Instructions for EMNLP 2020 Proceedings

Anonymous EMNLP submission

Abstract

todo

1 Introduction

todo

2 Background

We focus on language modeling, where we learn a distribution over sequences of tokens at the word level $\mathbf{x} = \{x_0, \ldots, x_T\}$, with each token x_t an element of the finite vocabulary $\mathcal{X}$.

2.1 Hidden Markov Models

Hidden Markov Models (HMMs) specify a joint distribution over the observed words $\mathbf{x}$ and discrete latent states $\mathbf{z} = \{z_0, \ldots, z_T\}$, each z_t from finite set $\mathcal{Z}$, with the following generative process: For every time step $t \in \{0, \ldots, T\}$, choose a state given the previous state $z_t \mid z_{t-1}$ from the transition distribution $p(z_t \mid z_{t-1})$, where the first state z_0 is chosen from the starting distribution $p(z_0)$. Then choose a word given the current state $x_t \mid z_t$ from the emission distribution $p(x_t \mid z_t)$. This yields the joint distribution

$$p(\mathbf{x}, \mathbf{z}) = p(z_0)p(x_0 \mid z_0) \prod_{t=1}^{T} p(x_t \mid z_t)p(z_t \mid z_{t-1}) \tag{1}$$

The distributions are parameterized as follows

$$\begin{aligned} p(z_0) &\propto e^{\phi_{z_0}} \\ p(z_t \mid z_{t-1}) &\propto e^{\phi_{z_t z_{t-1}}} \\ p(x_t \mid z_t) &\propto e^{\phi_{x_t z_t}} \end{aligned} \tag{2}$$

where each $\phi_{z_0}, \phi_{z_t z_{t-1}}, \phi_{x_t z_t} \in \mathbb{R}$ may be parameterized by a scalar or a neural network.

3 Model

For language modeling, we are interested in the distribution over the observed words obtained by marginalizing over the latent states $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$. This sum can be computed in time $O(T|\mathcal{Z}|^2)$ via dynamic programming, which becomes prohibitive if the number of latent states $|\mathcal{Z}|$ is large. However, in order to increase the capacity of HMMs, we must consider large $\mathcal{Z}$. We outline several techniques to manage the computational complexity of marginalization in HMMs.

3.1 Sparse Emission HMMs

We introduce a sparse emission constraint that allows us to efficiently perform conditional inference in large state spaces. Inspired by Cloned HMMs (Dedieu et al., 2019), we constrain each word x to be emit only by $z \in \mathcal{C}_x \subset \mathcal{Z}$:

$$p(x \mid z) \propto 1(z \in \mathcal{C}_x) e^{\phi_{zx}} \tag{3}$$

This allows us to perform marginalization as follows

$$\begin{aligned} p(\mathbf{x}) &= \sum_{z_0} p(z_0)p(x_0 \mid z_0) \sum_{z_1} p(z_1 \mid z_0)p(x_1 \mid z_1) \\ &\quad \cdots \sum_{z_T} p(z_T \mid z_{T-1})p(x_T \mid z_T) \\ &= \sum_{z_0 \in \mathcal{C}_{x_0}} p(z_0)p(x_0 \mid z_0) \sum_{z_1 \in \mathcal{C}_{x_1}} p(z_1 \mid z_0)p(x_1 \mid z_1) \\ &\quad \cdots \sum_{z_T \in \mathcal{C}_{x_T}} p(z_T \mid z_{T-1})p(x_T \mid z_T) \end{aligned} \tag{4}$$

which takes $O(T \max_x(|\mathcal{C}_x|)^2)$ computation. We choose sets $\mathcal{C}_x$ such that $\forall x, |\mathcal{C}_x| = k$ due to convenience of implementation.

We experiment with two constraint sets, both of which are specified as hyperparameters and remain

fixed for a given model. The first constraint set is obtained by sampling subsets of states for each word i.i.d. from a uniform distribution over subsets. The second constraint set uses Brown Clusters (Brown et al., 1992) to assign all words in a given Brown cluster the same subset of states. This constraint set is advantageous, as it admits further optimizations that we discuss below.

3.2 State Dropout

We introduce a variant of dropout called state dropout that operates on the start and transition distributions. The goal of state dropout is to encourage usage of the full state space in HMMs as well as to reduce the complexity of marginalization in a manner identical to the sparse emission constraints.

State dropout samples a mask over states $\mathbf{b}_{\mathcal{C}} \in \{0,1\}^{|\mathcal{C}|}$ for each partition $\mathcal{C} \subset \mathcal{Z}$, such that each sampled $|\mathbf{b}_{\mathcal{C}}| = n$. We only apply this if the $\mathcal{C}_x$ are disjoint partitions, as with the Brown constraint set.

Let $\mathbf{b}$ be the concatenation of the $\mathbf{b}_{\mathcal{C}}$ for all partitions $\mathcal{C}$, such that $b_z = 1$ if $b_{\mathcal{C}z} = 1$, where $z \in \mathcal{C}$. The start and transition distributions with dropout are given by

$$\begin{aligned} p(z_0) &\propto b_{z_0} e^{\phi_{z_0}} \\ p(z_t \mid z_{t-1}) &\propto b_{z_t} b_{z_{t-1}} e^{\phi_{z_t z_{t-1}}} \end{aligned} \tag{5}$$

Marginalizing over $\mathbf{z}$ with state dropout requires $O(Tn^2)$ computation, where n is the number of nonzero elements of $\mathbf{b}_{\mathcal{C}}, \forall \mathcal{C}$. We utilize state dropout on top of the Brown emission sparsity constraint, subsampling states of size n for each partition during training.

We also consider unstructured dropout which samples elements of masks $b_{z_0}, b_{z_t z_{t-1}} \overset{iid}{\sim} \text{Bern}(p)$ for all $z_0, z_t, z_{t-1} \in \mathcal{Z}$, yielding

$$\begin{aligned} p(z_0) &\propto b_{z_0} e^{\phi_{z_0}} \\ p(z_t \mid z_{t-1}) &\propto b_{z_t z_{t-1}} e^{\phi_{z_t z_{t-1}}}. \end{aligned} \tag{6}$$

Unstructured dropout results in sparsity patterns that are more difficult to take advantage of than state dropout, since there is variance in the number of nonzero elements in a mask. (Fix wording)

We find that state dropout has similar regularization effect as unstructured dropout, but additionally yields computational benefits.

3.3 Learning

We optimize the evidence of observed sentences $\log p(\mathbf{x}) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$ by first marginalizing over latent states $\mathbf{z}$ then performing gradient ascent. The emission sparsity constraints we introduce as well as state dropout allow us to both perform marginalization and compute the gradient of the evidence efficiently.

4 Experiments

We evaluate the HMM on two language modeling datasets.

4.1 Language Modeling

Datasets The two datasets we used are the `Penn Treebank` (Marcus et al., 1993) and `wikitext2` (Merity et al., 2016). `Penn Treebank` contains 929k tokens in the training corpus, with a vocabulary size of 10k. `Wikitext2` contains 2M tokens in the training corpus, with a vocabulary size of 33k. Both datasets contain inter-sentence dependencies, due to the use of un-shuffled documents.

Implementation We train two-layer LSTM recurrent neural networks with 256 or 512 units, as well as two-layer feed-forward neural networks with 256 or 512 units. The HMMs we train follow the sparsity constraints outlined in the previous section with a dropout rate of 0.5, and we vary the total number of states as well as states per word. We optimize all models with AdamW (Loshchilov and Hutter, 2017). See Appendix A for the hyperparameters for all models.

4.2 Results

We report validation (TODO: will include test) perplexities for `Penn Treebank` in Table 1 and for `wikitext2` in Table 2. The HMMs in all cases outperform the feedforward models (FF), but underperform the recurrent LSTMs.

Sparse emission constraint ablation We ablate the emission sparsity constraints in Table 3, and find that the Brown emission constraints outperforms the uniform emission constraints in all models.

Analysis

Table 1: Perplexities on the `Penn Treebank` dataset. The FF model is a 256-dim 2-layer feedforward neural network with a window of 4 previous tokens with 0.3 dropout. The LSTM is a 256-dim 2-layer recurrent neural network with 0.3 dropout. The HMM is a 32k-state HMM with 0.5 state dropout.

Model	Num Params	Valid PPL	Test PPL
FF		160	-
LSTM		90	-
HMM		124	-

Table 2: Perplexities on the `wikitext2` dataset. The FF model is a 256-dim 2-layer feedforward neural network with a window of 4 previous tokens with 0.3 dropout. The LSTM is a 256-dim 2-layer recurrent neural network with 0.3 dropout. The HMM is a 32k-state HMM with 0.5 state dropout.

Model	Num params	Valid PPL	Test PPL
FF		209	-
LSTM		125	-
HMM		167	-

- Hypothesis: Brown Cluster assigns words that are emit in similar contexts. Uniform assignment with too few states per word will not group words together resulting in needing more states.
- Evidence: check entropy of emission distribution / Uniform should be lower entropy.
- Conclusion:

Dropout ablation Graph 1: Show dropout helps but variants don't affect performance too much, and encourages better state usage (vs no dropout). Justify state dropout as the one with simplest implementation.

- Brown 16k 128 + unstructured dropout: ppl and state usage
- Brown 16k 128 + state dropout
- Brown 16k 128 + no dropout

Analysis 1: Discuss what happens with just state dropout, without partitioning.

States per word (k) ablation We next examine the sensitivity of HMMs to the number of Brown clusters.

Table 3: The perplexities for the different emission sparsity constraints in a 1024 state HMM as well as larger HMMs for which exact inference without sparsity is too expensive. The quantities $|\mathcal{Z}|$ and k are the number of hidden states and the number of states per word respectively. The HMMs with 1024 states do not have any dropout, while the 8k and 16k state HMMs have unstructured dropout.

Constraint	$\lvert\mathcal{Z}\rvert$	k	Valid PPL
Uniform	1024	32	-
Brown	1024	32	-
None	1024	1024	-
Uniform	8k	128	150
Brown	8k	128	142
Uniform	16k	128	146
Brown	16k	128	134

Table 4: The perplexities for HMMs with 16k states and different numbers of Brown clusters for constraining the emission distribution of the HMMs. $|\mathcal{Z}|$ is the total number of hidden states. All models have 0.5 state dropout.

$\lvert\mathcal{Z}\rvert$	Num clusters	Valid PPL
16k	32	136
16k	64	137
16k	128	133
16k	256	137

Weight tying ablation Analysis 1: Discuss performance using different tying methods (found this to not affect performance) and computational savings in terms of number of parameters. Discuss relative parameter inefficiency compared to LSTM / FF.

State size ablation Graph 1: Show performance improves as we increase the total number of states

- Brown HMM + state dropout, 16k, 128 clusters
- Brown HMM + state dropout, 32k, 128 clusters
- Brown HMM + state dropout, 65k, 128 clusters

Analysis 1: How does state usage change as clusters remain constant but the number of states (and states per word) increases?

Parameterization ablation Analysis 1: Is there a performance difference between neural / scalar parameterization, and is it consistent across state sizes?

Discussion 1: Memory savings when working with state dropout to not instantiate the full matrices during training, which makes working with a neural parameterization beneficial.

4.3 Error analysis

Acknowledgments

TBD

References

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Antoine Dedieu, Nishad Gothoskar, Scott Swingle, Wolfgang Lehrach, Miguel Lázaro-Gredilla, and Dileep George. 2019. Learning higher-order sequential structure with cloned hmms.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

A Hyperparameters

LSTM

- 2 layers

B Parameter estimation

We maximize the evidence of the observed sentences $\log p(\mathbf{x}) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$ via gradient ascent.

B.1 Gradient of the evidence

Let $\psi_0(z_0, z_1) = \log p(x_0, z_0)$ and $\psi_t(z_t, z_{t+1}) = \log p(x_{t+1}, z_{t+1} \mid z_t)$ for $t \in \{1, \ldots, T-1\}$. Additionally, let $\oplus$ and $\otimes$ be addition and multiplication in the log semiring. After conditioning on the observed $\mathbf{x}$, we can express the evidence as the following:

$$A_x = \log p(\mathbf{x}) = \bigoplus_{\mathbf{z}} \bigotimes_{t=0}^{T-1} \psi_t(z_t, z_{t+1}) \tag{7}$$

where $A_{\mathbf{x}}$ is the clamped log partition function (after observing $\mathbf{x}_{0:T}$).

We show that the gradient of the log partition function with respect to the $\psi_t(z_t, z_{t+1})$ is the first moment (a general result for the cumulant generating function of exponential family distributions). Given the value of the derivative $\frac{\partial A_{\mathbf{x}}}{\partial \psi_t(z_t, z_{t+1})}$, we can then apply the chain rule to compute the total derivative with respect to downstream parameters. For example, the gradient with respect to the transition matrix of the HMM is given by

$$\frac{\partial A_{\mathbf{x}}}{\partial \theta_{ij}} = \sum_t \frac{\partial A_{\mathbf{x}}}{\partial \psi_t(i,j)} \frac{\partial \psi_t(i,j)}{\partial \theta_{ij}}$$

Recall that the derivative of logaddexp ($\oplus$ in the log semiring) is

$$\begin{aligned} \frac{\partial}{\partial x} x \oplus y &= \frac{\partial}{\partial x} \log e^x + e^y \\ &= \frac{e^x}{e^x + e^y} \\ &= \exp(x - (x \oplus y)) \end{aligned} \tag{8}$$

while the derivative of addition ($\otimes$ in the log semiring) is

$$\frac{\partial}{\partial x} x \otimes y = 1 \tag{9}$$

The derivative of the clamped log partition func-

tion $A_{\mathbf{x}}$ is given by

$$
\begin{aligned}
&\frac{\partial A_{\mathbf{x}}}{\partial \psi_t(i,j)} \\
&= \frac{\partial}{\partial \psi_t(i,j)} \bigoplus_{\mathbf{z}} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \\
&= \frac{\partial}{\partial \psi_t(i,j)} \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right. \\
&\qquad \left. \bigoplus \left(\bigoplus_{\mathbf{z}: z_t \neq i, z_{t+1} \neq j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right) \\
&= \exp \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right. \\
&\qquad - \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right. \\
&\qquad\qquad \left. \left. \bigoplus \left(\bigoplus_{\mathbf{z}: z_t \neq i, z_{t+1} \neq j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right) \right) \\
&= \exp \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) - A_{\mathbf{x}} \right)
\end{aligned}
$$

which is the value at i and j of the edge marginal for z_t, z_{t+1} obtained via the forward-backward algorithm. The second equality is a result of the associativity of $\oplus$; the third equality is a result of Eqn. 8; and the fourth equality from Eqn. 9.

C Supplemental Material