Breaking the Softmax Bottleneck: a High Rank RNN Language Model

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, William W. Cohen

School of Computer Science

Carnegie Mellon University

RNN Language Modeling

- Language Modeling:

$$P(\mathbf{X}) = \prod_t P(X_t \mid X_{<t}) = \prod_t P(X_t \mid C_t)$$

- RNN Language Models: context –(RNN)-> fixed size vector –(word embedding)-> logits –(Softmax)-> categorical probability

- Question: Are the Softmax-based RNN language models expressive enough?

Language Modeling as Matrix Factorization

- Learning task: $P_\theta(X|c) = P^*(X|c)$ for all c in $\mathcal{L}$
- Softmax:

$$P_\theta(x|c) = \frac{\exp \mathbf{h}_c^\top \mathbf{w}_x}{\sum_{x'} \exp \underbrace{\mathbf{h}_c^\top \mathbf{w}_{x'}}_{\text{logit}}}$$

where h_c is context vector (hidden state), w_x is word embedding, all in dimension d.

- Matrix form:

$$\mathbf{H}_\theta = \begin{bmatrix} \mathbf{h}_{c_1}^\top \\ \mathbf{h}_{c_2}^\top \\ \cdots \\ \mathbf{h}_{c_N}^\top \end{bmatrix};\ \mathbf{W}_\theta = \begin{bmatrix} \mathbf{w}_{x_1}^\top \\ \mathbf{w}_{x_2}^\top \\ \cdots \\ \mathbf{w}_{x_M}^\top \end{bmatrix};\ \mathbf{A} = \begin{bmatrix} \log P^*(x_1|c_1), & \log P^*(x_2|c_1) & \cdots & \log P^*(x_M|c_1) \\ \log P^*(x_1|c_2), & \log P^*(x_2|c_2) & \cdots & \log P^*(x_M|c_2) \\ \vdots & \vdots & \ddots & \vdots \\ \log P^*(x_1|c_N), & \log P^*(x_2|c_N) & \cdots & \log P^*(x_M|c_N) \end{bmatrix}$$

where $\mathbf{H}_\theta \in \mathbb{R}^{N \times d}$, $\mathbf{W}_\theta \in \mathbb{R}^{M \times d}$, $\mathbf{A} \in \mathbb{R}^{N \times M}$, and the rows of $\mathbf{H}_\theta$, $\mathbf{W}_\theta$, and $\mathbf{A}$ correspond to context vectors, word embeddings, and log probabilities of the true data distribution respectively.

Language Modeling as Matrix Factorization

We further specify a set of matrices formed by applying *row-wise shift* to $\mathbf{A}$

$$F(\mathbf{A}) = \{\mathbf{A} + \mathbf{\Lambda J}_{N,M} | \mathbf{\Lambda} \text{ is diagonal and } \mathbf{\Lambda} \in \mathbb{R}^{N \times N}\},$$

where $\mathbf{J}_{N,M}$ is an all-ones matrix with size $N \times M$.

Property 1. *For any matrix* $\mathbf{A}'$, $\mathbf{A}' \in F(\mathbf{A})$ *if and only if* $\mathit{Softmax}(\mathbf{A}') = P^*$. *In other words,* $F(\mathbf{A})$ *defines the set of* ***all*** *possible logits that correspond to the true data distribution.*

Based on the Property 1 of $F(\mathbf{A})$, we immediately have the following Lemma.

Lemma 1. *Given a model parameter* θ, $\mathbf{H}_\theta \mathbf{W}_\theta^\top \in F(\mathbf{A})$ *if and only if* $P_\theta(X|c) = P^*(X|c)$ *for all* c *in* $\mathcal{L}$.

Now the expressiveness question becomes: does there exist a parameter θ and $\mathbf{A}' \in F(\mathbf{A})$ such that

$$\mathbf{H}_\theta \mathbf{W}_\theta^\top = \mathbf{A}'.$$

This is essentially a matrix factorization problem. We want the model to learn matrices $\mathbf{H}_\theta$ and $\mathbf{W}_\theta$

Softmax Bottleneck

- the rank of $\mathbf{H}_\theta \mathbf{W}_\theta^\top$ is strictly upper bounded by the embedding size d. since $\mathbf{H}_\theta \in \mathbb{R}^{N \times d}$ and $\mathbf{W}_\theta \in \mathbb{R}^{M \times d}$
- so $d \geq \text{rank}(\mathbf{A}')$

Property 2. *For any* $\mathbf{A}_1 \neq \mathbf{A}_2 \in F(\mathbf{A})$, $|rank(\mathbf{A}_1) - rank(\mathbf{A}_2)| \leq 1$. *In other words, all matrices in* $F(\mathbf{A})$ *have similar ranks, with the maximum rank difference being 1.*

Corollary 1. ***(Softmax Bottleneck)*** *If* $d < rank(\mathbf{A}) - 1$, *for any function family* $\mathcal{U}$ *and any model parameter* θ, *there exists a context* c *in* $\mathcal{L}$ *such that* $P_\theta(X|c) \neq P^*(X|c)$.

- Hypothesize: A is a high rank matrix. (context-dependent)
- Conclusion: when the dimension d is too small, Softmax does not have the capacity to express the true data distribution.
- Increase d?

A high rank language model

- Mixture of Softmax (MoS):

$$P_\theta(x|c) = \sum_{k=1}^{K} \pi_{c,k} \frac{\exp \mathbf{h}_{c,k}^\top \mathbf{w}_x}{\sum_{x'} \exp \mathbf{h}_{c,k}^\top \mathbf{w}_{x'}}; \quad \text{s.t.} \sum_{k=1}^{K} \pi_{c,k} = 1$$

$$\pi_{c_t,k} = \frac{\exp \mathbf{w}_{\pi,k}^\top \mathbf{g}_t}{\sum_{k'=1}^{K} \exp \mathbf{w}_{\pi,k'}^\top \mathbf{g}_t}$$

$$\mathbf{h}_{c_t,k} = \tanh(\mathbf{W}_{h,k} \mathbf{g}_t)$$

$(\mathbf{g}_1, \cdots, \mathbf{g}_T)$ is the sequence of RNN hidden states

A high rank language model

- Mixture of Softmax (MoS):

$$\hat{\mathbf{A}}_{\text{MoS}} = \log \sum_{k=1}^{K} \mathbf{\Pi}_k \exp(\mathbf{H}_{\theta,k} \mathbf{W}_\theta^\top)$$

where $\mathbf{\Pi}_k$ is an $(N \times N)$ diagonal matrix with elements being the prior $\pi_{c,k}$. Because $\hat{\mathbf{A}}_{\text{MoS}}$ is a nonlinear function (*log_sum_exp*) of the context vectors and the word embeddings, $\hat{\mathbf{A}}_{\text{MoS}}$ can be arbitrarily high-rank. As a result, MoS does not suffer from the rank limitation, compared to Softmax.

Experiments: Language Modeling

Model	#Param	Validation	Test
Mikolov & Zweig (2012) – RNN-LDA + KN-5 + cache	9M‡	-	92.0
Zaremba et al. (2014) – LSTM	20M	86.2	82.7
Gal & Ghahramani (2016) – Variational LSTM (MC)	20M	-	78.6
Kim et al. (2016) – CharCNN	19M	-	78.9
Merity et al. (2016) – Pointer Sentinel-LSTM	21M	72.4	70.9
Grave et al. (2016) – LSTM + continuous cache pointer†	-	-	72.1
Inan et al. (2016) – Tied Variational LSTM + augmented loss	24M	75.7	73.2
Zilly et al. (2016) – Variational RHN	23M	67.9	65.4
Zoph & Le (2016) – NAS Cell	25M	-	64.0
Melis et al. (2017) – 2-layer skip connection LSTM	24M	60.9	58.3
Merity et al. (2017) – AWD-LSTM w/o finetune	24M	60.7	58.8
Merity et al. (2017) – AWD-LSTM	24M	60.0	57.3
Ours – AWD-LSTM-MoS w/o finetune	22M	58.08	55.97
Ours – AWD-LSTM-MoS	22M	**56.54**	**54.44**
Merity et al. (2017) – AWD-LSTM + continuous cache pointer†	24M	53.9	52.8
Krause et al. (2017) – AWD-LSTM + dynamic evaluation†	24M	51.6	51.1
Ours – AWD-LSTM-MoS + dynamic evaluation†	22M	**48.33**	**47.69**

Table 1: Single model perplexity on validation and test sets on Penn Treebank. Baseline results are obtained from Merity et al. (2017) and Krause et al. (2017). † indicates using dynamic evaluation.

- the network size of MoS is adjusted to ensure a comparable number of parameters.

Experiments: Language Modeling

Model	#Param	Validation	Test
Inan et al. (2016) – Variational LSTM + augmented loss	28M	91.5	87.0
Grave et al. (2016) – LSTM + continuous cache pointer†	-	-	68.9
Melis et al. (2017) – 2-layer skip connection LSTM	24M	69.1	65.9
Merity et al. (2017) – AWD-LSTM w/o finetune	33M	69.1	66.0
Merity et al. (2017) – AWD-LSTM	33M	68.6	65.8
Ours – AWD-LSTM-MoS w/o finetune	35M	66.01	63.33
Ours – AWD-LSTM-MoS	35M	**63.88**	**61.45**
Merity et al. (2017) – AWD-LSTM + continuous cache pointer †	33M	53.8	52.0
Krause et al. (2017) – AWD-LSTM + dynamic evaluation†	33M	46.4	44.3
Ours – AWD-LSTM-MoS + dynamical evaluation†	35M	**42.41**	**40.68**

Table 2: Single model perplexity over WikiText-2. Baseline results are obtained from Merity et al. (2017) and Krause et al. (2017). † indicates using dynamic evaluation.

Model	#Param	Train	Validation	Test
Softmax	119M	41.47	43.86	42.77
MoS	113M	**36.39**	**38.01**	**37.10**

Table 3: Perplexity comparison on 1B word dataset. Train perplexity is the average of the last 4,000 updates.

Experiments: Dialog System

- Dialog: also context-dependent
- A seq2seq model with MoS added to the decoder RNN.

Model	Perplexity	BLEU-1		BLEU-2		BLEU-3		BLEU-4	
		prec	recall	prec	recall	prec	recall	prec	recall
Seq2Seq-Softmax	34.657	0.249	0.188	0.193	0.151	0.168	0.133	0.141	0.111
Seq2Seq-MoC	33.291	0.259	0.198	0.202	0.159	0.176	0.140	0.148	0.117
Seq2Seq-MoS	**32.727**	**0.272**	**0.206**	**0.213**	**0.166**	**0.185**	**0.146**	**0.157**	**0.123**

Table 4: Evaluation scores on Switchboard.

Verify the Role of Rank

- With tokens $\mathbf{X} = \{X_1, \ldots, X_T\}$, compute

  $$\{\log P(X_i \mid X_{<i}) \in \mathbb{R}^M\}_{t=1}^T$$ for each token
- Stack all T log-probability vectors into a T X M matrix,

Model	Validation	Test
Softmax	400	400
MoC	280	280
MoS	**9981**	**9981**

Table 6: Rank comparison on PTB. To ensure comparable model sizes, the embedding sizes of Softmax, MoC and MoS are 400, 280, 280 respectively. The vocabulary size, i.e., M, is 10,000 for all models.

#Softmax	Rank	Perplexity
3	6467	58.62
5	8930	57.36
10	9973	56.33
15	9981	55.97
20	9981	56.17

Table 7: Empirical rank and test perplexity on PTB with different number of Softmaxes.

Merit and Limitation

- Merit: Not only structural modification, but also theoretical support.
- Limitation: No strict proof that natural language is high-rank. But empirical evaluation can support the hypothesis.
- Inspiration: introduce non-linear transformation in the Transformer?

Proof 1

Proof of Property 1

Proof. For any $\mathbf{A}' \in F(\mathbf{A})$, let $P_{\mathbf{A}'}(X|C)$ denote the distribution defined by applying Softmax on the logits given by $\mathbf{A}'$. Consider row i column j, by definition any entry in $\mathbf{A}'$ can be expressed as $A'_{ij} = A_{ij} + \Lambda_{ii}$. It follows

$$P_{\mathbf{A}'}(x_j|c_i) = \frac{\exp A'_{ij}}{\sum_k \exp A'_{ik}} = \frac{\exp(A_{ij} + \Lambda_{ii})}{\sum_k \exp(A_{ik} + \Lambda_{ii})} = \frac{\exp A_{ij}}{\sum_k \exp A_{ik}} = P^*(x_j|c_i)$$

For any $\mathbf{A}'' \in \{\mathbf{A}'' \mid \text{Softmax}(\mathbf{A}'') = P^*\}$, for any i and j, we have

$$P_{\mathbf{A}''}(x_j|c_i) = P_{\mathbf{A}}(x_j|c_i)$$

It follows that for any i, j, and k,

$$\frac{P_{\mathbf{A}''}(x_j|c_i)}{P_{\mathbf{A}''}(x_k|c_i)} = \frac{\exp A''_{ij}}{\exp A''_{ik}} = \frac{\exp A_{ij}}{\exp A_{ik}} = \frac{P_{\mathbf{A}}(x_j|c_i)}{P_{\mathbf{A}}(x_k|c_i)}$$

As a result,

$$A''_{ij} - A_{ij} = A''_{ik} - A_{ik}$$

This means each row in $\mathbf{A}''$ can be obtained by adding a real number to the corresponding row in $\mathbf{A}$. Therefore, there exists a diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{N \times N}$ such that

$$\mathbf{A}'' = \mathbf{A} + \mathbf{\Lambda}\mathbf{J}_{N,M}$$

It follows that $\mathbf{A}'' \in F(\mathbf{A})$. $\square$

Proof 2

Proof of Property 2

Proof. For any $\mathbf{A}_1$ and $\mathbf{A}_2$ in $F(\mathbf{A})$, by definition we have $\mathbf{A}_1 = \mathbf{A} + \mathbf{\Lambda}_1 \mathbf{J}_{N,M}$, and $\mathbf{A}_2 = \mathbf{A} + \mathbf{\Lambda}_2 \mathbf{J}_{N,M}$ where $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are two diagonal matrices. It can be rewritten as

$$\mathbf{A}_1 = \mathbf{A}_2 + (\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2)\mathbf{J}_{N,M}$$

Let S be a maximum set of linearly independent rows in $\mathbf{A}_2$. Let $\mathbf{e}_N$ be an all-ones vector with dimension N. The i-th row vector $\mathbf{a}_{1,i}$ in $\mathbf{A}_1$ can be written as

$$\mathbf{a}_{1,i} = \mathbf{a}_{2,i} + (\Lambda_{1,ii} - \Lambda_{2,ii})\mathbf{e}_N$$

Because $\mathbf{a}_{2,i}$ is a linear combination of vectors in S, $\mathbf{a}_{1,i}$ is a linear combination of vectors in $S \cup \{\mathbf{e}_N\}$. It follows that

$$\mathrm{rank}(\mathbf{A}_1) \leq \mathrm{rank}(\mathbf{A}_2) + 1$$

Similarly, we can derive

$$\mathrm{rank}(\mathbf{A}_2) \leq \mathrm{rank}(\mathbf{A}_1) + 1$$

Therefore,

$$|\mathrm{rank}(\mathbf{A}_1) - \mathrm{rank}(\mathbf{A}_2)| \leq 1$$

$\square$