Introduction to Mixture of Experts (MoE)

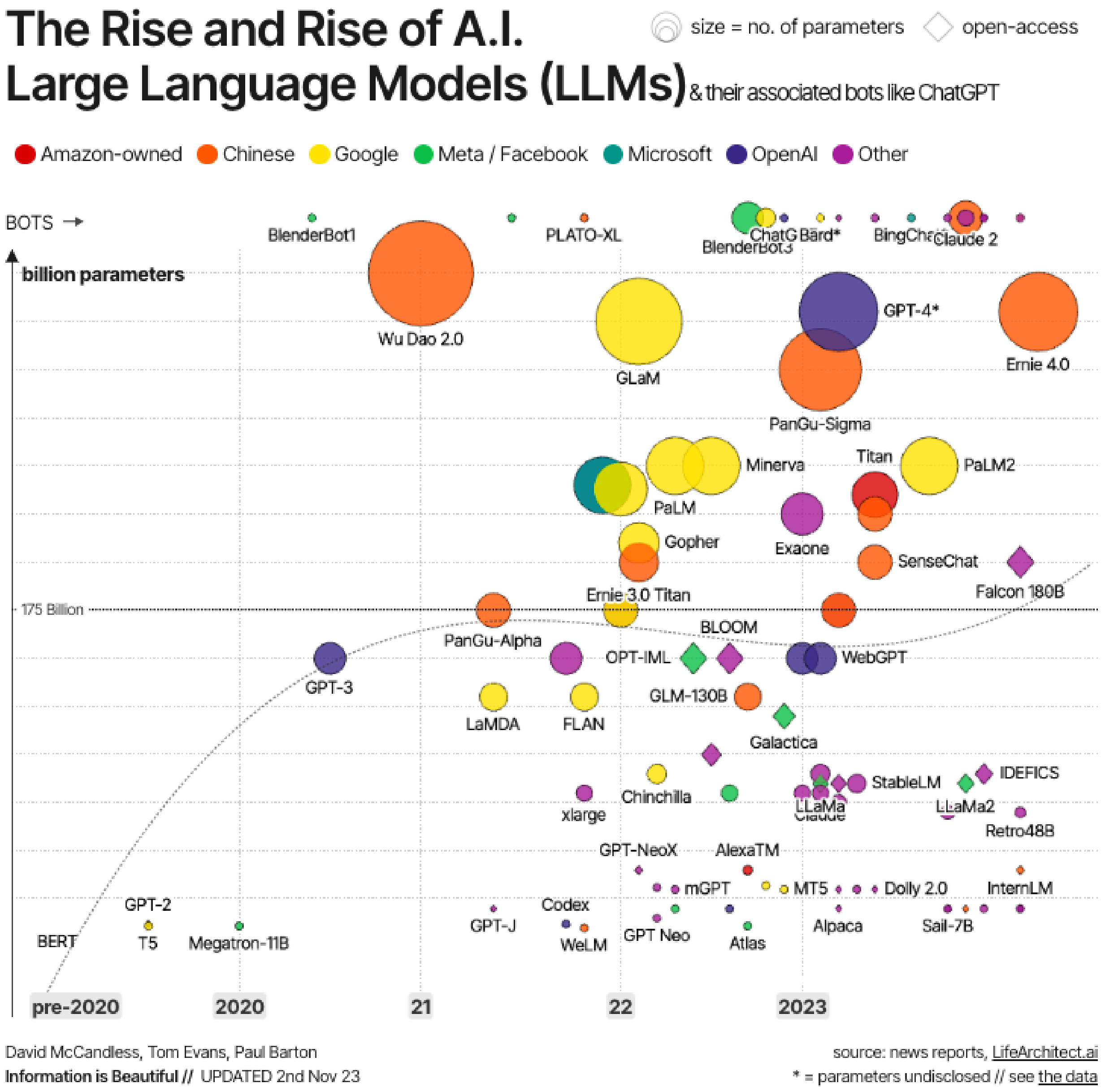
Presenter:
Yonsei Univ. PhD Student, NAVER Cloud (intern)
Jin-Duk Park

AX Writer Seminar Material

2024.04.04

Introduction

- LLMs are truly large
- LLMs require large computing demands (e.g., training/inference time)



*GPT-4 example

You
네이버 주가에 대해 설명해줘

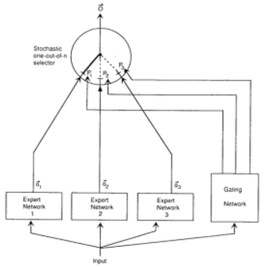
ChatGPT
네이버(Naver)의 주가는 최근 확인된 바에 따르면 한국 거래소에서 193,400 KRW로 마감했습니다. 이는 이전 거래일 대비 1,400 KRW (0.73%) 상승한 수치입니다. 52주 동안의 최저가는 178,000 KRW, 최고가는 241,500 KRW였으며 ● …

Introduction

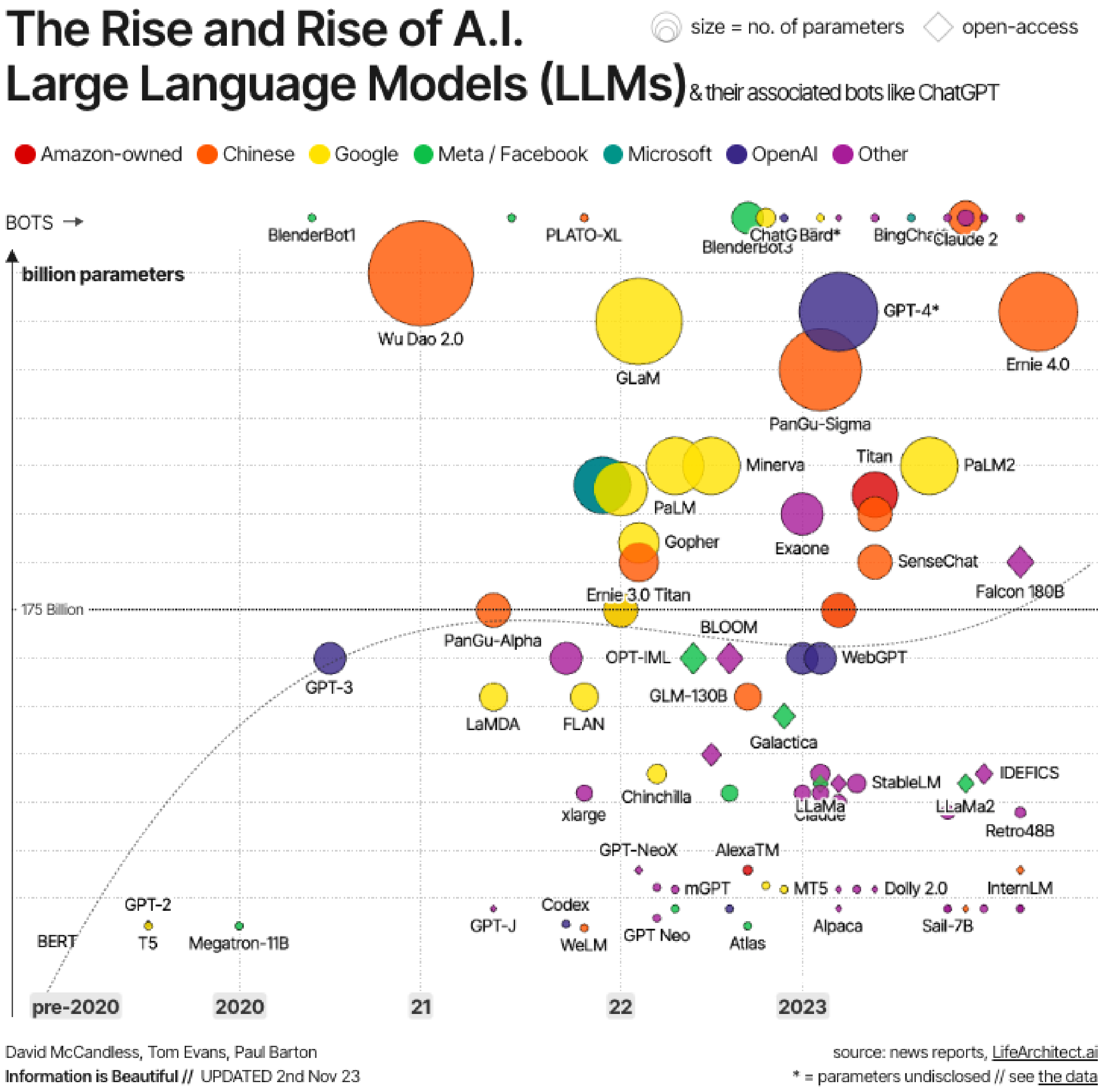
- LLMs are truly large
- LLMs require large computing demands (e.g., training/inference time)

⬇

Mixture of Experts (MoE) can be a solution!



* **Mixture of Experts (MoE):**
- The ensemble concept introduced in [Jacobs et al., 1991]
- Multiple sub-models (experts) are chosen per example, with gating mechanism



Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. Neural Computing, 1991.

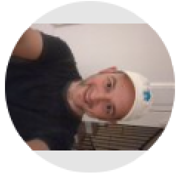
MoE for Deep Neural Networks

Published as a conference paper at ICLR 2017

OUTRAGEOUSLY LARGE NEURAL NETWORKS:
THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER

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

[1]Google Brain, {noam,azalia,andydavis,qvl,geoffhinton,jeff}@go
[2]Jagiellonian University, Cracow, krzysztof.maziarz@student.u



Noam Shazeer
Character.ai
character.ai의 이메일 확인됨
Deep Learning

팔로우

제목	인용	연도
Attention is all you need A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, ... Advances in neural information processing systems 30	103727	2017
Exploring the limits of transfer learning with a unified text-to-text transformer C Raffel, N Shazeer, A Roberts, K Lee, S Narang, M Matena, Y Zhou, W Li, ... The Journal of Machine Learning Research 21 (1), 5485-5551	12268	2020
Palm: Scaling language modeling with pathways A Chowdhery, S Narang, J Devlin, M Bosma, G Mishra, A Roberts, ... Journal of Machine Learning Research 24 (240), 1-113	2383	2023
Scheduled sampling for sequence prediction with recurrent neural networks S Bengio, O Vinyals, N Jaitly, N Shazeer Advances in neural information processing systems 28	2108	2015
Image transformer N Parmar, A Vaswani, J Uszkoreit, L Kaiser, N Shazeer, A Ku, D Tran International conference on machine learning, 4055-4064	1664	2018
Outrageously large neural networks: The sparsely-gated mixture-of-experts layer N Shazeer, A Mirhoseini, K Maziarz, A Davis, Q Le, G Hinton, J Dean arXiv preprint arXiv:1701.06538	1569	2017

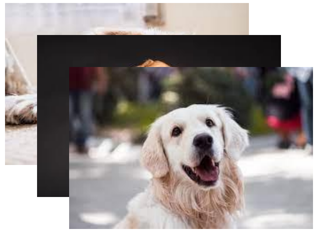
SparseMoE

Shazeer, Noam, et al. (Google Brain): "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." ICLR 2017

Motivation

- The capacity (# of parameters) of neural networks can give better accuracy

active **inactive**



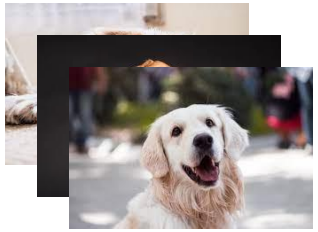
DNN

Simple task
with a small model

Motivation

- The capacity (# of parameters) of neural networks can give better accuracy
- Whole parts are active: quadratic blow-up in computing costs

active inactive



DNN

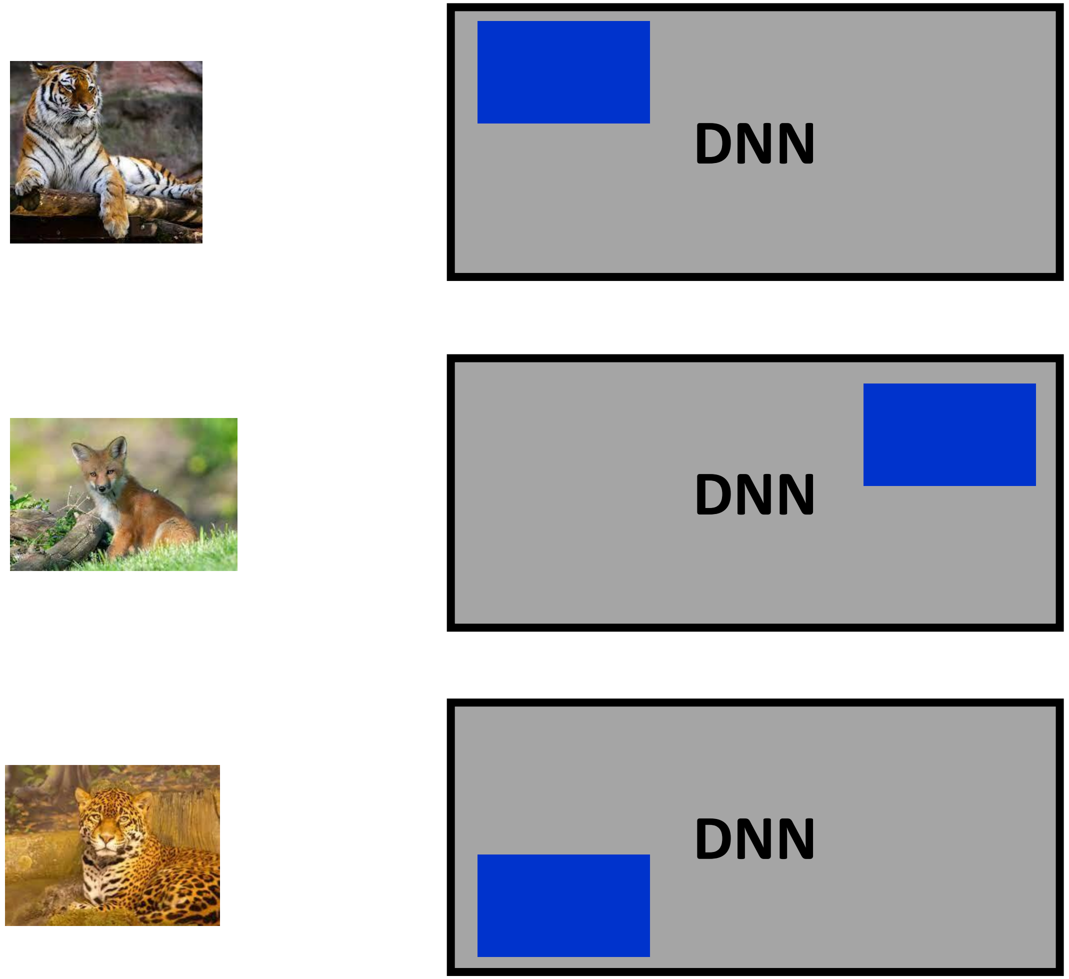
Simple task
with a small model

DNN

Complex task
with a massive model

Motivation

- Not all parts may be necessary for each data point



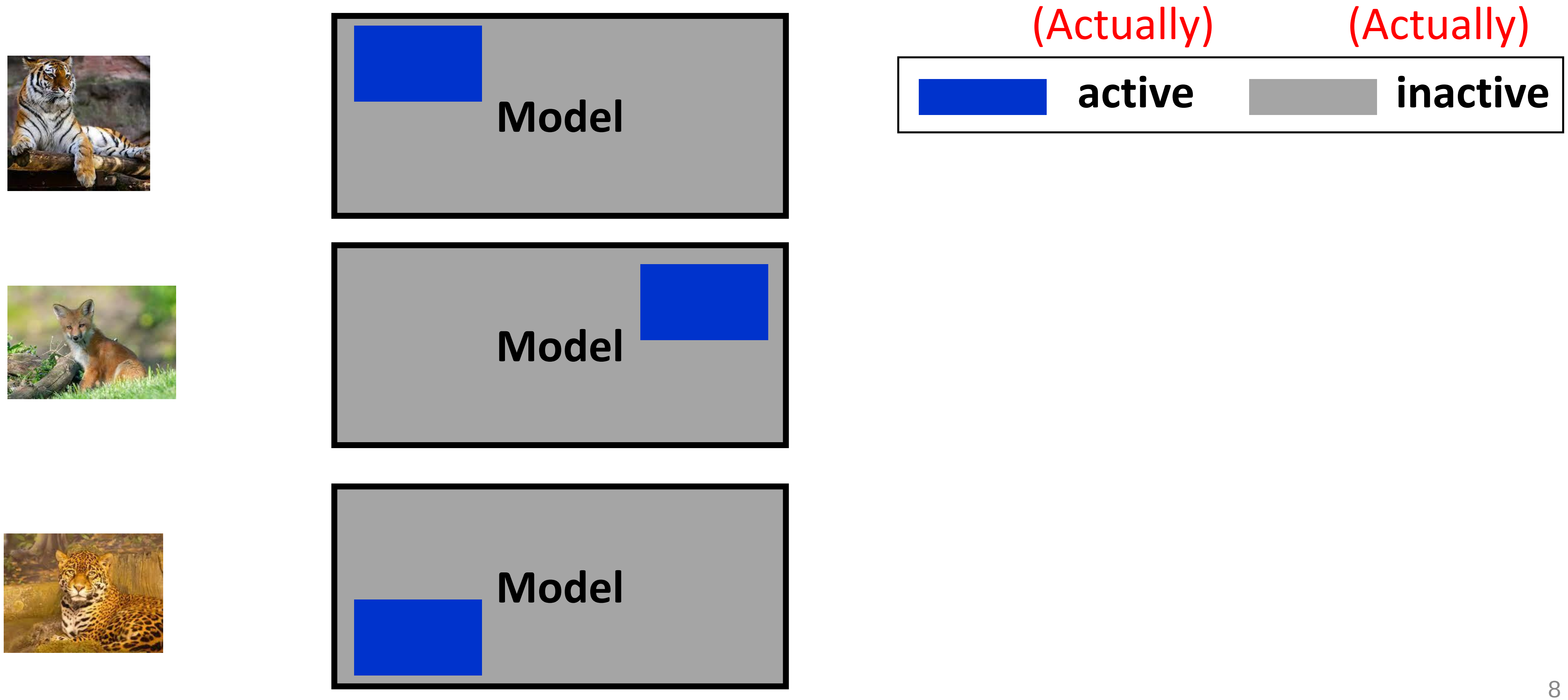
(Should be) (Should be)

active **inactive**

If so,
current architecture is
very inefficient

Motivation

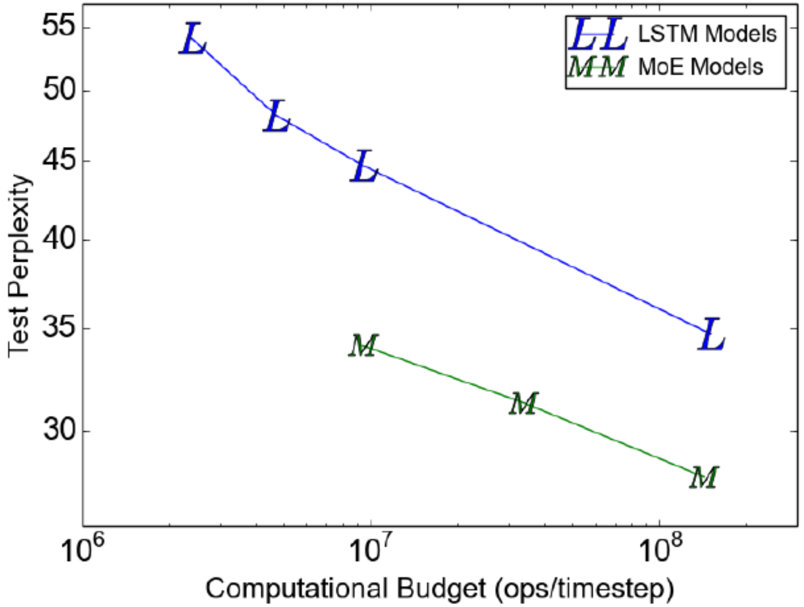
- **Mixture of Experts**, can be **a solution!**
- **Conditional computation:**
  Part of the network are active on per-example basis

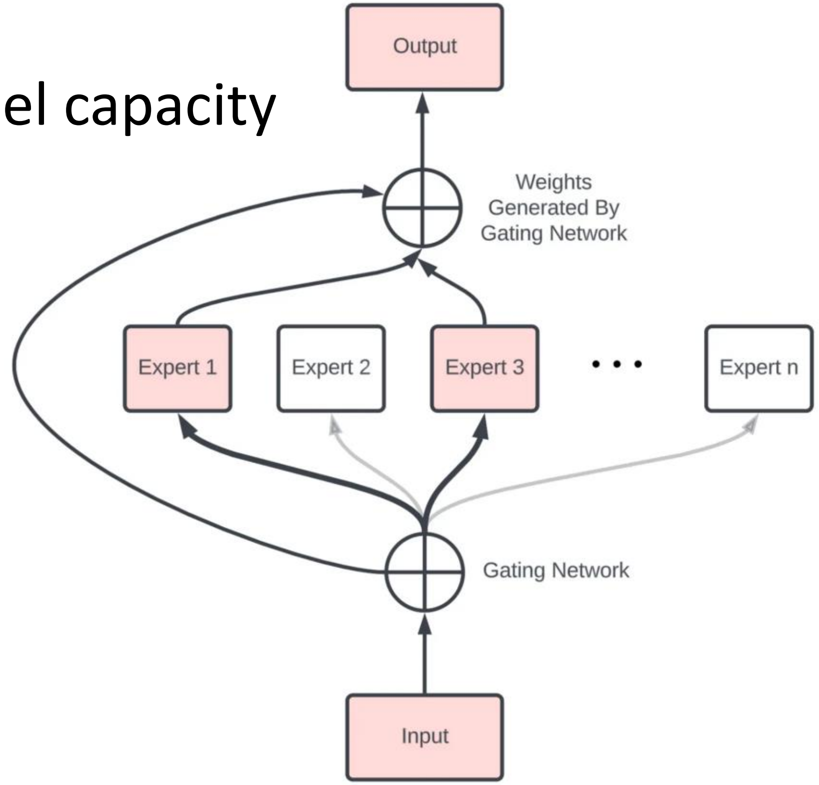


SparseMoE: The Deep Learning Way of MoE

- **Sparsely-gated Mixture-of-Experts layer (MoE):**
  Extension of MoE gatings to deep learning

- **Consisting of ~1000 sub-networks,**
  SparseMoE achieves **1000x** improvement in model capacity





https://deepgram.com/learn/mixture-of-experts-ml-model-guide

Challenges

- Limitations of dense models are clear
  but there are **challenges** in designing deep learning-based MoE:

  C1: Modern computing devices (e.g., GPU) are **much faster at arithmetic than branching**

  C2: **Sparsity levels** may be unstable

  C3: **Larger batch sizes** benefit performance in DNN but are reduced by conditional computation.

The Sparsely-Gated Mixture-of-Experts Layer (For C1)

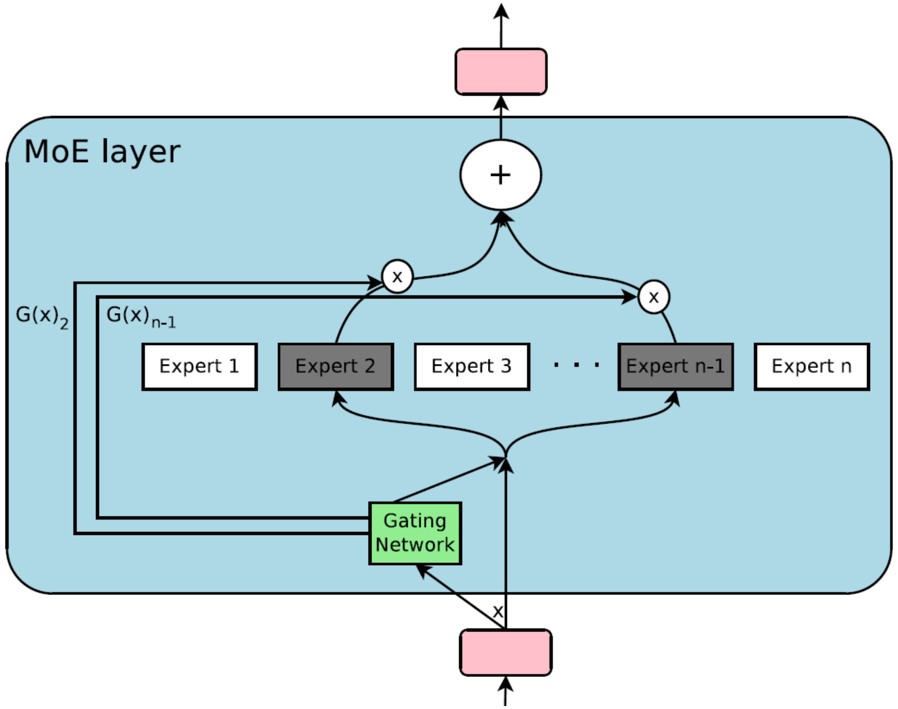
C1: Modern computing devices are much faster at arithmetic than branching

- Trainable gating network: selects a sparse combination of the experts to process each input

- $G(x)$: gating network
  $E_i(x)$: i-th expert network

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

➡ Simple arithmetic gating (solves **C1**)



Gating Network (For C2)

C2: Sparsity levels may be unstable

- Softmax gating
  - ✓ (Naïve approach) simple non-sparse gating function

$$G_\sigma(x) = Softmax(x \cdot W_g)$$

- Noisy top-K gating
  - ✓ Only top-K experts are activated
  - ✓ Maintain sparsity levels
  - ✓ The noise term helps with load balancing

prevent case when only few experts are repeatedly selected

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

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

➡ Maintain sparsity level (solves C2)

Balancing Expert Utilization (For C2)

- Empirically proven that MoE always produces large weights for the same few experts.

- Additional soft constraint (loss term)
  - ✓ Encourages all experts to have equal importance
  - ✓ Low variation : Even distribution

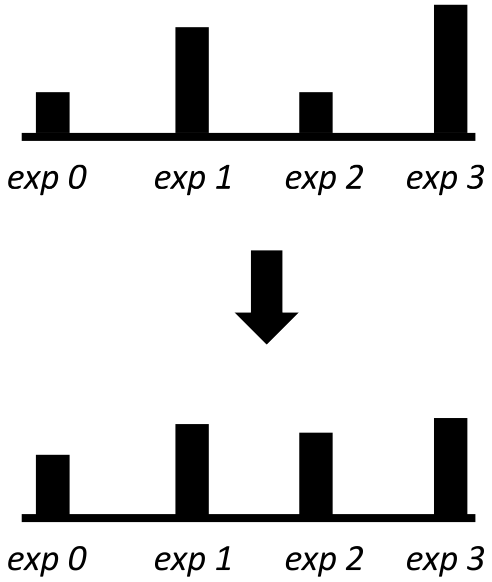
$$Importance(X) = \sum_{x \in X} G(x)$$

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

Coefficient of variation

Hyperparameter

$$CV = \frac{\sigma}{\mu}$$



The Shrinking Batch Problem (For C3)

C3: Larger batch sizes benefit performance but are reduced by conditional computation.

- Shrinking batch problem
  - ✓ As # experts increases, each experts receives only few batch data

⬇ Relieve the problems

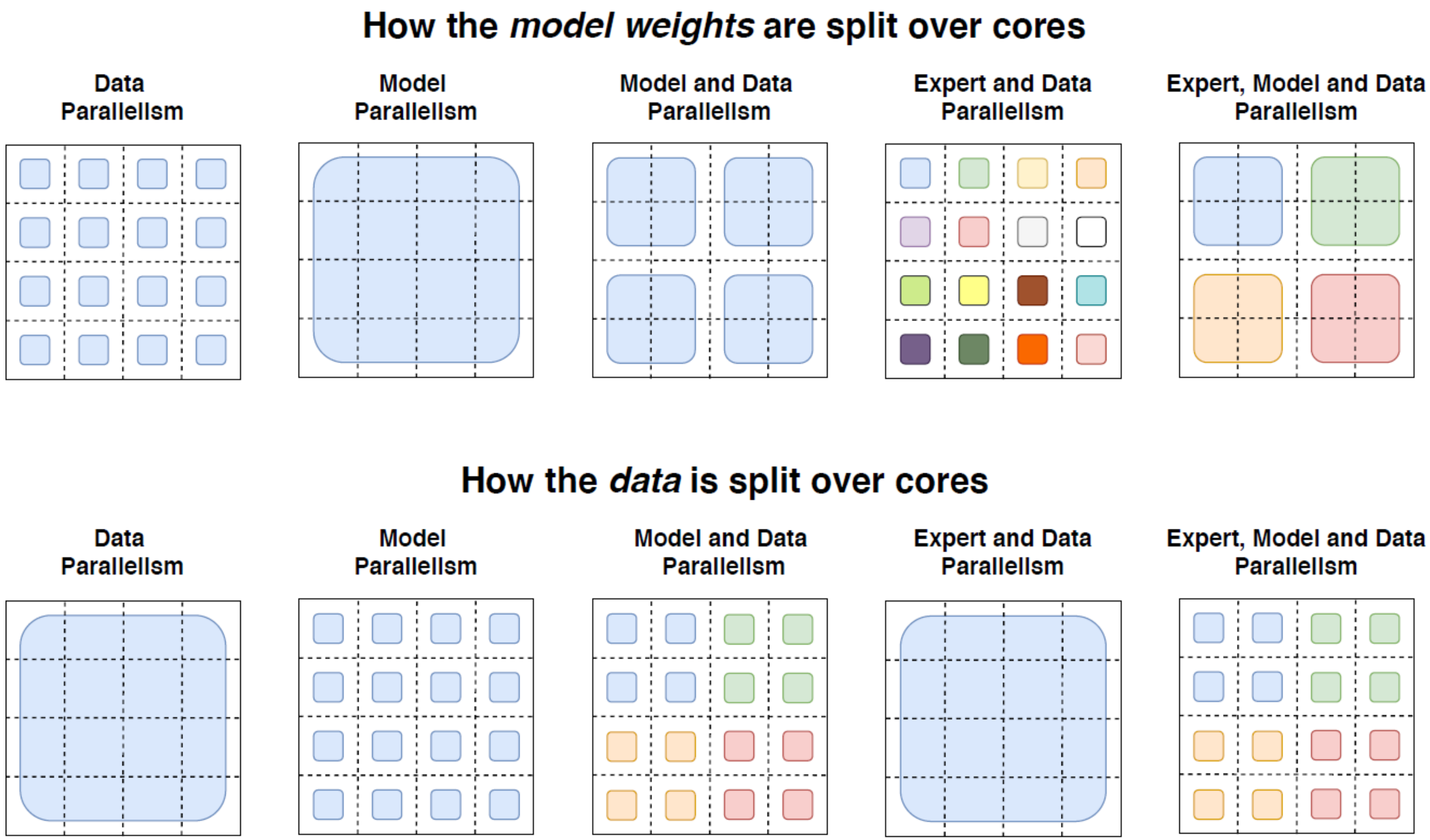
(solution 1) Increasing batch size further (* possible memory issue)

(solution 2) Distributed learning technique: Each expert receives a combined batch consisting of the relevant examples

➡ Reduce batch-relevant problem
(solves C3)

Expert parallelism

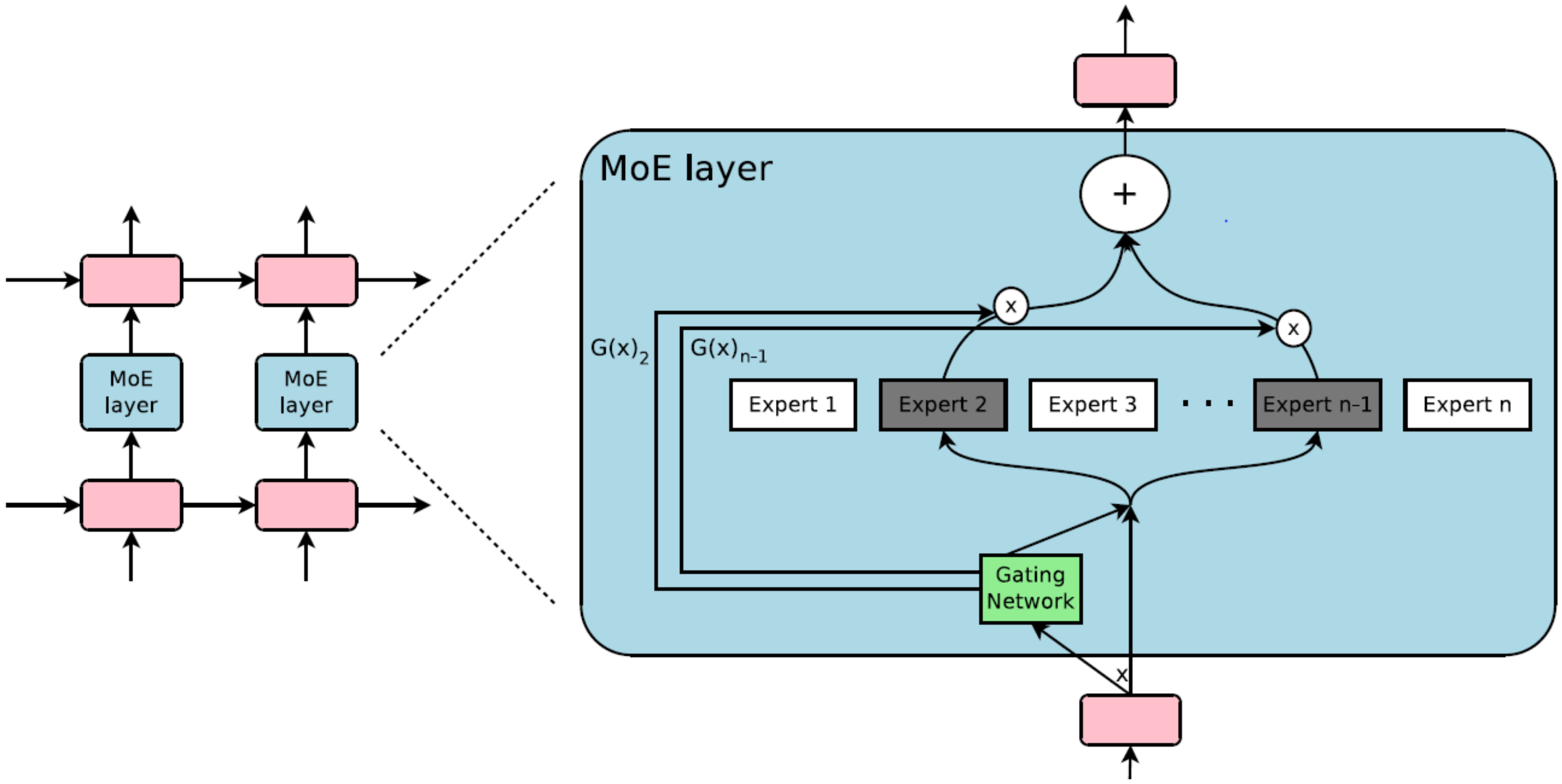
- MoE + Distributed learning
- Each expert is loaded on different device



Fedus, William, Barret Zoph, and Noam Shazeer. "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity." Journal of Machine Learning Research 23.120 (2022): 1-39.

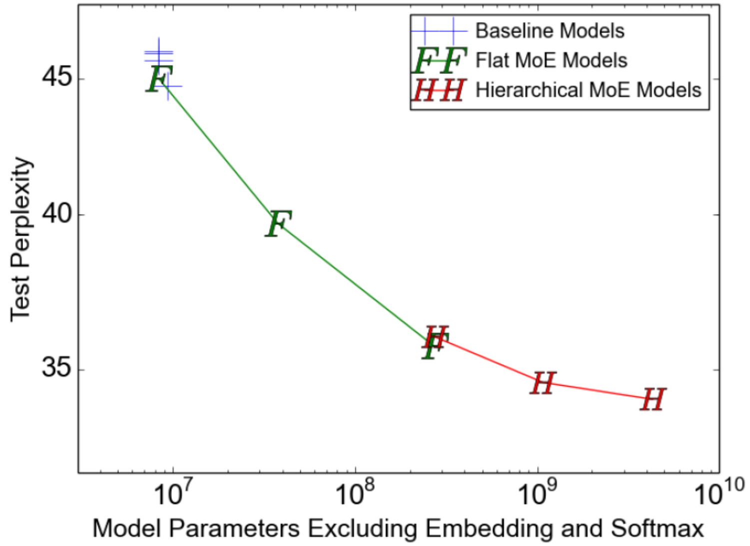
Experiments

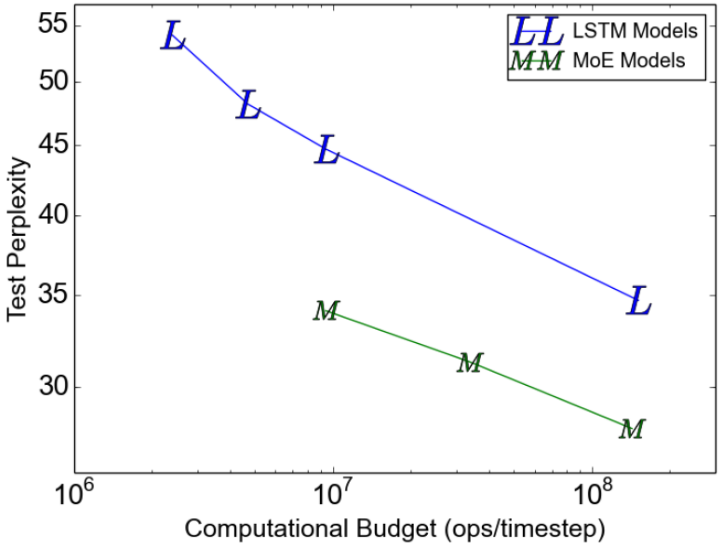
- Experiments on recurrent language model (whose task requires big model)



Experiments 1: Efficiency

- Better performs than SOTA baseline LMs
- Achieves better performance with fewer ops (# operations)

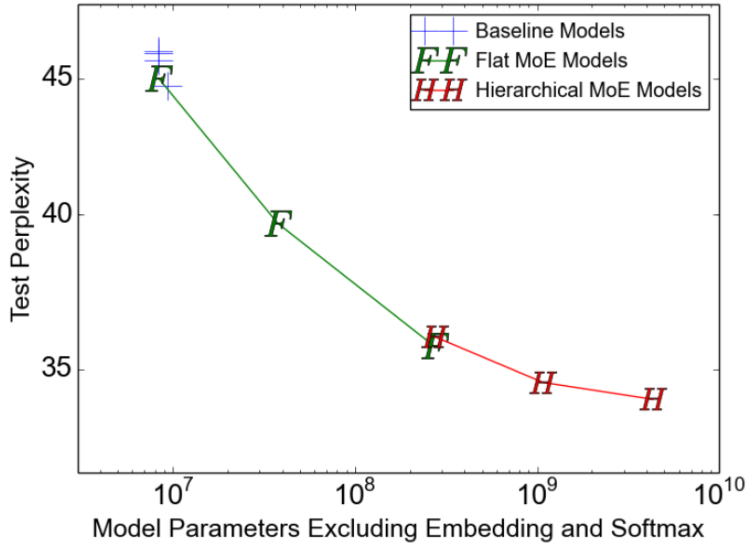


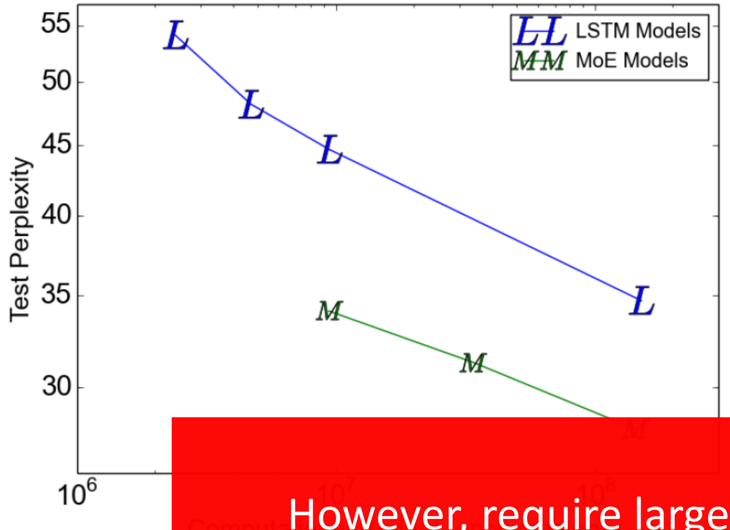


	Test Perplexity 10 epochs	Test Perplexity 100 epochs	#Parameters excluding embedding and softmax layers	ops/timestep	Training Time 10 epochs	TFLOPS /GPU
Best Published Results	34.7	30.6	151 million	151 million	59 hours, 32 k40s	1.09
Low-Budget MoE Model	34.1		4303 million	8.9 million	15 hours, 16 k40s	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**

Experiments 1: Efficiency

- Better performs than SOTA baseline LMs
- Achieves better performance with fewer ops (# operations)





However, require larger # parameters to achieve a good accuracy

	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**

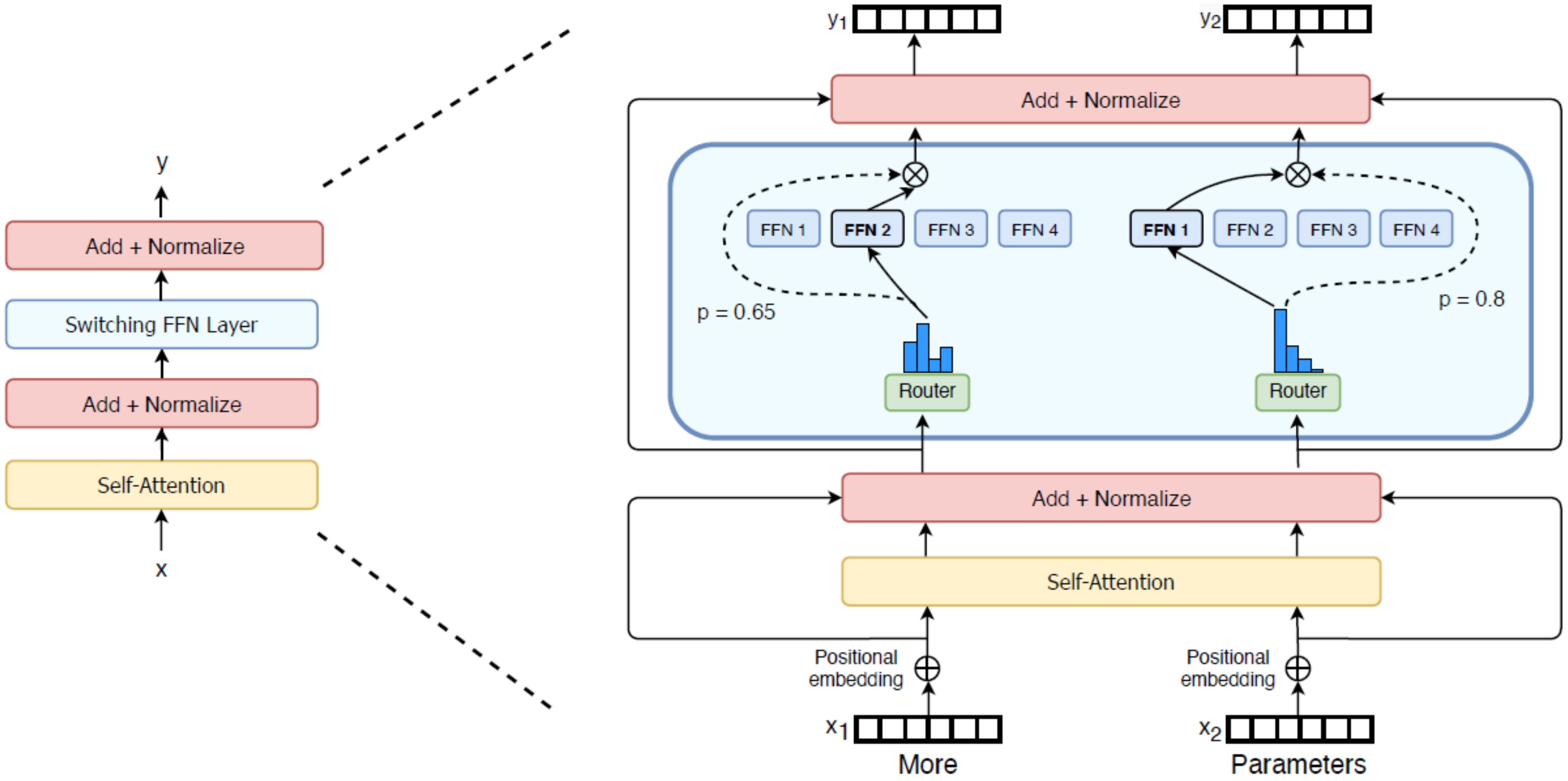
Experiments 2: Accuracy

- Achieves better results in LM tasks (machine translation)
- Because of larger # params?

	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

SwitchTransformer

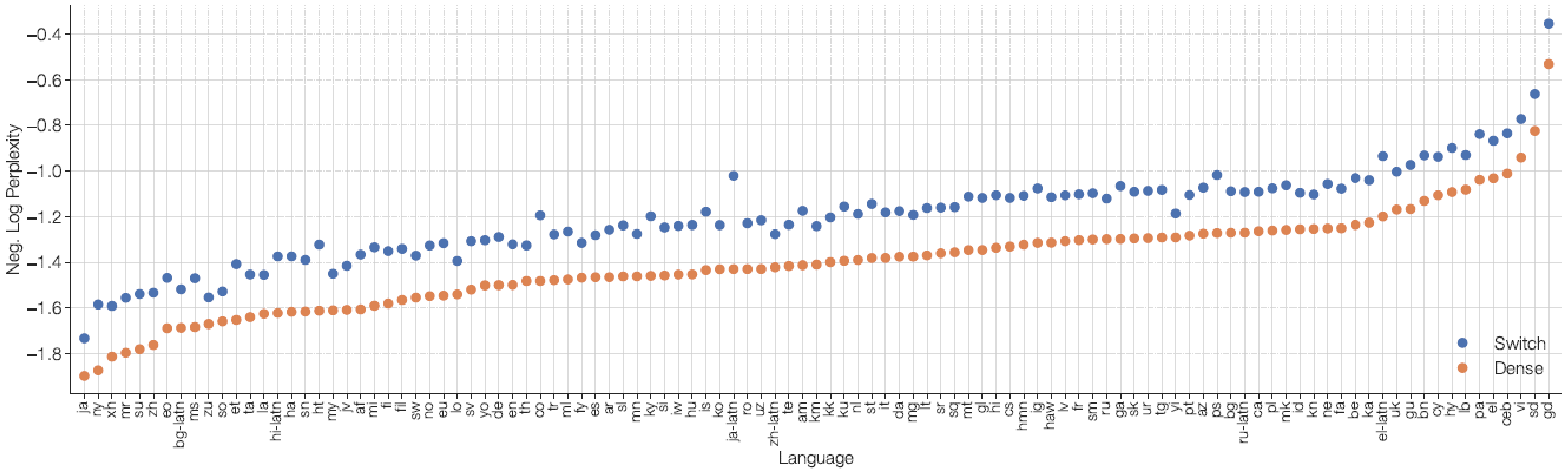
- **FFN layer** in transformer is **replaced with MoE (1.6T)**
- Simpler routing: Top-1 expert + differentiable load balancing loss



Fedus, William, Barret Zoph, and Noam Shazeer. "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity." Journal of Machine Learning Research 23.120 (2022): 1-39.

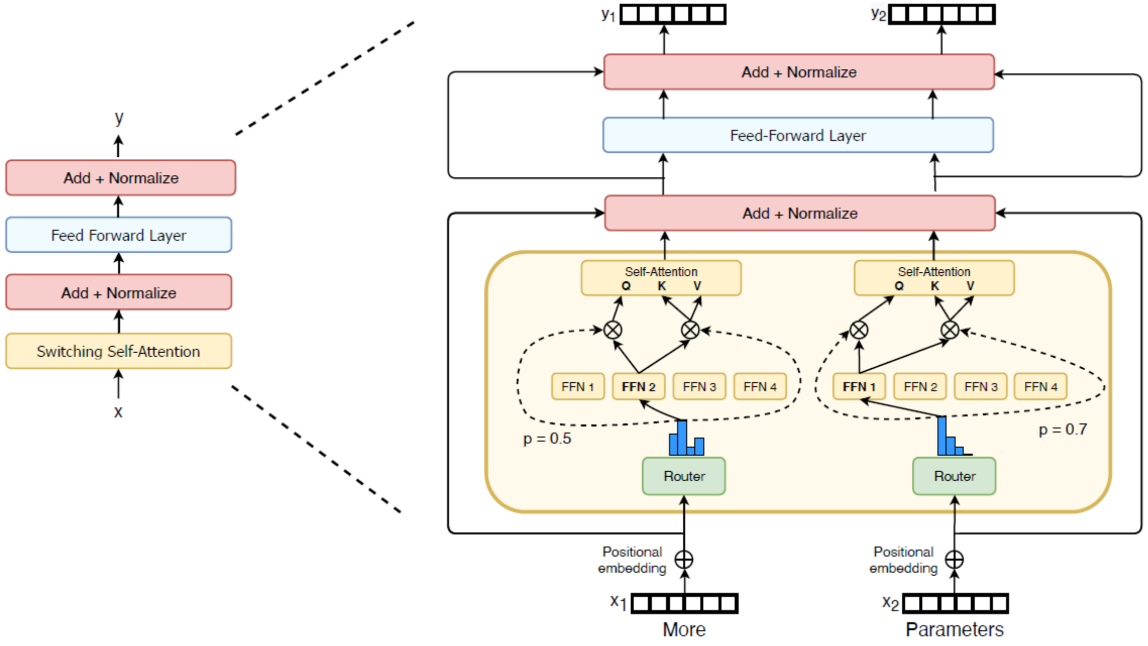
SwitchTransformer: The Results

- Multi-lingual training task
- Superior Multi-lingual translation performance than dense model (T5, [Raffel et al., 2020])



SwitchTransformer: The Results

- MoEs can replace different modules in transformer
- Replacing all achieves best



Model	Precision	Quality @100k Steps (↑)	Quality @16H (↑)	Speed (ex/sec) (↑)
Experts FF	float32	-1.548	-1.614	1480
Expert Attention	float32	-1.524	**-1.606**	1330
Expert Attention	bfloat16	[diverges]	[diverges]	–
Experts FF + Attention	float32	**-1.513**	-1.607	1240
Expert FF + Attention	bfloat16	[diverges]	[diverges]	–

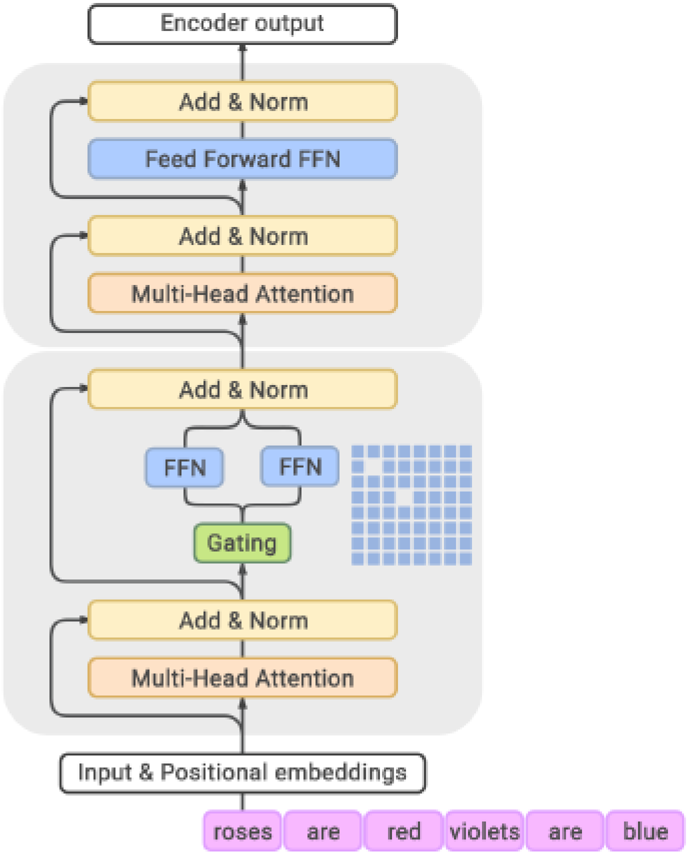
GLaM

- GLaM, the MoE-based language models
- Introduce 1.9B ~ 1.2T MoE models

Model Name	Model Type	n_{params}	$n_{\text{act-params}}$
BERT	Dense Encoder-only	340M	340M
T5	Dense Encoder-decoder	13B	13B
GPT-3	Dense Decoder-only	175B	175B
Jurassic-1	Dense Decoder-only	178B	178B
Gopher	Dense Decoder-only	280B	280B
Megatron-530B	Dense Decoder-only	530B	530B
GShard-M4	MoE Encoder-decoder	600B	1.5B
Switch-C	MoE Encoder-decoder	1.5T	1.5B
GLaM (64B/64E)	MoE Decoder-only	1.2T	96.6B

Table 1. Comparison between GPT-3 and GLaM. In a nutshell, GLaM outperforms GPT-3 across 21 natural language understanding (NLU) benchmarks and 8 natural language generative (NLG) benchmarks in average while using about half the FLOPs per token during inference and consuming about one third the energy for training.

		GPT-3	GLaM	relative
cost	FLOPs / token (G)	350	**180**	−48.6%
	Train energy (MWh)	1287	**456**	−64.6%
accuracy on average	Zero-shot	56.9	**62.7**	+10.2%
	One-shot	61.6	**65.5**	+6.3%
	Few-shot	65.2	**68.1**	+4.4%



Du, Nan, et al. "GLaM: Efficient scaling of language models with mixture-of-experts." International Conference on Machine Learning. PMLR, 2022.

Open MoE Models

Open Source MoEs

There are nowadays several open source projects to train MoEs:

- Megablocks: https://github.com/stanford-futuredata/megablocks
- Fairseq: https://github.com/facebookresearch/fairseq/tree/main/examples/moe_lm
- OpenMoE: https://github.com/XueFuzhao/OpenMoE

In the realm of released open access MoEs, you can check:

- Switch Transformers (Google): Collection of T5-based MoEs going from 8 to 2048 experts. The largest model has 1.6 trillion parameters.
- NLLB MoE (Meta): A MoE variant of the NLLB translation model.
- OpenMoE: A community effort that has released Llama-based MoEs.
- Mixtral 8x7B (Mistral): A high-quality MoE that outperforms Llama 2 70B and has much faster inference. A instruct-tuned model is also released. Read more about it in the announcement blog post.

Source: https://huggingface.co/blog/moe

Other Research Trends (1)

- Make MoE more efficiently

  - Faster [Belcak et al., 2023], [He et al., 2023]

  - Lower-memory [Franta et al., 2023]

FasterMoE: Modeling and Optimizing Training of Large-Scale Dynamic Pre-Trained Models

Jiaao He, Tsinghua University, hja20@mails.tsinghua.edu.cn
Jidong Zhai*, Tsinghua University, zhaijidong@tsinghua.edu.cn
Tiago Antunes, Tsinghua University, vazama10@mails.tsinghua.edu.cn
Haojie Wang, Tsinghua University, wanghaojie@tsinghua.edu.cn
Fuwen Luo, Tsinghua University, lfw19@mails.tsinghua.edu.cn
Shangfeng Shi, Tsinghua University, ssf20@mails.tsinghua.edu.cn
Qin Li, Tsinghua University, liqin20@mails.tsinghua.edu.cn

Abstract and integrate the above optimizations as a general system,

QMoE: Practical Sub-1-Bit Compression of Trillion-Parameter Models

Elias Frantar [1] Dan Alistarh [1 2]

Fast Feedforward Networks

Peter Belcak and Roger Wattenhofer
ETH Zürich
{belcak,wattenhofer}@ethz.ch

Abstract

We break the linear link between the layer size and its inference cost by introducing the fast feedforward[1] (FFF) architecture, a log-time alternative to feedforward networks.

We demonstrate that FFFs are up to 220x faster than feedforward networks, up to 6x faster than mixture-of-experts networks, and exhibit better training properties than mixtures of experts thanks to noiseless conditional execution.

Pushing FFFs to the limit, we show that they can use as little as 1% of layer neurons for inference in vision transformers while preserving 94.2% of predictive performance.

Introduction

The feedforward layer is a parameter-heavy building block of transformer models (Vaswani et al. 2017). Growing to tens of thousands of hidden neurons in recent years, the cost of feedforward layer inference is now in the sights of those seeking to make large models faster.

It has been recognized that in very large networks, only a small portion of the feedforward hidden neurons plays a role in determining the output for any single input, and that it is possible to design networks that are modular in order to utilize this fact (Bengio et al. 2015).

The most recent work on the modularization of feedforward layers aims at architectural designs that implicitly encourage sparsity (Shazeer et al. 2017; Lepikhin et al. 2020; Fedus, Zoph, and Shazeer 2022). They share the common approach of subdividing the feedforward layer into separate blocks of neurons – "experts" – and training a gating layer to determine the mixture of experts to be used in the forward pass. Inference acceleration is then achieved by using only the best-scoring k blocks, or a variant thereof. This approach scales down the inference time by a constant but remains linear in the width of the feedforward layer. Moreover, it relies on noisy gating to allow for load balancing among the experts, complicating training and encouraging duplicity.

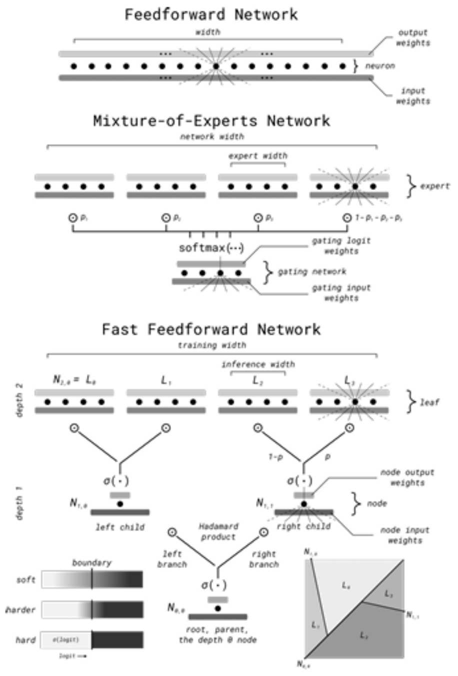


Figure 1: A fast feedforward network set in comparison to its peers. *Bottom.* Illustrations of the resulting regionalization of the input space and varying boundary hardness.

Other Research Trends (2)

- Multi-task learning [Shen et al., 2023], [Chen et al., 2023]

Mixture-of-Experts Meets Instruction Tuning: A Winning Combination for Large Language Models

Sheng Shen[♮*] Le Hou[†] Yanqi Zhou[†] Nan Du[†] Shayne Longpre[⊤*] Jason Wei[†],
Hyung Won Chung[†] Barret Zoph[†] William Fedus[†] Xinyun Chen[†] Tu Vu[‡*],
Yuexin Wu[†] Wuyang Chen[§*] Albert Webson[†] Yunxuan Li[†] Vincent Zhao[†] Hongkun Yu[†]
Kurt Keutzer[♮] Trevor Darrell[♮] Denny Zhou[†]

[†]Google [♮]University of California, Berkeley [⊤]Massachusetts Institute of Technology
[‡]University of Massachusetts Amherst [§]The University of Texas at Austin

Mod-Squad: Designing Mixtures of Experts As Modular Multi-Task Learners

Zitian Chen[1], Yikang Shen[2], Mingyu Ding[3], Zhenfang Chen[2],
Hengshuang Zhao[3], Erik Learned-Miller[1], Chuang Gan[1,2]
[1] University of Massachusetts Amherst, [2] MIT-IBM Watson AI Lab, [3] The University of Hong Kong

Abstract

Optimization in multi-task learning (MTL) is more challenging than single-task learning (STL), as the gradient from different tasks can be contradictory. When tasks are related, it can be beneficial to share some parameters among them (cooperation). However, some tasks require additional parameters with expertise in a specific type of data or discrimination (specialization). To address the MTL challenge, we propose ***Mod-Squad****, a new model that is* ***Mod****ularized into groups of experts (a '****Squad****'). This structure allows us to formalize cooperation and specialization as the process of matching experts and tasks. We optimize this matching process during the training of a single model. Specifically, we incorporate mixture of experts (MoE) layers into a transformer model, with a new loss that incorporates the mutual dependence between tasks and experts. As a result, only a small set of experts are activated for each task. This prevents the sharing of the entire backbone model between all tasks, which strengthens the model, especially when the training set size and the number of tasks scale up. More interestingly, for each task, we can extract the small set of experts as a standalone model that maintains the same performance as the large model. Extensive experiments on the Taskonomy dataset with 13 vision tasks and the PASCAL-Context dataset with 5 vision tasks show the superiority of our approach. The project page can be accessed at https://vis-www.cs.umass.edu/mod-squad.*

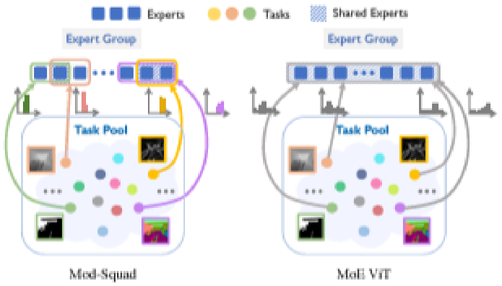


Figure 1. **A comparison between Mod-Squad and MoE ViT.** Our key motivation is that experts should leverage commonalities in some tasks (cooperation) but focus on a subset of tasks that require specific features and do not interfere with each other (specialization).

set of tasks. On the one hand, tasks often benefit by sharing parameters, i.e., **cooperation**. On the other hand, some tasks may require specialized expertise that only benefits that single task, i.e., **specialization**. A good MTL system should be flexible to optimize experts for the dual purposes of cooperation and specialization.

There are two well-known challenges in MTL: (1) gradient conflicts across tasks [5, 38]; and (2) how to design architectures that have both high accuracy and computational efficiency.

Takeaway Messages

- Recently, MoE receives huge attention because of the rise of LLMs
- Using MoE, we can get **accurate results with better efficiency**
- When to use MoE?

	Inference/training time	Memory (VRAM)
Dense model	Slow	Small
MoE (sparse model)	Fast	Large

"Success is not final, failure is not fatal:
it is the courage to continue that counts."
- Winston Churchill

Contact: jindeok6@yonsei.ac.kr
Web Page: https://jindeok.github.io/jdpark/