The Conditional Bernoulli and its Application to Speech Recognition

Sean Robertson

September 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 $\{y_\ell\}_{\ell \in [1,L]}$, where $L \ll T$. In hybrid architectures, y_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) [13] or sequence-to-sequence (seq2seq) architectures [2]. The former introduces a special blank label; performs a one-to-many mapping $y_\ell \mapsto \tilde{y}_t^{(i)}$ 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 y_ℓ.

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 $Y_t \sim P(y_t | x_{\leq t}, b_{<t})$. 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 Y_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.

b_1	b_2	b_3	Bias weight
~~0~~	~~0~~	~~0~~	
0	0	1	$2\times$
0	1	0	$2\times$
~~0~~	~~1~~	~~1~~	
1	0	0	$4\times$
~~1~~	~~0~~	~~0~~	
~~1~~	~~0~~	~~1~~	
~~1~~	~~1~~	~~1~~	

	Average $\sum R_t$
Unbiased	$R_1/3 + R_2/3 + R_3/3$
Biased	$R_1/2 + R_2/4 + R_3/4$

Figure 1: Example of the effect of sample bias on total reward under a uniform prior.

the authors rely on an estimate of the REINFORCE gradient [23]:

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

where

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

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 [16], 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) [16].

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 a bias in eq. (2) is responsible for the early training difficulty.

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. For example, if at point $t = T - L + \ell$ only ℓ samples emitted, $B_t = 1$ regardless of $P(b_t)$. Likewise, if L samples emitted before t, $B_t = 0$.

Though this bias appears harmless at first, it has great ramifications for the estimator early on during training. In short, the bias leads to earlier samples

[2] Forgoing the motivation for online speech recognition.

having greater impact on the total reward than later ones. Figure 1 provides an illustrative example for $T = 3$ and $L = 1$. At the beginning of training, $P(b_t|\ldots) \approx 0.5$ and each b is assumed to be equally likely. Though there are $2^3 = 8$ such equally-probable b, only $\binom{T}{L} = 3$ feature only one high Bernoulli value (i.e. $\sum b_t = L$). They are still equally probable, so an unbiased estimator should weigh the possibilities equally, leading to a total expected reward distributed evenly among the pointwise rewards. However, under the biased sampling procedure, the sequence $b = (1, 0, 0)$ will appear twice as often as the other two valid sequences, meaning R_1 has twice the impact on the total reward versus R_2 or R_3.

The bias also has a strong impact on the gradient estimates. We can see from eq. (1) that $R_{\geq t} = 0$ once $\sum b_{<t} = L$. Thus, parameter θ will receive no update from such t. For $T \gg L$, we expect a model to be done emitting high Bernoulli trials after about $2L$ frames, which would mean the tail $T - 2L$ frames would have no impact on the gradient. This could explain why, without an additional "entropy penalty," the model would learn to emit entirely at the beginning of the utterance [17].

To solve the problem of bias, 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 training. It avoids placing undue emphasis on earlier trials, which should curtail the convergence problems faced by Luo et al. [17]. In addition, the CB can be decomposed into Bernoulli trials that condition on past trial results, similar to eq. (2). We also show how the CB can be relaxed to a continuous variable for use in Straight-Through estimators [3, 14] or RELAX-like estimators [18, 10]. Finally, we outline under which conditions the likelihood of y can be exactly and efficiently calculated under the assumptions of the CB, and how it relates to CTC and RNN Transducers [12].

2 The Conditional Bernoulli

2.1 Definitions

The Conditional Bernoulli distribution [7, 6], 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 [6] 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. $B_t = 1 \iff t \in A$. Then the CB can be restated as

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

where

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

normalizes over all possible k-tuples of w_i in some set S. Equation (7) can be considered a generalization of the binomial coefficient, which 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 [6] 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 $A_j^c = I \setminus A_j = \{t_{j+1}, t_{j+2}, \ldots, t_T\}$. Then the probability that some $t \in A_{j-1}^c$ is the j-th sample to be drafted into A is defined as

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

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

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

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 [6] 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

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

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

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

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 $A_{r,j_r} \subseteq A_{j_r} = (t_1, t_2, \ldots t_{j_r})$ to be the C samples of A_{j_r} that have been added to A. Define the probability that the next sample $t_j \in A_{t_{j_{r-1}}}$ is the r-th drafted sample to be

$$P(j = j_r | k - r, j_{r-1}; w) = \frac{w_{t_j} C(k - r, A_{t_j}^c; w)}{C(k - r + 1, A_{t_{j_{r-1}}}^c; w)} \tag{12}$$

The draft is bound to the suffix $A_{t_{j_{r-1}}}^c = (t_{j_{r-1}+1}, t_{j_{r-1}+2}, \ldots, t_{j_T})$. 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 $C(k - r + 1, A_{t_j}^c; w) = w_{t_j+1} C(k - r, A_{t_j+1}^c; w) + C(k - r + 1, A_{t_j+1}^c; w)$, it is easily shown via induction that $C(k - r + 1, A_{t_{j_{r-1}}}^c; w) = \sum_{j=j_{r-1}+1}^{T} w_{t_j} C(k - r, A_{t_j}^c; w)$, proving that eq. (12) is a valid probability distribution. The probability of a draft prefix is calculated as

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

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 C using Method 2 from [6]. Observing eqs. (11) and (13), the probability of a prefix under the IDB matches a probability of some BB draft prefix whenever the last sampled Bernoulli from the IDB was high. Assuming $b_{t_j} = 1$ and $\sum_{j'=1}^{j} b_{t_{j'}} = r$, $b_{t_{\le j}} \mapsto A_{r,j_r}$ by the relation $b_{t'} = 1 \Leftrightarrow t' \in A_{r,j_r}$. The BB allows us to marginalize out prior drafted samples and ask what the probability is that t_j is the r-th drafted sample:

$$
\begin{aligned}
P(j = j_r|k-r;w) &= \sum_{A_{r,j_{r-1}}} P(A_{r,j_r}|k-r;w) \\
&= \left(\sum_{\{j_{<r}:j_{r'}<j\}} \prod_{r'=1}^{r-1} w_{t_{j_{r'}}}\right) \frac{w_{t_j} C(k-r, A^c_{t_j}; w)}{C(k, I; w)} \\
&= \frac{C(r-1, A_{t_{j-1}}; w) w_{t_j} C(k-r, A^c_{t_j}; w)}{C(k, I; w)}
\end{aligned} \tag{14}
$$

The second line features sums over the possible size-$(r-1)$ prefixes that could have been drafted prior to j, which means that each occurs within the subset $A_{t_{j-1}}$. Intuitively, the numerator enumerates all possible prefixes and all possible suffixes around w_{t_j}, subject to the constraint that $r-1$ elements come before and $k-r$ come after.

t_j being the r-th drafted sample and t_j being the $(r+1)$-th drafted sample are clearly disjoint events. Summing over these disjoint probabilities recovers the probability that t_j belongs to A:

$$
\begin{aligned}
\sum_{r=1}^{k} P(j = j_r|k-r;w) &= \frac{w_{t_j}}{C(k, I; w)} \sum_{r=1}^{k} C(r-1, A_{t_{j-1}}; w) C(k-r, A^c_{t_j}; w) \\
&= \frac{w_{t_j} C(k-1, I \setminus \{t_j\})}{C(k, I; w)}
\end{aligned} \tag{15}
$$

where the second line follows from noting $A_{t_{j-1}} \cup A^c_{t_j} = I \setminus \{t_j\}$ and applying

Proposition 1.c. from Chen et al. [7]:

$$\forall S \subseteq I \quad C(k, I; w) = \sum_{r=0}^{k} C(r, S; w) C(k - r, I \setminus S; w) \tag{16}$$

a generalization of Vandermonde's identity.

Outside of statistics, Swersky et al. [21] 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 [26] 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 [27], which has also been explored in the realm of gradient estimation [9]. 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 $y = \{y_\ell\}_{\ell \in [1,L]}$ 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.

$$\begin{aligned} P(y) &= P(y, L) \\ &= \sum_b P(y, b, L) \\ &= \sum_b P(b, L) P(y|b) \\ &= P(L) \sum_{b|L} P(b|L) P(y|b) \\ &= P(L) \mathbb{E}_{b|L} \left[P(y|b) \right] \end{aligned} \tag{17}$$

Where $P(y) = P(y, L)$ follows from the fact that L is a deterministic function of y.

Taking the log, we get

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

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

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \frac{\partial}{\partial \theta}\mathbb{E}_{b|L}\left[\log P(y|b)\right] \tag{18}$$

We have yet to make any assumptions about the distributions of any $P(\cdot)$, except to say that $|y| = L$. 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 $P(y, b)$ as [16]:

$$P(y, b) = \prod_{t=1}^{T} P(y_{\ell_t}|b_{\leq t}, y_{<\ell_t})^{b_t} P(b_t|b_{\leq t}, y_{<\ell_t}) \tag{19}$$

where $\ell_t = \sum_{t'=1}^{t} b'_t$.

Under these assumptions, the rightmost expectation in eq. (18) decomposes into[3]

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

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

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

However, the frame-wise - or "local" - signal is assumed to be less noisy [19].

[3]Thanks to Dieterich Lawson for this derivation.

The decomposition of $P(y, b)$ from eq. (19) is only well-defined when $|y| = \sum_{t=1}^{T} b_t$. This is not a problem during testing but it is during training when $|y|$ is fixed. For that reason, Luo et al. [17] hacks the Bernoulli sequence probabilities using the methods described in section 1. This produces a biased estimator with a variety of problems. If we can condition the joint on the number of highs in y, we can avoid the problem entirely.

The easiest fix to being ill-defined is to remove the auto-regressive property over Bernoulli trials and treat them as independent: $P(b) = \prod_{t=1}^{T} P(b_t)$. In this case, $P(b|L)$ is the CB a global REINFORCE objective can be defined as

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\frac{\partial \sum_{t=1}^{T} R_t}{\partial \theta} + \left(\sum_{t=1}^{T} R_t\right)\frac{\partial}{\partial \theta}\log P(b|L)\right] \tag{20}$$

Equation (20) is tractable and, unlike eq. (1), well-defined. Unfortunately, it is no longer autoregressive nor local.

The requirement that the model is not auto-regressive with respect to sequential B_t is a by-product of sampling from $P(b|L)$. If $P(b|L)$ is a Conditional Bernoulli, the odds of each Bernoulli trial w_t must be known before sampling a prefix. In an auto-regressive model, w_t depends on the prefix of samples. We know of no way to determine all w_t without iterating through all the sequences of Bernoulli trials with L highs, which would be intractable. Some distribution other than the CB could be chosen for $P(b|L)$, but this distribution would need to be able to sample both $P(b_t|b_{<t}, L)$ and $P(b_t|b_{<t})$ (i.e. with or without conditioning on the number of class labels) without conditioning on the odds of future events. To the best of our knowledge, existing research meets one, but not both, requirements.

That being said, even though w_t cannot condition on prior samples or class labels, $R_t = b_t \log P(y_{\ell_t}|\ldots)$ 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 $P(y_{\ell_t}|\cdots)$ 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 2.5. From now on, we assume $P(b_t|b_{<t}) = P(b_t)$.

If we still assume no prior dependence between Bernoulli trials B_t, the expectation is over the CB distribution $P(b|L)$. We can use the various decompositions of the CB defined in section 2.1 to derive local gradient estimates.

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

$$P(y, b|L) = P(y|b, L)P(B|L) = \prod_{t=1}^{T} P(y_{\ell_t}|b_{\leq t}, y_{<\ell_t})^{b_t} P(b_t|L - r_t) \tag{21}$$

Equation (21) is very similar to eq. (19), except the conditioning on the number of class labels L forces ℓ_t to be well-defined whenever $B_t = 1$. The derivation of the IDB REINFORCE gradient is almost identical to that for eq. (1), yielding

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t' \geq t} R_{t'}\right)\frac{\partial}{\partial \theta}\log P(b_t|L - \ell_t)\right)\right] \tag{22}$$

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

Equation (22) is very similar to eq. (1), but is unbiased. In the example given in fig. 1, the IDB estimate will treat each valid sequence of Bernoulli trials as equally likely. The sum of rewards over future trials is a function of the dependence of R_t on past trials $b_{<t}$.

We can prove that the variance of eq. (22) is no greater than that of eq. (20). Representing the expectation in eq. (22) as $\mathbb{E}_{b|L}[Y]$ and that in eq. (20) as $\mathbb{E}_{b|L}[Z]$,

$$\begin{aligned}
\mathbb{E}_{b|L}[Z] &= \mathbb{E}_{b|L}\left[\frac{\partial \sum_{t=1}^{T} R_t}{\partial \theta} + \left(\sum_{t'=1}^{T} R_{t'}\right)\left(\sum_{t=1}^{T}\frac{\partial}{\partial \theta}\log P(b_t|b_{<t}, L)\right)\right] \\
&= \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t'=1}^{T} R_{t'}\right)\frac{\partial}{\partial \theta}\log P(b_t|b_{<t}, L)\right)\right] \\
&= \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t'=t}^{T} R_{t'}\right)\frac{\partial}{\partial \theta}\log P(b_t|b_{<t}, L)\right) + \right. \\
&\qquad \left. \sum_{t=1}^{T}\left(\left(\sum_{t'=1}^{t-1} R_{t'}\right)\frac{\partial}{\partial \theta}\log P(b_t|b_{<t}, L)\right)\right] \\
&= \mathbb{E}_{b|L}[Y + X] \\
&= \mathbb{E}_{b|L}[Y] + \mathbb{E}_{b|L}[X]
\end{aligned} \tag{23}$$

We can see that $Z = X + Y$. We first prove that the expectation of the two estimators are equivalent by showing $\mathbb{E}_{b|L}[X] = 0$:

$$
\begin{aligned}
\mathbb{E}_{b|L}[X] &= \mathbb{E}_{b|L}\left[\sum_{t=1}^{T} R_{<t}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t}, L-\ell_t)\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b|L}\left[R_{<t}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t}, L-\ell_t)\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{<t}|L}\left[\mathbb{E}_{b_t|b_{<t},L}\left[R_{<t}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t}, L-\ell_t)\right]\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{<t}|L}\left[R_{<t}\mathbb{E}_{b_t|b_{<t},L}\left[\frac{\partial}{\partial\theta}\log P(b_t|b_{<t}, L-\ell_t)\right]\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{<t}|L}\left[R_{<t}\frac{\partial}{\partial\theta}\mathbb{E}_{b_t|b_{<t},L}[1]\right] \\
&= 0
\end{aligned}
\tag{24}
$$

Now we can treat the random variable Z as a function of X and Y, $Z = X+Y = J(X,Y)$ and calculate the marginal expectation of Y:

$$
\widehat{J}(Y) = \mathbb{E}_x[J(X,Y)|Y] = Y + \mathbb{E}_x[X] = Y \tag{25}
$$

which follows from eq. (24). By eq. (23), $\mathbb{E}_y[\widehat{J}(Y)] = \mathbb{E}_y[Y]$ is expectation in the IDB estimator of eq. (22). The remainder of the proof is merely an application of the Rao-Blackwell-Kolmogorov theorem:

$$
\begin{aligned}
Var(J(X,Y)) &= \mathbb{E}_{x,y}[J(X,Y)^2] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\
&= \mathbb{E}_y[\mathbb{E}_x[J(X,Y)^2|Y]] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\
&\geq \mathbb{E}_y[\mathbb{E}_x[J(X,Y)|Y]^2] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\
&= \mathbb{E}_y[\widehat{J}(Y)^2] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\
&= \mathbb{E}_y[\widehat{J}(Y)^2] - \mathbb{E}_y[\widehat{J}(Y)]^2 \text{ from eq. (23)} \\
&= Var(\widehat{J}(Y))
\end{aligned}
\tag{26}
$$

where the third line follows from the convexity of $(\cdot)^2$ and Jensen's Inequality.

The IDB REINFORCE gradient can be more efficiently calculated using the BB step function. Letting R_{t_ℓ} denote the reward for timestep t_ℓ whenever

$B_t = 1$, all remaining R_t have reward zero. Thus

$$
\begin{aligned}
\frac{\partial R}{\partial \theta} &= \frac{\partial \log P(L)}{\partial \theta} + \frac{\partial}{\partial \theta} \mathbb{E}_{b|L} \left[\log P(y|b) \right] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \sum_{t=1}^{T} \mathbb{E}_{b_{\leq t}|L} \left[\frac{\partial R_t}{\partial \theta} + R_t \frac{\partial}{\partial \theta} \log P(b_{\leq t}|L - t_\ell) \right] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \mathbb{E}_{b_{\leq t_\ell}|L} \left[\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell} \frac{\partial}{\partial \theta} \log P(b_{\leq t_\ell}|L - \ell) \right] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \mathbb{E}_{t_{\leq \ell}|L} \left[\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell} \frac{\partial}{\partial \theta} \log P(t_{\leq \ell}|L - \ell) \right] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L} \left[\sum_{\ell=1}^{L} \left(\frac{\partial R_{t_\ell}}{\partial \theta} + \left(\sum_{\ell' \geq \ell} R_{t_{\ell'}} \right) \frac{\partial}{\partial \theta} \log P(t_\ell | t_{\ell-1}, L - \ell) \right) \right]
\end{aligned} \tag{27}
$$

where $R_{t_\ell} = \log P(y_\ell | t_{\leq \ell}, y_{<\ell})$. The $P(t_\ell | \ldots)$ term is recognized as the BB step function of eq. (12). Equation (27) yields identical sample estimates as eq. (22), but requires calculation of far fewer terms.

Equations (22) and (27) allow R_t to condition on the history of Bernoulli trials $b_{<t}$ sampled. For example, $\log P(y_\ell | t_{\leq \ell}, y_{<\ell})$ can be parameterized by a decoder neural network which concatenates together a hidden state from an encoder network from time t_ℓ and an embedding of the previous class label $y_{\ell-1}$ as input to the RNN. Unfortunately, the dependence on $t_{\leq \ell}$ means that $t_{\ell'}$ is not only responsible for reward $R_{t_{\ell'}}$, but also for all rewards succeeding it $R_{t_{\ell'+1}}, R_{t_{\ell'+2}}, \ldots$. For this reason, $\frac{\partial}{\partial \theta} \log P(t_\ell | t_{\ell-1}, L - \ell)$ will tend to receive higher magnitude updates than $t_{\ell+1}$, which we expect to increase the variance of the estimator.

We can make the estimator more "local" if we make a conditional independence assumption $P(y_\ell | t_{\leq \ell}, y_{<\ell}) = P(y_\ell | t_\ell, y_{<\ell})$. This costs us, for example, the ability to feed an encoder hidden state as part of the input to a decoder RNN since that produces an implicit dependence on all $t_{\leq \ell}$. Our example decoder may still, however, condition its output on a hidden state at time t_ℓ and previous class labels $y_{<\ell}$, similar to [25].

Deriving the new estimator is similar to before. Letting $R_{t_\ell} = \log P(y_\ell | t_\ell, y_{<\ell})$,

$$
\begin{aligned}
\frac{\partial R}{\partial \theta} &= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \frac{\partial}{\partial \theta} \mathbb{E}_{b|L}[R_{t_\ell}] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \frac{\partial}{\partial \theta} \mathbb{E}_{t_\ell | L - \ell}[R_{t_\ell}] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L} \left[\sum_{\ell=1}^{L} \left(\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell} \frac{\partial}{\partial \theta} \log P(t_\ell | L - \ell) \right) \right]
\end{aligned} \tag{28}
$$

where $P(t_\ell|L-\ell)$ is the marginal probability of the t being the ℓ-th BB-drafted sample, i.e. eq. (14). We call this estimator the Marginal Bounded Bernoulli (MBB) REINFORCE estimator. Equation (14) is similar to eq. (27) but only multiplies the log-probability of the t_ℓ-th draft (not all $t_{\leq \ell}$ drafts) with its local reward R_{t_ℓ}. This should make the magnitude of the update more uniform with respect to the timestep.

When we make the conditional independence assumption above, we can prove that the MBB estimator has variance no greater than that of the IDB estimator. The proof is similar to that showing the IDB estimator has no greater variance than that of the CB estimator. First, we split the expectation of the IDB estimator into two random variables $Z = X + Y$:

$$\begin{aligned}
\mathbb{E}_{b|L}[Z] &= \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_{t_\ell}}{\partial\theta} + R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{\leq\ell}|L-\ell)\right)\right] \\
&= \mathbb{E}_{b|L}\Bigg[\sum_{\ell=1}^{L}\left(\frac{\partial R_{t_\ell}}{\partial\theta} + R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_\ell|L)\right) + \\
&\qquad \sum_{\ell=1}^{L}\left(R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right)\Bigg] \\
&= \mathbb{E}_{b|L}[Y+X] \\
&= \mathbb{E}_{b|L}[Y] + \mathbb{E}_{b|L}[X]
\end{aligned} \tag{29}$$

Where we note that the expectation over Y is that of eq. (28). We are left to prove, once again, that $\mathbb{E}_{b|L}[X] = 0$.

$$\begin{aligned}
\mathbb{E}_{b|L}[X] &= \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right)\right] \\
&= \sum_{\ell=1}^{L}\mathbb{E}_{t_{\leq\ell}|L}\left[R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right] \\
&= \sum_{\ell=1}^{L}\mathbb{E}_{t_\ell|L}\left[\mathbb{E}_{t_{<\ell}|t_\ell, L-\ell}\left[R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right]\right] \\
&= \sum_{\ell=1}^{L}\mathbb{E}_{t_\ell|L}\left[R_{t_\ell}\frac{\partial}{\partial\theta}\mathbb{E}_{t_{<\ell}|t_\ell, L-\ell}[1]\right] \\
&= 0
\end{aligned} \tag{30}$$

The remainder of the proof identically follows from eqs. (25) and (26). We emphasize that this only holds when R_ℓ is memoryless ($R_\ell \perp\!\!\!\perp t_{<\ell}|t_\ell$),. If R_ℓ is not memoryless eq. (28) is not an unbiased estimator of the total reward.

Once we have made the conditional independence assumption required for the MBB, however, we can efficiently calculate the exact expectation using dynamic programming. We will discuss how in section 2.4. The MBB may still be preferred over the exact form if the cost to compute R_ℓ is prohibitively expensive.

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 [18, 14], 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

$$Z_n = \frac{\exp((\log w_n + G_n)/\lambda)}{\sum_{n'=1}^N \exp((\log w_{n'} + G_{n'})/\lambda)} \tag{31}$$

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 [27]:

$$P(\forall n'.Z_n \geq Z_{n'}) = \frac{w_n}{\sum_{n'=1}^N w_{n'}} = P(B = n) \tag{32}$$

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 $\lambda \to 0$:

$$P(\lim_{\lambda\to 0} Z_n = 1) = \frac{w_n}{\sum_{n'=1}^N w_{n'}} = P(B = n) \tag{33}$$

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

$$Z = \frac{1}{1 + \exp(-(\log w + D)/\lambda)}, D = \log U - \log(1 - U) \tag{34}$$

and the deterministic mapping $B = H(Z) = I[Z > 0.5]$.

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. (31) 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.

We can, however, relax the CB indirectly by relaxing the intermediate variables defined in section 2.1. The IDB can be relaxed as a sequence of Bernoulli relaxations of eq. (34) according to the recursive step eq. (10). The BB can be relaxed into a sequence of categorical relaxations of eq. (31) according to the draft eq. (12). Finally, the marginal probability of t_ℓ under the BB (eq. (14)) is just one categorical relaxation per label ℓ.

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 y_ℓ.

We focus on two uses of continuous relaxations with a discrete objective. The first is to use a RELAX-based gradient estimator [10]. 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

$$\frac{\partial \mathbb{E}_b[f(b)]}{\partial \theta} = \mathbb{E}_b\left[\left(f(b) - \mathbb{E}_{z|b}[\gamma(z)]\right)\frac{\partial \log P(b)}{\partial \theta} - \frac{\partial \mathbb{E}_{z|b}[\gamma(z)]}{\partial \theta}\right] + \frac{\partial \mathbb{E}_z[\gamma(z)]}{\partial \theta} \tag{35}$$

Where $\gamma(z)$ 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 $\gamma(z)$ is the concrete distribution parameterized by a learnable λ, eq. (35) is the REBAR gradient [22].

RELAX-style estimators can be paired with eqs. (22) and (28). Equation (22) is preferred over eq. (27) as the latter would involve infinite values in the relaxed categorical draft for $t \leq t_{\ell-1}$. Each Bernoulli in the IDB format has a real relaxation except when $T - t = L - \ell_t$, at which point $\log P(b_{\leq t}|\ldots) = 0$ and hence does not need a baseline. Equation (28) is always real.

The second is the so-called Straight-Through (ST) estimator [3, 14]. 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. (19):

$$P(y|b, L) = \prod_{t=1}^{T} P(y_{\ell_t}|h_t, b_{\leq t})^{b_t} \tag{36}$$

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

2.4 Exact expectations

At the end of section 2.2, we mentioned that we can marginalize out the Bernoulli latent variables efficiently, assuming $P(y_\ell|t_{\leq\ell}, y_{<\ell}) = P(y_\ell|t_\ell, y_{<\ell})$. Further, it must be feasible to calculate that probability for all permutations of t and ℓ. In the case of the model proposed by [17], the distribution $\log P(y_t|t)$ 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 and are fully differentiable. The decoder structure of [25] is also a candidate as the distribution $P(y_\ell|t_\ell, y_{<\ell})$ is a simple two-layer feed-forward neural network on the combination of an encoder and a decoder hidden state.

Starting from eq. (17) and making the conditional independence assumption between t_ℓ and $t_{\ell-1}$, we manipulate $P(y)$ into a form suitable for dynamic programming.

$$
\begin{aligned}
P(y) &= P(L)\sum_b P(b|L)P(y|b) \\
&= P(L)\sum_b P(b|L)\prod_{\ell=1}^{L} P(y_\ell|b, y_{<\ell}) \\
&= P(L)\sum_{\{t_1,t_2,\ldots,t_\ell\}} P(t_1, t_2, \ldots, t_\ell|L)\prod_{\ell=1}^{L} P(y_\ell|t_\ell, y_{<\ell}) \\
&= P(L)\sum_{\ell=1}^{L}\sum_{t_\ell=t_{\ell-1}+1}^{T-L+\ell} P(t_\ell|t_{\ell-1}, L-\ell)P(y_\ell|t_\ell, y_{<\ell}) \\
&= P(L)\sum_{\ell=1}^{L}\sum_{t_\ell=1}^{T} P(t_\ell|t_{\ell-1}, L-\ell)P(y_\ell|t_\ell, y_{<\ell})
\end{aligned}
\tag{37}
$$

where the last line follows as $P(t_\ell|t_{\ell-1}, L-\ell) = 0$ 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 $P(y_\ell|t_\ell, y_{<\ell})$ as the emission probability, eq. (37) can be considered a Hidden Markov Model. Thus, $P(y)$ can be efficiently calculated using the forward algorithm.

Equation (37) is a first-order Markov model with respect to the "states" $[1, T]$. We can easily adapt the equation for higher-order models so that $P(y_\ell|\ldots)$ can depend on some arbitrary fixed-length history of emission points $t_\ell, \ldots, t_{\ell-W+1}$, but the number of states will grow exponentially with the size of the history T^W. For large T, higher-order models become increasingly infeasible.

There exists a relationship between eq. (37) and the CTC objective [13] when the history of class labels $y_{<\ell}$ is conditionally independent of the current class label $P(y_\ell|t_\ell, y_{<\ell}) = P(y_\ell|t_\ell)$. Recalling that $P(L)P(b|L) = P(b)$, the independent Bernoulli probabilities, then define a new distribution over an augmented class label set $\{y'_t\} = \{y_t\} \cup \{\_\}$ as

$$P(y'_t) = \begin{cases} P(B_t = 0) & y'_t = \_ \\ P(B_t = 1)P(y_t|t) & \text{otherwise} \end{cases} \tag{38}$$

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

$$\begin{aligned} P(y) &= \sum_b \sum_{t=1}^{T} P(b_t) P(y_{\ell_t}|t)^{b_t} \\ &= \sum_{\{c' : |c| = T \wedge \beta(y') = c\}} \sum_{t=1}^{T} P(y'_t|t) \end{aligned} \tag{39}$$

This expression of the data likelihood is almost identical to that of CTC [13], with two restrictions. First, it assumes the distribution over labels factors as described in eq. (38). In general, eq. (38) will lead to different gradient updates than directly parameterizing the augmented vocabulary $P(y'_t)$ since the blank label has its own parameterization. Second, $\beta(y')$ in eq. (39) does not reduce repeated labels in c'[4]. 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. The additional dependency on $y_{<\ell}$ can be considered a generalization of the CTC loss function. In fact, eq. (37) is nearly identical to the RNN Transducer generalization of CTC whereby an additional decoder-style structure is responsible for modelling the sequence y in an auto-regressive fashion. The only meaningful difference between an RNN-T loss and the maximum likelihood loss implicit in eq. (37) is that the latter explicitly factors the "blank" label into the decision to emit or not to emit.[5]

Thus, eq. (37) can be considered a generalization of CTC. Finally, the estimators from section 2 can also be used as single- or multi-sample approximations for CTC.

2.5 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

[4]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 [11] mention that reducing repeated labels existed prior to the blank label in the formation of the CTC objective.

[5]Look more into this. Double-check that the loss is indeed identical. I believe it is more efficient to use eq. (37) than the form used in RNN-T. It also has a more efficient best-path form.

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

3 Generalizations and approximations

The CB conditional and PB abandon the auto-regressive property of the distribution over Bernoulli trials. This implies each trial is independent though not necessarily identically distributed.

Suppose we really wish to maintain the auto-regressive property of the trials. In this case, we require $P(b_t|b_{<t})$ to be well-defined and easily calculable, such as the output of a recurrent neural network. The general probability over the number of emissions, $P(L)$, resembles the CB with additional conditional dependence requirements:

$$P(L|b_{<t}) = \sum_{b:\sum_t b_{t'}=L} \left(\prod_{t'=t}^{T} P(b_{t'}|b_{<t'}) \right) \tag{40}$$

where we have allowed conditioning on an arbitrary prefix of Bernoulli trials $b_{<t}$ that do not contribute to the count but can change the conditional probabilities per trial. Making the independence assumption over trials $P(b_t|b_{<t}) = P(b_t) = \frac{w_t}{1+w_t}$ recovers eq. (3).

We can also express the conditional probability mass function of the next Bernoulli trial given the remaining number of highs in terms of the unconditioned probabilities and the distribution over the number of highs:

$$\begin{aligned} P(b_t|b_{<t}, L) &= \frac{P(b_t, L|b_{<t})}{P(L|b_{<t})} \\ &= \frac{P(b_t|b_{<t})P(L - \sum_{t'=1}^{t} b_{t'}|b_{\leq t})}{P(L - \sum_{t'=1}^{t-1} b_{t'}|b_{<t})} \end{aligned} \tag{41}$$

If we consider $P(\ell|\ldots)$ a normalized, conditional version of the count function $C(\ell, \ldots)$, we can see how an independence assumption would make eq. (41) equivalent to the step distribution of the IDB eq. (10).

If $P(b_t|b_{<t})$ is unique for the given prefix $b_{<t}$, direct calculations of eqs. (40) and (41) are infeasible given the sheer number of paths yielding L emissions

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

(T choose L). To work around this, we have to simplify or approximate either $P(b_t|b_{<t})$ or $P(\ell|\ldots)$. The independence assumption applied is an example of simplifying $P(b_t|b_{<t})$.

Assuming we are willing to incur the bias to our estimators by approximating distributions, adapting the estimators from section 2.2 to the approximations is trivial: specifically, replace $P(L)$ and $P(b_t|b_{<t}, L)$ in the estimator with their approximations.

To begin, we start with approximating $P(b_t|b_{<t}, L)$. Because we must sample $B_t \sim P(b_t|b_{<t}, L)$ before we can determine the weights w_{t+1}. This precludes directly

4 Experiments

4.1 Toy problem

Check convergence and variance of different estimators.

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

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

The goal is to make $Q_t(b_t) \to P_t(b_t)$ and $Q_t(y_t|c_j) \to P_t(y_t|c_j)$ for all $t \in [1, T]$ using y 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 y)
4. M (Monte-Carlo sample, i.e. number of samples B per y)
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

4.2 Gigaword Abstractive Summarization, TIMIT, WSJ

Following Raffel et al. [20], 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. [17], Lawson et al. [16].
2. Autoregressive decoder with encoder hidden state input. Similar to Raffel et al. [20].
3. Monotonic attention with fixed- or variable-sized windows (former similar to Chiu and Raffel [8])
4. Autoregressive decoder with attention, but context vector is only used to produce emission distribution, similar to Wu et al. [25], Wu and Cotterell [24].

`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] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *4th International Conference on Learning Representations*, ICLR '16, San Juan, Puerto Rico, 2016. URL `http://arxiv.org/abs/1509.00519`.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Alex Graves. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711, 2012. URL http://arxiv.org/abs/1211.3711.

[13] 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.

[14] 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.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations*, ICLR '14, Banf, Canada, 2014. URL http://arxiv.org/abs/1312.6114.

[16] 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.

[17] 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.

[18] 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`.

[19] 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`.

[20] 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`.

[21] 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`.

[22] 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`.

[23] 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.

[24] 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, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1148.

[25] 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1473.

[26] 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.

[27] 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.

A Determining the bias of forced emissions for i.i.d. Bernoulli

From section 1, we assume $R_t = R$ if $B_t = 1$ else $R_t = 0$. If $B_t \sim P(b_t)$ are drawn i.i.d., then let $P(b_t) = p$. We also assume $0 < T \geq L \geq 0$. We ignore $T = 0$. The expected reward for timestep t given that there are L labels is

$$\begin{aligned}
\mathbb{E}_{b|L}[R_t] &= P(B_t = 1|L)R \\
&= P(B_t = 1, L)R/P(L) \\
&= \frac{R\binom{T-1}{L-1}p^L(1-p)^{T-L}}{\binom{T}{L}p^L(1-p)^{T-L}} \\
&= RL/T
\end{aligned} \tag{42}$$

Determining the expectation of the instantaneous reward R_t when the suffix of the sample is manipulated to be exclusively high or low is considerably more difficult. We will use P^* to denote the distributions over T samples forced to have L highs.

We begin by defining the two "force-points" in the Bernoulli process where it stops sampling like a simple Bernoulli process and forces the suffix to be made of exclusively zeros or ones. Define t_0 as the zero-force point, i.e. the minimal value in $[0, T-1]$ such that all $B_t = 0$ for $t_0 < t < T$. Likewise, define

$t_1 \in [0, T-1]$ to be the one-force point where $B_t = 1$ for $t_1 < t < T$. Note that $B_T = 0$ implies t_0 exists and $B_T = 1$ implies t_1 exists. Thus t_0 and t_1 are mutually exclusive events where one or the other must occur for a given sample. The exception is when $T = 0$, at which point both vacuously exist at $t_0 = t_1 = 0$, but we ignore this exception sine we've specified $T > 0$.

We can rephrase the "minimal" requirement of the force points to make it easier to define their distributions. If t_0 being the minimal non-negative integer such that $\forall t \in (t_0, T] B_t = 0$, either $t_0 = 0$ or $B_{t_0} = 1$ (if $B_{t_0} = 0$, then $t'_0 = t_0 - 1$ satisfies $\forall t \in (t'_0, T]\, B_t = 0$, a contradiction). Likewise, either $t_1 = 1$ or $B_{t_1} = 0$.

By introducing the variables $\ell_t = \sum_{t',1,t} b_t$ as the number of high emissions up to t, we can refine our definition of t_0 and t_1 to refer to the points when the suffix must all be zeros or ones explicitly. More formally, for some fixed-value sample $b \sim P^*(b)$:

$$\exists t_0 \in [0, T-1] \implies \ell_{t_0} = L \wedge (t_0 = 0 \vee b_{t_0} = 1) \tag{43}$$

$$\exists t_1 \in [0, T-1] \implies \ell_{t_1} = L - T + t_1 \wedge (t_1 = 0 \vee b_{t_1} = 0) \tag{44}$$

the conditions on ℓ_t reflect the number of high $b_{t'}$ remaining ($t' > t$). For t_0, no high Bernoulli values can follow, which means we must have previously sampled all L. For t_1, the number of remaining Bernoulli values must equal the number of remaining labels ($L - \ell_{t_0} = T - t_0$).

Letting $P^*(t_0)$ be the probability over samples b that t_0 exists and is equal to the value t_0:

$$P^*(t_0) = \begin{cases} \binom{t_0-1}{L-1} p^L (1-p)^{t_0-L} & t_0 > 0 \\ I[L=0] & t_0 = 0 \end{cases} \tag{45}$$

Likewise

$$P^*(t_1) = \begin{cases} \binom{t_1-1}{T-L-1} p^{t_1-T+L} (1-p)^{T-L} & t_1 > 0 \\ I[T=L] & t_1 = 0 \end{cases} \tag{46}$$

If t_0 exists, all highs must occur before or at t_0. The first equation can be interpreted as the probability of $L-1$ high Bernoulli values before t_0, then multiplying that with the probability that $B_{t_0} = 1$ (p). If t_1 exists, all *lows* must occur before or at t_1. The second can be interpreted as the probability of choosing $T - L - 1$ low Bernoulli values before t_1, then multiplying that with the probability that $B_{t_1} = 0$.

Because t_0 and t_1 are mutually exclusive when $T > 0$ and one must always occur at a given time, $\sum_{t=0}^{T-1} P^*(t_0 = t) + P^*(t_1 = t) = 1$[7], giving us a distribution over the forced suffixes.

Consider the conditional probability of $B_t = 1$ relative to the position of the forced points:

[7] This can be verified by observing that eqs. (45) and (46) are negative binomial distributions: $P^*(t_0)$ with success probability $p' = p$, $r' = L$ successes and $k' = t_0 + L$ failures; and $P^*(t_1)$ with success probability $p' = 1 - p$, $r' = T - L$ successes, and $k' = t_1 + T - L$ failures. The rest proceeds similarly to our calculations for $P^*(B_t = 1 | 0 < t < L)$.

- $P^*(B_t = 1 | t < t_0) = (L-1)/(t_0-1)$ since $L-1$ of the t_0-1 random variables are high *before* t_0 and each such r.v. has an equal chance of being chosen.
- $P^*(B_t = 1 | t = t_0) = 1$ since t_0 is high. Note that this is only possible when $t \neq T$ since $t_0 < T$.
- $P^*(B_t = 1 | t > t_0) = 0$ since all B_t following t_0 are low.
- $P^*(B_t = 1 | t < t_1) = (t_1 - T + L)/(t_1 - 1)$ since $T-L-1$ of the $t_1 - 1$ random variables are low (and thus $t_1 - T - L$ high) *before* t_1.
- $P^*(B_t = 1 | t = t_1) = 0$ since t_1 is low.
- $P^*(B_t = 1 | t > t_1) = 1$ since all B_t following t_1 are high.

Therefore the probability that $B_t = 1$ under the forced suffix distribution is:

$$
\begin{aligned}
P^*(B_t = 1) = &\sum_{t_0=t+1}^{T-1} P^*(t_0) P^*(B_t = 1 | t < t_0) + P^*(t_0 = t) P^*(B_t = 1 | t = t_0) + \\
&\sum_{t_1=t+1}^{T-1} P^*(t_1) P^*(B_t = 1 | t < t_1) + \sum_{t_1=0}^{t-1} P^*(t_1) P^*(B_t = 1 | t > t_1) \\
= &\sum_{t_0=t+1}^{T-1} \frac{L-1}{t_0-1} \binom{t_0-1}{L-1} p^L (1-p)^{t_0-L} + \\
&I[t \neq T] \binom{t-1}{L-1} p^L (1-p)^{t-L} + \\
&\sum_{t_1=t+1}^{T-1} \frac{t_1 - T + L}{t_1 - 1} \binom{t_1 - 1}{T-L-1} p^{t_1 - T + L} (1-p)^{T-L} + \\
&\sum_{t_1=1}^{t-1} \binom{t_1 - 1}{T-L-1} p^{t_1 - T + L} (1-p)^{T-L} + I[T = L] \\
= &I[L > 1] \sum_{t_0=t+1}^{T-1} \binom{t_0-2}{L-2} p^L (1-p)^{t_0-L} + \\
&I[t \neq T] \binom{t-1}{L-1} p^L (1-p)^{t-L} + \\
&\sum_{t_1=t+1}^{T-1} \binom{t_1 - 2}{T-L-1} p^{t_1 - T + L} (1-p)^{T-L} + \\
&\sum_{t_1=T-L}^{t-1} \binom{t_1 - 1}{T-L-1} p^{t_1 - T + L} (1-p)^{T-L} + I[T = L]
\end{aligned}
$$

The contributing summands differ according to t. Assume the usual case that $L < T - L < T$.

$$\begin{aligned}
P^*(B_t = 1|0 < t < L) = & \sum_{t_0=L}^{T-1} \binom{t_0 - 2}{L - 2} p^L (1-p)^{t_0 - L} + \\
& \sum_{t_1=T-L+1}^{T-1} \binom{t_1 - 2}{T - L - 1} p^{t_1 - T + L} (1-p)^{T-L} \\
= & p \sum_{t_0=0}^{T-L-1} \binom{t_0 + L - 2}{t_0} p^{L-1} (1-p)^{t_0} + \\
& p \sum_{t_1=0}^{L-2} \binom{t_1 + T - L - 1}{t_1} p^{t_1} (1-p)^{T-L} \\
= & p \left(\bar{B}(p; L-1, T-L) + \bar{B}(1-p; T-L; L-1) \right) \\
= & p \left(1 - \bar{B}(1-p; T-L; L-1) + \bar{B}(1-p; T-L; L-1) \right) \\
= & p
\end{aligned}$$

$$P^*(B_t = 1|L \leq t \leq T - L, L = 1) = p(1-p)^{t-1}$$

$$\begin{aligned}
P^*(B_t = 1|1 < L \leq t \leq T - L) = & \sum_{t_0=t+1}^{T-1} \binom{t_0 - 2}{L - 2} p^L (1-p)^{t_0 - L} + \\
& \binom{t-1}{L-1} p^L (1-p)^{t-L} + \\
& \sum_{t_1=T-L+1}^{T-1} \binom{t_1 - 2}{T - L - 1} p^{t_1 - T + L} (1-p)^{T-L} \\
= & p + \binom{t-1}{L-1} p^L (1-p)^{t-L} - \\
& p\bar{B}(p; L-1, t-L+1)
\end{aligned}$$

where $\bar{B}(p; r, k)$ is the regularized incomplete beta function, which is also the CDF of the negative binomial distribution with success probability p, r successes and k failures.

Noting $P^*(B_L = 1|L < T - L) = p$ both when $L = 1$ and $L \neq 1$, we can extend the first interval to include L, that is

$$P^*(B_t = 1|0 < t \leq L) = p$$

which allows us to simplify

$$p^*(B_t = 1|L < t \leq T - L) = p \left(1 - \bar{B}(p; L, t - L) \right)$$

Where we have dropped the restriction that $L > 1$ since the right-hand side evaluates to the same value as the right-hand side for $P^*(B_t = 1|L \leq t \leq T - L, L = 1)$.

$$
\begin{aligned}
P^*(B_t = 1|T-L<t<T) = &\sum_{t_0=t+1}^{T-1} \binom{t_0-2}{L-2} p^L (1-p)^{t_0-L} + \\
&\binom{t-1}{L-1} p^L (1-p)^{t-L} + \\
&\sum_{t_1=t+1}^{T-1} \binom{t_1-2}{T-L-1} p^{t_1-T+L} (1-p)^{T-L} + \\
&\sum_{t_1=T-L}^{t-1} \binom{t_1-1}{T-L-1} p^{t_1-T+L} (1-p)^{T-L} \\
=&p\left(1-\bar{B}(p;L,T-L)\right) - \\
&\sum_{t_1=0}^{t-T+L-1} \binom{t_1+T-L-1}{t_1} p^{t_1+1} (1-p)^{T-L} + \\
&\sum_{t_1=0}^{t-T+L-1} \binom{t_1+T-L-1}{t_1} p^{t_1} (1-p)^{T-L} \\
=&p\left(1-\bar{B}(p;L,t-L)\right) + \\
&(1-p)\bar{B}(1-p;T-L;t-T+L)
\end{aligned}
$$

$$
\begin{aligned}
P^*(B_T = 1) = &\sum_{t_1=T-L}^{T-1} \binom{t_1-1}{T-L-1} p^{t_1-T+L} (1-p)^{T-L} \\
=&\sum_{t_1=0}^{L-1} \binom{t_1+T-L-1}{t_1} p^{t_1} (1-p)^{T-L} \\
=&\bar{B}(1-p;T-L,L) \\
=&p\left(1-\bar{B}(p;L,T-L)\right) + \\
&(1-p)\bar{B}(1-p;T-L;T-T+L)
\end{aligned}
$$

The right-hand side of $P^*(B_T = 1)$ equals the right-hand side of $P^*(B_t = 1|T-L<t<T)$, meaning we can extend the latter interval to include T.

Collecting the various conditions on t together into a piecewise function for the probability of $B_t = 1$ under the forced-suffix distribution:

$$
\begin{aligned}
P^*(B_t = 1|L<T-L<T) =&p - I[t>L]p\bar{B}(p;L,t-L) + \\
&I[t>T-L](1-p)\bar{B}(1-p;T-L;t-T+L)
\end{aligned} \tag{47}
$$

We can interpret eq. (47) as follows. The probability of the first L samples will never be tampered with when $L < T-L$ because there will always be at least one remaining high sample possible and at least L samples following. Thus the probability for $B_t = 1$ for $t \leq L$ will be merely the Bernoulli probability.

Once $t > L$, some prefixes $b_{\leq t}$ (and all paths extending them) become invalid because they have too many high Bernoulli values. The $I[t > L] \ldots$ term sums the probabilities of all paths that would have contributed to the probability of $B_t = 1$, but have $L + 1$ or more high values (including B_t). When t approaches the end of the sequence, $t > T - L$, the probability of $B_t = 1$ claims some additional probability mass of invalidated paths which would otherwise set $B_t = 0$ but are forced to 1 in order to make the total number of emissions L. The $I[t > T - L] \ldots$ term sums the probability of those paths, which have $T - L + 1$ *low* values (including the $B_t = 0$ term).

B The ideal approximation

Suppose we want to design an ideal approximation for the conditional probability $P(B_t = 1|b_{<t}, L)$ via function family $f(b_{<t}, p_{\leq t}, T, L)$ and one for the prior $P(L)$ via function family $h(p_{\leq t}, T, L)$, $p_t = P(B_t = 1|b_{<t})$. The family f only has access to the prefix of samples $b_{<t}$ and probabilities $p_{\leq t}$, not any future probabilities, because p_t may depend on $b_{<t}$ at test time (during which we incrementally build the sample sequence b).

Critically, we require that f follow the restriction that only L high samples may be generated total per sequence. Formally, letting $\ell_{<t} = \sum_{t'=1}^{t-1} b_{t'}$, these restrictions can be stated as

1. $f(b_{<t}, p_{\leq t}, T, L) = 1$ if $T - t + 1 = L - \ell_{<t}$.
2. $f(b_{<t}, p_{\leq t}, T, L) = 0$ if $L - \ell_{<t} = 0$.

that is, the probability of sampling $B_t = 1$ becomes 1 when the number of remaining random variables equals the remaining number of highs and becomes 0 when all L highs have already occurred.

These restrictions gloss over the possibility that some $p_{\geq t}$ may be zero. Indeed, our discussion assumes that $\forall t \in [1, T] : p_t \in (0, 1)$, making all $b \in \{0, 1\}^T$ valid samples when L is free. Were we to allow for $p_t = 0$, the only f capable of producing valid samples would be

$$P(b_{<t}, p_{\leq t}, T, L) = I[p_t > 0]I[\ell_{<t} < L]$$

that is, emit all high samples as quickly as possible and stop when none remain. If f were not identically 1 for nonzero p_t, one could take $B_t = 0$ for one of those $p_t < 0$ and choose $p_{\geq t}$ such that only $L - \ell_t - 1$ remaining probabilities are nonzero. Suffice to say the above approximation is far from ideal. Restricting $p_t > 0$ is fair considering p_t is likely a function of some exponentiated real value from a neural neural network and cannot even take the value 0.

The other component of an ideal approximator is that, were we to multiply $P(b|L)$ with its prior $P(L)$, we would recover the sample probability $P(b) = P(b_1)P(b_2|b_1) \ldots P(b_T|b_{<T})$. Stated in terms of the approximations:

3. $h(p, T, \ell_T) \prod_{t=1}^{T} f(b_{<t}, p_{\leq t}, T, L)^{b_t} + (1 - f(\ldots))^{1-b_t} = \prod_{t=1}^{T} p_t^{b_t}(1 - p_t)^{1-b_t}$

Restrictions 1-3 cannot be met simultaneously. We prove this by counterexample. Choose $T = 2$ and $L = 1$. Note that $B_1 = 1 \iff B_2 = 0$. This means $f(b_1, p_{\leq 2}, 2, 1) = I[b_1 = 0]$ by restrictions 1-2. Let $f(\emptyset, p_1, 2, 1) = \alpha$. $B_1 = 1$ with probability α. Plugging in the samples $b \in \{0, 1\}^2$ into restriction 3 gives us a system of two equations with two unknowns (α and $h(\cdot)$):

$$
\begin{aligned}
h(p, 2, 1)\alpha &= p_1(1 - p_2) \\
h(p, 2, 1)(1 - \alpha) &= (1 - p_1)p_2
\end{aligned}
$$

The system has unique solutions

$$
\begin{aligned}
\alpha &= \frac{p_1 - p_1 p_2}{p_1 - 2p_1 p_2 + p_2} \\
h(p, 2, 1) &= p_1 - 2p_1 p_2 + p_3
\end{aligned}
$$

which are both valid probabilities. However, the solution to α is a function of p_2, which is not an input to $f(\emptyset, p_1, 2, 1)$. α must be constant for $p = (p_1, p_2)$ and $p' = (p_1, p_2')$, $p_2 \neq p_2'$, which means it will not satisfy restriction 3 for all choices of p.

This implies there are no f and h that can be decomposed nicely into $P(b|L)$ and $P(L)$ respectively. In the above counterexample, p_2 and p_2' represent the marginal probabilities of the second Bernoulli trial across two different distributions p and p'; they are not conditioned on b_1. Thus we cannot make the decomposition even with the independence assumption, i.e. $P(b_t|b_{<t}) = P(b_t)$

While no h can act as a proper prior $P(L)$, we can still recover the unconditioned probability of a sample b by relaxing the constraints on h. In particular, if we allow h to depend on the sample b, finding h is trivial:

$$
h(b, p, T) = \frac{\prod_{t=1}^{T} p_t^{b_t}(1 - p_t)^{1-b_t}}{\prod_{t=1}^{T} f(b_{<t}, p_{\leq t}, T, L)^{b_t} + (1 - f(\ldots))^{1-b_t}} \tag{48}
$$

In fact, were we to call the numerator $P(b, L) = P(b)$ and call the denominator the proposal distribution $\prod_{t=1}^{T} f(b_{<t}, p_{\leq t}, T, L)^{b_t} + (1 - f(\ldots))^{1-b_t} = Q(b, L) = Q(b)$, then maximizing h jointly with $p(y|b)$ gives us

$$
\mathbb{E}_{b\sim Q}\left[P(y|b)h(b)\right] = \mathbb{E}_{b\sim Q}\left[P(y|b)\frac{P(b)}{Q(b)}\right] = P(y) \tag{49}
$$

and applying Jensen's inequality gives us the a lower bound from Variational Inference (VI):

$$
\log P(y) \geq \mathbb{E}_{b\sim Q}\left[\log \frac{P(y, b)}{Q(b)}\right] \tag{50}
$$

Since both $P(b)$ and $Q(b)$ factor over time, a similar REINFORCE objective to those in Lawson et al. [16], but with the additional restrictions 1-2 on $Q(b)$.

If we consider the right-hand side of eq. (50) an estimator for the left-hand side $\log P(y)$, we can see that the estimator is biased unless both sides are equal.

This can only occur when $P(b) = Q(b)$ [15]. From the above proof involving h and f we know that $P(b) \neq Q(b)$ in general. Thus, for general P, variational lower bounds such as those in eq. (50) *will always be biased*.

We can mitigate the bias in eq. (50) by taking the mean ratio over many samples inside the log [5]:

$$\log P(y) \geq \mathbb{E}_{b \sim Q}\left[\log \frac{1}{K} \frac{P(y, b)}{Q(b)}\right] \geq \mathbb{E}_{b \sim Q}\left[\log \frac{P(y, b)}{Q(b)}\right] \tag{51}$$

A sum within the log term precludes factorizing $P(b)$ and $Q(b)$ over time. Thus, the decrease in bias of eq. (51) is at the cost of the higher variance of a global reward function.

Finally, we note that dropping the logs and maximizing the expectation of $P(y)$ directly will allow for an unbiased estimator, but such an estimator would be susceptible to underflow.

C TODO

More general bounds on the distribution over trials of the Luo et al. when $P(b_t) = 0.5$. A bit stronger than merely explaining the bias and giving an example when $T = 3$.

Push the proofs from the appendix into the text itself, and remove the "alternate proof" part (go straight to variance).

Motivations is not working as an intro

Hierarchical learning. Either through the estimators from section 2.2 or the exact form derived in section 2.4, the CB can be used to transduce a sequence of length T into one of length L. The advantage that the CB has over CTC is its ability to incorporate and simultaneously learn the dynamics of the L sequence while learning those of the T sequence. This does not prevent us from stacking a third sequence $M \ll L$ on top of the L sequence, or even deeper. For example, we could choose T to be acoustic frames, L to be sub-word units, and M to be full words. We could even recreate something akin to the FST composition used in hybrid speech recognition.

Prove/disprove that no distribution satisfies the following three requirements. Work in progress.

Let $B_t \in \{0, 1\}$ be a binary random variable. There are T such B_t and, if we condition on L, there must be $L \ll T$ (and only L) "high" B_t, i.e. $\sum_t B_t = L$.

1. *Conditional:* $P(b_t | b_{<t}, L, T)$ is well-defined and tractable.
2. *Marginal:* $P(b_t | b_{<t}, T) = P(b_t | b_{<t})$. That is, we can marginalize out the total number of "high" events L in the sequence in a way agnostic to the remaining number of total variables $T - t$.
3. *Prefix-dependent:* Let $p_t = P(B_t = 1 | b_{<t}, [L, T])$ (with or without conditioning on L and T). Then

$$\exists b_{<t}, b'_{<t} \quad \sum_t b_{<t} = \sum_t b'_{<t} \wedge p_t \neq p'_t$$

In other words, we can modify the distribution over B_t in a manner dependent on the sequence of $B_{<t}$, not just the number of highs in the sequence.

The BB and other CB variants satisfy the conditional and marginal requirements, but not the prefix-dependent one. The models of e.g. Luo et al. [17], Raffel et al. [20] satisfy the marginal and prefix-dependent requirements, but not the conditional.

Though I have yet to find a model in the literature that satisfies the conditional and prefix-dependent conditions simultaneously, I can provide a pathological distribution that does satisfy both: have some auto-regressive RNN parameterize each Bernoulli trial as a function of the previous trials ($b_t \sim w_t = \text{RNN}(b_{<t})$) up to and including event L, then make the suffix $b_{>L}$ a deterministic function of the remaining high events necessary such that $\sum_t b_t = L$. Since each suffix has probability 1, and $P(\sum_t b_{\leq L} \leq L) = 1$, the first L events can act as if there is no fixed number of trials. The model is free to adjust the probabilities of those trials in a prefix-dependent way.

A few thoughts so far:

"Tractable" needs to be defined. Specifically, we want to avoid iterating over all $B_{>t}$ s.t. $\sum_t B_t = L$ and setting $P(b_t|\ldots)$ to the truncated distribution. I think this will be difficult. If we can formulate this, we'll be a good chunk of the way done.

Being auto-regressive (i.e. parameterizing b_t as a function of $b_{<t}$) is certainly sufficient with respect to the prefix-dependent property. I believe they might be identical criteria.

The processes of Raffel et al. [20] (and, by extension, Chiu and Raffel [8]) reduce to an auto-regressive sequence of Bernoulli trials at test time, just like Luo et al.. The so-called "expectation" calculated by the authors during training ignores conditioning on L, just like Luo et al., and is subject to the same sort of bias exemplified in fig. 1.

I think the difficult part of finding a closed-form distribution that is both auto-regressive and does not enumerate the log-odds in the *prior*, not the posterior. For example, we can get a closed-form step by treating the parameterized probability p_t as a surrogate for all future probabilities and using the binomial distribution:

$$\begin{aligned} P(B_t = 1, \ell_t) &= \binom{T-t}{L-\ell_t-1} p_t^{L-\ell_t}(1-p_t)^{T-t-L+\ell_t+1} \\ P(\ell_t) &= \binom{T-t+1}{L-\ell} p_t^{L-\ell_t}(1-p_t)^{T-t-L+\ell_t+1} \\ P(B_t = 1|\ell_t) &= \frac{L-\ell_t}{T-t+1} \end{aligned}$$

where the last equation follows from dividing the first by the second. This conditional is trivial to calculate (it doesn't even involve p_t), but calculating $P(L)$ would still require summing the probability of all sequences that have L highs.

Thus, requirement 1 should probably be switched to something like "the prior is well-defined and tractable."

Some things to follow up on towards a proof:

Duration-based HMM sampling might give some ideas on how to proceed. Likewise with the *Cover & Thomas proof*.

Prove equivalence up to a negligible quality. If a distribution satisfies requirements 1 & 2, is it reducible to the CB distribution up to some irrelevant (e.g. linear) transformation? Likewise, if a distribution satisfies 1 & 3, does it reduce to a sequence of dependent binary trials?