Review Questions 5
ID2223
Group: Nicolas Oulianitski Essipova, Peter Lakatos, Marios Chatiras

1. Generative models in ML and statistics are models that are generated from data by modelling the conditional probability distribution. These models only apply to probabilistic methods and can learn the correlation between features and complicated distributions. Generative adversarial networks or GANs are generative autoencoders used to generate new data given a data set, with the same statistics.

2. The GAN functionality is based on the convergence of the two networks: the generator network and the discriminator network. At the equilibrium point, the generator produces samples that are discriminator cannot distinguish them apart from the dataset samples, and the discriminator is discarded. This equilibrium point represents the total minimum of the cost function but might not be reached because either network could be stuck in a local minimum of this function.

3. A mini-batch standard deviation layer in neural networks is a layer which calculates the standard deviations of features in a sample and uses it as an extra feature. In GAN training, a mini-batch standard deviation layer can be a part of the discriminator network, to enforce that the newly created samples share similar features with the original data set.

4. It is a solution to a non-cooperative game, where players are aware of each other's benefitting strategies but neither of them is gaining anything from changing only their own strategy. GAN is also a type of zero-sum non-cooperative game, where if the generator wins then the discriminator loses and vice versa. As we train the GAN model more and more, we could reach a state of Nash equilibrium, but it is not guaranteed.

5. There are numerous ways that reinforcement learning (RL) differs from both supervised- (SL) and unsupervised learning (USL). The ultimate goal for RL is to find a policy that generalizes well to the environment and takes the appropriate actions to maximize given reward function – in contrast to SL and USL wherein they try to find patterns in the data and minimize the loss function.

6. The credit assignment problem is when we cannot track back the connection between rewards and a sequence of actions. We have no idea of determining retroactively of a single action whether what effect it had on the outcome. To illustrate it in a simple example: when playing chess, we need to be able to identify, which steps helped our final outcome, and which did not. Three types of credit assignment problems can be identified: the temporal, the structural and the transfer credit assignment. Possible solutions to the problem are reward shaping, by introducing intermediate results or having more computational power.

7. A) Following are the results for the UCB1 strategy:
   S(i) = Q(i) + c*U(i)

   Q(1) = 4/7
   Q(2) = 3/5
   Q(3) = 0

   U(1) = 0.6140099266
   U(2) = 0.7265063427
   U(3) = 1.148707389

   S(1) = 1.185438498
   S(2) = 1.326506343
   S(3) = 1.148707389
   Slot machine two has the highest S(i) score, therefore we pick that for the next action.

   B) With the ε-greedy policy, we are looking for the highest average reward:
   A(1) = 4/7
   A(2) = 3/5
   A(3) = 0
   This strategy also chooses slot machine 2 for the next action.

8. Tabular version is particularly useful when approaching problems with a discrete state space and the Bellman equation could be used to calculate the state-value and action-value expected reward returns. However, in many cases, this is not possible. In cases where abstraction of the state space is necessary and computing the expected reward return for each action would hog resources with the brute force approach of implementing the Bellman equation; a neural network can be greatly beneficial.