Skysong Innovations
Advancing the Arizona State University Knowledge Enterprise

1475 N. Scottsdale Road, Suite 200
Scottsdale, AZ 85287-3538
Phone: 480 884 1996 Fax: 480 884 1984

Case ID:M20-171P^

Published: 3/5/2021

Inventors

Yixing Li

Fengbo Ren

Contact

Shen Yan
shen.yan@skysonginnovations.com

Pruning Binary Neural Networks (BNNs) for Model Compression and Inference Acceleration

Background

In the rapid evolution of deep learning, neural network models have been growing from several to over a hundred layers for handling more complex tasks. Network models are often trained with powerful GPUs in the cloud or on stand-alone servers, and the trained models are then deployed to certain hardware platforms for performing inference. For cloud applications where neural network models are both trained and deployed in the cloud, computational complexity is typically a secondary concern as there is little gap in computing resources between the training and the inference stages. However, in many emerging Internet-of-Things (IoT) applications where neural network models must be deployed onto resource-constrained edge devices for performing real-time inference, the computational complexity of neural network models becomes a major concern. Therefore, it is important to not only investigate how to build more compact neural network models that are friendly to hardware implementations but also rethink how to further compress compact neural network models for efficient hardware implementation on resource-constrained edge devices.

A binary neural network (BNN) is one of the most compact forms of neural networks. Both the weights and activations in BNNs can be binary values, which leads to a significant reduction in both parameter size and computational complexity compared to its full-precision counterpart. Such reductions can directly translate into reduced memory footprint and computation cost in hardware, making BNNs highly suitable for a wide range of hardware accelerators. However, it is unclear if and how a BNN can be further pruned for ultimate compactness. As both 0s and 1s are non-trivial in BNNs, it is not proper to adopt any existing pruning method that interprets 0s as trivial for full-precision networks.

Invention Description

Researchers at Arizona State University have developed a generic method of pruning binary neural networks. This innovation can be generally applied to all existing BNNs to further compress the model and accelerate the inference speed on a wide range of hardware platforms, including CPUs, GPUs, and FPGAs. Specifically, the weight-flipping frequency is used as an indicator to analyze the sensitivity of the binary weights to accuracy. This is based on the observation that when the training is sufficiently close to convergence, the weights with a high weight-flipping frequency are less sensitive to accuracy. Experiments performed on the binary versions of a 9-layer Network-in-Network (NIN) and the AlexNet with the CIFAR-10 dataset demonstrate that the BNN pruning method can achieve a 20-40% reduction in binary operations with a 0.5-1.0% accuracy drop, resulting in a 15-40% runtime speedup on a TitanX GPU.

Potential Applications

- BNNs for Internet-of-Things (IoT) applications
- Accelerating BNN on resource-constrained edge devices
- Accelerating BNN on dedicated hardware accelerators
- Artificial intelligence

Benefits and Advantages

- Generic nature of pruning method preserves flexibility in application
- Significantly reduces binary operations and boosts runtime speeds while imposing minimal sacrifice on accuracy

Related Publication: BNN Pruning: Pruning Binary Neural Network Guided by Weight Flipping Frequency

Research Homepage of Professor Fengbo Ren