MLP-Mixer: An all-MLP Architecture for Vision

I. Tolstikhim, et al. Google Research, Brain Team
Journal Club Discussion, REPU

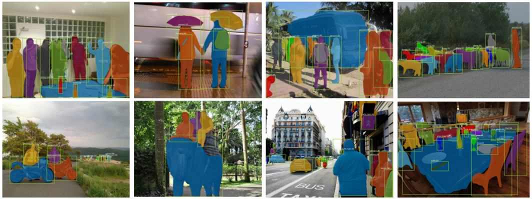
January 31, 2022

Index

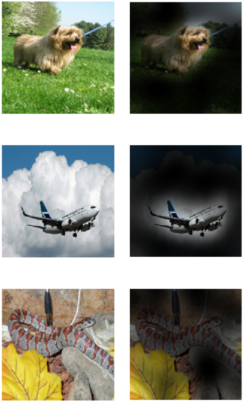
CNN and ViT are the go-to model for computer vision

- Convolution Neural Networks (CNN) are the current de-facto standard for computer vision applications (e.g. ResNet are the best solution for image classification).
- Vision Transformers (ViT) have reached a state-of-art performance and are based on self-attention layers (Google Research, Brain Team):
  - First success of Transformers in Natural Language Processing (NLP) [1].
  - Outperform by a short margin CNN's, with large enough dataset.

SOTA applications in image recognition



(a) Mask R-CNN applications for object instance segmentation [2]



(b) Attention of learned positions, ViT [3]

MLP-Mixer: a technically simple, but powerful solution

With modern hardware and larger available datasets, fully connected layer architectures such as MLP are working reasonably well [4].This architecture is conceptually and technically simple alternative:

- Relies on simple matrix multiplication, reshapes, and transpositions operations.
- It does not follow a convolution or self-attention approach.

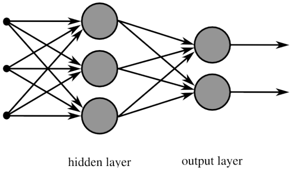


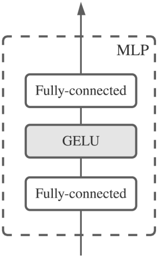
Figure 2: MLP basic concept that constitutes the MLP-Mixer [5]

Key definitions

Channel: Number of colors that have the color model (RGB = 3).
Self-Attention: Mechanism to weight neighboring instances to produce a contextual meaning.
TPU-v3: Google Cloud AI accelerator specifically for training NN.
Optimization: Seek to improve the accuracy of the model. Algorithms: SGD, Adam, Momentum, etc. Techniques: Batch-Normalization, Early Stopping, Gradient Clipping, etc.

Each Mixer's layer use 2 MLP blocks connected

Block 1: Channel/Image-mixing MLP, and executes per-location operation. It shares parameters among layers.
Block 2: Token/Patches-mixing MLP, and executes a cross-location operation among channels.



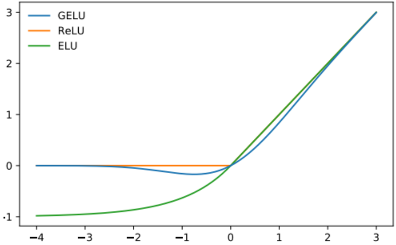


Figure 3: (Left) Structure of each MLP Block. (Right) GELU benchmarking [6].

MLP-Mixer Architecture is conceptually simple

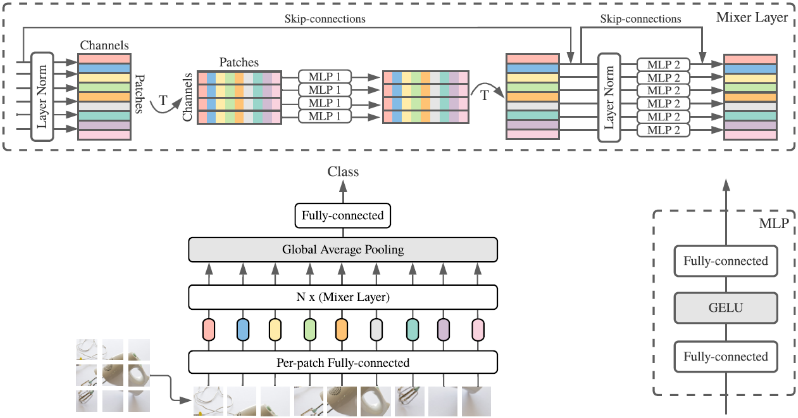


Figure 4: Mixer Python implementation available at: *https://keras.io/examples/vision*

Mixer is strongly influenced by CNNs and ViTs

Table 1: Similarities and differences between Mixer and modern vision architectures

	CNN	**ViT**
Similarities	- Parameter sharing - Skip connections by addition (ResNets) - Linear complexity according to image resolution	- Isotropic: same size input - Convert images to tokens (patches)
Differences	- Pyramidal Structure with decreasing input size	- Positional Embeddings: Mixer is already sensible to input order.

Mixer models were tested in various classification datasets

Objective of the experiment

1. Measure the **transfer accuracy** of performance for classification in popular datasets (ImageNet, CIFAR 100, etc.)
2. Obtain test-time **throughput**: data units processed in a specific time.
3. Analyze the **computation cost** of pre-training the model and then fine tuning the model.

Mixer's Experiments follow a transfer learning setup

Pre-training

Resolution of 224 pixels using Adam, linear warmup of 10k steps, batch size 4096, weight decay, and gradient clipping.

Fine-tuning

Use momentum, SGD, batch size 512, gradient clipping, cosine learning rate schedule, and increasing the input resolution.

Metrics

Evaluate the trade-off between the model's computational cost and quality using: 1) Total pre-training time on TPUv3 accelerators for each training setup and 2) Throughput in images/sec/core.

Models

Several Mixer models with different depths, patch resolutions, widths, input resolution, etc.

Several model scales and combinations were defined

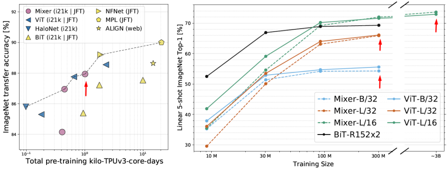
Specification	S/32	S/16	B/32	B/16	L/32	L/16	H/14
Number of layers	8	8	12	12	24	24	32
Patch resolution $P \times P$	32×32	16×16	32×32	16×16	32×32	16×16	14×14
Hidden size C	512	512	768	768	1024	1024	1280
Sequence length S	49	196	49	196	49	196	256
MLP dimension D_C	2048	2048	3072	3072	4096	4096	5120
MLP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

Figure 5: Specifications of the Mixer architectures, the "B","L" and "H" mean base, large and huge model scale. The brief notation "B/16" means the model of the base scale with patches of resolution 16x16 [5].

Mixer has competitive accuracy and good trade-off

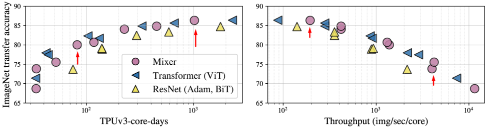
	ImNet top-1	ReaL top-1	Avg 5 top-1	VTAB-1k 19 tasks	Throughput img/sec/core	TPUv3 core-days
Pre-trained on ImageNet-21k (public)						
HaloNet [51]	85.8	—	—	—	120	0.10k
Mixer-L/16	84.15	87.86	93.91	74.95	105	0.41k
ViT-L/16 [14]	85.30	88.62	94.39	72.72	32	0.18k
BiT-R152x4 [22]	85.39	—	94.04	70.64	26	0.94k
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

(a) Big Mixer models has comparable metrics in most of the experiments



(b) Mixer performs in the Pareto frontier as well as extremely performant networks, in this case with ViT

Model scale and training influence Mixer performance

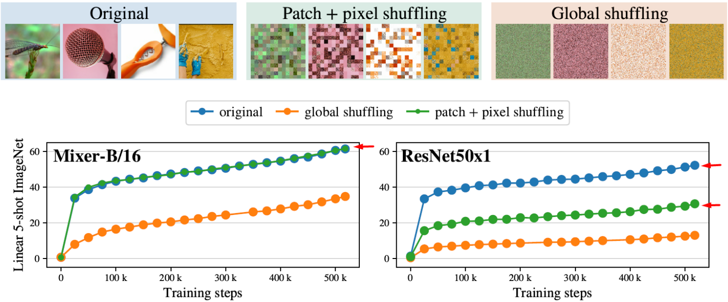


(a) Mixer is slightly below the frontier on the lower end of model scales, it sits confidently on the frontier at the high end

	Image size	Pre-Train Epochs	ImNet top-1	ReaL top-1	Avg. 5 top-1	Throughput (img/sec/core)	TPUv3 core-days
Pre-trained on ImageNet (with extra regularization)							
Mixer-B/16	224	300	76.44	82.36	88.33	1384	0.01k(‡)
ViT-B/16 (☎)	224	300	79.67	84.97	90.79	861	0.02k(‡)
Mixer-L/16	224	300	71.76	77.08	87.25	419	0.04k(‡)
ViT-L/16 (☎)	224	300	76.11	80.93	89.66	280	0.05k(‡)
Pre-trained on ImageNet-21k (with extra regularization)							
Mixer-B/16	224	300	80.64	85.80	92.50	1384	0.15k(‡)
ViT-B/16 (☎)	224	300	84.59	88.93	94.16	861	0.18k(‡)
Mixer-L/16	224	300	82.89	87.54	93.63	419	0.41k(‡)
ViT-L/16 (☎)	224	300	84.46	88.35	94.49	280	0.55k(‡)
Mixer-L/16	448	300	83.91	87.75	93.86	105	0.41k(‡)
Pre-trained on JFT-300M							
Mixer-S/32	224	5	68.70	75.83	87.13	11489	0.01k
Mixer-B/32	224	7	75.53	81.94	90.99	4208	0.05k
Mixer-S/16	224	5	73.83	80.60	89.50	3994	0.03k
BiT-R50x1	224	7	73.69	81.92	—	2159	0.08k
Mixer-B/16	224	7	80.00	85.56	92.60	1384	0.08k
Mixer-L/32	224	7	80.67	85.62	93.24	1314	0.12k
BiT-R152x1	224	7	79.12	86.12	—	932	0.14k
BiT-R50x2	224	7	78.92	86.06	—	890	0.14k
BiT-R152x2	224	14	83.34	88.90	—	356	0.58k
Mixer-L/16	224	7	84.05	88.14	94.51	419	0.23k
Mixer-L/16	224	14	84.82	88.48	94.77	419	0.45k
ViT-L/16	224	14	85.63	89.16	95.21	280	0.65k
Mixer-H/14	224	14	86.32	89.14	95.49	194	1.01k
BiT-R200x3	224	14	84.73	89.58	—	141	1.78k
Mixer-L/16	448	14	86.78	89.72	95.13	105	0.45k
ViT-H/14	224	14	86.65	89.56	95.57	87	2.30k
ViT-L/16 [14]	512	14	87.76	90.54	95.63	32	0.65k

(b) As the pre-training dataset grows, Mixer's performance steadily improves.

Model scale and training influence Mixer performance



(a) Mixer is invariant to the order of patches and pixels within the patches

They describe a very simple architecture for vision

Main Contribution: In the era of powerful processors, Mixer demonstrates that it is as good as existing SOTA methods in terms of the *trade-off* between accuracy and computational resources required for training and inference.

- ✓ The next steps are to see whether such a design works in NLP or other domains.

Mixer is the contextualization of my REPU project

Further analysis with a CKA kernel [7] for measure similarity among Mixer layer's activation might provide space for NN pruning techniques.

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\mathrm{HSIC}(L, L)}}.$$

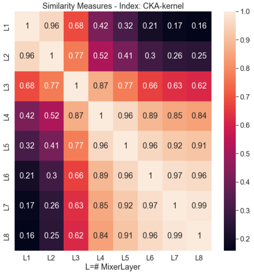


Figure 9: (Left) Definition of CKA kernel - Hilbert-Schimit Independence Criterion normalized. (Right) Some results running with a 8-layers Mixer. More updates in the Repo: *https://github.com/TheNewRobot/MixerCKA*.

References I

[1] A. Vaswani *et al.*, "Attention is all you need," 2017.

[2] K. He *et al.*, "Mask r-cnn," 2018.

[3] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[4] I. Goodfellow *et al.*, *Deep Learning*.
MIT Press, 2016.
http://www.deeplearningbook.org.

[5] I. Tolstikhin *et al.*, "Mlp-mixer: An all-mlp architecture for vision," 2021.

References II

[6] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2020.

[7] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," 2019.