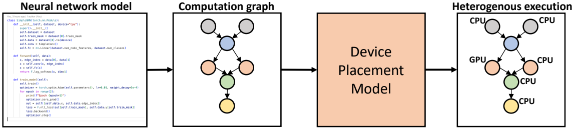
A Structure-Aware Framework for Learning Device Placements on Computation Graphs

Shukai Duan, Heng Ping, **Nikos Kanakaris**, Xiongye Xiao, Panagiotis Kyriakis, Nesreen K. Ahmed, Peiyu Zhang, Guixiang Ma, Mihai Capotă, Shahin Nazarian, Theodore L. Willke, Paul Bogdan

38^{th} Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouver Convention Center, Canada
December 10-15 2024

Background



Computation graphs

- $G = (V, E)$
- labeled, unweighted, directed and acyclic (DAG)
- A node v represents an operation applied to the input data and is associated with an operation type
- An edge $e = (v, u)$ represents the flow of data or dependency among node v and node u

Device placements

Given a list $\mathcal{D}$ of the available devices, a placement $P = \{p_1, p_2, ..., p_n\}$ assigns each operation v of a computation graph G to a device $p \in \mathcal{D}$, where $p \in \{1, 2, ..., |\mathcal{D}|\}$.

Problem definition

Our goal is to assign each part of a computation graph to the most suitable device, such that the overall execution time during the inference of the model is minimized.

$$\theta^* = \arg\min_{\pi,\theta} l(G; \pi, \theta)$$

Related work

Problems of the existing approaches

- Not capturing the directed interactions among nodes
- Heuristics or simple methods for graph partitioning
- Requiring hyperparameter tuning
- Grouper- or encoder-placer architectures
- End-to-end training is not allowed
- Ignoring topological features

Related work

Problems of the existing approaches

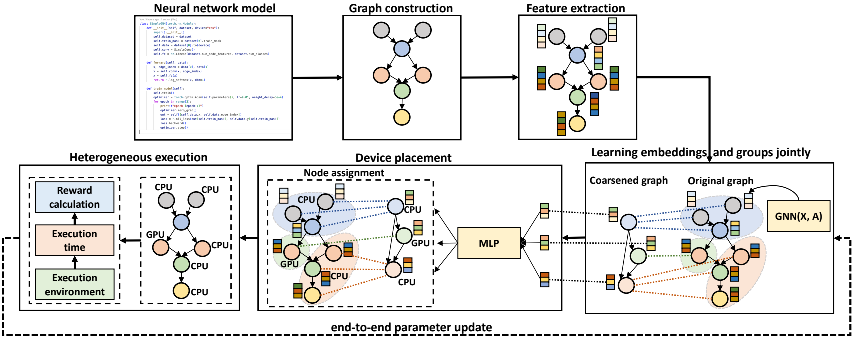
- Not capturing the directed interactions among nodes
- Heuristics or simple methods for graph partitioning
- Requiring hyperparameter tuning
- Grouper- or encoder-placer architectures
- End-to-end training is not allowed
- Ignoring topological features

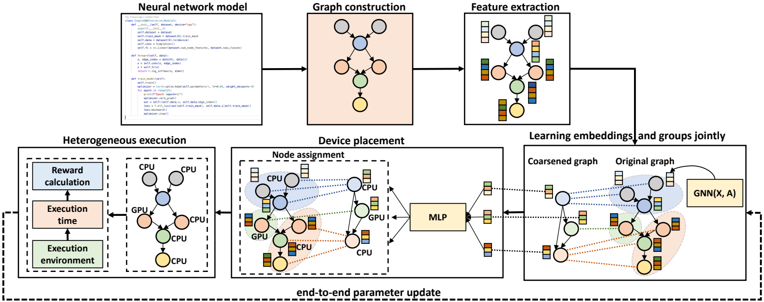
Our approach

- Local and global structural features
- Learning how to partition a graph
- Unspecified number of groups
- End-to-end learnable parameters
- Personalized partitioning
- Fusing encoder- and grouper-placer techniques

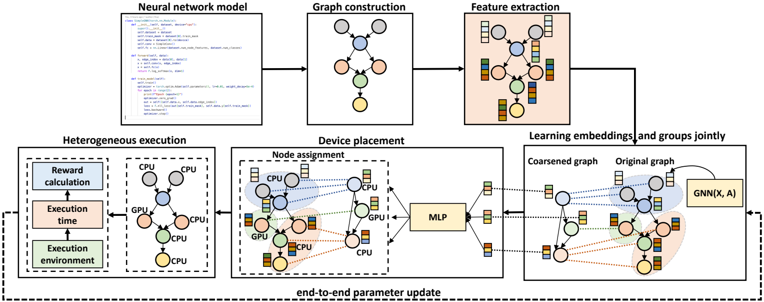
Proposed framework

The architecture





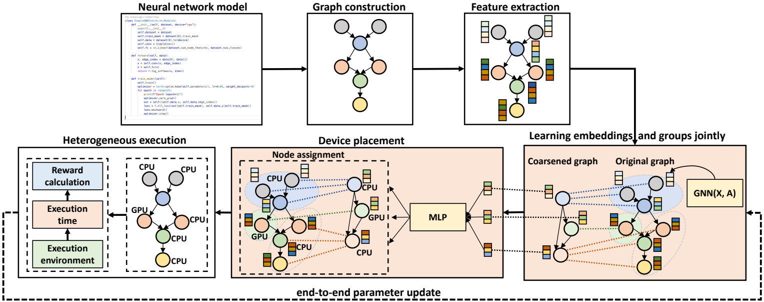
- Each computation graph is:
  - labeled
  - unweighted
  - directed and acyclic (DAG)
- Each node:
  - corresponds to an operation
  - has an associated operation type
- Each edge:
  - links two nodes
  - represents the flow of data
  - or a dependency among two operations



- Four categories of features:
  - Local structural features
  - Global structural features
  - Positional features
  - Node-specific features
- Examples of features:
  - in-degree and out-degree
  - operation type embedding
  - fractal dimension of nodes
  - positional encoding
  - node id or node embedding

Proposed framework

Learning embeddings and groups jointly and device placement

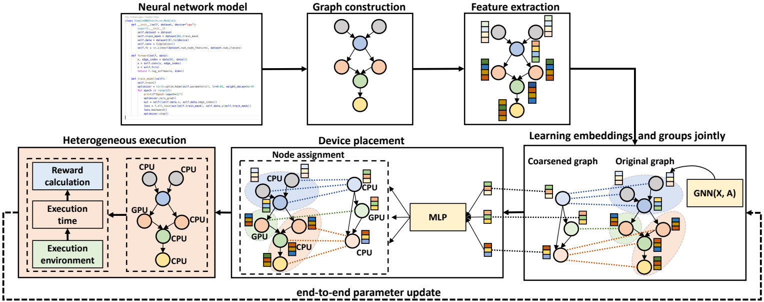


- Learns embeddings and groups jointly
- Further enrich node features
- Partitions a computation graph
- Unspecified number of groups
- Grouper-placer and encoder-placer

- Graph parsing network
  - Graph and node encoding
  - Edge score matrix calculation
  - Graph partitioning and pooling
- Original nodes to the available devices

Proposed framework

Heterogeneous execution



- Intel Server
- Intel OpenVINO toolkit
- Reinforcement learning
- Policy learning
- Inference time

- REINFORCE
- Reward aware of execution time
- $r_{P'}(G') = \frac{1}{l_{P'}(G')}$
- End-to-end parameter update

	Inception-V3		ResNet		BERT	
	$l_P(G)$	Speedup %	$l_P(G)$	Speedup %	$l_P(G)$	Speedup %
CPU-only	0.0128	0	0.0160	0	0.00638	0
GPU-only	0.0120	6.25	0.00781	51.2	0.00277	56.5
OpenVINO-CPU	0.0128	0	0.0234	-46.3	0.00657	-2.98
OpenVINO-GPU	0.0138	-7.81	0.00876	45.3	0.00284	55.5
Placeto	0.0116	9.38	0.00932	41.8	0.00651	-2.04
RNN-based	0.0128	0	0.00875	45.3	OOM	OOM
HSDAG	**0.0105**	**17.9**	**0.00766**	**52.1**	**0.00267**	**58.2**

Experiments

Ablation study

	Inception-V3		ResNet		BERT	
	$l_P(G)$	Speedup %	$l_P(G)$	Speedup %	$l_P(G)$	Speedup %
CPU-only	0.0128	0	0.0160	0	0.00638	0
Original	0.0105	**17.9**	0.00766	**52.1**	0.00267	**58.2**
w/o output shape	0.0117	8.59	0.00768	52.0	0.00278	56.4
w/o node ID	0.0117	8.59	0.00768	52.0	0.00279	56.4
w/o graph structural features	0.0109	14.8	0.00766	52.1	0.00268	58.2

That's all!

Code:

Paper: