PUCRS
Pontifícia Universidade Católica
do Rio Grande do Sul

GPIN

Neural Architecture Search Using Automated Planning

Felipe Roque Tasoniero

Automated Planning 2019/02

Objectives

- Explore the use of the MetaQNN algorithm (Baker et al. 2016) on MNIST dataset.
- Penalize the reward function when the model searched reaches a determined size, and when forward passes are slow.
- Constraint memory and incentivize quick inference, trying to reduce the amount of time spent on searching a high-performing architecture.

Neural Architecture Search (NAS)

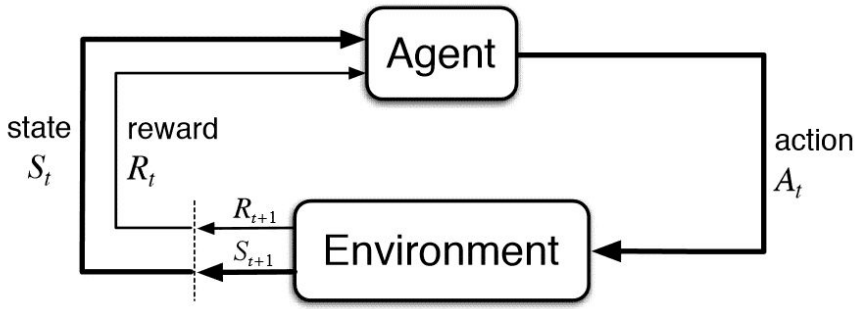
- Automatically search a neural network architecture that perform as well as ones designed manually by experts.
- Reinforcement Learning and Evolutionary Algorithms (widely used, but there are other techniques that has been used for Neural Architecture Search).
- Great amount of GPUs are necessary to find a high-performing architecture (In *Zoph et al. 2016,* they use 800 GPUs concurrently to train their model)

MetaQNN

- Algorithm based on Reinforcement Learning to automatically generate a high-performing CNN architecture for a given learning task.
- *Baker et al.* have shown that this algorithm is capable of searching a high-performing CNN architecture by using only 10 GPUs during 8~10 days.
- MetaQNN algorithm use Q-learning with ϵ-greedy exploration strategy and experience replay.

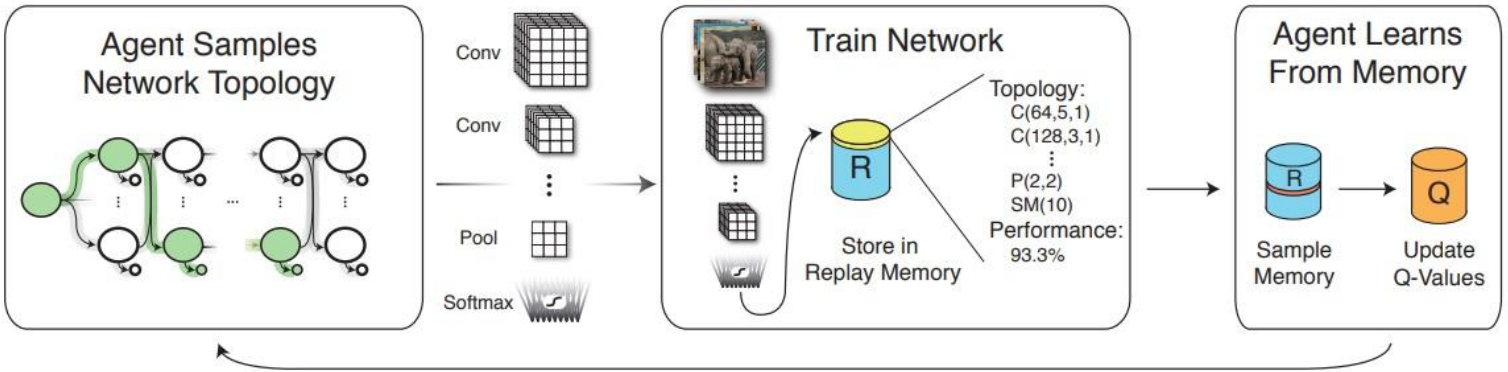
MetaQNN

- Q-learning
  - Used to design the CNN architecture.
  - Teaching an agent to find optimal paths as a Markov Decision Process (MDP).
  - Type of layers used: Convolution, Fully Connected, Pooling, Global Average Pooling, and Softmax.



MetaQNN

- ϵ-greedy exploration strategy and experience replay
  - We assume ϵ from 1 $\rightarrow$0 such as the agent begins in an exploration phase and slowly starts moving towards the exploitation phase.
  - Experience replay provide a memory of its past explored paths and rewards.



Project Management

- **Week 1:** Investigation phase that we are going to explore the use of MetaQNN and set up an initial penalize configuration.
- **Week 2:** Testing the algorithm and make some modification if necessary.
- **Week 3 and 4:** Due to the necessity of using multiple GPUs, we intend to use about two weeks to train and validate the model using our penalize configuration.
- **Week 5:** We let this week just for result analysis and final paper writing.

Final Considerations

- As we proposed for this work, we expect to generate a high-performing CNN architecture using fewer GPUs and less time of compute processing.
- It could be interesting, as an additional work, to investigate the use of MetaQNN for small and/or specifics images datasets.

THANK YOU!