Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

1. Which of the following meta-parameters can be tuned to improve performance of the agent? Performance refers to the cumulative reward the agent would receive *in expectation* across different runs. (Select all that apply)

1 / 1 point

- [x] The step size in the update rule of the learning algorithm (e.g., alpha in Q-learning)

Correct
Correct. If the step size is too low, learning might be very slow. But if it is too high, there might be a lot of variance in the learning behaviour.

- [ ] Random seed (for the random number generator)
- [x] Exploration parameter (e.g., epsilon in e-greedy or the temperature tau in the softmax policy)

Correct
Correct. We have to try different levels of exploration that the agent begins with, because different problems may require different extents of exploration. We do not know this beforehand.

- [x] Number of hidden-layer units in a neural network approximating the value function

Correct
Correct. If the number of hidden units is too small, the representational capacity may be insufficient for learning good behavioural policies. On the other hand, a large number of hidden units could help to learn a good representation, but learning progress might be very slow due to the sheer number of parameters.

2. Suppose a problem that you have formulated as an MDP has k continuous input dimensions. You are considering using tile coding as a function approximator. With T tilings and t tiles per dimension in each tiling, which of the following represent the resultant number of features? (Assume each tiling covers all k dimensions.)

1 / 1 point

- ○ T · t / k
- ○ k·T^t
- ○ T · t · k
- ◉ T·t^k

Correct
Correct. The number of features for a single tiling are t^k, and there are T such tilings, resulting in T t^k features in total.

3. Which of the following statements regarding feature-construction methods are TRUE? (Select all that apply)

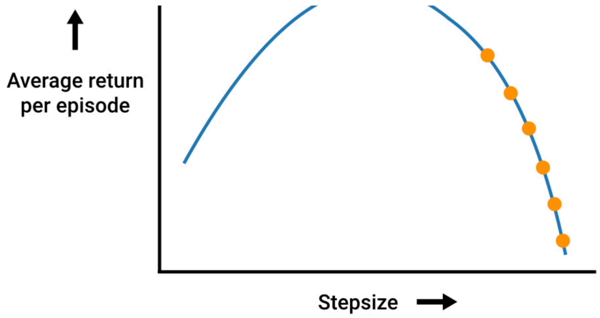
1 / 1 point

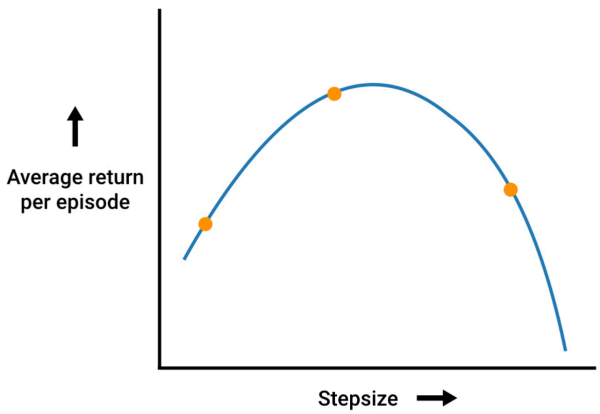
- [x] A simple implementation of tile coding leads to memory requirements that might be exponential in the number of features.

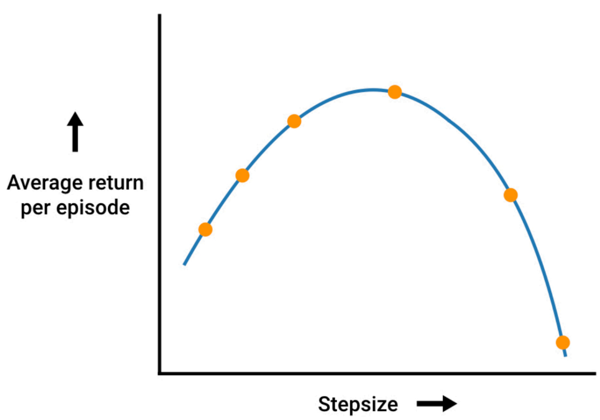
Correct
Correct. But through methods like hashing, the memory requirements can often be reduced by large factors with little loss of performance. Check out Section 9.5.4 of Sutton and Barto's textbook for a discussion on this.

- [x] In low-dimensional problems, tile coding is computationally efficient and provides good generalization and discrimination.

Correct
Correct. Recall the Tile coding lecture from Course 3. Tile coding is computationally efficient: with the use of binary feature vectors in tile coding, the weighted sum of features that make up the approximate value function is trivial to compute. For d number of features, one simply computes the indices of the n<<d active features and then adds up the n corresponding components of the weight vector. However, as the number of dimensions grows, the number of required tiles grows exponentially, and neural networks might be choice of function approximator.



- [ ] 

- [x] 

Correct
Correct. We should test a sufficient range and number of values for every meta-parameter to increase the likelihood of finding the best setting of meta-parameters for our algorithms.

12. True or False: Epsilon-greedy exploration uses information from all the action values of a particular state when choosing a *non-greedy* action in that state.

1 / 1 point

- (●) False
- ( ) True

Correct
Correct. When a non-greedy action is to be picked, epsilon-greedy disregards all the action values and picks one of the actions randomly. On the other hand, the probability of picking an action with a softmax operator is proportional to the (exponentiated) value of that action.