# CSE 471 Intro to AI (Kambhampati) Test 2

**[Must be submitted in hard copy by 5pm on Tuesday Nov 15th.]**

**Name:**____________________________ **Student ID:**__________________________

**Please read and sign the following declaration:**

*I hereby confirm, under penalty of academic dishonesty, that I did not use any resources beyond the class notes, lecture videos, class homework solutions and the text book in doing this take-home test. I also confirm that I have not discussed this exam with anyone other than the instructor and the TAs.*

*Signature:*__________________________

| | |
|---|---|
| Qn 1. MDPs [12] | |
| Qn 2. Reinforcement Learning [11] | |
| Qn 3. Neural Networks [10] | |
| Qn 4. Short Answer Questions [5x2] | |
| Extra Credit | |

Qn I [MDPs]
Consider an undiscounted MDP having three states (1,2,3), with rewards -1, -2 and 0 respectively. State 3 is a terminal (sink) state. In states 1 and 2, there are two possible actions: a and b. The transition model is as follows.

. in state 1, action a moves the agent to state 2 with probability 0.8 and makes the agent stay put with probability 0.2

. in state 2, action a moves the agent to state 1 with probability 0.8 and makes the agent stay put with probability 0.2

. in either state 1 or state 2, action b moves the agent to state 3 with probability 0.1 and makes the agent stay put with probability 0.9.

**[1]For this MDP, how many possible value functions are there? How many possible policies?**

[4] Assuming that $V_0$() of each node is just its immediate reward. Show one iteration of value iteration. Specifically compute $V_1(1)$ , $V_1(2)$ and $V_1(3)$

[4] Suppose we initialized the policy vector to be "**do action b in both states 1 and 2**". What is the value of this policy? (i.e., compute the value vector corresponding to this policy)

[3] Given the value vector in the previous part, what is the Greedy Policy corresponding to it? In particular, is the policy the same as the one you started with? Show your work.

**[Reinforcement Learning]**

**Part 1: [2+1+3 points]**

This gridworld MDP operates like to the one we saw in class. The states are grid squares, identified by their row and column number (row first). The agent always starts in state (1,1), marked with the letter S. There are two terminal goal states, (2,3) with reward +5 and (1,3) with reward -5. Rewards are 0 in non-terminal states. (The reward for a state is received as the agent moves into the state.) The transition function is such that the intended agent movement (North, South, West, or East) happens with probability .8. With probability .1 each, the agent ends up in one of the states perpendicular to the intended direction. If a collision with a wall happens, the agent stays in the same state.

| | | |
|---|---|---|
| | | **+5** |
| **S** | | **-5** |

The agent starts with the policy that always chooses to go right, and executes the following three trials: 1) (1,1)–(1,2)–(1,3), 2) (1,1)–(1,2)–(2,2)–(2,3), and 3) (1,1)–(2,1)–(2,2)–(2,3). What are the monte carlo (direct utility) estimates for states (1,1) and (2,2), given these traces? (5 points)

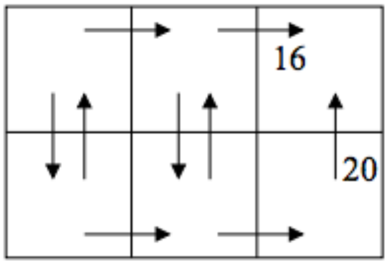
What is the model of the Go-Right action that you can learn from the traces above?

Using a learning rate of .1 and assuming initial values of 0, what updates does the TD-learning agent make after trials 1 and 2, above? (5 points)

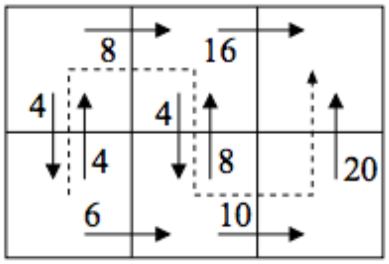
## Reinforcement Learning [Part 2][5pt]

Consider the deterministic world below (part (a)). Allowable moves are shown by arrows, and the numbers indicate the reward for performing each action. If there is no number, the reward is zero.

Given the $Q$ values in (b), show the changes in the $Q$ estimates when the agent take the path shown by the dotted line (the agent starts in the lower left cell) when $\gamma = 0.5$. Show all of your work.



(a)

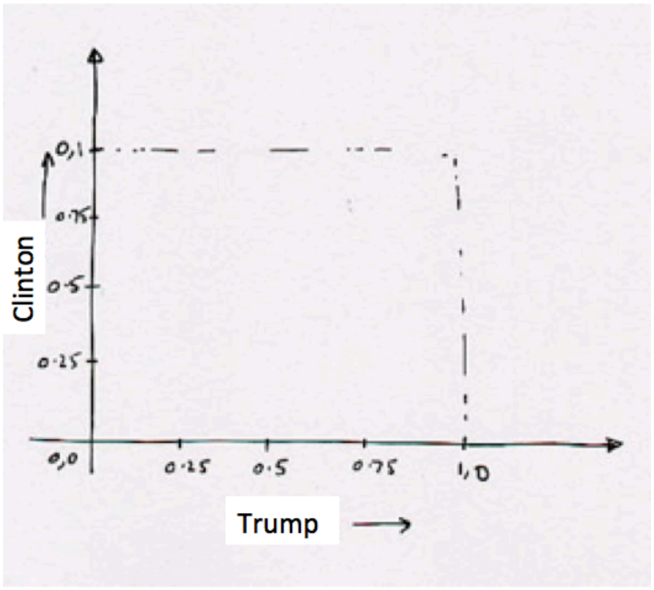


(b)

## [Neural Networks][Part 1]

Having failed to make the American public learn the concept **Neither Trump nor Clinton**, Gary Johnson bought a neural net program and decided to teach the idea to a network. He wants to go with a single unit perceptron with two inputs, corresponding to T and C, and the weights corresponding to wt and wc. The unit has a threshold w0. He wants to train the perceptron by giving examples in the following order:

|  | Trump | Clinton | Output |
|---|---|---|---|
| E1 | 1 | 0 | 0 |
| E2 | 0 | 1 | 0 |
| E3 | 1 | 1 | 0 |
| E4 | 0 | 0 | 1 |



A.[1] According to what you know, do you think Gary will have any more success in training a perceptron (than in training the unwashed masses, as it were)? How do you know that this is possible? (hint: consider plotting the examples on the graph above.

**Suppose the initial weights of the perceptron are**
**wt = -1 wc= -1 and w0= -1.1.** The **learning rate (alpha) is 0.1**

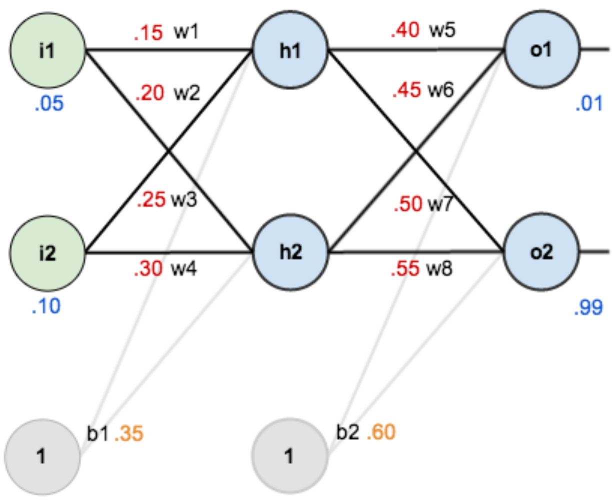
B.[2] Show the *decision surface* represented by these weights on the graph above. Using the decision surface, explain which of the examples are classified correctly by this network? Which are classified incorrectly?

C.[3] Suppose now the first example is presented to the network. What are the weights as computed by the perceptron update rule? (remember that alpha is 0.1). Show the new decision surface corresponding to new weights in the graph above. Which examples are classified correctly and which are classified incorrectly by the new decision surface?

**[Neural Networks][Part 2][4]**

Consider the multi-layer neural network below, where h1, h2, o1, o2 are threshold units. The current weights on the network are shown in red. The training example inputs and outputs are shown in blue under i1, i2, o1,o2 units.
Assume that the threshold units all use logistic activation function

1/(1 + exp(-x))



For the training example (0.5, 0.1) that is given what is the half-squared error at the units o1 and o2? (do this by showing the activations at h1, h2, o1 and o2. You can use calculators)

Clearly the weights w5,w6,w7 and w8 can be modified using the gradient update rule that is in terms of the total error at o1 and o2. To update the weights w1,w2,w3,w4 however, we need to know how much error is made at h1 and h2. The problem is that we don't know what the true values are supposed to be at h1 and h2 for the current input. How do neural network training algorithms handle this problem?

**Short answer questions:**

**1.** Is it true that increasing the discount factor, $\gamma$, is a way to make value iteration converge faster?

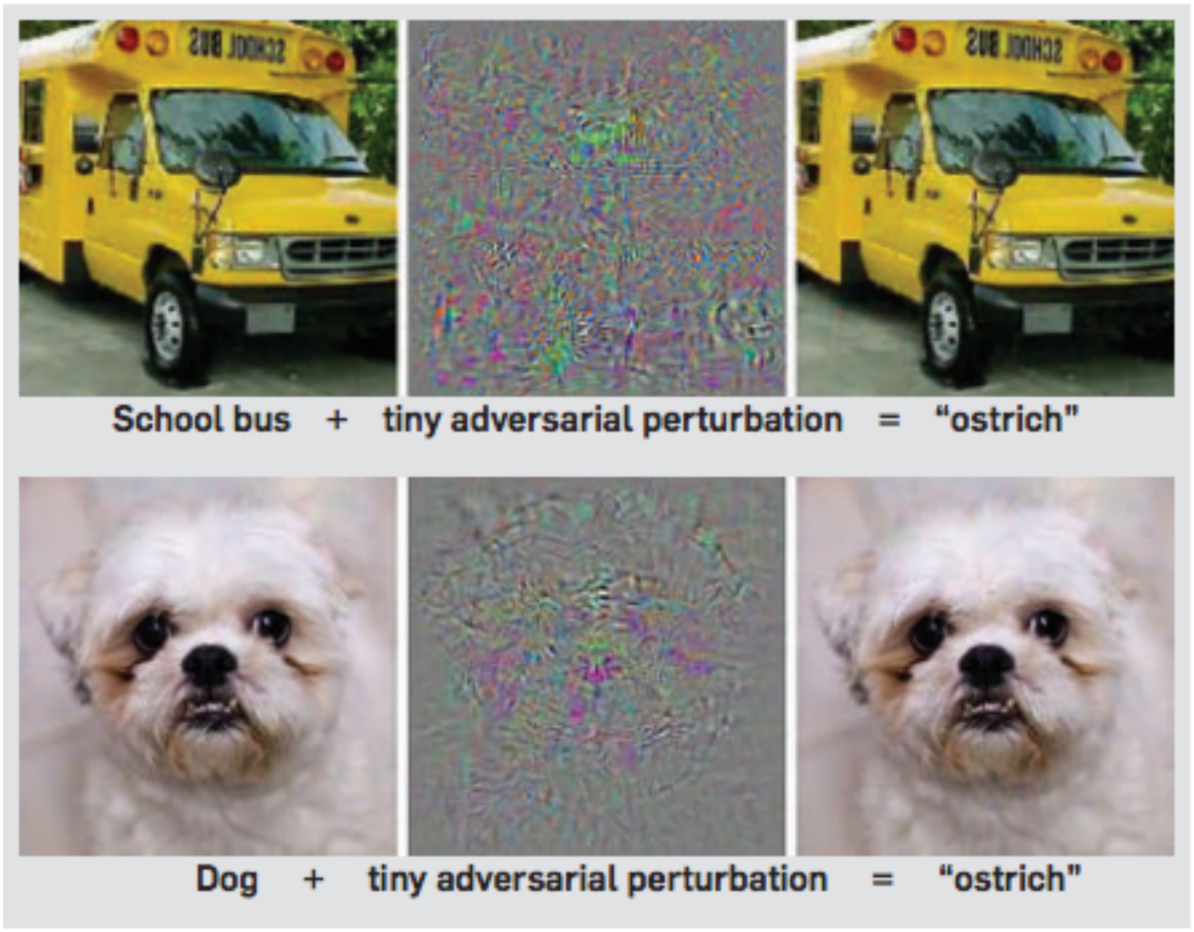
2. We saw that an MDP agent can find optimal policy by searching just in the space of pure policies (where each state is mapped to a single unique action). Can we also say that a Reinforcement Learning agent need only consider pure policies?

*3.* Your friend says that she trained two different networks on the same training data (i) a 4-layer neural network and (ii) a 15-layer network. What can you say about the error each of the networks makes on (a) the training data and (b) the test data.

*4.* You are given some two dimensional points (x1, x2) labeled as +ve or –ve according to some unknown rule. You find that the only way to learn a good classifier on the data is to use a neural network with two layers of neurons. Your friend however says that she could learn a classifier with a simple perceptron. Can you think of a way this is possible?

*5.* We can view an MDP agent as playing a game against nature: as the agent does an action with several potential outcomes, the nature decides which of the outcomes will actually be realized. Does it then make sense for the agent to use a Min-Max evaluation to decide its moves?

**[Fun Extra-Credit question]** An article in this month's Communications of ACM shows this example where a state of the art neural network for image recognition confidently makes the following predictions on slightly perturbed images.


School bus + tiny adversarial perturbation = "ostrich"
Dog + tiny adversarial perturbation = "ostrich"

In addition to the obvious conclusion—that everything in our world is really just a cleverly disguised Ostrich, what else do these examples tell us about neural network-based image recognition *vis a vis* human image recognition? Answer this briefly based on the discussion in the class as well as the short article at goo.gl/TaHrRe