Deep Networks with Stochastic Depth

Gao Huang[1], Yu Sun[1], Zhuang Liu[2], Daniel Sedral[1], Kilian Weinberger[1]

*Equal contribution

**Motivation**
Training very deep networks is difficult:
- Gradients vanish and forward signals diminish
- Long training time
- Overfitting

**Question:** Can we use short networks during training, but use deep networks during testing?

**Idea:** For each mini-batch, randomly drop a subset of layers and bypass them with the identity function!

**Method**

**Stochastic depth network at training time**

- Mini-batch 1
- Mini-batch 2
- Mini-batch 3

\[ H_\ell = \text{ReLU}(b_\ell f_\ell(H_{\ell-1} + \text{id}(H_{\ell-1}))) \]

- Bernoulli random variable
- Linear decay rule for survival probabilities
- Basic block (Similar to ResNets, He et al, CVPR'16)

**Expected network depth**

\[ E(\hat{L}) = \sum_{\ell=1}^{L} p_\ell = (3L - 1)/4 \approx 3L/4 \]

**Stochastic depth network at test time**

At test time

\[ H^{\text{Test}}_\ell = \text{ReLU}(b_\ell f_\ell(H^{\text{Test}}_{\ell-1}, W_\ell)) + H^{\text{Test}}_{\ell-1} \]

All layers are on, but outputs of \( f_\ell \) are down weighted by their corresponding survival probabilities.

**Advantages of stochastic depth**
- Alleviates the gradient and signal vanishing problem
- Speeds up the training process
- Performs regularization and improves generalization (implicit ensemble of \( 2^L \) models)

**Results**

**Classification**

Training time:
- CIFAR10+ CIFAR100+ SVHN
  - Constant Depth: 100 layers
  - Stochastic Depth: 15 layers

\[ \sim 25\% \text{ faster} \]

**Analysis**

**Gradient strength**

The gradient strength at the input layer

**Hyper-parameter \( p_L \)**

Varying \( p_L \) with fixed depth

Varying \( p_L \) with different depth

**Extension (DenseNets)**

Densely Connected Convolutional Networks (https://arxiv.org/abs/1608.06993)
- From implicit long-range connections to explicit long-range connections
- Learn more compact models!
- And more accurate!

**Code**

https://github.com/yueatsprograms/Stochastic_Depth