Regularization of polynomial networks for image recognition

Grigorios G Chrysos, Bohan Wang, Jiankang Deng, Volkan Cevher

École Polytechnique Fédérale de Lausanne (EPFL)

Poster #359 (Wednesday afternoon)

Background: Image generation with Polynomial Neural Networks

G Chrysos, S Moschoglou, G Bouritsas, Y Panagakis, J Deng, and S Zafeiriou. 'Π-nets: Deep Polynomial Neural Networks.' In CVPR, 2020.

Background: Large-scale recognition with Polynomial Neural Networks

Task	Dataset	Model	Metric Name	Metric Value	Global Rank
Face Recognition	AgeDB-30	Prodpoly	Accuracy	0.98467	# 2
Face Recognition	CALFW	Prodpoly	Accuracy	0.96233	# 1
Face Recognition	CFP-FF	Prodpoly	Accuracy	99.886	# 1
Face Recognition	CFP-FP	Prodpoly	Accuracy	0.98986	# 1

https://paperswithcode.com/paper/deep-polynomial-neural-networks (Jan. 2023)

Rank	Model	Accuracy	Paper	Code	Result	Year
1	**Prodpoly**	98.95%	Deep Polynomial Neural Networks			2020
2	**ElasticFace-Arc**	98.81%	ElasticFace: Elastic Margin Loss for Deep Face Recognition			2021
3	**ArcFace + MS1MV2 + R100 + R**	98.48%	ArcFace: Additive Angular Margin Loss for Deep Face Recognition			2018
4	**DiscFace**	97.44%	DiscFace: Minimum Discrepancy Learning for Deep Face Recognition			2020
5	**Dynamic AdaCos**	97.41%	AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations			2019

https://paperswithcode.com/sota/face-verification-on-megaface

G Chrysos, S Moschoglou, G Bouritsas, J Deng, Y Panagakis, and S Zafeiriou. 'Deep Polynomial Neural Networks.' In T-PAMI, 2021.

Background: Extrapolation with Polynomial Neural Networks

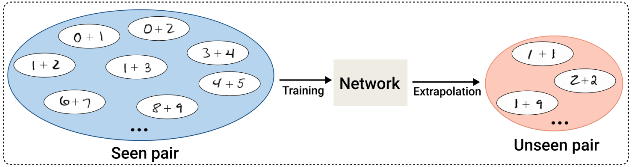

Table: Experimental evaluation of visual addition.

Method	Accuracy (Rounding)
NN(Dense)	0.436 ± 0.065
PNN (Dense)	$\mathbf{0.554 \pm 0.011}$
NN(Conv)	0.617 ± 0.103
PNN (Conv)	$\mathbf{0.825 \pm 0.109}$

Y Wu, Z Zhu, F Liu, G Chrysos, V Cevher, 'Extrapolation and Spectral Bias of Neural Nets with Hadamard Product: a Polynomial Net Study'. In NeurIPS, 2022.

Motivation

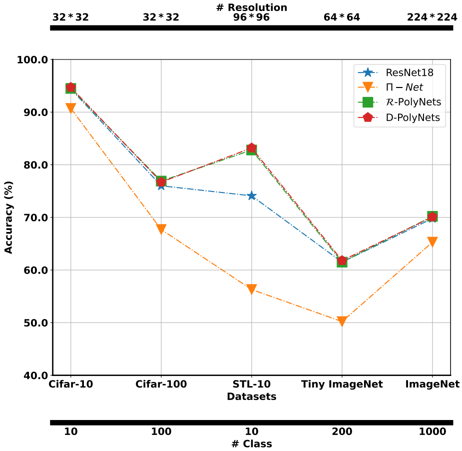

Figure: The proposed networks ($\mathcal{R}$-PolyNets, $\mathcal{D}$-PolyNets) reach the performance of powerful neural networks.

The proposed $\mathcal{R}$-PolyNets

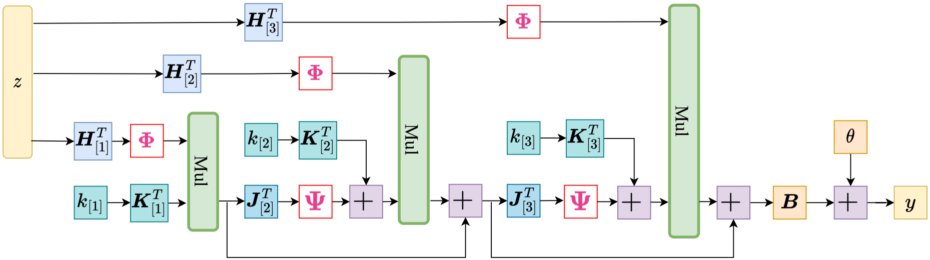

Figure: Fourth-degree expansion with respect to the input z. 'Mul' abbreviates the elementwise product.

$\mathcal{R}$-PolyNets for N degree polynomial expansions

$$\boldsymbol{x}_n = \left(\boldsymbol{\Phi}\boldsymbol{H}_{[n]}^T \boldsymbol{z}\right) * \left(\boldsymbol{\Psi}\boldsymbol{J}_{[n]}^T \boldsymbol{x}_{n-1} + \boldsymbol{K}_{[n]}\boldsymbol{k}_{[n]}\right) + \boldsymbol{x}_{n-1}, \quad (1)$$

- Linear mapping (of the input $\boldsymbol{z}$)
- Elementwise product
- Linear mapping (of the intermediate term $\boldsymbol{x}_{n-1}$)
- Skip connection

for $n = 2, \ldots, N$ with $\boldsymbol{x}_1 = \boldsymbol{z}$ and $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x}_N + \boldsymbol{\theta}$. The parameters $\left\{\boldsymbol{H}_{[n]}, \boldsymbol{J}_{[n]}, \boldsymbol{K}_{[n]}, \boldsymbol{k}_{[n]}\right\}_{n=2}^{N}$ and $\boldsymbol{B}, \boldsymbol{\theta}$ are learnable.

Model details

Critical components utilized in $\mathcal{R}$-PolyNets:

- Initialization scheme.
- Normalization scheme.
- Additional data augmentations and regularization.

The proposed $\mathcal{D}$-PolyNets

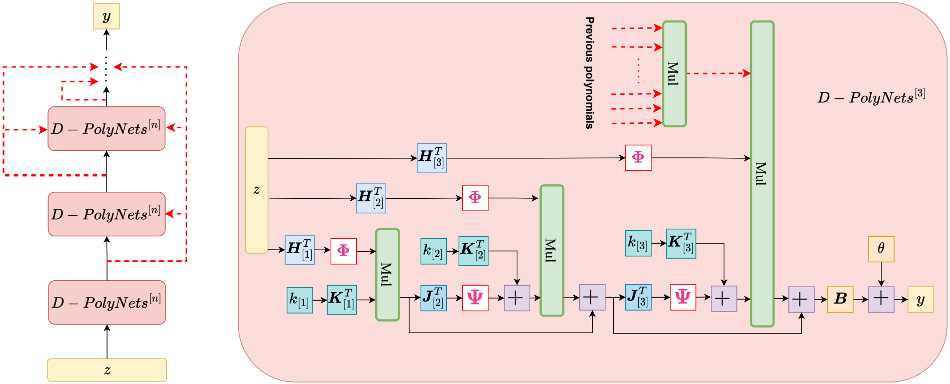

Figure: (Left) The complete architecture of $\mathcal{D}$-PolyNets. (Right) A single block.

Experimental validation

Method	Cifar-10	Cifar-100	STL-10	Tiny ImageNet	Oxford Flowers
ResNet18	0.944	0.760	0.741	0.615	0.877
Hybrid Π-Nets	0.944	0.765	0.775	0.611	0.889
PDC	0.909	0.689	0.681	0.452	0.885
Π-Nets	0.907	0.677	0.563	0.502	0.826
$\mathcal{R}$-PolyNets (ours)	0.945	**0.769**	0.828	0.615	**0.949**
$\mathcal{D}$-PolyNets (ours)	0.947	0.767	0.834	**0.618**	0.941
$\mathcal{R}$-PDC (ours)	0.947	0.757	0.833	0.560	0.947
$\mathcal{D}$-PDC (ours)	**0.949**	0.762	**0.855**	0.569	0.945

Experimental validation on ImageNet

Model	# par (M)	Top-1 Acc. (%)	Top-5 Acc. (%)
ResNet18	11.69	69.758	89.078
ResNet18 without activations	11.69	20.536	39.986
Hybrid Π-Nets	11.96	**70.740**	**89.548**
Π-Nets	12.38	65.280	85.958
$\mathcal{R}$-PolyNets (ours)	12.38	70.228	89.390
$\mathcal{D}$-PolyNets (ours)	11.36	70.090	89.424

Conclusion and future work

- We introduce $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets, which represent polynomial networks without elementwise activation functions.
- We exhibit how the proposed networks can be trained to obtain performance comparable to powerful neural networks.

Conclusion and future work

- We introduce $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets, which represent polynomial networks without elementwise activation functions.
- We exhibit how the proposed networks can be trained to obtain performance comparable to powerful neural networks.
- As a future step, we believe a more thorough understanding of the trainability of the models is essential.
- A deeper understanding of the inductive bias will enable the design of improved polynomial networks.
- Scaling polynomial networks to even higher degrees to outperform the largest transformers is still an open task.

Thank you for your attention

Come to our poster #359 (Wednesday afternoon).

1. Code: https://github.com/grigorisg9gr/regularized_polynomials
2. Contact us: grigorios.chrysos [at] epfl.ch.