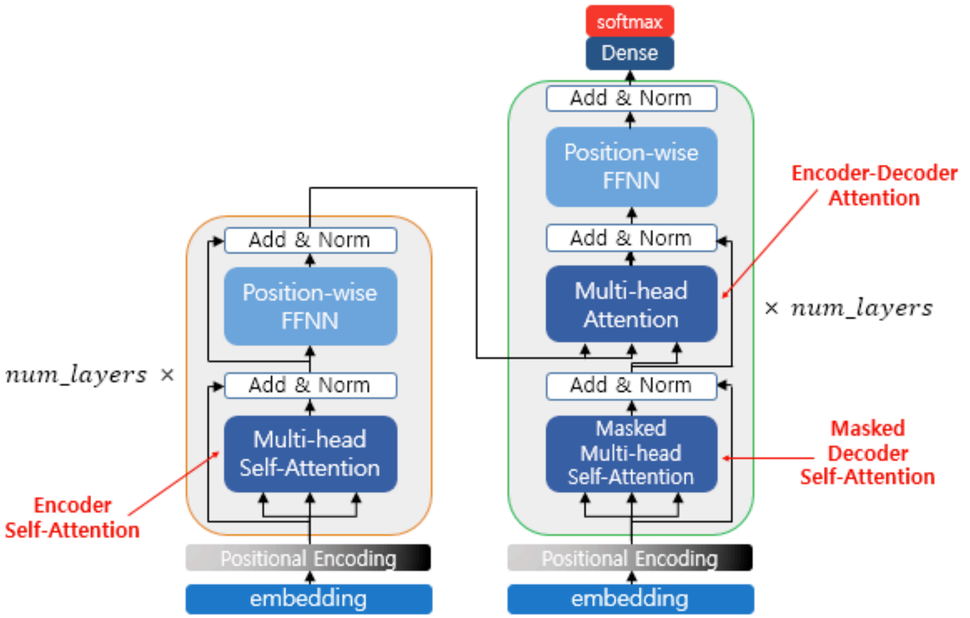
Are Pre-trained Convolutions Better than Pre-trained Transformers?

Tay, Yi, et al. "Are Pre-trained Convolutions Better than Pre-trained Transformers?." *arXiv preprint arXiv:2105.03322* (2021).

Introduction

- 트랜스포머는 현재 사전학습과 뗄 수 없는 관계를 가진 아키텍처. 최근 연구되는 거의 모든 아키텍처는 트랜스포머 기반
- 최근 CNN 기반 모델의 유망함이 제시됨 (Wu et al., 2019; Gehring et al., 2017). 트랜스포머의 필요성 의문을 제기
  - Wu et al.의 cnn + seq2seq은 트랜스포머 보다 LM, 기계번역에서 더 좋은 성능을 보임
  - Are only Transformers able to capitalize on the benefits of pre-training?
- convolutional architectures have not yet been rigorously evaluated under the pretrain-fine-tune paradigm

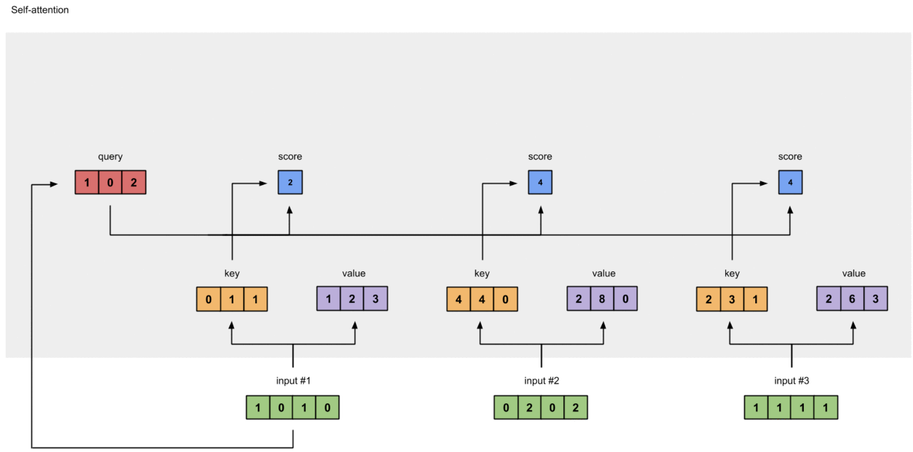


Introduction

- **CNN** 기반 모델이 갖는 장단점

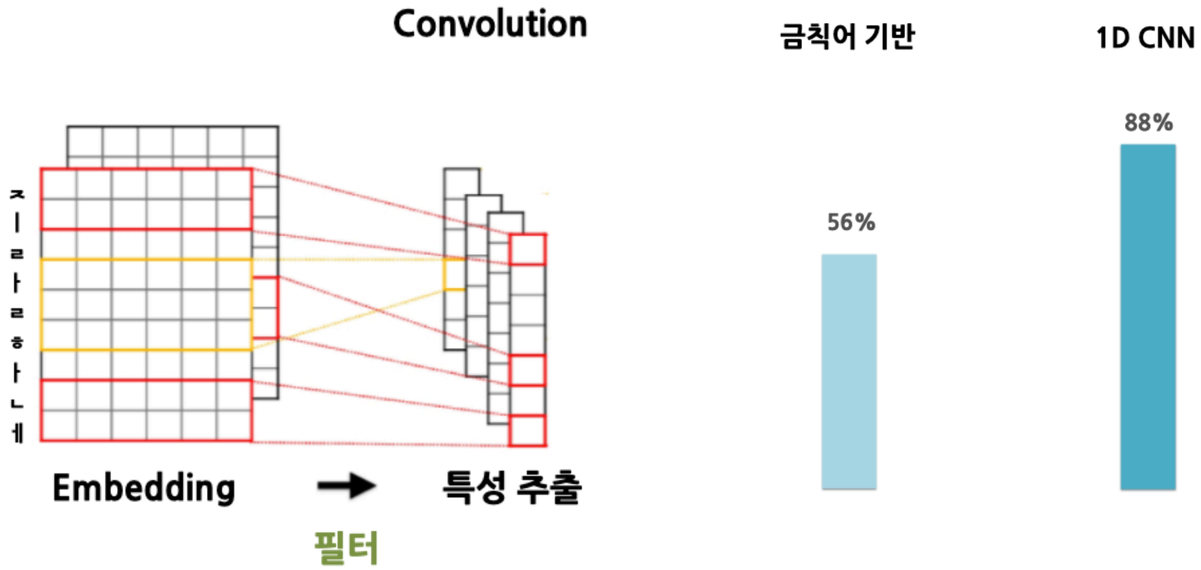
- 장점
  - self-attention 기반 모델과 달리, quadratic memory complexity O(n^2) 문제를 겪지 않는다
  - CNN은 positional encoding에 의존하지 않는다

- 단점
  - global information에 접근하지 못한다. 이 경우, 시퀀스 데이터 내에서 긴 글에 대한 맥락을 고려하지 못하고, 오직 각 토큰의 주변만을 고려할 뿐이다.



Related Work

- 큰 코퍼스로 사전학습 하는 것은 다양한 NLP 태스크를 해결하기 위해 보편적인 언어 표현을 학습시키는 주된 방법이 되었다.
  - Skip-Gram, Glove, ELMO, GPT, BERT 등
- Convolutional Model은 NLP에서 흥미로운 선택이었다. (주로 선택되지 않아왔음) 이것은 학습 및 추론 시 트랜스포머와 달리 가볍고 빠르다는 장점이 있다.



넥슨의 욕설 필터링 사례

Pre-trained Convolution Models

1. 표준적인 CNN 보다 메모리 효율성 향상 (Lightweight Depthwise Convolution)
2. Self-attention의 메모리 비효율성 극복하기 위해 제안됨

3.1 Depth-wise Convolutions

$$O_{i,c} = \text{DepthwiseConv}(X, W_{c,}, i, c) = \sum_{j=1}^{k} W_{c,j} \cdot X_{(i+j-\lceil \frac{k+1}{2} \rceil), c}$$

3.2 Lightweight Convolutions

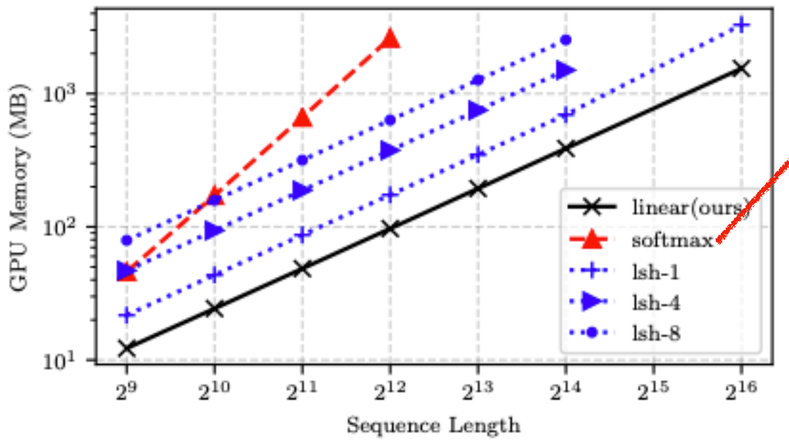
$$\text{LightConv}(X, W_{\lceil \frac{cH}{d} \rceil,:}, i, c) = \text{DepthwiseConv}(X, \text{softmax}(W_{\lceil \frac{cH}{d} \rceil,:}), i, c)$$

3.2.2 Optimization

$$L = \sum_{t=1}^{L} \sum_{i=1}^{n} \log(\pi_i^t) + (1 - y_i^t) \log(1 - \pi_i^t),$$

where π_i^t is the prediction of class i at time step t and y_i^t is the ground truth label of the class i at time step t.

"For many years, softmax has been the bottleneck for training classification models with a large number of categories"



Softmax: 일반적으로 사용하는 self-attention

$$Q = xW_Q,$$
$$K = xW_K,$$
$$V = xW_V,$$
$$A_l(x) = V' = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right) V.$$

그림: Katharopoulos, Angelos, et al. "Transformers are rnns: Fast autoregressive transformers with linear attention." *International Conference on Machine Learning*. PMLR, 2020.

Pre-trained Convolution Models

3.1 Depth-wise Convolutions

- 2D Conv는 기본적으로 3D data (C,H,W)를 다루는 데 사용됨
- Depth wise conv는 normal conv와 연산량은 동일하지만 conv를 조직하는 방식이 다름

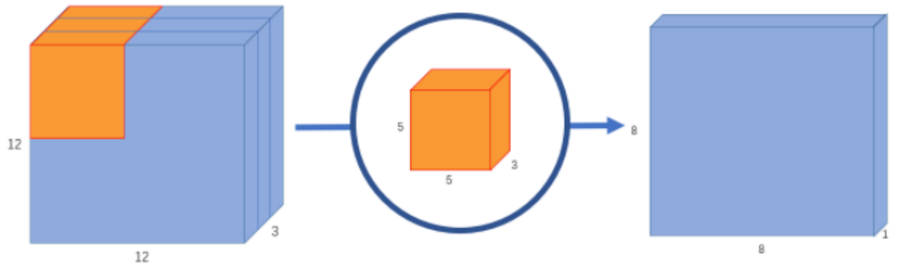


Image 4: A normal convolution with 8×8×1 output

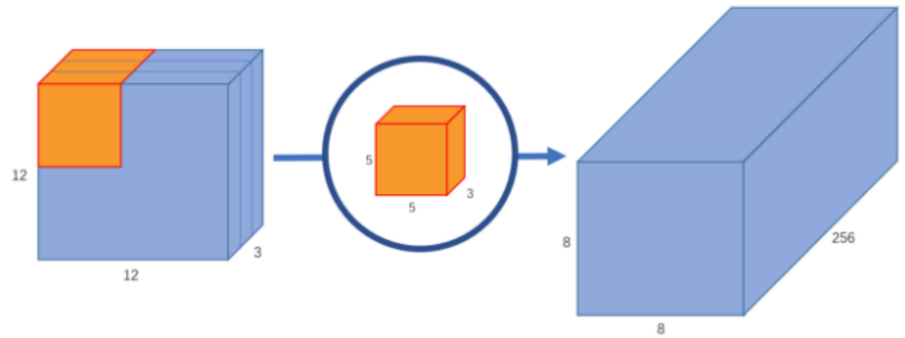


Image 5: A normal convolution with 8×8×256 output

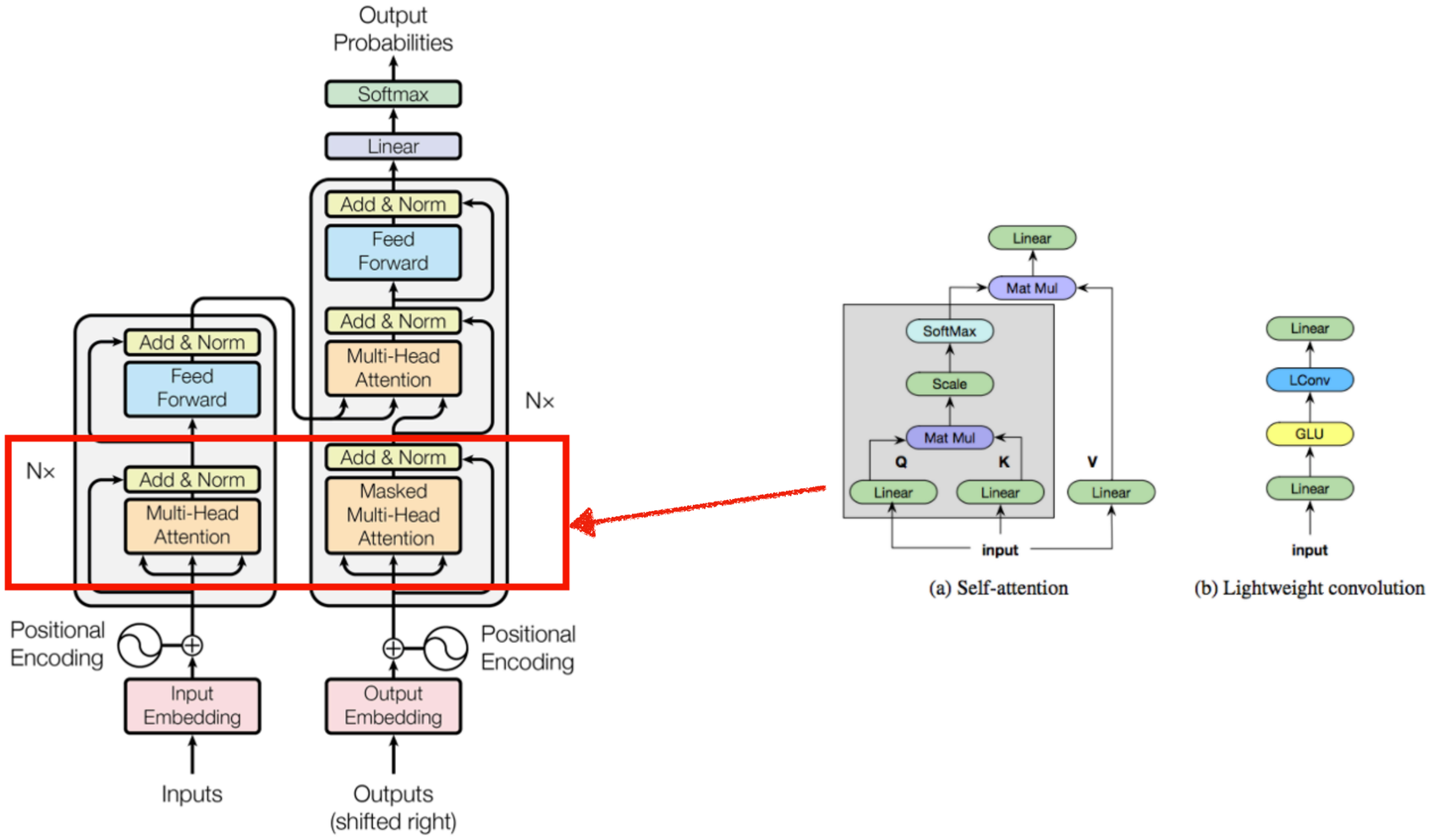
Image 6: Depthwise convolution, uses 3 kernels to transform a 12×12×3 image to a 8×8×3 image

instead of applying convolution of size **d×d×C**, we apply a convolution of size **d×d×1**

: we don't make the convolution computation over all the channels, but only 1 by 1.

Pre-trained Convolution Models

3.2 Convolutional Seq2Seq Architecture



Experiments and Analysis

Datasets

• **Toxicity Detection**	• **Sentiment Classification**	• **News Classification**	• **Question Classification**	• **Semantic Parsing / Compositional Gener- alization**
• WIKI TOXIC SUBTYPES dataset	• IMDb • Stanford Sentiment Treebank • Twitter Sentiment140	• AGNews dataset	• TREC fine-grained question classification dataset	COGS dataset

Dataset / Task	# Train	# Test	# Class
Civil Comments	3,820,210	205,781	2
Wiki Toxicity	561,808	234,564	2
IMDb	25,000	25,000	2
SST-2	67,000	1,800	2
S140	1,600,000	359	2
TREC	4,500	500	46
AGNews	120,000	7,600	4
COGS	24,000	3000	N/A

Experiments and Analysis

Models

	Transformer	CNN
Models	Transformer	- Light-weight Convolution model - Dynamic Convolution model - Dilated model
Pretrained training dataset	Colossal Cleaned Com- monCrawl Corpus	Colossal Cleaned Com- monCrawl Corpus

Experiments and Analysis

Experimental Results

Model	CIVILCOMMENT Acc	CIVILCOMMENT F1	WIKITOXIC Acc	WIKITOXIC F1	IMDb Acc	SST-2 Acc	S140 Acc	TREC Acc	News Acc
No pre-training									
Trans.	77.22	85.09	91.93	95.45	84.81	78.44	58.84	78.00	84.25
Light	78.58	85.82	91.05	94.65	**85.88**	81.65	60.64	**82.20**	**87.22**
Dilat.	**79.94**	**86.50**	**92.29**	94.91	85.84	79.01	55.62	79.60	81.24
Dyna.	78.49	84.71	90.06	**95.66**	85.69	**82.80**	**60.84**	80.20	85.13
With pre-training									
Trans.	81.16	86.56	91.46	95.12	**94.16**	92.09	61.65	93.60	93.54
Light	81.47	87.58	93.61	96.48	93.60	**92.20**	61.65	93.60	93.63
Dilat.	81.67	**87.78**	**93.84**	96.21	93.92	92.09	**62.85**	**94.20**	93.26
Dyna.	**81.83**	87.71	93.76	**96.53**	93.35	91.59	62.45	92.40	**93.93**
Gain from pre-training									
Trans.	+5.1%	+1.7%	-0.6%	-0.4%	+11.0%	+17.4%	+4.7%	+20.0%	+11.0%
Light	+3.7%	+2.1%	+2.8%	+1.9%	+9.0%	+13.0%	+1.7%	+14.0%	+7.3%
Dilat.	+2.1%	+1.5%	+1.7%	+1.4%	+9.4%	+17.0%	+13.0%	+18.0%	+14.8%
Dyn.	+4.3%	+3.5%	+4.1%	+1.0%	+8.9%	+10.6%	+2.6%	+15.2%	+10.4%

Table 2: Comparison of pre-trained Convolutions and pre-trained Transformers on toxicity detection, sentiment classification, question classification and news classification. All models have approximately $230M$ parameters and are 12 layered seq2seq architectures. Our findings show that convolutions (1) also benefit from pretraining and (2) are consistently competitive to transformer models with and without pretraining.

■ 거의 모든 분류 task에서 transformer 보다 CNN 기반 모델이 더 좋은 성능을 보임 (competitive)

■파싱 task: COGS dataset 사용한 경우, trans and CNN 둘다 95% acc

Experiments and Analysis

Summary of Results

Q. 사전학습되지 않은 CNN은 사전학습되지 않은 Transformer 보다 뛰어난가?
A. 7 개 task 중 6 개에서 CNN이 transformer 보다 뛰어났다.

Q. CNN도 Transformer 처럼 사전학습으로부터 이익을 얻는가?
A. 실험 결과에 따르면 그렇다.

Q. 어떤 variant CNN이 좋은가?
A. Lightweight Conv 보다 Dilated Conv와 Dynamic Conv가 일반적으로 더 좋은 성능을 보인다.

Discussion and Analysis

학습 시에 보지 못한(unseen) 데이터도 test 시에 잘 추론할 수 있도록 돕는 가정assumption의 집합

언제 **Pre-trained CNN**이 실패할 것이라고 예상하는가**?**

■ weakness of pre-trained convolutions are their lack of cross-attention inductive bias

☞ 이런 이유 탓에, 2 개 이상의 sequences 간 관계를 모델링해야 하는 task 에서는 pertained CNN을 사용하는 것은 좋은 생각이 아님

Ex) SQuAD, MultiNLI dataset으로 실험한 결과를 볼 때, CNN은 —missing inductive bias 탓에— Transformers 만큼의 성능을 보이지 못했다.

	MNLI	SQuAD
CNN	75% acc	90% F1
Transformer	84% acc	70% F1

What is the **cross attention?**

When query, key and values are generated from same embeddings, it's called **self-attention**.
When query is generated from one embedding and keys and values are generated from another embeddings is called **cross attention**

Discussion and Analysis

Pretrained CNN은 **Transformers** 보다 어떤 점이 더 좋은가**?**

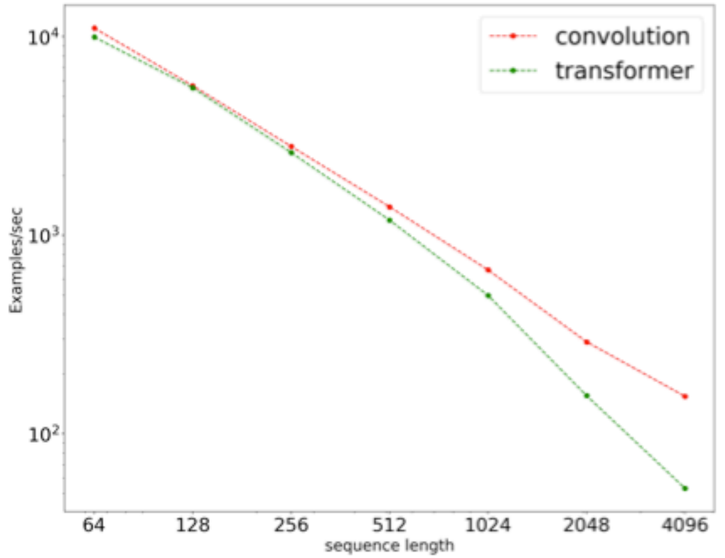


Figure 1: Effect of sequence length on processing speed (examples per second) on a seq2seq masked language modeling task. Results are benchmarked on 16 TPUv3 chips on C4 pre-training. Results are in log scale.

긴 시퀀스를 다룰 때, CNN are faster and scale better

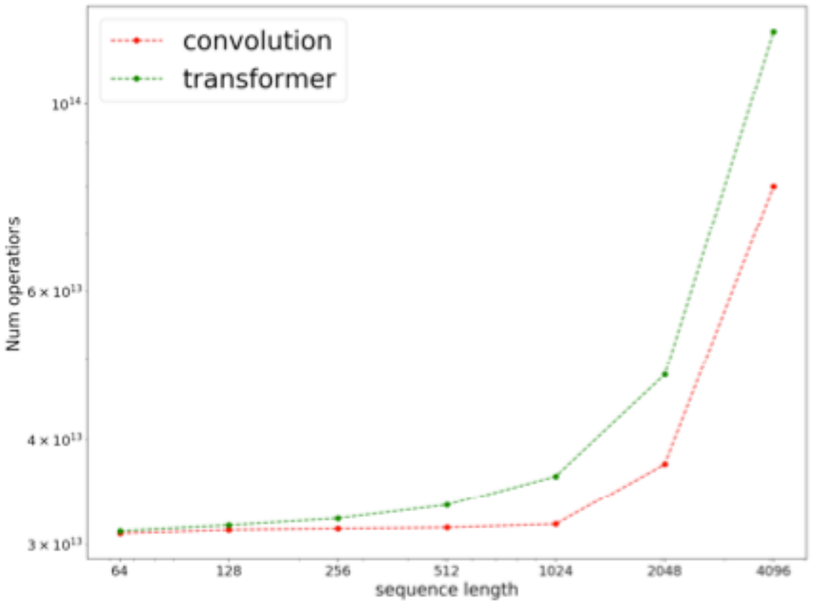


Figure 2: Effect of sequence length on number of FLOPs (einsum ops) on a seq2seq masked language modeling task. Results are benchmarked on 16 TPUv3 chips on C4 pre-training. Results are in log scale.

시퀀스 길이가 어떠하든지 간에, CNN are faster

Summary

- CNN도 사전학습의 이득을 얻을 수 있다.
- 몇 가지 task에서는 CNN이 Transformers 보다 좋은 성능을 낼 수 있다.
- 하지만 2 개 이상의 문장 관계를 다루는 모델, cross-attention을 요구하는 task에서 CNN을 사용하는 것은 좋은 선택이 아니다.

References

[1] Tay, Yi, et al. "Are Pre-trainBetter than Pre-trained Transformers?." arXiv preprint arXiv:2105.03322 (2021)
[2] http://ndcreplay.nexon.com/NDC2018/sessions/NDC2018_0033.html
[3] Katharopoulos, Angelos, et al. "Transformers are rnns: Fast autoregressive transformers with linear attention." International Conference on Machine Learning. PMLR, 2020.
[4] https://seewoo5.tistory.com/5
[5] https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9
[6] https://colab.research.google.com/github/ArthurChen189/ML/blob/master/Copy_of_Copy_of_Ada_Opt_Comparison.ipynb
[7] https://en.wikipedia.org/wiki/Inductive_bias
[8] http://www.secmem.org/blog/2020/01/12/Pay-Less-Attention-with-Lightweight-and-Dynamic-Convolutions-review/
[9] https://tvm.d2l.ai/chapter_common_operators/depthwise_conv.html
[10] https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728
[11] https://www.reddit.com/r/deeplearning/comments/nf08zz/what_is_the_cross_attention/
[12] Wu, Felix, et al. "Pay less attention with lightweight and dynamic convolutions." *arXiv preprint arXiv:1901.10430* (2019).