Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
100%

Impact of Parameter Choices in RL

LATEST SUBMISSION GRADE

100%

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] 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.

- [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] 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)

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
- (•) T·t^k
- ( ) k·T^t
- ( ) 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] The feature representation obtained using neural networks changes with time.

Correct
Correct. The weights of a neural network change during training, and hence the feature representation changes with time.

- [ ] The feature representation obtained using tile coding changes with time.
- [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.

4. True or False: Adding more hidden layers (of a fixed finite width) increases the representation capacity of neural network. For example, if you have a single-hidden layer neural network with 16 units and nonlinear activations, then adding another layer of 16 units to get a neural network with two hidden layers can represent more functions.

1 / 1 point

- (•) True
- ( ) False

Correct
Correct. With more hidden layers and nonlinear activation functions, the neural network can represent a larger class of nonlinear functions.

5. True or False: Adding more hidden layers to a neural network increases the number of parameters needed to be learned.

1 / 1 point

- (•) True
- ( ) False

Correct
Correct. More hidden layers leads to more parameters, which take more samples to train/learn

6. Which of the following statements regarding the exploration approach are TRUE? (Select all that apply)

1 / 1 point

- [ ] Both optimistic initial values and epsilon-greedy exploration can be easily used with neural networks, because they are simple exploration strategies.
- [ ] A softmax policy is a limited strategy for exploration because it can only be used with action preferences and policy-gradient methods.
- [x] Optimistic initial values are difficult to maintain when using neural networks as a function approximator.

Correct
Correct. This is because changing one weight of a neural network affects the values of many state-action pairs. This makes it hard to maintain optimistic values for all of the state-actions pairs that haven't been tried yet.

- [ ] Epsilon-greedy exploration is difficult to combine with neural networks.

7. Which of the following are TRUE about the softmax temperature parameter tau?

1 / 1 point

- [x] If tau is large, the agent's policy is more stochastic.

Correct
Correct. For large tau, the policy is nearly uniformly random. Such a policy is more stochastic as compared to the near-greedy deterministic policies when tau is small.

- [x] For very large tau, the agent's policy is nearly a uniformly-random policy.

Correct
Correct. For large tau, the differences between the action preferences/values become negligible. The resulting policy is nearly uniformly random.

- [ ] Tau does not affect the exploration at all.
- [x] For very small tau, the agent mostly selects the greedy action.

Correct
Correct. For small tau, the differences between the action preferences/values get exaggerated. As a result, the greedy action is picked more often.

8. Which of the following statements are true about activation functions? (Select all that apply)

1 / 1 point

- [ ] Rectified Linear Units (ReLUs) are linear activation functions.
- [x] The gradient of flat regions in the range of an activation function w.r.t. the input is zero.

Correct
Correct. This follows from basic calculus and is the root cause of vanishing gradients in sigmoidal activation functions.

- [ ] Linear activation functions (such as f(x)=x) have derivatives close to zero for inputs of large magnitude.
- [x] For inputs of large magnitude, the derivative of the sigmoid and tanh functions are close to zero.

Correct
Correct. The gradient of these S-shaped functions w.r.t. the input is non-zero only in a small range around 0. For inputs of large magnitude, sigmoid and tanh activation functions 'saturate'.

9. Consider you are using a neural network to approximate the action-value function of a reinforcement learning agent. You decide to use a neural network with two hidden layers. Now you want to choose the activation function for the hidden layers and the output layer. One option is to use a neural network with tanh activation functions in both hidden layers, and a linear activation in the output layer. Another option is to use a neural network with linear activations in both the hidden layers and the output layer.

True or False: In both cases (option one and option two), the neural network can represent the same class of action-value functions.

1 / 1 point

- ( ) True
- (•) False

Correct
Correct. A network comprised solely of linear activation functions can only represent linear functions. On the other hand, a network comprised of a combination of linear and nonlinear activation functions can represent some nonlinear functions.

10. Which of the following statements are TRUE regarding methods for selecting a stepsize for the learning update?

1 / 1 point

- [ ] A stepsize that reduces over time (such as 1/N, where N is the number of agent-environment interactions) is necessary when the environment changes over time.
- [x] An adaptive stepsize selection method like RMSProp uses a heuristic to change the stepsize during learning.

Correct
Correct. An adaptive stepsize selection method changes the stepsize during learning using some heuristic. For example, RMSProp does so based on the average recent magnitudes of the gradients.

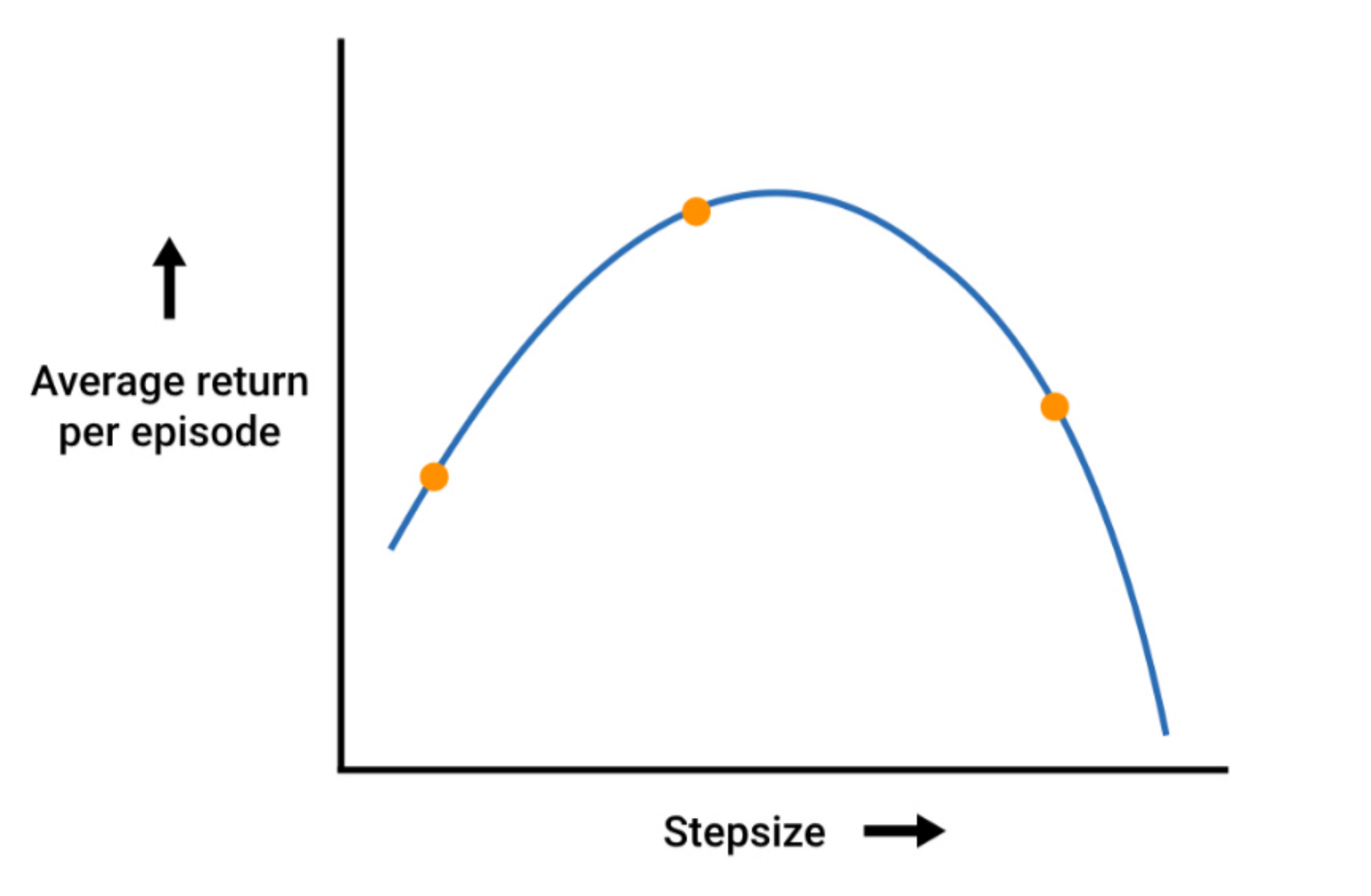
- [x] The heuristic to change the stepsize can be learned from the data collected from the agent-environment interactions.

Correct
Correct. Meta-learning techniques do not use a fixed heuristic to change the stepsize based on the data but learn the heuristic itself.

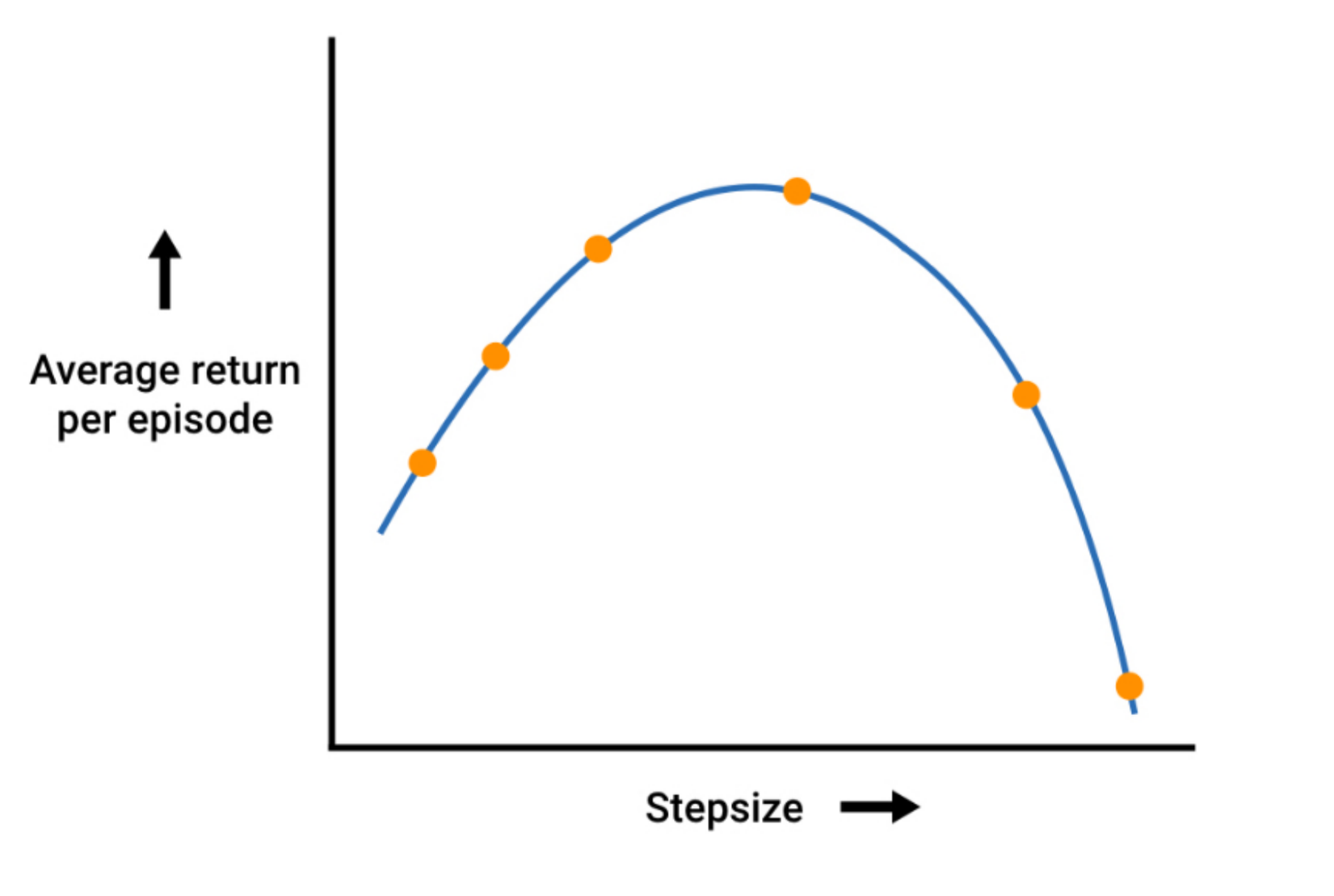
11. Suppose we want to find the optimal policy that obtains the maximum undiscounted return per episode in some task. We are using Expected Sarsa. With the rest of the meta-parameters fixed, we want to find the best setting of the stepsize that results in the best performance in this setting. In the following graph, the blue line represents how the performance measure varies with stepsize. Obviously, we do not have this information beforehand, and we are selecting a range of stepsizes to try out with our agent. Which of the following graphs best represent the range of stepsizes that should be tried out for a given experiment? (the orange points represent the selected stepsizes)

1 / 1 point

- [ ]

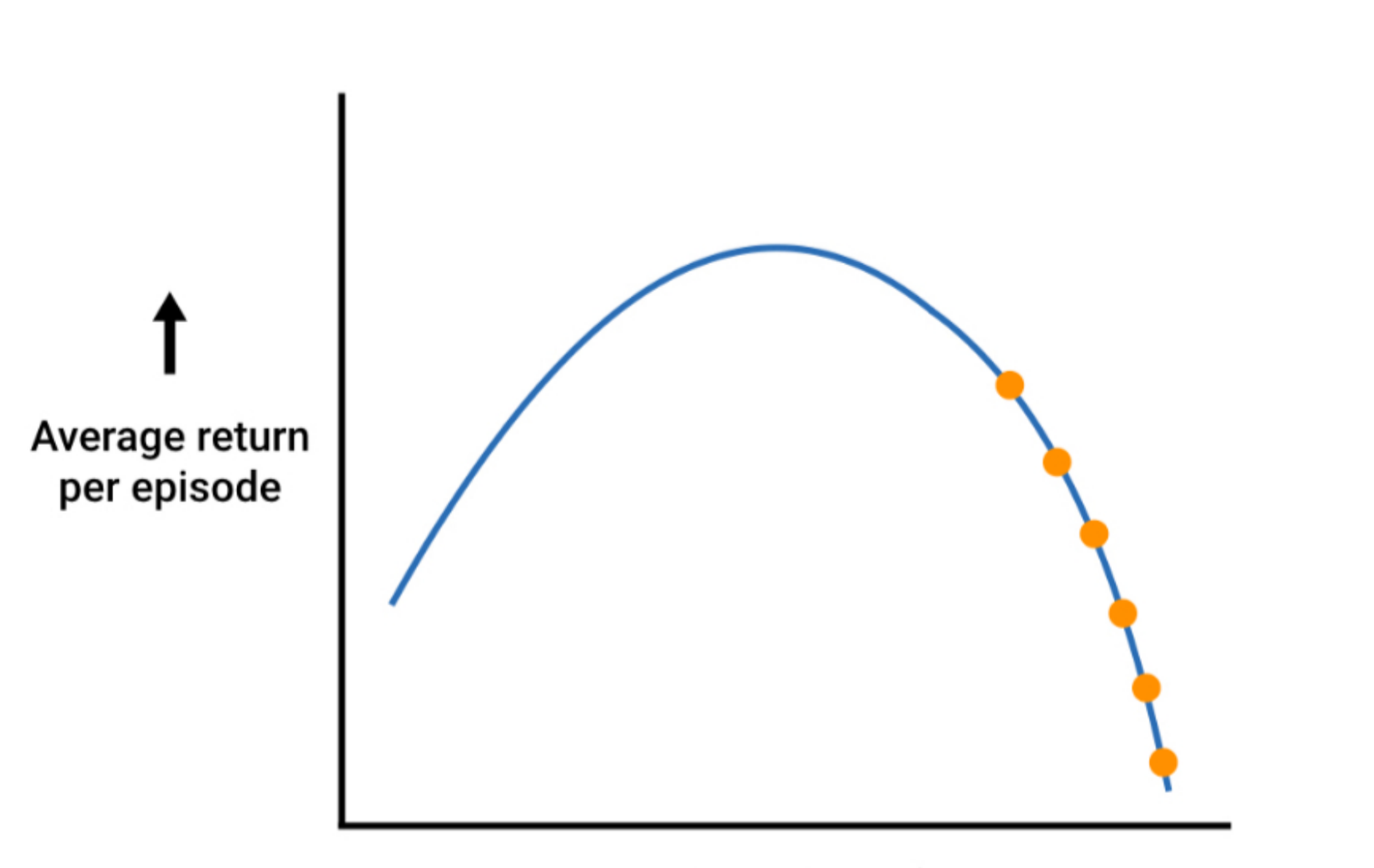


- [x]



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
Incorrect. 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.