Connectivity Learning in Multi-Branch Networks
Karim Ahmed, Lorenzo Torresani

Extended paper: arXiv:1709.09582

Multi-branch Networks
Multi-branch architecture are widely used. E.g., in image categorization:

- Modularized design: stacking multi-branch building blocks of identical/similar topology.
- Simple, uniform connectivity rules: feature maps from branches are always either added or concatenated.

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

- Modularized design
- Simple, uniform connectivity rules

Fixed connectivity Learned connectivity

ResNeXt [Xie et al., CVPR 2017] Our Approach

K (#Connected branches) = 1

Technical Approach
Goal: learn branch connectivity from data by optimizing training objective

- Minimize $\ell(y, \hat{y})$ using backpropagation over $g$ and $\theta$
- During training, we update auxiliary real-valued masks $g^{(i)}_j \in [0, 1]^C$
- Constrain the number of active input connections (fan-in) to each block to be a constant, $K$ (a hyperparameter)

Forward propagation:
1. Stochastically binarize $g^{(i)}_j \in [0, 1]^C$ into $g^{(i)}_j \in \{0, 1\}^C$ s.t. $\sum_{k=1}^C g^{(i)}_j k = K$
2. Perform forward pass using binary masks $g^{(i)}_j \in \{0, 1\}^C$

\[
x^{(i)}_j = \sum_{k=1}^C g^{(i)}_j k x^{(i-1)}_k
\]

\[
y^{(i)}_j = x^{(i)}_j + \beta f^{(i)}_j + \gamma_{\theta}^{(i)}
\]

Parameter update:
Compute $\frac{\partial \ell}{\partial g_{j,k}}$, and update real-valued masks $\tilde{g}_{j,k}$.

Results

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

Effect of fan-in ($K$)

Full, fixed connectivity, corresponding to original ResNeXt [Xie et al., 2016]

Architecture:
Depth: 20
Bottleneck width: 4
Cardinality: 8

CIFAR-10 (10 classes, 50K training examples)

ImageNet (1000 classes, 1.28M training examples)

References