Scaling Hidden Markov Language Models

Anonymous

2020

Hidden Markov Models in NLP

- Historically significant latent variable models
  - Applied to tagging, alignment, and language modeling in the 90s
- Are thought to be very poor language models
  - We show they are not!

Lessons from Large Neural Language Models

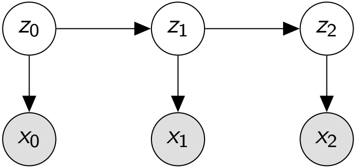
Large models perform better but are ...

- Slow to train
  - Parallelize computation and use GPUs
- Prone to overfitting
  - Regularize

Apply this to scaling HMMs

HMMs

For times $t \in [T]$, model states $z_t \in \mathcal{Z}$ and tokens $x_t \in \mathcal{X}$



We wish optimize

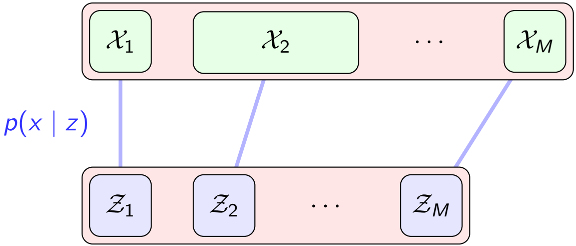
$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$$

3 Tricks for Training Large HMMs

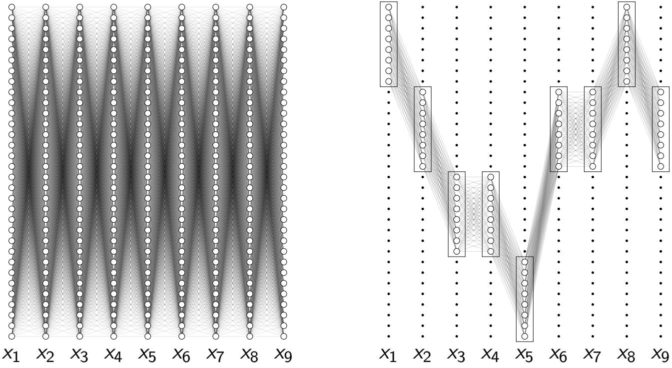
- Block-sparse emission constraints
  - ⇧ Speed
- Compact neural parameterization
  - ⇧ Generalization
- State dropout
  - ⇧ Speed ⇧ Generalization

Block-Sparse Emission Constraints

- Partition words and states jointly
- Words can only be emit by states in same group



Block-Sparse Emissions: Effect on Inference



(a) No constraints

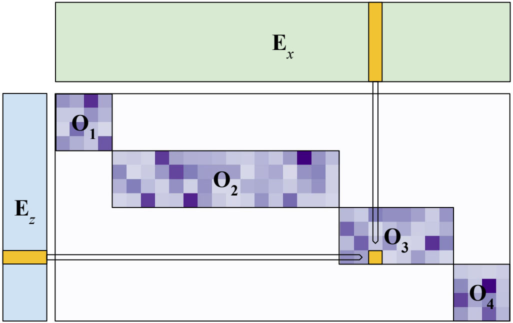
(b) Block-sparse emission constraints

After observing each x_t, only the states in the corresponding group have nonzero probability of occurring

Neural Parameterization

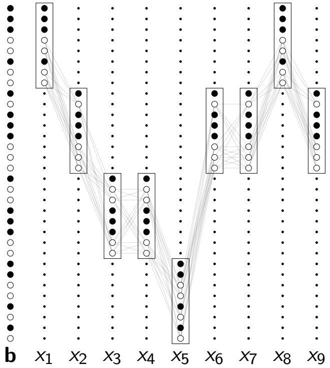
Generate transition and emission distributions using a neural network

- State embeddings $\mathbf{E}_z \in \mathbb{R}^{|\mathcal{Z}| \times h}$
- Token embeddings $\mathbf{E}_x \in \mathbb{R}^{|\mathcal{X}| \times h}$



State Dropout

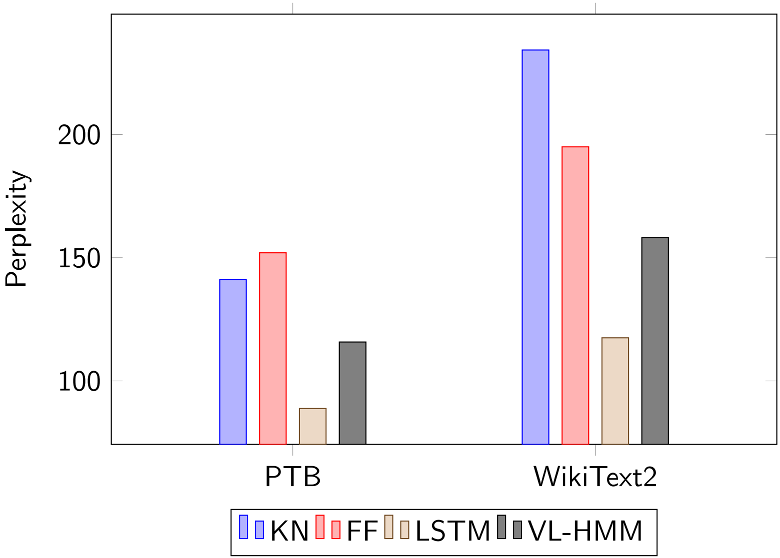
At each batch, sample dropout mask $\mathbf{b} \in \{0,1\}^{|\mathcal{Z}|}$



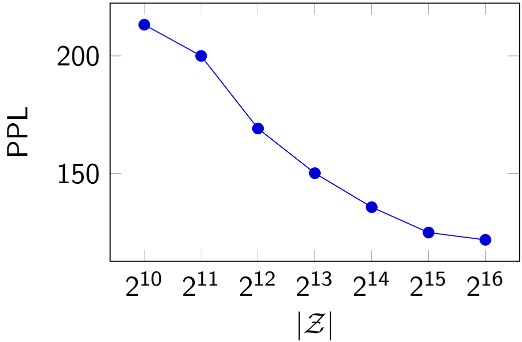
Experiments

- Language modeling on Penn Treebank and Wikitext-2
- Baselines
  - Knesey-Ney 5-gram model
  - Feedforward 5-gram model
  - 2-layer LSTM
- Model
  - 2^{15} (32k) state very large HMM (VL-HMM)
  - $M = 128$ groups (256 states each), obtained via Brown Clustering
  - Dropout rate of 0.5 during training

Results on PTB and WT2



State Size Ablation



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

Other Ablations

Model	Param	Train	Val	Time
VL-HMM (2^{14})	7.2M	115	134	40
- neural param	423M	119	169	14
- state dropout	7.2M	88	157	100

Conclusion

- HMMs can be scaled up to be competitive language models
- Introduced 3 tricks for tackling speed and overfitting
- HMMs are cool!

EOS

Citations