ResMLP : Feedforward networks for image classification with data efficient training

School of Industrial and Management Engineering, Korea University

Jong Kook, Heo

KOREA UNIVERSITY

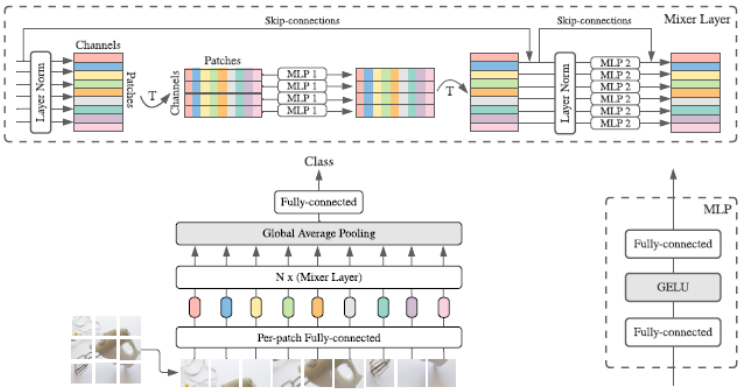
DMQA

Contents

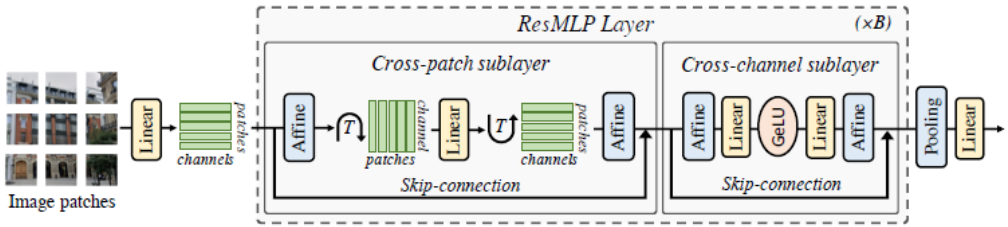
Research Purpose

❖ ResMLP : Feedforward networks for image classification with data-efficient training

- Facebook AI Resarch 에서 연구, 2021년 11월 13일 기준 약 13회 인용
- MLP Mixer(Tolstikhin et al, 2021)* 와 상당히 유사한 구조
  - ✓ Cross-patch sublayer : 각 채널마다 독립적으로 모든 패치에 대해 연산(= Token-mixing in MLP Mixer)
  - ✓ Cross-channel sublayer : 각 패치마다 독립적으로 모든 채널에 대해 연산(= channel-mixing in MLP Mixer)
- LayerNormalization 대신 Affine Transform 사용



MLP Mixer



ResMLP

Overview

ResMLP

❖ Main Observation

- Attention Block 없이 단순히 MLP Layer로만 이루어진 구조이지만, accuracy/complexity trade-off 상에서 상당히 좋은 결과를 나타냄
  - ✓ MLP Mixer 와 거의 같은 시기에 연구된 동일한 구조
  - ✓ 저자 왈 "MLP Mixer는 ImageNet-22k 와 JFT-300M 으로 훈련시킨 매우 큰 모델이지만, 우린 ImageNet-1k 로 훈련시켜 더 가볍고 Inference Time 이 빠르다는 것이 장점"

- 해당 구조는 이미지 뿐만 아니라 다른 도메인에서도 적용 가능
  - ✓ 기계 번역 벤치마크 WMT 에서 seq2seq Transformers 와 견줄만한 성능을 나타냈다고 함

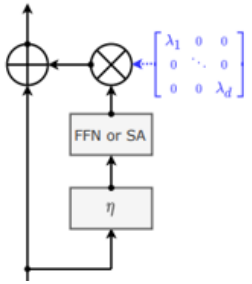
- 기존에 연구된 Distillation 이나 Self-SL 방법론을 적용하면 성능이 올라감

Overview

ResMLP

❖ The Residual Multi-Perceptron Layer

- LayerNormalization 대신에 learnable parameter 로 이루어진 Affine Transform 적용(CaiT* 에서 차용)
  - ✓ CaiT 에서 쓰였던 LayerScale 은 Layer Normalization -> SA/FFN Block -> Channel-wise 가중치를 준 후 Residual Connection

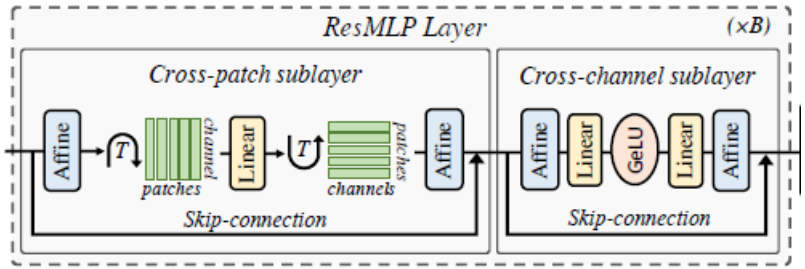


$$x'_l = x_l + \mathrm{diag}(\lambda_{l,1}, \ldots, \lambda_{l,d}) \times \mathrm{SA}(\eta(x_l))$$
$$x_{l+1} = x'_l + \mathrm{diag}(\lambda'_{l,1}, \ldots, \lambda'_{l,d}) \times \mathrm{FFN}(\eta(x'_l))$$

η : *LayerNorm*
Λ : *LayerScale*

LayerScale Block in CaiT

  - ✓ **Self-Attention 연산은 learning rate warm-up 이나 Layernorm 이 없으면 초기 학습이 매우 불안정!(CaiT 참고)**
  - ✓ **ResMLP 는 Self-Attention 연산이 없기 때문에 Layernorm 필요없이** Bias(β)가 추가된 Affine Transform 만 사용
  - ✓ 각 블록의 첫번째 Affine 은 기존의 Layernorm 역할(pre-norm), 대신 channel-wise statistic 이 필요없음
  - ✓ 각 블록의 두번째 Affine 은 CaiT 의 LayerScale 역할(post-norm)



$$\mathtt{Aff}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}) = \mathtt{Diag}(\boldsymbol{\alpha})\mathbf{x} + \boldsymbol{\beta},$$
$$\mathbf{Z} = \mathbf{X} + \mathtt{Aff}\left((\mathbf{A}\,\mathtt{Aff}(\mathbf{X})^\top)^\top\right),$$
$$\mathbf{Y} = \mathbf{Z} + \mathtt{Aff}\left(\mathbf{C}\,\mathtt{GELU}(\mathbf{B}\,\mathtt{Aff}(\mathbf{Z}))\right),$$

Affine Transform in ResMLP

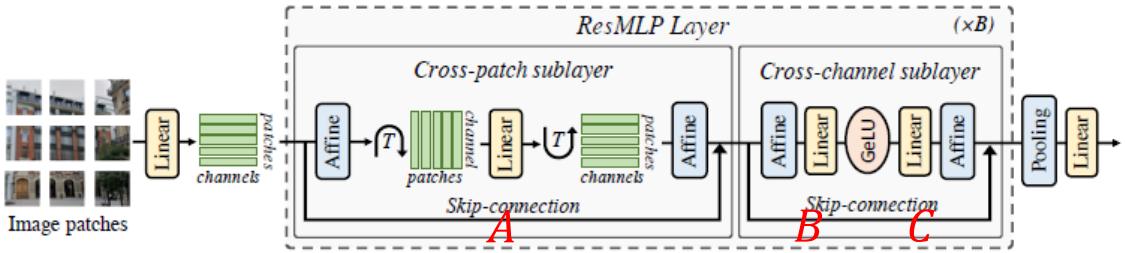
Overview

ResMLP

❖ The Residual Multi-Perceptron Layer

- Details
  - ✓ Cross-patch sublayer : Transformer 의 SA layer 역할(패치 간의 정보 교환)
  - ✓ Cross-channel sublayer : Transformer 의 FFN layer 역할(차원을 4배로 늘렸다가 줄이는 것까지 동일하게 적용)

- Difference with ViT architecture
  - ✓ Self-Attention, Positional embedding 존재X
  - ✓ CLS Token 을 사용하지 않고, 패치 임베딩 값에 대해 Average Pooling하여 사용
  - ✓ 배치 통계량을 이용한 정규화가 아닌 learnable parameter 사용



$$\mathbf{Z} = \mathbf{X} + \mathtt{Aff}\left(\left(\mathbf{A}\,\mathtt{Aff}\left(\mathbf{X}\right)^{\top}\right)^{\top}\right),$$
$$\mathbf{Y} = \mathbf{Z} + \mathtt{Aff}\left(\mathbf{C}\,\mathtt{GELU}(\mathbf{B}\,\mathtt{Aff}(\mathbf{Z}))\right),$$

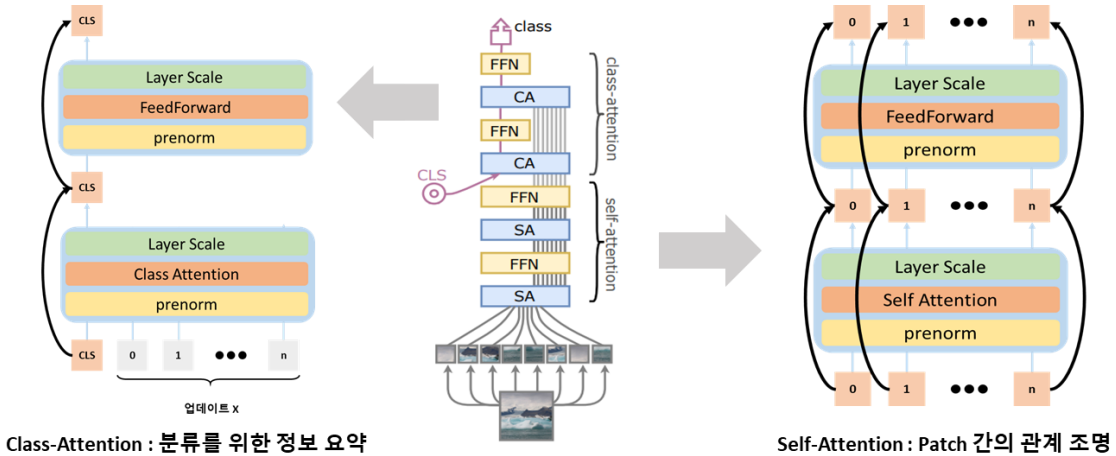
$$A \in R^{N^2 * N^2}$$
$$B \in R^{d*4d}$$
$$C \in R^{4d*d}$$

Overview

ResMLP

❖ Variants of ResMLP

- Class-MLP(An alternative to average pooling)
  - ✓ CaiT 의 Class Attention Layer(CA) 에서 차용(CLS Token 을 마지막에 넣어 Frozen embedding 으로부터 클래스 정보 요약)
  - ✓ Average Pooling 대신에 CLS Token 을 삽입한 후 패치 임베딩과 함께 Linear Layer 에 입력
  - ✓ FAIR github, Timm Library 에서는 해당 구현체 X



Class-Attention : 분류를 위한 정보 요약

Self-Attention : Patch 간의 관계 조명

CaiT Architecture

Overview

ResMLP

❖ Variants of ResMLP

- Seq2Seq ResMLP(For Machine Translation)
  - ✓ Self-Attention Layer 대신 ResMLP layer 를 쓴 인코더-디코더 구조
  - ✓ Vanilla Transformer 처럼 디코더에서 cross-attention layer 를 적용하여 인코더 아웃풋에 어텐션을 주었다고 하나 **구체적인 그림 설명이나 구현체는 없음**
  - ✓ **디코더의 Matrix A 는 Masked Attention 처럼 뒤의 토큰에 영향을 받지 않도록 Triangular Matrix 로 제약**
  - ✓ Variable Sequence Length 에 적용할 수 있도록 배치 내의 가장 긴 시퀀스 길이에 맞춰 zero padding 한 후, **submatrix A 를 추출**

Experiments

ResMLP

❖ Supervised Learning Results(ImageNet-1k Validation Set)

- ConvNet vs ViTs vs ResMLP
  - ✓ V100-32GB GPU 로 batch-size 32 고정
  - ✓ 이미지 사이즈(default): 14 by 14 patches of size 16 by 16
  - ✓ 정확도, throughput, FLOPS 등 다양한 지표의 trade-off 비교
  - ✓ 기존의 ConvNet 이나 ViT 계열의 성능보다 완전히 우세하지 않지만, 그래도 높은 정확도를 나타냄

    "충분한 데이터와 학습 스키마가 존재한다면 구조적 제약이 성능에 큰 영향을 미치지 않는다"

	Arch.	#params ($\times 10^6$)	throughput (im/s)	FLOPS ($\times 10^9$)	Peak Mem (MB)	Top-1 Acc.
State of the art	CaiT-M48↑448ϒ [57]	356	5.4	329.6	5477.8	86.5
	NfNet-F6 SAM [6]	438	16.0	377.3	5519.3	86.5
Convolutional networks	EfficientNet-B3 [53]	12	661.8	1.8	1174.0	81.1
	EfficientNet-B4 [53]	19	349.4	4.2	1898.9	82.6
	EfficientNet-B5 [53]	30	169.1	9.9	2734.9	83.3
	RegNetY-4GF [47]	21	861.0	4.0	568.4	80.0
	RegNetY-8GF [47]	39	534.4	8.0	841.6	81.7
	RegNetY-16GF [47]	84	334.7	16.0	1329.6	82.9
Transformer networks	DeiT-S [56]	22	940.4	4.6	217.2	79.8
	DeiT-B [56]	86	292.3	17.5	573.7	81.8
	CaiT-XS24 [57]	27	447.6	5.4	245.5	81.8
Feedforward networks	ResMLP-S12	15	1415.1	3.0	179.5	76.6
	ResMLP-S24	30	715.4	6.0	235.3	79.4
	ResMLP-B24	116	231.3	23.0	663.0	81.0

Experiments

ResMLP

❖ Self-Supervised Learning with DINO

- ResNet vs ViTs vs ResMLP
  - ✓ DINO 로 300 epoch 학습 후 Linear Evaluation 과 kNN Classifier 성능 비교(ImageNet-1k val)
  - ✓ ViT 에 비해 성능이 떨어지지만, kNN evaluation 에서는 ConvNet 과 pure MLP architecture 를 뛰어넘음
  - ✓ Pretraining 을 한 후 finetuning 한 모델이 지도 학습만으로 학습 시킨 모델보다 Acc 가 0.5% 높음(ResMLP-S24 기준)

Models	ResNet-50	ViT-S/16	ViT-S/8	ViT-B/16	ResMLP-S12	ResMLP-S24
Params. ($\times 10^6$)	25	22	22	87	15	30
FLOPS ($\times 10^9$)	4.1	4.6	22.4	17.5	3.0	6.0
Linear	75.3	77.0	79.7	78.2	67.5	72.8
k-NN	67.5	74.5	78.3	76.1	62.6	69.4

Experiments

ResMLP

❖ Knowledge distillation setting and Ablations

- Knowledge Distillation
  - ✓ RegNet 을 Teacher Model로 distillation 한 모델이 baselines 보다 더 우수한 성능을 나타냄(파란색)

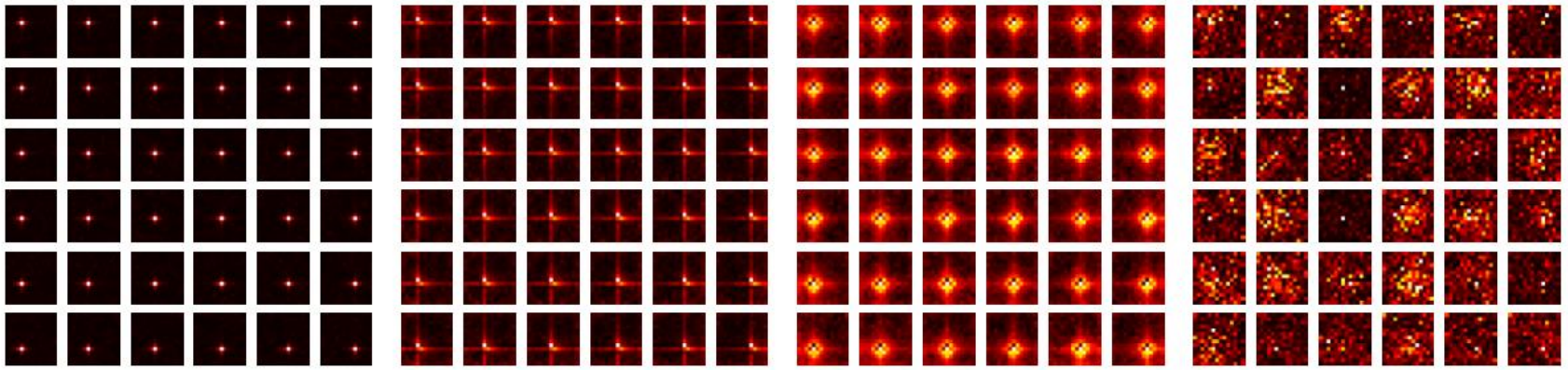
Ablation	Model	Patch size	Params $\times 10^6$	FLOPs $\times 10^9$	Variant	top-1 acc. on ImageNet val	real [4]	v2 [49]
Baseline models	ResMLP-S12	16	15.4	3.0	12 layers, working dimension 384	76.6	83.3	64.4
	ResMLP-S24	16	30.0	6.0	24 layers, working dimension 384	79.4	85.3	67.9
	ResMLP-B24	16	115.7	23.0	24 layers, working dimension 768	81.0	86.1	69.0
Normalization	ResMLP-S12	16	15.4	3.0	`Aff` → Layernorm	77.7	84.1	65.7
Pooling	ResMLP-S12	16	17.7	3.0	average pooling → Class-MLP	77.5	84.0	66.1
Patch communication	ResMLP-S12	16	14.9	2.8	linear → none	56.5	63.4	43.1
	ResMLP-S12	16	18.6	4.3	linear → MLP	77.3	84.0	65.7
	ResMLP-S12	16	30.8	6.0	linear → conv 3x3	77.3	84.4	65.7
	ResMLP-S12	16	14.9	2.8	linear → conv 3x3 depth-wise	76.3	83.4	64.6
	ResMLP-S12	16	16.7	3.2	linear → conv 3x3 depth-separable	77.0	84.0	65.5
Patch size	ResMLP-S12/14	14	15.6	4.0	patch size 16×16→14×14	76.9	83.7	65.0
	ResMLP-S12/8	8	22.1	14.0	patch size 16×16→8×8	79.1	85.2	67.2
	ResMLP-B24/8	8	129.1	100.2	patch size 16×16→8×8	81.0	85.7	68.6
Training	ResMLP-S12	16	15.4	3.0	old-fashioned (90 epochs)	69.2	76.0	56.1
	ResMLP-S12	16	15.4	3.0	pre-trained SSL (DINO)	76.5	83.6	64.5
	ResMLP-S12	16	15.4	3.0	distillation	77.8	84.6	66.0
	ResMLP-S24	16	30.0	6.0	pre-trained SSL (DINO)	79.9	85.9	68.6
	ResMLP-S24	16	30.0	6.0	distillation	80.8	86.6	69.8
	ResMLP-B24/8	8	129.1	100.2	distillation	83.6	88.4	73.4
	ResMLP-B24/8	8	129.1	100.2	pre-trained ImageNet-21k (60 epochs)	84.4	88.9	74.2

Experiments

ResMLP

❖ Knowledge distillation setting and Ablations

- Ablations – Patch Communication
  - ✓ Cross-patch sublayer 의 matrix A의 일부 row만 가져와 시각화
  - ✓ Convolution Filter 와 상당히 유사하며, depth 가 커질수록 전역적인 부분 고려
  - ✓ **그렇다면 Patch Communication 을 3 by 3 Conv 쓰는 것에 비해 어떤 점이 좋을까??**



Experiments

ResMLP

❖ Sparsity

- Ablations – Patch Communication
  - ✓ 3 by 3 Convolution 을 쓴 variant가 가장 좋은 결과를 나타내었지만, MLP-variant 나 Baselines 와 큰 차이는 없음
  - ✓ **3 by 3 Convolution 을 쓸 경우 FLOPs 와 parameter 수 증가가 거의 2배**

Ablation	Model	Patch size	Params $\times 10^6$	FLOPs $\times 10^9$	Variant	top-1 acc. on ImageNet val	real [4]	v2 [49]
Baseline models	ResMLP-S12	16	15.4	3.0	12 layers, working dimension 384	76.6	83.3	64.4
	ResMLP-S24	16	30.0	6.0	24 layers, working dimension 384	79.4	85.3	67.9
	ResMLP-B24	16	115.7	23.0	24 layers, working dimension 768	81.0	86.1	69.0
Normalization	ResMLP-S12	16	15.4	3.0	`Aff` → Layernorm	77.7	84.1	65.7
Pooling	ResMLP-S12	16	17.7	3.0	average pooling → Class-MLP	77.5	84.0	66.1
Patch communication	ResMLP-S12	16	14.9	2.8	linear → none	56.5	63.4	43.1
	ResMLP-S12	16	18.6	4.3	linear → MLP	77.3	84.0	65.7
	ResMLP-S12	16	30.8	6.0	linear → conv 3x3	77.3	84.4	65.7
	ResMLP-S12	16	14.9	2.8	linear → conv 3x3 depth-wise	76.3	83.4	64.6
	ResMLP-S12	16	16.7	3.2	linear → conv 3x3 depth-separable	77.0	84.0	65.5
Patch size	ResMLP-S12/14	14	15.6	4.0	patch size 16×16→14×14	76.9	83.7	65.0
	ResMLP-S12/8	8	22.1	14.0	patch size 16×16→8×8	79.1	85.2	67.2
	ResMLP-B24/8	8	129.1	100.2	patch size 16×16→8×8	81.0	85.7	68.6
Training	ResMLP-S12	16	15.4	3.0	old-fashioned (90 epochs)	69.2	76.0	56.1
	ResMLP-S12	16	15.4	3.0	pre-trained SSL (DINO)	76.5	83.6	64.5
	ResMLP-S12	16	15.4	3.0	distillation	77.8	84.6	66.0
	ResMLP-S24	16	30.0	6.0	pre-trained SSL (DINO)	79.9	85.9	68.6
	ResMLP-S24	16	30.0	6.0	distillation	80.8	86.6	69.8
	ResMLP-B24/8	8	129.1	100.2	distillation	83.6	88.4	73.4
	ResMLP-B24/8	8	129.1	100.2	pre-trained ImageNet-21k (60 epochs)	84.4	88.9	74.2

Experiments

ResMLP

❖ Sparsity

- 각 Linear Layer 에서 최대값의 절대값 대비 5% 미만의 가중치의 비율
  - ✓ 저자 왈 " Layer 의 Sparsity 가 높기 때문에, parameter pruning 이나 Quant-Noise, DiffQ 같은 quantization 방법을 적용할 수도 있다."

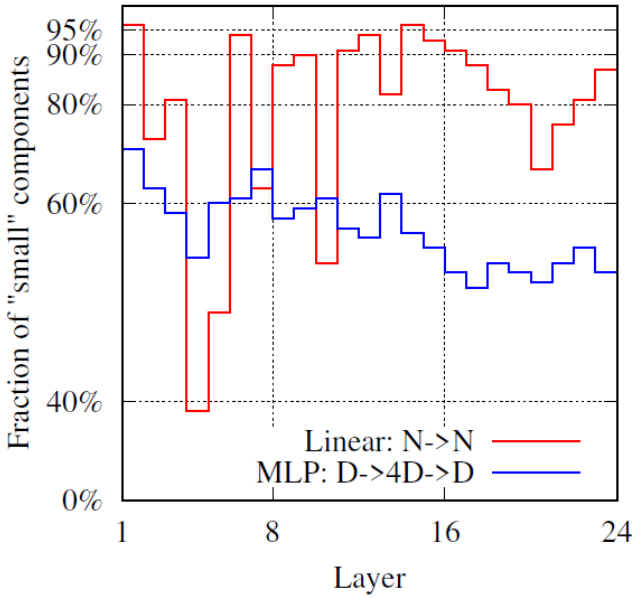


Figure 3: **Sparsity of linear interaction layers.** For each layer (linear and MLP), we show the rate of components whose absolute value is lower than 5% of the maximum. Linear interaction layers are sparser than the matrices involved in the per-patch MLP.

Conclusion

CaiT

- MLP Mixer 와 거의 같은 구조이기 때문에 큰 차별점은 느끼지 못했음

- CNN이나 기존 ViT 계열과 비교해봤을 때, 약간 성능은 떨어지지만 throughput 이나 Peak Memory 측면에서 확실히 가볍다는 것에 장점이 있어보임.

- ResMLP는 자연어 등 다른 도메인에도 사용할 수 있다고 주장하여 기계 번역 태스크에 대해 실험을 진행하였지만, 해당 variant architecture 나 실험에 대한 설명이 빈약한 것이 아쉬웠음.