Action Exploratoire (AEx)

Project:	LOw-rank Tensor representations of convolUtion neural networkS (LOTUS)
Principal investigator (PI):	Suraj KUMAR
Project team :	ROMA
Centre :	Inria Lyon
Duration :	4 years

Summary

Convolutional neural networks are currently the most popular models to classify objects in several fields. These networks have intense computational requirements due to the large number of parameters. Recent work has demonstrated that replacing some parts of popular models with their low-rank tensor representations drastically improves the number of parameters and achieves similar accuracy. Inspired by this progress, we view full models as large tensors and plan to represent them with their low-rank representations.

1 Project description

Convolutional neural networks (CNNs) are currently the state-of-the-art models to classify objects in several domains, such as computer vision, speech recognition, text processing etc. Thanks to improved computational capability, we witness several popular complex and deeper CNNs. For example, AlexNet is 8 layers deep, while ResNet employs short connections and is represented with 152 layers. Both have about 60M parameters. CNNs have intensive computational requirements due to their huge complexity and large number of parameters.

Tensors are a natural way to represent high dimensional data for numerous applications in computational science and data science. Tensor decompositions help to identify inherent structure of data, achieve data compression and enable various ways of data analysis [2]. CP, Tucker and Tensor Train are the widely used tensor decomposition methods in the literature. These decompositions represent a high dimensional object with a small set of low dimensional objects. These decompositions can be viewed as high order generalization of singular value decomposition. CP decomposition is used to understand the latent components of the data and well suited for tensors with small dimensions. Tucker decomposition is considered to be more appropriate for compression and multi-modal data analysis of small and moderate dimensional tensors. Tensor train decomposition captures the entanglement among dimensions and appropriate to work with high dimensional tensors.

Representing a high dimensional tensor with a set of smaller dimensional objects drastically reduces the overall number of parameters. This led to the use of low-rank tensor representations at different layers of CNNs. For example, it has been shown that replacing convolution kernels of ResNet with their low-rank approximations in Tucker tensor representations significantly reduces the number of parameters and improves the overall performance [5]. In a separate work, contributions have been made to replace dense weight matrices of the fully connected layers of AlexNet by their approximations in Tensor-train format [4]. This approach also significantly reduces the number of parameters while achieving the similar accuracy. Replacing a layer in CNN with its low-rank approximation requires to re-tune the network parameters and both the mentioned works adapted the existing training methods for low-rank tensor representations. The above contributions strongly advocate to employ the low-rank tensor representations in CNNs. We view a full CNN as a large tensor and aim to replace it with a network of smaller tensors. However the training methods for this large tensor and the decomposition structure is not intuitive to estimate. Therefore, we plan to follow bottom-up approach and represent multiple layers of existing successful CNN architectures – AlexNet and ResNet, with their low-rank tensor representations.

1.1 Objectives

The state-of-the-art CNN models are very complex to understand. The high level objective of this project is to express CNN models with simpler tensor based networks. The LOTUS project aims to drastically improve the following features of CNNs.

- *Simplicity and analysis capability* : CNNs represented with different tensor decompositions are simpler. Therefore it will allow one to gain more insights of the networks and describe what happens in each layer. We though do not explicitly focus on this aspect in the LOTUS project, but it is a part of our future work.

- *Reduction of parameters* : Some researchers have replaced certain portions of CNNs with networks of smaller tensors and achieved similar accuracies with significantly less number of parameters, as mentioned earlier. The LOTUS project explores this direction with the aim to represent a full CNN by a network of smaller tensors.

- *Parallel algorithms for training and prediction* : Representing CNNs with tensor based networks will allow one to take advantage of the existing parallel tensor algorithms. Now a days parallelization is ubiquitous in most computing systems. State-of-the-art CNNs usually rely on the efficient parallel implementation of matrix multiplication for parallelization. This approach processes only one layer at a time in backward or forward propagation. Tensor based networks will enable one to apply different parallelization schemes from Physics and Chemistry for training and inference.

1.2 Methodology

As mentioned earlier, this project aims to represent a full CNN by a network of smaller tensors and we plan to achieve it in a bottom-up fashion. In the beginning, we plan to focus on AlexNet architecture. It has 5 convolution layers and 3 fully connected layers. We first plan to replace convolution layers with a network of smaller tensors and then replace the fully connected layers with another network of smaller tensors. This will require us to re-tune the parameters of the full network. To do this, we plan to adapt gradient-descent method for training. We also plan to take advantage of the existing training methods for tensor network based frameworks in physic or chemistry. For example, density matrix renormalization group (DMRG) is a popular algorithm in molecular simulation community and it has demonstrated initial success to train neural networks [6]. Once we have a working architecture with low rank representations of two large tensors, then our next step would be to replace both tensors with a single one. Here first we plan to represent the large tensor in one of the popular low-rank tensor representations. Depending on the challenges faced at this level, we also plan to consider a combination of more than one low-rank tensor representation.

We first will work with MNSIT dataset for our training. After that we will also work with CIFAR and ImageNet datasets. After exploring AlexNet architecture, we will focus on ResNet architecture in the next step. In general, ResNet is more complex and consists of 152 convolution layers. Here we plan to replace 4-5 layers at once with a network of smaller tensors and then re-tune the parameters. Based on the outcome of this step, we plan to iterate our approach until we represent the full network with a low-rank representation of a large tensor. We also aim to take advantage of the work on parallel tensor computation and apply in our framework to improve the training/inference time.

1.3 Risks and their impacts

There are two main risks associated with this project. Here we mention how do we plan to mitigate the impact of these risks.

- *Design of new training methods* : As mentioned in the methodology section that we plan to modify the structure of CNNs with low-rank tensor representations and this approach requires to re-tune

the parameters. It requires to design new training methods. We plan to adapt the present popular algorithms for CNNs or tensor based networks. We also need to validate the robustness of the new adapted methods. To achieve this, in the beginning of the project, we plan to work with an intern and analyze what are the possible ways to adapt the popular training methods for tensor based networks. The goal of this step is to decide certain methods and try to obtain theoretical or probabilistic guarantees on the convergence of those methods.

- *Parallelization of the proposed training methods* : The proposed training methods may have limited parallelism. For example, it is well known that the DMRG method that is used in molecular simulation to work with tensor based networks is hard to parallelize. To overcome this limitation, we plan to consider/develop methods that have enough work at each step or can be parallelized in a tree structure.

1.4 Extension of our framework for heterogeneous systems

GPUs deliver increased processing capabilities and superior energy efficiency compared to CPUs. Therefore, they have become a crucial element of many computing systems over the past decade. The traditional computing cores of GPUs provide the accuracy and precision for the mathematical operations. These cores take huge amount of time for deep learning models as they require to process massive datasets. Thus, GPU vendors recently introduced Tensor (or Matrix) cores. These cores can perform multiple operations per cycle at the cost of limited precision, whereas traditional cores perform one operation per cycle with very accurate results. For deep learning models, the newly added Tensor/Matrix cores are much effective in terms of both cost and computation speed. Hence these cores are the preferred choice for CNN models [3]. However such approaches do not make full utilization of other units of a heterogeneous system. We also plan to parallelize our models for a heterogeneous systems composed of CPUs and GPUs. Task based runtime systems are a popular way to make efficient utilization of heterogeneous resources [1]. We intend to express computations of our parallel models in terms of tasks such that we can run them efficiently on a heterogeneous system with popular runtime systems, such as StarPU.

2 Requested resource

The recruitments of this project are a research intern (6 months), a PhD student (36 months) and an engineer (18 months). The research intern is expected to join the project from the beginning and work on the analysis of all the popular training methods for tensor based models. The PhD student is expected to join the project around the 6th month. He/She will focus on replacing full CNNs with networks of smaller tensors and design new robust training methods. The engineer is expected to be hired for 18 month, starting around the 30th month of the project. He/She will extend the proposed algorithms for heterogeneous systems and efficiently implement them.

We ask funding for the research intern and the PhD student through this call. We will look for some other funding sources to hire the engineer. We request 135.8k€ and Table 1 shows the breakdown of the requested amount.

	Amount
Research intern	3.3k €
PhD student	120k €
Inward billing (1 laptop)	2.5k €
Travel costs (missions)	10k €
Requested	135.8k€

Table 1: Requested amount for the project through this call.

3 How is the project exploratory?

In the last decade, CNNs have achieved tremendous success to classify objects in several domains. We view a full CNN as a large tensor and aim to represent it with a network of smaller tensors. This will allow one to take advantage of the work on low-rank compression of tensors and parallelization of tensor operations. Our idea seems promising and requires to explore several underneath issues, such as how to design efficient methods to train tensor based models that are easy to parallelize, what are the good alternatives for nonlinear components of a CNN in tensor format, what low-rank tensor representations to select for a portion of a CNN, and how to combine two different low-rank tensor representations to represent a large portion of a CNN, to name a few. To the best of PI's knowledge, he is not aware of any research group that attempts to represent a full CNN by a network of smaller tensors.

4 Follow-up of this project

As mentioned earlier, we plan to look for other funding sources to hire an engineer who will work on the parallelization of the proposed models. Towards the end of this project, we also plan to submit a proposal whose aim would be to get good interpretation of the proposed models, for example what is the role of each component in the full network. We are also interested to extend our framework for recurrent neural network models in future.

Upon successful completion of the project, we plan to adapt our models to work efficiently on low-end devices. This will enable us to identify objects quickly on any device. We will explore applications of our framework in several domains and aim to create a startup.

References

[1] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier. StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. *Concurrency and Computation: Practice and Experience*, 23(2):187–198, 2011.

[2] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[4] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[5] A.-H. Phan, K. Sobolev, K. Sozykin, D. Ermilov, J. Gusak, P. Tichavský, V. Glukhov, I. Oseledets, and A. Cichocki. Stable low-rank tensor decomposition for compression of convolutional neural network. In *Computer Vision – ECCV 2020*, pages 522–539, 2020.

[6] E. Stoudenmire and D. J. Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.