Fast and Efficient DNN Deployment via Deep Gaussian Transfer Learning

2021 ICCV OCTOBER 11-17 VIRTUAL

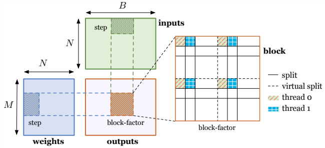
SmartMore

Qi Sun, Chen Bai, Tinghuan Chen, Hao Geng, Xinyun Zhang, Yang Bai, Bei Yu. *CUHK & SmartMore*.

Background & Motivation

- ✦ Heavy communication and computation workloads.
- ✦ Optimizing the model deployment is indispensable.

Preliminaries



Deployment Configuration

All of the settings (*e.g.*, blocks, threads, and *etc.*) to be determined are encoded as a feature vector $\boldsymbol{x}$ which is termed a deployment configuration.

Challenges:

- ✦ Extremely large design space
- ✦ Slow compilation process
- ✦ Underutilized historical information

Problem Formulation

Design Space

For each DNN layer, the design space $\mathcal{D}$ contains all of the candidate configurations.

Optimization Objective

For each layer, find the deployment configuration $x_* \in \mathcal{D}$ which has the best performance.

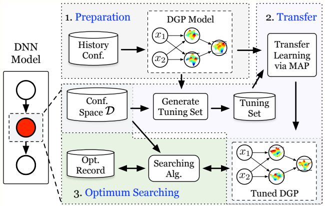
Deep Gaussian Transfer Learning

- ✦ Learn from the historical optimization records.
- ✦ Speedup the searching process.
- ✦ Find better deployment configurations.

Transfer Learning & Deep Gaussian Processes:

- ✦ Layer-wise optimization
- ✦ Stage 1 preparation: learn a deep Gaussian process model from historical data (model pre-training)
- ✦ Stage 2 transfer: transfer knowledge of the DGP model to new tasks (model tuning)
- ✦ Stage 3 optimal searching: guide the optimization of new tasks with the tuned DGP model

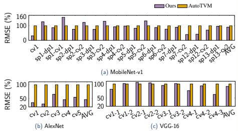
Our Flow:



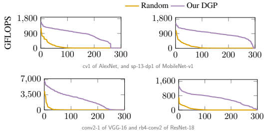
- ✦ Source Task: history tuning data
- ✦ Target Task: new deployment tasks
- ✦ Maximum-a-posteriori (MAP) estimation

Results

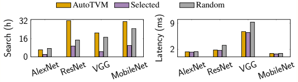
DGP Prediction Errors (Some Examples):



DGP-selected Tuning Set *vs.* Random Samples



Performance with Randomly Sampled Tuning Set:



(a) Search Time (b) Inference Latency

Final Results:

Table 1: Comparisons of Search Time and End-to-end Model Inference Latency

Model	AutoTVM		CHAMELEON			Ours				
	Search (h)	Inference (ms)	Search Redu. (%)	Inference Redu. (%)	HV	Search (h)	Search Redu. (%)	Inference (ms)	Inference Redu. (%)	HV
MobileNet-v1	31.14	0.8980	-	-	-	10.06	67.69	0.7664	**14.65**	**9.9168**
AlexNet	6.28	1.3467	72.16	5.88	4.2409	2.14	65.96	1.2537	**6.91**	**4.5573**
VGG-16	19.92	6.7847	82.56	3.44	2.8418	4.61	76.83	6.4972	**4.24**	**3.2556**
ResNet-18	32.04	1.8248	76.67	4.16	3.1915	9.47	70.43	1.7305	**5.17**	**3.6423**

Our method achieves the best inference performance while accelerating the optimization simultaneously.