Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity

William Fedus, Barret Zoph, Noam Shazeer

2020

Google Brain

1. Introduction

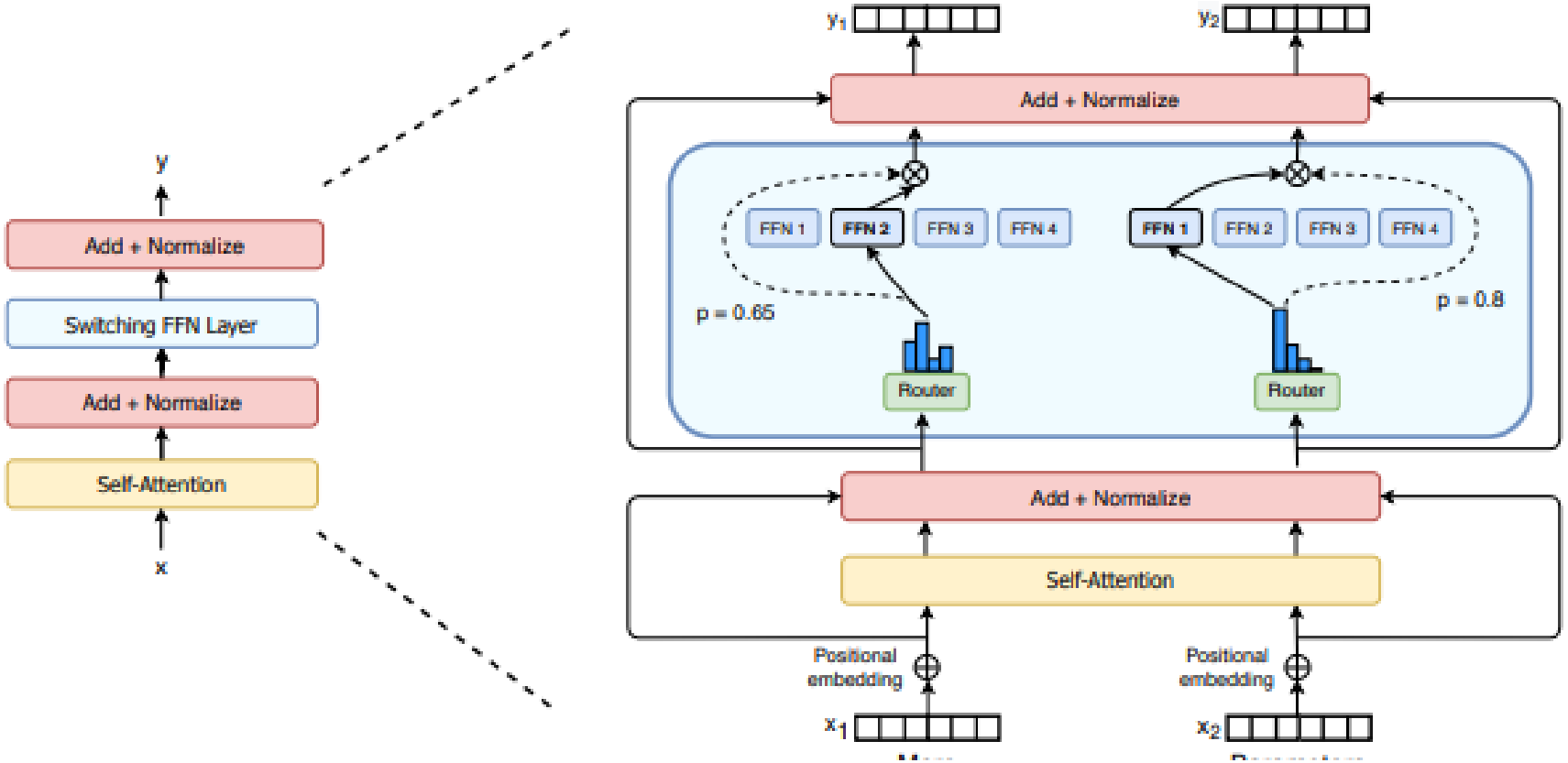
- 대규모 모델의 필요성

  최근 (2020년)에는 모델 크기를 늘려 성능을 개선하려는 경향이 강하다.
  전통적인 Transformer는 계산 비용이 기하 급수적으로 증가한다.
  따라서 동일한 연산량(FLOP) 내에서 파라미터 수를 늘릴 수 있는 sparse 모델
  즉, MoE 접근법이 주목받게 되었다.

- 기존 MoE의 한계

  기존 MoE는 여러 전문가 중 top-K를 선택하여 각 토큰을 여러 전문가에게 분배하였다.
  하지만 이 때 라우팅 계산, 통신비용 그리고 학습이 불안정하다는 문제점이 지적되었다.

2. Structure


y
Add + Normalize
Switching FFN Layer
Add + Normalize
Self-Attention
x
y1
y2
Add + Normalize
FFN 1
FFN 2
FFN 3
FFN 4
FFN 1
FFN 2
FFN 3
FFN 4
p = 0.65
p = 0.8
Router
Router
Add + Normalize
Self-Attention
Positional embedding
Positional embedding
x1
x2

2. Switch Transformer

2-1. Simplifying Sparse Routing

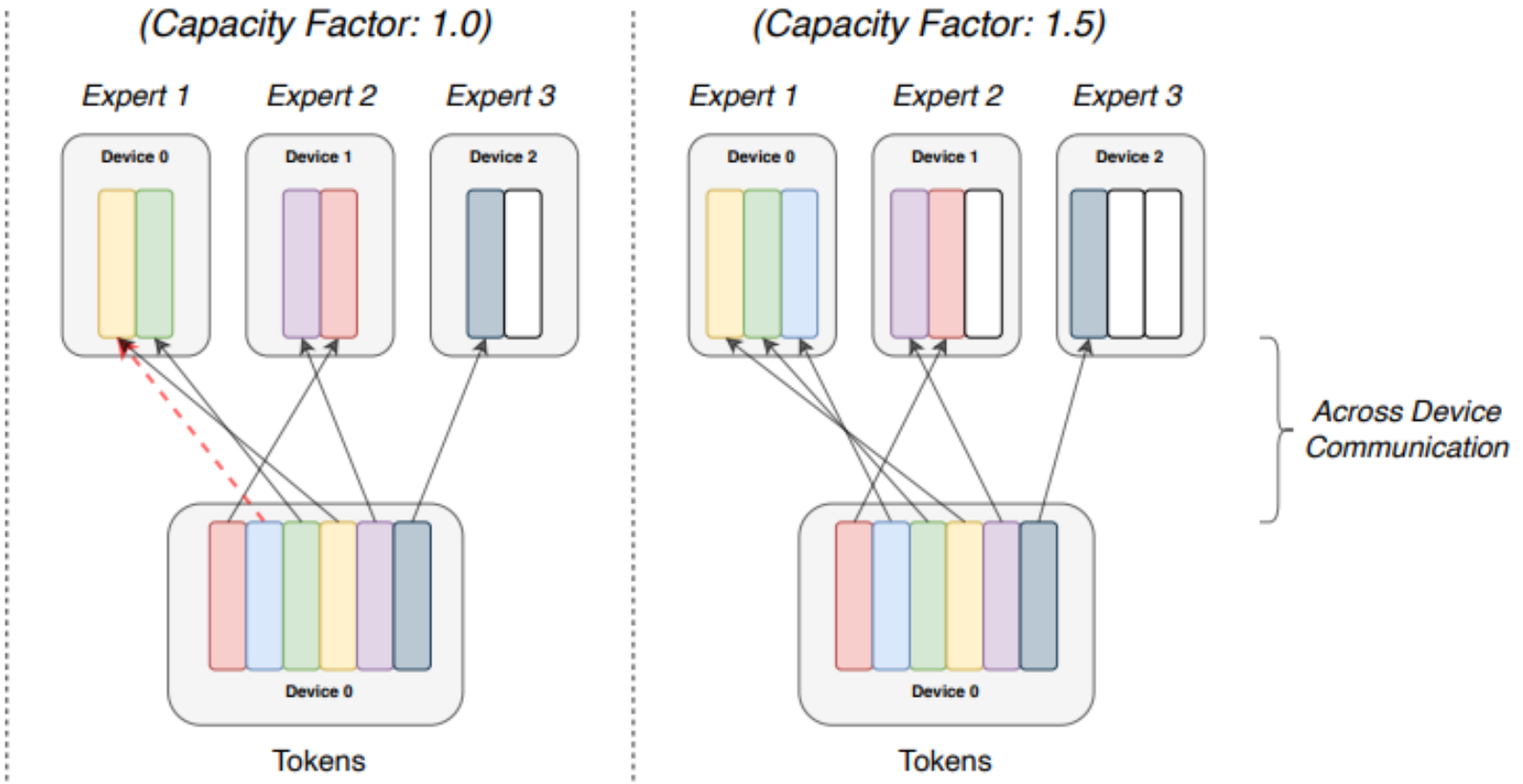
- 기존 MoE 모델은 입력을 여러 전문가에게 배분하는 Top-K 라우팅 방식
- Switch Transformer는 단일 전문가 (k=1) 선택
- 장점
  - 라우팅 연산 감소 -> 속도 증가
  - 각 전문가의 배치 크기를 줄여 효율적으로 활용 가능
  - 통신 비용 감소 및 구조의 단순화

2. Switch Transformer

2-1. Simplifying Sparse Routing

Terminology

- **Experts:** Split across devices, each having their own unique parameters. Perform standard feed-forward computation.
- **Expert Capacity:** Batch size of each expert. Calculated as
- (tokens_per_batch / num_experts) * capacity_factor
- **Capacity Factor:** Used when calculating expert capacity. Expert capacity allows more buffer to help mitigate token overflow during routing.

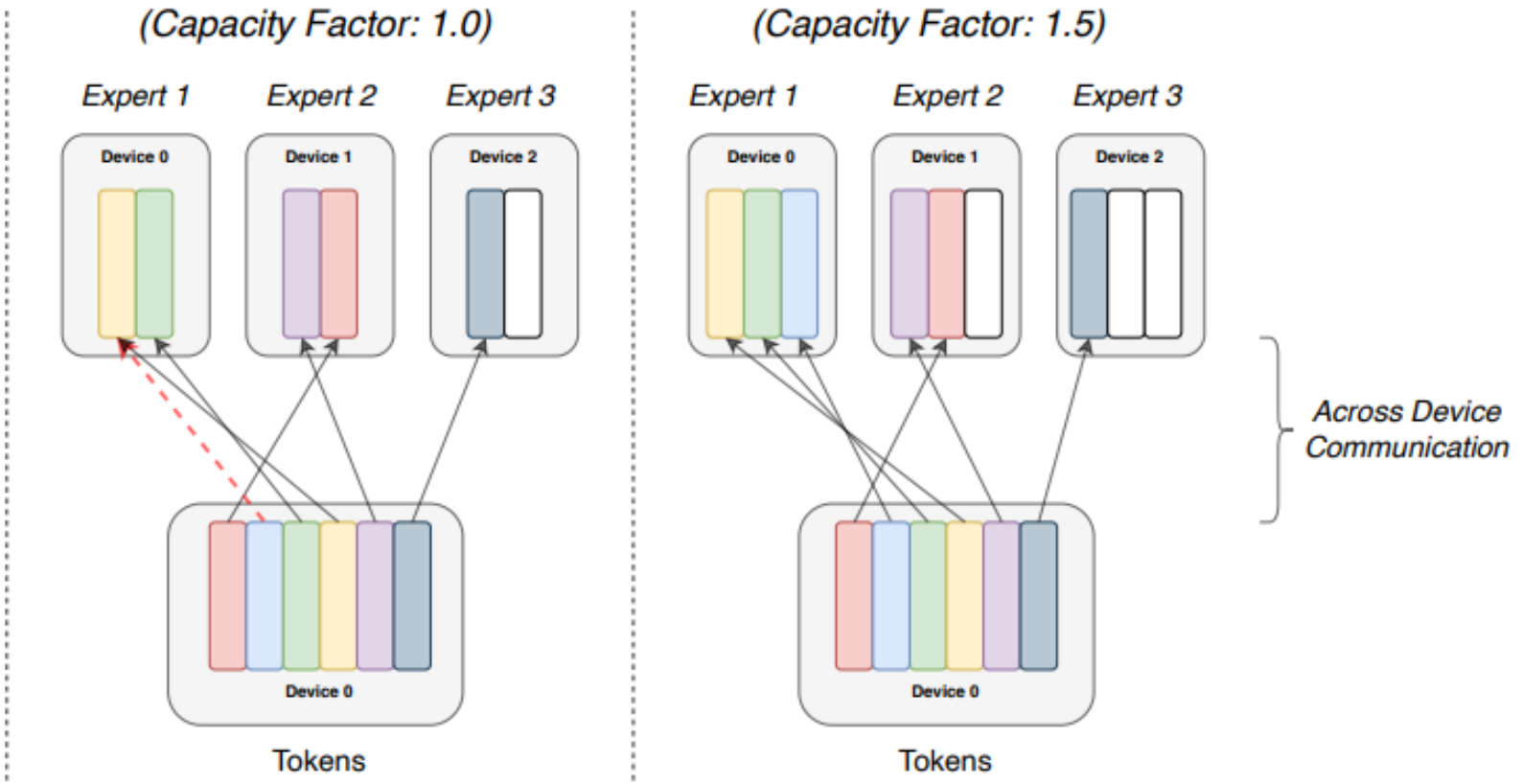


2. Switch Transformer

2-1. Simplifying Sparse Routing

Terminology

- **Experts:** Split across devices, each having their own unique parameters. Perform standard feed-forward computation.
- **Expert Capacity:** Batch size of each expert. Calculated as (tokens_per_batch / num_experts) * capacity_factor
- **Capacity Factor:** Used when calculating expert capacity. Expert capacity allows more buffer to help mitigate token overflow during routing.



2. Switch Transformer

2-2. Efficient Sparse Routing

- Mesh-Tensorflow(MTF) 라이브러리를 사용하여 효율적인 분산 학습 구현
- 라우팅 과정에서 전문가를 균등하게 유지하기 위해 Auxiliary Loss 추가

전문가별 토큰 분배 비율

$$f_i = \frac{1}{T} \sum_{x \in \mathcal{B}} \mathbb{1}\{\text{argmax}\, p(x) = i\}$$

전문가별 선택 확률 비율

$$P_i = \frac{1}{T} \sum_{x \in \mathcal{B}} p_i(x).$$

최종 Auxiliary Loss

$$\text{loss} = \alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i$$

손실값이 최소가 될수록 f와 p값이 유사해짐

전문가 간 부하가 균등해진다.

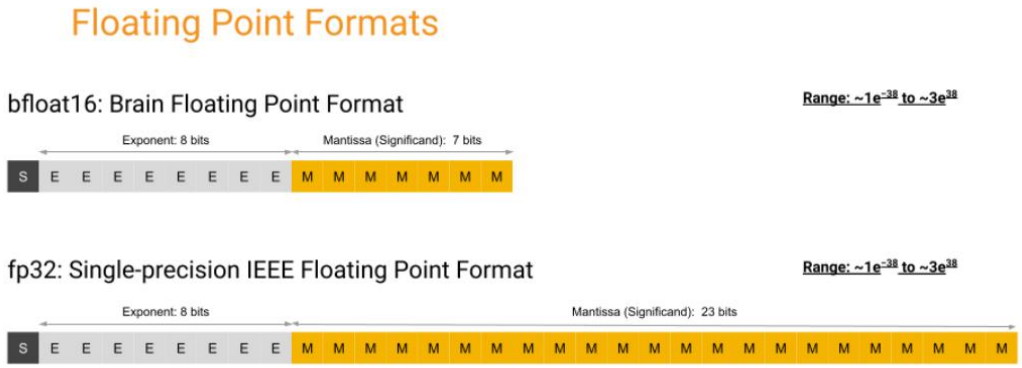
2-3. Switch Transfomer vs MoE

Model	Capacity Factor	Quality after 100k steps (↑) (Neg. Log Perp.)	Time to Quality Threshold (↓) (hours)	Speed (↑) (examples/sec)
T5-Base	—	-1.731	Not achieved†	1600
T5-Large	—	-1.550	131.1	470
MoE-Base	2.0	-1.547	68.7	840
Switch-Base	2.0	-1.554	72.8	860
MoE-Base	1.25	-1.559	80.7	790
Switch-Base	1.25	-1.553	65.0	910
MoE-Base	1.0	-1.572	80.1	860
Switch-Base	1.0	-1.561	**62.8**	1000
Switch-Base+	1.0	**-1.534**	67.6	780

2-4. Improved Training & Fine-Tuning Tech.

1. Selective Precision with large sparse models

- Bfloat16, float32 선택적으로 사용
- **라우터의 연산**은 float32를 사용



Model (precision)	Quality (Neg. Log Perp.) (↑)	Speed (Examples/sec) (↑)
Switch-Base (float32)	-1.718	1160
Switch-Base (bfloat16)	-3.780 [*diverged*]	**1390**
Switch-Base (Selective precision)	**-1.716**	1390

2-4. Improved Training & Fine-Tuning Tech.

2. Smaller parameter initialization for stability

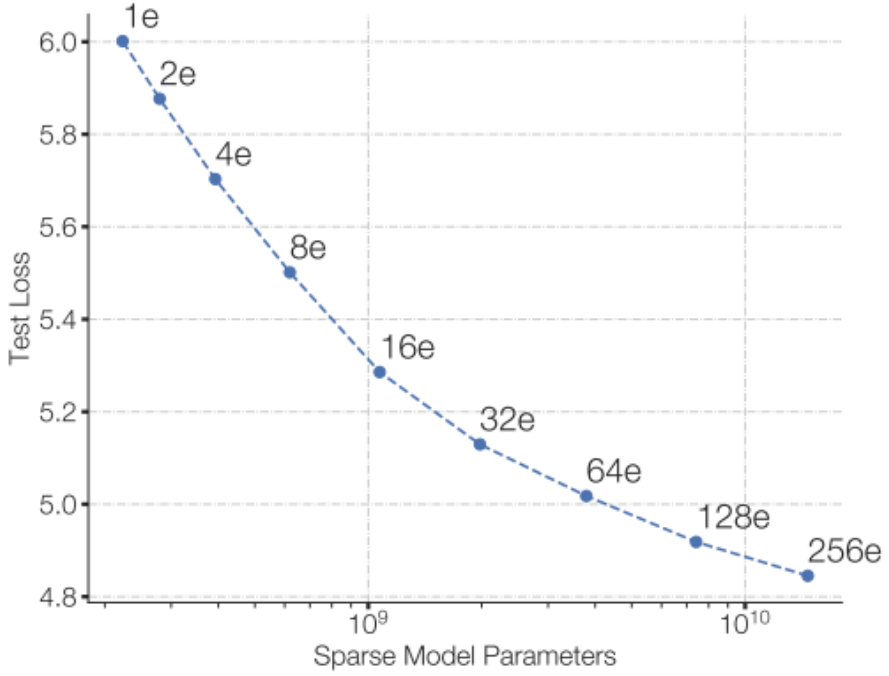
Model (Initialization scale)	Average Quality (Neg. Log Perp.)	Std. Dev. of Quality (Neg. Log Perp.)
Switch-Base (0.1x-init)	**-2.72**	**0.01**
Switch-Base (1.0x-init)	-3.60	0.68

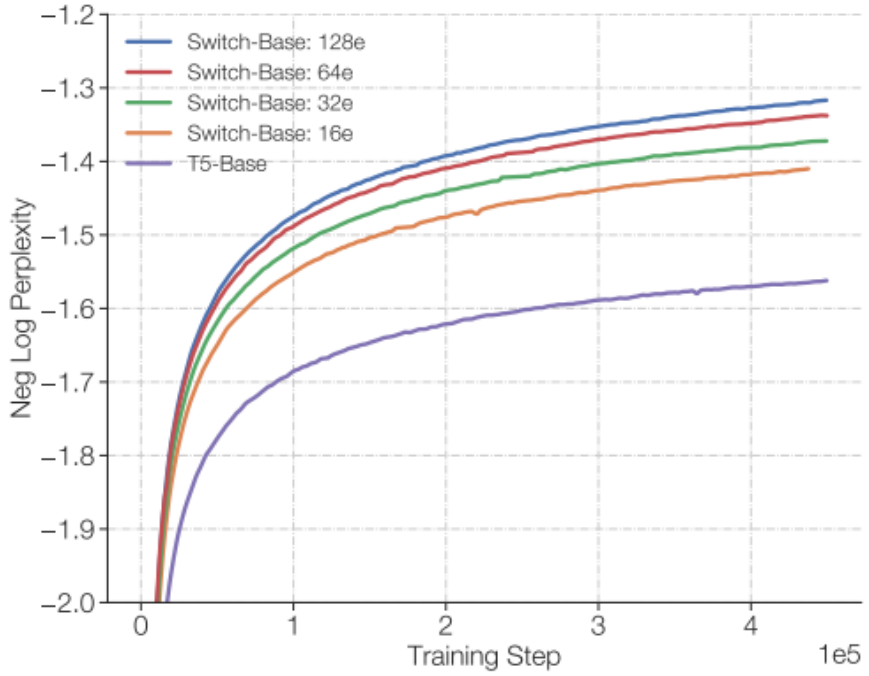
3. Expert Dropout

Model (dropout)	GLUE	CNNDM	SQuAD	SuperGLUE
T5-Base (d=0.1)	82.9	**19.6**	83.5	72.4
Switch-Base (d=0.1)	84.7	19.1	**83.7**	**73.0**
Switch-Base (d=0.2)	84.4	19.2	**83.9**	**73.2**
Switch-Base (d=0.3)	83.9	19.6	83.4	70.7
Switch-Base (d=0.1, ed=0.4)	**85.2**	**19.6**	**83.7**	**73.0**

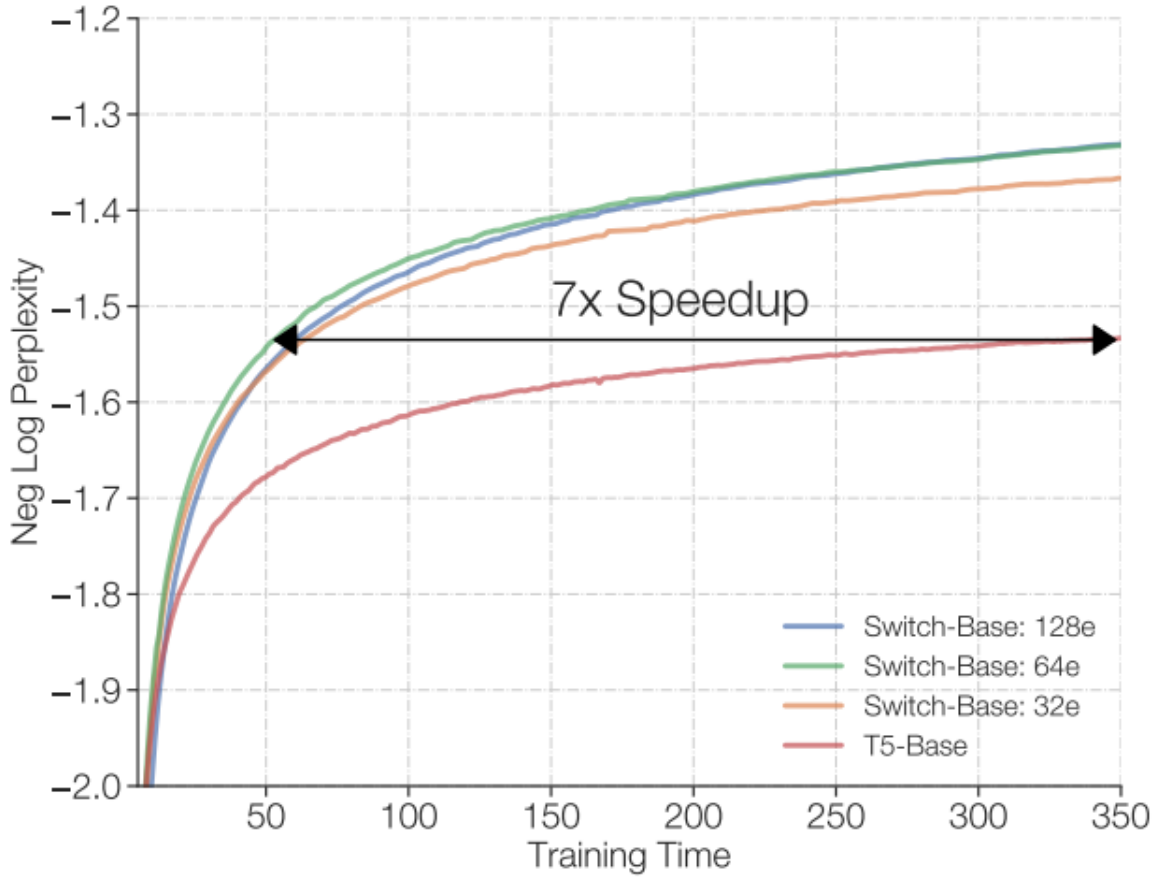
3-1. Scaling Results on a Step-Basis

- 동일한 FLOP budget에서 전문가 수를 늘리면 학습 샘플 효율이 개선된다





3-2. Scaling Results on a Time-Basis



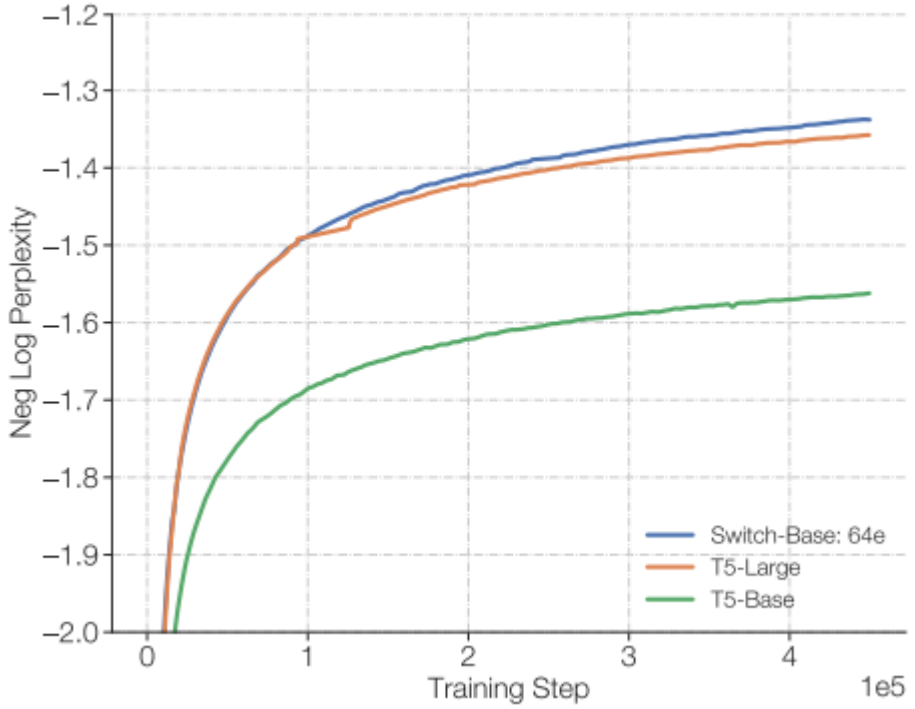
동일한 계산 자원과 학습 시간을

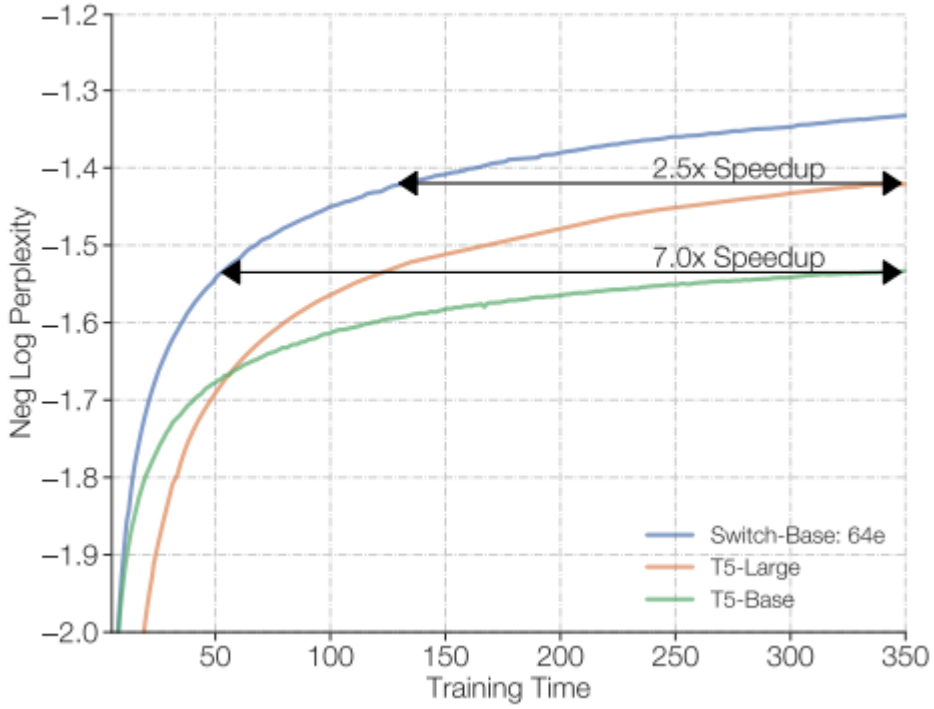
사용했을 때 기존 모델에 비해

훨씬 빠르게 동일한 성능에 도달

3-3. Scaling Versus a Larger Dense Model

- 기존 모델, Large 모델과 비교해도 훨씬 빠르게 동일 성능에 도달한다.





4. Downstream Results

4-1. Fine-Tuning

Model	GLUE	SQuAD	SuperGLUE	Winogrande (XL)
T5-Base	84.3	85.5	75.1	66.6
Switch-Base	**86.7**	**87.2**	**79.5**	**73.3**
T5-Large	87.8	88.1	82.7	79.1
Switch-Large	**88.5**	**88.6**	**84.7**	**83.0**

Model	XSum	ANLI (R3)	ARC Easy	ARC Chal.
T5-Base	18.7	51.8	56.7	**35.5**
Switch-Base	**20.3**	**54.0**	**61.3**	32.8
T5-Large	20.9	56.6	**68.8**	**35.5**
Switch-Large	**22.3**	**58.6**	66.0	**35.5**

Model	CB Web QA	CB Natural QA	CB Trivia QA
T5-Base	26.6	25.8	24.5
Switch-Base	**27.4**	**26.8**	**30.7**
T5-Large	27.7	27.6	29.5
Switch-Large	**31.3**	**29.5**	**36.9**

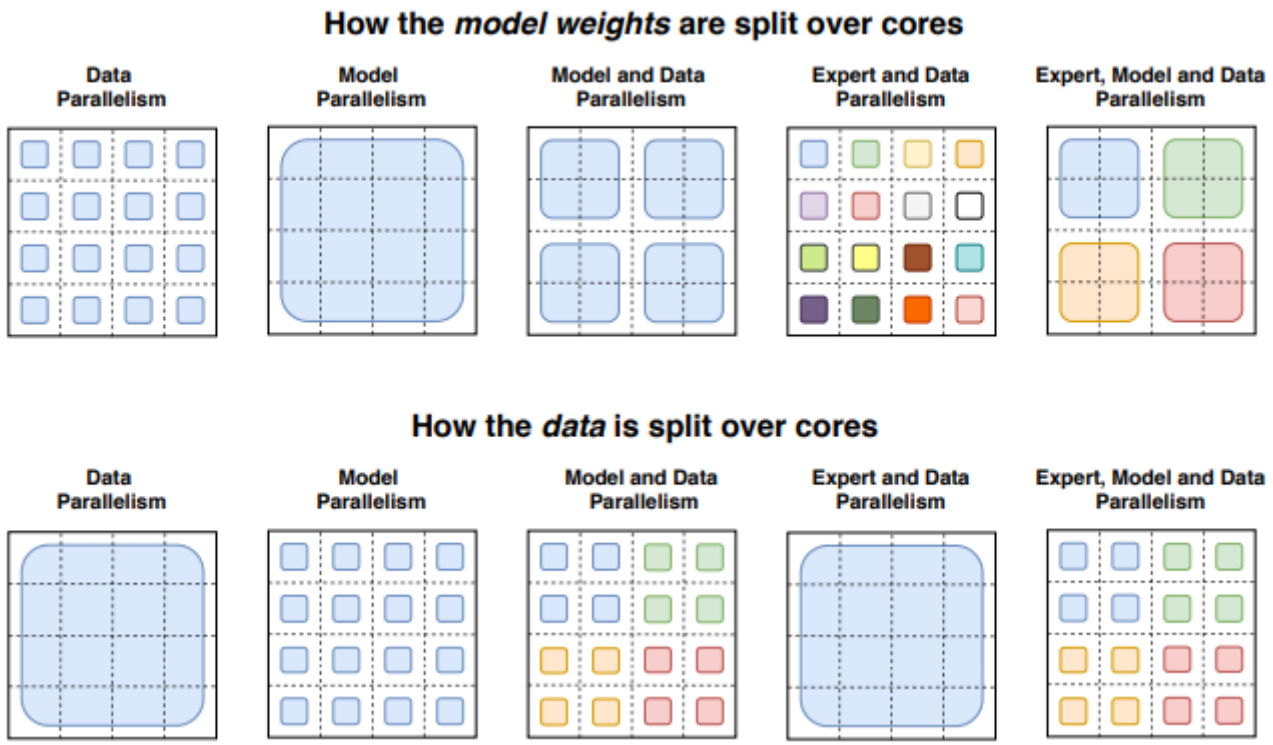
4-2. Distillation

Technique	Parameters	Quality (↑)
T5-Base	223M	-1.636
Switch-Base	3,800M	-1.444
Distillation	223M	(3%) -1.631
+ Init. non-expert weights from teacher	223M	(20%) -1.598
+ 0.75 mix of hard and soft loss	223M	(29%) -1.580
Initialization Baseline (no distillation)		
Init. non-expert weights from teacher	223M	-1.639

4-3. Multilingual Learning

	Dense	Sparse				
Parameters	223M	1.1B	2.0B	3.8B	7.4B	14.7B
Pre-trained Neg. Log Perp. (↑)	-1.636	-1.505	-1.474	-1.444	-1.432	-1.427
Distilled Neg. Log Perp. (↑)	—	-1.587	-1.585	-1.579	-1.582	-1.578
Percent of Teacher Performance	—	37%	32%	30 %	27 %	28 %
Compression Percent	—	82 %	90 %	95 %	97 %	99 %

5. Model Parellelism



6. Conclusion

요약
Switch Transformer는 기존 MoE 모델의 복잡성과 통신 부담을 대폭 줄이면서, 동일한 FLOP 내에서 파라미터 수를 획기적으로 늘릴 수 있음을 보입니다.
결과
전반적으로 pre-training, 파인튜닝, 증류, 다국어 학습 등에서 dense 모델 대비 우수한 성능과 학습 속도 향상을 달성하였습니다.
Future Work
더욱 안정적인 초대형 sparse 모델 학습, 다운스트림 태스크에서의 추가 성능 개선, 그리고 전문가 분할 방식의 최적화 등이 연구 과제로 남아 있습니다.