Hopfield/Mixer correspondence

towards a better understanding of MetaFormers architecture design

Toshihiro Ota

CyberAgent, Inc. / RIKEN iTHEMS

December, 2023

Mainly based on 2304.13061 with Masato Taki (Rikkyo Univ./RIKEN iTHEMS)

Biography

Apr. 2016 - Mar. 2021	Osaka Univ., Ph.D. in Physics
	AdS/CFT, class $\mathcal{S}$, integrability
Apr. 2021 - July 2021	UTokyo, Math. Sci.
	Low-dim. topology, quantum algebra
Aug. 2021 - Nov. 2022	TokyoTech, School of Computing
	Machine Learning, Deep Learning
Dec. 2022 - present	CyberAgent, AI Lab
	Machine Learning, Deep Learning
Apr. 2019 - present	RIKEN, iTHEMS

Today's main message:

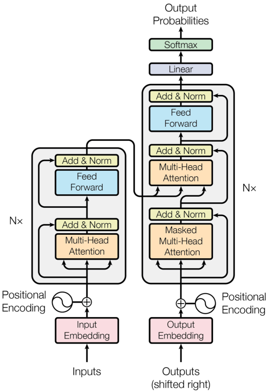
- Hopfield/Mixer correspondence as an approach for MetaFormes architecture design

Based on the correspondence, we theoretically predict

iMixer: a novel MetaFormer model from hierarchical Hopfield network [TO-Taki, 2304.13061]

Attention is All You Need

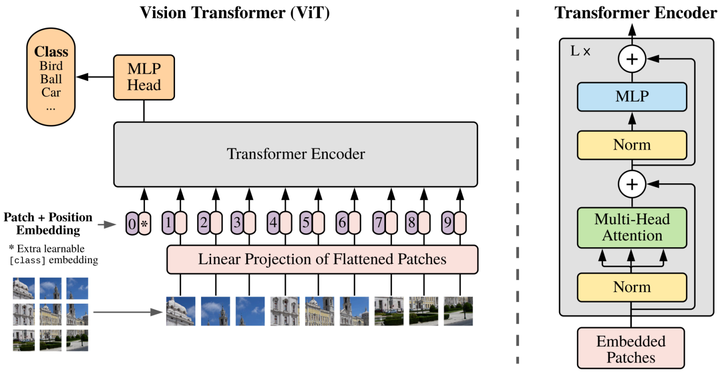
Transformer in our everyday life [Vaswani+ NeurIPS17, Fig. 1]



Large success across nearly all domains

Attention is All You Need

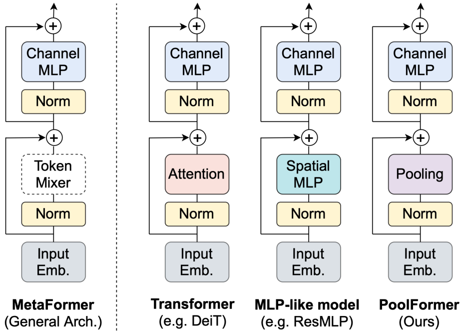
An image is worth 16x16 words: Vision Transformer



[Dosovitskiy+ ICLR21; Touvron+ ICML21; . . .]

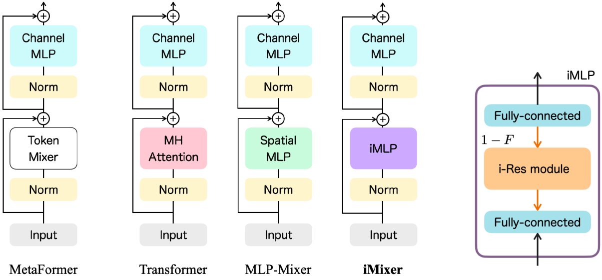
ATTENTION IS ALL YOU NEED?

MetaFormers (MLP-Mixer, Conv/Pool/Rand/Identity-Former, ...) [Tolstikhin+ NeurIPS21; Melas-Kyriazi 21; Yu+ 22]



[Yu+ CVPR22, Fig. 1a]

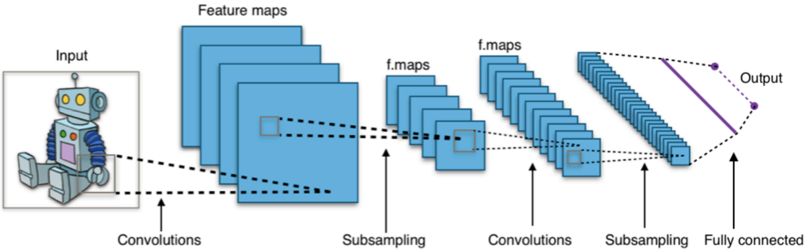
iMixer: invertible, implicit and iterative MLP-Mixer



- *Derive* a new MetaFormer model from Hopfield/Mixer correspondence
- Provide a direction for incorporating *implicit* NNs
- Empirical study supports the validity of our formulation

Contents

Convolutional Neural Network



https://en.wikipedia.org/wiki/Convolutional_neural_network

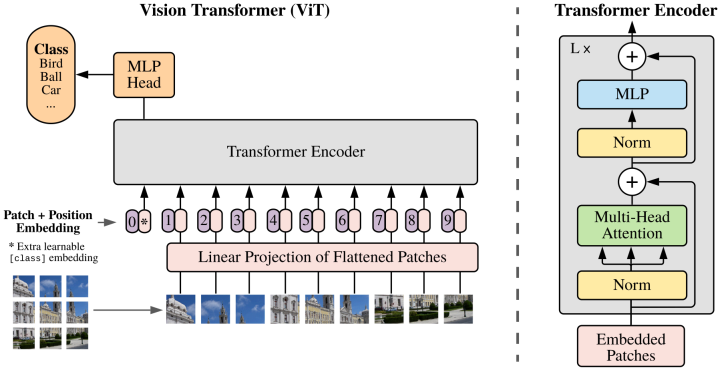
respects

- Locality
- Translation invariance

$\rightsquigarrow$ "inductive bias"

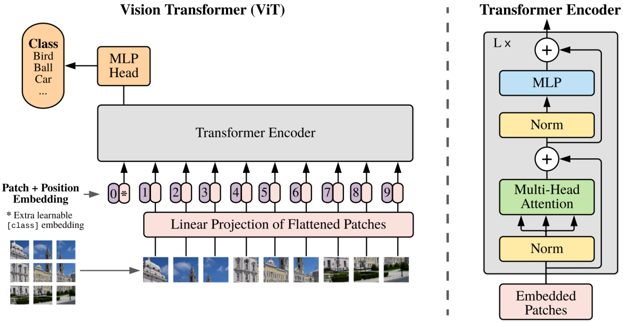
Vision Transformer

An image is worth 16x16 words [Dosovitskiy+ ICLR21, Fig. 1]



Quite less inductive bias than CNN

Vision Transformer



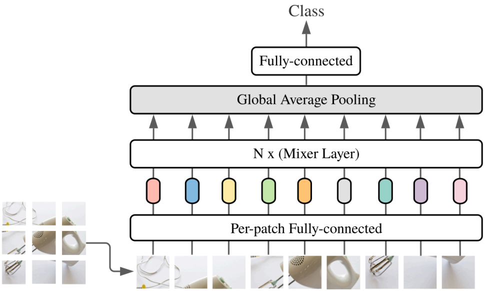
Attention mechanism:

$$Y = \mathrm{Attn}(X) = V^{\top}\mathrm{softmax}\left(KQ^{\top}\right)$$

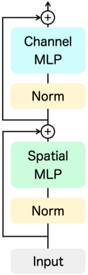
$$Q = W_Q X, \quad K = W_K X, \quad V = W_V X$$

Attention is All You Need?

Casted doubt on the role of attention module: MLP-Mixer



Mixer Layer



[Tolstikhin+ NeurIPS21, Fig. 1]

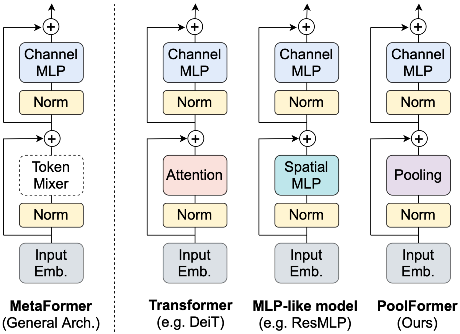
Spatial MLP:

$$Y = W_2 \sigma(W_1 X)$$

Simpler than attention mechanism and yet less inductive bias

Attention is All You Need?

MetaFormers [Yu+ CVPR22, Fig. 1a]



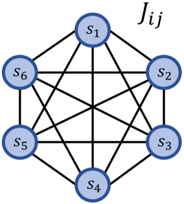
Token-mixing block:

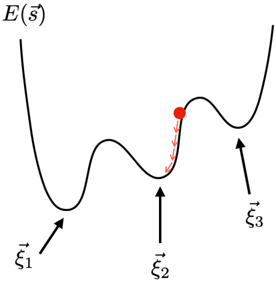
$$Y = X + \mathrm{TokenMixer}(\mathrm{Norm}(X))$$

Contents

Classical Hopfield network

A classical associative memory model [Hopfield 82]



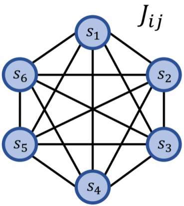


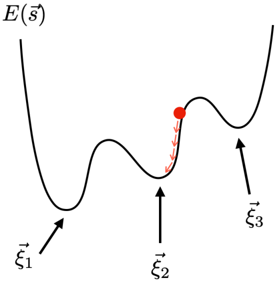
Update rule $\vec{s} \leftarrow \mathrm{sgn}(J\vec{s})$ minimizes the energy function,

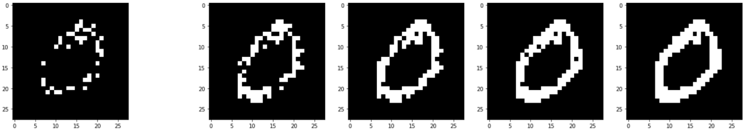
$$E(\vec{s}) = -\sum_{i \neq j} J_{ij} s_i s_j, \quad J := \sum_{\mu} \vec{\xi}_\mu \vec{\xi}_\mu^{\top}, \quad s_i \in \{\pm 1\}$$

Classical Hopfield network

A classical associative memory model [Hopfield 82]





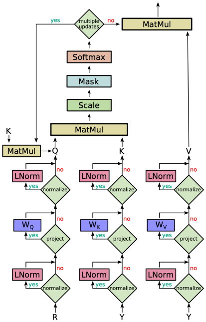


Hopfield Networks is All You Need

Attention = a Hopfield update rule [Ramsauer+ ICLR21, Fig. A.7]

$$v_i \in \mathbb{R},$$

$$\xi = (\vec{\xi}_1, \ldots, \vec{\xi}_N)^\top$$



Update rule

$$v_i \leftarrow \sum_{\mu} \xi_{i\mu} \mathrm{softmax}\left(\sum_j \xi_{\mu j} v_j \right)$$

minimizes an energy function

$$E(\{v_i\}) = \frac{1}{2} \sum_i v_i^2 - \log \sum_{\mu} \exp\left(\sum_i \xi_{\mu i} v_i \right)$$

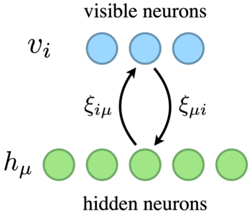
Generalized Hopfield network

Unification of energy-based associative memory models

[Krotov-Hopfield ICLR21]

$$\tau_v \frac{dv_i(t)}{dt} = \sum_{\mu=1}^{N_h} \xi_{i\mu} f_\mu(h(t)) - v_i(t)$$

$$\tau_h \frac{dh_\mu(t)}{dt} = \sum_{i=1}^{N_v} \xi_{\mu i} g_i(v(t)) - h_\mu(t)$$



Activation functions f, g are determined by "Lagrangians":

$$f_\mu(h) = \frac{\partial L_h(h)}{\partial h_\mu}, \quad g_i(v) = \frac{\partial L_v(v)}{\partial v_i}$$

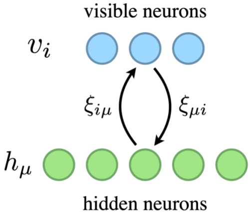
Generalized Hopfield network

Unification of energy-based associative memory models

[Krotov-Hopfield ICLR21]

$$\tau_v \frac{dv_i(t)}{dt} = \sum_{\mu=1}^{N_h} \xi_{i\mu} f_\mu(h(t)) - v_i(t)$$

$$\tau_h \frac{dh_\mu(t)}{dt} = \sum_{i=1}^{N_v} \xi_{\mu i} g_i(v(t)) - h_\mu(t)$$



Activation functions f, g are determined by “Lagrangians”:

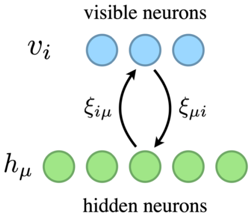
$$f_\mu(h) = \frac{\partial L_h(h)}{\partial h_\mu}, \quad g_i(v) = \frac{\partial L_v(v)}{\partial v_i}$$

Generalized Hopfield network

The dynamical equations (update rules for the neurons)

$$\tau_v \frac{dv_i(t)}{dt} = \sum_{\mu=1}^{N_h} \xi_{i\mu} f_\mu(h(t)) - v_i(t)$$

$$\tau_h \frac{dh_\mu(t)}{dt} = \sum_{i=1}^{N_v} \xi_{\mu i} g_i(v(t)) - h_\mu(t)$$



minimize the energy function

$$E(v,h) = \sum_i v_i g_i - L_v + \sum_\mu h_\mu f_\mu - L_h - \sum_{\mu,i} f_\mu \xi_{\mu i} g_i$$

Lagrangians L_v, L_h define a model

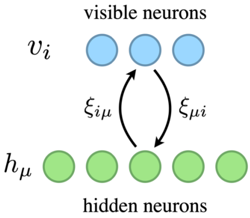
Generate a family of Hopfield networks

Generalized Hopfield network

The dynamical equations (update rules for the neurons)

$$\tau_v \frac{dv_i(t)}{dt} = \sum_{\mu=1}^{N_h} \xi_{i\mu} f_\mu(h(t)) - v_i(t)$$

$$\tau_h \frac{dh_\mu(t)}{dt} = \sum_{i=1}^{N_v} \xi_{\mu i} g_i(v(t)) - h_\mu(t)$$



minimize the energy function

$$E(v,h) = \sum_i v_i g_i - L_v + \sum_\mu h_\mu f_\mu - L_h - \sum_{\mu,i} f_\mu \xi_{\mu i} g_i$$

Lagrangians L_v, L_h define a model

Generate a family of Hopfield networks

Attention as a modern Hopfield network

"Model B" in [Krotov-Hopfield ICLR21]

$$L_v(v) = \frac{1}{2}\sum_i v_i^2, \quad L_h(h) = \log \sum_\mu \exp(h_\mu)$$

Integrate out hidden neurons h_μ, discretize the ODE, then

$$v_i(t+1) = \sum_\mu \xi_{i\mu} \mathrm{softmax}\left(\sum_j \xi_{\mu j} v_j(t)\right)$$

$$E(\{v_i\}) = \frac{1}{2}\sum_i v_i^2 - \log \sum_\mu \exp\left(\sum_i \xi_{\mu i} v_i\right)$$

reproduce [Ramsauer+ ICLR21]

Attention as a modern Hopfield network

Applications along this line:

- Immune repertoire classification [Widrich+ NeurIPS20]
- Exponential capacity of dense associative memories [Lucibello-Mezard 23]
- Learning with partial forgetting in modern Hopfield networks [TO-Sato-Kawakami-Tanaka-Inoue AISTATS23]
- A family of Boltzmann machines from modern Hopfield networks [TO-Karakida NECO23]
  - Attentional Boltzmann machine is an exactly solvable model

Contents

Hopfield/Mixer correspondence

MLP-Mixer as Model C of the generalized Hopfield network

[Krotov-Hopfield ICLR21; Tang-Kopp 21]

$$L_v(v) = \sqrt{\sum_i (v_i - \bar{v})^2}, \quad L_h(h) = \sum_\mu \phi(h_\mu)$$

Integrate out hidden neurons h_μ, discretize the ODE, then

$$v_i(t+1) = v_i(t) + \sum_\mu \xi_{i\mu} \phi' \left(\sum_j \xi_{\mu j} \mathrm{LayerNorm}(v(t))_j \right)$$

Token-mixing block of MLP-Mixer [Tolstikhin+ NeurIPS21]

$$Y = X + W_2 \sigma(W_1 \mathrm{LayerNorm}(X))$$

HOPFIELD/MIXER CORRESPONDENCE

MLP-Mixer as Model C of the generalized Hopfield network

[Krotov-Hopfield ICLR21; Tang-Kopp 21]

$$L_v(v) = \sqrt{\sum_i (v_i - \bar{v})^2}, \quad L_h(h) = \sum_\mu \phi(h_\mu)$$

Integrate out hidden neurons h_μ, discretize the ODE, then

$$v_i(t+1) = v_i(t) + \sum_\mu \xi_{i\mu} \phi' \left(\sum_j \xi_{\mu j} \mathrm{LayerNorm}(v(t))_j \right)$$

Token-mixing block of MLP-Mixer [Tolstikhin+ NeurIPS21]

$$Y = X + W_2 \sigma(W_1 \mathrm{LayerNorm}(X))$$

The generalized Hopfield network can *reproduce* many of known NN models. So far so good

A natural question:

> The generalized Hopfield network can even *predict* a novel MetaFormer architecture?

Model-C Hopfield network	$\rightsquigarrow$	MLP-Mixer
Model-C hierarchical extension	$\rightsquigarrow$	???

The generalized Hopfield network can *reproduce* many of known NN models. So far so good

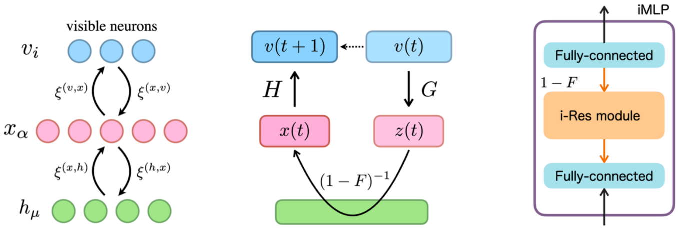
A natural question:

> The generalized Hopfield network can even *predict* a novel MetaFormer architecture?

Model-C Hopfield network	$\rightsquigarrow$	MLP-Mixer
Model-C hierarchical extension	$\rightsquigarrow$	???

iMixer

Hierarchical extension



$$L_v(v) = \sqrt{\sum_i (v_i - \bar{v})^2}, \quad L_x(x) = \sum_\alpha \phi_x(x_\alpha), \quad L_h(h) = \sum_\mu \phi_h(h_\mu)$$

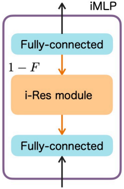
$$v(t+1) = v(t) + \xi^{(v,x)} \phi'_x \Big((1-F)^{-1} \big(\xi^{(x,v)} \mathrm{LayerNorm}(v(t)) \big) \Big)$$

$$F = (\xi^{(x,h)} \phi'_h) \circ (\xi^{(h,x)} \phi'_x)$$

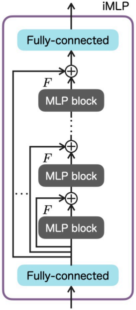
iMixer

Inverted ResNet is an example of implicit NNs

[Behrmann+ ICML19; Bai+ NeurIPS19; El Ghaoui+ 19]



$\rightsquigarrow$



Algorithm 1 Feedforward computation of the iMLP module.

Input: input x, fully-connected layer G, contractive MLP block F, fully-connected layer H, number of fixed-point iterations n
Init: $x^0 := G(x)$
for $a = 0, \ldots, n-1$ **do**
  $x^{a+1} := x^0 + F(x^a)$
end for
return: $H(x^n)$

Fixed-point iteration method enables us to easily implement & train the model

iMixer

The iMLP module looks somewhat unconventional from CV viewpoint. Experimental evaluation?

Top-1 accuracy (%), trained on CIFAR-10 from scratch

Model	Small	Base	Large
Mixer (baseline)	88.08 $\pm$0.51	89.03 $\pm$0.24	86.67 $\pm$0.30
iMixer (ours)	**88.56** $\pm$0.30	**89.07** $\pm$0.33	**87.48** $\pm$0.40

Top-1 accuracy (%) for other datasets, trained from scratch for Small models

Model	CIFAR-100	Food-101	ImageNet-1k
Mixer-S	68.13 $\pm$0.46	76.11 $\pm$0.32	73.91
iMixer-S	68.26 $\pm$0.30	76.08 $\pm$0.20	74.10

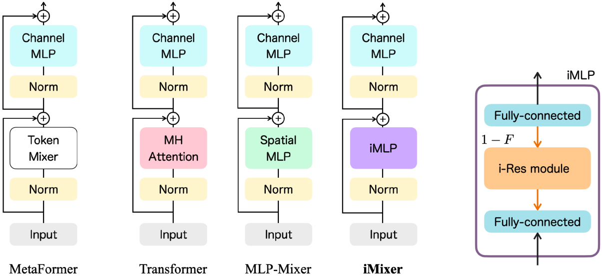
Outlook

Lots of further directions like

- More hidden layers and different Lagrangians
- Practical applications for real computer vision tasks
- Boltzmann machine counterparts of hierarchical Hopfield networks
- More direct relation with associative memory model (in progress with Taki and Karakida)

Any discussions/comments are very welcome

iMixer: invertible, implicit and iterative MLP-Mixer



- *Derive* a new MetaFormer model from Hopfield/Mixer correspondence
- Provide a direction for incorporating *implicit* NNs
- Empirical study supports the validity of our formulation

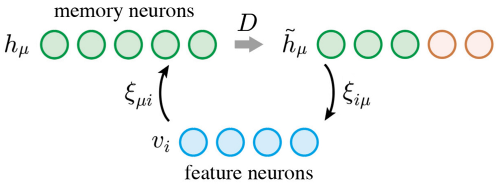
Contents

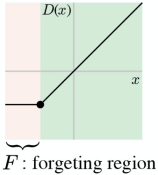
Backup

LwPF

Learning with partial forgetting in modern Hopfield networks

[TO-Sato-Kawakami-Tanaka-Inoue AISTATS23]



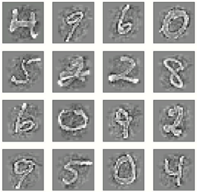


- Propose *learning with partial forgetting* (LwPF) mechanism
- Derive the expression for *partially forgetting attention*
- Demonstrate the effectiveness of LwPF in diverse domains

AttnBM

Attention in a family of Boltzmann machines emerging from modern Hopfield networks [TO-Karakida NECO23]

Image reconstruction



Receptive fields

- Propose a family of Boltzmann machines from the generalized Hopfield network
- Investigate the basic properties of *attentional BM* and verify its integrability and trainability

Generalized Hopfield network

Model A: Dense associative memory models [Hopfield 82; Krotov-Hopfield NeurIPS16; Demircigil +17]

$$L_v(v) = \sum_i |v_i|, \quad L_h(h) = \sum_\mu F(h_\mu)$$

Integrate out hidden neurons h_μ, discretize the ODE, then

$$v_i(t+1) = \sum_\mu \xi_{i\mu} F'\left(\sum_j \xi_{\mu j} \operatorname{sgn}\big(v_j(t)\big) \right)$$

$$E(\{v_i\}) = -\sum_\mu F\left(\sum_i \xi_{\mu i} \operatorname{sgn}(v_i) \right)$$

Generalized Hopfield network

Model A: Dense associative memory models [Hopfield 82; Krotov-Hopfield NeurIPS16; Demircigil+ 17]

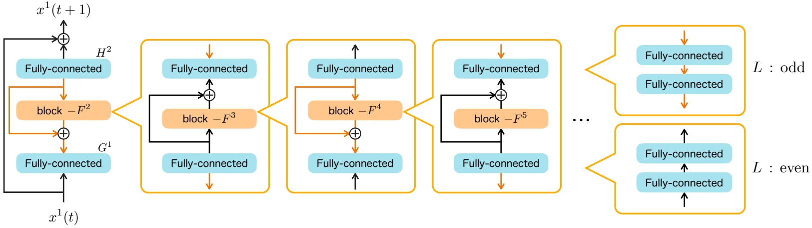
$$v_i(t+1) = \sum_\mu \xi_{i\mu} F'\left(\sum_j \xi_{\mu j}\,\mathrm{sgn}\big(v_j(t)\big)\right)$$

$$E(\{v_i\}) = -\sum_\mu F\left(\sum_i \xi_{\mu i}\,\mathrm{sgn}(v_i)\right)$$

- $F(x) = x^2$: the classical Hopfield network, $\mathrm{sgn}(v_i(t)) =: s_i(t)$
- $F(x) = x^n$: the network can store $\mathcal{O}(N_v^{n-1})$ memories
- $F(x) = e^x$: exponential storage capacity

iMixer: a general formulation

One of the most general formulations of iMixer from L-layer hierarchical Hopfield network:



$$x^1(t+1) = x^1(t) + \text{iMLPs}(x^1(t))$$

iMixer: experimental details

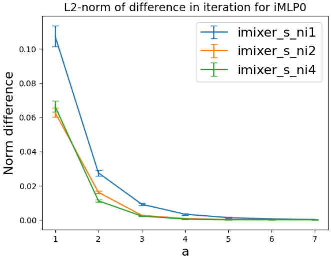
Hyperparameters commonly used for the vanilla Mixer and iMixer for fair comparison.

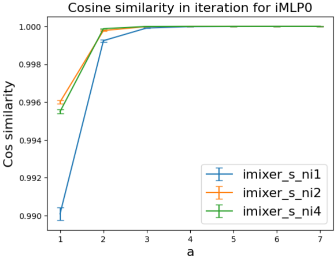
Training configuration	Small/Base/Large
optimizer	AdamW
training epochs	300
batch size	512/256/64
base learning rate	5e-4/2.5e-4/6.25e-5
weight decay	0.05
optimizer ϵ	1e-8
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
learning rate schedule	cosine decay
lower learning rate bound	1e-6
warmup epochs	20
warmup schedule	linear
warmup learning rate	1e-6
cooldown epochs	10
crop ratio	0.875
RandAugment	(9, 0.5)
mixup α	0.8
cutmix α	1.0
random erasing	0.25
label smoothing	0.1
stochastic depth	0.1/0.2/0.3

iMixer: experimental details

Hyperparameter search for h_r and n in iMixer-S, trained on CIFAR-10 from scratch

h_r	$n=1$	$n=2$	$n=4$
0.25	88.26 $\pm$0.28	88.22 $\pm$0.33	88.29 $\pm$0.37
0.5	88.32 $\pm$0.39	88.21 $\pm$0.45	88.22 $\pm$0.43
1	88.36 $\pm$0.31	88.32 $\pm$0.32	88.32 $\pm$0.32
2	**88.54** $\pm$0.34	**88.56** $\pm$0.30	**88.46** $\pm$0.26





Convergence rate of L_2-norm (left) and cosine similarity (right)
between two successive feature vectors in fixed-point iteration in iMLP-0