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

Motivation

One key factor that leads to the success of modern deep neural networks is the increasing amount of computational power available. It has been shown that scaling both in dataset and model capacity is crucial to boosting model performance on various tasks including computer vision, text, and audio.
In the field of NLP, the success of bigger model is especially shining. Many researches are devoted to building giant models in different domains, such as the BERT[1] on natural language understanding tasks, the GPT-2[2] model on language generation, and the Meena[3] for open-domain dialogues (2.6 billion parameters).

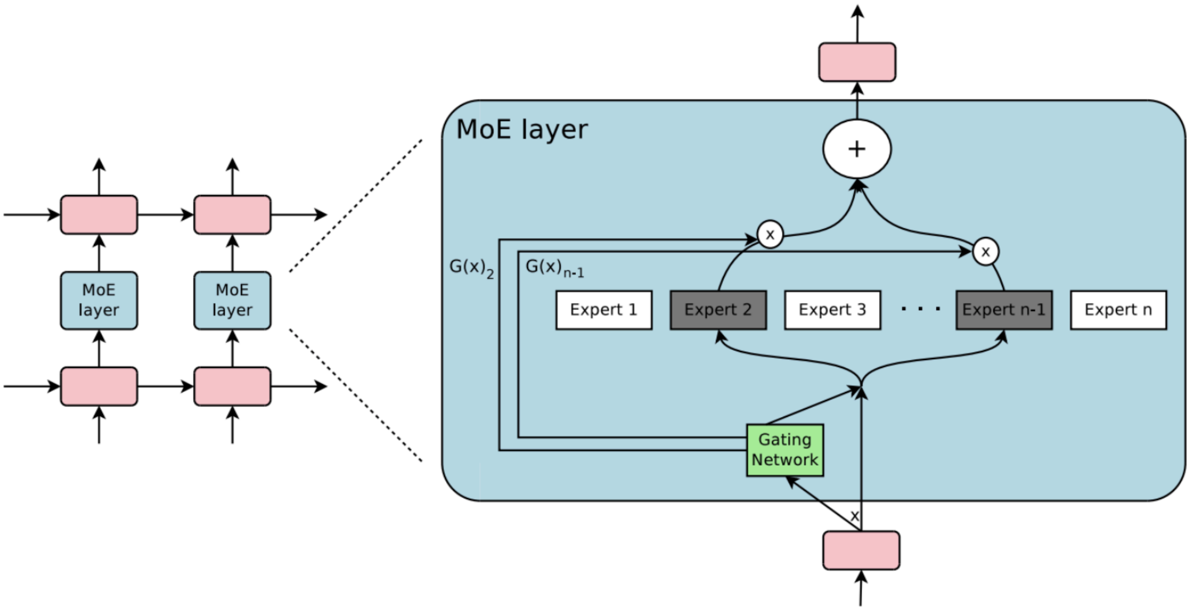
Challenges

However, as the dataset and model capacity increases, the training cost increases quadratically, making building "Outrageously Large Neural Networks" a challenging research topic. Conditional computation is one way that has been proved to increase model capacity without a proportional increase in computational costs. The idea of conditional computation is that most parts of neural network are inactive for one instance.
Previous work failed to achieve massive model capacity due to various reasons:

- Large batch size is critical to amortize the parameter transfer and update
- The network bandwidth can be a bottleneck
- Large models require a larger dataset
- Distributing the data sparsely and equally for model quality and load-balancing.

Idea of MoE

The paper from Google Brain proposes the sparsely-gated Mixture-of-Experts (MoE) as a new way for conditional computation, which solves all above challenges and gains a significant improvement on language modeling and machine translating. In their setting, each expert is a independent neural network model. Each expert is good at a specific perspective. For example some experts are responsible for semantics while others are professional at syntactics. There might be thousands of expert models. For each input sentence, only a small fraction of experts are activated. While experts are more suitable for the task is determined by the MoE layer. The results from all activated experts are then aggregated together to make a final prediction.



Mixure-of-Expert Layer

One straightforward choice for MoE layer is applying the softmax function after a linear transportation. To achieve sparsity, the Nosiy Top-K Gating is proposed. Adding a trainable Gaussian noise before applying softmax. Then, only the top-K values are reserved. The rest are all set to negative infinite. Note that the noise per component also depends on the input data, which is controlled by another trainable matrix W_noice. The noise term also helps improve load balancing according to the paper. The process is illustrated in the below table.

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

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

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

Input data	(0.8, 1.2)	(-0.3, -1.5)	(0.5, 0.3)	(0.7, -0.1)
Linear Transfer	0.8	-0.2	0.2	0.1
Get Noise	-0.2	-0.3	-0.1	0.2
Add Noice	0.6	-0.5	0.1	0.3
Top-K (K=2)	0.6	-0.5	-∞	-∞
Softmax	0.75	0.25	0	0

Address Performance Challenges

This part will introduce several design that help solve challenges mentioned before.

- As there might be thousands of experts, the whole model is too big to fit into the memory for single machine. The MoE model mixes the model parallelism and data parallelism. The standard input/output layer and MoE layer are shared on all machine (data-parallel replicas) while the expert models are splitted and stored on individual single machine (model-parallel shards). A given input batch is distributed to all machines. Each machine runs its experts on the data when necessary. And the prediction results are then aggregated.
- Network bandwidth. Using a larger hidden layer, or more hidden layers can increase computation efficiency,
- Balance expert utilization. It is observed the gating network tend to converge to a state where it always produces large weight for a small set of experts. The imbalance is "self-reinforcing", as those experts are trained faster and thus been chosen by the MoE layer more frequently. The paper designs an importance loss, which aims to ensure each expert would end up with equal importance.

Experiment Results

The paper focuses on language modelling and machine translation task, which are known can benefit from large model capacity. For language modeling, the perplexity (the lower is better) on test set for MoE is 28.0, while the best published result is 34.7. Remarkably, the training time for MoE is similar to the compared method on same amount of GPUs (47h vs 59h on 32 k40s), although MoE has a large model size (4 billion vs 0.15 billion). For machine translation, MoE outperformance 1.0~1.3 BLEU score than state-of-the-art models on various datasets. Both results are significant improvements.

Related Work

There are many related work for training large model or increasing model capacity. Transformer [4] is proposed attention-based neural networks to replace LSTM and speed up the training. The transformer, is orthogonal to MoE, as the expert can be models with any structure. [5] improves model capacity with mixture of softmaxes. [6] designs a system for efficiently implementing model parallel.

Reference

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[2] Language Models are Unsupervised Multitask Learners

[3] Towards a Human-like Open-Domain Chatbot

[4] Attention Is All You Need

[5] Breaking the Softmax Bottleneck: A High-Rank RNN Language Model

[6] Mesh-TensorFlow: Deep Learning for Supercomputers