Pay Attention to MLPs

Liu, Hanxiao, et al.

arXiv preprint arXiv:2105.08050 (2021).

Seungone Kim

Department of Computer Science

Yonsei University

louisdebroglie@yonsei.ac.kr

2021.08.03

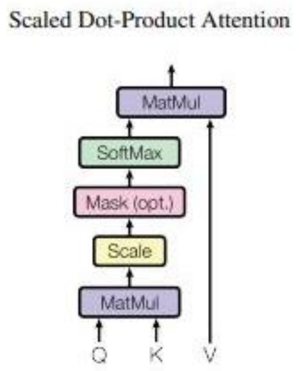
Outline

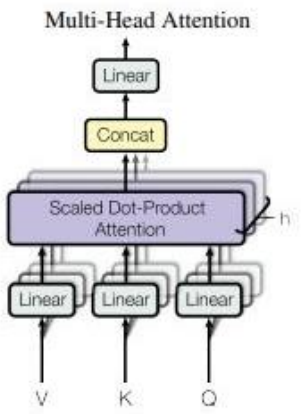
- 문제 정의
- 제안하는 방법
- 관련 연구
- 실험 결과

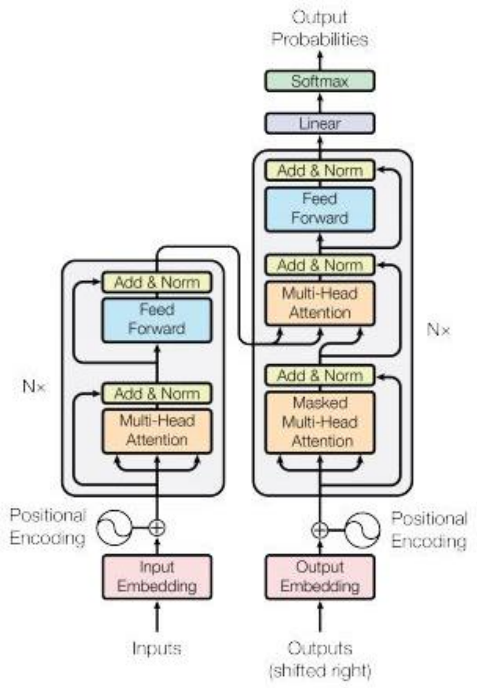
문제 정의

- Transformers combines two important concepts
    - Recurrent-free architecture which allows Parallelism
    - Multi-head Self-Attention Blocks which aggregate spatial information

- It is still unclear what empowers the success of Transformer based Models
    - Feedforward nature of Transformers
    - Multi-head self-attention layers



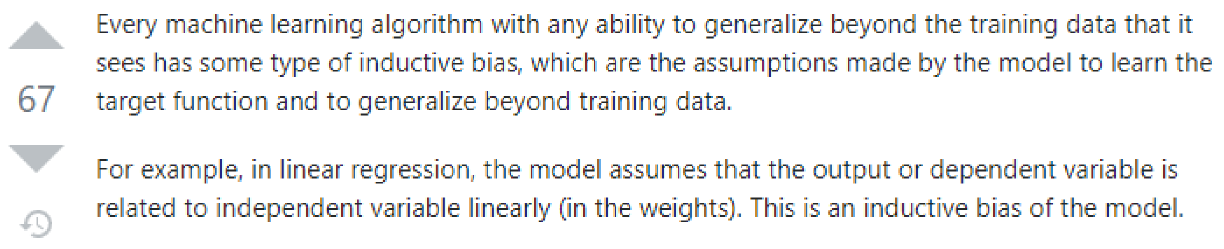




문제 정의

- Attention Mechanism
  - *Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." (ICLR 2015)*
  - Attention Mechanism introduces the inductive bias that the spatial interactions should be dynamically parameterized based on the input representations

- MLPs
  - *Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.*
  - MLPs meanwhile are statically parameterized, which means independent of the input
  - MLPs can represent arbitrary functions


67

Every machine learning algorithm with any ability to generalize beyond the training data that it sees has some type of inductive bias, which are the assumptions made by the model to learn the target function and to generalize beyond training data.

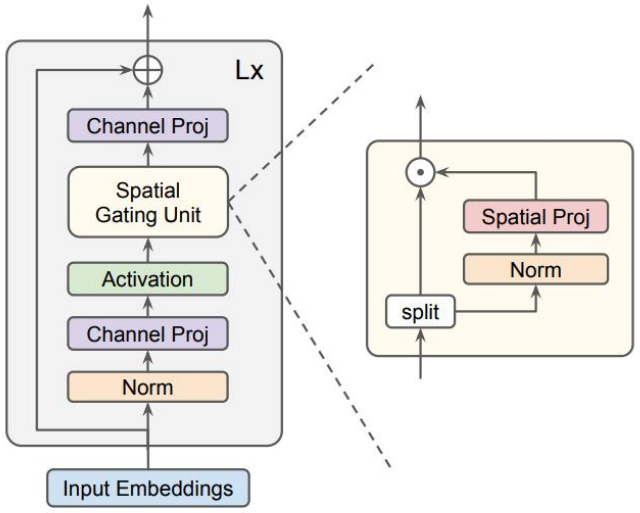
For example, in linear regression, the model assumes that the output or dependent variable is related to independent variable linearly (in the weights). This is an inductive bias of the model.

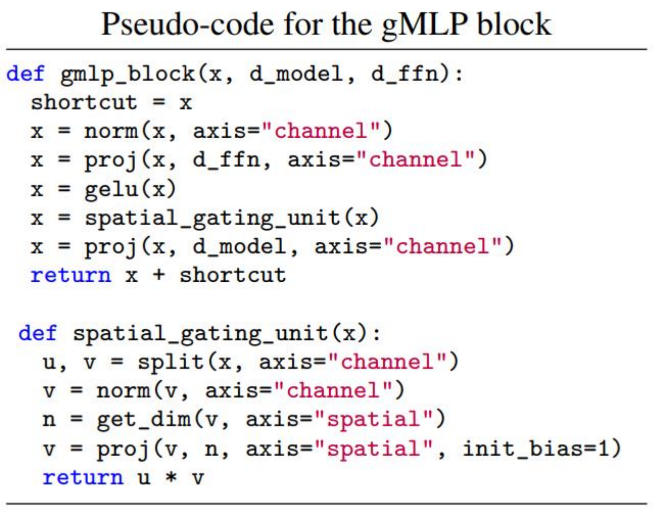
연세대학교 YONSEI UNIVERSITY | SOFT COMPUTING LABORATORY

제안하는 방법

- gMLP
  - MLP-based alternative to Transformers without Self-Attention
  - Simply consists of Channel Projections and Spatial Projections with Static Parameterization
  - Built out of MLP layers with gating

- Model Architecture
  - $Z = \sigma(XU),\ \tilde{Z} = s(Z),\ Y = \tilde{Z}V$
  - $X \in \mathbb{R}^{n\times d},\ U \in \mathbb{R}^{d\times model\_dim},\ V \in \mathbb{R}^{model\_dim\times d},\ \sigma : Gelu,\ s : SGU$





Pseudo-code for the gMLP block

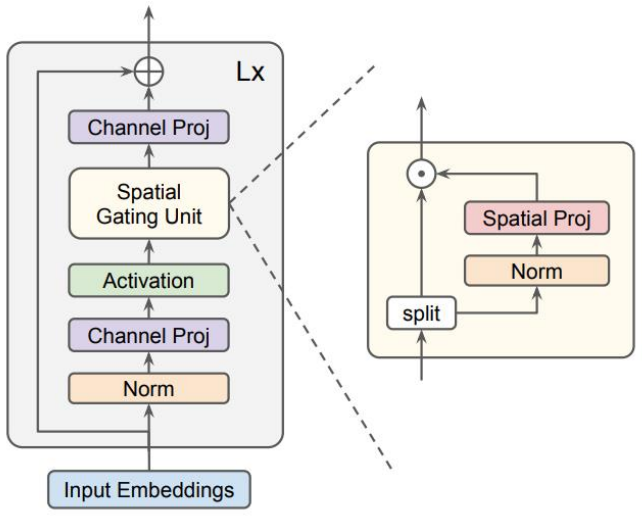
```
def gmlp_block(x, d_model, d_ffn):
  shortcut = x
  x = norm(x, axis="channel")
  x = proj(x, d_ffn, axis="channel")
  x = gelu(x)
  x = spatial_gating_unit(x)
  x = proj(x, d_model, axis="channel")
  return x + shortcut

def spatial_gating_unit(x):
  u, v = split(x, axis="channel")
  v = norm(v, axis="channel")
  n = get_dim(v, axis="spatial")
  v = proj(v, n, axis="spatial", init_bias=1)
  return u * v
```

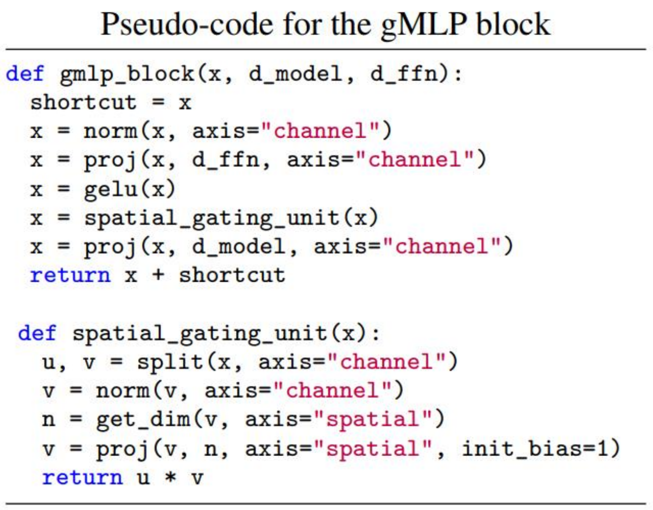
연세대학교 YONSEI UNIVERSITY | SOFT COMPUTING LABORATORY

제안하는 방법

- Spatial Gating Unit (SGU)
  - Statically Parameterized
  - $s(Z) = z_1 \odot f_{W,b}(z_2)$ $where$ $f_{W,b}(z) = WZ + b,$ $W \in \mathbb{R}^{model\_dim \times model\_dim}$
  - Initialize W near zero values and b as ones at the beginning of training
  - gMLP blocks will behave like a regular FFN at early stage of training
  - Gradually injects spatial information across tokens during the course of learning

  - SGUs is an alternative to capture high order relationships other than self-attention
  - 2nd order interactions $(z_i\ z_j)$ where as self-attention composes of 3rd order interaction $(q_i\ k_j\ v_k)$

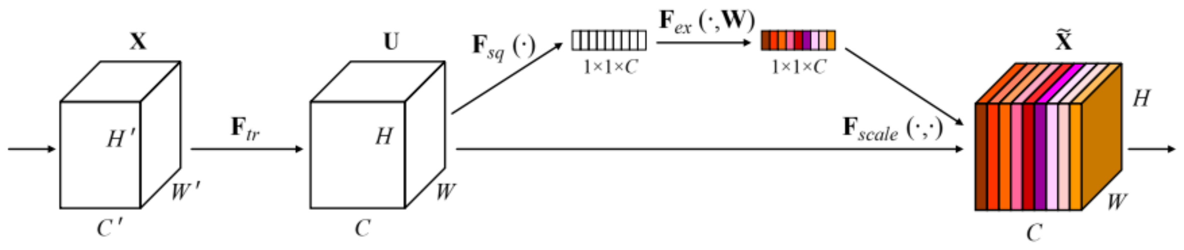



Pseudo-code for the gMLP block

```
def gmlp_block(x, d_model, d_ffn):
  shortcut = x
  x = norm(x, axis="channel")
  x = proj(x, d_ffn, axis="channel")
  x = gelu(x)
  x = spatial_gating_unit(x)
  x = proj(x, d_model, axis="channel")
  return x + shortcut

def spatial_gating_unit(x):
  u, v = split(x, axis="channel")
  v = norm(v, axis="channel")
  n = get_dim(v, axis="spatial")
  v = proj(v, n, axis="spatial", init_bias=1)
  return u * v
```

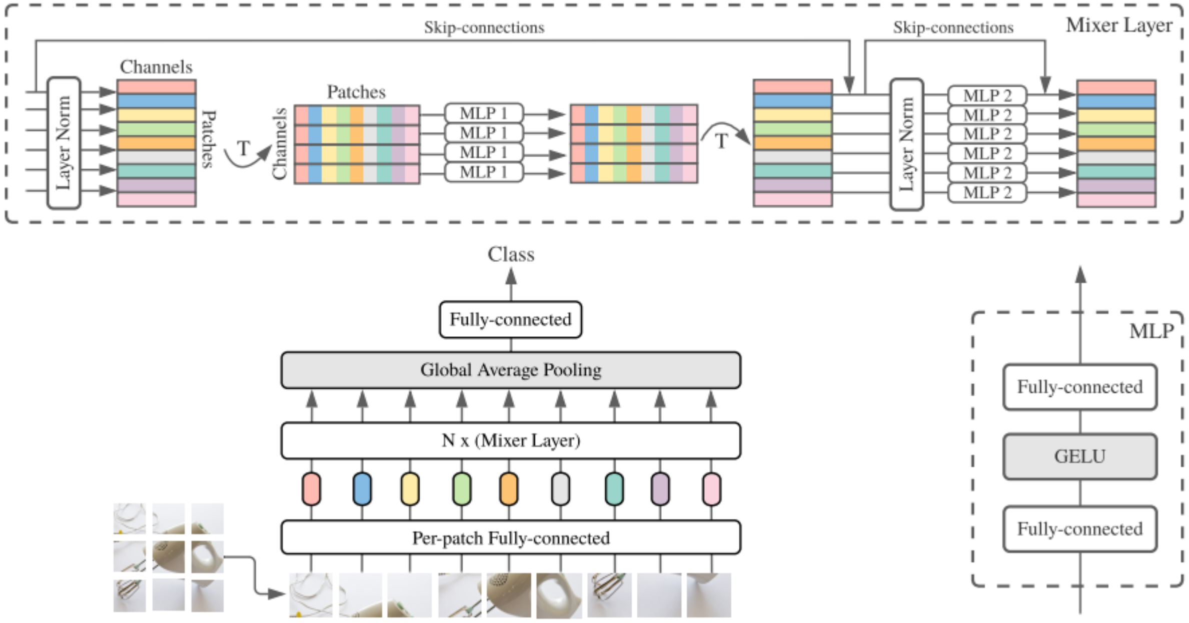
관련 연구

- Gated Linear Units / Highway Networks / LSTM-RNNs
  - SGU projects over spatial(cross-token) dimension rather than channel(hidden) dimension
  - *Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In ICML, 2017*
  - *Noam Shazeer. Glu variants improve transformer. arXiv preprint arXiv:2002.05202, 2020.*
  - *Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In ICLR, 2019.*
  - *Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. arXiv preprint arXiv:1505.00387, 2015.*
  - *Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 1997.*
- Squeeze-and-Excite Blocks
  - *Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018.*
  - Unlike SE Blocks, SGU does not contain cross-channel projections at all, nor does it enforce permutation invariance
  - Unlike typical depthwise CNNs with channel-specific filters, SGU learns only a single transformation shared across channels



관련 연구

- MLP Mixer
  - Tolstikhin, Ilya, et al. "Mlp-mixer: An all-mlp architecture for vision." *arXiv preprint arXiv:2105.01601* (2021).
  - Architecture based exclusively on MLPs, one applied to image patches (mixing per location features) and one applied across patches (mixing spatial information)

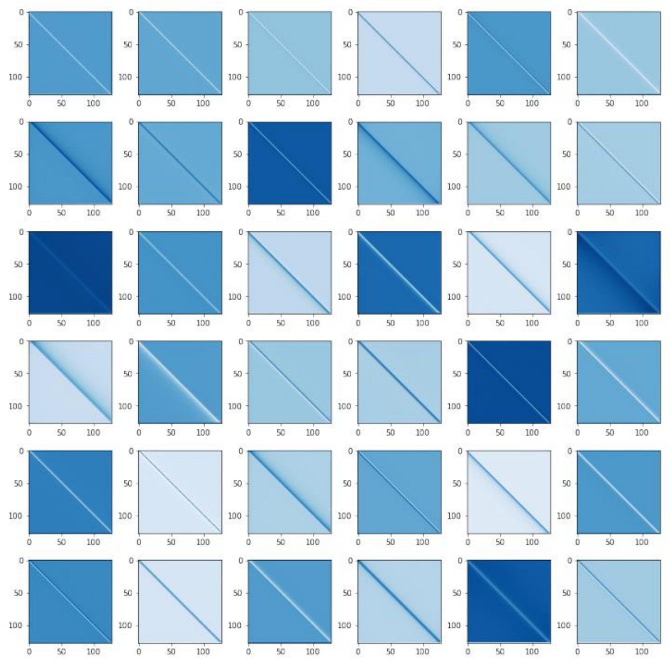


실험 결과

- Masked Language Modeling
  - Do not use Positional Encodings
  - Unnecessary to mask out [PAD] tokens during finetuning as model learns to ignore them
  - Batch size : 2048, Max length : 128, Training Steps : 125k, Pretraining Dataset : C4

- Toeplitz-like matrices
  - gMLPs always learn Toeplitz-like matrices as the spatial weights
  - This means gMLPs are able to learn the notion of Shift Invariance from data
  - Shift Invariance is a property naturally implied by MLM where offset of input sequence does not affect slot filling outcome

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & \cdots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_2 & a_1 & a_0 \end{bmatrix}$$



실험 결과

- Importance of Gating in gMLP for BERT's Pretraining
  - Difference between strongest baseline and gMLPs are insignificant
  - The learned filters appear to be smooth and have several types
  - e.g. forward looking (1st in 2nd row), backward-looking (5th in 2nd row) and bi-directional (2nd last in the last row)

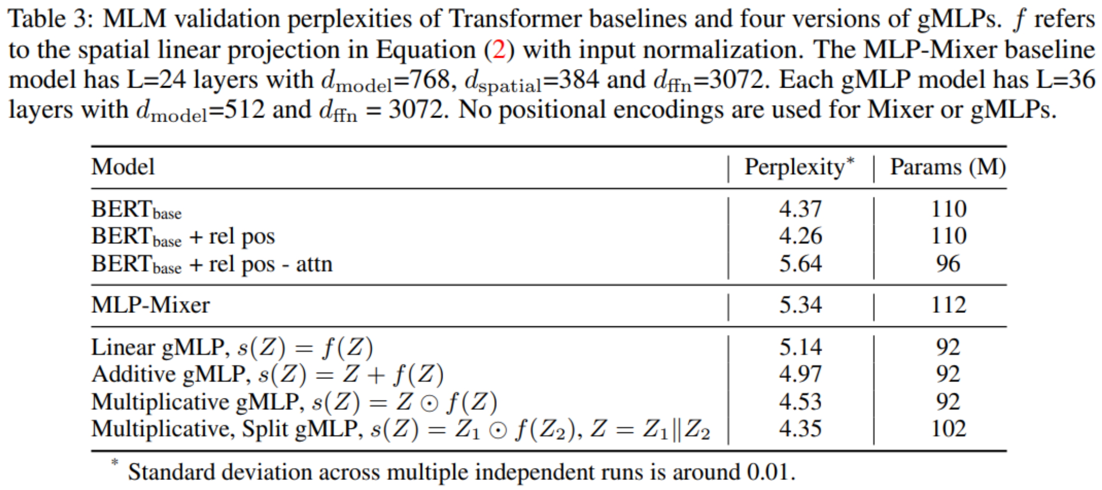
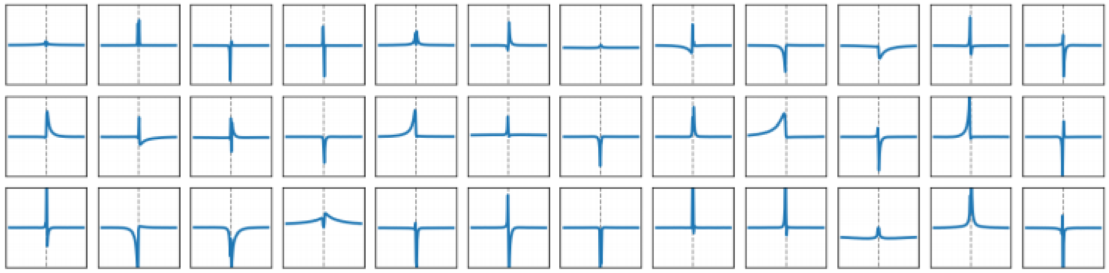


Table 3: MLM validation perplexities of Transformer baselines and four versions of gMLPs. f refers to the spatial linear projection in Equation (2) with input normalization. The MLP-Mixer baseline model has L=24 layers with d_{model}=768, d_{spatial}=384 and d_{ffn}=3072. Each gMLP model has L=36 layers with d_{model}=512 and d_{ffn} = 3072. No positional encodings are used for Mixer or gMLPs.

Model	Perplexity*	Params (M)
BERT$_{\text{base}}$	4.37	110
BERT$_{\text{base}}$ + rel pos	4.26	110
BERT$_{\text{base}}$ + rel pos - attn	5.64	96
MLP-Mixer	5.34	112
Linear gMLP, $s(Z) = f(Z)$	5.14	92
Additive gMLP, $s(Z) = Z + f(Z)$	4.97	92
Multiplicative gMLP, $s(Z) = Z \odot f(Z)$	4.53	92
Multiplicative, Split gMLP, $s(Z) = Z_1 \odot f(Z_2)$, $Z = Z_1 \| Z_2$	4.35	102

* Standard deviation across multiple independent runs is around 0.01.



실험 결과

- Behavior of gMLP as Model size increases
  - Deep enough gMLPs are able to match and even outperform the perplexity of Transformers
  - gMLPs outperform Transformers on SST-2, but are worse on MNLI
  - Suspect the role of self-attention during finetuning is related to cross-sentence alignment

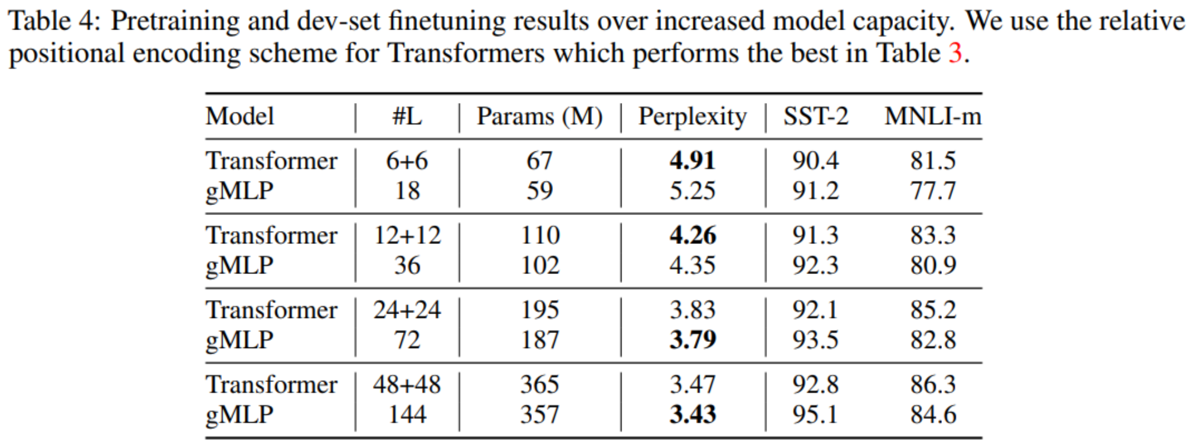
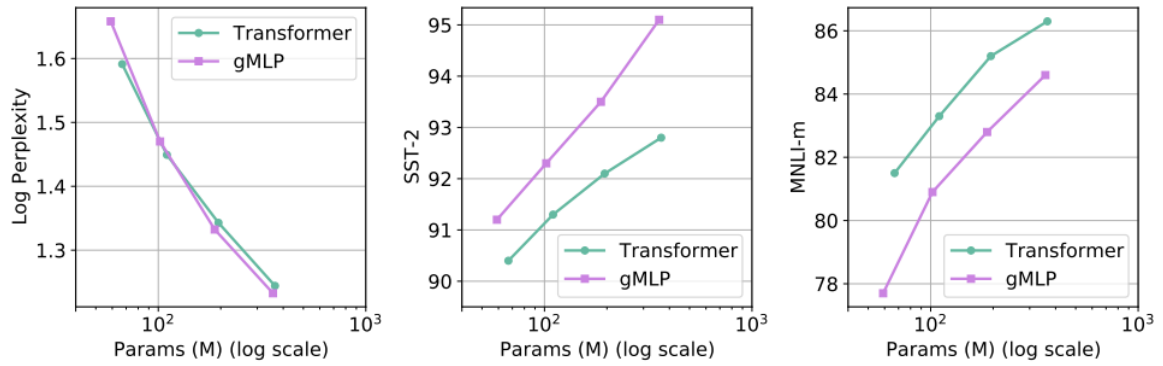


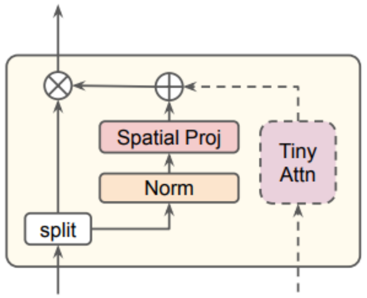
Table 4: Pretraining and dev-set finetuning results over increased model capacity. We use the relative positional encoding scheme for Transformers which performs the best in Table 3.

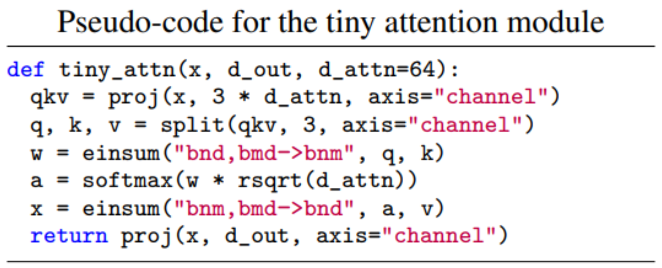
Model	#L	Params (M)	Perplexity	SST-2	MNLI-m
Transformer	6+6	67	**4.91**	90.4	81.5
gMLP	18	59	5.25	91.2	77.7
Transformer	12+12	110	**4.26**	91.3	83.3
gMLP	36	102	4.35	92.3	80.9
Transformer	24+24	195	3.83	92.1	85.2
gMLP	72	187	**3.79**	93.5	82.8
Transformer	48+48	365	3.47	92.8	86.3
gMLP	144	357	**3.43**	95.1	84.6



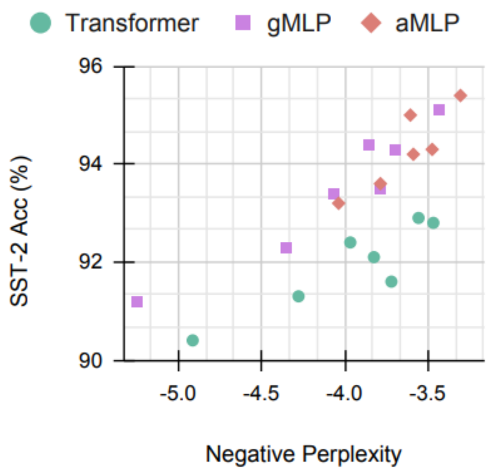
제안하는 방법

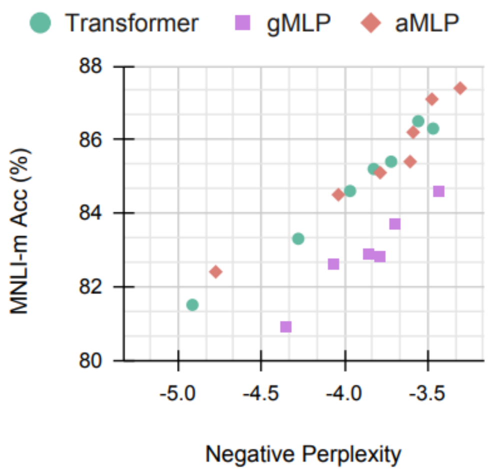
- Usefulness of Tiny Attention in BERT's Finetuning (aMLP)
  - Found self-attention is not a required component to achieve strong MLM perplexity or scalability
  - For MNLI, suspect role of self-attention during finetuning is related to cross-sentence alignment
  - Hypothesize adding a tiny self-attention(size 64) that is very small next to SGU Block could help
  - aMLP consistently outperforms Transformer on both finetuning tasks




Pseudo-code for the tiny attention module

```
def tiny_attn(x, d_out, d_attn=64):
  qkv = proj(x, 3 * d_attn, axis="channel")
  q, k, v = split(qkv, 3, axis="channel")
  w = einsum("bnd,bmd->bnm", q, k)
  a = softmax(w * rsqrt(d_attn))
  x = einsum("bnm,bmd->bnd", a, v)
  return proj(x, d_out, axis="channel")
```



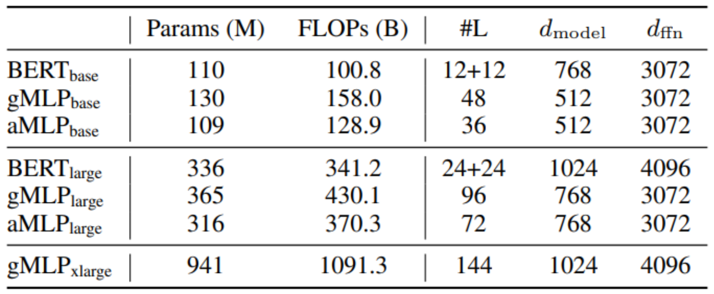


연세대학교 YONSEI UNIVERSITY | SOFT COMPUTING LABORATORY

제안하는 방법

- Main Results for MLM in BERT Setup
  - Performance gap tends to narrow as the model capacity increases
  - gMLP outperforms BERT large on SQuAD 2.0

Table 5: Model specifications in the full BERT setup.



	Params (M)	FLOPs (B)	#L	d_{model}	d_{ffn}
$\text{BERT}_{\text{base}}$	110	100.8	12+12	768	3072
$\text{gMLP}_{\text{base}}$	130	158.0	48	512	3072
$\text{aMLP}_{\text{base}}$	109	128.9	36	512	3072
$\text{BERT}_{\text{large}}$	336	341.2	24+24	1024	4096
$\text{gMLP}_{\text{large}}$	365	430.1	96	768	3072
$\text{aMLP}_{\text{large}}$	316	370.3	72	768	3072
$\text{gMLP}_{\text{xlarge}}$	941	1091.3	144	1024	4096

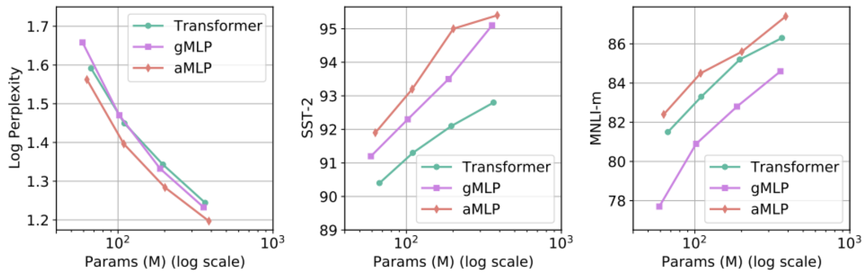
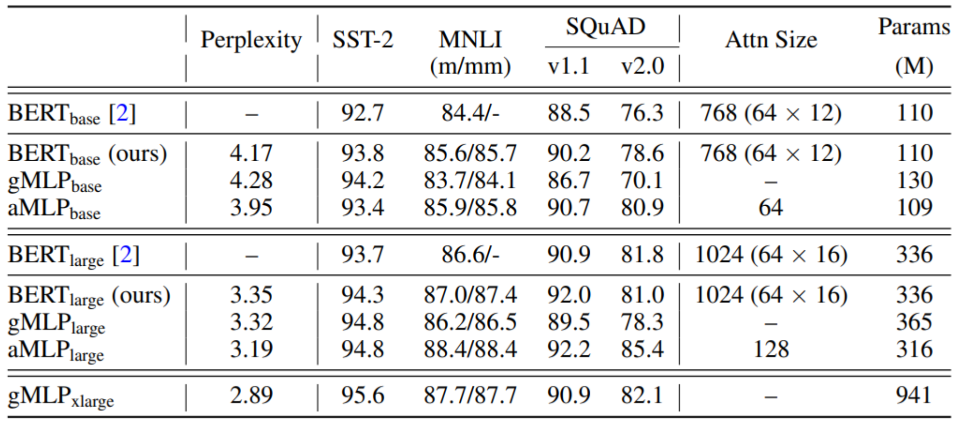


Figure 8: Comparing the scaling properties of Transformers, gMLPs and aMLPs (with 64-d, single-head attention). Results were obtained using the same setup in Section 4.2.



	Perplexity	SST-2	MNLI (m/mm)	SQuAD v1.1	SQuAD v2.0	Attn Size	Params (M)
$\text{BERT}_{\text{base}}$ [2]	–	92.7	84.4/-	88.5	76.3	768 (64 × 12)	110
$\text{BERT}_{\text{base}}$ (ours)	4.17	93.8	85.6/85.7	90.2	78.6	768 (64 × 12)	110
$\text{gMLP}_{\text{base}}$	4.28	94.2	83.7/84.1	86.7	70.1	–	130
$\text{aMLP}_{\text{base}}$	3.95	93.4	85.9/85.8	90.7	80.9	64	109
$\text{BERT}_{\text{large}}$ [2]	–	93.7	86.6/-	90.9	81.8	1024 (64 × 16)	336
$\text{BERT}_{\text{large}}$ (ours)	3.35	94.3	87.0/87.4	92.0	81.0	1024 (64 × 16)	336
$\text{gMLP}_{\text{large}}$	3.32	94.8	86.2/86.5	89.5	78.3	–	365
$\text{aMLP}_{\text{large}}$	3.19	94.8	88.4/88.4	92.2	85.4	128	316
$\text{gMLP}_{\text{xlarge}}$	2.89	95.6	87.7/87.7	90.9	82.1	–	941

결론

- It is still unclear what empowers the success of Transformers
    - Is it the feedforward nature or is it the multi-head self-attention layers?

- gMLPs, a simple variant of MLPs with gating, can be competitive with Transformers
    - Capacity in the multi-head self-attention of Transformers can be largely redundant

- Inductive bias in Transformer's multi-head self-attention are useful on downstream tasks that require cross-sentence alignment
    - Blending small single-head self-attention into gMLP allows for even better architecture without extra model size