Appendix for Power-law in Sparsified Deep Neural Networks

Anonymous Author(s)
Affiliation
Address
email

A Procedure for TPL Fitting

We follow the method in [1] to estimate $x_{\min}$, $x_{\max}$ and α in (4). Specifically, $x_{\min}$ and $x_{\max}$ are chosen by minimizing the difference between the probability distribution of the observed data and the best-fit power-law model as measured by the Kolmogorov-Smirnov (KS) statistic [2]:

$$D = \max_{x \in \{x_{\min}, x_{\min}+1, \dots, x_{\max}-1, x_{\max}\}} |S(x) - P(x)|,$$

where $S(x)$ and $P(x)$ are the CCDF's of the observed data and fitted power-law model for $x \in \{x_{\min}, x_{\min}+1, \dots, x_{\max}-1, x_{\max}\}$. To reduce the search space, we search $x_{\min}$ in the $\lfloor (n \times k\%) \rfloor$ smallest degree values, where n is the number of nodes in that layer, and $x_{\max}$ in the $\lfloor (n \times k\%) \rfloor$ largest degree values, respectively. In the experiments, $k = 30$ is used. For each $(x_{\min}, x_{\max})$ pair, we estimate α using the method of maximum likelihood as in [1].

B Degree Calculation in Convolutional Neural Networks

As an example, consider a feature map (node) in the first convolutional layer (conv1) for CNN on MNIST in Section 3. As the input image is of size 28×28, each such feature map is of size 24×24. To illustrate the counting more easily, we consider the unpruned network. We first count its connections to the input layer. Recall that in conv1, (i) the filter size is 5×5; (ii) each filter weight is used $24 \times 24 = 576$ times; and (iii) there is only one channel in the grayscale MNIST image. Thus, each node has $25 \times 576 \times 1 = 14,400$ connections to the input layer. Similarly, for connections to the conv2 layer, (i) the conv2 filter size is 5×5; (ii) each filter weight is used $8 \times 8 = 64$ times (size of each conv2 feature map); and (iii) there are 32 feature maps in conv2. Thus, each conv1 node has $25 \times 64 \times 32 = 51,200$ connections to the conv2 layer. Hence, the degree of each conv1 node (in an unpruned network) is $14,400 + 51,200 = 65,600$. Note that the pooling layers do not have learnable connections. They are never sparsified, and we do not need to study their degree distributions.

C Proofs

C.1 Proposition 4.1

Proof 1 *For node i at layer l, let $d_i(t)$ be its degree at time t. Out of these $d_i(t)$ connections, let $d_i^{\uparrow}(t)$ be connected to the upper layer, and $d_i^{\downarrow}(t)$ to the lower layer*[1]*. Using (6), the increase of its degree due to new connections to the upper layer is:*

$$\frac{\mathrm{d}d_i^{\uparrow}(t)}{\mathrm{d}t} = \sum_m \Delta_t^l(d_i(t), d_m(t)) = N^l a^l d_i(t) \frac{\sum_m d_m(t)}{(\sum_s d_s(t))(\sum_m d_m(t))} = \frac{N^l a^l d_i(t)}{\sum_s d_s(t)},$$

[1]For simplicity, we only consider hidden layers here. Analysis for the other layers can be easily modified and are not detailed here.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

where m and s are indices to all the nodes in layer $(l+1)$ and layer l, respectively. Similarly, the increase of degree due to new connections to the lower layer is:

$$\frac{\mathrm{d}d_i^{\downarrow}(t)}{\mathrm{d}t} = \sum_r \Delta_t^{l-1}(d_r(t), d_i(t)) = \frac{N^{l-1}a^{l-1}d_i(t)}{\sum_s d_s(t)},$$

where r and s are indices to all the nodes in layer $(l-1)$ and layer l, respectively. The total number of new connections for nodes in layer l can be obtained as:

$$\sum_s d_s(t) = \sum_s d_s(0) + \int_0^t (N^l a^l + N^{l-1}a^{l-1})dt = \sum_s d_s(0) + (N^l a^l + N^{l-1}a^{l-1})t.$$

Combining all these, we have

$$\frac{\mathrm{d}d_i(t)}{\mathrm{d}t} = \frac{\mathrm{d}d_i^{\uparrow}(t)}{\mathrm{d}t} + \frac{\mathrm{d}d_i^{\downarrow}(t)}{\mathrm{d}t} = \frac{(N^l a^l + N^{l-1}a^{l-1})d_i(t)}{\sum_s d_s(0) + (N^l a^l + N^{l-1}a^{l-1})t}.$$

After integration and simplification, we obtain

$$d_i(t) = d_i(0)c^l(t). \tag{1}$$

C.2 Corollary 1

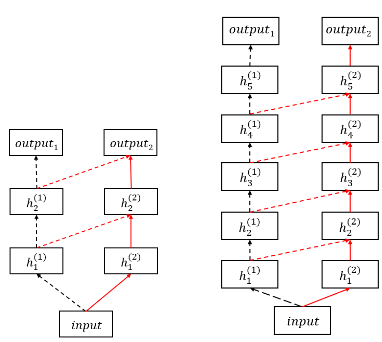
Proof 2 *If the degree distribution of layer l at $t=0$ (denoted p_0^l) follows the power law (standard or TPL), i.e., $p_0^l(d) = Ad^{-\alpha}$ for some $A>0$ and $\alpha>1$. Then, from (1), its degree distribution at time t is*

$$p_t^l(d) = p_0^l\left(\frac{d}{c^l(t)}\right) = A\left(\frac{d}{c^l(t)}\right)^{-\alpha} = (c^l(t))^{\alpha}Ad^{-\alpha}, \tag{2}$$

which also follows the same power law as p_0^l, but scaled by the factor $(c^l(t))^{\alpha}$.

D Progressive Neural Network

In a progressive neural network, we first train a column for task A, using the method in [3]. connections of this column are then fixed. a new column is instantiated for task B. let $h_l^{(1)}$ and $h_l^{(2)}$ be the lth hidden layer of the first and second columns, respectively. layer $h_l^{(2)}$ receives input from both $h_{l-1}^{(2)}$ and $h_{l-1}^{(1)}$ via lateral connections. The progressive neural networks for MNIST and CIFAR are shown in Figure 1.



(a) MLP on MNIST. (b) CNN on CIFAR.

Figure 1: Progressive network. In each model, the left column (with black dashed arrows for its connections) is for task A, while the right column is for task B (with red solid lines for connections within the task B column, and red dashed lines for lateral connections connecting to the task A column).

References

[1] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[2] Anna Deluca and Álvaro Corral. Fitting and goodness-of-fit test of non-truncated and truncated power-law distributions. *Acta Geophysica*, 61(6):1351–1394, 2013.

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