MLP-Mixer: An all-MLP Architecture for Vision Ilya

Radek Bartyzal

GLAMI AI

15. 6. 2021

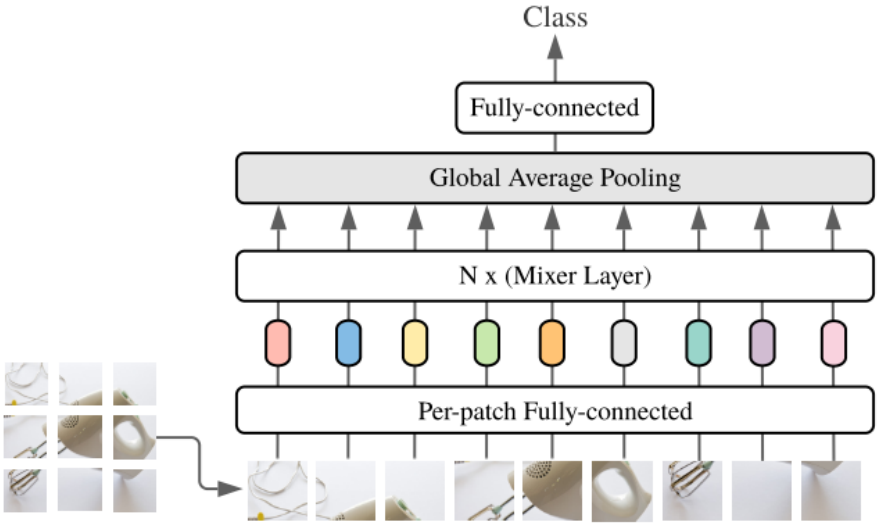
Motivation

Paper is by Google Brain from 2021.

Goal: Learn useful representations of images.

Contribution: Competitive model using only MLPs = fully connected FF layers with non-linearities.

MLP-Mixer

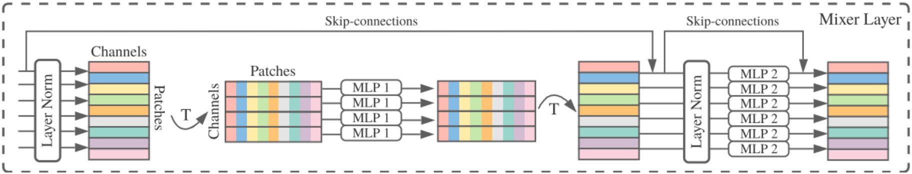

Class
Fully-connected
Global Average Pooling
N x (Mixer Layer)
Per-patch Fully-connected

MLP-Mixer overall architecture

- split image to patches 16x16 pixels
- pass each patch through **shared** FC layer to get embedding
- apply N mixer layers to embeddings

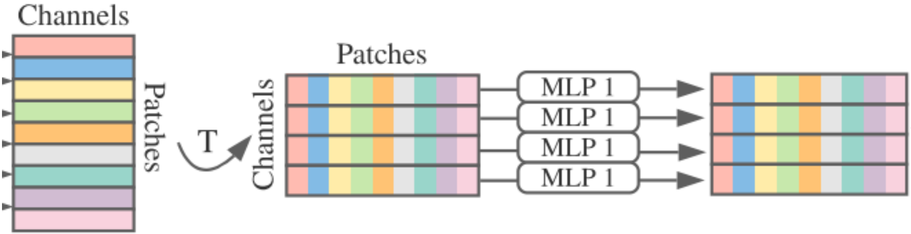
- very similar to Visual Transformers
- mixer block instead of attention between patches
- per patch FC = Conv layer with stride 16x16

Mixer Block



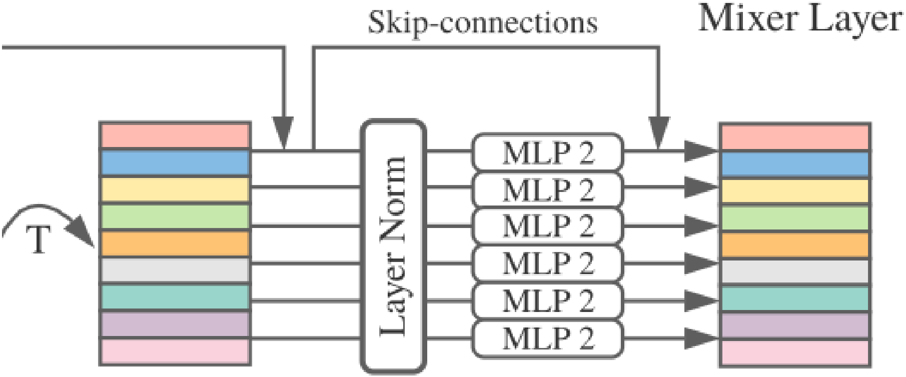
Skip-connections
Skip-connections
Mixer Layer
Channels
Layer Norm
Patches
T
Channels
Patches
MLP 1
MLP 1
MLP 1
MLP 1
T
Layer Norm
MLP 2
MLP 2
MLP 2
MLP 2
MLP 2
MLP 2

Mixer Block: Token Mixing MLP



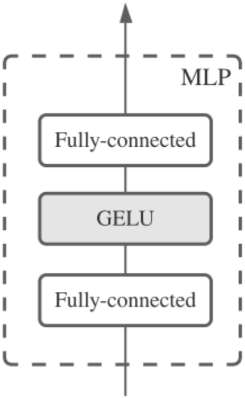
- combine information across patches = channel *i* from all patches
- same = shared MLP for each channel
- each channel is some feature detector - because of shared projection of each patch into channels

Mixer Block: Channel Mixing MLP



- combine information across channels = for each patch separately
- same = shared MLP for each patch = **1x1 Conv**

MLP Block



Mixer Block

- LayerNorm and skip connections before each **per patch** operation

- mix information between patches = instead of attention
- mix information between channels = identical to 1x1 Conv
- self-attention in ViT does both

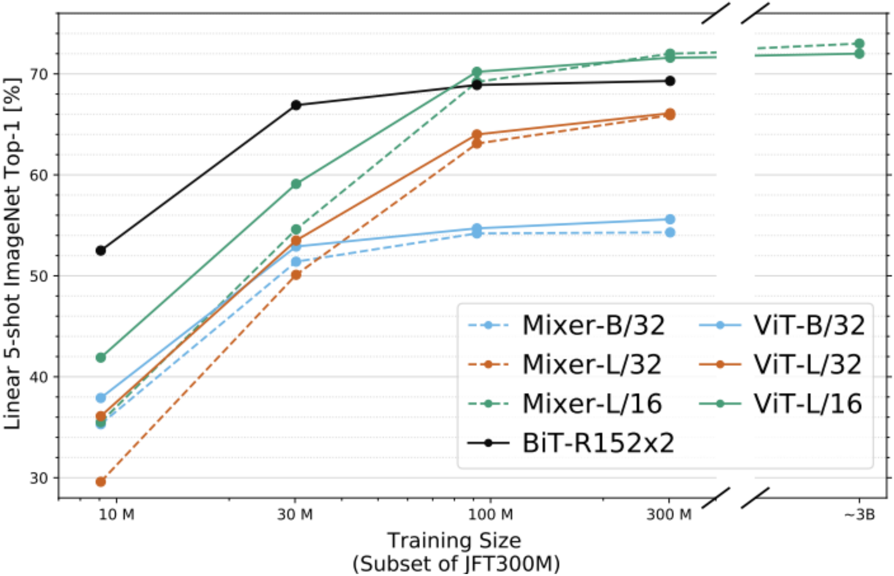
Models

Specification	S/32	S/16	B/32	B/16	L/32	L/16	H/14
Number of layers	8	8	12	12	24	24	32
Patch resolution $P \times P$	32×32	16×16	32×32	16×16	32×32	16×16	14×14
Hidden size C	512	512	768	768	1024	1024	1280
Sequence length S	49	196	49	196	49	196	256
MLP dimension D_C	2048	2048	3072	3072	4096	4096	5120
MLP dimension D_S	256	256	384	384	512	512	640
Parameters (M)	19	18	60	59	206	207	431

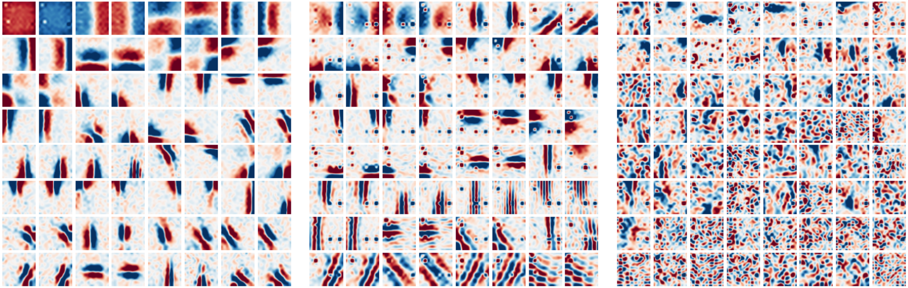
Fast inference + competitive accuracy

	ImNet top-1	ReaL top-1	Avg 5 top-1	VTAB-1k 19 tasks	Throughput img/sec/core	TPUv3 core-days
Pre-trained on ImageNet-21k (public)						
• HaloNet [51]	85.8	—	—	—	120	0.10k
• Mixer-L/16	84.15	87.86	93.91	74.95	105	0.41k
• ViT-L/16 [14]	85.30	88.62	94.39	72.72	32	0.18k
• BiT-R152x4 [22]	85.39	—	94.04	70.64	26	0.94k
Pre-trained on JFT-300M (proprietary)						
• NFNet-F4+ [7]	89.2	—	—	—	46	1.86k
• Mixer-H/14	87.94	90.18	95.71	75.33	40	1.01k
• BiT-R152x4 [22]	87.54	90.54	95.33	76.29	26	9.90k
• ViT-H/14 [14]	88.55	90.72	95.97	77.63	15	2.30k
Pre-trained on unlabelled or weakly labelled data (proprietary)						
• MPL [34]	90.0	91.12	—	—	—	20.48k
• ALIGN [21]	88.64	—	—	79.99	15	14.82k

Scales very well with more pretraining data

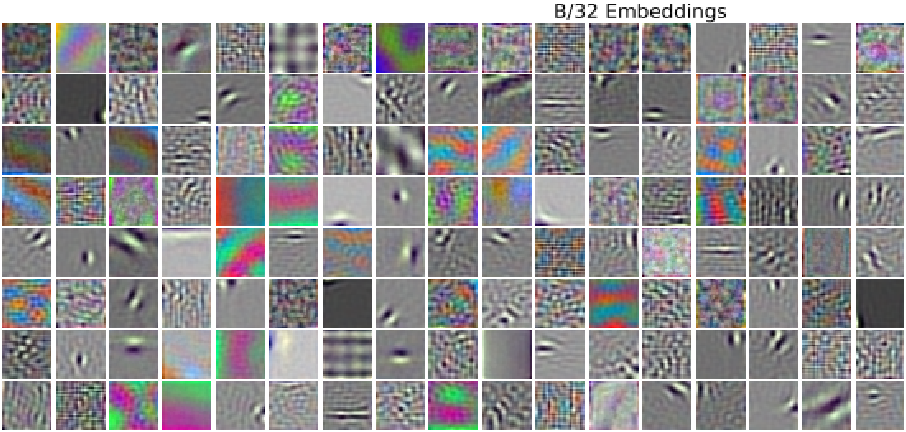


Token Mixing weights in first, second, third layer



- first layer = edge detector
- later layers = more complex detectors = like CNN

Embedding projection of the patches = patch channels



- 32x32 learns nice high level patterns
- 16x16 (not shown) learns low level noisy patterns

Conclusion

- uses patches = specific to images, learned projections like CNN
- separate token and channels mixing
- fast inference = faster than big ResNet which has better performance
- competitive accuracy = not a SoTA
- linear scaling with image size = like CNNs
- **best scaling with pretraining dataset size** = biggest advantage

My 2 cents:

- adds biases toward images = not a general arch.
- interesting for efficiency reasons but does not feel like a direction toward a breakthrough

Sources

1.Tolstikhin, Ilya, et al. "Mlp-mixer: An all-mlp architecture for vision." arXiv preprint arXiv:2105.01601 (2021).
https://arxiv.org/abs/2105.01601