NetTailor: Tuning the architecture, not just the weights

Pedro Morgado **Nuno Vasconcelos**
University of California, San Diego

1 Introduction

Real-world applications of object recognition often require the solution of *multiple* tasks in a single platform. Under the standard paradigm of network fine-tuning, an entirely new CNN is learned per task, and the final network size is independent of task complexity. This is wasteful, since simple tasks require smaller networks than more complex tasks, and limits the number of tasks that can be solved simultaneously. To address these problems, we propose a transfer learning procedure, denoted NETTAILOR, in which layers of a pre-trained CNN are used as universal blocks that can be combined with small task-specific layers to generate new networks. Besides minimizing classification error, the new network is trained to mimic the internal activations of a strong unconstrained CNN, and minimize its complexity by the combination of 1) a soft-attention mechanism over blocks and 2) complexity regularization constraints. In this way, NETTAILOR can adapt the network architecture, not just its weights, to the target task. Experiments show that networks adapted to simple tasks, such as character or traffic sign recognition, become significantly smaller than those adapted to hard tasks, such as fine-grained recognition. More importantly, due to the modular nature of the procedure, this reduction in network complexity is achieved without compromise of either parameter sharing across tasks, or classification accuracy.

2 NetTailor

A CNN implements a function $f(\mathbf{x}) = (G_L \circ G_{L-1} \circ \cdots \circ G_1)(\mathbf{x})$ by composing L computational blocks $G_l(\mathbf{x})$ consisting of simple operations, such as convolutions, spatial pooling, normalization among others. While the blocks $G_l(\mathbf{x})$ differ with the CNN model, they are often large, both in terms of computation and storage. Thus, in order to accomodate multiple tasks, it is desirable to share network resources between them (2). In this work, we introduce a new transfer technique, denoted NETTAILOR, which adapts the architecture of a pre-trained model to a target task, while reusing layers of a pre-trained CNN as universal blocks shared across tasks. The NETTAILOR procedure can be summarized as follows.

1. Train the *teacher network* by fine-tuning a pre-trained network on the target task.
2. Define the *student network* by *augmenting* the pre-trained network with task-specific low-complexity *proxy layers*.
3. Train the task-specific parameters of the student network on the target task to *mimic* the internal activations of the teacher, while imposing *complexity constraints* that encourage the use of low-complexity proxy layers over high-complexity pre-trained blocks.
4. Prune layers with low impact on network performance, and fine-tune the remaining task-specific parameters.

Student network architecture The main architectural component introduced in this work is the augmentation of the pre-trained network. Each layer G_l of the pre-trained network is augmented with a set of lean proxy layers $\{A_p^l(\cdot)\}_{p=1}^{l-1}$ that introduce a skip connection between layers p and l. As the name suggests, proxy layers aim to approximate and substitute the large pre-trained blocks $G_l(\cdot)$ whenever possible. The output activation $\mathbf{x}_l$ of layer l is computed by pooling all blocks

$$\mathbf{x}_l = \alpha_l^l G_l(\mathbf{x}_{l-1}) + \textstyle\sum_{p=1}^{l-1} \alpha_p^l A_p^l(\mathbf{x}_p), \tag{1}$$

where $\{\alpha_p^l\}_{p=1}^l \in [0,1]$ are a set of scalars that enable or disable the different network paths.

Proxy layers are forced to compete with each other to minimize the propagation of redundant information through the network. This is accomplished by introducing a set of auxiliary parameters a_p^l and computing α_p^l as the softmax across all paths merging into layer l, i.e. $\alpha_p^l = \frac{e^{a_p^l}}{\sum_k e^{a_k^l}}$.

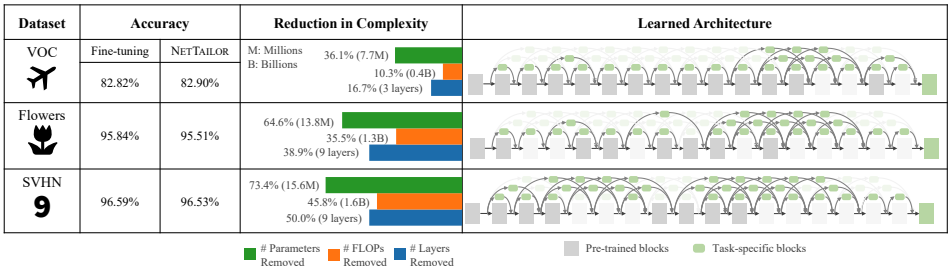

Figure 1: Reduction of network complexity and final architecture after adapting ResNet34 using NETTAILOR.

Differential complexity constrains In the complexity-aware pooling block, scalars $\{\alpha_p^l\}_{p=1}^l$ act as a soft-attention mechanism that selects which blocks to use for the target task. Let $B_i^j(\cdot)$ denote the computational block associated with path $i \rightarrow j$, and $\mathcal{C}_i^j$ its complexity. A block $B_i^j(\cdot)$ can be removed if α_i^j is close to zero. Under the modeling assumption that the probability of exclusion is given by $P(R_i^j) = 1 - \alpha_i^j$, the expected complexity of block B_i^j is $E[\mathcal{C}_i^j] = \mathcal{C}_i^j(1 - P(R_i^j)$. Complexity constrains can thus be imposed by minimizing the expected network complexity $E[\mathcal{C}] = \sum_{i,j} E[\mathcal{C}_i^j]$. Note that unlike recent network architecture search approaches (3), NetTailor relies on differentiable optimization.

Mimicking the teacher The teacher network is obtained by fine-tuning a pre-trained network for the target task. To transfer this knowledge to the student network, the latter is encouraged to match the internal activations of the teacher, by adding the L_2 regularizer, $\Omega = \sum_l \|\mathbf{x}_l^t - \mathbf{x}_l\|^2$, where $\mathbf{x}_l^t$ is the activation of l^{th} block of the teacher network, $\mathbf{x}_l$ the corresponding activation of the student network given by (1), and the sum is carried over all internal blocks as well as network outputs (prior to the softmax).

3 Evaluation

To study the effectiveness of NETTAILOR, we tuned the ResNet34 architecture to three classification datasets of varying characteristics (SVHN, VGG-Flowers and Pascal VOC 2012) and measured the maximum achievable reduction in network complexity that retains performance similar to fine-tuning. Global blocks are obtained by pre-training ResNet34 on ImageNet and remain unchanged afterward in order to share them across tasks. The student is assembled by augmenting the pre-trained blocks with three skip connections per layer. The complexity $\mathbb{C}_i^j$ is defined as the number of FLOPs of each block. After training the student network, all proxies with $\alpha_i^j < 0.05$ are removed andremaining task-specific parameters finetuned without complexity constraints.

Fig. 1 shows the results. We list the total number of layers, parameters (global and task-specific) and FLOPs removed from the pre-trained network by NETTAILOR. We also display the final learned architecture for each task. Fig. 1 shows that networks trained for simpler tasks, such as SVHN, are the most heavily pruned, with 9 out of 18 pre-trained blocks removed. This results in a drastic 73.4 % reduction in total parameters and a 45.8 % reduction in FLOPs. For simpler tasks, most residual blocks are unnecessary and fine-tuning likely converts them into transformations close to the identity, which can be replaced by low complexity proxies. NETTAILOR also obtains significant reductions for the more complex Flowers and VOC datasets. Overall, the results of Fig. 1 show that, for many applications, large pre-trained networks can be significantly reduced, both in size and speed, without loss of performance. Furthermore, because the pre-trained blocks remain unchanged, only a small number of new parameters is introduced per task: 1.90M (million) for VOC, 1.88M for flowers and 1.85M for SVHN (pre-trained ResNet34 blocks have a total of 21.29M parameters).

Full technical details and additional experiments are shown in the full version of this abstract paper (1).

References

[1] Pedro Morgado and Nuno Vasconcelos. Nettailor: Tuning the architecture, not just the weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3044–3054, 2019. 2

[2] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1

[3] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2(6), 2017. 2