The Conditional Bernoulli and its Application to Speech Recognition

Sean Robertson

April 17, 2020

1 Motivations

A major challenge in speech recognition involves converting a variable number of speech frames $\{x_t\}_{t \in [1,T]}$ into a variable number of transcription tokens $\{c_\ell\}_{\ell \in [1,L]}$, where $L \ll T$. In hybrid architectures, c_ℓ are generated as a by-product of transitioning between states s_t in a weighted finite-state transducer. In end-to-end neural ASR, this process is commonly achieved either with Connectionist Temporal Classification (CTC) [11] or sequence-to-sequence (seq2seq) architectures [2]. The former introduces a special blank label; performs a one-to-many mapping $c_\ell \mapsto \tilde{c}_t^{(i)}$ by injecting blank tokens until the transcription matches length T in all possible configurations (i) during training; and removes all blank labels during testing. Seq2seq architectures first encode the speech frames x_t into some encoding h, then some separate recurrent neural network conditions on h to generate the token sequence c_t.

In 2017, Luo et al. developed a novel end-to-end speech recognizer. Given a prefix of acoustic feature frames including the current frame $\{x_{t'}\}_{t' \in [t,T]}$ and a prefix of Bernoulli samples excluding the current frame $\{b_{t'}\}_{t' \in [t+1,T]}$, the recognizer produces a Bernoulli sample for the current frame $B_t \sim P(b_t | x_{\leq t}, b_{<t})$, plus or minus some additional conditioned terms. Whenever $B_t = 1$, the model "emits" a token drawn from a class distribution conditioned on the same information $C_t \sim P(c_t | x_{\leq t}, b_{<t})$. The paper had two primary motivations. First, though it resembles a decoder in a *seq2seq* architecture [2], it does not need to encode the entire input sequence x_t before it can start making decisions about what was said, making it suitable to online recognition. Second, we can interpret the emission points, or "highs," of the Bernoulli sequence $B_t = 1$ as a form of hard alignment: the token output according to C_t is unaffected by speech $x_{>t}$[1].

Because of the stochasticity introduced by sampling B_t discretely, the network cannot determine the exact gradient for parameterizations of B_t. Thus,

[1] This is not necessarily a synchronous alignment. $B_t = 1$ may occur well after whatever caused the emission. The last high $\arg\max_{t' < t} B_{t'} = 1$ cannot be assumed to bound the event to times after t' for the same reason. Finite t and vanishing gradients will force some synchronicity, however.

the authors rely on an estimate of the REINFORCE gradient [20]:

$$\frac{\partial R}{\partial \theta} = \mathbb{E}_b \left[\sum_{t=1}^{T} \left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t' \geq t} R_{t'} \right) \frac{\partial}{\partial \theta} \log P(b_t | b_{<t}, c_{<\ell_t}) \right) \right] \tag{1}$$

where

$$R_t = \begin{cases} \log P(C_t = c_{\sum_{t'<t} b_{t'}} | x_{\leq t}, b_{<t}, c_{\sum_{t'<t-1} b_{t'}}) & \text{if } B_t = 1 \\ 0 & \text{if } B_t = 0 \end{cases} \tag{2}$$

The reward (eq. (2)) is the log probability of the k-th class label, where k counts the number of high Bernoulli values up to and including time step t. The return for time step t accumulates the instantaneous rewards for all non-past time steps $t' \geq t$.

In practice, using eq. (1) is very slow to train and yields mixed results. The authors found it was necessary to add a baseline function and an entropy function in order to converge. In a later publication [13], a bidirectional model[2] used Variational Inference to speed up convergence, though this failed to improve the overall performance of the model on the TIMIT corpus. The mixed performance and convergence of these models was blamed on the high-variance gradient estimate of eq. (1) [13].

We believe that the performance and convergence issues of these models are not due, at least in whole, to a high-variance estimate. Instead, we propose that the training objective has two other critical issues.

First, in order to ensure the total number of high Bernoulli values matched the total number of labels L during training, i.e. $\sum_t b_t = L$, the authors would force later samples to some specific value. This implies that $B_t \not\sim P_B(b_t|\ldots)$, making the Monte Carlo estimate of eq. (1) biased.

Second, under the current regime, $B_t = 0$ maximizes the per-step reward by avoiding a negative reward associated with $\log P(c_t|\ldots)$. Equation (1) accumulates future rewards to mitigate this, but the choice to do so biases the system to emit as soon as possible to reduce the number of total frames accumulating negative rewards. This bias could explain why, without an additional "entropy penalty," the model would learn to emit entirely at the beginning of the utterance [14]. It could interact with the previous issue to produce a model that learns to immediately and repeatedly emit without stopping, since $b_{\geq L} = 0$, pushing $P_B(b_t|\ldots) \to 1$.

To solve both of these problems, we propose replacing the T independent Bernoulli random variables B_t sampled during training with a single sample B from the Conditional Bernoulli (CB) distribution during training. The CB conditions on the required number of high trials, which will make the objective well-defined during testing and avoiding the first problem entirely. The CB can be decomposed into a series of T Bernoulli trials that condition on past trial results, similar to eq. (2), but with an arbitrary ordering of trials. We can alternate forward and backward conditioning to mitigate the system's tendency to

[2] Forgoing the motivation for online speech recognition.

emit early. If we also train the model to fit a certain prior, CB training elegantly segues into the same Bernoulli testing scheme. In addition to modifying eq. (1), we will also show how the CB can be relaxed to a continuous variable for use in Straight-Through estimators [3, 12] or RELAX-like estimators [15, 9].

2 The Conditional Bernoulli

2.1 Definitions

The Conditional Bernoulli distribution [6, 5], sometimes called the Conditional Poisson distribution [1, 4], is defined as

$$P\left(b \middle| \sum_t b_t = k; w\right) = \frac{\prod_t w_t^{b_t}}{\sum_{\left\{b' : \sum_t b'_t = k\right\}} \prod_t w_t^{b'_t}} \tag{3}$$

Where $w_t = p_t/(1-p_t)$ are the odds/weights of a Bernoulli random variable $B_t \sim P(b_t; w_t) = p_t^{b_t}(1-p_t)^{(1-b_t)} = w_t^{b_t}/(1+w_t)$. Equation (3) reads as "what is the probability that Bernoulli random variables $B = \{B_t\}_{t \in [1,T]}$ have values $\{b_t\}_t$, given that exactly k of them are high ($\sum_t b_t = k$)?" Letting $K = \sum_t B_t$, K is a random variable that counts the total number of "highs" in a series of Bernoulli trials. K is distributed according to the Poisson-Binomial (PB) distribution, a generalization of the Binomial distribution for when $p_i \neq p_j$. It is defined as

$$\begin{aligned} P(K = k; w) &= \sum_{\left\{b : \sum_t b_t = k\right\}} P(b; w) \\ &= \left(\prod_{t=1}^{T} 1 + w_t\right)^{-1} \sum_{\left\{b : \sum_t b_t = k\right\}} \prod_{t=1}^{T} w_t^{b_t} \end{aligned} \tag{4}$$

If we use eq. (4) to marginalize out K from eq. (3), we recover the independent Bernoulli probabilities:

$$\begin{aligned} P(b; w) &= \sum_{k=0}^{T} P(b, k; w) = \sum_{k=0}^{T} P(b|k; w) P(k; w) \\ &= P(b|k'; w) P(k'; w) \text{ for exactly one } k' = \sum_t b_t \\ &= \left(\prod_t 1 + w_t\right)^{-1} \frac{\prod_{t=1}^{T} w_t^{b_t}}{\sum_{\left\{b' : \sum_t b'_t = k'\right\}} \prod_{t=1}^{T} w_t^{b'_t}} \left(\sum_{\left\{b' : \sum_t b'_t = k'\right\}} \prod_{t=1}^{T} w_t^{b'_t}\right) \\ &= \prod_{t=1}^{T} (1 + w_t)^{-1} w_t^{b_t} \end{aligned} \tag{5}$$

Which is to say that, if we do not have knowledge of the number of highs *a priori*, assuming a Poisson-Binomial prior, the probability of sample B is the product of the probabilities of the outcomes of T independent Bernoulli trials.

Direct calculation of equation eq. (3) involves summing over T-choose-k products of k odds, making it infeasible for large T and k. To combat this, Chen and Liu [5] propose a number of alternative algorithms where the sample B is constructed by iteratively deciding on the individual values of B_i. We will not only use these algorithms for efficiency: we will also use them to factor the CB distribution into useful forms for different objectives.

To better describe these algorithms, we define the set of indices $t \in [1, T] = I$ s.t. $B = \{B_t\}_{t \in I}$. The set $A \subseteq I$ maps to some sample B such that all the high Bernoulli variables' indices can be found in A, i.e. $B_t = 1 \Leftrightarrow t \in A$. Then the CB can be restated as

$$P(A|k; w) = \frac{\prod_{a \in A} w_a}{R(k, I; w)} \tag{6}$$

where

$$R(v, S; w) = \sum_{\{A' \subseteq S : |A'| = v\}} \prod_{a \in A'} w_a \tag{7}$$

normalizes over all possible k-tuples of w_i in some set S. Equation (7) can be considered a generalization of T-choose-k: T-choose-k can be recovered by setting all $w_t = 1$. If we identify the product of weights from a set A as a weight indexed by A (i.e. $\prod_{a \in A} w_a \mapsto w'_A$), we can interpret eq. (6) as a categorical distribution.

The Draft Sampling procedure [5] recursively builds A by choosing a new weight to add to an ordered set. We use $j \in [1, T]$ to index elements of I in the order in which they are drafted into A: $I = \{t_j\}_j$, $A_j = (t_1, t_2, \ldots t_j)$, and $A^c_j = I \setminus A_j = \{t_{j+1}, t_{j+2}, \ldots, t_T\}$. Then the probability that some $t \in A^c_{j-1}$ is the j-th sample to be drafted into A is defined as

$$P(t \in A_j | A_{j-1}, k; w) = \frac{w_t R(k - j, A^c_{j-1} \setminus \{t\}; w)}{(k - j + 1) R(k - j + 1, A^c_{j-1}; w)} \tag{8}$$

Terms in both the numerator and denominator of eq. (8) sum over suffix sets of length $k - j + 1$ that could be appended to A_{j-1} to get a k-tuple A. The numerator is the sum of products of odds including w_t. The conditional probability is conditioned on the remaining ("future") odds with respect to j, as well as whatever samples t_j were chosen in the past. The total probability of a drafted sample is

$$\begin{aligned}
P(A_k|k;w) &= \prod_{j=1}^{k} P(t_j \in A_j|A_{j-1},k;w) \\
&= \prod_{j=1}^{k} \frac{w_{t_j} R(k-j, A_j^c, k)}{(k-j+1)R(k-j+1, A_{j-1}^c)} \\
&= \left(\prod_{j=1}^{k} w_{t_j} \right) \frac{R(0, A_k^c)}{k! R(k, I)} \\
&= \frac{1}{k!} P(A|k,w) \qquad (9)
\end{aligned}$$

Section 2.1 produces almost the same probability as the Conditional Bernoulli, except for the factorial term. The factorial term accounts for the fact that samples are drafted into A_k in some fixed order. Summing over the probabilities of the $k!$ possible permutations of A_k yields the Conditional Bernoulli. We will call the distribution defined in the Draft Bernoulli (DB). Though the DB is not the same distribution as the CB, an expected value over the DB will be the same as that over the CB as long as the order of samples in A_k is ignored by the value function.

The ID-Checking Sampling procedure [5] is another useful treatment of the CB. This procedure builds A by iterating over Bernoulli trials and making binary decisions whether to include the trial in A. First, choose and fix an order j in which samples I will potentially be added to A. Let $A_{r_j,j} \subseteq A_j = (t_1, t_2, \ldots, t_j)$ be the subset of r_j samples ($|A_{r_j,j}| = r_j$) that have been added to A. At every step j, we choose to either add t_j to $A_{r_{j-1},j-1}$ and recurse on $A_{r_j,j} = A_{r_{j-1},j-1} \cup \{t_j\}$ or exclude t_j and recurse on $A_{r_j,j} = A_{r_{j-1},j-1}$. The probability of including t_j is

$$P(t_j \in A_{r_j,j}|A_{r_{j-1},j-1},k;w) = \frac{w_{t_j} R(k - r_{j-1} - 1, A_j^c; w)}{R(k - r_{j-1}, A_{j-1} u^c; w)} \qquad (10)$$

From the perspective of Bernoulli trials, $P(t_j \in A_{r_j,j}|\ldots) = P(B_{t_j} = 1|k - r_j; w)$. Equation (10) can be interpreted as the probability that B_{t_j} is high, given that $k - r_j$ remaining trials must be high. Like in eq. (8), the numerator and denominator of eq. (10) consist of products of weights of possible suffixes. The numerator only includes suffixes where w_{t_j} is a multiplicand.

The joint probability of a prefix of Bernoulli trials $b_{t_{\leq j}} = (b_{t_1}, b_{t_2}, \ldots, b_{t_j})$ using eq. (10) equals

$$\begin{aligned}
P(b_{t_{\leq j}}|k - r_j; w) &= \prod_{j'=1}^{j} P(b_{t_{j'}}|k - r_{j'}; w) \\
&= \prod_{j'=1}^{j} \frac{w_{t_{j'}}^{b_{t_{j'}}} R(k - r_{j'}, A_j^c; w)}{R(k - r_{j'-1}, A_{j-1}^c; w)} \qquad (11)
\end{aligned}$$

The dependence on prior trials is implicit in the $r_{j'}$ term. We will call the family of distributions over different prefixes the ID-checking Bernoulli (IDB). When the prefix is the length of the entire sequence $j = T$, $P(b_{t_{\leq T}}|k - r_T; w) = P(b|k; w)$ and the IDB distribution matches the CB distribution.

We will find a novel third decomposition useful. This method combines the ID-Checking and Drafting methods so that the draft at a given step must come from a bounded suffix of weights. Define $A_{r,j_r} \subseteq A_{j_r} = (t_1, t_2, \dots t_{j_r})$ to be the r samples of A_{j_r} that have been added to A. Define the probability that the next sample j is drafted from $A^c_{j_r}$ to be

$$P(t_j \in A_{r,j_r}|k - r, j_{r-1}; w) = \frac{w_{t_j} R(k - r, A^c_{t_j}; w)}{R(k - r + 1, A^c_{t_{j_{r-1}}}; w)} \tag{12}$$

The draft is bound to the suffix $A^c_{t_{j_{r-1}}} = (t_{j_{r-1}+1}, t_{j_{r-1}+2}, \dots, t_{j_T})$. Further, the draft requires that if t_j is the r-th draft, the remaining drafts must come from indexed values $t_{>j}$. To balance the restriction, earlier t_j will be more probable than later t_j to be drafted earlier. Using the fact that $R(k - r + 1, A^c_{t_j}; w) = w_{t_j+1} R(k - r, A^c_{t_j+1}; w) + R(k - r + 1, A^c_{t_j+1}; w)$, it is easily shown via induction that $R(k-r+1, A^c_{t_{j_{r-1}}}; w) = \sum_{j=j_{r-1}+1}^{T} w_{t_j} R(k-r, A^c_{t_j}; w)$, proving that eq. (12) is a valid probability distribution. The probability of a draft prefix is calculated as

$$\begin{aligned} P(A_{r,j_r}|k - r; w) &= \prod_{r'=1}^{r} P(t_{j_{r'}} \in A_{r',j_{r'}}|k - r', j_{r'-1}; w) \\ &= \prod_{r'=1}^{r} \frac{w_{t_{j_{r'}}} R(k - r', A^c_{t_{j_{r'}}}; w)}{R(k - r' + 1, A^c_{t_{j_{r'-1}}}; w)} \\ &= \left(\prod_{r'=1}^{r} w_{t_{j_{r'}}} \right) \frac{R(k - r, A^c_{t_{j_r}}; w)}{R(k, I; w)} \end{aligned} \tag{13}$$

We call this distribution the Bounded Bernoulli (BB). When $r = k$, the BB matches the CB. The BB fixes the multiple orderings problem of the DB. The conditional probabilities of eq. (12) can be efficiently calculated using intermediate values when calculating R using Method 2 from [5].

Outside of statistics, Swersky et al. [18] linked the CB distribution with the goal of choosing a subset of k items from a set of N alternatives. In this case, the N alternatives are class labels, where one or more class labels may be active at a time. Models could be trained in a Maximum-Likelihood setting using the CB distribution: $B_n = 1$ implies class n is present and the probability of the data can be estimated via eq. (3). The authors note that it was insufficient to rely on the implicit prior induced by training via eq. (3) and had to explicitly learn and condition on it.

Xie and Ermon [23] approximates the T-choose-k sampling procedure by using a top-k procedure called Weighted Reservoir Sampling. This procedure produces samples in an identical fashion to the Plackett-Luce (PL) distribution

[24], which has also been explored in the realm of gradient estimation [8]. While the PL distribution has a similar construction to the DB, its top-k rankings do not have a uniform distribution over permutations and, as such, the PL does not match the expectation of the CB. Nonetheless, estimators involving the DB can be trivially modified to sample from the PL.

2.2 REINFORCE Objective

From section 1, we are interested in sampling T Bernoulli random variables such that the total number of emissions/highs matches the number of tokens L during training. We will start by considering the probability of a token sequence $c = \{c_\ell\}_{\ell \in [1,L]}$ under a model and work our way to a REINFORCE objective. For brevity, we supress conditioning on the acoustic data $\{x_t\}_{t\in[1,T]}$ and model parameters.

$$\begin{aligned} P(c) &= P(c, L) \\ &= \sum_b P(c, b, L) \\ &= \sum_b P(b, L) P(c|b) \\ &= \mathbb{E}_{b;L}\left[P(c|b)\right] \end{aligned} \tag{14}$$

Where $P(c) = P(c, L)$ follows from the fact that L is a deterministic function of c. We use the notation $b; L$ to indicate that the expectation is over the joint probability of variable B and some fixed L, but not B given L.

Taking the log, we get

$$\begin{aligned} \log P(c) &= \log \mathbb{E}_{b;L}\left[P(c|b)\right] \\ &\geq \mathbb{E}_{b;L}\left[\log P(c|b)\right] \end{aligned}$$

Where we have used Jensen's Inequality to establish a lower bound. Calling the bound R and differentiating with respect to some parameter θ, we get

$$\frac{\partial R}{\partial \theta} = \frac{\partial}{\partial \theta} \mathbb{E}_{b;L}\left[\log P(c|b)\right] \tag{15}$$

We have yet to make any assumptions about the distributions of any $P(\cdot)$, except to say that $|c| = L$. To recover the REINFORCE objective of eq. (1), we assume B is a sequence of independent Bernoulli trials. Further, we approximate $P(b, L) \approx P(b)$. Then we factor $P(c, b)$ as [13]:

$$P(c, b) = \prod_{t=1}^{T} P(c_{\ell_t} | b_{\leq t}, c_{<\ell_t})^{b_t} P(b_t | b_{\leq t}, c_{<\ell_t}) \tag{16}$$

where $\ell_t = \sum_{t'=0}^{t} b'_t$.

Under these assumptions, the rightmost expectation in eq. (15) decomposes into[3]

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\mathbb{E}_b\left[\log P(c|b)\right] &= \frac{\partial}{\partial\theta}\mathbb{E}_b\left[\sum_{t=1}^{T} b_t \log P(c_{\ell_t}|b_{\leq t}, c_{<\ell_t})\right] \\
&= \sum_{t=1}^{T}\frac{\partial}{\partial\theta}\mathbb{E}_b\left[R_t\right] \text{ from eq. (2)} \\
&= \sum_{t=1}^{T}\frac{\partial}{\partial\theta}\mathbb{E}_{b_{\leq t}}\left[R_t\right] \text{ since } R_t \text{ not based on } b_{>t} \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{\leq t}}\left[\frac{\partial R_t}{\partial\theta} + R_t\frac{\partial}{\partial\theta}\log P(b_{\leq t}|c_{<\ell_t})\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{\leq t}}\left[\frac{\partial R_t}{\partial\theta} + R_t\sum_{t'\leq t}\frac{\partial}{\partial\theta}\log P(b_{t'}|b_{t'-1}, c_{<\ell_{t'}})\right] \\
&= \mathbb{E}_b\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial\theta} + \left(\sum_{t'\geq t} R_{t'}\right)\frac{\partial}{\partial\theta}\log P(b_t|b_{<t}, c_{<\ell_t})\right)\right]
\end{aligned}
$$

The expectation of the sum of frame-wise objectives is the same as the expectation of the "global" objective, where no subset of B are attributed to a given class label c_ℓ:

$$
\frac{\partial}{\partial\theta}\mathbb{E}_b\left[\log P(c|b)\right] = \mathbb{E}_b\left[\sum_{\ell=1}^{L}\left(\frac{\partial \log P(c_\ell|b)}{\partial\theta} + \log P(c_\ell|b)\frac{\partial}{\partial\theta}\log P(b)\right)\right]
$$

However, the frame-wise - or "local" - signal is assumed to be less noisy [16].

We can see the two issues with the above REINFORCE objective discussed in section 1 by observing eq. (16). First, c_{ℓ_t} is undefined when ℓ_t exceeds L. This is only a problem because we ignored the L term in the expectation $\mathbb{E}_{b;L}[\cdot]$. Given the restrictions that $\sum_{t=1}^{T} B_t = L$ and that $P(b, L) = P(b)$, we can define a distribution over the restriction of samples to those of length L as

$$
P_L(b) = \frac{P(b)I\left[\sum_{t=1}^{T} b_t = L\right]}{\sum_{b'} P(b')I\left[\sum_{t=1}^{T} b'_t = L\right]} \tag{17}
$$

Where $I[\cdot]$ is the indicator function. Since $P_L(b)$ differs from $P(b, L)$ only by a normalization constant, we can equivalently sample from $P_L(b)$ or $P(b, L)$. Were $B_t \perp\!\!\!\perp B_{t'}$, $P_L(b) = P(b|L)$, the CB distribution, and we could sample from that. This is not true since the model is autoregressive. We could brute force $P_L(b)$ by calculating $P_L(b)$ over all B such that $\sum_t B_t = L$, or draw an

[3] Thanks to Dieterich Lawson for this derivation.

unaltered sample B from $P(b)$ and reject it whenever $\sum_t B_t \neq L$. The former is infeasible. The latter can be feasible when $\sum_{t=1}^T p_t = L$, which we would expect for a model near convergence, but not at initialization.

Second, $P(c, b)$ is maximized whenever $B_t = 0$ for all t. The second problem (a tendency to emit early) may be solved by skipping the factorization of class label probabilities and using the aforementioned global objective. However, in this case, L is still ignored and the first problem is still a problem. Furthermore, we would lose the ability to attribute credit to the t-th frame for classifying label c_{ℓ_t}.

The easiest fix is to remove the auto-regressive property over Bernoulli trials and treat each as independent: $P(b) = \prod_{t=1}^T P(b_t)$. This allows us to sample B from $P_L(b) = P(b|L)$. Further, since each trial is independent, the credit for c_{ℓ_t} belongs exclusively to the first emitting B_t for that label (i.e. when $\ell_t \neq \ell_{t-1}$). The the REINFORCE objective becomes

$$\frac{\partial R}{\partial \theta} = \mathbb{E}_{b;L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial \theta} + R_t \frac{\partial}{\partial \theta} \log P(b_t, L)\right)\right] \tag{18}$$

where $R_t = b_t \log P(c_{\ell_t}|b_{\leq t}, c_{<\ell_t})$. This objective is well-defined, tractable, and is not maximized when $\forall t. B_t = 0$. However, we can no longer feed the previous Bernoulli sample into the model for each time step, nor will w_t be updated unless $R_t \neq 0$. We expect both issues to slow down training.

The former requirement is a by-product of sampling from $P(b, L) = P(b)I[\sum_t b_t = L]$. Since the only drawn samples b have L emissions, we either accept the possibility that the entire sample could be rejected or somehow incorporate L into $P(b)$, requiring knowledge of future events. That being said, even though w_t cannot condition on prior samples or class labels, $R_t = P(c_{\ell_t}|\ldots)$ can. The model can still be auto-regressive, as long as that auto-regression does not impact the odds of a given Bernoulli sample. We can re-inject the auto-regressive property into the model by treating $P(c_{\ell_t}|\cdots)$ as the output of an auto-regressive decoder whose decision to step forward depends on whether $B_t = 1$. We discuss some additional possibilities for auto-regressive dependencies in section 2.6.

To address the problem of update sparsity, we can factor the $P(L)$ term out of the expectation as:

$$\log P(c) \geq \log P(L) + \mathbb{E}_{b|L}\left[\log P(c|b)\right] \tag{19}$$

Assuming the model is not auto-regressive and $P(b)$ is the product of probabilities of independent Bernoulli trials, $P(L)$ is PB-distributed and $P(b|L)$ is CB-distributed. Factoring $P(L)$ out of the expectation is desirable because a) expectations of the same value will have lower variance:

$$\begin{aligned}
\mathbb{E}_{b;L}[X^2] - (\mathbb{E}_{b;L}[X])^2 &= P(L)\mathbb{E}_{b|L}[X^2] - P(L)^2\mathbb{E}_{b|L}[X]^2 \\
&= P(L)\left(\mathbb{E}_{b|L}[X^2] - P(L)\mathbb{E}_{b|L}[X]^2\right) \\
&\geq P(L)\left(\mathbb{E}_{b|L}[X^2] - \mathbb{E}_{b|L}[X]^2\right)
\end{aligned}$$

and b) it ensures all w receive an error signal to backpropagate, regardless of whether they emit. Letting t_ℓ be the inverse mapping of ℓ_t, we define

$$P(c, b|L) = P(c|b, L)P(b|L) = \left(\prod_{\ell=1}^{L} P(c_\ell|b_{t_{\leq \ell}}, c_{<\ell})\right) P(b|L) \tag{20}$$

and plug the conditional probability $P(c|b, L)$ into the expectation in eq. (15) to get the "global" CB REINFORCE gradient:

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_\ell}{\partial \theta} + R_\ell \frac{\partial}{\partial \theta}\log P(b|L)\right)\right] \tag{21}$$

Where $R_\ell = \log P(c_\ell|b_{t_{\leq \ell}}, c_{<\ell})$.

Since the set of all samples $B \sim P(b|L)$ will have exactly L highs $\sum_t B_t = L$, the decomposition of the class label sequence probability is well-defined. The pathological case where reward is maximized when B_t is no longer a problem because we have switched to a global reward rather than a per-frame reward. $P(L)$ can be optimized without estimation, and $P(b|L)$ involves all the odds w rather than just those that are high, like in eq. (18). Unfortunately, eq. (21) may result in a higher-variance estimate than eq. (18) because the former uses a global signal rather than a local[4].

In Section 2.1, we mentioned that an expectation over a DB variable will yield the same expected value as the same expectation over a CB variable assuming that the value function in the expectation is not conditioned on the order in which samples are drafted. The total reward in eq. (21) satisfies this criterion. Thus the global DB REINFORCE objective maximizes the same expectation as the CB REINFORCE objective:

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{A_L|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_\ell}{\partial \theta} + R_\ell \frac{\partial}{\partial \theta}\log P(A_L|L)\right)\right] \tag{22}$$

The advantage of the DB REINFORCE objective over the CB REINFORCE objective is it can leverage the relaxation of the DB, discussed in section 2.3.

Our first frame-wise objective is courtesy of the IDB decomposition of the CB from eq. (11). Though a given trial sample B_{t_j} is conditioned on non-past weights $w_{t_{\geq j}}$, it is only conditioned on samples from the past $b_{t_{<j}}$. Setting $t_j = j$, we decompose the joint probability of the class label sequence and the CB sample as

$$P(c, b|L) = P(c|b, L)P(B|L) = \prod_{t=1}^{T} P(c_{\ell_t}|b_{\leq t}, c_{<\ell_t})^{b_t} P(b_t|L - r_t) \tag{23}$$

Equation (23) is very similar to eq. (16), except the conditioning on the number of class labels L forces ℓ_t to be well-defined whenever $B_t = 1$. The

[4] Though the expected values are the same, the values need not be. Hence they can have different variances.

derivation of the IDB REINFORCE gradient is almost identical to that for eq. (1), yielding

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t' \geq t} R_{t'}\right)\frac{\partial}{\partial \theta}\log P(b_t|L - r_t)\right)\right] \quad (24)$$

where $R_t = b_t \log P(c_{\ell_t}|b_{\leq t}, c_{\ell_t})$.

The IDB REINFORCE gradient solves the problem of ill-defined ℓ_t, provides a frame-wise gradient update, and avoids the pathological case of maximum probability when $\forall t. B_t = 0$. However, were we to use eqs. (23) and (24) on their own, the model would likely still learn to emit early so as to minimize future accumulated (negative) rewards.

To mitigate this tendency, we can leverage the fact that the IDB factors the CB in a fixed but arbitrary order of trials $\{t_j\}_j$. Denoting eq. (23) as the forward IDB joint probability, we define the backward IDB joint probability as

$$P(c, b|L) = \prod_{t=1}^{T} P(c_{L-\ell_t+1}|b_{>t}, c_{>L-\ell_t+1})^{b_t} P(b_t|L - r_{T-t+1}) \quad (25)$$

where we use the mapping $t_j = T - j + 1$. We define the backward IDB reinforce gradient analogously. To perform the backward gradient updates, we need merely to reverse the Bernoulli sequence of weights $w_t \mapsto w_{T-t+1}$ and classes $c_\ell \mapsto c_{L-\ell+1}$ and perform the forward update. We have switched the direction of the dependency on labels from $c_{<\ell}$ to $c_{>L-\ell_t+1}$, likewise for $b_{<t}$ and $b_{>t}$. While optimizing only one direction allows for sequential architectures of $P(c_t|\ldots)$, such as auto-regressive decoders, optimizing both directions either requires distributions which ignore history, e.g. pointwise $P(c_{\ell_t}|b_{<t}, c_{<\ell_t}) = P_t(c_{\ell_t})$.

We hypothesize that, though the forward and backward objectives tend to emit at the beginning and end of the utterance, respectively, alternating between them will cancel out the tendencies. By alternating directions, on average, every weight w_t should receive a signal from $L/2$ nonzero rewards R_t.

While alternating between eqs. (23) and (25) should mitigate or solve the problem of emitting too early or late, doing so precludes defining sequential $P(c_{\ell_t}|\ldots)$. The IDB REINFORCE objective samples T random variables despite having a nonzero reward for only L of them. Since the probability of the t-th Bernoulli trial being high is dependent on all previous Bernoulli samples, the reward associated with sample t is confounded with the results of all prior trials and thus ends up assigned to every one of them. Because the : it cannot distinguish between the high Bernoulli trial that "caused" the reward expectation for class label at time t requires conditioning on all Bernoulli samples $b_{\leq t}$ so that ℓ_t may be determined. In order to get there, of which exact Bernoulli trials were high up to time t is superfluous: all we need to know to determine a distribution over the ℓ-th class label is the time t at which the ℓ-th high Bernoulli (ordered by time t) occurred. This is the t_ℓ term in eq. (20). Decomposing

$P(b|L)$ in eq. (20) in terms of t_ℓ, we get

$$\begin{aligned} P(c,b|L) &= \prod_{\ell=1}^{L} P(c_\ell|t_{\leq\ell}, c_{<\ell}) P(t_\ell|t_{<\ell}, L) \\ &= \prod_{\ell=1}^{L} P(c_\ell|t_{\leq\ell}, c_{<\ell}) P(t_\ell|t_{\ell-1}, L-\ell) \end{aligned} \tag{26}$$

The first line is a simple application of the chain rule of probabilities. $t_\ell \perp\!\!\!\perp t_{\ell-2}|t_{\ell-1}$ follows from the fact that $t_\ell > t_{\ell-1} > t_{\ell-2}$, so $t_{\ell-1}+1$ will be the minimum value that t_ℓ can take.

The $P(t_\ell|t_{\ell-1}, L-\ell)$ term is identical in definition to the step function of the BB from eq. (12): t_ℓ must be drafted from $t \in [t_{\ell-1}+1, T]$ subject to the condition that $T-\ell-1$ more samples must be drafted after t_ℓ. The joint probability of $t_{\leq\ell}$ is the BB, matching the CB when $\ell = L$.

Deriving the BB REINFORCE gradient is similar to before. Letting $R_\ell = \log P(c_\ell|t_{\leq\ell}, c_{<\ell})$,

$$\begin{aligned} \frac{\partial R}{\partial\theta} &= \frac{\partial \log P(L)}{\partial\theta} + \sum_{\ell=1}^{L} \frac{\partial}{\partial\theta} \mathbb{E}_b[R_\ell] \\ &= \frac{\partial \log P(L)}{\partial\theta} + \sum_{\ell=1}^{L} \frac{\partial}{\partial\theta} \mathbb{E}_{\{t_1,t_2,\ldots t_\ell\}|L-\ell}[R_\ell] \\ &\ldots \\ &= \frac{\partial \log P(L)}{\partial\theta} + \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_\ell}{\partial\theta} + \left(\sum_{\ell'\geq\ell} R_{\ell'}\right)\frac{\partial}{\partial\theta}\log P(t_\ell|t_{\ell-1}, L-\ell)\right)\right] \end{aligned} \tag{27}$$

Equation (27), like eqs. (1) and (24), must accumulate future rewards due to t_ℓ's dependency on past drafted samples $t_{<\ell}$. A critical difference of eq. (27), however, is that it does not assign a reward to a non-event - a low Bernoulli trial - directly. This will avoid the tendency to emit early in order to maximize future non-events. The $P(L)$ term manages the parameters that would otherwise not receive an error signal.

The sum of future rewards reflects the uncertainty of the impact of the current decision on future ones. We expect earlier t_ℓ to receive a higher variance, higher magnitude signal than later ones because of this sum. We can spread out this variance in the BB REINFORCE objective by alternating between forward and backward decompositions of $P(c,b|L)$, just like we did for the IDB REINFORCE objective.

2.3 Continuous relaxations

A continuous relaxation is a continuous random variable that approximates (relaxes) some discrete random variable. Of particular note is the Concrete/Gumbel-

Softmax distribution [15, 12], which approximates a categorical random variable $B \in [1, N]$ with odds $\{w_n\}_{n \in [1,N]}$, Gumbel noise $G_n = -\log(-\log U_n), U_n \sim Uniform(0, 1)$, and a scalar temperature $\lambda \in \mathbb{R}^+$. The Concrete random variable $Z \in \{x \in [0, 1]^N; \sum_n x_n = 1\}$ is defined as

$$Z_n = \frac{\exp((\log w_n + G_n)/\lambda)}{\sum_{n'=1}^{N} \exp((\log w_{n'} + G_{n'})/\lambda)} \tag{28}$$

A categorical sample $B \sim P(n; N)$ can be recovered from a Concrete sample in two equivalent manners. First, by the Gumbel-Max trick [24]:

$$P(\forall n'.Z_n \geq Z_{n'}) = \frac{w_n}{\sum_{n'=1}^{N} w_{n'}} = P(B = n) \tag{29}$$

Which implies that $B = H(Z) = \arg_n \max(Z_n)$ is a Categorical sample. Alternatively, Z approaches a one-hot representation of B as $\lambda \to 0$:

$$P(\lim_{\lambda \to 0} Z_n = 1) = \frac{w_n}{\sum_{n'=1}^{N} w_{n'}} = P(B = n) \tag{30}$$

When $N = 2$, $P(B = n)$ is Bernoulli, the Concrete variable is defined as

$$Z = \frac{1}{1 + \exp(-(\log w + D)/\lambda)}, D = \log U - \log(1 - U) \tag{31}$$

and the deterministic mapping $B = H(Z) = 1_{Z>0.5}$.

Using the mapping $\prod_{a \in A} w_a = w'$, the CB can be considered a categorical distribution and suitable for a Concrete relaxation. Unfortunately, using this mapping directly would convert an N-length vector of weights w_n to a vector of N-choose-k weights, which is intractable for large N. The numerator in eq. (28) cannot be teased into a combination of random variables $W_1(w_1), W_2(w_2), \ldots$, because the Gumbel noise G_n, which would now represent the combination of noise of the W_a terms, would no longer be independent of $G_{n'}, n' \neq n$. Thus, the CB is not directly suited to continuous relaxation.

However, reframing the CB in terms of intermediate variables defined in the DB and IDB recursive definitions will allow us a tractable number of independent, non-identical random variables to relax. Specifically, we will relax the recursive step distributions in eqs. (8) and (10). The drafting procedure in eq. (8) can be reframed as a categorical distribution over $T - j$, where $P(i \in A_j | \ldots) \mapsto w'_i$. We can use the Concrete distribution to relax the draft and repeat the relaxation L times, with a combined representation of size $\sum_{\ell=1}^{L} T - \ell + 1$ values. Similarly, the choice of including t_j in the sample $A_{r_j,j}$ from eq. (10) is a binary decision and can be reframed as a Bernoulli distribution where $P(t_j \in A_{r_j,j} | \ldots) \mapsto p_{t_j}$. Again, the Concrete distribution can be used to relax each of the T decisions, with a combined representation of size T. The BB relaxation is similar to the draft relaxation, but involves $\sum_{\ell=1}^{L} T - t_{\ell-1} \leq \sum_{\ell=1}^{L} T - \ell + 1$ values.

While the DB, IDB, and BB relaxations are continuous within a single step, discontinuities will arise between steps. This is because the probabilities of the next step are conditioned on a discrete decision $H(Z)$ made in the previous step. Better continuous relaxations that smooth the decision function across steps may exist, and are left for future work.

When the objective can be reframed in terms of the relaxation Z, a network can start by optimizing a high temperature λ, then slowly lower it over the course of training so that Z approaches the discrete distribution. At test time, the deterministic mapping $H(Z)$ can be used. For our objective, a relaxed emission does not make sense. We need to come up with L distinct distributions for each of the class labels c_ℓ.

We focus on two uses of continuous relaxations with a discrete objective. The first is to use a RELAX-based gradient estimator [9]. RELAX-based gradient estimators augment the REINFORCE estimator with some additional terms that are intendent to reduce its variance. Letting B be a discrete random variable of a continuous relaxation Z, the gradient of the expected value of some f (where f can be a reward, e.g.) is defined as

$$\frac{\partial \mathbb{E}_b[f(b)]}{\partial \theta} = \mathbb{E}_b\left[\big(f(b) - \mathbb{E}_{z|b}[c(z)]\big)\frac{\partial \log P(b)}{\partial \theta} - \frac{\partial \mathbb{E}_{z|b}[c(z)]}{\partial \theta}\right] + \frac{\partial \mathbb{E}_z[c(z)]}{\partial \theta} \quad (32)$$

Where $c(z)$ is a control variate, e.g. a neural network trained on the values of the relaxation to minimize the difference between the objective $f(b)$ and itself. $P(z|b)$ is the truncated distribution over Z such that the value of Z obeys the relationship $H(Z) = b$. If $c(z)$ is the concrete distribution parameterized by a learnable λ, eq. (32) is the REBAR gradient [19].

The second is the so-called Straight-Through (ST) estimator [3, 12]. An ST estimator uses the discrete sample $H(X)$ during the forward pass, and estimates the partial derivative of $H(X)$ in the backward pass with that of X, i.e. $\frac{\partial H(X)}{\partial \theta} \approx \frac{\partial X}{\partial \theta}$. This estimator is biased, but can work well in practice. If we output a one-hot representation $H(X^{(\ell)}) = b^{(\ell)} \in \{0,1\}^T, b_t^{(\ell)} = 1_{t=n^{(\ell)}}$ for the ℓ-th drafted (DB or BB) sample, adding them together $b = \sum_{\ell=1}^{L} b^{(\ell)}$ produces our CB sample. If we substitute $\frac{\partial b_t^{(\ell)}}{\partial \theta} \approx \frac{\partial X_t^{(\ell)}}{\partial \theta}$ then $\frac{\partial b_t}{\partial \theta} = \sum_\ell \frac{\partial b_t^{(\ell)}}{\partial \theta}$ is well-defined. Alternatively, we can construct b by concatenating together the relaxed Bernoulli trials of the IDB, $b = [b^{(1)}, b^{(2)}, \ldots, b^{(T)}], b^{(t)} = H(X^{(t)})$. Again, the partial derivatives are well-defined: $\frac{\partial b_t}{\partial \theta} = \frac{\partial b^{(t)}}{\partial \theta}$. From there, we maximize the likelihood of the data using the conditional distribution derived from eq. (16):

$$P(c|b, L) = \prod_{t=1}^{T} P(c_{\ell_t}|h_t, b_{\leq t})^{b_t} \quad (33)$$

where h_t is a hidden state of the network at timestep t. Conditioning on $b_{\leq t}$ is implicit in the definition of c_{ℓ_t}, though this conditioning is ignored by the ST estimator.

By taking the log-probability, only the timesteps where $B_t = 1$ will have nonzero loss. Were the relaxation a series of independent Bernoulli trials, only the odds w_t of those trials would be updated in backpropagation. Since eqs. (8), (10) and (12) involve all "future" odds, the CB-based ST estimators will update more weights with the likelihood of the data. The DB has a clear advantage in this regard: drafting involves $T - j + 1$ odds for the j-th draft, whereas the BB involves $T - t_j$, and the IDB involves only $T - t + 1$ odds. Since the ST estimators already ignore conditioning on past labels, we can ensure that, on average, the each weight receives $L/2$ updates from the data by shuffling the order in which timesteps are processed t_j. In any case, optimizing the PB term $\frac{\partial P(L)}{\partial \theta}$ will give a blanket update to all weights.

2.4 Exact expectations

Sample estimates of the expectations in section 2.2 should only be used when the value in the expectation is not differentiable or it is infeasible to marginalize out the sample variable. In the case of the model proposed by [14], the distribution $\log P(c_t|\ldots)$ is calculated by a simple linear transformation of the RNN hidden state h_t followed by a softmax. These calculations can be parallelized across t. Further, they are fully differentiable. Therefore if the latent variable B can be efficiently marginalized and that process is differentiable, we can use $P(c)$ to directly maximize the data log likelihood. To show this is possible, we will use the BB distribution:

$$\begin{aligned}
P(c) &= P(L) \sum_b P(b|L) P(c|b) \\
&= P(L) \sum_b P(b|L) \prod_{\ell=1}^{L} P(c_\ell | b, c_{<\ell}) \\
&= P(L) \sum_{\{t_1, t_2, \ldots, t_\ell\}} P(t_1, t_2, \ldots, t_\ell | L) \prod_{\ell=1}^{L} P(c_\ell | t_\ell, c_{<\ell}) \\
&= P(L) \sum_{\ell=1}^{L} \sum_{t_\ell = t_{\ell-1}+1}^{T-L+\ell} P(t_\ell | t_{\ell-1}, L - \ell) P(c_\ell | t_\ell, c_{<\ell}) \\
&= P(L) \sum_{\ell=1}^{L} \sum_{t_\ell=1}^{T} P(t_\ell | t_{\ell-1}, L - \ell) P(c_\ell | t_\ell, c_{<\ell}) \qquad (34)
\end{aligned}$$

where the last line follows assuming $P(t_\ell | t_{\ell-1}, L - \ell) = 0$ when $t_\ell \leq t_{\ell-1}$.

Treating $P(t_\ell | t_{\ell-1}, L)$ as the transition probability between states $t \in [1, T]$ and $P(c_\ell | t_\ell, c_{<\ell})$ as the emission probability, eq. (34) can be considered a Hidden Markov Model. Thus, $P(c)$ can be efficiently calculated using the forward algorithm.

[14] conditioned the current Bernoulli distribution on prior Bernoulli samples and prior class labels. We disallowed conditioning on prior Bernoulli samples when employing the CB, but allow conditioning on prior labels in viz. $P(c_\ell | t_\ell, c_{<\ell})$. If the distribution is derived directly from h_t then, under the CB, $P(c_\ell | t_\ell, c_{<\ell}) = P(c_\ell | t_\ell) = P_t(c_\ell; h_t)$. The

Alternatively, recalling that $P(L)P(b|L) = P(b)$, the independent Bernoulli probabilities, then define a new distribution over augmented class label set $c'_t = c_t \cup \{\_\}$ as

$$P(c'_t) = \begin{cases} P(B_t = 0) & c'_t = \_ \\ P(B_t = 1)P(c_t) & \text{otherwise} \end{cases} \tag{35}$$

where the label "_" acts as a stand-in for choosing not to emit at a given time step. Letting $\beta(c')$ remove all the "_" labels from the augmented label set,

$$\begin{aligned} P(c) &= \sum_b \sum_{t=1}^{T} P(b_t)P(c_{\ell_t})^{b_t} \\ &= \sum_{\{c' : |c| = T \wedge \beta(c') = c\}} \sum_{t=1}^{T} P(c'_t) \end{aligned} \tag{36}$$

This expression of the data likelihood is almost identical to that of CTC [11], with two restrictions. First, it assumes the distribution over labels factors as described in eq. (35). In general, eq. (35) will lead to different gradient updates than directly parameterizing the augmented vocabulary $P(c'_t)$ since the blank label has its own parameterization. Second, $\beta(c')$ in eq. (36) does not reduce repeated labels in c'[5]. Assuming it allows for the non-standard adjustment to β, the data likelihood marginalized over latent Bernoulli sequences can be trivially implemented using an existing CTC loss function.

2.5 Inference

2.6 Fake it until you make it

In section 2.2, we discussed how it is infeasible to sample B in an iterative, auto-regressive fashion, i.e. sample B_t conditioned on $B_{<t}$ given length restriction L. While this is still true, it is often the case that, at test time, "samples" are the result of some deterministic process. For example, the decision to emit at time step t occurs whenever $w_t > 1$. If we replace stochastic sample $B_t \sim P(b_t|b_{<t}, x)$ as the input to our neural network with the test-time deterministic function $\tilde{B}_t = f(\tilde{B}_{<t}, x)$ *during* training, then it is still possible to build an auto-regressive model. However, there is no guarantee that the deterministic samples $\tilde{B}$ will be the same as the sampled ones B. At the beginning of training, $\sum_t \tilde{B}_t \approx T/2 \gg \sum_t B_t$, meaning the model will (hopefully) learn to ignore $\tilde{B}$ early on. As the model converges and the distribution over B becomes more sparse, $\tilde{B} \to B$ and the model can start to rely on $\tilde{B}$. We can also imitate

[5] To the best of our knowledge, there has been no attempt to explore whether the reduction operation leads to any performance benefits over just using the blank label. Graves [10] mention that reducing repeated labels existed prior to the blank label in the formation of the CTC objective.

conditioning B_t on $c_{<\ell_t}$ by constructing $\tilde{\ell}_t$ from $\tilde{B}_{\leq t}$ and filling $c_{\tilde{\ell}_t}$ for $\tilde{\ell}_t > L$ arbitrarily[6]. Doing so will not bias the expectation over B.

3 Experiments

3.1 Toy problem

Check convergence and variance of different estimators.

Choose a fixed-size sequence T and vocabulary size C. Define the population distribution via T binary random variables $\widehat{B}_t \sim P_t(\widehat{b}_t)$ and T categorical random variables $C_t \sim P_t(c_t|c_j)$. Draw a sequence of categorical random variables C by first sampling $\widehat{B}$ and then adding $C_t \sim P(c_t|c_{\ell_t-1})$ to the sequence whenever $\widehat{B}_t = 1$. The resulting sequences c of size L and b of size T was sampled with probability

$$P(c, \widehat{b}) = \prod_{t=1}^{T} P_t(c_{\ell_t}|c_{\ell_t-1})^{\widehat{b}_t} P_t(\widehat{b}_t)$$

The goal is to make $Q_t(b_t) \to P_t(b_t)$ and $Q_t(c_t|c_j) \to P_t(c_t|c_j)$ for all $t \in [1, T]$ using c and one of the estimators in section 2.2 or the exact expectation from section 2.4. Letting Q_t and P_t belong to the same parameterized family of statistical models (i.e. Bernoulli or categorical), we can determine the distance between the distributions via mean-squared-error over parameters.

Hyperparameters:

1. Estimators
2. T
3. N (batch size, i.e. number of sequences c)
4. M (Monte-Carlo sample, i.e. number of samples B per c)
5. σ (standard deviation of population parameters)

Should fix the number of trials to something very high. Measure for each sample

1. Sample reward
2. Estimator variance
3. MSE between all P_t and Q_t

[6] Factoring $P(L)$ out of the expectation is critical here to ensure the model is pushed to emit the correct number of labels.

3.2 Gigaword Abstractive Summarization, TIMIT, WSJ

Following Raffel et al. [17], we can also get to one or all of these tasks. The models and training are fairly interchangeable between corpora (GGWS needs an additional embedding layer, TIMIT + WSJ might use some language modelling).

Decoder structures:

1. Pointwise (feed-forward from encoder hidden state). Similar to Luo et al. [14], Lawson et al. [13].
2. Autoregressive decoder with encoder hidden state input. Similar to Raffel et al. [17].
3. Monotonic attention with fixed- or variable-sized windows (former similar to Chiu and Raffel [7])
4. Autoregressive decoder with attention, but context vector is only used to produce emission distribution, similar to Wu et al. [22], Wu and Cotterell [21].

`https://github.com/j-min/MoChA-pytorch`

References

[1] Unequal probability exponential designs. In *Sampling Algorithms*, pages 63–98. Springer New York, New York, NY, 2006. ISBN 978-0-387-34240-5. doi: 10.1007/0-387-34240-0_5.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*, ICLR '15, San Diego, USA, 2015. URL `http://arxiv.org/abs/1409.0473`.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. URL `http://arxiv.org/abs/1308.3432`.

[4] Lennart Bondesson, Imbi Traat, and Anders Lundqvist. Pareto Sampling versus Sampford and Conditional Poisson Sampling. *Scandinavian Journal of Statistics*, 33(4):699–720, December 2006. ISSN 0303-6898. doi: 10.1111/j.1467-9469.2006.00497.x.

[5] Sean X. Chen and Jun S. Liu. Statistical applications of the Poisson-Binomial and Conditional Bernoulli distributions. *Statistica Sinica*, 7(4): 875–892, 1997. ISSN 10170405, 19968507. URL `www.jstor.org/stable/24306160`.

[6] Xiang-Hui Chen, Arthur P. Dempster, and Jun S. Liu. Weighted finite population sampling to maximize entropy. *Biometrika*, 81(3):457–469, 1994. ISSN 00063444. doi: 10.2307/2337119.

[7] Chung-Cheng Chiu and Colin Raffel. Monotonic Chunkwise Attention. In *6th International Conference on Learning Representations*, ICLR '18, Vancouver, Canada, 2018. URL `https://openreview.net/forum?id=Hko85plCW`.

[8] Artyom Gadetsky, Kirill Struminsky, Christopher Robinson, Novi Quadrianto, and Dmitry Vetrov. Low-variance black-box gradient estimates for the Plackett-Luce distribution. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, AAAI '20, 2020.

[9] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *6th International Conference on Learning Representations*, ICLR '18, Vancouver, Canada, 2018. URL `https://openreview.net/forum?id=SyzKd1bCW`.

[10] Alex Graves. Connectionist Temporal Classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 61–93. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-24797-2. doi: 10.1007/978-3-642-24797-2_7.

[11] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143891.

[12] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations*, ICLR '17, Toloun, France, 2017. URL `https://openreview.net/forum?id=rkE3y85ee`.

[13] Dieterich Lawson, Chung-Cheng Chiu, George Tucker, Colin Raffel, Kevin Swersky, and Navdeep Jaitly. Learning hard alignments with variational inference. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '18, pages 5799–5803, April 2018. ISBN 2379-190X. doi: 10.1109/ICASSP.2018.8461977.

[14] Yuo Luo, Chung-Cheng Chiu, Navdeep Jaitly, and Ilya Sutskever. Learning online alignments with continuous rewards policy gradient. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '17, pages 2801–2805, March 2017. doi: 10.1109/ICASSP.2017.7952667.

[15] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A continuous relaxation of discrete random variables. In *5th*

International Conference on Learning Representations, ICLR '17, Toloun, France, 2017. URL `https://openreview.net/forum?id=S1jE5L5gl`.

[16] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of the 31st International Conference on Machine Learning*, ICML '14, pages 1791–1799. PMLR, January 2014. URL `http://proceedings.mlr.press/v32/mnih14.html`.

[17] Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 2837–2846, Sydney, Australia, August 2017. PMLR. URL `http://proceedings.mlr.press/v70/raffel17a.html`.

[18] Kevin Swersky, Brendan J Frey, Daniel Tarlow, Richard S. Zemel, and Ryan P Adams. Probabilistic n-choose-k models for classification and ranking. In *Advances in Neural Information Processing Systems*, number 25 in NIPS, pages 3050–3058. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4702-probabilistic-n-choose-k-models-for-classification-and-ranking.pdf`.

[19] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems 30*, pages 2627–2636. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6856-rebar-low-variance-unbiased-gradient-estimates-for-discrete-latent-variable-models`
`pdf`.

[20] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696.

[21] Shijie Wu and Ryan Cotterell. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 1530–1537, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1148.

[22] Shijie Wu, Pamela Shapiro, and Ryan Cotterell. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 4425–4438, Brussels, Belgium, 10. Association for Computational Linguistics. doi: 10.18653/v1/D18-1473.

[23] Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, IJCAI '19, pages 3919–3925. International Joint Conferences on Artificial Intelligence Organization, 2019. doi: 10.24963/ijcai.2019/544.

[24] John I. Yellott. The relationship between Luce's Choice Axiom, Thurstone's Theory of Comparative Judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, April 1977. ISSN 0022-2496. doi: 10.1016/0022-2496(77)90026-8.