



i Exam: DAT440 / DIT471 Advanced topics in machine learning**Examiner and teacher:** Morteza Haghir Chehreghani, morteza.chehreghani@chalmers.se

- The final exam is in the form of a digital exam (to be done on Inspera).
- The exam will take four hours.
- There are in total 34 questions: Most of the questions have one point/mark, and there are few questions with two or three points/marks. The mark of each question is specified.
- The total score is 40, which will be normalized to 60 and then the final grade will be computed as described in the course canvas page.
- The exam must be done individually, and you cannot use cheat-sheet or any other resources.
- You can have an ordinary calculator for the exam.
- The exam consists of only multiple-choice questions, where for each question, at least one option (and possibly more than one) can be correct. The correct number of options is specified for each question.
- There will be no negative score for wrong answers, but you need to choose all (and only) the correct answers to get the mark/score of the question.
- The examiner will visit the exam premises on two occasions to answer the clarification questions: i) one hour after the start of the exam, and ii) when an hour of the exam remains.
- You do not need to show your calculations for the questions.

1 For regret analysis of a bandit algorithm, in what settings is the bad event often negligible?

[Number of correct options: 2]

- The total number of rounds is large. 
- The total number of arms is large.
- The probability of the clean event is significantly larger. 
- The upper confidence interval is small.

Rätt. 1 av 1 poäng.

2 Choose the correct statements about an arm chosen by the UCB1 algorithm in bandit problems.
[Number of correct options: 2]

- The arm is likely to have a large high reward.
- The arm has been sufficiently explored.
- The arm has not been explored enough.
- The arm has a small confidence radius.



Rätt. 1 av 1 poäng.

3 Consider a stochastic K -armed bandit problem with mean reward $\mu(a)$ for each arm $a \in \mathcal{A}$, where \mathcal{A} is the set of arms and $|\mathcal{A}| = K$. We define, for an arbitrary arm $\hat{a} \in \mathcal{A}$, the observed average reward $\bar{\mu}_t(\hat{a})$ of arm \hat{a} before round t . Moreover, $r_t(\hat{a})$ is the confidence radius at round t for arm \hat{a} , and $\text{UCB}_t(\hat{a})$ and $\text{LCB}_t(\hat{a})$ are the upper and lower confidence bounds, respectively, of arm \hat{a} at round t . Choose the correct statements about the confidence bounds/radius.

[Number of correct options: 2]



- Under bad event, $\text{LCB}_t(a) < \text{LCB}_t(a')$ for $a, a' \in \mathcal{A}$, $a \neq a'$ always indicates $\mu(a) < \mu(a')$.
- The confidence radius is a random variable because of randomness in the rewards and algorithm.
- If $\text{UCB}_t(a) < \text{UCB}_t(a')$ for arms $a, a' \in \mathcal{A}$ where $a \neq a'$, the mean reward of arm a' is larger than the mean reward of arm a .
- Under the event $\mathcal{E} := \{\forall a \in \mathcal{A} \forall t \quad |\bar{\mu}_t(a) - \mu(a)| \leq r_t(a)\}$, $\text{UCB}_t(a)$ provides upper bound for $\mu(a)$.



Rätt. 1 av 1 poäng.

4 Choose the correct statements about Thompson sampling in bandit problems.



[Number of correct options: 2]

- Takes a sample from the posterior distribution over mean rewards and chooses best arm according to this sample. 
- Samples a mean reward vector within the confidence intervals and chooses the best arm according to this vector.
- Samples an arm from the posterior distribution of it being the optimal arm. 
- Chooses the arm with the highest upper confidence bound according to the posterior distribution over mean rewards.

Rätt. 1 av 1 poäng.

5 Choose the correct statements about Thompson sampling with independent priors in bandit problems.

[Number of correct options: 2]

- Mean rewards only need to be sampled for one arm in each round.
- Needs to update only the played arm at each round. 
- Mean rewards can be sampled independently for each arm. 
- The posterior can be computed directly without summation over all possible mean rewards.

Rätt. 1 av 1 poäng.

6

Choose the correct statements about the reward for a chosen arm in stochastic multi armed bandit problems.

[Number of correct options: 2]

- Depends on previously chosen arms
- Depends on the chosen arm
- Depends on an unknown distribution
- Depends on the state of the environment



Rätt. 1 av 1 poäng.

7 Choose the correct statements about Bayesian bandit problems.

[Number of correct options: 2]

- Tries to identify an arm that is optimal over all problem instances.
- Parameters of the reward distribution are drawn from a prior distribution.
- Uses maximum likelihood estimation to determine the reward distribution.
- Tries to infer the most probable reward distribution with respect to observations and prior beliefs.



Rätt. 1 av 1 poäng.

8

Consider a Bayesian K -armed bandit problem with a set of arms \mathcal{A} , $|\mathcal{A}| = K$. Let $r_t \sim \mathcal{D}_{\mu(a_t)}$ denote the reward of arm $a_t \in \mathcal{A}$ chosen at round t , where the reward distribution $\mathcal{D}_{\mu(a_t)}$ is specified by its expected reward $\mu(a_t)$. Let $\mathbb{P}_H(\tilde{\mu}) := \Pr[\mu = \tilde{\mu} | H_t = H]$ denote the Bayesian posterior distribution after round t . Let H be a feasible t -history that is a concatenation of some feasible $(t-1)$ -history H' and an action-reward pair (a, r) , i.e., $H = H' \oplus (a, r)$. For a H -consistent algorithm, $\pi(a) = \Pr[a_t = a | H_{t-1} = H']$ is the probability that arm $a \in \mathcal{A}$ is chosen at round t given the history H' .

Assume that arm $a \in \mathcal{A}$ is chosen in round t and, subsequently, the reward r is observed. For a mean reward vector $\tilde{\mu} \in [0, 1]^K$, choose the correct statements about the joint distribution $\Pr[\mu = \tilde{\mu}, H_t = H]$.

[Number of correct options: 2]

- $\Pr[\mu = \tilde{\mu}, H_t = H] = \Pr[H_{t-1} = H'] \times \Pr[(a_t, r_t) = (a, r) | H_{t-1} = H']$
- $\Pr[\mu = \tilde{\mu}, H_t = H] = \Pr[H_{t-1} = H'] \times \mathbb{P}_{H'}(\tilde{\mu}) \times \mathcal{D}_{\tilde{\mu}(a)}(r) \times \pi(a)$ ✓
- $\Pr[\mu = \tilde{\mu}, H_t = H] = \Pr[H_{t-1} = H'] \times \mathbb{P}_{H'}(\tilde{\mu}) \times \pi(a)$ ✗
- $\Pr[\mu = \tilde{\mu}, H_t = H] = \Pr[H_{t-1} = H'] \times \Pr[\mu = \tilde{\mu}, (a_t, r_t) = (a, r) | H_{t-1} = H']$ ✓

Delvis rätt. 0 av 2 poäng.

- 9 Consider a Bayesian K -armed bandit problem with a set of arms \mathcal{A} , $|\mathcal{A}| = K$. Assume independent priors $\mathbb{P}(\boldsymbol{\mu})$ given by the Beta distribution, likelihood $\mathbb{P}(r|\boldsymbol{\mu})$ is given by the Bernoulli distribution and $K = 4$ arms.

Assume you sample the following mean reward vectors from the posterior distribution in rounds $t = 1, \dots, 4$:

Round 1: $\tilde{\boldsymbol{\mu}}_1 = [0.7, 0.3, 0.6, 0.4]$ where each entry corresponds to one of the arms.

Round 2: $\tilde{\boldsymbol{\mu}}_2 = [0.5, 0.4, 0.2, 0.3]$ where each entry corresponds to one of the arms.

Round 3: $\tilde{\boldsymbol{\mu}}_3 = [0.4, 0.8, 0.6, 0.1]$ where each entry corresponds to one of the arms.

Round 4: $\tilde{\boldsymbol{\mu}}_4 = [0.3, 0.4, 0.3, 0.5]$ where each entry corresponds to one of the arms.

You use Thompson sampling to choose one arm in each round and observe the following sequence of rewards $\{0, 1, 1, 0\}$ at different rounds (i.e., 0 in the first round, 1 in the second round, 1 in the third round, and finally, 0 in the fourth round).

Given that the prior is uniform at the first round, what are the values of posterior parameters in the next round $t = 5$?

Hint 1: For observations (of random variables) $\boldsymbol{x}_1, \dots, \boldsymbol{x}_t$, the parameters of a Beta posterior distribution are given by $(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (\boldsymbol{\alpha}_0 + \sum_{i=1}^t \boldsymbol{x}_i, \boldsymbol{\beta}_0 + t - \sum_{i=1}^t \boldsymbol{x}_i)$, where $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are the prior parameters.

Hint 2: A Beta distribution with parameters $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{1}$ yields a uniform distribution.

Note: Options are given in the following format: $\{(\alpha^1, \beta^1), (\alpha^2, \beta^2), (\alpha^3, \beta^3), (\alpha^4, \beta^4)\}$, where α^i and β^i are the posterior parameters of arm i .

[Number of correct options: 1]

$\{(2,4),(2,2),(2,1),(1,2)\}$

$\{(1,1),(1,0),(0,0),(0,1)\}$

$\{(2,2),(2,1),(1,1),(1,2)\}$

$\{(2,2),(3,2),(2,2),(2,3)\}$



Rätt. 2 av 2 poäng.

- 10 Consider a stochastic bandit problem with $K = 2$ arms with Gaussian rewards with means μ_1 and μ_2 , respectively. Assume that the first arm is the optimal arm, i.e., $\mu^* = \mu_1$, and $\mu_1 = \mu_2 + \Delta$ with $\Delta > 0$. It can be shown that the regret of the Explore-First algorithm will be upper bounded by

$$R(T) \leq \Delta \left(N + T \Phi \left(-\Delta \sqrt{\frac{N}{2}} \right) \right),$$

where Φ denotes the cumulative distribution function (cdf) of the standard Gaussian distribution.

Provide a value N^* of N that minimizes the above regret upper bound.

Hint 1: $\frac{\partial}{\partial N} \Phi(a(N)) = \phi(a(N)) \frac{\partial a(N)}{\partial N}$, where $\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$ denotes the probability density function (pdf) of the standard Gaussian distribution and $a(N)$ is an arbitrary function of N .

Hint 2: Use the Lambert function W , defined as for $y > 0$, $W(y) \exp(W(y)) = y$.

Note: $\lceil x \rceil$ is the **ceiling function** which maps x to the least integer greater than or equal to x .

[Number of correct options: 1]

$N^* = \left\lceil W \left(\frac{T^2 \Delta^4}{32\pi} \right) \right\rceil$



$N^* = \left\lceil \frac{T^2 \Delta^2}{16\pi} \right\rceil$

$N^* = \left\lceil \frac{2}{\Delta^2} \right\rceil$

$N^* = \left\lceil \frac{1}{\Delta} W \left(\frac{T^2 \Delta^2}{16\pi} \right) \right\rceil$

$N^* = \left\lceil \frac{T^2 \Delta^4}{32\pi} \right\rceil$

$N^* = \left\lceil \frac{2}{\Delta^2} W \left(\frac{T^2 \Delta^4}{32\pi} \right) \right\rceil$





Fel. 0 av 3 poäng.

11 Consider the definition of return in reinforcement learning as the following.

$$G_t \doteq \sum_{k=t+1}^T \gamma^{k-t-1} R_k$$

Note that T indicates the horizon or the total number of time steps. Choose the correct statements.




[Number of correct options: 2]

- For $T = \infty$ we have $\gamma = 1$.
- We have either $T = \infty$ or $\gamma = 1$ 
- We have $G_t = R_{t+1} + G_{t+1}$
- We have $G_t = R_{t+1} + \gamma G_{t+1}$ 
- We have $G_t = G_{t+1} + \gamma R_{t+1}$

Rätt. 1 av 1 poäng.

12 Choose the correct statement about Markov property in reinforcement learning.



[Number of correct options: 2]

- The current state depends only on the first state, not the others.
- The current state depends only on the most recent state, and not on any of the actions. 
- The current state depends on the most recent state and the action taken at that state. 
- The future is independent of the past given the present. 

Delvis rätt. 0 av 1 poäng.

13 What is the concept of "reward hypothesis" in reinforcement learning?



[Number of correct options: 2]

- It determines what should be achieved in a reinforcement learning task. 
- It specifies how to reach the goals/purposes.
- It is equivalent to choosing actions with maximal expected/estimated reward at each time step.
- It indicates the goals and purposes can be modeled by the maximization of the expected value of the cumulative sum of rewards. 

Rätt. 1 av 1 poäng.

14 Consider a Markov decision process for reinforcement learning. Choose the correct statements about the expected reward $r(\mathbf{s}, \mathbf{a}) \doteq \mathbb{E}[R_t | \mathcal{S}_{t-1} = \mathbf{s}, \mathbf{A}_{t-1} = \mathbf{a}]$ for state-action pairs (\mathbf{s}, \mathbf{a}) .

[Number of correct options: 2]

- $r(\mathbf{s}, \mathbf{a}) = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(r | \mathbf{s}, \mathbf{a}, s')$
- $r(\mathbf{s}, \mathbf{a}) = \sum_{r \in \mathcal{R}} r p(r | \mathbf{s}, \mathbf{a})$ 
- $r(\mathbf{s}, \mathbf{a}) = \sum_{r \in \mathcal{R}} r p(s', r | \mathbf{s}, \mathbf{a})$
- $r(\mathbf{s}, \mathbf{a}) = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r | \mathbf{s}, \mathbf{a})$ 


Rätt. 1 av 1 poäng.


- 15 Consider the Iterative Policy Evaluation in reinforcement learning (under a given MDP) to be performed as following, where v refers to state value, π is the policy, s refers to a state, r is a reward value, and a is an action.

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_{\pi}[R_{t+1} + X \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} Y [r + \gamma v_k(s')] \end{aligned}$$

Choose the correct statements about the missing parts X and Y.

[Number of correct options: 2]

$X = \gamma v_k(S_{t+1})$ 

$Y = p(s', r | s, a)$ 

$Y = p(r | s, a, s')$


$X = v_k(S_{t+1})$


Rätt. 1 av 1 poäng.

- 16 Choose the correct statements about Policy Improvement Theorem.

[Number of correct options: 2]

Let π and π' be any pair of deterministic policies such that for all $s \in \mathcal{S}$ we have: $q_{\pi}(s, \pi'(s)) \geq v_{\pi}(s)$, where q_{π} and v_{π} respectively correspond to state value and action value functions. Then, the policy π cannot be worse than π' .



There is always at least one optimal policy. 

Let π and π' be any pair of deterministic policies such that for all $s \in \mathcal{S}$ we have: $q_{\pi}(s, \pi'(s)) \geq v_{\pi}(s)$, where q_{π} and v_{π} respectively correspond to state value and action value functions. Then, the policy π' is as good as or better than π . 

It implies there is always exactly one optimal policy.

Rätt. 1 av 1 poäng.

17 Choose the correct statements about Monte Carlo (MC) methods in reinforcement learning.
[Number of correct options: 2]

- MC methods learn from experience. 
- MC methods need some prior knowledge of the actual environment's dynamics.
- Unlike Dynamic Programming, MC methods need an MDP.
- MC methods work based on averaging sample returns. 
- MC methods perform the policy evaluation similar to Dynamic Programming methods.

Rätt. 1 av 1 poäng.

18 Consider the following Monte Carlo (MC) prediction method for computing state values in reinforcement learning.

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π : $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$



Unless S_t appears in S_0, S_1, \dots, S_{t-1} :

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Choose the correct statements about this algorithm.

[Number of correct options: 2]

- The policy is updated to a greedy policy.
- It is a first-visit MC method. 
- The update target is $R_{t+1} + \gamma V(S_{t+1})$.
- It is an every-visit MC method.
- The policy is not updated during different episodes. 





Rätt. 1 av 1 poäng.

19 Consider the following reinforcement learning algorithm.

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
 Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$
 Loop for each episode:
 Initialize S
 Loop for each step of episode:
 Choose A from S using policy derived from Q (e.g., ε -greedy)
 Take action A , observe R, S'
 $Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$
 $S \leftarrow S'$
 until S is terminal

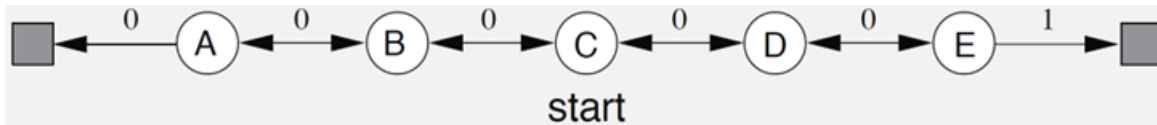
Choose the correct statement(s) about this algorithm.

[Number of correct options: 3]

- This algorithm is an on-policy method. 
- This algorithm is for control (policy improvement). 
- This algorithm is a model-free method. 
- This algorithm is a model-based method.
- This algorithm does not improve the policy.
- This algorithm is an off-policy method. 

Delvis rätt. 0 av 1 poäng.

- 20** Consider the following random walk problem, where the agent starts from the start state C and moves randomly to the right or to the left until it reaches one of the terminal states at the extreme left or right ends (shown by gray squares). The numbers indicate the rewards for the respective transitions. Assume the state values are initially set to zero and the discount factor is 1.



Consider the following episode: C, 0, D, 0, E, 1, Termination.

Choose the correct statements about the application of n -step Temporal Difference (n -step TD) to estimate state values from this episode.

[Number of correct options: 4]



- Any n -step method with $n > 5$, affects the values of all states A, B, C, D, E.
- Three-step TD, affects the values of C, D, and E. ✔
- Any n -step method with $n > 4$, affects the values of all states A, B, C, D, E.
- Any n -step method with $n > 2$, affects the values of C, D, E. ✔
- One-step TD affects only the estimate for the last state $V(E)$. ✔
- Two-step TD affects the values of the two states preceding termination: $V(D)$ and $V(E)$. ✔

Rätt. 1 av 1 poäng.

21 Consider this generic update scheme for state values in reinforcement learning $V(S_t) \leftarrow V(S_t) + \alpha[X - V(S_t)]$, where X represents the target variable for the update of the state value S_t .

Choose the correct statement(s).


[Number of correct options: 2]

- For TD we have $X = G_t$ where G_t is the return.
- For MC we have $X = G_t$ where G_t is the return. 
- For TD we have $X = R_{t+1} + \gamma V(S_t)$ where R_{t+1} is the reward at time $t + 1$ and γ is the discount factor.
- For MC we have $X = R_{t+1} + \gamma V(S_{t+1})$ where R_{t+1} is the reward at time $t + 1$ and γ is the discount factor.
- For TD we have $X = R_{t+1} + \gamma V(S_{t+1})$ where R_{t+1} is the reward at time $t + 1$ and γ is the discount factor. 
- For MC we have $X = R_{t+1} + \gamma V(S_t)$ where R_{t+1} is the reward at time $t + 1$ and γ is the discount factor.

Rätt. 1 av 1 poäng.

22 Which item specifies the advantages of TD (Temporal Difference) over MC (Monte Carlo) in reinforcement learning?



[Number of correct options: 1]

- TD, unlike MC, is naturally implemented in an online, fully incremental fashion. 
- TD, unlike MC, does not require a model of the environment.
- TD, unlike MC, learns from return samples.
- TD, unlike MC, does not require that the reward and next-state probability distributions are known.

Rätt. 1 av 1 poäng.

23 Choose the correct statements about double learning in reinforcement learning (in Temporal Difference methods).

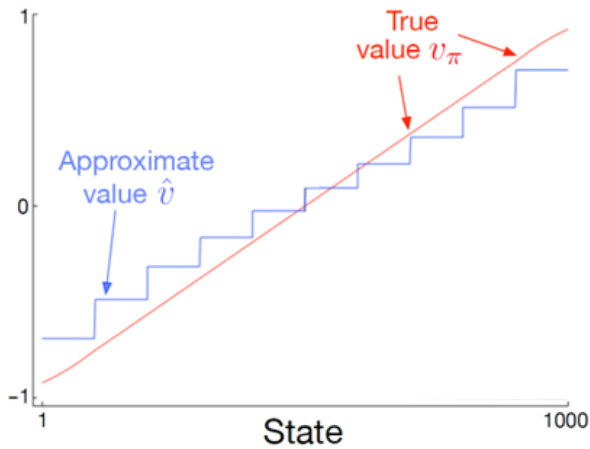
[Number of correct options: 2]

- It does not affect the amount of computation per step. 
- It does not affect the memory requirements.
- It separates the maximizing action from the estimate of its value. 
- Two estimates for each state-action pair are learned, and both of them are updated on each play.

Rätt. 1 av 1 poäng.

24 Consider function approximation for prediction in reinforcement learning, applied for state value estimation, i.e., $\hat{v}(\mathcal{S}_t, \mathbf{w})$, where \mathbf{w} corresponds to the parameters of the approximate function. We use the "state aggregation" method for $\hat{v}(\mathcal{S}_t, \mathbf{w})$ along with SGD (Stochastic Gradient Descent) to learn the parameters.

We apply this method to a task with 1000 states (e.g., the 1000-state Random Walk problem). We obtain the following results about the estimated (approximate) and true state values.



Choose the correct statements about this problem.

[Number of correct options: 2]

- The update target used here is biased. ✔
- The update target used here is unbiased.
- The update target used here is based on Temporal-Difference (TD). ✔
- The update target used here is based on Monte Carlo (MC).

Rätt. 1 av 1 poäng.

- 25 Choose the correct statement(s) about function approximation in reinforcement learning.
[Number of correct options: 2]

- The approximate function must always be implemented using a deep neural network.
- Compared to tabular reinforcement learning, it supports better the transfer of state values between similar states. ✓
- It is useful for huge state spaces including visual images. ✓
- It is only applicable when the model of the environment is known.

Rätt. 1 av 1 poäng.

- 26 Consider the following semi-gradient TD(0) algorithm for estimating the state values with function approximation (i.e., via $\hat{v}(S_t, \mathbf{w})$).

Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
 Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\hat{v}(\text{terminal}, \cdot) = 0$
 Algorithm parameter: step size $\alpha > 0$
 Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
 Initialize S
 Loop for each step of episode:
 Choose $A \sim \pi(\cdot | S)$
 Take action A , observe R, S'
 Update \mathbf{w}
 $S \leftarrow S'$
 until S is terminal



Choose the correct statements about this algorithm.

[Number of correct options: 2]

- In the update, the target is $R + \gamma \hat{v}(S', \mathbf{w})$. ✓
- In the update, the target is $R + \gamma \nabla \hat{v}(S', \mathbf{w})$. ✗
- It uses bootstrapping. ✓
- It learns the action values too.

Delvis rätt. 0 av 1 poäng.

27 Consider the DQN model designed for Atari games and choose the correct statements.
[Number of correct options: 2]




- It clips the TD error to be between -1 and +1. 
- It uses two completely independent neural networks instead of one.
- It clips the reward to be between -1 and +1.
- It uses experience replay. 

Rätt. 1 av 1 poäng.

28 Consider the episodic semi-gradient sarsa algorithm in reinforcement learning described as following.




Choose the correct statements about this algorithm.

[Number of correct options: 2]

- The approximate function is a linear function.
- This algorithm performs both prediction and control. 
- The control is based on a completely greedy policy improvement. 
- Selecting the policy based on ϵ -greedy leads to improving the policy within the family of ϵ -greedy policies. 
- For the update of the parameters, the algorithms needs to wait until the end of episodes.



Delvis rätt. 0 av 1 poäng.

29 Choose the correct statements about "experience replay" in reinforcement learning.
[Number of correct options: 3]

- It leads to learning multiple action values (for every state-action pair), instead of one.
- It helps to reduce variance and instability. 
- It helps to learn more efficiently from the experiences. 
- After an update, the next state always receives the next update.
- This technique makes sense for an off-policy reinforcement learning method. 

Rätt. 1 av 1 poäng.

30 Choose the correct statements about the Policy Gradient Theorem in policy-based reinforcement learning. Assume the policy parameters are θ and the respective performance measure (objective) is $J(\theta)$ defined as the state value of the initial state.
[Number of correct options: 2]

- It says that $\nabla J(\theta)$ does not involve the derivative of the state distribution. 
- It says that $\nabla J(\theta)$ does not involve the gradient of the policy function $\pi(a|s, \theta)$.
- It says that the performance measure $J(\theta)$ does not depend on any state.
- It provides a way to compute the gradient of performance measure $J(\theta)$ for performing gradient ascent. 

Rätt. 1 av 1 poäng.

31 Which of the following statements are correct about policy-based reinforcement learning methods.

[Number of correct options: 2]

- The action probabilities usually change dramatically.
- Obtained knowledge generalizes well.
- They are not good for continuous action spaces.
- The action probabilities usually change smoothly.
- They can learn stochastic policies.



Delvis rätt. 0 av 1 poäng.

32 Consider the one-step actor–critic method in reinforcement learning. Choose the correct statements about that.

[Number of correct options: 2]

- The critic is based on state value function.
- The critic is based on action value function.
- It uses the same update target as the REINFORCE algorithm.
- The actor is responsible for learning the policy.
- The policy improvement is based on ϵ -greedy.



Delvis rätt. 0 av 1 poäng.

33 Consider the following n-step Sarsa for estimating action values (control).

```




Initialize  $Q(s, a)$  arbitrarily, for all  $s \in \mathcal{S}, a \in \mathcal{A}$ 
Initialize  $\pi$  to be  $\varepsilon$ -greedy with respect to  $Q$ , or to a fixed given policy
Algorithm parameters: step size  $\alpha \in (0, 1]$ , small  $\varepsilon > 0$ , a positive integer  $n$ 
All store and access operations (for  $S_t, A_t$ , and  $R_t$ ) can take their index mod  $n + 1$ 

Loop for each episode:
  Initialize and store  $S_0 \neq$  terminal
  Select and store an action  $A_0 \sim \pi(\cdot|S_0)$ 
   $T \leftarrow \infty$ 
  Loop for  $t = 0, 1, 2, \dots$  :
    | If  $t < T$ , then:
    |   Take action  $A_t$ 
    |   Observe and store the next reward as  $R_{t+1}$  and the next state as  $S_{t+1}$ 
    |   If  $S_{t+1}$  is terminal, then:
    |      $T \leftarrow t + 1$ 
    |   else:
    |     Select and store an action  $A_{t+1} \sim \pi(\cdot|S_{t+1})$ 
    |    $\tau \leftarrow t - n + 1$  ( $\tau$  is the time whose estimate is being updated)
    |   If  $\tau \geq 0$ :
    |     X
    |     If  $\tau + n < T$ , then  $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$  ( $G_{\tau:\tau+n}$ )
    |     Y
    |     If  $\pi$  is being learned, then ensure that  $\pi(\cdot|S_\tau)$  is  $\varepsilon$ -greedy wrt  $Q$ 
  Until  $\tau = T - 1$ 

```

Choose the correct statements about the missing parts X and Y.

[Number of correct options: 2]

- $Y : Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha[G - Q(S_\tau, A_\tau)]$ 
- $X : G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_i$ 
- $X : G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_{i-\tau-1}$
- $X : G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^i R_i$ 
- $Y : Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha G$
- $Y : Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha[G - \max_a Q(S_\tau, a)]$

Delvis rätt. 0 av 2 poäng.

- 34 Consider an episodic reinforcement learning problem where we apply Temporal Difference (TD). As you know, the TD error is defined as: $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ where R_{t+1} is the reward at time $t + 1$, γ is the discount factor, and $V(S_t)$ is the state value function for state S_t . Assume that $V(\cdot)$ does not change during the episode. Then, fill in the missing parts X and Y in the following expression (note that G_t is the return at time t).

$$G_t = V(S_t) + \delta_t + \gamma X + \gamma^2(Y - V(S_{t+2})).$$

[Number correct options: 1]

- $X = G_{t+1}, Y = \delta_{t+1}$
- $X = V(S_{t+1}), Y = G_{t+2}$
- $X = \delta_t, Y = G_t$
- $X = \delta_{t+1}, Y = G_{t+2}$
- $X = G_{t+1}, Y = V(S_{t+1})$



Fel. 0 av 2 poäng.