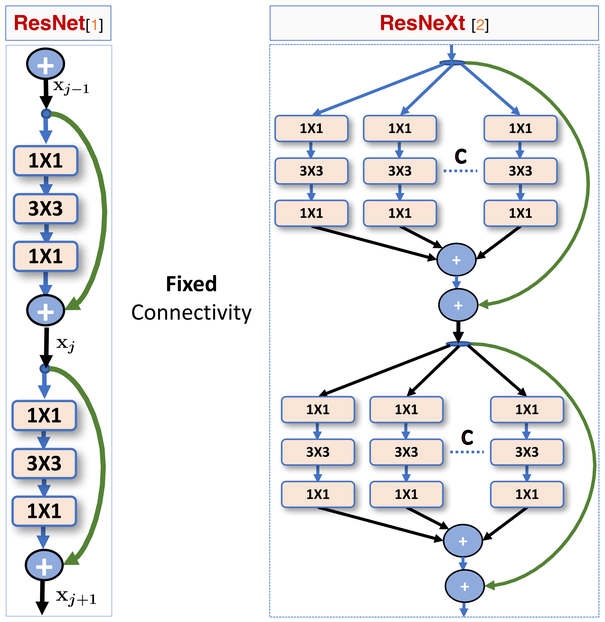
MaskConnect: Connectivity Learning by Gradient Descent

Karim Ahmed, Lorenzo Torresani

Dartmouth

Modular Networks



To combat the complexity in hand-designing architectures, prior work has adopted:

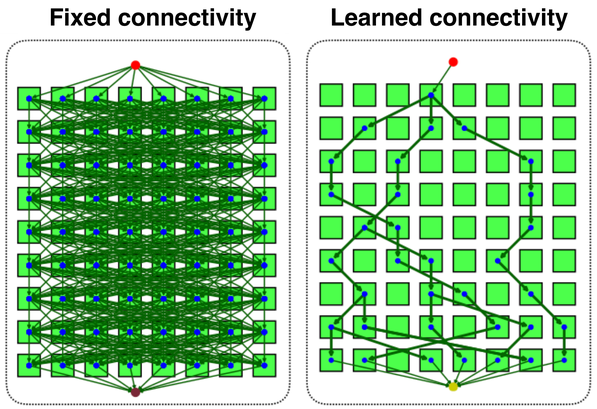
❑ Modularized design

→ Stacking building blocks of identical/similar topology

❑ Simple hand-defined connectivity rules

→ The input $\mathbf{x}_j$ of the j-th module is the output $\mathbf{y}_{j-1}$ of the preceding module: $\mathbf{x}_j \leftarrow \mathbf{y}_{j-1}$

→ Outputs of parallel blocks are added together



Technical Approach

Goal: learn branch connectivity from data by optimizing training objective

❑ The input of the j-th module is given by:

$$\mathbf{x}_j = \sum_{k=1}^{j-1} m_{j,k} \cdot \mathbf{y}_k$$

Learned binary mask defining active connections between module j and a preceding module k

active if: $m_{j,k} = 1$

inactive if: $m_{j,k} = 0$

❑ During training, we update auxiliary real-valued masks $\tilde{\mathbf{m}}_j$

- Stochastically binarize $\tilde{\mathbf{m}}_j \in [0,1]^{j-1}$ into $\mathbf{m}_j \in \{0,1\}^{j-1}$

  Constraint: $\sum_{k=1}^{j-1} m_{j,k} = K$ → "fan-in" hyperparameter: # of active entries in each mask

  $a_1, a_2, \ldots, a_K \leftarrow \mathrm{Mult}(\tilde{m}_{j,1}, \tilde{m}_{j,2}, \ldots, \tilde{m}_{j,j-1})$

  Set active binary masks based on drawn samples: $m_{j,a_k} \leftarrow 1$ for $k = 1, ..., K$

- Minimize $\ell(\theta, \mathbf{m})$ using backpropagation

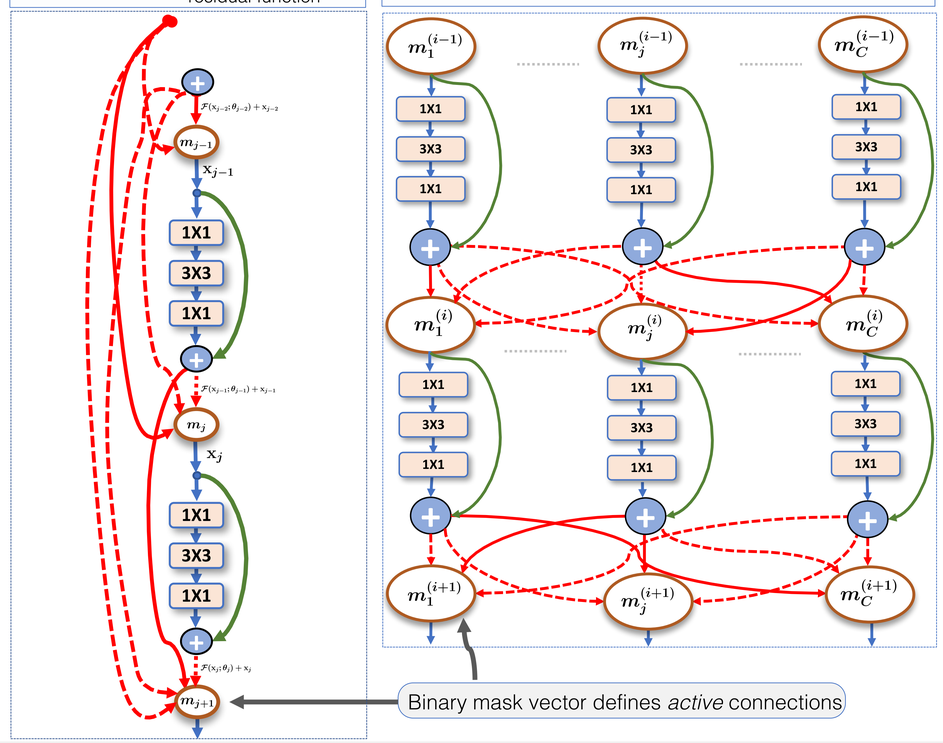
❑ **MaskConnect** applied to ***ResNet***

$$\mathbf{x}_j \leftarrow \sum_{k=1}^{j-1} m_{j,k} \underbrace{[\mathcal{F}(\mathbf{x}_k; \theta_k) + \mathbf{x}_k]}_{\mathbf{y}_k}$$

residual function

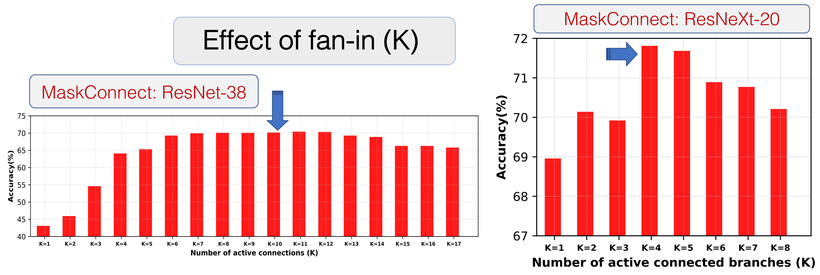
❑ **MaskConnect** applied to ***ResNeXt***

$$\mathbf{x}_j^{(i)} \leftarrow \sum_{k=1}^{C} m_{j,k}^{(i)} \underbrace{\left[\mathcal{F}(\mathbf{x}_k^{(i-1)}; \theta_k^{(i-1)}) + \mathbf{x}_k^{(i-1)}\right]}_{\mathbf{y}_k^{(i-1)}}$$



Results

CIFAR-100 (100 classes, 50,000 training examples)



Model	Connectivity	Accuracy (%)
ResNet-38	**Learned, K=10**	**70.40**
	Fixed, K=1 [1]	68.54
	Fixed-Random, K=10	62.67
ResNet-74	**Learned, K=15**	**72.81**
	Fixed, K=1 [1]	70.64
	Fixed-Random, K=15	66.93
ResNet-110	**Learned, K=20**	**73.15**
	Fixed, K=1 [1]	71.21
	Fixed-Random, K=20	67.22

Model {depth,width,#branches}	Connectivity	Params Train	Params Test	Accuracy (%) best (mean± std)
ResNeXt {29,8,8}	**Learned, K=1**	0.86M	0.65M	**73.91** (73.76± 0.14)
	Learned, K=4	0.86M	0.81M	**75.89** (75.77± 0.12)
	Fixed-Full, K=8 [2]	0.86M	0.86M	73.52 (73.37± 0.13)
	Fixed-Random, K=4	0.86M	0.85M	72.85 (72.66± 0.24)
ResNeXt {29,64,8}	**Learned, K=1**	34.4M	20.5M	**82.31** (82.15± 0.15)
	Learned, K=4	34.4M	32.1M	**84.05** (83.94± 0.11)
	Fixed-Full, K=8 [2]	34.4M	34.4M	82.23 (82.12± 0.12)
	Fixed-Random, K=4	34.4M	34.3M	81.96 (81.73± 0.20)

- MaskConnect with K=4 improves ResNeXt by 2.2%
- MaskConnect with K=1 reduces # parameters by up to 40% compared to ResNeXt

ImageNet (1000 classes, 1.28M training examples)

Model {depth, width, #branches}	Connectivity	Accuracy (%) Top-1	Accuracy (%) Top-5
ResNeXt {50,4,32}	**Learned, K=16**	**79.1**	**94.1**
	Fixed-Full, K=32 [2]	77.8	93.3
ResNeXt {101,4,32}	**Learned, K=16**	**79.5**	**94.5**
	Fixed-Full, K=32 [2]	78.8	94.1
ResNeXt {101,4,64}	**Learned, K=32**	**79.8**	**94.8**
	Fixed-Full, K=64 [2]	79.6	94.8

Note: although we tested MaskConnect on ResNet and ResNeXt, our method can be applied to any modularized architecture and any loss function.

References

[1] He et al. "*Deep residual learning for image recognition.*" CVPR. 2016.

[2] Xie et al. "*Aggregated residual transformations for deep neural networks.*" CVPR 2017.