Scaling Hidden Markov Language Models

Anonymous

2020

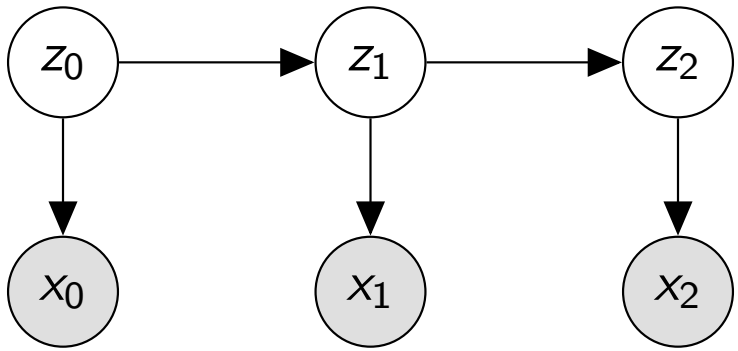
Motivation for HMMs

- Generative process separates the generation of the latent states from the observed tokens
  - LSTMs couple the two
- Discrete latent representations
  - May improve data efficiency (Jin et al., 2020)

HMM LMs

- Previously, HMMs were thought to be very poor language models
  - Past work improved performance only by departing from the HMM structure and turning them into RNNs (Buys et al., 2018)
- We show that HMM performance can be vastly improved by scaling the number of hidden states
  - While remaining faithful to the original structure!

HMMs



Joint distribution

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

We refer to the emission / observation matrix $p(x_t \mid z_t)$ as $\mathbf{O}$.

Training HMMs

- Computing the likelihood of the observed sentence $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$ requires $O(T|\mathcal{Z}|^2)$ computation, which makes thousands of states infeasible
- Parameterization of the transitions and emissions may make optimization difficult
- We present three tricks for training large HMMs by targeting these issues

3 Tricks for Scaling HMMs

- A block-sparse emission matrix reduces the computational cost of computing the likelihood
- A compact (neural) parameterization of the transitions and emissions aides optimization
- State dropout further reduces computational cost and reduces overfitting

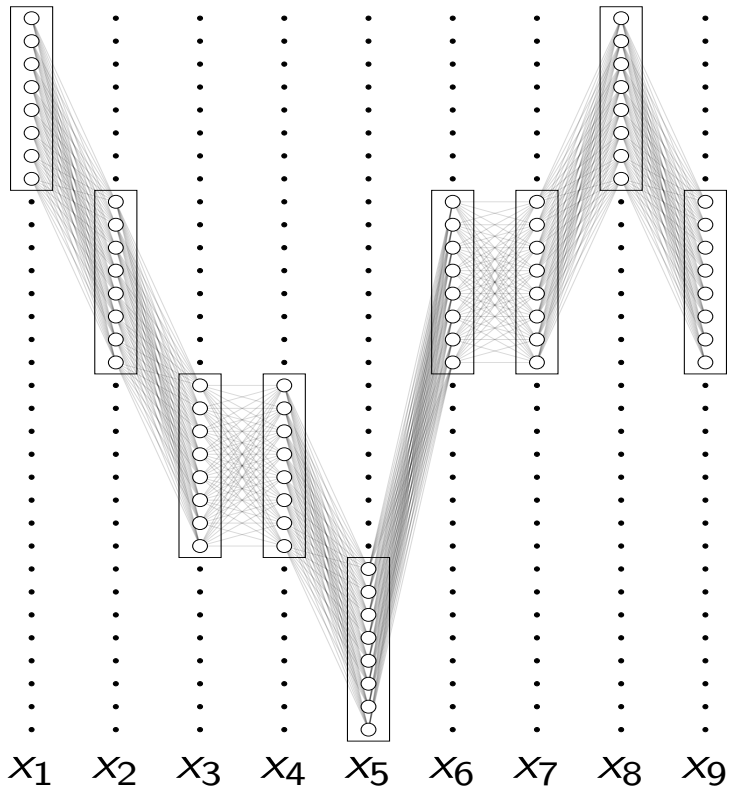
Block-sparse Emissions

- If each state z represents a context, reasonable to assume that not every word is appropriate in every context
- Constrain emissions to

$$\mathbf{O} = \begin{bmatrix} \mathbf{O}^1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \mathbf{O}^M \end{bmatrix}$$

- Each block $\mathbf{O}_m$ contains k latent states and a variable number of tokens
- Results in a serial complexity of $O(Tk^2)$ for computing the likelihood

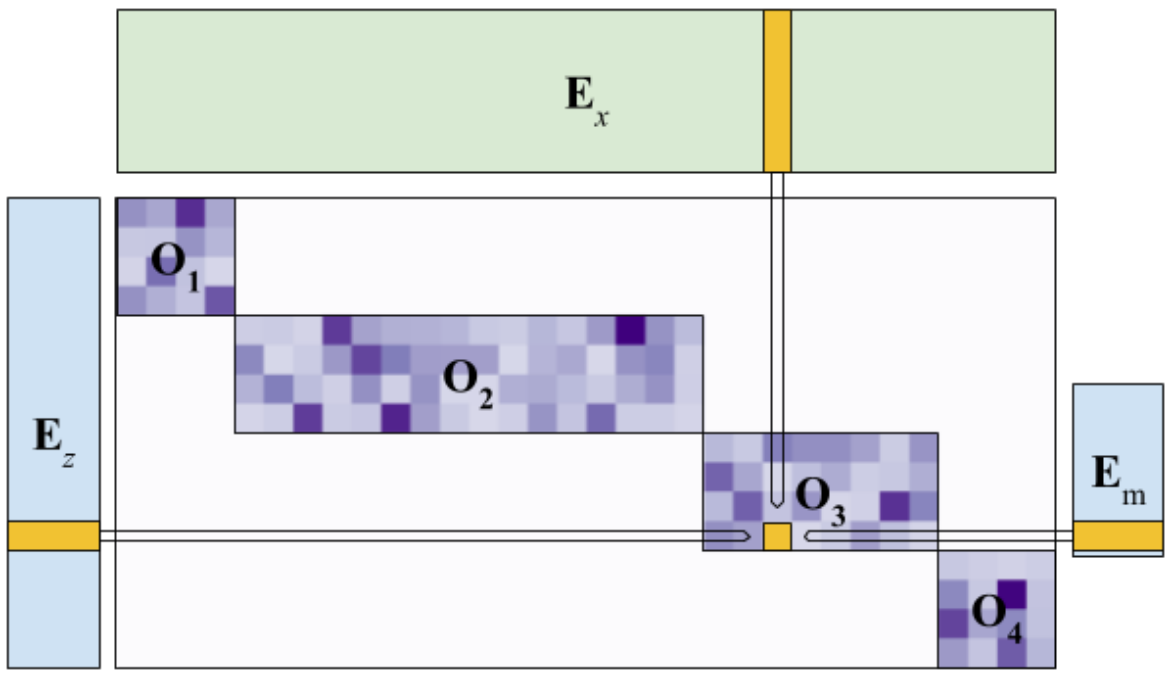
Block-sparse Emissions



After observing $\mathbf{x}$, only the states that emit each x_t have nonzero probability of occurring

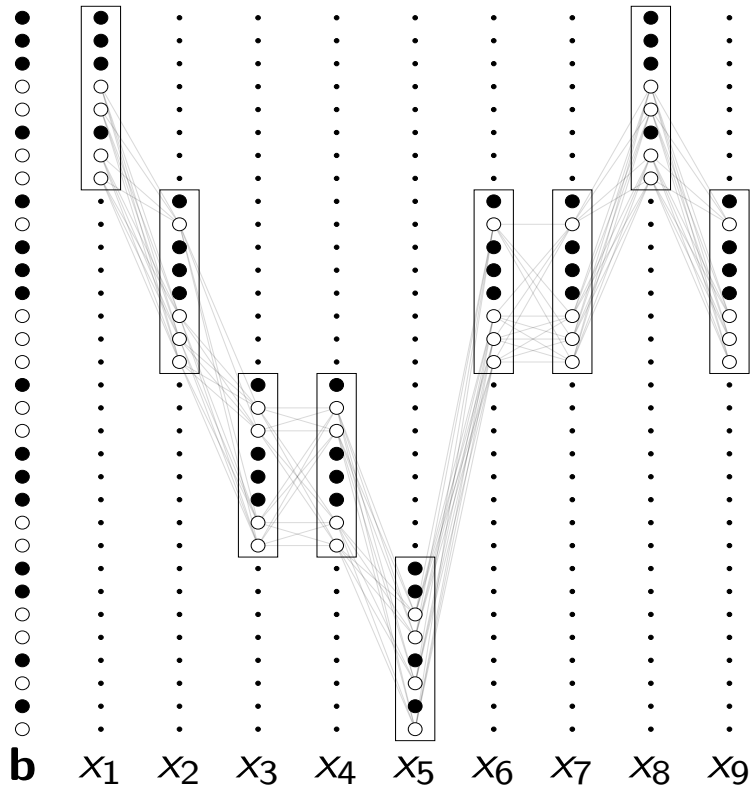
Neural Parameterization

- Compute transition and emission parameters using a neural network
  - State embeddings $\mathbf{E}_z \in \mathbb{R}^{|\mathcal{Z}| \times h/2}$
  - Token embeddings $\mathbf{E}_x \in \mathbb{R}^{|\mathcal{X}| \times h}$
  - Block embeddings $\mathbf{E}_m \in \mathbb{R}^{M \times h/2}$



State Dropout

- Sample a dropout mask $\mathbf{b}_m \in \{0,1\}^k$ for each block $\mathbf{O}_m$
- Concatenate into a global vector $\mathbf{b} = \langle \mathbf{b}_1, \ldots, \mathbf{b}_M \rangle$



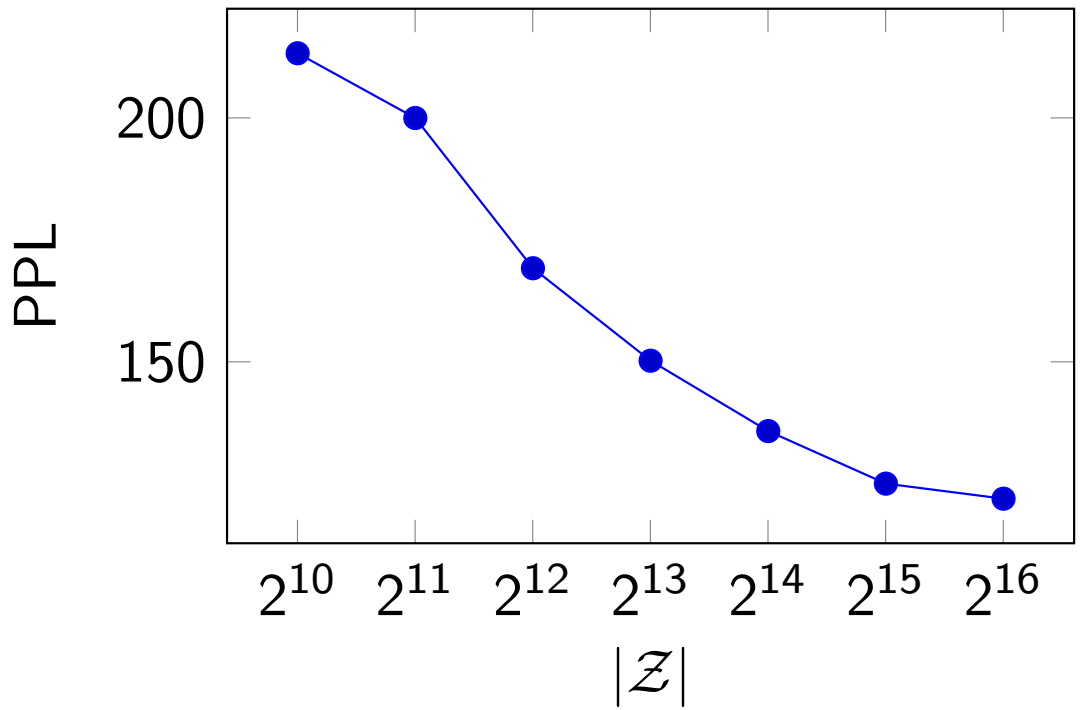
Results on PTB

Model	# Params	Val PPL	Test PPL
KN 5-gram	2M	-	141.2
AWD-LSTM	24M	60.0	57.3
256 FF 5-gram	2.9M	159.9	152.0
2x256 dim LSTM	3.6M	93.6	88.8
HMM+RNN	10M	142.3	-
HMM ($\lvert\mathcal{Z}\rvert$=900)	10M	284.6	-
VL-NHMM ($\lvert\mathcal{Z}\rvert = 2^{15}$)	7.7M	125.0	115.8

Results on WikiText2

Model	# Param	Val PPL	Test PPL
KN 5-gram	5.7M	248.7	234.3
AWD-LSTM	33M	68.6	65.8
256 FF 5-gram	8.8M	210.9	195.0
2x256 LSTM	9.6M	124.5	117.5
VL-NHMM ($\lvert\mathcal{Z}\rvert = 2^{15}$)	13.7M	169.0	158.2

State Size Ablation



Perplexity on PTB by state size $|\mathcal{Z}|$ ($\lambda = 0.5$ and $M = 128$)

Citations

Jan Buys, Yonatan Bisk, and Yejin Choi. 2018. Bridging hmms and rnns through architectural transformations.

Shuning Jin, Sam Wiseman, Karl Stratos, and Karen Livescu. 2020. Discrete latent variable representations for low-resource text classification. In *ACL*.