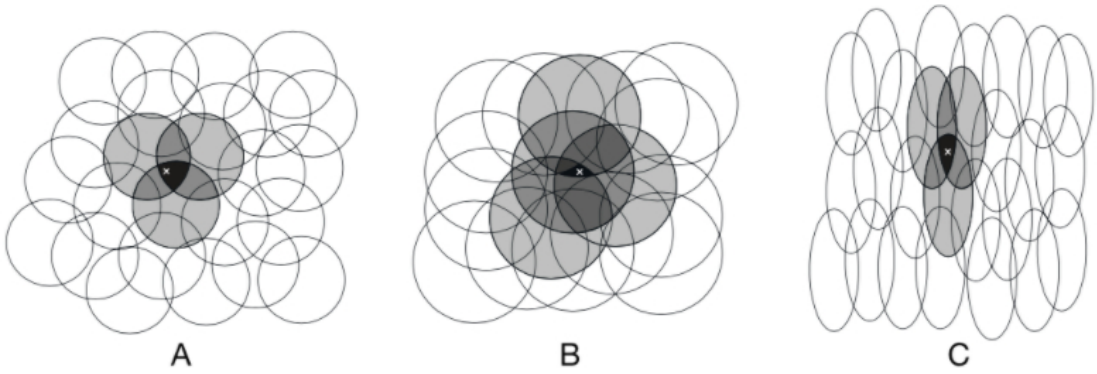
Constructing Features for Prediction

TOTAL POINTS 12

1. Which of the following is TRUE about coarse coding? (Select all that apply) 1 point

- [ ] In coarse coding, generalization occurs between states that have features with overlapping receptive fields.
- [ ] In coarse coding, generalization between states depend on the size and shape of the receptive fields.
- [ ] When using features with large receptive fields, the function approximator cannot make discriminations that are finer than the width of the receptive fields.
- [ ] When training at one state, the learned value function will be updated over all states within the intersection of the receptive fields.

2. Consider a continuous two-dimensional state space. Assuming linear function approximation with the coarse-codings in either A, B or C, which of the following is TRUE? (Select all that apply) 1 point



- [ ] Generalization is broader in case A as compared to case B.
- [ ] In case B, when updating the state marked by an 'x' , the value function will be affected for a larger number of states as compared to case A.
- [ ] In case C, each update results in more generalization along the vertical dimension, as compared to horizontal dimension.
- [ ] In case C, each update results in more generalization along the horizontal dimension, as compared to vertical dimension.

3. Which of the following is TRUE about tile coding? (Select all that apply) 1 point

- [ ] Tile coding is a form of coarse coding with tiles as its receptive fields.
- [ ] Tile coding with one tiling is an instance of state aggregation.
- [ ] The size of the feature vector equals the number of tilings: each feature corresponds to one tiling.
- [ ] With only one tiling, generalization occurs only between states within the same tile.

4. When tile coding is used for feature construction, the number of active or non-zero features 1 point

- ○ is the number of tiles.
- ○ is the number of tilings.
- ○ is the number of tilings multiplied by the number of tiles.
- ○ depends on the state.

5. Which of the following is TRUE about neural networks (NNs) ? (Select all that apply) 1 point

- ☐ A NN is feedforward if there are no paths within the network by which a unit's output can influence its input.
- ☐ Hidden layers are layers that are neither input nor output layers.
- ☐ The output of the units in NNs are typically a linear function of their input signals.
- ☐ NNs are parameterized functions that enable the agent to learn a nonlinear value function of state.
- ☐ The nonlinear functions applied to the weighted sum of the input signals are called the activation function.

6. Which of the following is the rectified linear activation function? 1 point

- ○ $f(x) = \frac{1}{1+e^{-x}}$
- ○ $f(x) = 1$ if $x > 0$ and 0 otherwise
- ○ $f(x) = max(0, x)$
- ○ $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

7. Which of the following is TRUE about neural networks (NNs)? 1 point

- ○ A NN with a single hidden layer can represent a smaller class of functions compared to a NN with two hidden layers.
- ○ The universal approximation property of one-hidden-layer NNs is not true when linear activation functions are used for the hidden layer.
- ○ Given the universal approximation property of one-hidden-layer NNs, there is no benefit to including more layers in the network.

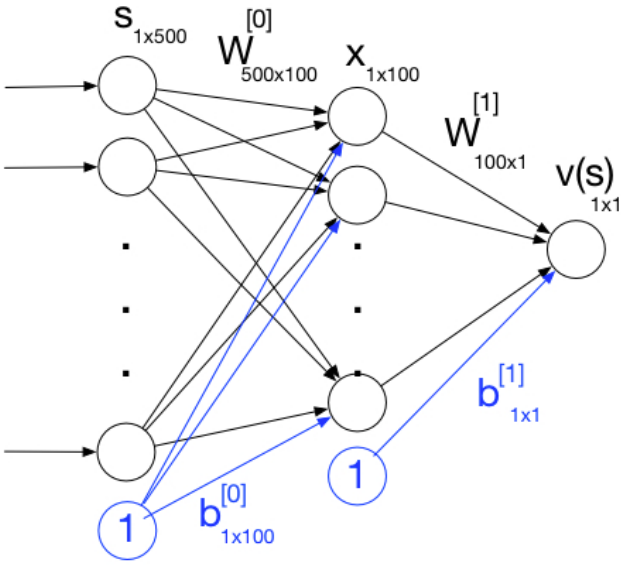
8. Which of the following is TRUE about backpropagation? (Select all that apply) 1 point

- ☐ Backpropagation corresponds to updating the parameters of a neural network using gradient descent.
- ☐ Backpropagation involves computing the partial derivatives of an objective function with respect to the weights of the network.
- ☐ The forward pass in backpropagation updates the weights of the network using the partial derivatives computed by the backward passes.
- ☐ Backpropagation computes partial derivatives starting from the last layer in the network, to save computation.

9. Training neural networks (NNs) with backpropagation can be challenging because (Select all that apply) 1 point

- [ ] the loss surface might have flat regions, or poor local minima, meaning gradient descent gets stuck at poor solutions.
- [ ] the initialization can have a big impact on how much progress the gradient updates can make and on the quality of the final solution.
- [ ] neural networks cannot accurately represent most functions, so the loss stays large.
- [ ] learning can be slow due to the vanishing gradient problem, where if the partial derivatives for later nodes in the network are zero or near zero then this causes earlier nodes in the network to have small or near zero gradient updates.

10. Consider the following network: 1 point



where for a given input s, value of s is computed by:

$\psi = sW^{[0]} + b^{[0]}$

$x = max(0, \psi)$

$v = xW^{[1]} + b^{[1]}$

What is the partial derivative of $v(s)$ with respect to $W_{ij}^{[0]}$?

- s_i
- $W_j^{[1]} s_i$ if $x_j > 0$ and 0 otherwise
- x_j
- $W_j^{[1]} x_j$ if $x_j > 0$ and 0 otherwise

11. Which of the following is TRUE? (Select all that apply) 1 point

- [ ] When using stochastic gradient descent, we often completely eliminate the error for each example.

- [ ] The difference between stochastic gradient descent methods and batch gradient descent methods is that in the former the weights get updated using one random example whereas in the latter they get updated based on batches of data.

- [ ] Adagrad, Adam, and AMSGrad are stochastic gradient descent algorithms with adaptive step-sizes.

- [ ] Setting the step-size parameter for stochastic gradient descent can be challenging because a small step-size makes learning slow and a large step-size can result in divergence.

12. Which of the following is TRUE about artificial neural networks (ANNs)? (Select all that apply) 1 point

- [ ] It is best to initialize the weights of a NN to large numbers so that the input signal does not get too small as it passes through the network.

- [ ] It is best to initialize the weights of a NN to small numbers so that the input signal does not grow rapidly as it passes through the network.

- [ ] If possible, it would be best to initialize the weights of an NN near the global optimum.

- [ ] A reasonable way to initialize the NN is with random weights, with each weight sampled from a normal distribution with the variance scaled by the number of inputs to the layer for that weight.
