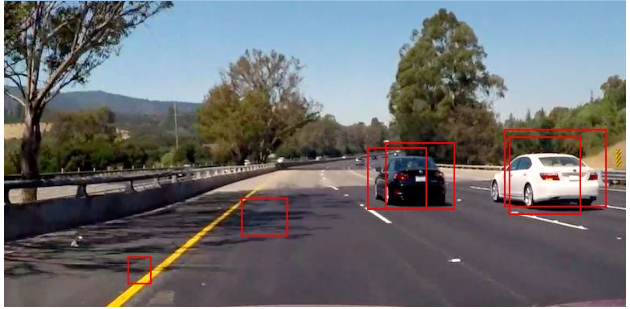
Topology-Aware Network Pruning using Multi-stage Graph Embedding and Reinforcement Learning

Sixing Yu[1], Arya Mazaheri[2], Ali Jannesari[1]
[1]Iowa State University, [2]Technical University of Darmstadt

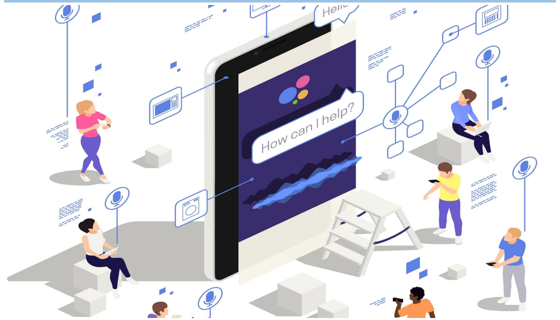
Modern AI model applications

Machine Translation

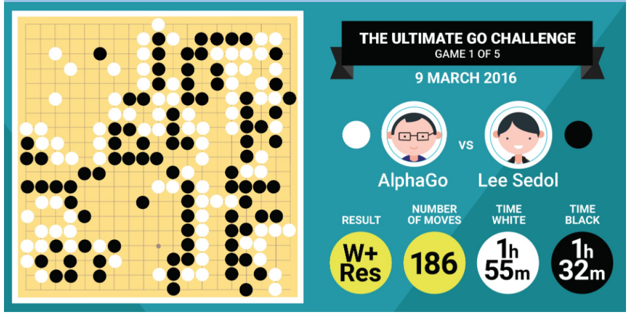
Autonomous Driving

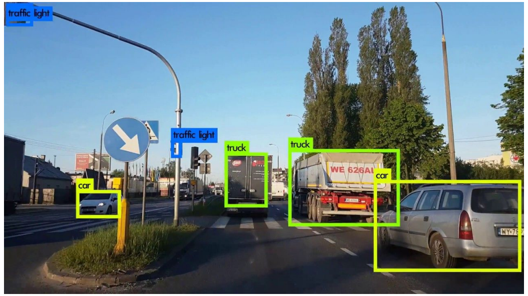


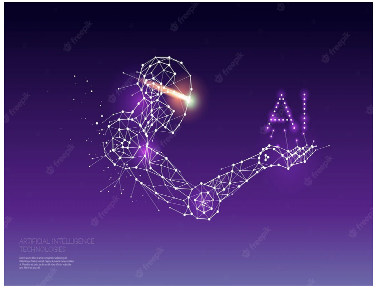
Speech Recognition



Intelligent Agent







Challenges

- AI models are over-parameterized and resource hungry
- Limit computational power and resource on deployed devices
- Inference-sensitive applications

Solutions

- Increase resource capacity
- Model compression: network pruning

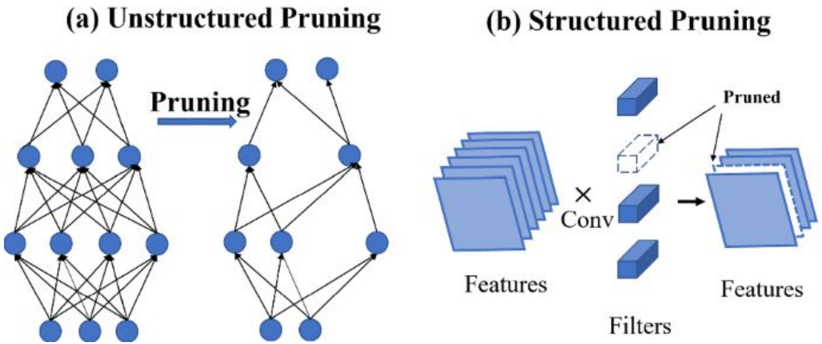


Fig 2. Chen, Liyang et al. (2021). Knowledge from the original network: restore a better pruned network with knowledge distillation.

SoTA pruning method

Traditional methods

- Labor-costly
- Expertise knowledge required for a specific task

RL-based methods

- Manually design vectors to represent DNN’s hidden layer.
- Rigid RL environment.

Background & Motivation

DNNs are essentially computational graph

Every pruning causing topology changes

Learn pruning policy from DNN's topology

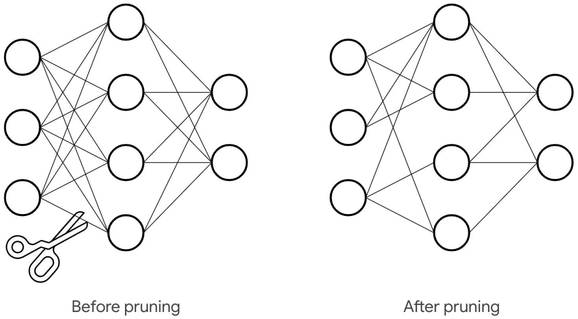


Fig 1.Illustrate of network pruning. Raziel Alvarez et al. https://blog.tensorflow.org/2019/05/tf-model-optimization-toolkit-pruning-API.html

Computational graph topology changes → Leverage GNN to perceive the topology changes → Use RL agent to optimize pruning policy

Objective

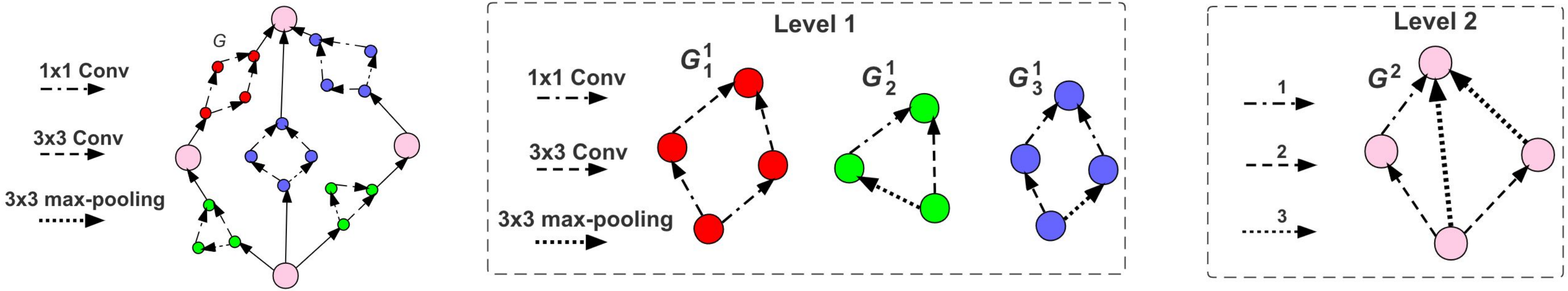
- Model DNN as a graph
- Use Graph Neural network to learn DNN representation
- Construct RL environment

Modeling hierarchical computational graph

- DNNs are essentially computational graphs.
- DNNs often contains various patterns (a.k.a. motifs).
- Motifs (such as conv 3x3) repeated throughout the network topology.
- Repeated Motifs have same topology.

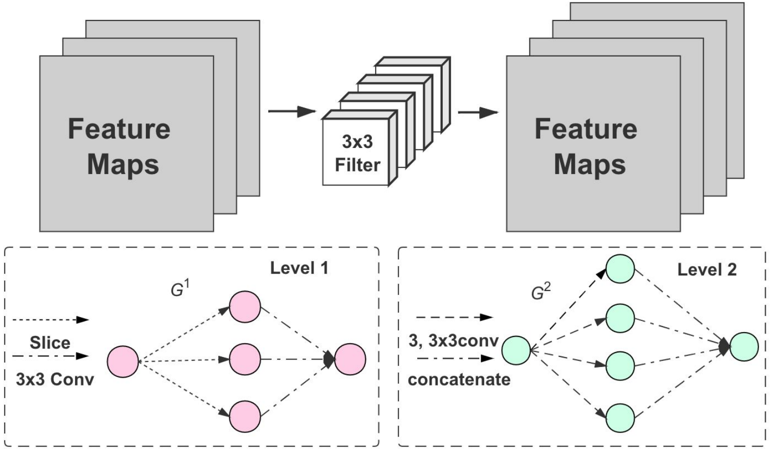
Modeling hierarchical computational graph

- Plain computational graph are huge Memory explosion
- A computational graph with motifs (the sub-graph painted red, blue, and green).
- Embed it hierarchically



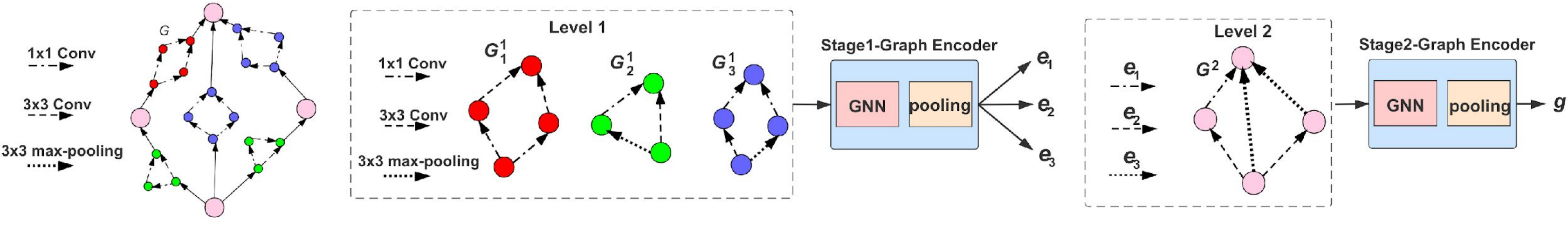
Modeling hierarchical computational graph

- Example



Multi-stage graph embedding

- Motifs have same topology, embedding motif first.
- (b) We extract motifs from G and split the G into 2 hierarchical levels.
- (c) The edges in G correspond to motifs at level-1.



(a) (b) (c)

Multi-stage graph embedding (m-GNN)

Message passing of m-GNN

$$h_i^{l+1} = \sum_{j \in N_i} \frac{1}{c_i} W^l (h_j^l \circ e_k^{l-1})$$

- m-GNN embeds the example hierarchical computational graph into two stages. At stage one, m-GNN embeds the motifs.
- Second stage, m-GNN applies motifs embeddings as the edge features.

Reinforcement Learning Environment

Environment states
- DNN's computational graph representation

Action space
- Pruning policy

Reward
- Pruned model's performance

Episode exit
- Target model size

RL Policy
- PPO

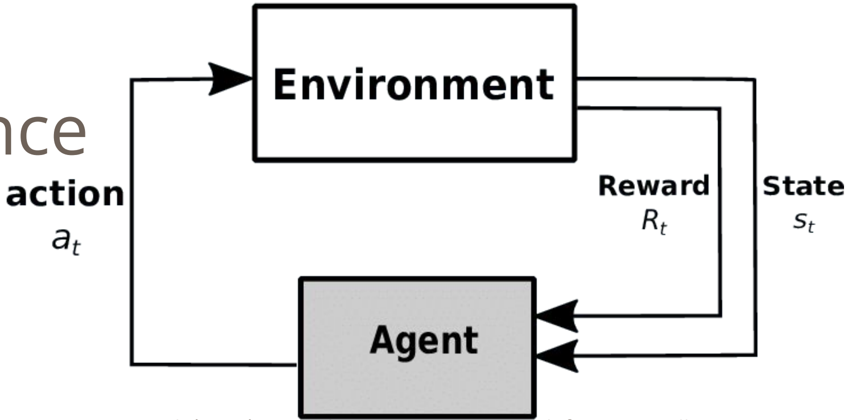
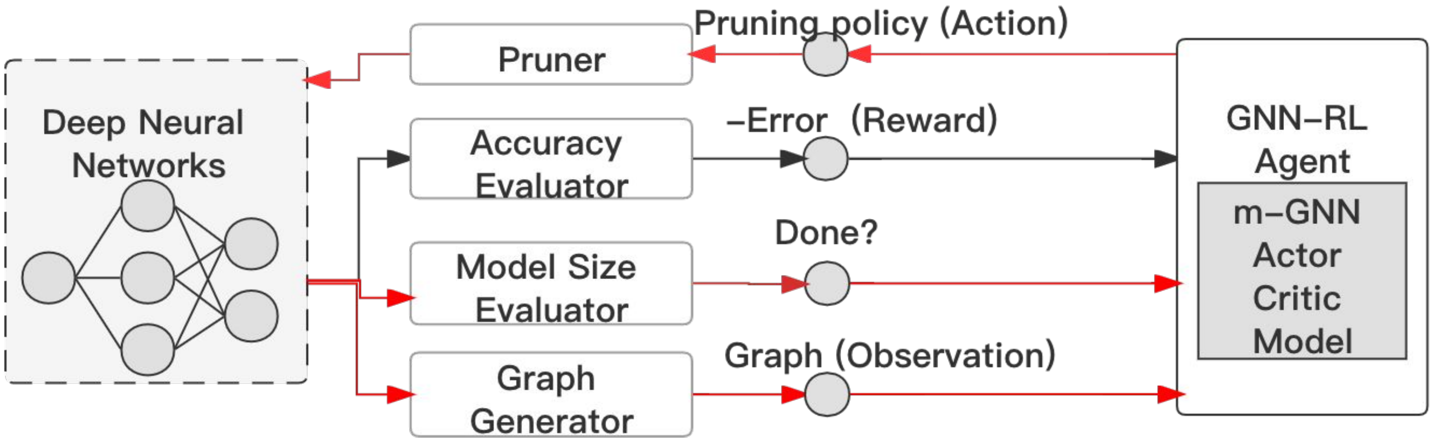


Fig 1. Amiri et al. (2018). A Machine Learning Approach for Power Allocation in HetNets Considering QoS.

GNN-RL Overview



Pruning Results – CIFAR-10/100

Model	Dataset	FLOPs↓	Top-1 Acc. %	ΔAcc. %
ResNet-100	CIFAR-10	52%	94.31	+0.63
ResNet-56	CIFAR-10	54%	93.49	+0.10
ResNet-32	CIFAR-10	51%	92.58	-0.05
ResNet-20	CIFAR-10	51%	91.31	-0.42
ShuffleNet-V1	CIFAR-100	42%	67.10	-2.84
ShuffleNet-V2	CIFAR-100	46%	66.64	-2.21

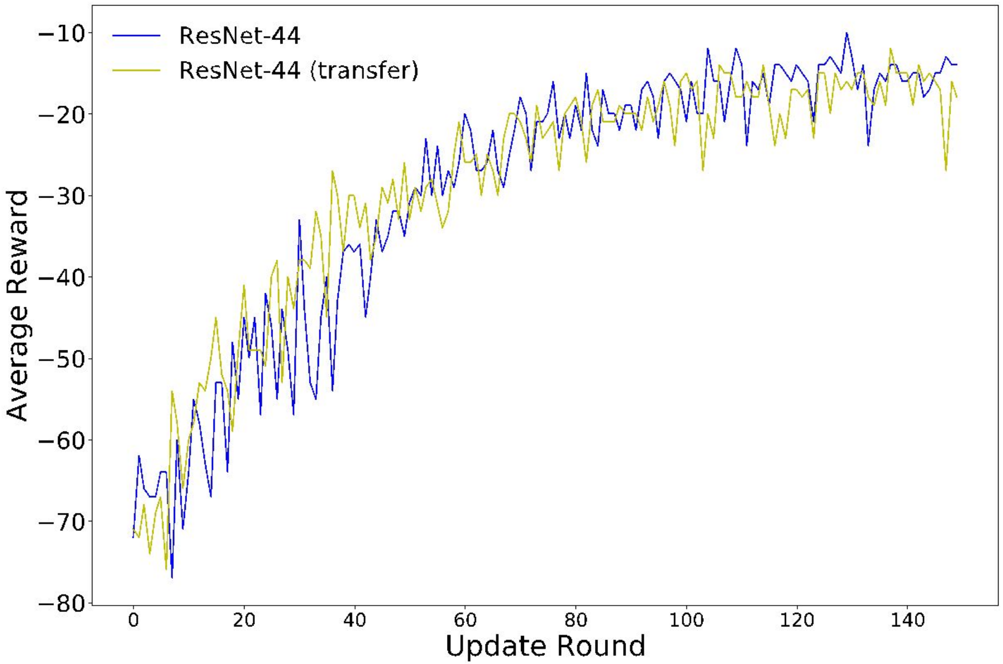
Pruning Results – ImageNet

Model	Dataset	FLOPs↓	Top-1 Acc. %	ΔAcc. %
VGG-16	ImageNet	80%	70.99	+0.49
ResNet-18	ImageNet	51%	68.66	-1.10
ResNet-50	ImageNet	53%	74.28	-1.82
MobileNet-V1	ImageNet	60%	69.50	-1.40
MobileNet-V2	ImageNet	42%	70.04	-1.83

GNN-RL achieves comparable results with SoTA!

Topology transfer

- GNN-RL trained on a topology can be transferred to another topology.
- We train GNN-RL on ResNet-56 then transfer it to ResNet-44
- Topology transfer offers a rapid pruning process (1.12X faster for each round) with much less computing time



Extension

GNN-RL is not limited on Model compression.

You can customize GNN-RL by define your customized RL task (e.g., action space, environment states, rewards).

Currently, our colleagues are testing GNN-RL on job scheduling task.

Extension -- m-GNN

Multi-stage graph embedding

Protein molecular

Thank you!