Safety Aware Neural Pruning for Deep Reinforcement Learning (Student Abstract)

Briti Gangopadhyay, Pallab Dasgupta and Soumyajit Dey*

Indian Institute of Technology Kharagpur
IIT Kharagpur, Kharagpur, West Bengal 721302
briti_gangopadhyay@iitkgp.ac.in

Abstract

Neural network pruning is a technique of network compression by removing weights of lower importance from an optimized neural network. Often, pruned networks are compared in terms of accuracy, which is realized in terms of rewards for Deep Reinforcement Learning (DRL) networks. However, networks that estimate control actions for safety-critical tasks, must also adhere to safety requirements along with obtaining rewards. We propose a methodology to iteratively refine the weights of a pruned neural network such that we get a sparse high-performance network without significant side effects on safety.

Introduction

Deep Reinforcement Learning (DRL) has shown promise in learning human-like control policies for safety-critical domains such as autonomous driving. However, these policies often utilize large DNN architectures with millions of neurons, adversely affecting latency and memory utilization during inference. Real-time (or lower latency) and memory-optimized models are essential for the deployment of DRL policies in embedded systems with memory limitations. Neural network pruning is one of the most widely studied techniques for network compression. Neural network pruning refers to the act of removing parameters from a pre-trained and optimized neural network. It may entail removing specific parameters or groupings of parameters leading to enhanced efficiency while maintaining performance. There has been extensive research on the utility of sparse networks in computer vision and sparse networks have been shown to perform as well as dense networks (Frankle and Carbin 2019). Although the need for network compression is very significant for DRL-based controllers, the subject has been largely unexplored.

Research in computer vision compares the performance of sparse networks in terms of classification accuracy. In line with the advances made in computer vision, performance comparisons for sparse networks in DRL have been solely made in terms of the maximum reward obtained (Graesser et al. 2022). For safety-critical control tasks, this evaluation metric is not enough. For DRL agents that learn policies for real-world applications, respecting safety requirements is just as important as obtaining high rewards. A compressed network achieving high rewards may still have safety violations since violations having minimal effects on reward reduction may be difficult to uncover. Even when safety constraints are included in the optimization objective of the dense network pruning weights may undo the effects of safe training. In this work, we propose a methodology for iterative refinement of the pruned network such that we obtain a maximal sparse network with high rewards and no safety infractions with respect to failures uncovered through testing.

Methodology

Given an optimized dense network π_{opt} parameterized by θ for an RL task, our aim is to construct a sparse network π_s parameterized by θ' such that $\theta' \subset \theta$ and $\forall s_0 \in S_0$, $\mathcal{R}(\tau') \geq \delta$ and $\tau' \models \phi$ where s_0 is an initial state belonging to the set of all initial states S_0, τ' is any trajectory starting from s_0 following π_s, $\mathcal{R}$ denotes the reward of the trajectory, ϕ denotes given safety specifications and δ is the reward cutoff.

The pruning and refinement process is divided into the following broad steps (see Fig. 1):

1. We choose a k and perform one-shot pruning of the weights of π_{opt} to obtain π_s. Here, $k\%$ of the weights with minimum activation are set to 0 and the rest of the weights are retained. k is iteratively decreased until a network π_s is obtained with an average reward, over 100 executions, at least equivalent to the reward threshold δ.
2. We obtain a set of counterexample trajectories ξ_f against a safety specification through testing of the pruned network. We choose Bayesian Optimization (BO) as the testing technique as it has been extensively studied in the literature for testing intelligent controllers (Gangopadhyay et al. 2021).
3. For each counterexample trajectory $\xi_{f_i} \in \xi_f$ we calculate the recovery state s_{r_i}. A recovery state is defined as follows:

   Definition 0.1 ***Recovery State****: A recovery state s_{r_i} in a counterexample trajectory $\xi_{f_i} \in \xi_f$ is defined as the last state $s_i \in \xi_{f_i}$ such that trajectory τ' starting from s_i*

*This work was partially supported by the TCS Research Scholarship.

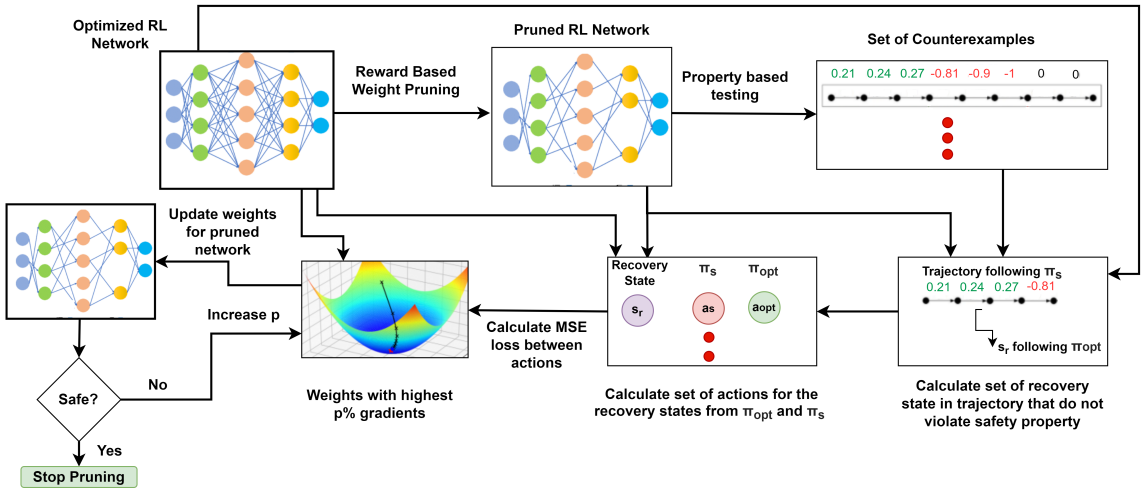

Figure 1: Illustration of the Safety Aware Neural Pruning Methodology

following π_{opt} is safe i.e, $\tau' \models \phi$ and $\mathcal{R}(\tau) \geq \delta$ where τ is a trajectory starting from s_{i_0} following τ'.

This is achieved through backtracking. We start from the last state $s_{n_i} \in \xi_{f_i}$ and set this as the initial state of the environment and obtain a trajectory τ' executing π_{opt}. This process continues iteratively until a state s_{r_i} is discovered starting from which trajectory τ' is safe.

4. For each unique s_{r_i} we calculate the action proposed by the original network $a_{opt_i} = \pi_{opt}(s_{r_i})$ and sparse network $a_{s_i} = \pi_s(s_{r_i})$. The objective is to shift each a_{s_i} towards a_{opt_i} in π_s so that failures ξ_f are corrected in π_s.
5. To achieve the correction we calculate the Mean Square Error (MSE) loss between all a_s and a_{opt} obtained from s_r, $L(a_s, a_{opt}) = \frac{1}{N}\sum_{i=0}^{N}(a_{s_i} - a_{opt_i})^2$. We calculate the Jacobian matrix of the loss with respect to all the weights in π_s. This gives us the gradient information of the locations where weights of π_s need to be replaced to achieve a_{opt}. Layer-wise, we select $p\%$ of the weights from π_{opt} with the highest gradient information. These weights are made non-zero in π_s. p is iteratively increased until all the trajectories in ξ_f are corrected.

Empirical Studies

We test our methodology on the Cartpole, Lunar-Lander, and Bipedal Walker environment from the Open AI Gym environment suite. For Cartpole, the objective is to control a pole on top of the cart by applying a force of +1 or -1 to the cart and the goal is to prevent the pole from falling over. We choose $\delta = 195$ and the following safety specifications:

1. The cart should not go beyond -2.4 (left) or +2.4 (right).
2. The cart maintains a momentum between -2.0 and 2.0
3. The angle made by the pole should be greater than 0.2 with respect to the rest position

We first obtain a pruned network π_s with 95% of the weights set to 0. We then test π_s against the given safety specifications and a total of 37 counterexamples are uncovered.

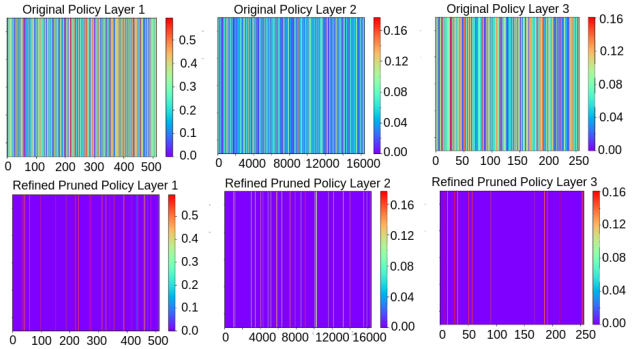

Figure 2: The original and sparse networks for Cartpole. The violate regions denote weights of magnitude 0. The x-axis denotes parameters and the y-axis denotes weight magnitude. The refined sparse network has 94.6% weights pruned.

Then π_s is iteratively refined using the proposed methodology to obtain the refined network which has 94.6% of the weights set to 0 as illustrated in Fig 2. For Lunar-Lander and Bipedal Walker, we obtain a network with 39.48% and 42.35% weights pruned respectively. This work reports our preliminary but promising results on safety-aware network compression for RL. In the future, we aim to formally verify the compressed network to guarantee no safety infractions.

References

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *7th ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.*

Gangopadhyay, B.; et al. 2021. Counterexample Guided RL Policy Refinement Using Bayesian Optimization. In *NeurIPS*, volume 34, 22783–22794.

Graesser, L.; et al. 2022. The State of Sparse Training in Deep Reinforcement Learning. In *ICML*, 7766–7792. PMLR.