FractalNet: Ultra-Deep Neural Networks without Residuals

Gustav Larsson — *University of Chicago* — larsson@cs.uchicago.edu

Michael Maire — *TTI-Chicago* — mmaire@ttic.edu

Gregory Shakhnarovich — *TTI-Chicago* — greg@ttic.edu

Introduction

FractalNet is a neural network macro-architecture based on truncated fractals. A simple expansion rule leads to a powerful architecture with interesting properties:

Recursive The following rule describes how a fractal network is made deeper:

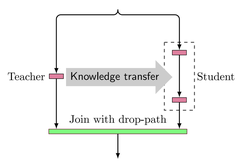
$$f_{C+1}(z) = [(f_C \circ f_C)(z)] \oplus [\mathrm{conv}(z)] \qquad f_1(z) = \mathrm{conv}(z)$$

Paths are joined ($\oplus$) with *element-wise mean*. An increase in depth is coupled with an increase in parallel paths, facilitating training.

Scalable in Depth Deeper network are generally harder to train. Fractal networks, however, are robust to choice of overall depth; make them deep enough and training will carve out a useful assembly of subnetworks.

Deep Supervision In order to tackle diminishing gradients for early layers, deep supervision uses auxiliary pathways with weights unused at test time. In fractal networks, deep supervision emerges naturally with no throw-away weights.

Student-Teacher Fractal networks with drop-path exhibit a student-teacher behavior between paths. Shallow paths converge quickly. Drop-path then promotes deeper paths (students) to be compatible with the shallow paths (teachers), creating a lateral knowledge transfer. See Figure 2.



Modules A common technique is to have modules of parallel paths with different receptive field sizes (Szegedy et al. (2015)). Fractal networks exhibit a similar property since the deeper paths have increasingly large receptive fields. A block expanded once can be seen as a parallel module with a 3×3 convolution and a non-linear 5×5 convolution.

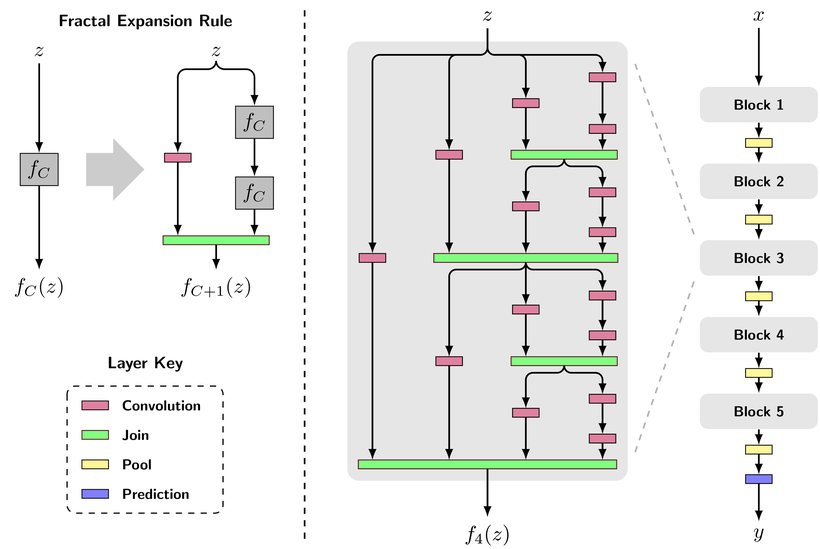


Figure 1: Fractal networks are based on a simple fractal expansion rule.

Results

Method	CIFAR-100	CIFAR-100+	CIFAR-100++	CIFAR-10	CIFAR-10+	CIFAR-10++	SVHN
FitResNet (Mishkin and Matas, 2016)	-	27.66	-	-	5.84	-	-
ResNet (He et al., 2016a)	-	-	-	-	6.61	-	-
ResNet by (Huang et al., 2016b)	44.76	27.22	-	13.63	6.41	-	2.01
Stochastic Depth (Huang et al., 2016b)	37.80	24.58	-	11.66	5.23	-	1.75
Identity Mapping (He et al., 2016b)	-	22.68	-	-	4.69	-	-
ResNet in ResNet (Targ et al., 2016)	-	22.90	-	-	5.01	-	-
Wide (Zagoruyko and Komodakis, 2016)	-	20.50	-	-	4.17	-	-
DenseNet-BC (Huang et al., 2016a)[1]	19.64	17.60	-	5.19	3.62	-	1.74
FractalNet (20 layers, 38.6M params)	35.34	23.30	22.85	10.18	5.22	5.11	2.01
+ drop-path + dropout	28.20	23.73	23.36	7.33	4.60	4.59	1.87
↳ deepest column alone	29.05	24.32	23.60	7.27	4.68	4.63	1.89
FractalNet (40 layers, 22.9M params)[2]	-	22.49	21.49	-	5.24	5.21	-

Table 1: **CIFAR-100/CIFAR-10/SVHN.** We compare test error (%) with other leading methods, trained with either no data augmentation, translation/mirroring (+), or more substantial augmentation (++).

Method	Top-1 (%)	Top-5 (%)
VGG-16	28.07	9.33
ResNet-34 C	24.19	7.40
FractalNet-34	24.12	7.39

Table 2: **ImageNet** (validation set, 10-crop).

[1]Densely connected networks (DenseNets) are concurrent work, appearing subsequent to our original arXiv paper on FractalNet.

[2]This deeper FractalNet has fewer parameters. We make deeper columns thinner and incorporate elements from Iandola et al. (2016).

References

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. *CVPR*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. *ECCV*.

Huang, G., Liu, Z., and Weinberger, K. Q. (2016a). Densely connected convolutional networks. *arXiv:1608.06993*.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. (2016b). Deep networks with stochastic depth. *ECCV*.

Iandola, F. N., Moskewicz, M. W., Ashraf, K., Han, S., Dally, W. J., and Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *arXiv:1602.07360*.

Mishkin, D. and Matas, J. (2016). All you need is a good init. *ICLR*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *CVPR*.

Targ, S., Almeida, D., and Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures. *arXiv:1603.08029*.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *BMVC*.

Regularization of Fractal Networks

We propose **drop-path**: Randomly drop subnetworks (paths). It provides:

Regularization Prevents co-adaptation between paths

Knowledge Transfer Promotes all operands of joins to be similar or compatible. As a result, if subsequent layers have grown dependent on the signal from a shallow path, drop-path will promote a lateral knowledge transfer to its deeper siblings.

Dropout: Randomly drop **activations**

Drop-connect: Randomly drop **weights**

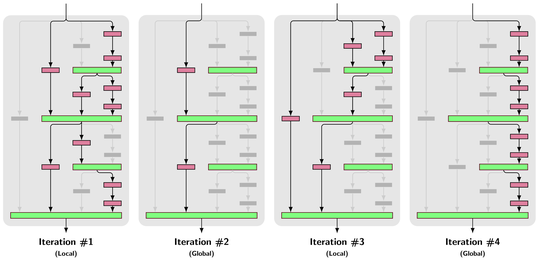
Drop-path: Randomly drop **paths**

We drop paths in two ways:

Local Operands of all joins are dropped with some fixed probability (we use 15%)

Global All paths are dropped except for a single-path (we use straight columns)

We use an even mixture between Local and Global.



Fractal Structure as a Training Apparatus

Fractal training with drop-path facilitates training deep networks. Even single-column performance is improved through fractal training:

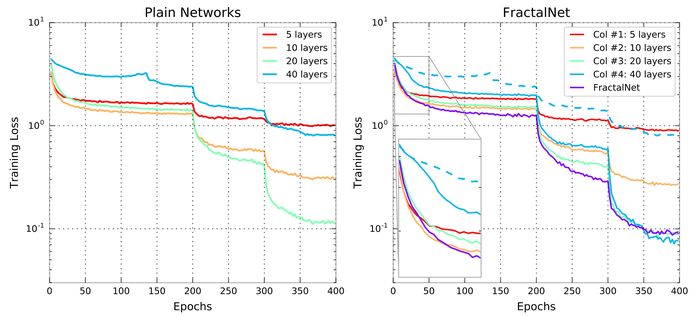


Figure 2: **Implicit deep supervision.** *Left:* Evolution of loss for plain networks of various depths on CIFAR-100. At 40 layers, we are unable to train the network satisfactorily. *Right:* A 4-column fractal network, with individual columns identical at test time to the plain networks on the left. Drop-path is used to facilitate training and promote single-column performance. Col #4 trains well and achieves the lowest loss among all.

Model	Depth	Train Loss	Error (%)
Plain	5	0.786	36.62
Plain	10	0.159	32.47
Plain	20	0.037	31.31
Plain	**40**	**0.580**	**38.84**
Fractal Col #1	5	0.677	37.23
Fractal Col #2	10	0.141	32.85
Fractal Col #3	20	0.029	31.31
Fractal Col #4	**40**	**0.016**	**31.75**
Fractal Full	**40**	**0.015**	**27.40**

Table 3: **Fractal structure as a training apparatus** (CIFAR-100++). Plain networks perform well if moderately deep, but exhibit worse convergence during training if instantiated with great depth. However, as a column trained within, and then extracted from, a fractal network with mixed drop-path, we recover a plain network that overcomes such depth limitation (possibly due to a student-teacher effect).

Columns	Depth	Parameters	Error (%)
1	5	0.3M	37.32
3	20	2.1M	27.69
4	40	4.8M	27.38
5	80	10.2M	26.46
6	160	21.1M	27.38

Table 4: **Ultra-deep fractal networks** (CIFAR-100++). Increasing depth greatly improves accuracy until eventual diminishing returns. Contrast with plain networks, which are not trainable if made too deep (Table 3).

github.com/gustavla/fractalnet

LaTeX TikZposter