Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer

Noam Shazeer[1], Azalia Mirhoseini[*†1], Krzysztof Maziarz[*2], Andy Davis[1], Quoc Le[1], Geoffrey Hinton[1] and Jeff Dean[1]

[1]Google Brain, {noam,azalia,andydavis,qvl,geoffhinton,jeff}@google.com
[2]Jagiellonian University, Cracow, krzysztof.maziarz@student.uj.edu.pl

*on ICLR 2017

Motivation

- Neural Networks: more capacity(parameters), better performance.
- Increase model capacity -> increase computation costs.
- Conditional computation: only parts of the nets are active for each example.
- Significant algorithmic and performance challenges (GPUs, bandwidth, …).
- This paper: algorithmic and engineering solutions to conditional computation in deep nets: 1000X improvements in model capacity with minor losses in computational efficiency on GPU clusters.

Method

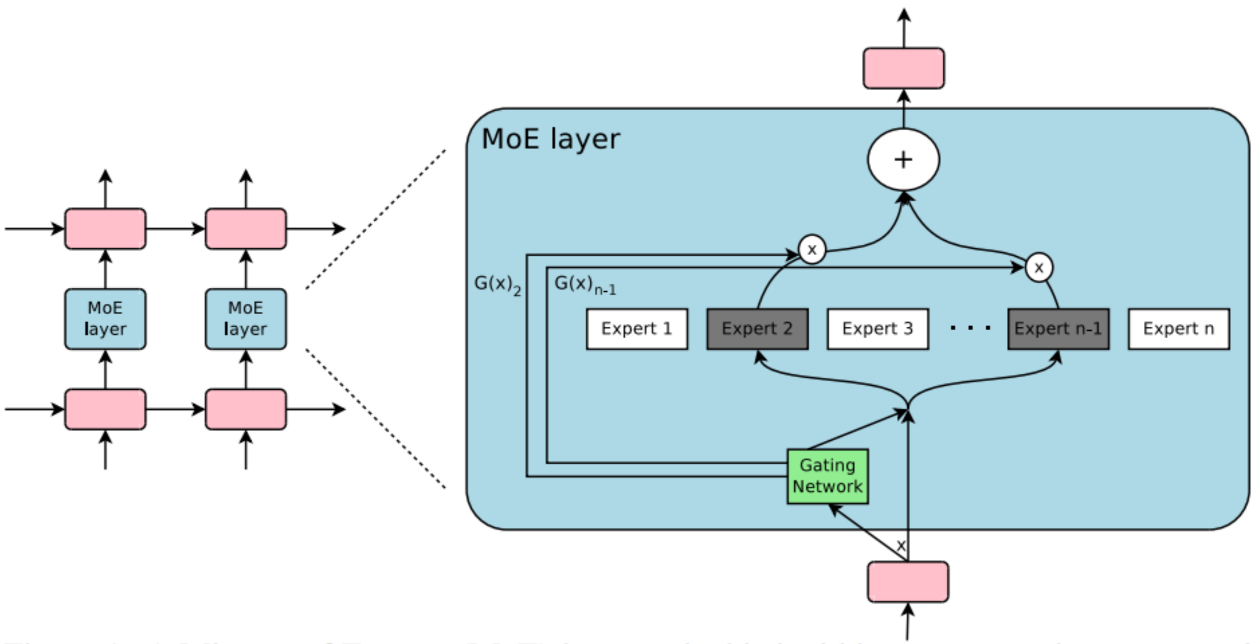


Figure 1: A Mixture of Experts (MoE) layer embedded within a recurrent language model. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network.

- Sparsely-Gated Mixture-of-Experts Layer (MoE): a number of simple FFN (thousands).
- Gating Net: select a sparse&different combination of the experts to process each input.

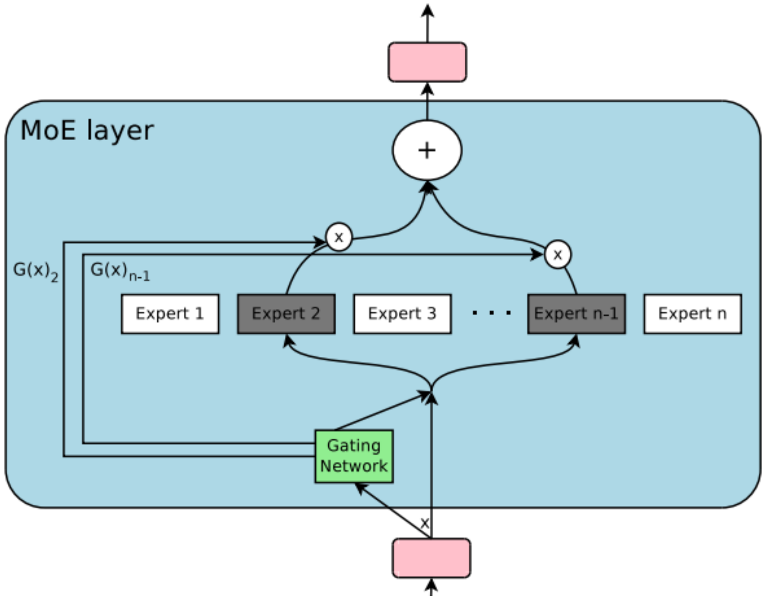
The different experts tend to become highly specialized based on syntax and semantics.

Method

- Output of MoE:

$$y = \sum_{i=1}^{n} G(x)_i E_i(x)$$

- Sparse: when G(x)=0, no need to compute E(x)



- Softmax Gating: $G_\sigma(x) = Softmax(x \cdot W_g)$

- Noisy Top-K Gating (K=4):

$$G(x) = Softmax(KeepTopK(H(x), k))$$

$$H(x)_i = (x \cdot W_g)_i + StandardNormal() \cdot Softplus((x \cdot W_{noise})_i)$$

$$KeepTopK(v, k)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ elements of } v. \\ -\infty & \text{otherwise.} \end{cases}$$

Method

- Engineering solution in distributed computing: batch size, network bandwidth.
  - Batch size for each expert: $\frac{k*b}{n} \ll b$, use model parallelism to increase.
  - Increase FFN hidden layer size to increase computational efficiency.

Method

- Balancing expert utilization: gates tend to select the same few experts.
- Define the *Importance* of an expert and add another loss to encourage all experts to have equal importance.

$$Importance(X) = \sum_{x \in X} G(x)$$

batchwise sum of the gate values for that expert.

$$L_{importance}(X) = w_{importance} \cdot CV(Importance(X))^2$$

square of the coefficient of variation of the set of importance values

Results

- 1 BILLION WORD LANGUAGE MODELING:

Table 1: Summary of high-capacity MoE-augmented models with varying computational budgets, vs. best previously published results (Jozefowicz et al., 2016). Details in Appendix C.

	Test Perplexity 10 epochs	Test Perplexity 100 epochs	#Parameters excluding embedding and softmax layers	ops/timestep	Training Time 10 epochs	TFLOPS /GPU
Best Published Results	34.7	30.6	151 million	151 million	59 hours, 32 k40s	1.09
Low-Budget MoE Model	34.1		4303 million	8.9 million	15 hours, 16 k40s	0.74
Medium-Budget MoE Model	31.3		4313 million	33.8 million	17 hours, 32 k40s	1.22
High-Budget MoE Model	**28.0**		4371 million	142.7 million	47 hours, 32 k40s	**1.56**

Model Capacity

Computational Efficiency

Results

- Machine Translation:

Table 2: Results on WMT'14 En→ Fr newstest2014 (bold values represent best results).

Model	Test Perplexity	Test BLEU	ops/timenstep	Total #Parameters	Training Time
MoE with 2048 Experts	2.69	40.35	85M	8.7B	3 days/64 k40s
MoE with 2048 Experts (longer training)	**2.63**	**40.56**	85M	8.7B	6 days/64 k40s
GNMT (Wu et al., 2016)	2.79	39.22	214M	278M	6 days/96 k80s
GNMT+RL (Wu et al., 2016)	2.96	39.92	214M	278M	6 days/96 k80s
PBMT (Durrani et al., 2014)		37.0			
LSTM (6-layer) (Luong et al., 2015b)		31.5			
LSTM (6-layer+PosUnk) (Luong et al., 2015b)		33.1			
DeepAtt (Zhou et al., 2016)		37.7			
DeepAtt+PosUnk (Zhou et al., 2016)		39.2			

Table 3: Results on WMT'14 En → De newstest2014 (bold values represent best results).

Model	Test Perplexity	Test BLEU	ops/timestep	Total #Parameters	Training Time
MoE with 2048 Experts	**4.64**	**26.03**	85M	8.7B	1 day/64 k40s
GNMT (Wu et al., 2016)	5.25	24.91	214M	278M	1 day/96 k80s
GNMT +RL (Wu et al., 2016)	8.08	24.66	214M	278M	1 day/96 k80s
PBMT (Durrani et al., 2014)		20.7			
DeepAtt (Zhou et al., 2016)		20.6			

Table 4: Results on the Google Production En→ Fr dataset (bold values represent best results).

Model	Eval Perplexity	Eval BLEU	Test Perplexity	Test BLEU	ops/timestep	Total #Parameters	Training Time
MoE with 2048 Experts	**2.60**	**37.27**	**2.69**	**36.57**	85M	8.7B	1 day/64 k40s
GNMT (Wu et al., 2016)	2.78	35.80	2.87	35.56	214M	278M	6 days/96 k80s

Summarization

- Marks in ICLR 2017: 6, 7, 7.
- Meta Comments: The paper uses mixtures of experts to increase the capacity of deep networks, and describes the implementation of such a model on a cluster of GPUs. The proposed mixture model achieves strong performances in language modeling and machine translation.
- Inspiration: gating mechanism to fuse information.