Jenish Maharjan **Week 6**

Paper: *Playing Atari with Deep Reinforcement Learning*

Summary:

The paper "Playing Atari with Deep Reinforcement Learning" by Mnih et al describes an Atari game playing program created by the company Deep Mind which learned to play seven Atari games without game specific directions from the programmers. The deep learning model used by the program was able to successfully learn to play games based on high-dimensional sensory input using reinforcement learning. The model consisted of a convolutional neural network with which was trained with a variant of Q-learning. The model was trained to learn a value function estimating future rewards based on the input raw pixels.

Reinforcement Learning (RL) faces multiple challenges – they depend on a scalar reward signal that is frequently sparse, noisy and delayed, the data samples are usually highly correlated states and the data distribution is not stationary. The authors use a convolutional neural network than can overcome these challenges to learn from raw video in complex RL environments. To solve the issue of correlated data and non-stationary distributions they implement an experience reply mechanism which randomly samples previous transitions to smooth the training distribution over many past behaviors. They describe that an agent is able to observe an image from the game emulator without access to the internal state and the reward which depends on the whole prior sequence of actions and observations. Since it's impossible to understand the current situation fully from only the current screen, they use sequences of actions and observations to learn game strategies. Since all sequences in the emulator are assumed to terminate in a finite number of time-steps standard RL methods can be used for Markov decision process (MDP) by using the complete sequence as the state representation at a particular instant of time.

The goal of the agent is to maximize future rewards and the standard assumption is that the future rewards are discounted. The optimal action-value function is defined as the maximum expected return achievable by following any strategy, after seeing some sequence and then some action. The basic idea behind many RL is to "estimate the action value function by using the Bellman equation as an iterative update" and such value iteration converge to optimal action value function. RL methods use non-linear function approximator with weights as a Q-network. It implements a model-free approach which uses samples from the emulator without construction an estimate of emulator. They also implement a greedy strategy with some adequate exploration of the state space. The goal of deep reinforcement learning is to connect a RL algorithm to a deep neural network which operates directly on RGB images and uses stochastic gradient updates. They utilize experience replay which applies Q-learning updates to samples of experience randomly from the pool of stored samples. The advantages of deep reinforcement learning over standard online Q-learning are that each step of experience is potentially used in many weight updates, randomizing the samples breaks the correlation between the data samples and reduces the variance of the updates.

The input raw frames are converted from RGB to grey-scale, down-sampled and finally cropped to 84x84 region of rough captures of the playing area before feeding them to the network. Thus, the input to the neural network is 84x84x4 image since it uses the last 4 frames of the history. The first hidden layer uses a 168x8 filters with stride 4, the second hidden layer uses a 324x4 filters with stride 2 and the final hidden layer is a fully-connected layer with 256 rectifier units. The output of network is a fully connected linear

layer with a single output for each valid action. They trained the network using RMSProp with mini batches of size 32 and the behavior policy during training was epsilon-greedy. The authors have compared the results with the previous best performing RL methods. They demonstrate that their method achieves better performance than an expert human player on three of the Atari games and achieves close to human performance on the fourth. The performance on the remaining three games were not comparable as they require the network to find a strategy that extends over long time scales.

Strengths:

- They provide an elaborate introduction and background section which helps readers understand existing standards and norms in reinforcement learning before they dive into their contribution to the field.
- They present a table to compare the performance of their model with many existing models which helps in evaluation of the model.

Weaknesses:

- While having elaborate introduction and background sections helps to setup for the description of their work, it does take away the focus from their contributions. More concise introduction and background sections covering the same topics would have been better.
- A diagrammatic representation of the model architecture and how the different components interact to produce the game playing program would have assisted in the understanding of their work.

Confusions:

- I didn't quite understand the Bellman equation.
- How is the "adequacy" of "exploration of the state space" determined?

Discussion Questions:

- How would using RGB or Gray-scaled images change the model's performance?
- How would fine tuning the hyperparameters for each game change the performance?
- A discussion on online Q-learning would be helpful.