Appendix

BMVC 2018 Submission # 870

1 L_1 norm vs. L_2 norm

We compare the L_1 and L_2 norm as importance criterion in SPP. Intuitively, it is better to make each weight in a column contribute equally to the importance criteria. However, L_m norm tends to emphasize more the weights with greater magnitude when m is larger (to the extreme, when m is infinity, L_m norm equals to the maximum absolute value of the weights in a column). From this point of view, L_1 norm makes more sense than L_2 norm as importance criteria. The comparison with ConvNet on CIFAR-10 is listed as Tab.1.

Method	2×	4×	6×	8×	10×
SPP (L_1)	**81.7**	81.2	80.3	**80.0**	**79.1**
SPP (L_2)	81.5	**81.3**	**80.4**	79.4	78.5

Table 1: Comparison of the criteria of L_1 norm and L_2 norm with ConvNet on CIFAR-10. The baseline accuracy is 81.5%.

From the result, we do not see significant difference between the L_1 and L_2 norm, which was also reported by other pruning methods [1, 2]. When the speedup ratio is greater, *e.g.* 8× and 10×, L_1 norm is slightly better than L_2 norm.

2 Sensitivity Analysis of Hyper-parmaters

In SPP, the default pruning interval t is 180 iterations, and we keep it for all our experiments. Here we compare the influence of different pruning intervals to see the sensitivity of SPP's performance with the choice of t. We vary t within 50% deviation from the default value 180, i.e. from 90 to 270, at a step of 30.

Pruning interval t	90	120	150	180	210	240	270
Pruning iteration (k)	**6.7**	8.8	11.1	13.3	15.3	17.8	20.0
Accuracy (%)	81.2	81.3	81.3	81.2	81.3	81.2	**81.4**

Table 2: Comparison of different pruning intervals of SPP with ConvNet on CIFAR-10, under speedup ratio of 4×. The baseline accuracy is 81.5%. Pruning iteration is the number of iterations needed to achieve target pruning ratio. $t = 180$ is the default setting.

From the result (Tab. 2), there is no significant accuracy difference between large intervals and small intervals. So the result is not very sensitive to the hyper-parmater t, namely,

A	0.01	0.025	0.05	0.075	0.1
Pruning iteration (k)	6.6	2.6	1.3	0.9	**0.7**
Accuracy (%)	81.2	**81.3**	81.2	**81.3**	81.2
u	0.05	0.125	0.25	0.375	0.5
Pruning iteration (k)	2.4	1.7	1.3	1.1	**1.0**
Accuracy (%)	81.3	81.2	81.2	**81.4**	81.1

Table 3: The comparison of different A and u of SPP with ConvNet on CIFAR-10, under speedup ratio of $4\times$. Default settings are $A = 0.05$ and $u = 0.25$.

there is no need for elaborate hyper-paramter tuning in SPP to achieve comparable results. Meanwhile, we note that, the number of pruning iterations increases linearly with t. For example, the number of pruning iteration with $t = 270$ is about three times as large as that with $t = 90$, while the accuray is almost the same. So with similar accuracy, it's better to choose relatively small interval t for saving time.

The robustness of A and u is shown in Tab.3. Results indicate that performance is not sensitive to reasonable A and u changes. In addition, *all* our experiments (with different network architectures on different datasets) are conducted with the *same* settings of A, u, t, which also shows the robustness of these hyper-parameters.

The robustness of the hyper-parameters, we think, should be owed to the use of the ranks of L_1 norms rather than L_1 norms themselves, because ranks play a role like smoothing, and its value is independent of network architectures or datasets, therefore can coordinate the whole pruning process. An illustration of this coordination is shown in Fig.1, where we compared SPP using ranks and SPP using the L_1 norm as criterion. It is observed that the layer sparsity during pruning of 'SPP+rank' (dot marker) grows much more stable than that of 'SPP+L1' (diamond marker) and with less variations across layers.

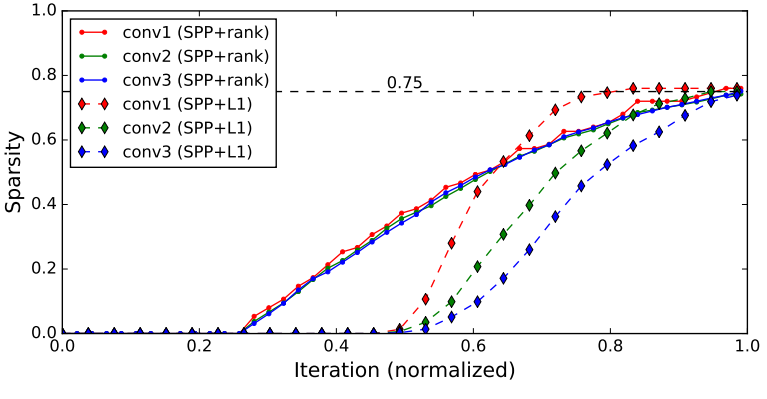

Figure 1: The layer sparsity changes with iterations, where sparsity refers to the ratio of columns whose corresponding pruning probability $p = 1$. Iteration numbers are normalized because different methods have different iteration numbers. Experiment is conducted with ConvNet on CIFAR-10 with target pruning ratio $r = 0.75$. The final test accuracy is 81.2% for 'SPP+rank' and 80.7% for 'SPP+L_1'.

References

[1] S. Han, J. Pool, and J. Tran. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems, NIPS*, pages 1135–1143, Montréal, Canada, 2015.

[2] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations, ICLR*, 2017.