19D070052 – Sheel Shah

a. Why are convolutional neural networks a good choice for both these applications?
-> For Atari games, CNNs are a good choice because they can process images well, and the state in an Atari game is one frame which is an image. This is mainly because there is relevant information in a physically close set of pixels than in just single pixels, and CNNs use exactly this to get feautures of the image. For Go, the input to the neural network is a 17 layer image, where each layer has only binary values. Hence, the CNN can again use proximity based information, and since the 17 layers are similar (time shifted), the feature extraction done by the CNN can generalize well over the layers.

b. Why do you think the DQN algorithm proposed by Mnih et al. is unable to perform well on Ms. Pac-Man?
-> I think this is mainly because the state space is too large, and while the DQN generaizes enough to perform well in standard scenarios, there are several edge cases that rarely occur and hence the network is unable to learn about those scenarios.

c. What is the main difference between the result achieved by Silver et al. (2016) and that described in the paper below by Schaeffer et al. (2007)?
-> Schaeffer et al. have computationally prooved that optimal checkers players end the match in a draw, and have come up with an explicit strategy that won't ever lose the game. On the other hand, Silver et al. have come up with an agent that plays really well, but is not guaranteed to win. There are no guarantes on the optimality of the agent, or on what the result of a game with two optimal players would be.