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

Presenter: Shijia Wang

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

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

ICLR, 2017

Outline

Outline

1. Introduction
   - Conditional Computation
2. The Mixture-of-Experts Layer (MoE)
   - Approach
   - Structure
   - Performance Challenges
   - Balancing Expert Utilization
3. Conclusion
   - Experiments

Problem

- When datasets are large, increasing the number of parameters of neural networks can give much better prediction accuracy.
- Roughly quadratic growth in training cost as both the model size and the number of training examples increase

Previous Solutions

- *Conditional computation schemes* - parts of a network are used depending on the example
- Gating decisions could be binary or sparse and continuous, stochastic or deterministic
- Various forms of reinforcement learning and back-propagation for training the gating decisions
- None has demonstrated massive improvements

Challenges

- Most computing devices are much faster at arithmetic than branching
- Conditional computing reduces the batch sizes due to the conditionally active chunk
- Network bandwidth speed is slower than computation speed
- Loose information to achieve the desired level of sparsity.
- Small model capacity for acceptable datasets

Outline

The Mixture-of-Experts Layer

- Consists of a number of experts, each a simple feed-forward neural network
- A trainable gating network which selects a combination of the experts to process each input
- All parts are trained jointly by back-propagation

MoE Diagram

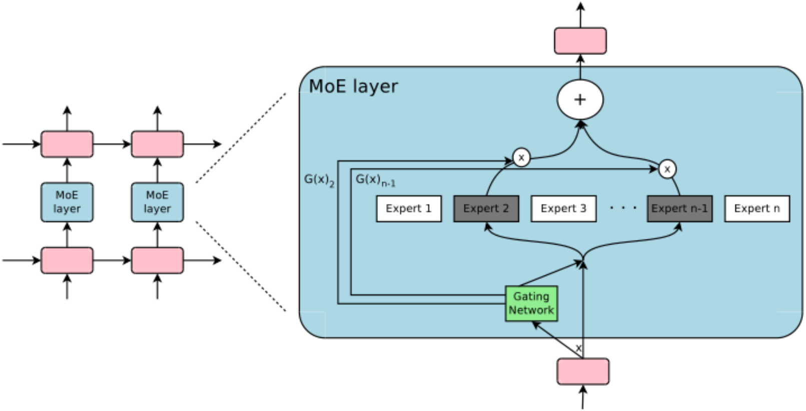

Figure 1: A Mixture of Experts (MoE) layer embedded within a recurrent language model. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network.

Outline

1. Introduction
   - Conditional Computation
2. The Mixture-of-Experts Layer (MoE)
   - Approach
   - Structure
   - Performance Challenges
   - Balancing Expert Utilization
3. Conclusion
   - Experiments

Consituents

- A set of n expert networks $E_1, \ldots, E_n$
- A gating network G that outputs a sparse n-dimensional vector

Output of the MoE Module

- Given input x
- $G(x)$ the output of the gating network
- $E_i(x)$ the output of the i-th expert
- The output y is:

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

- Whenever $G(x)_i = 0$, the $E_i(x)$ does not need to be calculated

Large Number of Experts

- Reduce branching factor by creating a two-level hierarchy of experts
- Each expert itself is a MoE

Softmax Gating

- Gating function with trainable weight matrix W_g:

$$G_\sigma(x) = Softmax(x * W_g) \tag{2}$$

Noisy Top-K Gating

- Add sparsity and noise to the Softmax gating network:

$$G(x) = Softmax(KeepTopK(H(x), k)) \tag{3}$$

$$H(x)_i = (x * W_g)_i + StandardNormal() * Softplus((x * W_{noise})_i) \tag{4}$$

$$KeepTopK(v, k)_i = \begin{cases} v_i, & \text{if } v_i \text{ is in the top } k \text{ elements of } v. \\ -\infty, & \text{otherwise.} \end{cases} \tag{5}$$

Training the Gating Network

- Back-propagation with the rest of the model
- The gate values of the top experts have nonzero derivatives with respect to the weights of the gating network
- Gradients also back-propagate through the gating network to its inputs

Outline

Shrinking Batch Problem

- Want large batch sizes
- If the gating chooses k out of n experts for a batch of b examples, each expert receives a batch of approximately $kb/n \ll b$ examples
- Extremely large batch sizes limited by memory

Mixing Data Parallelism and Model Parallelism

- Solution: run the standard layers in parallel with different batches of data
- Feed into only 1 shared MoE layer
- Each expert receives a combined batch from all the parallel inputs
- If there are d parallel devices, each expert receives kbd/n examples

Taking Advantage of Convolutionality

- Solution: wait until all timesteps of the previous layer finish
- Experts receive a big batch from all the timesteps

Increasing Batch Size for a Recurrent MoE

- Solution: wait until all timesteps of the previous layer finish
- Experts receive a big batch from all the timesteps

Network Bandwidth

- Problem: overhead cost for communcating inputs and outputs
- Use a larger hidden layer or more hidden layers within memory limit

Outline

Favors Few Experts

- Could converge to a state that favors a few experts
- Favored experts train more rapidly and are selected even more

Soft Constraint

- Batch-wise sum of the gate values for an expert:

$$Importance(X) = \sum_{x \in X} G(x) \tag{6}$$

- Loss added to the overall loss, where CV is the coefficient of variance function:

$$L_{importance}(X) = w_{importance} * CV(Importance(X))^2 \tag{7}$$

- As the gating favors a few experts, the overall loss increases

Outline

1 Billion Word Language Modeling Benchmark

- 829 million words, with a vocabulary of 793,471 words
- Flat MoEs containing 4, 32, and 256 experts
- Hierarchical MoEs containing 256, 1024, and 4096 experts
- Each expert had about 1 million parameters

1 Billion Word Language Modeling Benchmark

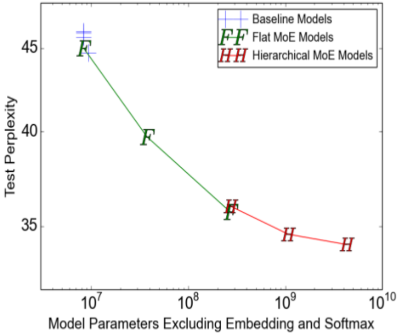

Figure 2: Model comparison on 1-Billion-Word Language-Modeling Benchmark. On the left, we plot test perplexity as a function of model capacity for models with similar computational budgets of approximately 8-million-ops-per-timestep. On the right, we plot test perplexity as a function of computational budget. The top line represents the LSTM models from (Jozefowicz et al., 2016). The bottom line represents 4-billion parameter MoE models with different computational budgets.

1 Billion Word Language Modeling Benchmark

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**

100 Billion Word Language Modeling Benchmark

- For a larger training set, high capacities would continue to produce significant quality improvements

100 Billion Word Language Modeling Benchmark

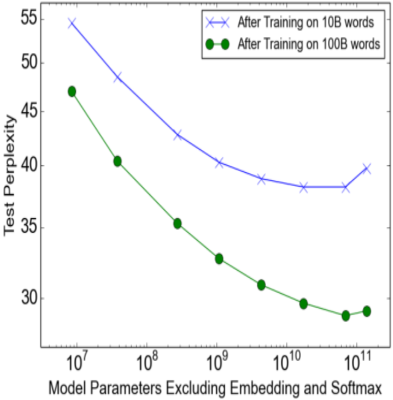

Figure 3: Language modeling on a 100 billion word corpus. Models have similar computational budgets (8 million ops/timestep).

100 Billion Word Language Modeling Benchmark

- The WMT14 EnFr with 36M sentence pairs
- The EnDe with 5M sentence pairs
- BLEU (bilingual evaluation understudy) higher is better

Machine Translation (Single Language Pair)

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			

Machine Translation (Single Language Pair)

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			

Machine Translation (Single Language Pair)

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

Multilingual Machine Translation

- About 3B sentence pairs

Multilingual Machine Translation

Table 5: Multilingual Machine Translation (bold values represent best results).

	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

Summary

- Algorithmic and engineering solution
- Focused on text experiments but can be applied for other situations