Paper: *Playing Atari with Deep Reinforcement Learning*

Nirajan Koirala

CSC 9010_001

Summary

Reinforcement Learning is an area within Machine Learning concerned with agents who take actions in an environment in order to maximize some notion of cumulative reward. It presents several challenges from a deep learning perspective as RL algorithms must be able to learn from a scalar reward signal that is frequently sparse, noisy and also occurrence of delay between actions and rewards, sequences of highly correlated states and constant change in data distribution are major issues for RL learning algorithms. Authors demonstrate in this paper that a CNN can overcome these challenges to learn successful control policies from raw video data in complex RL environments. The network they use is trained with a variant of Q-learning algorithm with SGD to update weights. They have also used a experience replay mechanism to tackle problems of correlated data and non-stationary distributions. They have applied this approach to a wide range of Atari 2600 games implemented in the Arcade Learning Environment and are able to outperform all previous RL algorithms on 6 of the 7 games and surpassed human player on 3 of them.

In section 2, authors provide a background on their methods and describe the keywords like environment, agents, observation, states and rewards and how they are used in the method of learning. The agent only observes an image from the emulator and receives rewards based on the sequence of actions and observations. Since it is not possible to fully understand the current situation using only the current screen, they consider sequences of actions and observations and learn game strategies which depend on these sequences. The goal of the agent is to interact with the environment and select actions that maximizes future rewards. Using discount factors they define the optimal action-value function $Q^*(s,a)$ which obeys the Bellman equation. A function approximator is used to estimate the action-value function as using a value iteration algorithm to converge to a optimal action-value function is an impractical approach. The Q-network which they use to approximate the function is trained by minimizing a sequence of loss functions that changes at every iteration. They provide further explanation about their loss function and learning method and mention that the algorithm they use is model-free which means it solves the RL task directly using samples from the emulator and it is off-policy as well which means the algorithm follows a greedy strategy while following a behavior distribution that ensures adequate exploration of the state space. In the next section, they provide some previous success stories of RL and how their approach differ from the approaches used previously. In particular, they consider SGD updates that have a low constant cost per iteration and scale to large sets and they apply RL end-to-end directly from visual inputs.

In section 4, authors talk about their deep reinforcement learning approach which is to connect a RL algorithm to a deep NN which operates directly on RGB images and efficiently process training data using stochastic gradient updates. A new technique called experience replay is employed and agent’s experiences at each time step is stored in a data set and pooled over many episodes into a replay memory. Later they provide the full description of their full algorithm, deep Q-learning. They discuss about its advantages like greater data efficiency and randomization of the sample breaks. It is also pointed that using experience replay the behavior distribution is averages over many previous states which smooths out learning and avoids oscillations or divergence in the parameters. Since the algorithm only stores the last N experience tuples in the replay memory and samples uniformly at random from memory D, it is limited as memory buffer does not differentiate important transitions and always overwrites using recent transitions due to finite memory size. They also mention that uniform sampling gives equal importance to all transitions in the memory and a more sophisticated sampling strategy might emphasize transitions from which we can learn the most. In the subsection, they talk about all the pre-processing that they apply to images and the model's architecture which consists of separate output for each possible action of agent and only the state representation is inputted to the neural network. They keep the architecture same for all the Atari games and the CNN trained with their approach is called Deep Q-Networks.

Even though different games have different winning objectives, they use the same network architecture, learning algorithm and hyperparameters across all 7 games. It shows that their approach is robust and they also clip the rewards to limit the scale of error derivative which makes it easier to use the same learning rate across multiple games. They also use a frame-skipping technique to help decrease the overall computation and this technique also allows the agent to play games more times without significantly increasing the run-time. They discuss the challenges relating to progress of an agent during training and point out their metric which is suggested by the total reward the agent collects in an episode or game averaged over a number of games. They also talk about other stable metric which provides an estimate of how much discounted reward the agent can obtain by following its policy from any given state but later suggest that their method is able to train large neural networks using a RL signal and SGD in a stable manner. Later in the sub-sections they discuss their results with the best performing methods from the RL literature. They point out that other methods incorporated significant prior knowledge about the visual problem by using background subtraction and treating each of the 128 colors as separate channel but in contrast agents in their method only receive the raw RGB screenshots as input and must learn to detect objects on their own. DQN outperforms the other learning methods by a substantial margin on all 7 Atari games despite incorporating almost no prior knowledge about the inputs.

The algorithm they use is evaluated on ϵ-greedy control sequences generalizes across a wide variety of possible situations. They show that on all games except Space Invaders, their average as well as max evaluation results achieve better performance than all other learning methods. Finally, they mention that their method achieves better performance than an expert human players on some of the games and their approach gave the state-of-art-results in 6 of the 7 games it was tested on without any adjustment of the architecture or hyperparameters.

Strengths

- The methods they have discussed are very robust the architecture used doesn’t need to be modified for different games.
- I think the approaches mentioned in the paper are very clear, concise and to the point.
- The experiments they’ve performed are on different type of 7 games which is able to highlight the strengths as well as weaknesses of their methods.

Weaknesses

- I think the section 3 (related work) could have been presented before the background section because it provides a little history on RL methods.
- The figures provided in the paper could have been enlarged a little.

Points of Confusion

- I am confused about the Markov Decision Process and how it works.
- It was confusing to grasp the role of discount factor in future rewards.

Discussion Questions

- A discussion on the workings of their algorithm would be beneficial.
- How would a sophisticated sampling strategy work so that it emphasizes the transitions from which learning can be the most?
- What is a HyperNEAT evolutionary architecture and how is it designed?