# SkipNet: Learning Dynamic Routing in Convolutional Networks

Xin Wang[1] Fisher Yu[1] Zi-Yi Dou[2] Joseph E. Gonzalez[1]
[1]UC Berkeley [2]Nanjing University

## 1 INTRODUCTION

Deep convolutional neural networks are the enabling technology behind the recent rapid progress in computer vision. A growing body of research in convolutional network design [4, 8, 12, 13] reveals a clear trend: *deeper networks are more accurate*. Consequently, the best-performing image recognition networks have tens of millions of parameters and hundreds of layers. While commodity GPUs are able to substantially accelerate training, the high computation cost of very deep networks hinders their deployment in latency sensitive end-user applications and on low-power devices. Moreover, the depth of these networks results in fundamental and significant increases in prediction latency.

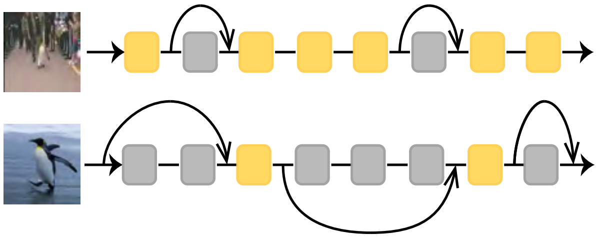


**Figure 1: The SkipNet learns to route images through a subset of layers on a per-input basis. Challenging images (top) are routed through more layers than easy images (bottom).**

The continuous improvements in accuracy, while significant, are small relative to the growth in model complexity. A network that doubles in depth may only improve by a few percentage points on key benchmarks. These small improvements are critical to the adoption of these models in real-world applications; however, they imply that only a small fraction of images require very deep representations and thus the vast majority of images could be accurately processed using shallower architectures.

In this paper, we study the design of *dynamically routed networks* (SkipNets), convolutional networks that determine which layers of a convolutional neural network should be included when processing a given image, illustrated in Fig. 1. We frame the routing problem as a sequential decision problem in which the outputs of previous layers are used to decide whether to bypass the subsequent layer. The objective in the routing problem is then to bypass as many layers as possible while retaining the accuracy of the full network. Not only can routing policies significantly reduce the average cost of model inference they also provide insight into the diminishing return and role of individual layers.

While conceptually simple, learning an efficient routing policy is challenging. To achieve a reduction in computation, we need to bypass the correct layers in the network. This inherently discrete decision is not differentiable, and therefore precludes the application of established supervised learning methods based on gradient based optimization. Although one could introduce a soft approximation similar to soft-attention techniques [1, 14, 15], we show that the subsequent hard thresholding required to achieve a reduction in the cost of computation results in low prediction accuracy. Similar to our goals, recent research has made progress in applying reinforcement learning (RL) techniques to address hard attention in recurrent models [2, 9]. While these techniques are promising, in our experiments we find that these RL based techniques are brittle, often getting stuck in poor local minima and producing networks that are not competitive with the state-of-the-art.

## 2 APPROACH

Dynamically routed networks are convolutional networks in which layers are selectively included or excluded for a given input. The per-input selection of layers is accomplished using small gating networks that are interposed between layers. The gating networks map the output of the previous layer or group of layers to a binary decision to execute or bypass the subsequent layer or group of layers as illustrated in Fig. 2.

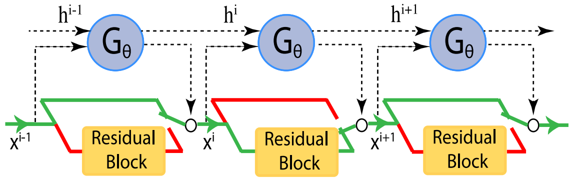


**Figure 2: SkipNet with recurrent gates. A unified recurrent gate is shared across all the blocks.**

More precisely, let $\mathbf{x}^i$ be the input and $F^i(\mathbf{x}^i)$ be the output of the $i^{\text{th}}$ layer or group of layers, then we define the output of the gated layer (or group of layers) as:

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

where $G^i(\mathbf{x}^i) \in \{0, 1\}$ is the gating function for layer $i$. In order for Eq. 1 to be well defined, we require $F^i(\mathbf{x}^i)$ and $\mathbf{x}^i$ to have the same dimensions. This requirement is satisfied by commonly used residual network architectures and can be easily addressed by pooling or up-sampling $\mathbf{x}^i$ so its dimensions match that of $F^i(\mathbf{x}^i)$. In our work, we use ResNets [4] as our base models.

When designing the gating modules, we explore gating network designs including both basic feed-forward convolutional architectures and recurrent networks with varying degrees of parameter sharing to address the trade-off between expressivity and computational cost. In our experiments, we find the light-weighted recurrent gate (roughly 0.04% of the computation of residual blocks) whose main buiding block is a single layer *Long Short Term Memory* (LSTM) [5] with hidden unit size of 10 outperform other designs.

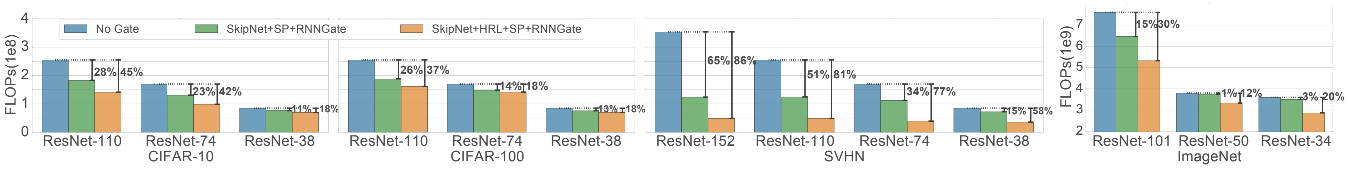


**Figure 3:** **Computation reduction of SkipNet +SP and SkipNet +HRL+SP with recurrent gate on benchmark datasets.**

## 2.1 Routing Policy Learning with Hybrid RL

Because SkipNets make a sequence of discrete decisions, one at each gated layer, we frame the task of estimating the gating function in the context of policy estimation through reinforcement learning. We define the routing policy:

$$\pi(\mathbf{x}^i, i) = \mathbb{P}(G^i(\mathbf{x}^i) = g_i) \qquad (2)$$

as a function from the input $\mathbf{x}^i$ to the probability distribution over the gate action $g_i$ to execute ($g_i = 1$) or skip ($g_i = 0$) layer $i$. We define a sample sequence of gating decisions from the routing policy as $\mathbf{g} = [g_1, \ldots, g_N] \sim \pi_{F_\theta}(\mathbf{x})$ where $F_\theta = \left[F_\theta^1, \ldots, F_\theta^N\right]$ is the sequence of network layers parameterized by $\theta$.

We define the joint objective function to maximize accuracy and gate rewards:

$$\mathcal{J}(\theta, \pi) = \mathbb{E}_{\mathbf{g}}[\mathcal{L}_\theta(\hat{y}(\mathbf{x}, F_\theta, \mathbf{g}), y)] + \mathbb{E}_{\mathbf{g}}\left[\sum_{i=1}^{N} R_i\right] \qquad (3)$$

$$\approx \mathcal{L}_\theta(\hat{y}(\mathbf{x}, F_\theta, \mathbf{g} \sim \pi(\mathbf{x})), y) + \mathbb{E}_{\mathbf{g}}\left[\sum_{i=1}^{N} R_i\right], \qquad (4)$$

where $\mathcal{L}$ is the log likelihood of the true label $y$ given the SkipNet prediction $\hat{y}$. The second component in Equation 4 (referred as $\mathcal{J}_{\text{hybrid}}$) is the expected rewards for gate decisions, whose gradients can obtained by REINFORCE [16] algorithm, but the first component becomes the loss of supervised classification and its gradient can be calculated directly. We refer optimizing this hybrid objective function as *hybrid reinforcement learning*

The gate reward $R_i$ for gate $i$ is defined as the future rewards for that gate:

$$R_i = \frac{\alpha}{N} \sum_{j=0}^{N-i} (1 - g_i) C_i + \mathcal{L}(\hat{y}(\mathbf{x}, F_\theta, \mathbf{g}), y). \qquad (5)$$

The constant $C_i$ is the cost of executing $F^i$ and the term $(1 - g_i)C_i$ reflects the reward associated with *skipping* $F^i$. In our experiments, all $F^i$ have the same cost and so we set $C_i = 1$. Finally, $\alpha$ is a tuning parameter that allows us to trade-off the competing goals of maximizing prediction accuracy and minimizing the computation.

## 2.2 Supervised Pre-training

Optimizing Eq. 4 starting from random parameters also consistently produces models with poor prediction accuracy. We conjecture that the degraded ability to learn an accurate model is due to interference between policy learning and image representation learning.

We relax the gate outputs $G(\mathbf{x})$ in Eq. 1 to continuous values, i.e. approximating $G(\mathbf{x})$ by $S(\mathbf{x}) \in [0, 1]$. We round the output gating probability of the routing modules to 0 or 1 in the forward pass. During backpropagation we use the soft-max approximation [6]. That is, the gate is restored to soft outputs and the gradients to the soft-max outputs can be calculated accordingly.

## 3 RESULTS

We evaluate SkipNets, using ResNets [4] as the base models, on the CIFAR-10 [7], CIFAR-100 [7], SVHN [10] and ImageNet 2012 [11] datasets. We show that with the hybrid learning procedure, SkipNets learn routing policies that significantly reduce model inference costs (45% on the CIFAR-10 dataset, 37% on the CIFAR-100 dataset, 86% on the SVHN dataset and 30% on the ImageNet dataset) while preserving accuracy in Fig. 3. We show SkipNet outperforms spatially adaptive computation time (SACT) and adaptive computation time (ACT) networks [3] on the ImageNet benchmark shown in Fig. 4.

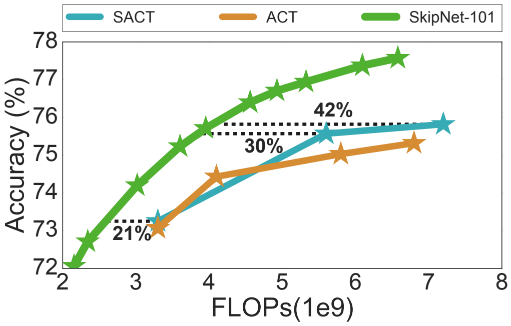
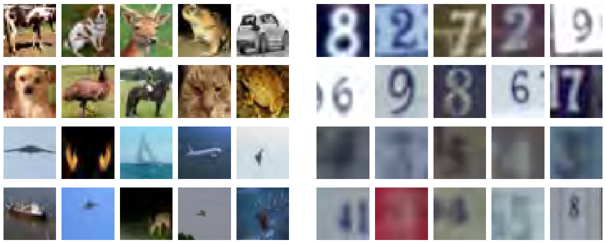


**Figure 4:** **Comparison of SkipNet-101 with SACT and ACT. SkipNet reduces up to 42% of the computation of SACT with the same accuracy.**



**(a) SkipNet +RNNGate CIFAR-10**

**(b) SkipNet +RNNGate SVHN**

**Figure 5:** **Visualization of *easy* (top two rows) and *hard* (bottom two rows) images in the CIFAR-10 and SVHN. Easy examples are more bright and clear while hard examples tend to be dark and blurry.**

We also visualize the routing behavior of the learned routing policy in Fig. 5 by grouping images by the number of skipped layers. It reveals that it learns to identify more challenging images and route those images through more layers of the network.

# REFERENCES

[1] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. 2015. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*.

[2] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vol. 1, 484–495.

[3] Michael Figurnov, Maxwell D. Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. 2017. Spatially adaptive computation time for residual networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*. (July 2017).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9, 8, 1735–1780.

[6] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

[7] Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Tech. rep.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

[9] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212.

[10] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning* number 2. Vol. 2011, 5.

[11] Olga Russakovsky et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 3, 211–252.

[12] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

[15] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, 2692–2700.

[16] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8, 3-4, 229–256.