Summary

Zhang Heartbright

August 16, 2018

Abstract

Nowadays, convolutional neural networks(CNNs) have achieved great success in many recognition tasks. However, existing deep convolutional neural network models are computationally expensive and memory intensive. Therefore, Model Compression and Acceleration are required. for Deep Neural Networks

Key Words: Pruning , Low-Rank , Compact , Knowledge Distillation

1 Introduction

Applied to different applications and having achieved dramatic accuracy improvements in many tasks, deep neural networks have received a lot of attentions. Yet these works are based on deep neural networks with millions or even billions of parameters, which challenge the hardwares today with the sizeable amount of computation.

As larger networks with more layers and nodes, to reduce the storage and computational costs becomes critical, paticularly in real-time applications such as,online learning and incremental learning

These approaches are classfied into four categories : parameter pruning and sharing, low-rank factorization, transferred/compact convolutional filters, and knowledge distillation. The Table below is the summarize of these methods.

Table
SUMMARIZATION OF DIFFERENT APPROACHES FOR NETWORK COMPRESSION.

Theme Name	Description	Applications	More details
Parameter pruning and sharing	Reducing redundant parameters which are not sensitive to the performance	Convolutional layer and fully connected layer	Robust to various settings, can achieve good performance, can support both train from scratch and pre-trained model
Low-rank factorization	Using matrix/tensor decomposition to estimate the informative parameters	Convolutional layer and fully connected layer	Standardized pipeline, easily to be implemented, can support both train from scratch and pre-trained model
Transfered/compact convolutional filters	Designing special structural convolutional filters to save parameters	Only for convolutional layer	Algorithms are dependent on applications,usually achieve good performance only support train from scratch
Knowledge distillation	Training a compact neural network with distilled knowledge of a large model	Convolutional layer and fully connected layer	Model performances are sensitive to applications and network structure only support train from scratch

2 Main Methods

2.1 Pruning

- Structured Pruning
  - Yoon et al. Combined Group and Exclusive Sparsity for Deep Neural Networks. ICML2017
  - Ren et al. SBNet: Sparse Blocks Network for Fast Inference. CVPR2018
- Filter Pruning
  - Luo et al. Thinet: A filter level pruning method for deep neural network compression. ICCV2017
  - Liu et al., Learning efficient convolutional networks through network slimming. ICCV2017
  - He et al. Channel Pruning for Accelerating Very Deep Neural Networks. ICCV2017
- Gradient Pruning

  - Sun et al. meProp: Sparsified Back Propagation for Accelerated Deep Learning with Reduced Overfitting. ICML2017

- Fine-grained Pruning in a Bayesian View
  - Molchanov et al. Variational Dropout Sparsifies Deep Neural Networks. ICML2017

2.1.1 DrawBacks:

one potential problem of this kind of approaches is that the structural constraint will cause loss in accuracy since the constraint might bring bias to the model. On the other hand, how to find a proper structural matrix is difficult. There is no theoretical way to derive it out.

2.2 Low-rank Factorization

- SVD:
  - Zhang et al., Accelerating Very Deep Convolutional Networks for Classification and Detection. IEEE TPAMI 2016.
- CP decomposition:
  - Lebedev et al., Speeding-up Convolutional Neural Networks Using Fine-tuned CP- Decomposition. ICLR 2015.
- Tucker decomposition:
  - Kim et al., Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications. ICLR 2016.
- Tensor Train Decomposition
  - Novikov et al., Tensorizing Neural Networks. NIPS 2016.
- Block Term Decomposition
  - Wang et al., Accelerating Convolutional Neural Networks for Mobile Applications. ACM MM 2016.
- Tensor Ring (TR) factorizations:
  - Wang et al., Wide Compression: Tensor Ring Nets. CVPR2018
- Block Term Decomposition For RNN
  - Ye et al., Learning Compact Recurrent Neural Networks with Block-Term Tensor Decomposition . CVPR2018.

2.2.1 DrawBacks

low-rank approaches are straightforward for model compression and acceleration. The idea complements recent advances in deep learning, such as dropout, rectified units and maxout. However, the implementation is not that easy since it involves decomposition operation, which is computationally expensive. Another issue is that current methods perform low-rank approximation layer by layer, and thus can not perform global parameter compression, which is important as different layers hold different information. Finally, factorization requires extensive model retraining to achieve convergence when compared to the original model.

2.3 Quantization

- Low-bit Quantization
  - Leng et al. Extremely Low Bit Neural Network: Squeeze the Last Bit Out with ADMM. AAAI2018
  - Hu et al. From Hashing to CNNs: Training Binary Weight Networks via Hashing. AAAI2018
  - Wang et al. A General Two-Step Quantization Approach for Low-bit Neural Networks with High Accuracy. CVPR2018
- Quantization for general training acceleration
  - Kster et al. Flexpoint: An Adaptive Numerical Format for Efficient Training of Deep Neural Networks. NIPS2017
- Gradient Quantization for distributed training
  - Alistarh et al. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. NIPS2017
  - Wen et al. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. NIPS2017

2.3.1 DrawBacks:

Not Found

2.4 Knowledge Distillation

- KD (Knowledge Distillation)
  - Hinton et al. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
- FitNets
  - Romero et al. Fitnets: Hints for thin deep nets.ICLR 2015
- Current Proccess
  - Yim et al. A Gift From Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning . CVPR2017
  - Zagoruyko et al. Pay More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. ICLR2017
  - Chen et al. Learning Efficient Object Detection Models with Knowledge Distillation. NIPS2017

2.4.1 DrawBacks:

KD-based Approaches can make deeper models thinner and help significantly reduce the computational cost. However, there are a few disadvantages. One of them is that KD can only be applied to classification tasks with softmax loss function, which hinders its usage. Another drawback is that the model assumptions sometimes are too strict to make the performance competitive with other type of approaches.

2.5 Compact Network Design

- MobileNet
  - Howard et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CVPR2017
  - Sandler et al. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. CVPR2018
- ShuffleNet
  - Zhang et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. CVPR2018

2.5.1 DrawBacks:

there are few issues to be addressed for approaches that apply transfer information to convolutional filters. First, these methods can achieve competitive performance for wide/flat architectures (like VGGNet) but not narrow/special ones (like GoogleNet, Residual Net). Secondly, the transfer assumptions sometimes are too strong to guide the algorithm, making the results unstable on some datasets.

3 Trend and Future

- Non-fine-tuning or Unsupervised Compression
- Self-adaptive Compression
- Network Acceleration for other tasks
- Hardware-Software Co-design
- Binarized Neural Networks