KI DELTA LEARNING
Scalable AI for Automated Driving

Deploying AI models under safety constraints

Domenik Helms, Adrian Osterwind
OFFIS

WP 4.2

Use-Case in Delta Learning

WP4 in total regards the delta between an AI system, running in a lab environment and the same system, running in the real world. While AP4.1 regards the AI's reaction towards unexpected, unpredictable and previously unseen input data, the AP4.2 focusses on the necessary changes, when trying to run an AI on an actual embedded system inside a car: As the performance of an AI system is usually safety critical its execution has to regard certain constraints such as worst case execution time, worst case memory demand or energy dissipation. The goal of AP4.2 is thus to develop tools and methodologies, helping to regard and to optimize for these constraints when deploying an AI application onto an automotive embedded system.

Technical Problem

OFFIS is focusing on two problems in AP4.2: The first one is predicting the to be expected execution time of a given neural network, when being deployed onto and executed on a certain hardware component. The prediction should be done already before the neural network is even trained.
The second one is to develop transformation methodologies, reading in a given trained neural network and converting it into another neural network which is then in accordance to the safety constraints.

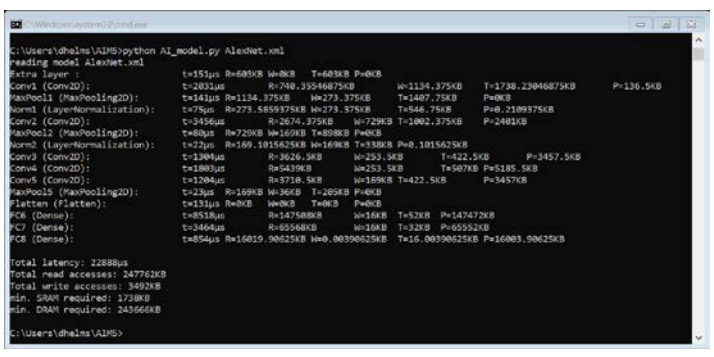

Figure 1: Based on an analysis of the neural network's topology and its configuration (i.e. meta parameters), the execution time of each network layer can be predicted and summed up to an overall prediction.

Technical Solution

In order to determine the execution time of a neural network before training, we developed a methodology, reading in the general network's topology and its meta-parameters. We separate each layer of the neural network and compare its parameter set with earlier measurements of the hardware under analysis. From this we can extrapolate the per layer execution time and sum that up to the overall execution time with high accuracy.
In order to optimize a given network towards the constraints, we developed a configurable tensor compression module, and connected it to a neural architecture search system, which is used to optimize the remaining degrees of freedom of the tensor compression. The entire system enables to trade off accuracy of the network with its execution time and memory footprint.

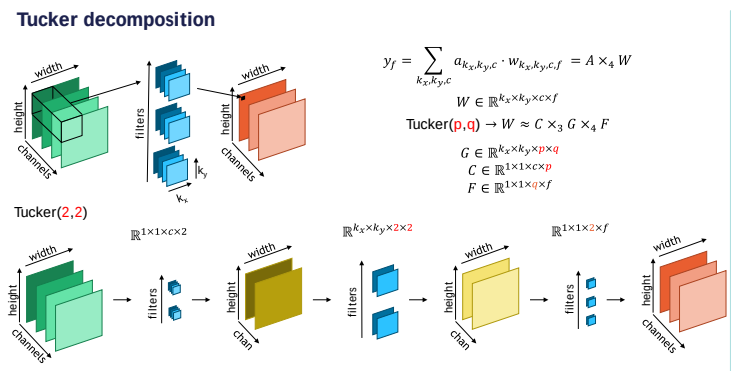

Figure 2: Visualization of the Tucker decomposition, powering the tensor compression: Tucker decomposition produces two small interface tensors and a remaining core tensor. In sum, the tensors are significantly smaller than the original weight tensor.

Evaluation

We applied the model to the convolution layers of several benchmarks. Other layers such as pooling, activations, normalization and dense are not regarded yet. We applied the modelling approach to all convolutional layers in the benchmarks and compared it to a measurement of these layers in the real hardware.
As can be seen in Table 1, for large networks, the prediction accuracy is already quite good.

Benchmark	Mobilenet	Densenet121	ResNet152
Convolution layers	14	120	156
Execution time measured	7641ms	22570ms	107265ms
Time prediction	5629ms	23551ms	106028ms

Table 1: Analysis of the neural network model, so far just working for convolution layers: For large networks, the prediction is already quite good, but for small networks and especially for those, using depthwise convolution, there are still high deviations, making a model update necessary.

Networks

OFFIS Verkehr DLR

For more information contact:
domenik.helms@offis.de

Partners

External partners

KI Delta Learning is a project of the KI Familie. It was initiated and developed by the VDA Leitinitiative autonomous and connected driving and is funded by the Federal Ministry for Economic Affairs and Energy.

Supported by: Federal Ministry for Economic Affairs and Energy
on the basis of a decision by the German Bundestag

www.ki-deltalearning.de @KI_Familie KI Familie