Supplementary Material: Collaborative Channel Pruning for Deep Networks

In this appendix, we first derive approximation of Hessian matrix for other losses. Then we give the realistic speedup of ResNet-50 on ILSVRC-12, and the results of MobileNet-v1 and Inception-v1 on CIFAR-10. Finally we provide the details of auxiliary classifier.

A. Approximated Hessian Matrix

Let $f(\mathbf{w},\mathbf{x}) \in \mathbb{R}^p$ be the output of $\mathbf{x}$, and $\mathbf{y}$ denotes the corresponding ground-truth label vector. We represent the loss function as $\mathcal{L}$. Here we consider the general formulation to estimate Hessian matrix. First, we give the gradient $\nabla\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y})$ as

$$\begin{aligned}\nabla\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y}) &= \sum_i \frac{\partial\mathcal{L}}{\partial f_i}\frac{\partial f_i}{\partial \mathbf{w}} \\ &= \nabla^T f(\mathbf{w},\mathbf{x})\frac{\partial\mathcal{L}}{\partial f}.\end{aligned} \tag{1}$$

where $\nabla f(\mathbf{w},\mathbf{x}) \in \mathbb{R}^{p\times d}$ and $\mathbf{w} \in \mathbb{R}^d$. The second-order derivative is given by:

$$\begin{aligned}\nabla^2\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y}) = &\sum_i \frac{\partial\mathcal{L}}{\partial f_i}\nabla^2 f_i(\mathbf{w},\mathbf{x}) \\ &+ \nabla^T f(\mathbf{w},\mathbf{x})\frac{\partial^2\mathcal{L}}{\partial^2 f}\nabla f(\mathbf{w},\mathbf{x}).\end{aligned} \tag{2}$$

where $\nabla^2 f_i(\mathbf{w},\mathbf{x}) \in \mathbb{R}^{d\times d}$. For deep networks, $\nabla^2 f_i(\mathbf{w},\mathbf{x})$ often has large computational complexity and is intractable to compute. Here we omit it and use the second term in (2) to approximate $\nabla^2\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y})$, and the final approximated Hessian matrix is given by:

$$\nabla^2\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y}) \approx \nabla^T f(\mathbf{w},\mathbf{x})\frac{\partial^2\mathcal{L}}{\partial^2 f}\nabla f(\mathbf{w},\mathbf{x}). \tag{3}$$

For the least-square loss, since $\frac{\partial^2\mathcal{L}}{\partial^2 f} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix, we approximately compute the Hessian matrix as $\nabla^2\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y}) \approx \nabla^T f(\mathbf{w},\mathbf{x})\nabla f(\mathbf{w},\mathbf{x})$. For the cross-entropy loss, the Hessian matrix can be approximated via:

$$\nabla^2\mathcal{L}(f(\mathbf{w},\mathbf{x}),\mathbf{y}) \approx \nabla^T f(\mathbf{w},\mathbf{x})\Sigma\nabla f(\mathbf{w},\mathbf{x}). \tag{4}$$

where $\Sigma = diag((\mathbf{y} \oslash (f(\mathbf{w},\mathbf{x}) \odot f(\mathbf{w},\mathbf{x}))))$, $\odot$ stands for element-wise multiplication, $\oslash$ denotes element-wise division, and $diag$ denotes converting the vector into diagonal matrix whose entries along diagonal are the entries of vector $(\mathbf{y} \oslash (f(\mathbf{w},\mathbf{x}) \odot f(\mathbf{w},\mathbf{x})))$.

B. Latency of ResNet-50 on ILSVRC-12

To evaluate the realistic speedup, we provide the latency of ResNet-50 on ILSVRC-12 before and after channel pruning. When the pruning ratio $r = 0.40$ for all the layers, the evaluation time is reduced from 146ms to 125ms, and we test the latency on a Nvidia P40 GPU with batch size 64. Note the latency depends on I/O operations, buffer switches and efficiency of linear algebra libraries.

C. MobileNet-v1 and Inception-v1 on CIFAR-10

In this section, we provide the results of Inception-v1 and MobileNet-v1 on CIFAR-10. The baseline classification accuracy of pre-trained MobileNet-v1 and Inception-v1 on CIFAR-10 are 95.43% and 93.71%, we note that accuracy of uncompressed MobileNet-v1 varies in different papers. In (Zhuang et al., 2018), the baseline accuracy is 93.96%, while in (Kim et al., 2019), the baseline accuracy is 89.97%. we test our algorithm with pruning ratio $r = 0.30$. After pruning 30% channels of each layer, the accuracy of Inception-v1 model decreases to 94.54% and the accuracy of MobileNet-v1 model decreases to 92.41%.

D. Auxiliary Classifier

The auxiliary classifier helps boost the performance, its computational cost is quite small. We utilize the average pooling operation over the feature maps to reduce the size to 1×1 in width and height, then we impose a fully-connected layer to project the features to auxiliary classifier. We use cross-entropy loss as auxiliary classifier.

References

Kim, D. H., Lee, S. H., Lee, M. K., and Song, B. C. Macro unit-based convolutional neural network for very light-weight deep learning. *Image and Vision Computing*, 2019.

Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Cao, J., Wu, Q., Huang, J., and Zhu, J. Discrimination-aware channel pruning for deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 883–894. Curran Associates, Inc., 2018.