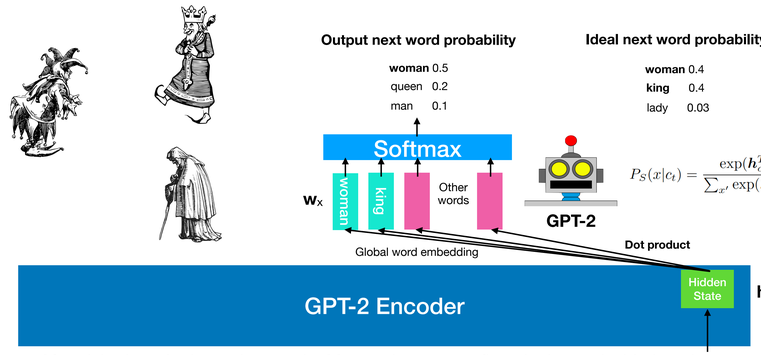
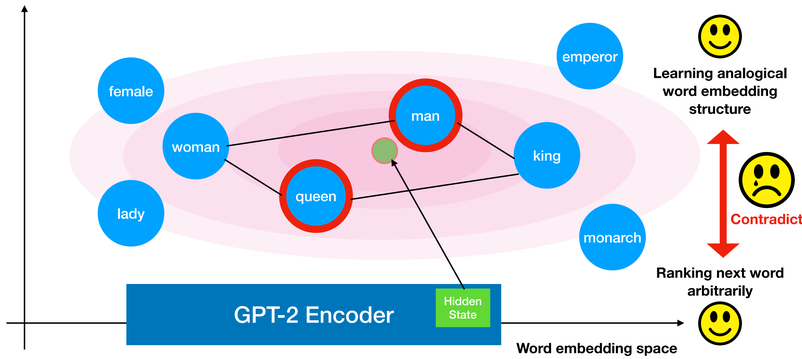
Softmax Bottleneck Makes Language Models Unable to Represent Multi-mode Word Distributions

Haw-Shiuan Chang and Andrew McCallum

Introduction



c_t After debating whether to bow to the **king** or the **woman** first, the jester decided on the



Architecture Enhancement

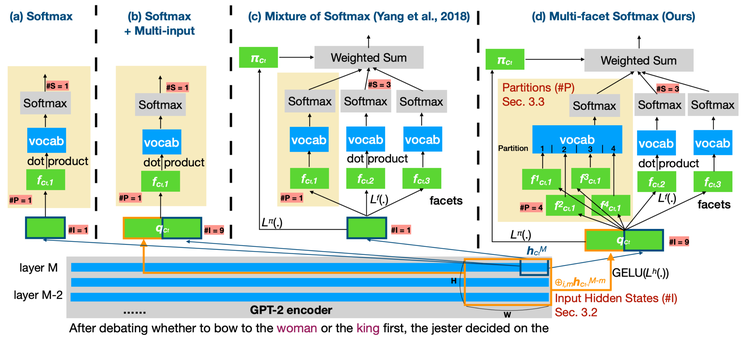


Figure 2: Comparison between different architectures. The #S, #I, and #P are the number of softmaxes, input hidden states, and partitions, respectively. The green boxes refer to embeddings/vectors. The vocab means the embeddings of all words in the vocabulary. $\oplus$ refers to concatenation. L^h, L^f, and L^π are linear projection layers.

Theoretical Analysis

Theorem 1 (simplified): If many word embeddings are linearly dependent, the softmax in a LM cannot rank the words arbitrarily

Example: If "*woman* - *man* = *queen* - *king*", GPT-2 cannot rank the word *woman* and *king* as the top 2 words

Example: If "*UMass* = 0.2 *University* + 0.2 *Massachusetts*", GPT-2 cannot rank a rare word *UMass* on top of the similar popular words *University* and *Massachusetts* (Demeter et al., 2020).

Linear Algebra Intuition: N+1 words are linear dependent ➡

They are in subspace with d < N ➡ cannot have arbitrary probabilities

- 1 w_{king} - 1 w_{queen} = 1 w_{man} - 1 w_{woman}
- 1 w_{king} + 1 w_{woman} = 1 w_{queen} + 1 w_{man}

  × h^T (hidden state) on both side

  Large Large Small Small
- 1 $h^T w_{king}$ + 1 $h^T w_{woman}$ = 1 $h^T w_{queen}$ + 1 $h^T w_{man}$
- If ∃h, s.t min($h^T w_{king}$, $h^T w_{woman}$) > max($h^T w_{queen}$, $h^T w_{man}$)

  Large Large
  - 1 $h^T w_{king}$ + 1 $h^T w_{woman}$ ≥ 2 min($h^T w_{king}$, $h^T w_{woman}$) > 2 max($h^T w_{queen}$, $h^T w_{man}$) ≥ 1 $h^T w_{queen}$ + 1 $h^T w_{man}$ (→ ←)

  Small Small
- Thus, the logits of LM cannot rank both *king* and *woman* on top of *queen* and *man*

Generalization

- Linearly dependent among {w_{l_1},…, w_{l_k}, w_{r_1},…,w_{r_m}}
- $a_{l_1} w_{l_1} + … + a_{l_k} w_{l_k} = a_{r_1} w_{r_1} + … + a_{r_m} w_{r_m}$
  - All coefficient $a_l > 0$, $a_r > 0$
  - WLOG $a_{l_1} + … + a_{l_k} ≥ a_{r_1} + … + a_{r_m}$
- $a_{l_1} h^T w_{l_1} + … + a_{l_k} h^T w_{l_k} = a_{r_1} h^T w_{r_1} + … + a_{r_m} h^T w_{r_m}$
- If ∃h, s.t min$_i$($h^T w_{l_i}$) > max$_j$($h^T w_{r_j}$)
  - $a_{l_1} h^T w_{l_1} + … + a_{l_k} h^T w_{l_k}$ ≥ ($a_{l_1} + … + a_{l_k}$)min$_i$($h^T w_{l_i}$) > ($a_{r_1} + … + a_{r_m}$)max$_j$($h^T w_{r_j}$) ≥ $a_{r_1} h^T w_{r_1} + … + a_{r_m} h^T w_{r_m}$ (→ ←)
- Thus, the logits of LM cannot rank all the left words on top of the right words.

- Improvements over Yang et al., (2018)
  - Serious among which words? -> Among words in a small subspace
  - Affect the top words? If yes, when? -> Yes. When the ideal distribution is multi-mode
  - Disappears after making D>V? -> No, if some words are in a small subspace

Theorem 2 (simplified): If many word embeddings are approximately linearly dependent and the magnitude of the hidden state has a upperbound, the softmax in a LM cannot assign very small probabilities to some words

Example: If "*woman* + *king* = *queen* + *man* + *ε*", GPT-2 cannot make the logits of *queen* and *man* much smaller than the logits of *king* and *woman*

Example: If "*woman* = *man* + *ε*", GPT-2 cannot make the logits of *man* much smaller than the logits of *woman*

Intuition: h^T *king* + h^T *woman* = h^T *queen* + h^T *man* + h^T *ε*, and we can ignore h^T *ε* if ||h|| and ||*ε*|| are both small

Empirical Analysis

Only adding nonlinearity is not enough (Parthiban et al., 2021)

Models ↓	Configuration #S	#I	#P	GPT-2 Small Size	Time	OWT	Wiki	GPT-2 Medium Size	Time	OWT	Wiki
Softmax (GPT-2)	1	1	1	163.6M	84ms	18.72	24.06	407.3M	212ms	15.89	20.34
SigSoftmax (Kanai et al., 2018)	1	1	1	163.6M	91ms	18.63	24.06	407.3M	221ms	16.07	20.65
Softmax + Multi-input	1	9	1	169.5M	87ms	18.50	23.89	417.8M	219ms	15.76	20.29
Softmax + Multi-partition	1	1	4	165.4M	88ms	18.77	24.08	410.5M	218ms	15.89	20.30
MoS (Yang et al., 2018) (4)	4	1	1	165.4M	152ms	18.61	23.77	410.5M	299ms	15.75	20.08
MoS (Yang et al., 2018) (3)	3	1	1	164.8M	130ms	18.63	23.81	409.4M	270ms	15.79	20.11
DOC (Takase et al., 2018)	3	3	1	164.8M	130ms	18.69	24.02	409.4M	270ms	15.88	20.34
MFS w/o Multi-partition	3	9	1	171.9M	133ms	18.37	23.56	422.0M	276ms	15.65	20.06
MFS w/o Multi-input	3	1	4	166.6M	134ms	18.60	23.72	412.6M	275ms	15.71	20.08
MFS (Ours)	3	9	4	175.4M	138ms	**18.29**	**23.45**	428.3M	283ms	**15.64**	**20.02**

Multiple input hidden states help

Multiple partitions help

Table 1: Perplexity comparison between MFS (Ours) and baselines. #S, #I, #P are the number of softmaxes (i.e., K), input hidden states, and partitions, respectively. The top four baselines use a single softmax. OWT and Wiki are the test set perplexity of OpenWebText and Wikipedia 2021, respectively. The standard errors of all models are smaller than 0.02 perplexity. We also compare the number of parameters and the inference time on one batch.

Improvement of MFS over Softmax is around 15% between GPT-2 Small and GPT-2 Medium (with 3x parameters)

Corpus →	OpenWebText	Wikipedia 2021	Analogy in Templates (Section 5)
Input Context	... The Elastic Endpoint Security and Elastic SIEM solutions mentioned in this post are now referred to as **Elastic**	... law and chance working together cannot generate CSI, either. Moreover, he claims that **CSI**	I went to Paris and Germany before, and I love one of the places more, which is **Germany**
Softmax (GPT-2)	the 0.087, E 0.043, End 0.039	the 0.174, this 0.054, if 0.038	Paris 0.893, France 0.045, **Germany** 0.033
MFS (Ours)	**Elastic** 0.220, the 0.089, EC 0.033	CSI 0.186, the 0.140, there 0.033	Paris 0.544, **Germany** 0.389, France 0.064
MFS Softmax 1	end 0.051, the 0.043, security 0.023	the 0.191, law 0.127, if 0.053	Paris 0.979, France 0.013, **Germany** 0.007
MFS Softmax 2	**Elastic** 0.652, EC 0.080, ES 0.046	the 0.191, there 0.049, this 0.047	Paris 1.000 Berlin 0.000 ##Paris 0.000
MFS Softmax 3	the 0.193, E 0.040, a 0.014	**CSI** 0.677, law 0.029, laws 0.019	**Germany** 0.852, France 0.139, China 0.004

Worse word similarity (Non-English text in OpenWebText): 2.5x Improvement

Top 10% most diverse facets: 3x Improvement

Candidates have an analogical relation: Perplexity 2.3 -> 1.7

Conclusion and Related Work

Theory

Multi-mode distribution must exist if some word embeddings are in a small subspace

Stolen Probability (Demeter et al., 2020) + Softmax Bottleneck (Yang et al., 2018) → Generalization → Multi-mode Distribution (Ours)

Method

We propose two enhancements for mixture of softmax (MoS)

MoS (Yang et al., 2018) → Adding Nonlinearity (Kanai et al., 2018) → Improvement → MFS (Ours)

Analysis

Our proposed method improves softmax layer in GPT-2 especially when the ideal next word distribution is multi-mode

Softmax bottleneck cannot explain the improvement of MoS (Parthiban et al., 2021) → Explanation → Multi-mode is a good explanation (Ours)

References

[1] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. "Breaking the Softmax Bottleneck: A High-Rank RNN Language Model." In ICLR. 2018.
[2] David Demeter, Gregory Kimmel, and Doug Downey. Stolen probability: A structural weakness of neural language models. In ACL. 2020
[3] Sekitoshi Kanai, Yasuhiro Fujiwara, Yuki Yamanaka, and Shuichi Adachi. Sigsoftmax: Reanalysis of the softmax bottleneck. In NeurIPS 2018
[4] Dwarak Govind Parthiban, Yongyi Mao, and Diana Inkpen. On the softmax bottleneck of recurrent language models. In AAAI 2021
[5] Sho Takase, Jun Suzuki, and Masaaki Nagata. 2018. Direct output connection for a high-rank language model. In EMNLP 2018
[6] Sharan Narang, Hyung Won Chung, Yi Tay, Liam Fedus, Thibault Févry, Michael Matena, Karishma Malkan et al. "Do Transformer Modifications Transfer Across Implementations and Applications?." In *EMNLP 2021*