Title: EffConv: Efficient Learning of Kernel Sizes for Convolution Layers of CNNs

JX-Ma

2024/8/17

Astract

This paper design a novel efficient kernel size learning method in which a size predictor model learns to predict optimal kernel sizes for a classifier given a desired number of parameters. The implement can be found in https://github.com/Alii-Ganjj/EffConv .

Introduce

This section introduced how some people designed CNN networks in the past and the work done in this article. such as One famous design mentioned is a multi-layer digital block with a predetermined kernel size. However, doing so would result in the network's predictions being indistinguishable from the kernel size.

The work of this article is as follows:

1. They have introduced a new CNN kernel size learning method that requires much lower computational budget than existing methods.
2. They designed a size predictor model that is trained in conjunction with a kernel predictor model to predict the optimal kernel size for the classifier. Training these models only requires a small portion of the CNN training cycle.
3. Experiments on several benchmark datasets have shown that our method can achieve more proficient architectures in lower training cycles than other kernel size learning methods, providing an efficient and effective solution to the problem of kernel size learning methods.

method

Size Predictor

The model dynamically predicts kernel sizes during training, it takes a fixed input z and predicts the kernel sizes as follows:

$$v = \sigma(f_{sp}(z; \theta sp)) \qquad s = round(v)$$

the f_{sp} represents the size predictor, and round($\cdot$) rounds the input to its nearest integer.   is the nonlinearity that we use to ensure the predicted kernel sizes are in the designed range.

Kernel Predictor

it estimates proper weights for the CNN given the kernel sizes determined by the size predictor.

the formulation is :

$$\breve{w}_l = f^l_{kp}(z^{`}_l; s_l; \theta^l_{kp})$$

the $z^{`}_l$ is the constant input of the GRU.

Modulating Kernel Size in the Predicted Weights

they propose modulating the predicted kernel size into the predicted weights as a workaround.

the formulation is:

$$m_l(i,j) = exp(-\frac{(i-c_l)^2 + (j-c_l)^2}{v_l^2})$$

the m_l is the soft kernel size, the v_l is a mask m_l,

Experiment

This section mainly introduces the dataset, architecture, and some training parameters used in the experiment, and finally introduces the evaluation conditions, it use the models' accuracy/training time per epoch to compare the performance/efficiency of the models, Afterwards, the experimental results on various datasets were introduced

Summary

This article mainly introduces a kernel size learning method that overcomes the computational inefficiency of previous methods while also improving performance. When evaluating performance in this article, precision was included. I regard that in future GPU experiments, precision can also become a criterion for evaluating whether an algorithm is good or not.