Channel Pruning Algorithm DNS-CP

1 Introduction

Based on the improved channel clipping algorithm of the DNS algorithm, the number of input channels per layer is first reduced, and then fine-tuned on the tailored network to reduce the amount of computation while maintaining high precision.

2 Algorithm Principle

2.1 Weight Update

$$\boldsymbol{W}_k^{(i,j)} = \boldsymbol{W}_k^{(i,j)} - \beta \frac{\partial}{\partial \left(\boldsymbol{W}_k^{(i,j)} \boldsymbol{T}_k^{(i,j)} \right)} L(\boldsymbol{W}_k \odot \boldsymbol{T}_k), \quad \forall (i,j) \in I \tag{1}$$

Where $W_k^{(i,j)}$ represents the weight coefficient of the (i,j) angle in the kth layer of the neural network; $T_k^{(i,j)}$ represents the angle in the kth layer of the neural network as (i,j) weight binary mask, ie mask blob, its value is 0 or 1, 0 means its corresponding weight is deleted, 1 means its corresponding weight is retained, T_k is the same size as W_k; β represents the positive learning rate; $L(\cdot)$ represents the loss function; $\odot$ represents the Hadamard product operator; I represents the angular range of the weight coefficient matrix W_k.

2.2 Update Formula for Binary Mask Matrix $\boldsymbol{T_k}$ (mask blob)

$\boldsymbol{T_k}$ is updated according to a certain probability. When $\sigma(iter) > r$, then $\boldsymbol{T}_k$ is updated; when $\sigma(iter) < r$, it is not updated, r is between [0, 1] Random number.The expression of the probability function is as follows:

$$\sigma(iter) = \frac{1}{(1+\gamma * iter)^{power}} \tag{2}$$

Where, iter is the number of steps in the current iteration, γ and $power$ are hyperparameters, which need to be defined by the user, usually a real number greater than 0.

$$h_k\left(\boldsymbol{W}_k^{(i,j)}\right)\begin{cases}0 & if\ a_k > |\mu_k| \\ \boldsymbol{T}_k^{(i,j)} & if\ a_k \le |\mu_k| \le b_k \\ 1 & if\ b_k < |\mu_k|\end{cases} \tag{3}$$

Where $a_k < b_k$ are respectively the boundaries for determining whether the binary mask is updated. The function $h_k(\cdot)$ indicates that if the absolute value of the weight μ_k is smaller than a_k, the binary mask $\boldsymbol{T}_k^{(i,j)}$ becomes 0, meaning that $\boldsymbol{W}_k^{(i,j)}$ will be cropped. If the absolute value of μ_k is greater than b_k, the binary mask $\boldsymbol{T}_k^{(i,j)}$ becomes 1, meaning that $\boldsymbol{W}_k^{(i,j)}$ will be retained; if μ_k is between a_k and b_k, the value of $\boldsymbol{T}_k^{(i,j)}$is temporarily unchanged, which means that $\boldsymbol{W}_k^{(i,j)}$ is retained depending on $\boldsymbol{T}_k^{(i,j)}$ The value before the update. μ_k is the arithmetic mean of the absolute values of all parameters of the kth channel.

$$\begin{cases}a_k = \max(0, \mu - c\_rate \times std\ ) \\ b_k = \max(0, \mu + c\_rate \times std\ )\end{cases} \tag{4}$$

Among them, μ and std respectively represent the arithmetic mean and standard deviation of the absolute values of all parameters in the current layer, and c_rate is the hyperparameter input by the user, generally taking 0.1, max(·) function returns the maximum value among its parameters.

2.3 DNS Algorithm Flow

Input: **X**: training datum (with or without label), $\widehat{W}_k$: $0 \le k \le C$: the reference model,

α: base learning rate, *f*: learning policy.

Initialize $\boldsymbol{W}_k \leftarrow \widehat{W}_k, \boldsymbol{T}_k \leftarrow 1,\ \forall\, 0 \le k \le C,\ \beta \leftarrow 1$, and $iter \leftarrow 0$.

repeat

Choose a minibatch of network input from **X**

Forward propagation and loss calculation with $(W_0 \odot T_0)$,..., $(W_C \odot T_C)$

Backward propagation of the model output and generate ∇L

for $k = 0, \ldots, C$ do

Update $\boldsymbol{T}_k$ by function $h_k(\cdot)$ and the current $\boldsymbol{W}_k$, with a probability of $\sigma(\text{iter})$

Update $\boldsymbol{W}_k$ by formula (1) and the current loss function gradient ∇L

end for

Update: iter $\leftarrow$ iter + 1 and $\beta \leftarrow f(\alpha;\ \text{iter})$

until iter reaches its desired maximum

Output:$\{\boldsymbol{W}_k; \boldsymbol{T}_k : 0 \le k \le C\}$: the updated parameter matrices and their binary masks.

2.4 Experimental Results：

We test pruned rensnet50 on Imagenet2012 dataset, the results are shown in the following table. When pruned ratio reaches 50%, top1 and top5 increased by 0.13% and 0.17% respectively. When pruned ratio reaches 60%, top1 and top5 decreased by 0.93% and 0.22% respectively.

Table1 Channel prune test

resnet50				
pruned	top1	top5	top1-gap	top5-gap
0	0.727662	0.910144		
0.5	0.728943	0.911824	0.13%↑	0.17%↑
0.6	0.718322	0.907944	0.93%↓	0.22%↓

2.5 Reference

1) Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In NIPS, 2016.