Adaptive Input Representations for Neural Language Modeling

Alexei Baevski & Michael Auli
Facebook AI Research,
Menlo Park, CA, USA

Accepted in ICLR 2019
Scores: 7, 7, 8

Motivation

- Neural language models with large vocabulary require heavy computation for computing the probabilities of all words.
  - 800K words in BILLION WORD
- In our transformer model, word embedding accounts for most parameters.
- Intuition: assign more parameters/capacity to frequent words and reduce the parameters for less frequent words.

Previous Work: Adaptive Softmax

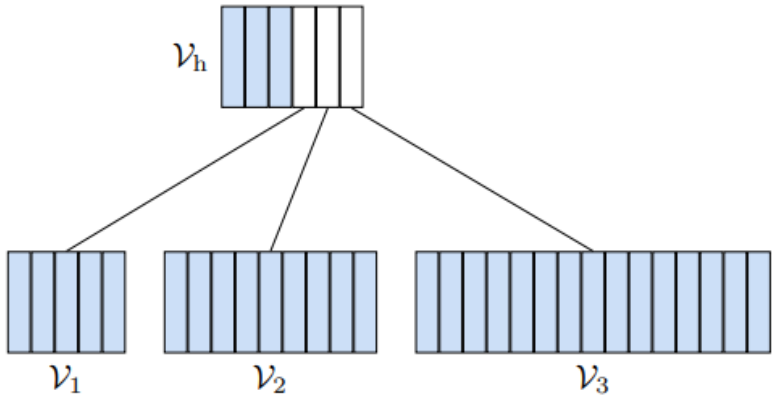

Figure 3. Our hierarchical model is organized as (i) a first level that includes both the most frequent words and vectors representing clusters, and (ii) clusters on the second level that are associated with rare words, the largest ones being associated with the less frequent words. The sizes are determined so as to minimize our computational model on GPU.

- Divide the vocabulary into several clusters according to their frequencies.
- Frequent words need high capacity to be predicted correctly (large output embedding).
- Map hidden states to different sizes of classifiers.

Grave et al., *Efficient softmax approximation for GPUs, in ICML 2017.*

This Paper: Adaptive Embeddings

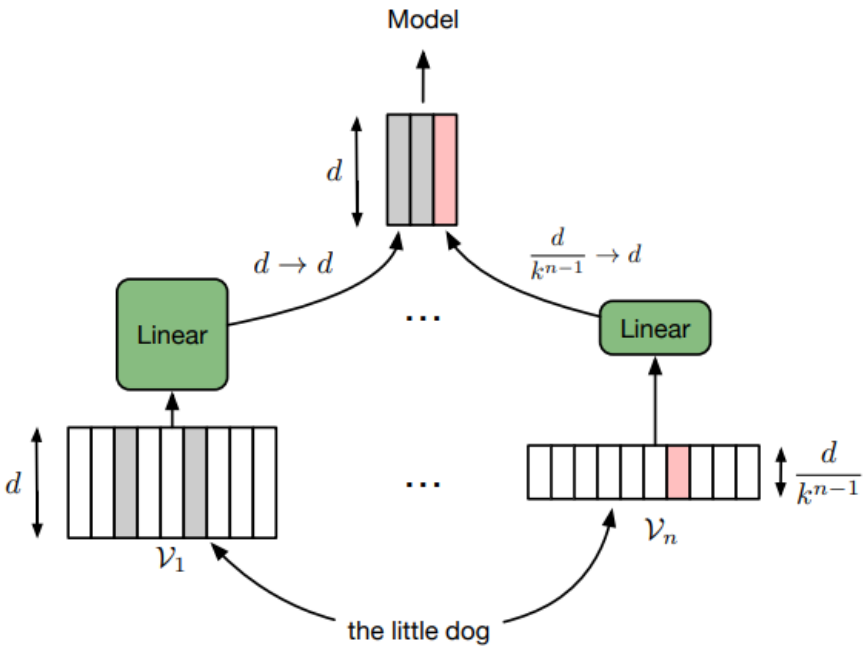

Figure 1: Illustration of adaptive input representations. Words are assigned to clusters $\mathcal{V}_i$ based on their frequency which determines the size of the representations. Embeddings are projected to a common dimension d before being fed to the model.

- Extend from output embeddings to input embeddings.
- Project embeddings to the same hidden size.
- Weight sharing: combine with adaptive softmax, sharing words and projections.

Results

	Input	Output	Valid	Test	Train Time (hours)	Params
SM	Embedding	Softmax	23.87	24.92	57*	476.8M
BPE	BPE Embedding	BPE Softmax	23.13	24.25	30	270M
BPE-T	BPE Embedding	BPE Softmax (tied)	22.46	23.45	30	235.7M
SM-T	Embedding	Softmax (tied)	22.63	23.38	56*	339.7M
ASM	Embedding	Adaptive	21.23	22.18	35	263.1M
CNN	Char-CNN	Adaptive	20.86	21.79	70	266.3M
ADP	Adaptive	Adaptive	20.95	21.74	34	291.3M
ADP-T	Adaptive	Adaptive (tied)	**19.79**	**20.51**	30	246.9M

Table 3: Test perplexity on WIKITEXT-103 for various input and output layer factorizations. Training speed was measured on a single 8-GPU machine. (*) indicates a modified training regime

- Adaptive input embeddings reduce parameters while achieving higher accuracy.
- Tied with adaptive softmax further reduce parameters by a total of 61%.

Summary

- Simple and doable idea.

- People are caring about efficiency more and more.