GroSS: Group-Size Series Decomposition for Grouped Architecture Search

A Appendix

A.1 Network Definitions

In this section, we will provide the explicit definitions of the networks used for our experiments. In Table 1, we detail the structure of our 4-layer network. Table 2 lists the non-standard classifier structure used for VGG-16 on CIFAR-10. All convolutional and fully-connected layers are followed by a ReLU non-linearity, with the exception of the final fully-connected layers in the classifiers.

Table 1: Architecture of our 4-layer network. Each convolution has a 3×3 kernel and is followed by a ReLU non-linearity and a 2×2 max pooling layer

Conv 1	**Conv 2**	**Conv 3**	**Conv 4**	**Classifier**
conv($3 \rightarrow 32$)	conv($32 \rightarrow 32$)	conv($32 \rightarrow 64$)	conv($64 \rightarrow 64$)	fc(256→256) fc(256→10)

Table 2: VGG-16 classifier structure for CIFAR-10 and ImageNet

Dataset	**Layers**
CIFAR-10	fc(512→512) fc(512→10)
ImageNet	fc(25088→4096) fc(4096→4096) fc(4096→1000)

A.2 Training From Scratch

4-layer Network. Convolutional weights in the network are initialised with the He initialisation [1] in the "fan out" mode with a ReLU non-linearity. The weights of the fully-connected layers are initialised with a zero-mean, 0.01-variance normal distribution. All bias terms in the network are initialised to 0. The network is trained from scratch on our CIFAR-10 training split for 100 epochs using stochastic gradient descent (SGD). We adopt a initial learning rate of 0.1 and momentum of 0.9. The learning rate is decayed by a factor of 0.1 after 50 and 75 epochs. We train the network 5 times and use the weights with median accuracy for further experiments.

VGG-16. For our CIFAR-10 variant of VGG-16, the weights are initialised with identical strategy to the 4-layer network. We train this full network on CIFAR-10 for a total of 200 epochs, again using stochastic gradient descent. The initial learning rate is set to 0.05 and momentum to 0.9. The learning rate is decayed by a factor of 0.1 after 100 and 150 epochs. For ImageNet, we take the pretrained model from the Pytorch (Torchvision) [2] model zoo. Specifically, we take the variant without batch-normalisation layers.

ResNet-18. We again make use of the Torchvision model zoo, and use their ResNet-18 model trained on ImageNet.

A.3 High-Compression Decomposition Structure

We recreate the exact structure used for VGG-16 acceleration in [3] with GroSS, which is listed in 3. The constraint that is used in our other experiments, where bottlenecks should be constant width, is relaxed. Since the group-size must be a factor of both bottleneck dimensions (in and out), the bottleneck dimensions chosen by Wang *et al.*do limit the choice of ranks in GroSS. We perform decomposition as with our other VGG-16 experiments, without any bells or whistles. We found that a longer fine-tuning schedule was required to best recover accuracy. Therefore, finetuning consists of 14 epochs with a learning rate of 5×10^{-4} and decay after 8 and 12 epochs. This led to an accuracy consistent with [3].

Table 3: Decomposition structure as used in [3]. Group-sizes marked with * represent the original choice

Layer	Bottleneck (in → out)	Group-sizes
`conv1_2`	11 → 18	11*
`conv2_1`	10 → 24	5 10*
`conv2_2`	28 → 28	1 7 14* 28
`conv3_1`	36 → 48	3 9* 18
`conv3_2`	60 → 48	15* 30 60
`conv3_3`	64 → 56	16* 32 64
`conv4_1`	64 → 100	16* 32 64
`conv4_2`	116 → 100	29* 58 116
`conv4_3`	132 → 132	3 33* 66
`conv5_1`	224 → 224	7 28 56* 112
`conv5_2`	224 → 224	7 28 56* 112
`conv5_3`	224 → 224	7 28 56* 112

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034 (2015)
2. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
3. Wang, P., Hu, Q., Fang, Z., Zhao, C., Cheng, J.: DeepSearch: A fast image search framework for mobile devices. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **14**(1), 6 (2018)