y_k^t = 0$ for all k except the blank symbol. $\gamma_t(k)$, is determined mostly by the known sequence of the correct labels, $\mathbf{z}$.
- After some training (b), y_k^t has started to converge. Its convergence guides the forward-backward algorithm, so $\gamma_t(k)$ is also much more localized.
- When fully converged, $y_k^t \approx \gamma_t(k)$, so the error is nearly zero.

Left column: network outputs y_k^t. Right column: error signal $\gamma_t(k) - y_k^t$.



Outline

1. Review: CTC in Testing Mode
2. Maximum Likelihood Training of a CTC Network
3. Summary

Conclusions

- A CTC network is trained to minimize

$$\mathcal{L}_{\text{CTC}} = -\ln p(\mathbf{z}|\mathbf{x})$$

- Differentiating, we discover that

$$\frac{d\mathcal{L}}{dy_k^\tau} = \frac{1}{y_k^\tau} p(\pi_\tau = k|\mathbf{z}, \mathbf{x})$$

- $p(\pi_\tau = k|\mathbf{z}, \mathbf{x})$ can be computed using the forward-backward algorithm.
- Even in the very first epoch of training, the known sequence $\mathbf{z}$ distributes error uniformly across the waveform. For this reason, CTC training converges smoothly and quickly (compared, e.g., to Transformer loss).