KnowledgeNet: Disaggregated and Distributed Training and Serving of Deep Neural Networks

Saman Biookaghazadeh, Yitao Chen, Kaiqi Zhao, and Ming Zhao, ***Arizona State University***

https://www.usenix.org/conference/opml19/presentation/biookaghazadeh

This paper is included in the Proceedings of the 2019 USENIX Conference on Operational Machine Learning (OpML '19).

May 20, 2019 • Santa Clara, CA, USA

ISBN 978-1-939133-00-7

Open access to the Proceedings of the 2019 USENIX Conference on Operational Machine Learning is sponsored by USENIX.

KnowledgeNet: Disaggregated and Distributed Training and Serving of Deep Neural Networks

Saman Biookaghazadeh
Arizona State University

Yitao Chen
Arizona State University

Kaiqi Zhao
Arizona State University

Ming Zhao
Arizona State University

Abstract

Deep Neural Networks (DNNs) have a significant impact on numerous applications, such as video processing, virtual/augmented reality, and text processing. The ever-changing environment forces the DNN models to evolve, accordingly. Also, the transition from the cloud-only to edge-cloud paradigm has made the deployment and training of these models challenging. Addressing these challenges requires new methods and systems for continuous training and distribution of these models in a heterogeneous environment.

In this paper, we propose KnowledgeNet (KN), which is a new architectural technique for a simple disaggregation and distribution of the neural networks for both training and serving. Using KN, DNNs can be partitioned into multiple small blocks and be deployed on a distributed set of computational nodes. Also, KN utilizes the knowledge transfer technique to provide small scale models with high accuracy in edge scenarios with limited resources. Preliminary results show that our new method can ensure a state-of-the-art accuracy for a DNN model while being disaggregated among multiple workers. Also, by using knowledge transfer technique, we can compress the model by 62% for deployment, while maintaining the same accuracy.

1 Introduction

Deep Neural Networks (DNNs) have achieved tremendous accuracy improvements for various tasks, such as decision making, text processing, and video processing. Edge and cloud applications are adopting DNNs to assist users and other systems in better decision making. For example, a recent effort [1] is using neural networks on surveillance cameras on the roads to detect objects of interest. However, state-of-the-art DNN models are computationally heavy and need to continuously adopt to the environment.

Some related works have proposed new methods for distribution and acceleration of DNNs on distributed heterogeneous systems, for both training, and inference. One approach is distributed synchronous SGD algorithm [4], where each computing node executes the complete model on different batches of data. This method suffers from lack of scalability in the training process. Also, in the edge-cloud paradigm, the edge nodes are not powerful enough to take care of the whole model. Another method is *model-parallelism* [3, 9]. In this method, different layers of the model are distributed among several accelerators (on the same machine). Need to mention, the feasibility of such a model in a distributed edge-cloud environment with average connections speed is not yet evaluated.

Other related works have studied several methods to prepare DNN models for edge deployment. These methods can be broadly classified into four categories: (1) *Weight Sharing*, (2) *Quantization*, (3) *Pruning*, and (4) *Knowledge Transfer*. The weight sharing techniques [2, 5] reduce the memory occupied by the model by grouping weights and replacing them with a single value. The quantization techniques [5, 7] reduce the size of the model by shrinking the number of bits needed by the weights. The pruning techniques [5, 8, 10] reduce the complexity of a model significantly by removing weights or connections that produce a negligible response. Finally, with the knowledge transfer techniques, a small model is being supervised by a large model during the training to achieve a much higher accuracy. Unfortunately, all these methods (except knowledge transfer) are only feasible for inference scenarios.

2 Approach

Following the previous discussions, we propose KnowledgeNet (KN), which enables a disaggregated and distributed training/serving process while supporting heterogeneous environments, such as the edge-cloud paradigm. KN can disaggregate and deploy large DNN models on a set of heterogeneous processors to enable continuous training based on the user data and ever-changing environment.

The KN utilizes two specific methods to enable disaggregated and distributed model training and serving. First,

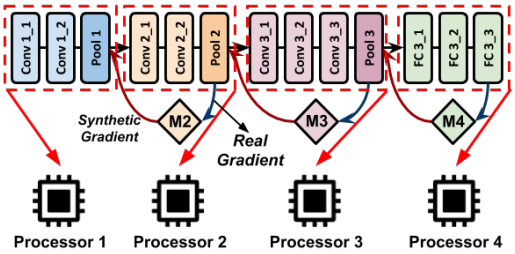

Figure 1: Representation of the model in the KN setting. Each dash box is being mapped onto a distinct processor. Also, the synthetic gradients are being generated asynchronously, using extra components (represented as M blocks).

KN can split a large DNN model into several small models (Figure 1). Each of these small models can be deployed on an independent processing node. The conventional training method suffers from the locking among the layers in a model, based upon their prior or next layer, during the forward and backward propagation. To alleviate this problem, we use the synthetic gradient method [6] to generate the target gradient for each section, asynchronously. Using synthetic gradients, each individual or set of layers can continue their progress, by adding a new module, which is responsible to generate synthetic gradients, approximating the true gradients in the conventional training model. As a result, the training process can be seamlessly offloaded onto a set of heterogeneous process without compromising the accuracy.

Second, while disaggregation can overcome the distribution problem, it may still need compression to be deployed on the edge devices. Edge devices are usually equipped with small processors with limited computational capabilities. We develop a new knowledge transfer (KT) technique in KN, which enables fine-grained supervision from the oracle model on the cloud and the model deployed on edge. Using this technique, the DNN model can be transformed into two equivalent models: (1) A large-scale oracle model on the cloud and (2) a small-scale counterpart model on the edge. Our novel KT technique provides state-of-the-art accuracy for the small-scale model, while receiving supervision from the oracle model. In the KN, unlike conventional KT techniques, where the only knowledge comes from only the final layer's loss, each section of the small model can constantly receive supervision from a specific section of the oracle model, in order to adopt the same representation.

3 Evaluation

In this section we provide preliminary evaluations on the feasibility of the KN, based on the its capability to maintain a state-of-the-art accuracy, while enabling distributed training over a set of heterogeneous devices.

Our experimental result for the synthetic gradient approach can achieve comparable accuracy as the conventional backpropagation approach. We use a simple four-layered model which consists of one convolutional layer (including max pooling layer) and three fully-connected layers with MNIST dataset for the evaluation. After training for 500K iterations, the backpropagation approach achieves 98.4% accuracy whereas the synthetic gradient approach achieves 97.7% accuracy.

Our knowledge transfer result shows that we can compress a model significantly and maintain the same accuracy by leveraging the knowledge from a large network. The teacher model is VGG16 and the student model is a network that is shorter than the teacher and consists of much fewer parameters (3.2M vs. 8.5M). After training for 100 epochs, the accuracy of teacher model and independent student model is 74.12% and 61.24%, respectively. The dependent student model that uses our proposed knowledge transfer method achieves almost comparable accuracy as the teacher model.

4 Conclusions and Future Work

The emerging trend of heterogeneous systems, both in the cloud-only and in the form of edge-cloud systems, necessitates rethinking the current methods for training and deploying deep learning models. Current available methods cannot enable efficient serving and continuous training of the complex models on the distributed heterogeneous systems. KN seeks to address the above challenges, through our novel disaggregation and distribution of the DNNs, and also our new layer-by-layer knowledge transfer techniques. Our preliminary results suggest our new method as a promising approach for training DNNs for the emerging heterogeneouscomputing paradigms.

References

[1] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *computer*, 50(10):58–67, 2017.

[2] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pages 2285–2294, 2015.

[3] Yuanxiang Gao, Li Chen, and Baochun Li. Spotlight: Optimizing device placement for training deep neural networks. In *International Conference on Machine Learning*, pages 1662–1670, 2018.

[4] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[5] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[6] Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1627–1635. JMLR. org, 2017.

[7] Deepak Kadetotad, Sairam Arunachalam, Chaitali Chakrabarti, and Jae-sun Seo. Efficient memory compression in deep neural networks using coarse-grain sparsification for speech applications. In *Proceedings of the 35th International Conference on Computer-Aided Design*, page 78. ACM, 2016.

[8] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[9] Azalia Mirhoseini, Anna Goldie, Hieu Pham, Benoit Steiner, Quoc V. Le, and Jeff Dean. A hierarchical model for device placement. In *International Conference on Learning Representations*, 2018.

[10] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.