Compact and Optimal Deep Learning with Recurrent Parameter Generators

Berkeley UNIVERSITY OF CALIFORNIA · MICHIGAN · Meta · NYU · Massachusetts Institute of Technology

Jiayun Wang* Yubei Chen* Stella X. Yu Brian Cheung Yann LeCunn

* Indicates equal contribution

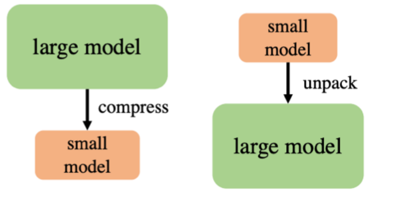
WACV WAIKOLOA HAWAII JAN 3-7 • 2023

Project website

Motivations and Overview

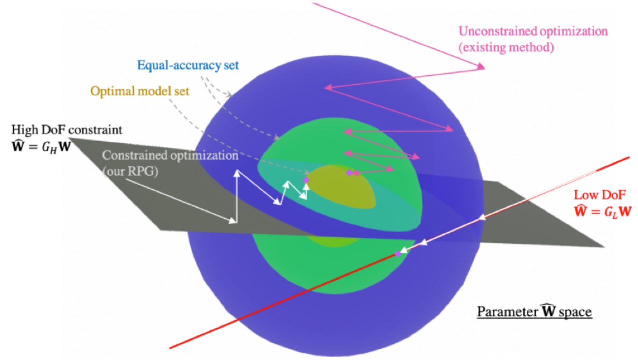
A novel approach to compact and optimal deep learning by decoupling model degree of freedom (DoF) and model parameters.

Comparison to Current Methods

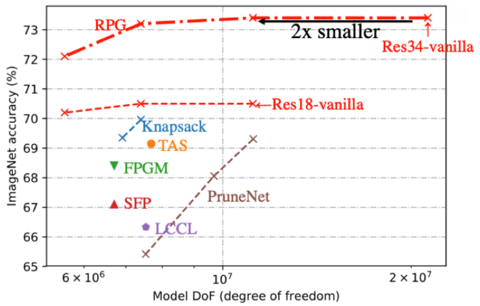


a) Existing method **b)** Our method (RPG)

a) Existing methods first finds the optimal in a large model space and then compress it.
b) We start with a small (DoF) model of free parameters, use recurrent parameter generator (RPG) to unpack them onto a large model with predefined random linear projections.

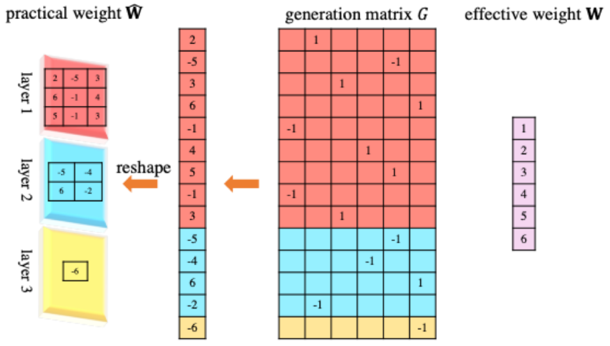


Linearly constrained neural optimization: Gradient descent finds the optimal model of a small DoF under our linear constraints with faster converge than training the large unpacked model. If the DoF is too small, the optimal large model may fall out of the constrained subpsace. However, at a sufficiently large DoF, RPG gets rid of redundancy and often finds a model with little loss in accuracy.



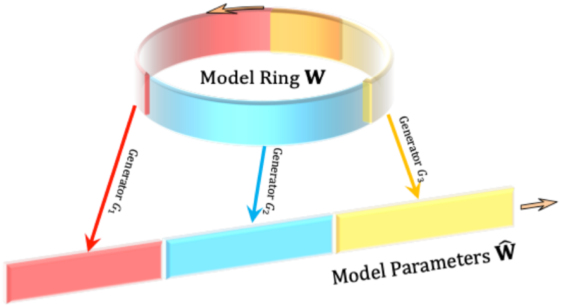
Results: RPG achieves the same ImageNet accuracy with half of the ResNet-vanilla DoF. RPG also outperforms other state-of-the-art compression approaches.

Recurrent Parameter Generator (RPG)



$$\hat{W} = GW$$

Linearly Constrained Neural Optimization (general case)
Networks are optimized with a linear constraint $\hat{W} = GW$, where the constrained parameter $\hat{W}$ of each network layer was generated by the generating matrix G from the free parameter W, which is directly optimized. $\hat{W}$ is unpacked large model parameter while the size of $\hat{W}$ is the model DoF.

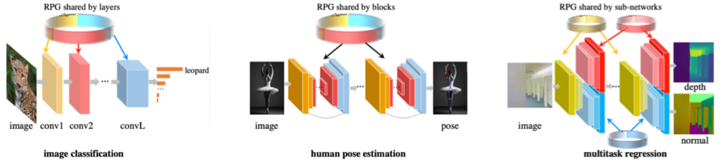


Recurrent Parameter Generator (RPG, special case)
RPG shares a fixed set of parameters in a ring and uses them to generate parameters of different parts of a neural network, whereas in the standard neural network, all the parameters are independent of each other, so the model gets bigger as it gets deeper. The third section of the model starts to overlap with the first section in the model ring, and all later layers share generating parameters for possibly multiple times.

Destructive weight sharing: $G_i \in \{b \circ p | b \in B(N_i), p \in P(N_i)\}$

Random sign flip permutation

RPG Performs Better at the Same DoF



We apply RPG to tasks including image classification, human pose estimation and multitask regression. RPGs are shared at multiple scales: a network can either have a global RPG or multiple local RPGs that are shared within blocks or sub-networks.

CIFAR100 and ImageNet Classification

Comparisons to baselines

	DoF	Acc. (%)
R18-vanilla	11M	77.5
R34-RPG.blk	11M	78.5
R34-RPG	11M	**78.9**
R34-random weight share	11M	74.9
R34-DeepCompression [23]	11M	72.2
R34-Hash [42]	11M	75.6
R34-Lego [67]	11M	78.4
R34-vanilla	21M	**79.1**

Model DoF v.s. accurary

Acc. (%)	R18-RPG			R18-vanilla
ImageNet	40.0	67.2	70.5	70.5
CIFAR100	60.2	75.6	77.6	77.6
Model DoF	45K	2M	5.5M	11M
Acc. (%)	**R34-RPG**			**R34-vanilla**
ImageNet	41.6	69.1	**73.4**	**73.4**
CIFAR100	61.7	76.5	**78.9**	**79.1**
Model DoF	45K	2M	11M	21M

• ResNet-RPG outperforms existing DoF reduction methods on CIFAR100. Also, a global RPG outperforms block-wise local RPGs.
• ResNet-RPG consistently achieves higher performance at the same model DoF.

Pose estimation

Acc. (DoF)	CPM [62]	RPG	No shared w.
1x sub-net	84.7 (3.3M)		
2x sub-nets	86.1 (3.3M)	86.5 (3.3M)	87.1 (6.7M)
4x sub-nets	86.5 (3.3M)	87.3 (3.3M)	88.0 (13.3M)

Multi-Task Regression

RMSE (%)	Depth	Normal
Vanilla model	25.5	41.0
RPG with shared BN	24.7	40.3
Reuse & new BN	24.0	39.4
Reuse & new BN & perm. and reflect.	**22.8**	**39.1**

• RPG outperforms model at the same DoF for both pose estimation and multi-task regression on the Stanford 3D indoor scenes dataset.

RPG Increases the Model Generalizability

ImageNet train-val gap

Acc gap (%)	vanilla	RPG
R18	-0.7	**-2.7**
R34	1.1	**-2.3**

pose estimation train-val gap

Acc gap (%)	no shared w	shared w	RPG
2x sub-nets	1.15	1.13	**0.64**
4x sub-nets	1.98	1.70	**1.15**

direct evaluation on ObjectNet

	R18	R34-RPG	R34
DoF	11M	11M	21M
Acc. (%)	13.4	**16.5**	16.0

• ResNet-RPG has lower training-validation accuracy gap on ImageNet classification and pose estimation.
• ResNet with RPG has higher performance on out-of-distribution dataset ObjectNet. RPGis trained on ImageNet only and directly evaluated on ObjectNet.

Accelerating RPG

RPG reduces model DoF. Could we prune or quantize it to reduce computation/inference time as well?

Pruning RPG

fine-grained pruning

	acc before	acc after ↓ DoF	acc drop	model DoF
R18-IMP [18]	92.3	90.5	1.8	274k
R18-RPG	95.0	93.0	2.0	274k

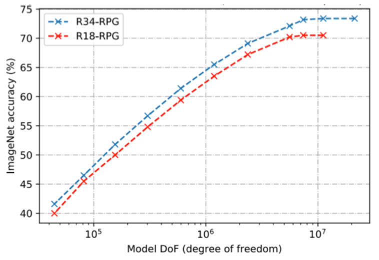
coarse-grained pruning

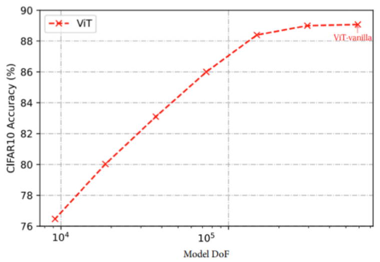
	DoF before pruning	Pruned acc.	FLOPs
R18-Knapsack	11.2M	69.35%	1.09e9
Pruned R18-RPG	5.6M	69.10%	1.09e9

Quantize RPG

	# Params	Acc before	Acc after ↓ quantization	Acc drop
R18-vanilla	11M	69.8	69.5	0.3
R18-RPG	5.6M	70.2	70.1	0.1

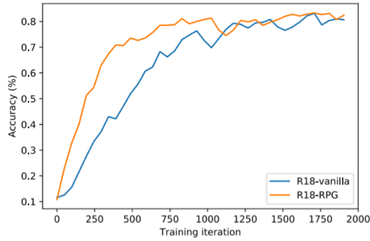
Log-Linear DoF-Accuracy Relationship



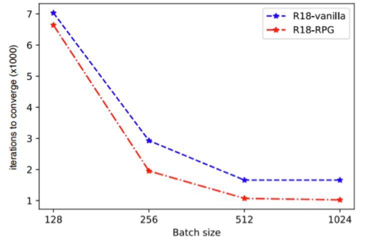


• Accuracy and model DoF follow a *power law* for both CNN and ViT.
• The exponents of the power laws are the same for ResNet18-RPG and ResNet34-RPG on ImageNet. The scaling law may be useful for estimating the network accuracy without training the network.
• RPG enables *under-parameterized* models for large-scale datasets such as ImageNet, which may unleash new studies and findings.

RPG Converges Faster



RPG converges faster than the vanilla model



RPG converges faster for different batch sizes