# SkipNet: Learning Dynamic Routing in Convolutional Networks

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, Joseph E. Gonzalez

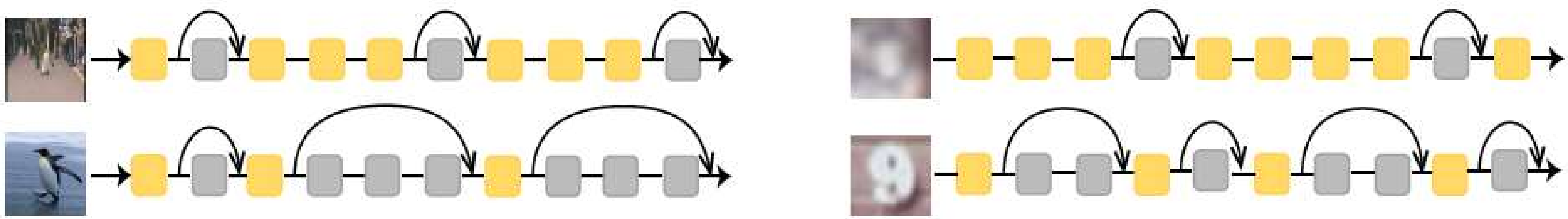
University of California, Berkeley
Nanjing University

ECCV 2018 conference paper

Representor: Qiyun Cheng

# Background and Motivations

1. The convolutional neural networks become deeper and deeper.
2. The high cost only benefits the accuracy for a few percentage points.
3. The optimal number of layers is decided by the input.



The SkipNet learns to skip convolutional layers on a per-input basis.
More layers are executed for challenging images than easy images.

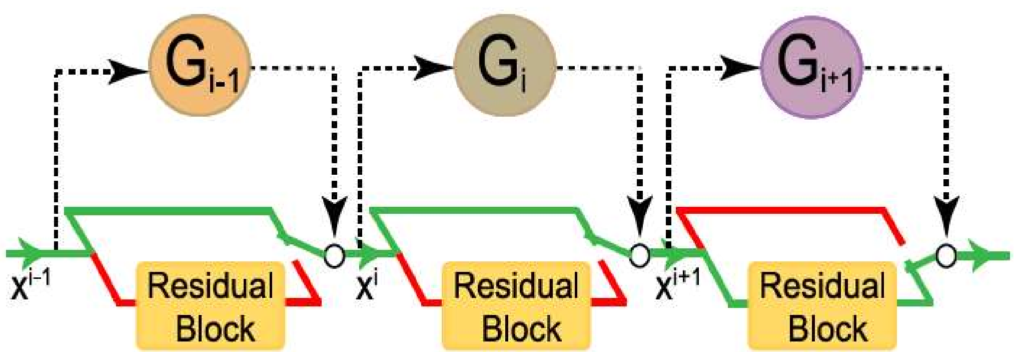
# SkipNet Model Design

$$\mathbf{x}^{i+1} = G^i(\mathbf{x}^i)F^i(\mathbf{x}^i) + (1 - G^i(\mathbf{x}^i))\mathbf{x}^i$$

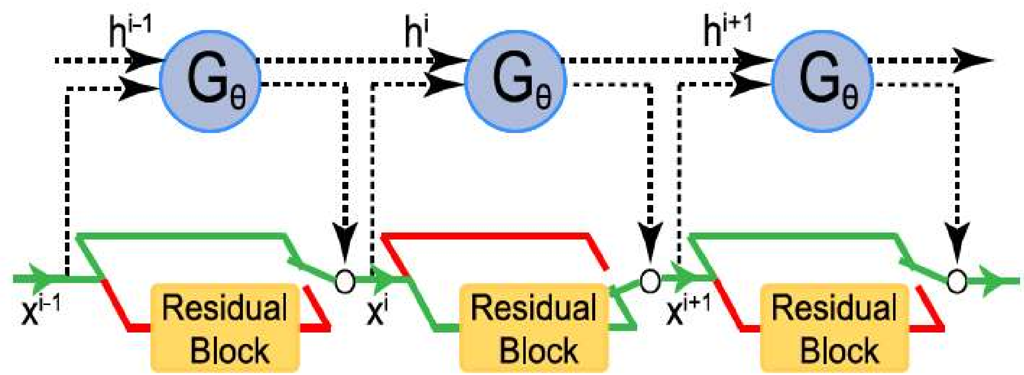
$x^i$ *is the input and* $F^i(x^i)$ *is the output of the* $i^{th}$ *layer*
$G^i(x^i) \in \{0, 1\}$ *is the gating function for the layer* $i$
$x^{i+1}$ *is the output of the gated layer*

$$\mathbf{x}^{i+1}_{\text{ResNet}} = F^i(\mathbf{x}^i_{\text{ResNet}}) + \mathbf{x}^i_{\text{ResNet}}$$

*Pooling* $x^i$ *to match the dimensions of* $F^i(x^i)$



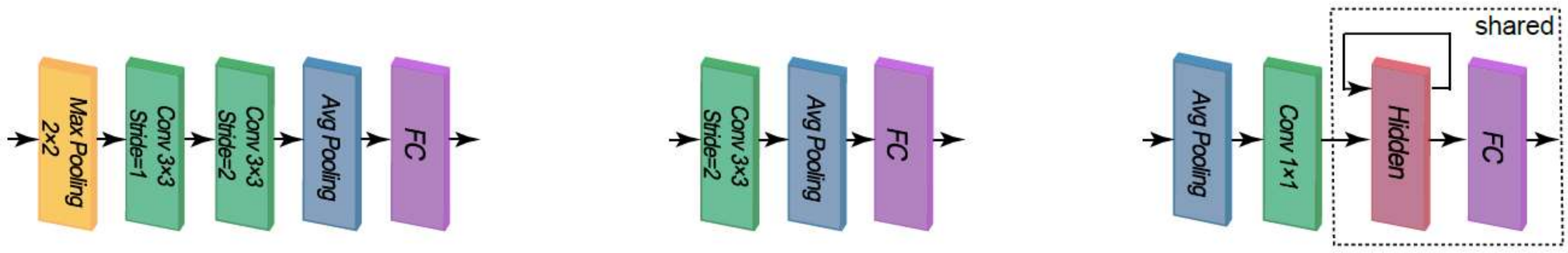

(a) Feed-forward Gate




(b) Recurrent Gate

Gating must be both computationally cheap and accuracy enough

# Gating Network Design




(a) FFGate-I (b) FFGate-II (c) RNNGate

FFGate-I : 19% computational cost of the residual blocks, better for shallower networks;
FFGate-II: 12.5% computational cost of the residual blocks, better for more than 100 layers;

RNNGate: Contains a one-layer Long Short Term Memory with both input and hidden unit size of 10.
The cost is 0.04% of the cost of the residual blocks.

In later experiments, the recurrent gate dominates the feed-forward gates in both prediction accuracy and computation cost. Therefore, the recurrent gate design is considered to be better in capturing the cross-layer dependencies.

# Skipping Policy

Discrete decision process for gates: whether the layer is skipped or executed.

Skipping policy: $\pi(\mathbf{x}^i, i) = \mathbb{P}(G^i(\mathbf{x}^i) = g_i)$ Execute: $g_i = 1$ Skip: $g_i = 0$

Gating sequence: $\mathbf{g} = [g_1, \ldots, g_N] \sim \pi_{F_\theta}$

$F_\theta = [F_\theta^1, \ldots, F_\theta^N]$ is the sequence of network layers (including the gating modules) parameterized by θ

Objective function:

$$\min \mathcal{J}(\theta) = \min \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{g}} L_\theta(\mathbf{g}, \mathbf{x})$$
$$= \min \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{g}} \left[ \mathcal{L}(\hat{y}(\mathbf{x}, F_\theta, \mathbf{g}), y) - \frac{\alpha}{N} \sum_{i=1}^{N} R_i \right]$$

$R_i = (1 - g_i)C_i$: reward of each gating module;
$C_i$: constant for the cost of executing $F_i$ (all $F_i$ are same so $C_i = 1$);
α: tuning parameter to trade-off minimizing the prediction loss and maximizing the gate rewards.

# Skipping Policy

Gradient: $\pi_{F_\theta}(\mathbf{x}) = p_\theta(\mathbf{g}|\mathbf{x})$ $\mathcal{L} = \mathcal{L}(\hat{y}(\mathbf{x}, F_\theta, \mathbf{g}), y)$ $r_i = -[\hat{\mathcal{L}} - \frac{\alpha}{N}\sum_{j=i}^{N} R_j]$

$$\begin{aligned}\nabla_\theta \mathcal{J}(\theta) &= \mathbb{E}_{\mathbf{x}} \nabla_\theta \sum_{\mathbf{g}} p_\theta(\mathbf{g}|\mathbf{x}) L_\theta(\mathbf{g}, \mathbf{x}) \\ &= \mathbb{E}_{\mathbf{x}} \sum_{\mathbf{g}} p_\theta(\mathbf{g}|\mathbf{x}) \nabla_\theta \mathcal{L} + \mathbb{E}_{\mathbf{x}} \sum_{\mathbf{g}} p_\theta(\mathbf{g}|\mathbf{x}) \nabla_\theta \log p_\theta(\mathbf{g}|\mathbf{x}) L_\theta(\mathbf{g}, \mathbf{x}) \\ &= \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{g}} \nabla_\theta \mathcal{L} - \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{g}} \sum_{i=1}^{N} \nabla_\theta \log p_\theta(g_i|\mathbf{x}) r_i.\end{aligned}$$

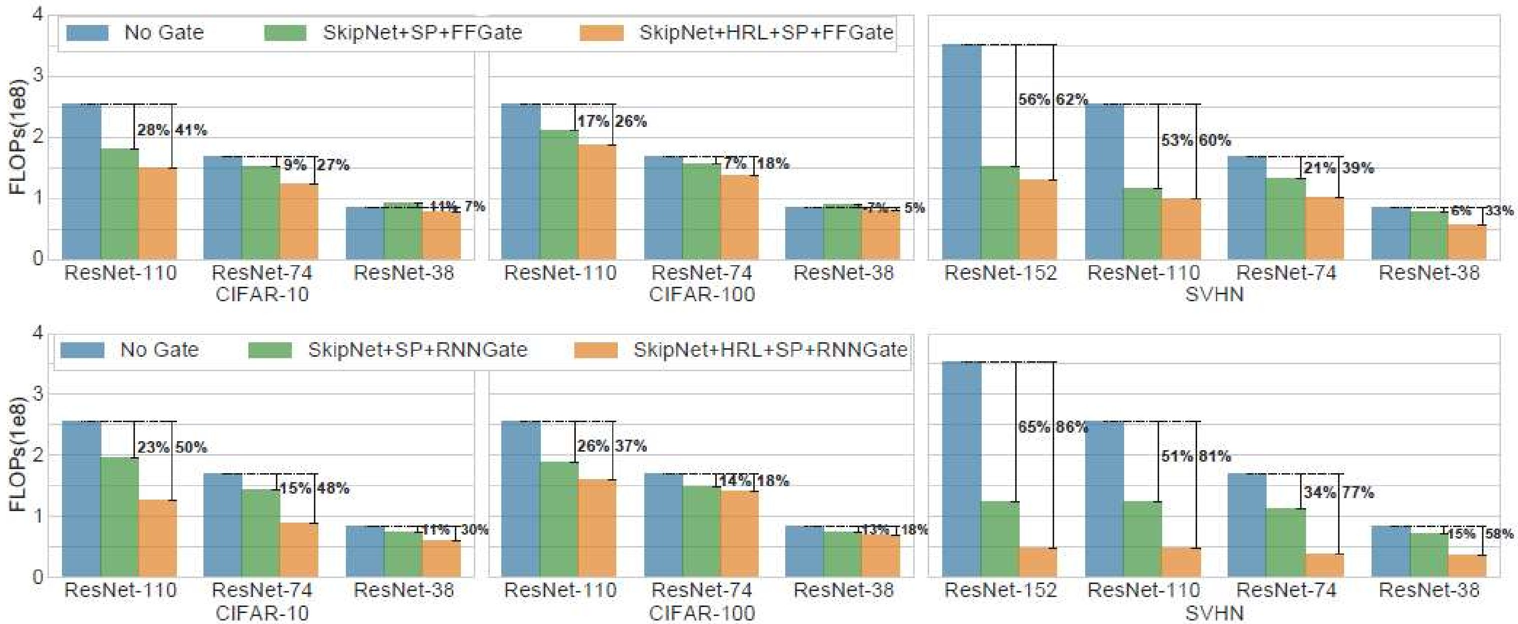
supervised learning loss

reinforce gradient

$r_i$ is the cumulative future rewards associated the gating modules

# Experiments

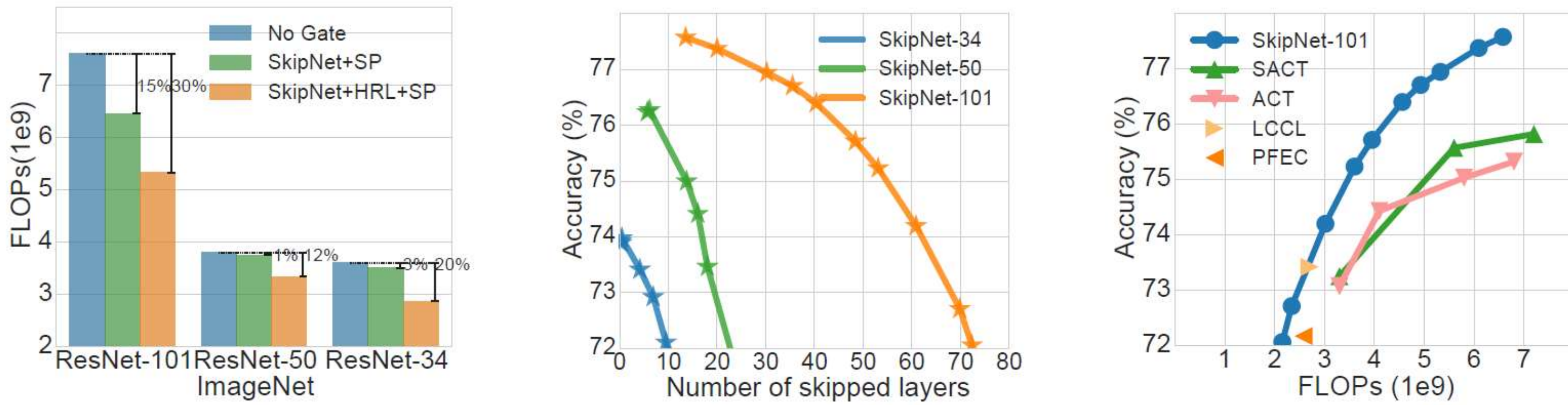
Computation reduction while preserving full network accuracy




Computation reduction of SkipNet+SP and SkipNet+HRL+SP with feed-forward gates and recurrent gates while preserving the full network accuracy. The computation cost includes the computation of gates. We are able to reduce computation costs by 50%, 37% and 86% of the deepest models on the CIFAR-10, 100 and SVHN data. Compared to using SP only, fine-tuning with HRL can gain another 10% or more computation reduction. Since feed-forward gates are more expensive, SkipNets with recurrent gates generally achieve greater cost savings

# Experiments

Trade-off computational cost and accuracy

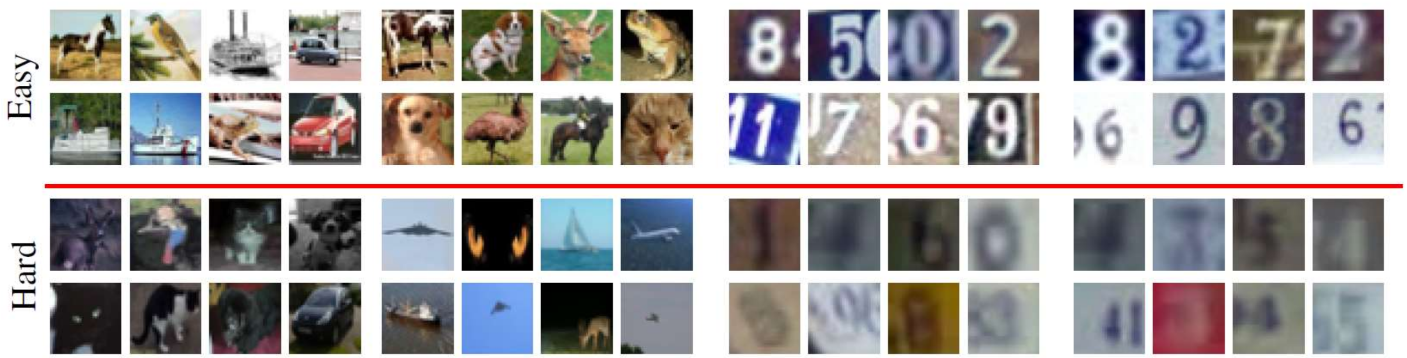



(a) Computation Reduction (b) Acc.-compt. Trade-off (c) Comparison with Others

(a) Computation reduction (12 - 30%) achieved by SkipNets with RNNGates while preserving full network accuracy.
(b) Trade-off between accuracy and cost under different α. With small α , the computation drops faster than the decrease of accuracy.
(c) Comparison of SkipNet with state-of-the-art models. SkipNet consistently outperforms existing approaches on both benchmarks under various trade-off between computational cost and prediction accuracy.

# Experiments

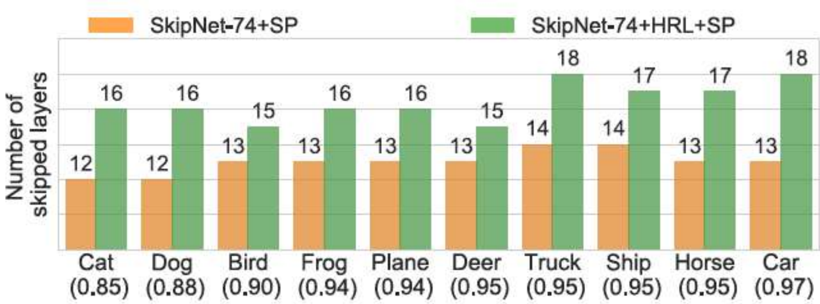
Skipping Behavior Analysis and Visualization

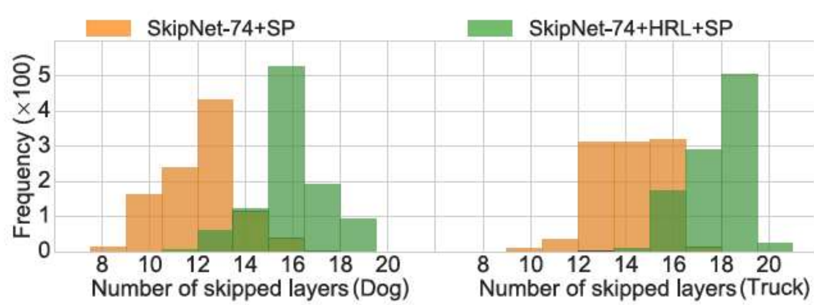



(a) SkipNet +FFGate CIFAR-10 (b) SkipNet +RNNGate CIFAR-10 (c) SkipNet +FFGate SVHN (d) SkipNet +RNNGate SVHN

Easy (brighter and clearer): more than 15 layers skipped.

Hard (dark and blurry): less than 8 layers skipped.




(a) Median of number of skipped layers (b) Distribution of number of skipped layers

# Conclusion

1. SkipNet architecture learns to dynamically skip redundant layers on a per-input basis, without sacrificing prediction accuracy.
2. Evaluated on four benchmark datasets, SkipNet is able to reduce computation substantially while preserving the original accuracy.
3. The dynamic architectures offer the potential to be more computationally efficient and improve accuracy by specializing and reusing individual components.

Rensselaer

why not change the world?®