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] Jeff Dean[1]

[1]Google Brain [2]Jagiellonian University, Cracow

ICLR 2017

Background

- Practices in various domains including text, images, and audio have shown that with sufficiently large datasets, increasing the capacity (number of parameters) of neural networks can give much better prediction accuracy.
- For typical deep learning models, where the entire model is activated for every example, the training costs are quadratic, as both the model size and the number of training examples increase.

Motivation

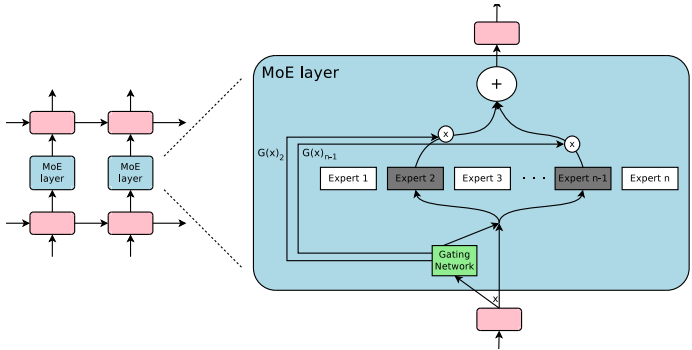
- Various forms of conditional computation have been proposed as a way to increase model capacity without a proportional increase in computational costs.
- In these schemes, large parts of a network are active or inactive on a per-example basis.
- However, While these ideas are promising in theory, no work to date has yet demonstrated massive improvements in model capacity, training time, or model quality.

Challenges

- GPUs are much faster at arithmetic than at branching.
- Conditional computation reduces the batch sizes for the conditionally active chunks, resulting in lower GPU utilization.
- Network bandwidth could be more limiting than computation power.
- Additional loss terms may be necessary to achieve the desired level of sparsity.
- Model capacity is most critical for very large datasets. Previous studies use datasets up to 600,000 images, which may not be able to demonstrate the necessity of huge models.

Sparsely-Gated Mixture-of-Experts (MoE) Layers

A new type of general purpose neural network componenet, Sparsely-Gated Mixture-of-Experts (MoE) Layer, which consists of a number of experts, each a simple feed-forward neural network, and a trainable gating network.



Sparsely-Gated Mixture-of-Experts (MoE) Layers

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

$G(x)$ is the gating network. Wherever $G(x)_i = 0$, the computation of $E_i(x)$ can be skipped.
$E_i(x)$ is the output of the i-th expert.

Gating Network

- A simple choice (Jordan et al., 1994) is $G(x) = Softmax(x \cdot W_g)$
- However, to achieve sparsity, a new Noisy Top-K Gating is introduced.

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

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

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

- The gating network can be trained with back-propagation.

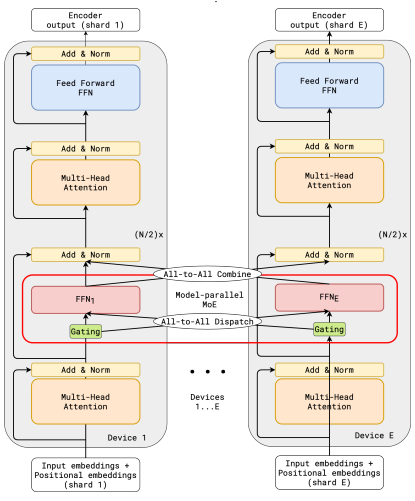
The Shrinking Batch Problem

Large batch sizes are necessary for computation efficiency on mordern GPUs. If the gating network chooses k out of n experts, then for a batch of b examples, each experts receives approximately $\frac{kb}{n} \ll b$ examples.

The solution is to make the original batch size as large as possible. However, batch size tends to be limited by the memory necessary to store activations between the forwards and backwards passes.

Mixing Data Parallelism and Model Parallelism

- Distribute the standard layers and the gating networks according to conventional data-parallel schemes, allowing large total batch size.
- Each expert in the MoE layer receives a combined batch consisting of the relevant examples from all of the data-parallel input batches.



Taking Advantage of Convolutionality

In the language model used in the experiment, another trick to the shrinking batch size problem is to accumulate the data of all the time steps of the previous layer as a big batch before feeding into the MoE layer. This is because the MoE layer is not recurrent and is applied to all the time steps convolutionally.

Network Bandwidth

- Sending the inputs and outputs of the experts across the network takes significant time.
- To maintain computational efficiency, the hidden layer of each expert can be set to very big, or more hidden layers can be used.

Balancing Experts Utilization

- Imbalanced choice of experts wastes capacity and slows down training.
- The imbalance of gating network is self-reinforcing, as the favored experts are trained more rapidly.
- Therefore, an importance loss is introduced to encourage all experts to have equal importance.

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

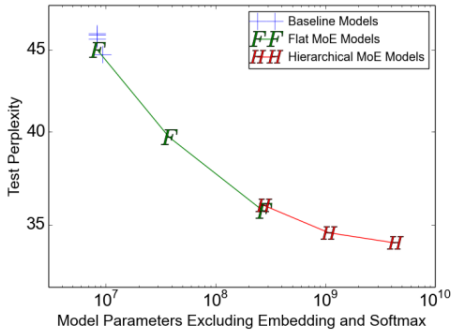
$$L_{importance}(X) = \lambda \cdot CV(Importance(X))^2$$

1 Billion Word Language Modeling Benchmark

- Dataset: shuffled sentences that have approximately 829 million words with a vocabulary of 793,471 words.
- Baseline: one or more stacked LSTM layers.
- MoE model: two stacked LSTM with a MoE layer between. 4 experts were active per input.

Low Computation, Varied Capacity

The first experiment shows the effect of adding capacity while keeping the computation cost roughly the same.



Varied Computation, High Capacity

The next experiment alters the number of experts and the width of each expert, to adjust the computation cost while keeping the capacity the same.

	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**

Machine Translation

The MoE model is modified from the GNMT model. It shows better performance in the WMT'14 English to Franch translation task with lower computation cost.

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			

Multilingual Machine Translation

	GNMT-Mono	GNMT-Multi	MoE-Multi	MoE-Multi vs. GNMT-Multi
Parameters	278M / model	278M	8.7B	
ops/timestep	212M	212M	102M	
training time, hardware	various	21 days, 96 k20s	**12 days, 64 k40s**	
Perplexity (dev)		4.14	**3.35**	-19%
French → English Test BLEU	36.47	34.40	**37.46**	+3.06
German → English Test BLEU	31.77	31.17	**34.80**	+3.63
Japanese → English Test BLEU	23.41	21.62	**25.91**	+4.29
Korean → English Test BLEU	25.42	22.87	**28.71**	+5.84
Portuguese → English Test BLEU	44.40	42.53	**46.13**	+3.60
Spanish → English Test BLEU	38.00	36.04	**39.39**	+3.35
English → French Test BLEU	35.37	34.00	**36.59**	+2.59
English → German Test BLEU	**26.43**	23.15	24.53	+1.38
English → Japanese Test BLEU	**23.66**	21.10	22.78	+1.68
English → Korean Test BLEU	**19.75**	18.41	16.62	-1.79
English → Portuguese Test BLEU	**38.40**	37.35	37.90	+0.55
English → Spanish Test BLEU	34.50	34.25	**36.21**	+1.96

Conclusion

The paper presented a conditional computation layer and showed its power in increasing the model capacity without quadratic computation cost. It also introduces new engineering challenges like shrinking batchsize problem and expert imbalance. There are follow up papers like GShard in ICLR 2021 that further investigate these problems. We can also think about them and try to find better solutions.

Thank you!