Why does CTC result in peaky behavior?

Albert Zeyer [1 2] **Ralf Schlüter** [1 2] **Hermann Ney** [1 2]

Abstract

The peaky behavior of CTC models is well known experimentally. However, an understanding about *why* peaky behavior occurs is missing, and whether this is a good property. We provide a formal analysis of the peaky behavior and gradient descent convergence properties of the CTC loss and related training criteria. Our analysis provides a deep understanding why peaky behavior occurs and when it is suboptimal. On a simple example which should be trivial to learn for any model, we prove that a feed-forward neural network trained with CTC from uniform initialization converges towards peaky behavior with a 100% error rate. Our analysis further explains why CTC only works well together with the blank label. We further demonstrate that peaky behavior does not occur on other related losses including a label prior model, and that this improves convergence.

1. Introduction

The peaky behavior of connectionist temporal classification (CTC) (Graves et al., 2006) (Figure 1) was often observed experimentally. However, it is completely unknown *why* models trained with CTC get peaky. Also, other training criteria for the same models and label sets will not result in the same peaky behavior, so this is a result of the CTC training criterion.

We will formally define our understanding of peaky behavior. Then we provide a formal analysis as to in what cases and *why* we will get such behavior. We will see that the peaky behavior results as a corollary from the training criterion and its local convergence properties, where gradient descent from a uniform initialization tends towards suboptimal local optima with peaky behavior.

We demonstrate that in the case of a training criterion with peaky behavior like CTC, it is crucial to use a label topology with a blank label, and having a silence label in case of speech recognition is suboptimal. This is an important new understanding of the blank label, which has been observed experimentally before (Bluche et al., 2015).

Peaky behavior can be problematic in certain cases, e.g. when an application requires to not use the blank label, e.g. to get meaningful time accurate alignments of phonemes to a transcription. Also, we will mathematically demonstrate that local-context models like feed-forward neural networks (FFNNs) are suboptimal for such kind of training criterion like CTC, which is due to the peaky behavior. We show variations of the training criterion by including a label prior, and we demonstrate that this solves convergence problems and does not lead to peaky behavior anymore.

Some of the mathematical proofs and demonstrations were assisted using the computer algebra system SymPy (Meurer et al., 2017) via symbolic computation. We also performed synthetic experiments using TensorFlow (TensorFlow Development Team, 2015) and RETURNN (Zeyer et al., 2018). We publish all the symbolic computation code, and all the code and configs of our experiments.[1]

2. Related Work

While this work focuses on CTC, we have a similar complete marginalization over all possible alignments in the training criterion of recurrent neural network transducer (RNN-T) (Graves, 2012a), recurrent neural aligner (RNA) (Sak et al., 2017), lattice-free maximum mutual information (MMI) (Povey et al., 2016) and AutoSegCriterion (ASG) (Collobert et al., 2016).

Training with the full-sum over all alignment paths with neural networks is not novel (Bengio et al., 1991; Haffner, 1993; Senior & Robinson, 1996; Hennebert et al., 1997; Yan et al., 1997; LeCun et al., 1998; Li & Wu, 2014). From-scratch (flat-start, Gaussian mixture model (GMM)-free) training with frequent realignments was also discussed in (Zhang & Woodland, 2014; Senior et al., 2014; Bacchiani et al., 2014).

A label prior model has been used together with CTC-trained models at decoding time (Naoyuki Kanda, 2016; Miao et al., 2015).

To the best of our knowledge, no prior work exists which analyzes the reasons of peaky behavior. It is completely

[1]Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, Aachen, Germany [2]AppTek GmbH, Aachen, Germany. Correspondence to: Albert Zeyer <zeyer@cs.rwth-aachen.de>.

[1]https://github.com/rwth-i6/returnn-experiments/tree/master/2021-formal-peaky-behavior-ctc

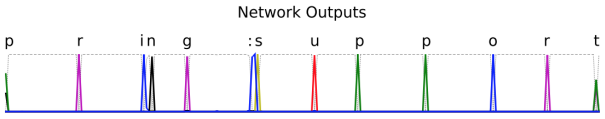


Figure 1. CTC peaky output, Figure 7.9 from (Graves, 2012b). The colored lines depict the output activations for different labels over time. The greyed dotted line represents the blank label.

unknown why such training criteria results in this unnatural behavior, and in fact other training criteria do not.

3. Definition of Peaky Behavior

The peaky behavior of CTC is best illustrated in Figure 1. It shows the neural network (NN) output probability distribution over a subset of the labels, including the blank label. It can be seen that along the time axis, the blank label is dominating most of the time, while the other labels are observed only as a spike event. I.e. the probability distribution over the time is peaky.

Preliminaries 3.1. Let S be a finite set of labels. Let $\mathcal{M}$ be a model such that it defines the probability distribution $p_t(s|x_1^T, \mathcal{M})$ for some input signal x_1^T, over time frames $t \in \{1, \dots, T\}$, $s \in S$. For notational simplicity, we partially leave out the condition on $\mathcal{M}$. The probability of the label sequence y_1^N is defined assuming label-independence

$$p(y_1^N|x_1^T) = \sum_{s_1^T : y_1^N} p(s_1^T|x_1^T) = \sum_{s_1^T : y_1^N} \prod_t p_t(s_t|x_1^T).$$

We call $s_1^T \in S^T$ an alignment. The elements of y are not relevant here – what matters are all the allowed alignments s_1^T given y_1^N. We denote the set of possible alignments as $\mathcal{A}_T(y_1^N) := \left\{s_1^T \,\middle|\, s_1^T : y_1^N\right\}$. $\mathcal{A}$ is also called the *label topology* and is usually defined by a finite state transducer (FST). In our case, we always use a FST which is equivalent to a regular expression (RE) of the form $y_1^{1|+|*} \dots y_n^{1|+|*}$, $y \in S$. For simplicity, we assume this defines a unique mapping $(T, s_1^T) \mapsto (N, y_1^N)$. Let $\mathcal{D} = \left\{(x_1^T, y_1^N)\right\}$ be the training dataset. In the following, we focus the analysis only on a single training sample (x_1^T, y_1^N). This is not really a restriction, as you can concatenate multiple sequences into one. Also, in most cases, the analysis would trivially generalize to multiple sequences but would complicate the notation.

The *label topology* $\mathcal{A}$ will be of central importance for the convergence behavior. CTC defines such a topology by allowing blank anywhere, and by allowing label repetitions. We explicitly define some possible topologies for some simple examples.

Example 3.2. Consider the target sequence consisting only of a single label, $y_1^N = (\mathtt{a})$, and define our label set as $S = \{\mathtt{B}, \mathtt{a}\}$, and

$$\mathcal{A}_T(y_1^N) := \left\{s_1^T \,\middle|\, s_1^T \text{ matches reg. expr. } \mathtt{B}^*\mathtt{a}^+\mathtt{B}^*\right\}.$$

Despite an empty target sequence, Example 3.2 is arguably one of the simplest possible non-trivial CTC examples. The label $\mathtt{B}$ plays the role of the blank label.

Definition 3.3 (Peaky alignment)**.** Let $s_1^T \in \mathcal{A}_T(y_1^N)$. We say that the *alignment* s_1^T is *peaky with dominant label* $\mathring{s}$, if

$$|\{t \mid s_t = \mathring{s}\}| > |\{t \mid \tilde{s}_t = \mathring{s}\}| \quad \forall\, \tilde{s}_1^T \in \mathcal{A}_T(y_1^N).$$

With Example 3.2, and $T = 100$, $s_1^T = (\mathtt{B}^{49}, \mathtt{a}, \mathtt{B}^{50})$ is a peaky alignment with dominant label $\mathring{s} = \mathtt{B}$. Label $\mathtt{a}$ occurs only in one single frame, i.e. is peaky. I.e. a peaky alignment is peaky w.r.t. all the non-dominant labels.

Definition 3.4 (Viterbi)**.** Given a model $\mathcal{M}$ and a sample (x_1^T, y_1^N), a *Viterbi* alignment is an alignment $s_1^T \in \mathcal{A}_T(y_1^N)$ such that it maximizes $\prod_t p_t(s_t|x_1^T, \mathcal{M})$.

Definition 3.5 (Peaky behavior)**.** Given a sample (x_1^T, y_1^N) and a model $\mathcal{M}$. If all Viterbi alignments $s_1^T \in \mathcal{A}_T(y_1^N)$ are *peaky*, then the *model* $\mathcal{M}$ has *peaky behavior for* (x_1^T, y_1^N). If that holds true for all $(x_1^T, y_1^N) \in \mathcal{D}$, then we simply say that the model has *peaky behavior*.

Definition 3.6 (Alignment count)**.** Let

$$\mathcal{C}(s, t, T) := \left|\left\{s_1^T \,\middle|\, s_t = s, s_1^T \in \mathcal{A}_T(y_1^N)\right\}\right|$$

be the *count* of all alignments with label s in frame t. Let

$$\mathcal{C}(T) := \left|\mathcal{A}_T(y_1^N)\right|$$

be the total *count* of all alignments.

Remark 3.7. Note that the alignment count is totally independent from the input features x. It only depends on the possible alignments in the label topology $\mathcal{A}_T(y_1^N)$.

Definition 3.8 (Label count)**.** Let

$$\mathcal{C}(s, T) := \sum_t \mathcal{C}(s, t, T)$$

be the total *count* of label $s \in S$ in all frames.

Definition 3.9 (Dominant label)**.** The label $\mathring{s} \in S$ is *dominant in* $\mathcal{A}_T(y_1^N)$, if

$$\mathcal{C}(\mathring{s}) > \mathcal{C}(s) \quad \forall\, s \in S, s \neq \mathring{s}.$$

Remark 3.10. Note that the dominance property of $\mathring{s}$ is defined depending on T, y_1^N and the label topology $\mathcal{A}$, *independent from the input* x. For the CTC topology, blank always has this property. For the common hidden Markov model (HMM) topology in speech recognition, silence almost always gets this property, simply by the same counting arguments.

Let us recall the Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$). This example is simple enough that we can exactly calculate these counts.

Lemma 3.11. *For Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$),*

$$\mathcal{C}(T) = \frac{T \cdot (T+1)}{2}$$
$$\mathcal{C}(s{=}\mathtt{a}, t, T) = t \cdot (T - t + 1)$$
$$\mathcal{C}(s{=}\mathtt{B}, t, T) = \frac{T^2}{2} - T \cdot t + \frac{T}{2} + t^2 - t$$

$$\mathcal{C}(s{=}\mathtt{a}, T) = \frac{T \cdot (T^2 + 3T + 2)}{6}$$
$$\mathcal{C}(s{=}\mathtt{B}, T) = \frac{T \cdot (T^2 - 1)}{3}$$

Thus the dominant label is $\mathring{s} = \mathtt{B}$ *for* $T \geq 5$.

Corollary 3.12. *Following from Example 3.2 (*$\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$*) and Lemma 3.11 we get*

$$\frac{\mathcal{C}(\mathring{s}, T)}{\sum_s \mathcal{C}(s, T)} = \frac{2 \cdot (T-1)}{3T}$$

which is the average count of the dominant label $\mathring{s} = \mathtt{B}$ *per frame. I.e. for* $T \geq 5$ *we have* $\frac{\mathcal{C}(\mathring{s},T)}{\sum_s \mathcal{C}(s,T)} > 50\%$.

Corollary 3.13. *We can count the number of frames where some label* s *dominates, i.e. define*

$$C_T^{\textit{Frames}}(s) := |\{t \mid \mathcal{C}(s,t,T) > \mathcal{C}(s',t,T)\ \forall s' \neq s\}| \leq T.$$

Following further from Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$*) and Lemma 3.11 we get*

$$C_T^{\textit{Frames}}(\mathring{s}) = 2\lceil \tfrac{1}{2}T - \tfrac{1}{2}\sqrt{T+1} - \tfrac{1}{2} \rceil \geq T - \sqrt{T+1} - 1.$$

I.e. $\lim_{T\to\infty} \frac{C_T^{\textit{Frames}}(\mathring{s})}{T} = 1$*, i.e. in the limit,* $\mathring{s} = \mathtt{B}$ *will strongly dominate per frame. For* $T \geq 8$*, we have* $\frac{C_T^{\textit{Frames}}(\mathring{s})}{T} \geq 50\%$*. For* $T \geq 24$*, we have* $\frac{C_T^{\textit{Frames}}(\mathring{s})}{T} \geq 75\%$.

Recall again that these corollaries are just about the dataset $\mathcal{D}$, or more precisely just the target label sequences y_1^N and the input sequence length T (but not the input itself, x_1^T), and the topology $\mathcal{A}$. They are independent from any training criterion or any model. However, based on these, we will show that models trained with the CTC criterion with gradient descent show peaky behavior, i.e. all their Viterbi alignments are peaky.

4. Convergence to Peaky Behavior

Now we study the convergence behavior of the training criterion (loss) when trained with gradient descent. I.e. we have some model initialization and we locally modify the model parameters such that the loss decreases.

Definition 4.1. The CTC loss is defined as

$$L := -\log \sum_{s_1^T : y_1^N} p(s_1^T | x_1^T) = -\log \sum_{s_1^T : y_1^N} \prod_t p_t(s_t | x_1^T).$$

Remark 4.2. We always have $L \geq 0$. Let s_1^T be any valid alignment (peaky or not), and assume a model $\hat{\mathcal{M}}(s_1^T)$ with

$$p_t(s | x_1^T, \hat{\mathcal{M}}(s_1^T)) := \begin{cases} 1 & \text{if } s = s_t, \\ 0 & \text{else.} \end{cases}.$$

Then we have reached a global optimum with $L = 0$. If any p_t is not sharp like this, we have $L > 0$.

Remark 4.3. Let θ be the model parameters of $\mathcal{M}$. The gradient of L with respect to the model parameters θ is given (compare (Graves, 2012b ↗; Zeyer et al., 2017 ↗)) as

$$\frac{\partial}{\partial \theta} L = -\sum_{s,t} q_t(s | x_1^T, y_1^N, \theta) \cdot \frac{\partial}{\partial \theta} \log p_t(s_t | x_1^T, \theta) \quad (1)$$

with

$$q_t(s | x_1^T, y_1^N, \theta) = \frac{\sum_{s_1^T : y_1^N, s_t = s} p(s_1^T | x_1^T, \theta)}{\sum_{s_1^T : y_1^N} p(s_1^T | x_1^T, \theta)}.$$

The quantity q can be efficiently computed using the forward-backward (Baum-Welch) algorithm and is also known as soft-alignment.

Remark 4.4. If p_t is a uniform distribution for all t, it cancels out in L and also in q_t. We simply get

$$q_t(s | x_1^T, y_1^N, \theta) = \frac{\sum_{s_1^T : y_1^N, s_t = s} 1}{\sum_{s_1^T : y_1^N} 1} = \frac{\mathcal{C}(s,t,T)}{\mathcal{C}(T)}.$$

As a first model to understand the convergence behavior, we analyze a model which is totally independent from the input x_1^T and just consists of a bias term. We would expect that this model learns a prior over the labels as they occur in the training targets. The model is also relevant, as every neural network usually has a bias term in the output softmax, and this bias term will get exactly the same gradient.

Definition 4.5 (Bias model)**.** The model $\mathcal{M}^b$ just consists of a single bias parameter, and is totally independent from the input x_1^T, i.e. for $\theta = b \in \mathbb{R}^S$,

$$p_t(s | x_1^T, \mathcal{M}^b) := \mathrm{softmax}(b).$$

Theorem 4.6. *Let* $\mathring{s}$ *be dominant in* $\mathcal{A}_T(y_1^N)$*. Starting with the model* $\mathcal{M}^{b_0}$ *initialized with uniform distribution, for example* $b_0 = 0$*, then gradient descent on* L *will converge to a model with peaky behavior.*

Proof. First observe that

$$\frac{\partial}{\partial b_i} L = \sum_t p_t - q_t = T \cdot (\mathrm{softmax}(b_i) - \mathbb{E}_t[q_t])$$

for a gradient step i. By Remark 4.4 for $i = 0$ we get

$$\mathbb{E}_t[q_t][s] = \frac{\mathcal{C}(s,T)}{\sum_{s' \in S} \mathcal{C}(s',T)}.$$

I.e. $\mathrm{argmin}_s \frac{\partial}{\partial \theta_0} L = \mathring{s}$. One gradient step will result in $b_1[\mathring{s}] > b_1[s] \quad \forall s \neq \mathring{s}$. For $|S| = 2$, it is clear that we cannot escape from that region of b anymore where we always have peaky behavior. For the case $|S| > 2$, for some $s \neq \mathring{s}$, when comparing the relative difference of $q[\mathring{s}]$ vs. $q[s]$ in the forward-backward computation through the FST, we can disregard any paths not contributing to $\{\mathring{s}, s\}$, as they will be shared. Thus we can reduce the case to $|S| = 2$, and it follows that $b[\mathring{s}] > b[s]$. □

Simulation 4.7. Consider Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$), $T = 5$, with dominant label $\mathring{s} = \mathtt{B}$ (via Lemma 3.11). We can simulate that the bias model uniformly initialized converges to the probability distribution $p_t(s|x_1^T) \approx \{\mathtt{B} \mapsto 0.72, \mathtt{a} \mapsto 0.28\}$ (i.e. peaky behavior), which does not reflect the target label prior distribution $\frac{\mathcal{C}(s,T)}{\sum_{s'} \mathcal{C}(s',T)} \approx \{\mathtt{B} \mapsto 0.53, \mathtt{a} \mapsto 0.47\}$. I.e. *peaky behavior reinforces itself.*

Now we consider a very simple model with dependence on the input x. This can be interpret as a FFNN with a single softmax layer and no bias.

Definition 4.8 (FFNN)**.** Define the model $\mathcal{M}^W$ as

$$p_t(s|x_1^T) := \mathrm{softmax}(x_t W)[s],$$

where $W \in \mathbb{R}^{D_x, S}$ and $x_t \in \mathbb{R}^{D_x}$.

Example 4.9. For Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$), define

$$x_1^T := \begin{pmatrix} \underbrace{0 \cdots 0}_{n \text{ times}} & \underbrace{1 \cdots 1}_{2n \text{ times}} & \underbrace{0 \cdots 0}_{n \text{ times}} \\ \underbrace{1 \cdots 1}_{n \text{ times}} & \underbrace{0 \cdots 0}_{2n \text{ times}} & \underbrace{1 \cdots 1}_{n \text{ times}} \end{pmatrix},$$

for some $n \in \mathbb{N}$, $T = 4n$, i.e. $x_t \in \mathbb{R}^2$ and either $x_t = x_\mathtt{B} := \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$ or $x_t = x_\mathtt{a} := \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$. These constructed x_t can be interpreted as corresponding to the label $\mathtt{B}$ or label $\mathtt{a}$.

Remark 4.10. Note that Example 4.9 is constructed in such a way that the probability distribution $p(x)$ over possible inputs x is uniform. This is optimistic because in practice, e.g. in audio, silence frames often dominate. Such input domination contributes further to peaky behavior. However, we will show that we get peaky behavior even for this constructed case where there is no dominating input feature.

Remark 4.11. With Example 4.9 and the FFNN model, we can see that we reach the optimum $L = 0$ with $W = \left(\begin{smallmatrix} \infty & 0 \\ 0 & \infty \end{smallmatrix}\right)$ and get as close as we want with a matrix over $\mathbb{R}$. Any such solution has 0% error rate. This trivially generalizes to similarly constructed more complex examples.

Theorem 4.12. *Consider the FFNN model uniformly initialized, e.g. $W = 0$, and Example 4.9 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$) with $n \geq 4$, i.e. $T \geq 16$, i.e. the dominant label is $\mathring{s} = \mathtt{B}$. When trained with L with gradient descent, the model converges to peaky behavior, which is a suboptimal local optima, and yields 100% error rate.*

Proof. We can reparameterize the FFNN by the very generic model-free setting:

$$p_t(s|x_t{=}x_\mathtt{a}) := \mathrm{softmax}((\theta_\mathtt{a}, -\theta_\mathtt{a}))[s],$$
$$p_t(s|x_t{=}x_\mathtt{B}) := \mathrm{softmax}((-\theta_\mathtt{B}, \theta_\mathtt{B}))[s].$$

This can parameterize any possible discriminative distribution (if we allow ∞ as well), and specifically exactly the same probability distributions as the FFNN. In this parameterization, we have 2 scalar parameters $\theta_\mathtt{a}, \theta_\mathtt{B}$, i.e. our parameter space is in $\mathbb{R}^2$. We get the initial uniform distribution with $\theta_\mathtt{a} = \theta_\mathtt{B} = 0$. We visualize the loss function over the parameters in Figure 2a. There are two global optima for this loss under the parameters, which are $\theta_\mathtt{a} = \theta_\mathtt{B} = \infty$ and $\theta_\mathtt{a} = -\theta_\mathtt{B} = \infty$. In the first case, the discriminative model would output the label $\mathtt{a}$ at input $x_\mathtt{a}$, and label $\mathtt{B}$ at input $x_\mathtt{B}$. In the second case, the model would always output label $\mathtt{a}$. From the figure, we can see that there is a local optima in the region $\theta_\mathtt{B} > 0, \theta_\mathtt{a} < 0$, and this is the local optima which we reach when we start in $\theta_\mathtt{a} = \theta_\mathtt{B} = 0$. For all parameters in this region $\theta_\mathtt{B} > 0, \theta_\mathtt{a} < 0$, all peaky alignments have higher scores than all other alignments, i.e. the model always has peaky behavior. Decoding with this FFNN with peaky behavior always yields $\mathrm{argmax}_s\, p_t(s \mid x_1^T) = \mathring{s} = \mathtt{B}$, i.e. the model has 100% error rate.

We can explicitly calculate the gradients by Remark 4.4 and SymPy. For the ("blank") frames t with $x_\mathtt{B} = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$, we get

$$\mathbb{E}_{t, x_t = x_\mathtt{B}}[q_t(\mathring{s}|x_1^T, W_0)] = \frac{19n^2 - 1}{6n(4n+1)} > 74\%$$

and for the ("label") frames t with $x_a = \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$, we get

$$\mathbb{E}_{t, x_t = x_\mathtt{a}}[q_t(\mathring{s}|x_1^T, W_0)] = \frac{13n^2 - 1}{6n(4n+1)} > 50\%.$$

We see from the figure that we can never escape that local minima, because the gradients on the lines $\theta_\mathtt{a} = 0, \theta_\mathtt{B} > 0$ and $\theta_\mathtt{a} < 0, \theta_\mathtt{B} = 0$ points towards the same region $\theta_\mathtt{B} > 0, \theta_\mathtt{a} < 0$, which means that gradient descent can not escape from this region.

Case 1, $\theta_\mathtt{a} < 0, \theta_\mathtt{B} = 0$: Define

$$C_\mathtt{a}(s_1^T) := |\{t \mid s_t = \mathring{s}, x_t = x_\mathtt{a}\}| \in \{0, \dots, 2n\},$$
$$p_\mathtt{a} := \mathrm{softmax}((\theta_\mathtt{a}, -\theta_\mathtt{a}))[\mathtt{B}] > 0.5,$$
$$p'_\mathtt{a} := \frac{p_\mathtt{a}}{1 - p_\mathtt{a}} > 1.$$

Then

$$\mathbb{E}_{t, x_t = x_\mathtt{B}}[q_t(s|x_1^T, W_0)]$$
$$= \frac{1}{p(y_1^N|x_1^T)} \frac{1}{2n} \sum_{\substack{t, \\ x_t = x_\mathtt{B}}} \sum_{c=0}^{2n} \sum_{\substack{s_1^T : y_1^N, \\ s_t = \mathring{s}, \\ C_\mathtt{a}(s_1^T) = c}} 0.5^{2n} {p_\mathtt{a}}^c (1 - p_\mathtt{a})^{2n-c}$$
$$= \frac{1}{p(y_1^N|x_1^T)} \frac{1}{2n} (0.5(1 - p_\mathtt{a}))^{2n} \sum_{\substack{t, \\ x_t = x_\mathtt{B}}} \sum_{c=0}^{2n} \sum_{\substack{s_1^T : y_1^N, \\ s_t = \mathring{s}, \\ C_\mathtt{a}(s_1^T) = c}} {p'_\mathtt{a}}^c$$
$$= \frac{1}{p(y_1^N|x_1^T)} \frac{1}{2n} (0.5(1 - p_\mathtt{a}))^{2n} \sum_{c=0}^{2n} {p'_\mathtt{a}}^c \sum_{\substack{t, \\ x_t = x_\mathtt{B}}} \sum_{\substack{s_1^T : y_1^N, \\ s_t = \mathring{s}, \\ C_\mathtt{a}(s_1^T) = c}} 1.$$

Now we are back at counting. Define

$$\mathcal{C}_\mathtt{a}(s, c) := \sum_{\substack{t, \\ x_t = x_\mathtt{B}}} \sum_{\substack{s_1^T : y_1^N, \\ s_t = s, \\ C_\mathtt{a}(s_1^T) = c}} 1,$$
$$\Delta\mathcal{C}_\mathtt{a}(c) := \mathcal{C}_\mathtt{a}(s{=}\mathtt{B}, c) - \mathcal{C}_\mathtt{a}(s{=}\mathtt{a}, c).$$

If $\sum_{c=0}^{2n} \Delta\mathcal{C}_\mathtt{a}(c) {p'_\mathtt{a}}^c > 0$, we have shown that $\mathbb{E}_{t, x_t = x_\mathtt{B}}[q_t(s{=}\mathtt{B}|x_1^T, W_0)] > \mathbb{E}_{t, x_t = x_\mathtt{B}}[q_t(s{=}\mathtt{a}|x_1^T, W_0)]$. Via SymPy, we can calculate that

$$\Delta\mathcal{C}_\mathtt{a}(c) = \begin{cases} 0, & c = 0, \\ 4n(n^2 - 1)\frac{1}{3}, & c = 2n, \\ 2n(c + n), & 0 < c < 2n. \end{cases}$$

Given that we have $n \geq 4$, we get $\Delta\mathcal{C}_\mathtt{a}(c) > 0$ for all $c > 0$, and thus $\sum_{c=0}^{2n} \Delta\mathcal{C}_\mathtt{a}(c) {p'_\mathtt{a}}^c > 0$. I.e. gradient descent will increase $\theta_\mathtt{B}$, i.e. increase $p(s{=}\mathtt{B}|x{=}x_\mathtt{B})$.

Case 2, $\theta_\mathtt{a} = 0, \theta_\mathtt{B} > 0$: Analogous to the other case, we define

$$C_\mathtt{B}(s_1^T) := |\{t \mid s_t = \mathtt{B}, x_t = x_\mathtt{B}\}| \in \{0, \dots, 2n\},$$

$$p_{\mathtt{B}} := \mathrm{softmax}((-\theta_{\mathtt{B}}, \theta_{\mathtt{B}}))[\mathtt{B}] > 0.5,$$
$$p'_{\mathtt{B}} := \frac{p_{\mathtt{B}}}{1 - p_{\mathtt{B}}} > 1.$$

Then we get

$$\mathbb{E}_{t, x_t = x_{\mathtt{a}}}[q_t(s|x_1^T, W_0)]$$
$$= \frac{1}{p(y_1^N|x_1^T)} \frac{1}{2n} \left(0.5(1 - p_{\mathtt{B}})\right)^{2n} \sum_{c=0}^{2n} {p'_{\mathtt{B}}}^c \sum_{\substack{t, \\ x_t = x_{\mathtt{a}}}} \sum_{\substack{s_1^T : y_1^N, \\ s_t = s, \\ C_{\mathtt{B}}(s_1^T) = c}} 1.$$

Now we are back at counting. Define

$$\mathcal{C}_{\mathtt{B}}(s, c) := \sum_{\substack{t, \\ x_t = x_{\mathtt{a}}}} \sum_{\substack{s_1^T : y_1^N, \\ s_t = s, \\ C_{\mathtt{B}}(s_1^T) = c}} 1,$$
$$\Delta\mathcal{C}_{\mathtt{B}}(c) := \mathcal{C}_{\mathtt{B}}(s{=}\mathtt{B}, c) - \mathcal{C}_{\mathtt{B}}(s{=}\mathtt{a}, c).$$

Via SymPy, we can calculate that

$$\Delta\mathcal{C}_{\mathtt{B}}(c) = \begin{cases} 2n(2n^2 - 3n - 2)\frac{1}{3}, & c = 2n, \\ 4n(n-1), & c = 2n-1, \\ 2n(3c - 4n + 1), & n \le c < 2n-1, \\ -2n^2, & c = n-1, \\ -2n(c+1), & 0 \le c < n-1, \end{cases}$$

$$\sum_{c=0}^{2n} \Delta\mathcal{C}_{\mathtt{B}}(c) = 4n(n^2 - 3n - 1)\frac{1}{3} > 0. \tag{2}$$

Define $c^* := \frac{4n-1}{3}$. We see that

$$\Delta\mathcal{C}_{\mathtt{B}}(c) = 0, \quad c = c^*,$$
$$\Delta\mathcal{C}_{\mathtt{B}}(c) > 0, \quad \forall c > c^*,$$
$$\Delta\mathcal{C}_{\mathtt{B}}(c) < 0, \quad \forall c < c^*.$$

Now choose any $\tilde{c} \in \mathbb{N}$ with

$$\Delta\mathcal{C}_{\mathtt{B}}(c) \ge 0, \quad \forall c \ge \tilde{c},$$
$$\Delta\mathcal{C}_{\mathtt{B}}(c) \le 0, \quad \forall c \le \tilde{c} - 1.$$

Via Equation (2), we know that

$$\sum_{c=\tilde{c}}^{2n} \Delta\mathcal{C}_{\mathtt{B}}(c) > -\sum_{c=0}^{\tilde{c}-1} \Delta\mathcal{C}_{\mathtt{B}}(c).$$

And we get

$$-\sum_{c=0}^{\tilde{c}-1} \Delta\mathcal{C}_{\mathtt{B}}(c) {p'_{\mathtt{B}}}^c \le \left(-\sum_{c=0}^{\tilde{c}-1} \Delta\mathcal{C}_{\mathtt{B}}(c)\right) {p'_{\mathtt{B}}}^{\tilde{c}}$$
$$< \left(\sum_{c=\tilde{c}}^{2n} \Delta\mathcal{C}_{\mathtt{B}}(c)\right) {p'_{\mathtt{B}}}^{\tilde{c}} \le \sum_{c=\tilde{c}}^{2n} \Delta\mathcal{C}_{\mathtt{B}}(c) {p'_{\mathtt{B}}}^c,$$

and thus

$$\sum_{c=0}^{2n} \Delta\mathcal{C}_{\mathtt{B}}(c) {p'_{\mathtt{B}}}^c > 0.$$

As before, it follows that

$$\mathbb{E}_{t, x_t = x_{\mathtt{a}}}[q_t(s{=}\mathtt{B}|x_1^T, W_0)] > \mathbb{E}_{t, x_t = x_{\mathtt{a}}}[q_t(s{=}\mathtt{a}|x_1^T, W_0)].$$

I.e. gradient descent will decrease $\theta_{\mathtt{a}}$, i.e. increase $p(s{=}\mathtt{B}|x{=}x_{\mathtt{a}})$. This results in peaky behavior, and in 100% error rate. □

These observation were shown for this specific constructed simple example, however it can be argued that a similar behavior will usually be observed in other cases. To emphasize: *A uniformly initialized FFNN trained with gradient descent on the CTC loss does not converge to a global optimum, but to a local optima with peaky behavior and 100% error rate.* The global optima of L and all parameters close to that have a perfect 0% error rate without peaky behavior. So this is mostly a problem of the gradient, which tends towards peaky behavior, and the model is too weak to be able to handle peaky behavior.

Simulation 4.13. We use Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$) and Example 4.9, and $T = 16$. We see that the converged FFNN model has peaky behavior, more specifically $p_t(s{=}\mathtt{B}|x_1^T) > 88\%$ for all t, and 100% error rate.

Remark 4.14. If there is a global bias like it is usually the case for neural networks before the $\mathrm{softmax}$, it will reinforce the convergence to peaky behavior because the gradient to the bias will be as in Theorem 4.6.

The FFNN converges towards peaky behavior but cannot learn the peaky alignment because it has only local context. We can argue that a more powerful model with global context can always learn such alignment. As a synthetic experiment, we introduce the *memory model*, which has perfect memory. This is the equivalent behavior of any model which can perfectly overfit. I.e. by construction this is the most powerful model possible. This model is independent from the input x.

Definition 4.15 (Memory model)**.** Define the model $\mathcal{M}^M$ with perfect memory as

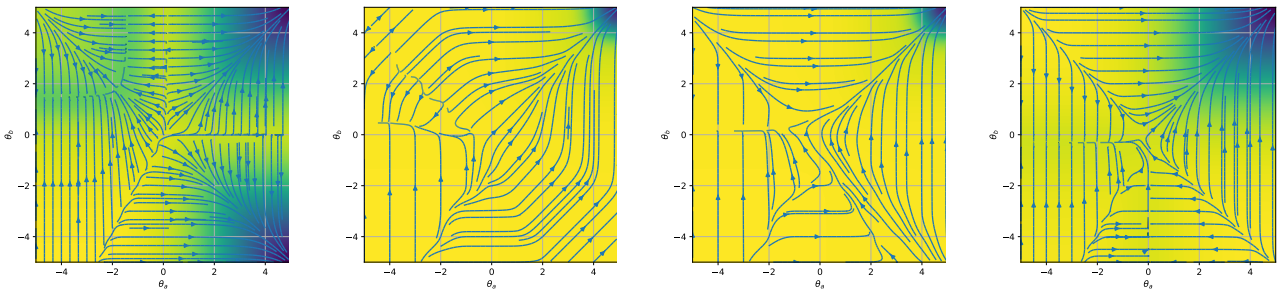
$$p_t(s|x_1^T, \mathcal{M}^M) := \mathrm{softmax}(M[t])[s]$$

for $\theta = M \in \mathbb{R}^{T \times S}$.

Simulation 4.16. For Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$), $T = 100$, the memory model starting from uniform initialization trained with L with gradient descent converges to peaky behavior with $p_t(s{=}\mathtt{B}|x) > 93\%\ \forall t$, i.e. 100% error rate.

5. Role of the blank Label

Recall Remark 3.10. The blank label plays a special role in the CTC topology. We have seen that it is the dominant label, and models tend to become peaky w.r.t. the blank label. It is important to point out that blank can occur anywhere in the alignment, between all other labels. In the common HMM topology in speech recognition, there is no blank, but silence instead. Simply by counting, silence is also the dominant label. Note that in the Wav2Letter (Collobert et al., 2016 ↗) label topology, we have a special repetition label if a label is supposed to repeat on the target



(a) L (CTC) for FFNN (Theorem 4.12).

(b) L_{hybrid} for FFNN (Remark 7.4).

(c) L_{hybrid} for FFNN, stop-grad. on prior (Remark 7.6).

(d) $L_{\text{generative}}$ for generative model (Remark 7.11).

Figure 2. Different loss functions, plotted over the model parameter space, for Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$) & Example 4.9 ($x_1^T$). A darker color represents a lower loss value. We also plot the negative gradient map, such that we can see for every possible parameter setting, where gradient descent leads to. A uniform distribution initialization starts in the center with $\theta_{\mathtt{a}} = \theta_{\mathtt{B}} = 0$. We can also see local optima. All models with parameters in the left upper area have peaky behavior. The constructed optimal solution is in the right upper corner.

side. The Chain model (Povey et al., 2016) label topology has two states per phoneme, where the second optional looping label is interpret as blank – however, it is not shared, and thus not dominant. Both Wav2Letter and Chain have a dominant silence label as well. However, they use other training criteria which do not necessarily lead to peaky behavior.

CTC trained with dominant silence label results in peaky models. However, the label topology usually allows that silence can only occur before or after whole words, not within words, where a word consists of multiple phonemes. We will argue that this label topology is suboptimal for a loss like CTC with peaky behavior. We will construct an even simpler example for the further demonstration.

Example 5.1. Let us consider the single word "ping" which consists of the phoneme sequence "p ih ng". With the CTC topology, blank is allowed anywhere. With the standard HMM topology, silence is allowed only before "p" and after "ng". We will demonstrate that this restriction is suboptimal together with peaky behavior which results by the loss L, and a blank label which can occur everywhere is better. We further construct corresponding input features

$$x_1^T = \begin{pmatrix} 0 & 0\,1 & 1\,0 & 0\,0 & 0\,0 & 0 \\ 0 & 0\,0 & 0\,1 & 1\,0 & 0\,0 & 0 \\ 0 & 0\,0 & 0\,0 & 0\,1 & 1\,0 & 0 \\ 1 & 1\,0 & 0\,0 & 0\,0 & 0\,1 & 1 \end{pmatrix}^{\underbrace{\quad}_{20\times}\ \underbrace{\quad}_{10\times}\ \underbrace{\quad}_{30\times}\ \underbrace{\quad}_{20\times}\ \underbrace{\quad}_{20\times}}$$

with $T = 100$. This example is constructed such that the time accurate (optimal) alignment is $s_1^T = (\mathtt{B}^{20}, \mathtt{a}^{10}, \mathtt{b}^{30}, \mathtt{c}^{20}, \mathtt{B}^{20})$, and an optimal model is

$$p(s|x) := \begin{cases} 1, & x = x_s \\ 0, & \text{else} \end{cases}$$

with x_s accordingly. The posteriors of this model are visualized in Figure 3a.

Remark 5.2. Consider the case for the HMM topology with silence, i.e. we allow all alignments matching the regular expression $(\mathtt{B}^*, \mathtt{a}^+, \mathtt{b}^+, \mathtt{c}^+, \mathtt{B}^*)$. Peaky behavior results in alignments of the form $(\mathtt{B}^+, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{B}^+)$. Optimal posteriors of this alignment are visualized in Figure 3b. With the CTC topology, i.e with blank, we allow all alignments matching the regular expression $(\mathtt{B}^*, \mathtt{a}^+, \mathtt{B}^*, \mathtt{b}^+, \mathtt{B}^*, \mathtt{c}^+, \mathtt{B}^*)$. We get the peaky behavior with alignments of the form $(\mathtt{B}^+, \mathtt{a}, \mathtt{B}^+, \mathtt{b}, \mathtt{B}^+, \mathtt{c}, \mathtt{B}^+)$. Optimal posteriors of this alignment are visualized in Figure 3c. Comparing both possible posteriors and alignments, we see that the HMM topology is much more restricted, and peaky behavior compresses a whole word as short as possible. This is clearly suboptimal, as it was also experimentally observed (Zeyer et al., 2017).

Remark 5.3. This implies that a blank label can help in general for CTC training. This is also true if the modeling is performed on phone-level, and in fact this seems to work well in practice (Sak et al., 2015; Miao et al., 2016).

6. Role of the Ratio T/N

From Corollary 3.13 we can see that the peaky behavior is amplified the higher the ratio $\frac{T}{N}$ purely due to the label topology.

Simulation 6.1. We use the same example from Example 5.1 for the target $y_1^N = \mathtt{abc}$ ($N = 3$), CTC label topology (including blank) and x_1^T synthetically constructed for varying T, where $p(x)$ stays uniform. We want to study the effect of the ratio $\frac{T}{N}$ on the peaky behavior and convergence behavior. We can measure the average $q(\mathtt{B})$ for a uniform distribution p to get the the initial gradient due to the label topology and $\frac{T}{N}$. We train a simple long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) model with CTC, and measure the resulting average $q(\mathtt{B})$ which shows how dominant $\mathtt{B}$ has become. The model learns perfectly in all cases, although with varying convergence speed. We plot

(a) Optimal with silence. (b) Peaky with silence. (c) Peaky with blank.

Figure 3. Like in Figure 1, colors represents the posterior outputs for the labels "p", "ih" and "ng". The dotted gray output represents silence or blank. The posteriors are constructed such that they represent a Viterbi alignment.

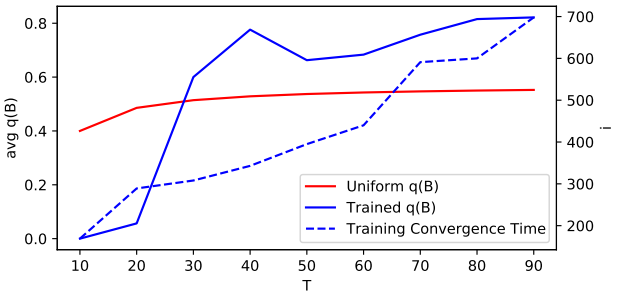


Figure 4. This uses the target sequence abc with $N = 3$, and x_1^T constructed as in Example 5.1 but downscaled accordingly. We plot $\frac{1}{T}\sum_t q_t(\mathtt{B})$ for a uniform distribution p, and of a CTC-trained LSTM model. We also plot the convergence time, which measures the number of steps i until $L < 1$, i.e. lower is better.

the results in Figure 4. For $T \leq 20$ ($\frac{T}{N} < 7$), we observe that the model does not converge to peaky behavior, while it tends to for larger T. Also, we see that the convergence speed decreases with increasing T, which indicates that a high $\frac{T}{N}$ ratio is harder to learn.

7. Avoiding Peaky Behavior by Other Losses

For good error rate performance, avoiding peaky behavior might not be needed. However, peaky behavior can be problematic in certain cases, e.g. when an application requires to not use the blank label (e.g. for time accurate phoneme or word boundaries in the alignment), or for the usage of local-context models like FFNNs, as we have shown.

We extend the training criterion by a label prior and show that this does not lead to peaky behavior. We will demonstrate that this solves the convergence issues for FFNNs. This loss is originated in the hybrid NN-HMM model case (Bourlard & Morgan, 1989; Franzini et al., 1990). where the generative acoustic model integrates a discriminative NN by

$$p(x_1^T|s_1^T) \propto \frac{p(s_1^T|x_1^T)}{p(s_1^T)}.$$

The difference here to the usual CTC model is the label prior model $p(s_1^T)$ in the denominator. It can even be useful for decoding with CTC models (Miao et al., 2015). We usually simplify the *prior model* $p(s_1^T)$ as

$$p(s_1^T|\mathcal{M}_{\text{prior}}) = \prod_t p(s_t|\mathcal{M}_{\text{prior}}).$$

Other prior variants are possible (Naoyuki Kanda, 2016).

Definition 7.1. Define the *hybrid model loss* as

$$L_{\text{hybrid}} := -\log \sum_{s_1^T:y_1^N} \frac{p(s_1^T|x_1^T,\mathcal{M})}{p(s_1^T|\mathcal{M}_{\text{prior}})}$$
$$= -\log \sum_{s_1^T:y_1^N} \prod_t \frac{p_t(s_t|x_1^T,\mathcal{M})}{p(s_t|\mathcal{M}_{\text{prior}})}.$$

L_{hybrid} was used in (Haffner, 1993; Zeyer et al., 2017). There are multiple options how to estimate the prior $p(s)$.

Remark 7.2. Having a prior model $p(s|\mathcal{M}_{\text{prior}}) = \text{softmax}(b_{\text{prior}})[s]$ as a separate model with its own parameters, and trained jointly with the posterior model $p_t(s|x_1^T,\mathcal{M}_{\text{posterior}})$ will lead exactly to an inverse prior estimation. I.e. consider that a label $\mathring{s}$ maximizes the posterior model $p(s_1^T|x_1^T)$, i.e. it would occur most often in Viterbi alignments (disregarding the prior model). Then, the prior model would be optimal for minimizing L_{hybrid} when it *minimizes* $p(\mathring{s})$, i.e. $p(\mathring{s}) < p(s)$. This is counter intuitive and does not reflect what the prior model should represent. Also, it would only reinforce the peaky behavior.

Given this remark, it becomes clear that $p(s)$ should be estimated based on the posterior model in some way.

Definition 7.3. Letting the prior model $p(s)$ be estimated as the expected output of the posterior model, which we also call *softmax prior*, cf. (Manohar et al., 2015), i.e.

$$p(s) := \frac{1}{T}\sum_t p_t(s|x_1^T).$$

Remark 7.4. Just as in Theorem 4.12, for the same example, with the same parameterization, we plot the loss function L_{hybrid} in Figure 2b. We can see that there is only a single global optimum at $\theta_a = \theta_b = \infty$, and also that we reach that global optimum at a uniform distribution initialization ($\theta_a = \theta_b = 0$).

Simulation 7.5. We use the FFNN model (Definition 4.8) with softmax prior (Definition 7.3), and Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$), x_1^T as in Example 4.9. We can see that training with the loss L_{hybrid} will not get peaky behavior. The model converges to the time accurate (optimal) alignment. I.e. it converges towards $W = \left(\begin{smallmatrix}\infty & 0\\ 0 & \infty\end{smallmatrix}\right)$ with 0% error rate.

Remark 7.6. The common training of hybrid NN-HMM models would keep the prior model $p(s)$ fixed while updating the posterior model $p(s|x)$. In our formulation, that is equivalent by defining

$$p(s|\mathcal{M}_{\text{prior-sg}}) := \text{stop-gradient}\left(\frac{1}{T}\sum_t p_t(s|x_1^T,\mathcal{M})\right),$$

where $\mathrm{stop\text{-}gradient}$ is the identity function, but the gradient is defined as zero. In that case, the gradient of L_{hybrid} will look different. We can see the effect in Figure 2c. We observe a slightly different behavior of the gradient map. However, starting with uniform distribution initialization ($\theta_a = \theta_b = 0$) will converge to the same global optimum.

Remark 7.7. When the prior is kept fixed but the dominance of $\mathring{s}$ is strong enough, this still can lead to peaky behavior. Alternatively, if the prior is too strong or not well estimated, this can result in the dominance of another label $\hat{s} \neq \mathring{s}$ in $\frac{p(\hat{s}|x)}{p(\hat{s})}$, and can get peaky behavior where this other label $\hat{s} \neq \mathring{s}$ dominates. We observed this behavior in some cases experimentally, where we used an online moving average of $p(s \mid x)$ for the prior. This online moving average estimation can be unstable, esp. in early stages of training.

Remark 7.8. A stable recipe is to estimate the prior on the whole training data as in Definition 7.3, then to calculate the soft alignment q_t for the whole training data, and then to update the posterior model while keeping the soft alignments fixed. An approximation of using the soft alignment are hard Viterbi alignments. This is very similar to the standard training procedure for hybrid NN-HMMs with framewise cross entropy (CE).

The peaky behavior was only observed for discriminative models, while similar training criteria have been used for generative models. We now study the convergence behavior and peaky behavior of generative models. We use a simple generative model (without transition probabilities)

$$\begin{aligned} p(x_1^T|y_1^N, \mathcal{M}) &\propto \sum_{s_1^T : y_1^N} p(x_1^T|s_1^T, \mathcal{M}) \\ &= \sum_{s_1^T : y_1^N} \prod_t p(x_t|s_t, \mathcal{M}). \end{aligned}$$

Definition 7.9 (Loss for generative model)**.** Define the loss

$$\begin{aligned} L_{\text{generative}} &:= -\log \sum_{s_1^T : y_1^N} p(x_1^T|s_1^T, \mathcal{M}) \\ &= -\log \sum_{s_1^T : y_1^N} \prod_t p(x_t|s_t, \mathcal{M}). \end{aligned}$$

We follow a similar construction as for the FFNN (Theorem 4.12) for Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$), x_1^T as in Example 4.9.

Definition 7.10 (Generative model)**.** For $S = \{\mathtt{B}, \mathtt{a}\}$, $x \in \{x_\mathtt{B}, x_\mathtt{a}\}$, we define

$$p(x|s{=}\mathtt{a}) = \mathrm{softmax}((\theta_\mathtt{a}, -\theta_\mathtt{a})) \left[\begin{cases} 1, & x = x_\mathtt{a} \\ 2, & x = x_\mathtt{B} \end{cases} \right],$$

$$p(x|s{=}\mathtt{B}) = \mathrm{softmax}((-\theta_\mathtt{B}, \theta_\mathtt{B})) \left[\begin{cases} 1, & x = x_\mathtt{a} \\ 2, & x = x_\mathtt{B} \end{cases} \right].$$

Remark 7.11. We plot the loss $L_{\text{generative}}$ in Figure 2d. We can see that there is only a single global optimum at $\theta_\mathtt{a} = \theta_\mathtt{B} = \infty$ with $L = 0$. When we start with uniform distribution ($\theta_\mathtt{a} = \theta_\mathtt{B} = 0$), we reach that global optimum. This global optimum is the optimal non-peaky solution, and the error rate becomes 0%.

Remark 7.12. We can reparameterize the model as

$$p(x|s) := \begin{cases} \theta[s], & x = x_s \\ 1 - \theta[s], & x \neq x_s \end{cases}.$$

For $\theta \equiv 1$, we get the unique global optimum with $L = 0$, and this global optimum has no peaky behavior. We get our initial uniform distribution with $\theta_0 \equiv 0.5$. We will get a similar gradient as in Equation (1), however, this error signal is for the generative model $p(x_t|s_t)$.

$$\begin{aligned} \frac{\partial L_{\text{generative}}}{\partial \theta} &= -\sum_{s,t} q_t(s) \frac{\partial}{\partial \theta} \log p(x_t|s, \theta) \\ &= -\sum_{s,x} \left(\sum_{t, x_t = x} q_t(s) \right) \frac{\partial}{\partial \theta} \log p(x|s, \theta). \end{aligned}$$

Assume $\theta_i[s] \notin \{0, 1\}$. Then for any s,

$$\frac{\partial L_{\text{generative}}}{\partial \theta_i[s]} = -\frac{\sum_{t, x_t = x_s} q_t(s, \theta_i)}{\theta_i[s]} + \frac{\sum_{t, x_t \neq x_s} q_t(s, \theta_i)}{1 - \theta_i[s]}.$$

Following a similar calculation as in Theorem 4.12, we can explicitly calculate that for $\theta_0 = 0.5$,

$$\begin{aligned} \frac{\sum_{t, x_t = x_\mathtt{B}} q_t(s{=}\mathtt{B}, \theta_0)}{\sum_t q_t(s{=}\mathtt{B}, \theta_0)} &= \frac{19n^2 - 1}{32n^2 - 2} > 59\% \\ \frac{\sum_{t, x_t = x_\mathtt{a}} q_t(s{=}\mathtt{a}, \theta_0)}{\sum_t q_t(s{=}\mathtt{a}, \theta_0)} &= \frac{11n^2 + 6n + 1}{16n^2 + 12n + 2} \geq 60\%. \end{aligned}$$

Thus $\frac{\partial L_{\text{generative}}}{\partial \theta_0[s]} < 0$, i.e. $\theta_1[s] > \theta_0[s]$.

Simulation 7.13. For $L_{\text{generative}}$, for Example 3.2 ($\mathtt{B}^*\mathtt{a}^+\mathtt{B}^*$), x_1^T as in Example 4.9, $T = 16$, and the model parameterized as in Definition 7.10, we see that the model converges to the global optimum with time accurate (optimal) alignment, i.e. it does not get peaky behavior and has 0% error rate.

8. Conclusions

We contribute a formal analysis to discover the causes for peaky behavior. We found this is a property of local convergence which tends towards peaky behavior when starting from a uniform distribution. This is due to the label topology and the dominance of one label such as blank or silence. We also explained the role of the blank label, the role of the ratio $\frac{T}{N}$, and the role of a label prior model in CTC and full-sum training. We have shown that peaky behavior should be avoided without a blank label. Even with the blank label and CTC topology, peaky behavior can be suboptimal, as was demonstrated on a simple example with a simple FFNN. We extended the training criterion to handle and avoid the peaky behavior by including a label prior.

References

Bacchiani, M., Senior, A. W., and Heigold, G. Asynchronous, online, GMM-free training of a context dependent acoustic model for speech recognition. In *INTERSPEECH*, pp. 1900–1904, 2014.

Bengio, Y., De Mori, R., Flammia, G., and Kompe, R. Global optimization of a neural network-hidden markov model hybrid. In *IEEE International Joint Conference on Neural Networks*, pp. 789–794, Seatle, WA, USA, November 1991.

Bluche, T., Ney, H., Louradour, J., and Kermorvant, C. Framewise and CTC training of neural networks for handwriting recognition. In *Document analysis and recognition (icdar), 2015 13th international conference on*, pp. 81–85. IEEE, 2015.

Bourlard, H. and Morgan, N. A continuous speech recognition system embedding MLP into HMM. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 186–193, Denver, CO, USA, November 1989.

Collobert, R., Puhrsch, C., and Synnaeve, G. Wav2letter: an end-to-end convnet-based speech recognition system. Preprint arXiv:1609.03193, 2016.

Franzini, M., Lee, K.-F., and Waibel, A. Connectionist viterbi training: a new hybrid method for continuous speech recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 425–428, Albuquerque, NM, USA, April 1990.

Graves, A. Sequence transduction with recurrent neural networks. Preprint arXiv:1211.3711, 2012a.

Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012b. ISBN 978-3-642-24796-5. doi: 10.1007/978-3-642-24797-2. URL http://dx.doi.org/10.1007/978-3-642-24797-2.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376. ACM, 2006.

Haffner, P. Connectionist speech recognition with a global MMI algorithm. In *EUROSPEECH*, 1993.

Hennebert, J., Ris, C., Bourlard, H., Renals, S., and Morgan, N. Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems. In *Eurospeech*, pp. 1951–1954. International Speech Communication Association, 1997.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, X. and Wu, X. Labeling unsegmented sequence data with DNN-HMM and its application for speech recognition. In *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*, pp. 10–14. IEEE, 2014.

Manohar, V., Povey, D., and Khudanpur, S. Semi-supervised maximum mutual information training of deep neural network acoustic models. In *Proceedings of INTERSPEECH*, 2015.

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roučka, v., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., and Scopatz, A. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

Miao, Y., Gowayyed, M., and Metze, F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 167–174. IEEE, 2015.

Miao, Y., Gowayyed, M., Na, X., Ko, T., Metze, F., and Waibel, A. An empirical exploration of CTC acoustic models. In *ICASSP*, pp. 2623–2627. IEEE, 2016.

Naoyuki Kanda, Xugang Lu, H. K. Maximum a posteriori based decoding for CTC acoustic models. In *Interspeech*, pp. 1868–1872, 2016. doi: 10.21437/Interspeech.2016-71. URL http://dx.doi.org/10.21437/Interspeech.2016-71.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Interspeech*, pp. 2751–2755, 2016.

Sak, H., Senior, A., Rao, K., and Beaufays, F. Fast and accurate recurrent neural network acoustic models for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Sak, H., Shannon, M., Rao, K., and Beaufays, F. Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping. In *Proc. of Interspeech*, 2017.

Senior, A. and Robinson, T. Forward-backward retraining of recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 743–749. Citeseer, 1996.

Senior, A., Heigold, G., Bacchiani, M., and Liao, H. GMM-free DNN acoustic model training. In *ICASSP*, 2014.

TensorFlow Development Team. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Yan, Y., Fanty, M., and Cole, R. Speech recognition using neural networks with forward-backward probability generated targets. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 4, pp. 3241–3241. IEEE Computer Society, 1997.

Zeyer, A., Beck, E., Schlüter, R., and Ney, H. CTC in the context of generalized full-sum HMM training. In *Interspeech*, pp. 944–948, Stockholm, Sweden, August 2017.

Zeyer, A., Alkhouli, T., and Ney, H. RETURNN as a generic flexible neural toolkit with application to translation and speech recognition. In *Annual Meeting of the Assoc. for Computational Linguistics*, Melbourne, Australia, July 2018.

Zhang, C. and Woodland, P. C. Standalone training of context-dependent deep neural network acoustic models. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5597–5601. IEEE, 2014.