

# Efficient representations of deep neural networks



Project Manager  
Rahim Mammadli

Principal Investigator  
Prof. Dr. Felix Wolf

Project Term  
2012 - 2012

Clusters  
Lichtenberg Cluster Darmstadt

Software  
PyTorch

Institute  
Parallel Programming

University  
Technische Universität Darmstadt

## Introduction

Deep neural networks (*DNNs*) have gained extreme popularity in recent years, advancing state-of-the-art results in computer vision and natural language processing tasks. Their success can be partially attributed to substantial increase in computational power of parallel processors over the last decade. However best-performing models in terms of accuracy tend to have many thin consecutive processing layers which makes them generalize better but also limits the level of parallelism that can be exploited during inference. In this work our aim was to build a tool that produces neural networks with high accuracy while maintaining efficient use of compute resources.

## Methods

We build a tool named *Shape\_DNN*, which takes user input in form of constraints on size, latency, energy consumption and accuracy of the neural network as well as the optimization goal, which dictates which of the four aforementioned properties of the neural network will be optimized. *Shape\_DNN* tunes two important hyper-parameters of a *DNN*: its depth and width, which are together referred to as the shape of a *DNN*. First, the design space of *DNNs* is initialized. After that, the non-functional constraints, i.e. the size, speed, and energy consumption are processed, removing the non-conforming *DNNs* from the design space. Then, in order to produce the model predicting the accuracy of the neural network, some of the *DNNs* in the design space are trained. Once the accuracy prediction model is built, it is used to process the accuracy constraint and/or the optimization goal provided by the user.

## Results

We compare the performance of Shape\_DNN to exhaustive search in the design space of different shapes of residual networks. Our results show that Shape\_DNN achieves comparable results to exhaustive search, while spending up to 5x less time on the training of DNNs. Moreover, when comparing the speed of the DNNs produced by Shape\_DNN to depth scaled residual networks we notice up to 4.22x faster execution time with the same level of accuracy.

## Discussion

DNN training time is by far the most time-consuming part of our approach. We believe it can be further reduced using various methods in addition to training several shapes in parallel on a multi-server cluster, e.g. by modelling the learning process or using knowledge transfer techniques to avoid training each shape from scratch. We plan to also integrate recently developed DNN architectures such as DenseNet and NASNet into Shape\_DNN. We believe the abovementioned features will make Shape\_DNN more useful for real-life applications.

## Publications

Mammadli, R.; Wolf, F.; Jannesari, A.: The Art of Getting Deep Neural Networks in Shape. ACM Transactions on Architecture and Code Optimization (TACO), 15(4):62:1- 62:21, January 2019.  
<https://doi.org/10.1145/3291053>

*Last Update:* 2022-05-06 17:40