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

Choose the correct statements about the reward for a chosen arm in stochastic multi armed bandit problems.
[Number of correct options: 2]

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

Rätt. 1 av 1 poäng.

2 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. ✓
- ☐ Chooses the arm with the highest upper confidence bound according to the posterior distribution over mean rewards.
- ☐ 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. ✓

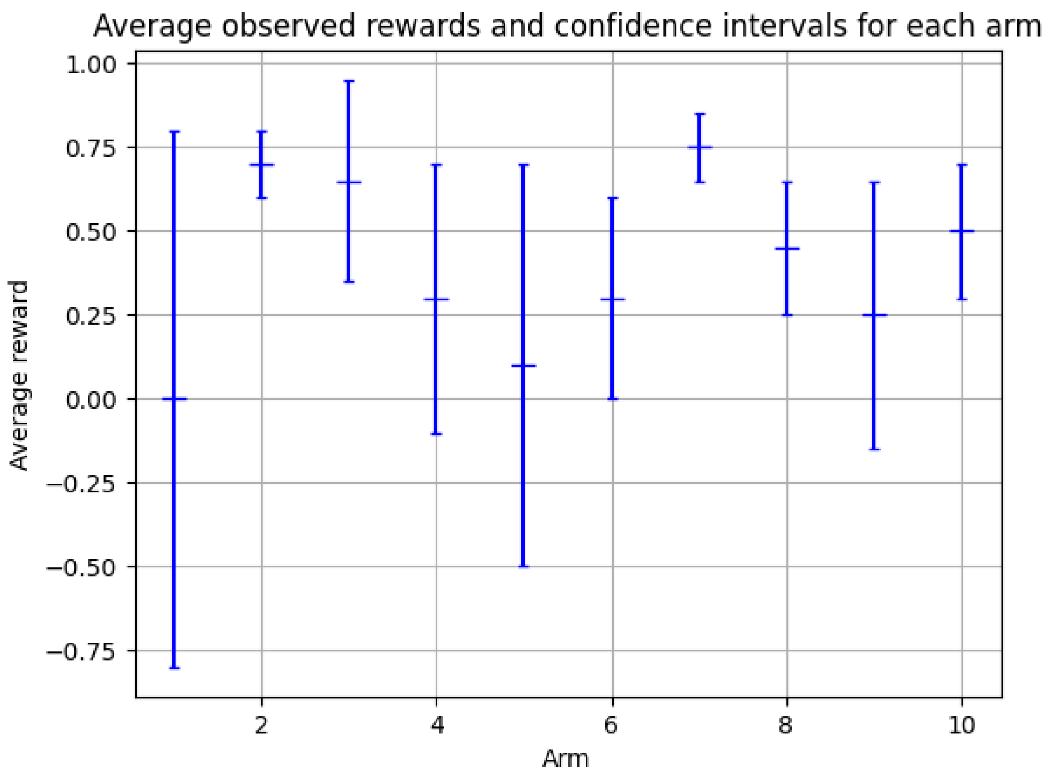
Rätt. 1 av 1 poäng.

3 Choose the correct statements about Epsilon-Greedy (ϵ-greedy) in bandit problems.
[Number of correct options: 2]

- [✓] The exploration does not depend on the history of observed rewards. ✓
- [ ] Epsilon-Greedy performs better than Greedy when the rewards are deterministic (fixed) but unknwn.
- [ ] Compared to Greedy, Epsilon-Greedy becomes less effective when the rewards are noisy.
- [✓] The exploration is uniform over arms. ✓

Rätt. 1 av 1 poäng.

4



Consider a stochastic bandit problem with a set of arms $\mathcal{A}$, and $|\mathcal{A}| = 10$. For an arbitrary round, the figure displays the average reward $\bar{\mu}(a)$ up to this round for every arm $a \in \mathcal{A}$ and corresponding confidence intervals. Which arm(s) will algorithm UCB1 and Greedy choose in this round?

Algorithm UCB1 will choose [Arm 3] ✓ (Arm 2, Arm 9, Arm 8, Arm 10, Arm 3, Arm 7, Arm 1, Arm 6, Arm 4, Arm 5).

Greedy will choose [Arm 7] ✓ (Arm 5, Arm 9, Arm 3, Arm 10, Arm 2, Arm 6, Arm 8, Arm 4, Arm 1, Arm 7).

Rätt. 1 av 1 poäng.

5 Consider the following game that proceeds over n rounds: In each round $t \in \{1, \dots, n\}$, you choose either to play or do nothing. If you do nothing, then your reward is $X_t = 0$. If you play, then your reward is $X_t = 1$ with probability p and $X_t = -1$ otherwise.

In terms of regret, what is the optimal way of choosing actions, i.e., the best algorithm, when p is known?

[Number of correct options: 1]

- Always choose to play.
- Always choose to do nothing.
- [✓ If $p > 0.5$, choose to play. Otherwise, choose to do nothing.] ✓
- If $p > 0.5$, choose to do nothing. Otherwise, choose to play.

Rätt. 1 av 1 poäng.

6

```
Exploration phase: try each arm $N$ times;
Select the arm $\hat{a}$ with the highest average reward (break ties arbitrarily);
Exploitation phase: play arm $\hat{a}$ in all remaining rounds.
```

Algorithm 1.1: Explore-First with parameter N.

Consider a stochastic K-armed bandit problem with a set of arms $\mathcal{A}$, and $|\mathcal{A}| = K$. Assume rewards are bounded in $[0, 1]$ and a total number of rounds T. Choose the correct statements about the Explore-First algorithm in bandit problems.

[Number of correct options: 2]

- If $K \gg T$, then the bad event can be always neglected.
- [✓ The cumulative regret in the exploration phase is upper bounded proportional to the number of arms K and the number of times N that each arm is tried in the exploration phase.] ✓
- [✓ Hoeffding's inequality can be used to define the clean event of the exploration phase.] ✓
- The regret is constant in the exploitation phase.

Rätt. 1 av 1 poäng.

7 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.

8 Consider the following game that proceeds over n rounds: In each round $t \in \{1, \ldots, n\}$, you choose either to play or do nothing. If you do nothing, then your reward is $X_t = 0$. If you play, then your reward is $X_t = 1$ with probability p and $X_t = -1$ otherwise.
Assume that you use an algorithm that chooses to play with probability q and otherwise chooses to do nothing.
What is the expected regret of this algorithm?
[Number of correct options: 2]

- $p > 0.5 \Rightarrow \mathbb{E}[R(n)] = np(1-q)$
- $p < 0.5 \Rightarrow \mathbb{E}[R(n)] = -q(2p-1)$
- [✓ $p > 0.5 \Rightarrow \mathbb{E}[R(n)] = n(1-q)(2p-1)$] ✓
- [✓ $p < 0.5 \Rightarrow \mathbb{E}[R(n)] = -nq(2p-1)$] ✓
- $p < 0.5 \Rightarrow \mathbb{E}[R(n)] = pqn$
- $p > 0.5 \Rightarrow \mathbb{E}[R(n)] = n(2p-1)$

Rätt. 2 av 2 poäng.

9 Table 1: Number of samples $n_t(a)$ and total observed reward (return) $s_t(a)$ of each a arm before round t.

a	1	2	3	4	5	6	7	8	9	10
$n_t(a)$	10	52	24	17	10	18	24	21	14	109
$s_t(a)$	0	36	10	5	0	6	10	8	3	92
$x_t(a)$	1	1	1	1	0	0	1	0	0	1

Consider a stochastic K-armed bandit problem with a set of arms $\mathcal{A}$, and $|\mathcal{A}| = K$. We define, at round t: $a_t \in \mathcal{A}$ is the arm played by the agent, $x_t(a) \in \{0, 1\}$ is the reward received by an agent if it plays arm $a \in \mathcal{A}$, $s_t(a) = \sum_{j=1}^{t-1} \mathbf{1}\{a_j = a\} x_j(a)$ is the cumulative reward of arm a before round t, and $n_t(a) = \sum_{j=1}^{t-1} \mathbf{1}\{a_j = a\}$ is the number of rounds that arm a has been played before round t. Consider the scenario in Table 1, where an agent has played in a multi-armed bandit environment with $K = 10$ arms up to round t. Note that the reward $x_t(a)$ is revealed to the agent *if and only if* arm $a \in \mathcal{A}$ is played in round t. With fixing the time horizon $T = 1000$, which arm will be chosen in rounds t and $t + 1$ by the UCB1 algorithm with confidence radius $r_t(a) = \sqrt{\frac{2 \log_e T}{n_t(a)}}$?

Round t: Arm 6 ✓ (Arm 1, Arm 2, Arm 3, Arm 4, Arm 5, Arm 6, Arm 7, Arm 8, Arm 9, Arm 10)

Round $t + 1$: Arm 9 ✗ (Arm 1, Arm 2, Arm 3, Arm 4, Arm 5, Arm 6, Arm 7, Arm 8, Arm 9, Arm 10)

Delvis rätt. 1 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]

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

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

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

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

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

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

Fel. 0 av 3 poäng.

11 Consider a Markov decision process for reinforcement learning. Choose the correct statements about the expected reward $r(s, a, s') \doteq \mathbb{E}[R_t | S_{t-1} = s, A_{t-1} = a, S_t = s']$ for state–action–next–state triples (s, a, s').
[Number of correct options: 2]

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

☑ $r(s, a, s') = \sum_{r\in\mathcal{R}} r p(r | s, a, s')$ ✓

☐ $r(s, a, s') = \sum_{r\in\mathcal{R}} r \frac{p(s'|s,a)}{p(s',r|s,a)}$

☑ $r(s, a, s') = \sum_{r\in\mathcal{R}} r \frac{p(s',r|s,a)}{p(s'|s,a)}$ ✓

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

Rätt. 1 av 1 poäng.

13 What is the concept of "reward hypothesis" in reinforcement learning?
[Number of correct options: 2]

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

Rätt. 1 av 1 poäng.

14 Choose the correct statements about state values $v_\pi(s)$ for state s under policy π in reinforcement learning.
In these equations, a refers to an action, s refers to a state, r is a reward value, γ is the discount factor, and G_t is the return.
[Number of correct options: 3]

- [ ] $v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma G_t | S_t = s]$
- [x] $v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s]$ ✓
- [ ] $v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma G_t]$
- [x] $v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma v_\pi(s')]$ ✓
- [x] $v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$ ✓

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.

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

Choose the correct statements about the missing parts X and Y.
[Number of correct options: 2]

- [✓] $Y = p(s', r|s, a)$ ✓
- [ ] $Y = p(r|s, a, s')$
- [✓] $X = \gamma v_k(S_{t+1})$ ✓
- [ ] $X = v_k(S_{t+1})$

Rätt. 1 av 1 poäng.

16 Choose the correct statements about Dynamic Programming (DP) in reinforcement learning.
[Number of correct options: 3]

- [✓] It is based on a perfect model of the environment as a Markov decision process (MDP). ✓
- [✓] It uses bootstrapping. ✓
- [ ] It is a model-free method.
- [ ] It learns from real experiences (interactions between the agent and environment).
- [✓] It may have great computation expense. ✓
- [ ] It always assumes the discount factor is 1.

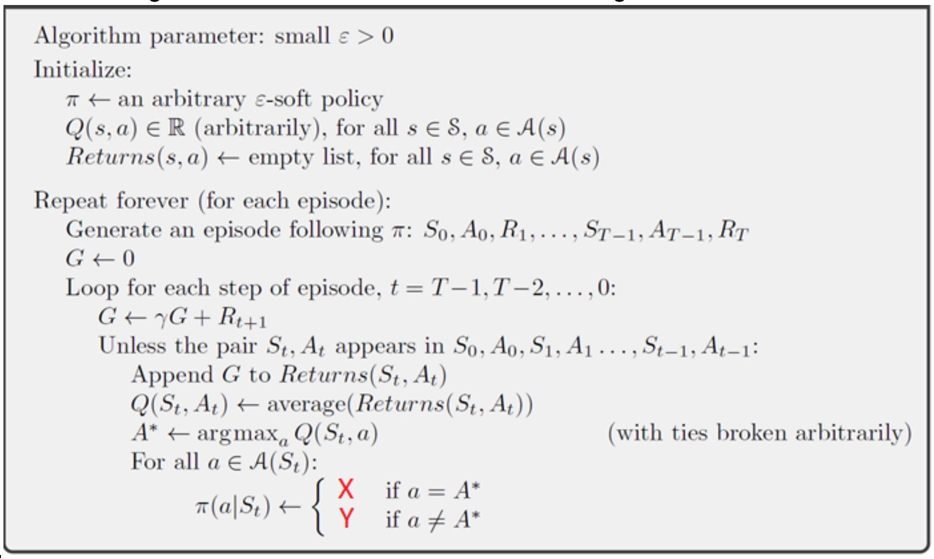
Rätt. 1 av 1 poäng.

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

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

Rätt. 1 av 1 poäng.

18 Consider the following control method in reinforcement learning. Choose the correct statements about this



```
Algorithm parameter: small $\varepsilon > 0$
Initialize:
    $\pi \leftarrow$ an arbitrary $\varepsilon$-soft policy
    $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
    $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):
    Generate an episode following $\pi$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        Unless the pair $S_t, A_t$ appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
            Append $G$ to $Returns(S_t, A_t)$
            $Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$
            $A^* \leftarrow \operatorname{argmax}_a Q(S_t, a)$          (with ties broken arbitrarily)
            For all $a \in \mathcal{A}(S_t)$:
                $\pi(a|S_t) \leftarrow \begin{cases} X & \text{if } a = A^* \\ Y & \text{if } a \neq A^* \end{cases}$
```

method.
[Number of correct options: 3]

- ☑ $X = 1 - \epsilon + \epsilon/|\mathcal{A}(S_t)|$ ✓
- ☐ It is based on Temporal Difference (TD) target.
- ☐ $X = \epsilon/|\mathcal{A}(S_t)|$
- ☐ $Y = 1 - \epsilon + \epsilon/|\mathcal{A}(S_t)|$
- ☑ It is based on Monte Carlo (MC) target. ✓
- ☑ $Y = \epsilon/|\mathcal{A}(S_t)|$ ✓

Rätt. 1 av 1 poäng.

19 Consider n-step Expected Sarsa. Which option specifies its target update for state values?
R_{t+1} corresponds to reward at time $t+1$, γ is the discount factor, π is the policy, S_t is the state at time t, Q is the action value function, and a is an action.
[Number of correct options: 1]

- ☐ $R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum_a \pi(a|S_t) Q_t(S_t, a).$
- ☐ $R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum_a \pi(a|S_{t+n}) Q_t(S_{t+n}, a).$
- ☐ $R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n-1} + \gamma^n \sum_a \pi(a|S_t) Q_t(S_t, a).$
- ☐ $R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum_a \pi(a|S_t) Q_{t+n-1}(S_t, a).$
- [✓] $R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum_a \pi(a|S_{t+n}) Q_{t+n-1}(S_{t+n}, a).$ ✓

Rätt. 1 av 1 poäng.

20 Consider the backup diagram for n-step TD (n-step Temporal Difference) prediction.
How many actions do appear in the backup diagram?
[Number of correct options: 1]

- ☐ 2
- ☐ n+2
- ☐ n+1
- ☐ n-2
- [✓] n ✓
- ☐ 1
- ☐ n-1

Rätt. 1 av 1 poäng.

21 Consider the backup diagram for n-step TD (n-step Temporal Difference) Sarsa.
How many states do appear in the backup diagram?
[Number of correct options: 1]

- ☐ n ✓
- ☐ 2
- ☑ n+1 ✗
- ☐ n-1
- ☐ 1

Fel. 0 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, does not require that the reward and next-state probability distributions are known.
- ☑ TD, unlike MC, is naturally implemented in an online, fully incremental fashion. ✓
- ☐ TD, unlike MC, learns from return samples.
- ☐ TD, unlike MC, does not require a model of the environment.

Rätt. 1 av 1 poäng.

23 Consider the following Bellman equation in reinforcement learning, where G_t is the return at time t, γ is the discount factor, R_{t+1} corresponds to reward, and $v_\pi(s)$ is the value of state s under policy π.

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \end{aligned}$$

Choose the correct statements.
[Number of correct options: 2]

- [✓] DP uses an estimate of the third line (in the equation) as a target. ✓
- [ ] MC uses a sample of the second line (in the equation) as a target.
- [ ] DP uses the expectation in the second line (in the equation) as a target.
- [ ] DP uses the expectation in the first line (in the equation) as a target.
- [ ] MC uses the expectation in the first line (in the equation) as a target.
- [✓] MC uses a sample of the first line (in the equation) as a target. ✓

Rätt. 1 av 1 poäng.

24 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 $\pi$ to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \to \mathbb{R}$ 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]

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

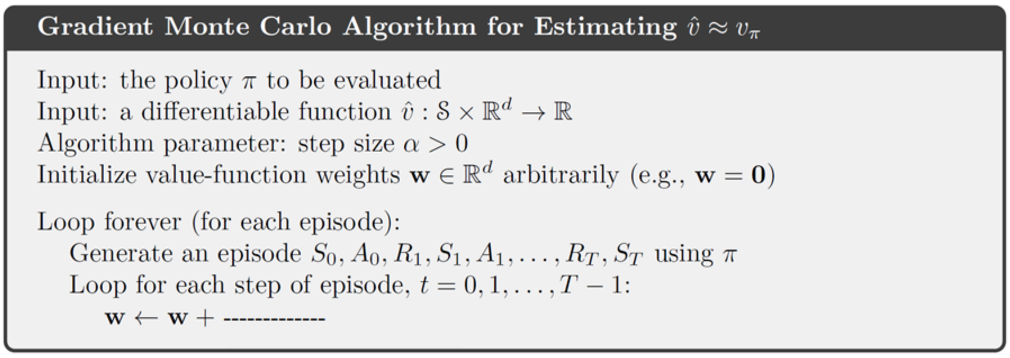
Rätt. 1 av 1 poäng.

25 Consider function approximation for prediction in reinforcement learning, applied for state value estimation, i.e., $\hat{v}(S_t, \mathbf{w})$, where $\mathbf{w}$ corresponds to the parameters of the approximate function.
We use SGD (Stochastic Gradient Descent) to learn the parameters. We assume the update target used here is the bootstrapping target, i.e., it is based on the ordinary TD (Temporal Difference) target.
Choose the correct statements about this problem.
[Number of correct options: 2]

- [ ] The target is unbiased.
- [✓] The target is biased. ✓
- [✓] This choice of target provides continual/online learning. ✓
- [ ] To compute the target, the agent needs to wait until the end of the episode.

Rätt. 1 av 1 poäng.

26 Consider the following Gradient Monte Carlo algorithm for estimating the state values with function approximation (i.e., via $\hat{v}(S_t, \mathbf{w})$).



```
Gradient Monte Carlo Algorithm for Estimating $\hat{v} \approx v_\pi$

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

Loop forever (for each episode):
    Generate an episode $S_0, A_0, R_1, S_1, A_1, \dots, R_T, S_T$ using $\pi$
    Loop for each step of episode, $t = 0, 1, \dots, T-1$:
        $\mathbf{w} \leftarrow \mathbf{w} +$ -------------
```

Choose the correct statements.
[Number of correct options: 2]

- [ ] The missing part is $\alpha[G_t - \hat{v}(S_t, \mathbf{w})]\hat{v}(S_t, \mathbf{w})$
- [ ] The missing part is $\alpha[G_t - \nabla\hat{v}(S_t, \mathbf{w})]\hat{v}(S_t, \mathbf{w})$
- [✓] The missing part is $\alpha[G_t - \hat{v}(S_t, \mathbf{w})]\nabla\hat{v}(S_t, \mathbf{w})$ ✓
- [ ] This is a first-visit method.
- [✓] This is an every-visit method. ✓

Rätt. 1 av 1 poäng.

27 Consider the following DQN algorithm and choose the correct statements about that.

```
Initialize network $\hat{q}$
Initialize target network $\tilde{q}$
Initialize experience replay memory $D$
Initialize the $Agent$ to interact with the Environment
while $not\ converged$ do
    /* Sample phase
    $\epsilon \leftarrow$ setting new epsilon with $\epsilon$-decay
    Choose an action $a$ from state $s$ using policy $\epsilon$-greedy($\hat{q}$)
    $Agent$ takes action $a$, observe reward $r$, and next state $s'$
    Store transition $(s, a, r, s', done)$ in the experience replay memory $D$

    if $enough\ experiences\ in\ D$ then
        /* Learn phase
        Sample a random $minibatch$ of $N$ transitions from $D$
        for $every\ transition\ (s_i, a_i, r_i, s'_i, done_i)\ in\ minibatch$ do
            if $done_i$ then
                $y_i = r_i$
            else
                $y_i = r_i + \gamma \max_{a' \in A} \tilde{q}(s'_i, a')$
            end
        end
        Calculate the loss $\mathcal{L} = 1/N \sum_{i=0}^{N-1} (\hat{q}(s_i, a_i) - y_i)^2$
        Update $\hat{q}$ using the SGD algorithm by minimizing the loss $\mathcal{L}$
        Every $C$ steps, copy weights from $\hat{q}$ to $\tilde{q}$
    end
end
```

[Number of correct options: 2]

- [ ] When $done_i$ is *true* then it does not use experience replay.
- [ ] It uses the MC (Monte Carlo) returns as update target.
- [✓] It uses the TD (Temporal Difference) as update target. ✓
- [ ] It does not use experience replay.
- [✓] The target network is used as the duplicate network. ✓

Rätt. 1 av 1 poäng.

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

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

Delvis rätt. 0 av 1 poäng.

29 Choose the correct statements about duplicate network used in DQN.
[Number of correct options: 2]

- [ ] It makes the target for a Q-learning update depend on the most recent update of the neural network.
- [ ] It requires training two neural networks independently in parallel, but using the same data.
- [✓] The use of duplicate network can avoid oscillations or divergence. ✓
- [✓] With this trick, the neural network used in the target (the target of Q-learning) is updated less frequ✓y.

Rätt. 1 av 1 poäng.

30 Choose the correct statements about value-based and policy-based methods in reinforcement learning.
[Number of correct options: 2]

- [✓] ϵ-greedy is usually related to value-based methods. ✓
- [ ] ϵ-greedy is commonly used in Actor-Critic methods.
- [ ] ϵ-greedy is commonly used in both policy-based and value-based methods.
- [✓] Actor-Critic methods learn both policy and value functions. ✓
- [ ] ϵ-greedy is usually related to policy-based methods.

Rätt. 1 av 1 poäng.

31 Choose the correct statements about policy gradient methods in reinforcement learning. Assume the parameters of the policy objective (performance measure) J are specified by $\boldsymbol{\theta}$.
[Number of correct options: 2]

☐ The objective (performance measure) $J(\theta)$ cannot depend on state value of any state under the investigated policy.

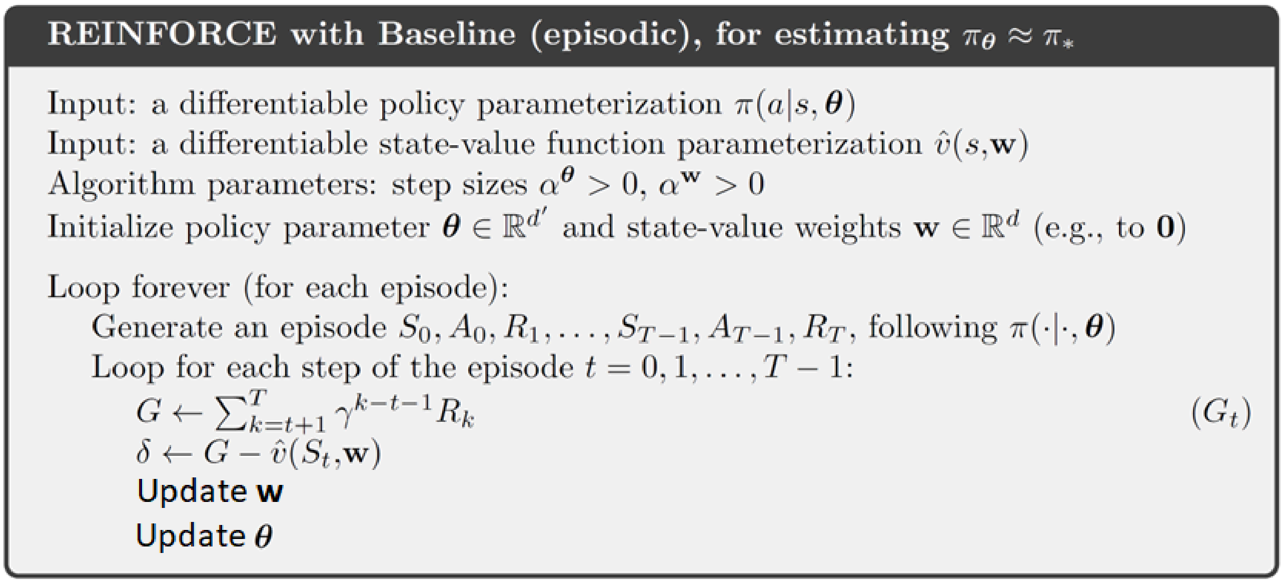
☐ The parameters can be updated as: $\theta_{t+1} = \theta_t - \alpha \nabla J(\theta)$, where $J(\theta)$ is the objective function (performance measure).

☑ The parameters can be updated as: $\theta_{t+1} = \theta_t + \alpha \nabla J(\theta)$, where $J(\theta)$ is the objective function (performance measure). ✓

☑ The actions selected at different steps affect the update of the parameters $\boldsymbol{\theta}$. ✓

Rätt. 1 av 1 poäng.

32 Consider the REINFORCE method with baseline, described below.



```
REINFORCE with Baseline (episodic), for estimating $\pi_\theta \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Algorithm parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
    Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
    Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
        $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$                    $(G_t)$
        $\delta \leftarrow G - \hat{v}(S_t,\mathbf{w})$
        Update $\mathbf{w}$
        Update $\boldsymbol{\theta}$
```

Choose the correct statement about this algorithm.
[Number of correct options: 1]

☐ The baseline term is δ.

☐ The baseline must be constant and fixed for all episodes.

☐ The policy is based on a greedy or ϵ-greedy policy.

☑ The baseline term is $\hat{v}(S_t, \mathbf{w})$. ✓

Rätt. 1 av 1 poäng.

33 Consider the following Policy Gradient Theorem in policy-based reinforcement learning. Assume the policy parameters are θ and the respective performance measure (objective) is $J(\theta)$ which is defined as the value of the start state, i.e., $J(\theta) = v_{\pi_\theta}(s_0)$.

$$\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi\left[\sum_a \pi(a|S_t,\boldsymbol{\theta})q_\pi(S_t,a)\frac{\nabla\pi(a|S_t,\boldsymbol{\theta})}{\pi(a|S_t,\boldsymbol{\theta})}\right]$$
$$= \mathbb{E}_\pi\left[\text{ X } \nabla \text{Y} \right]$$

Choose the correct expressions about **X** and **Y**.
[Number of correct options: 3]

- [ ] $X = \pi(A_t|S_t,\theta)$
- [✓] $X = q_\pi(S_t, A_t)$, where q_π is the action value function under policy π. ✓
- [ ] $Y = G_t$ where G_t is the return.
- [✓] $X = G_t$ where G_t is the return. ✓
- [✓] $Y = \ln \pi(A_t|S_t,\theta)$ ✓
- [ ] $Y = \pi(A_t|S_t,\theta)$

Rätt. 2 av 2 poäng.

34 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]

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

Rätt. 2 av 2 poäng.