CS 3600 Final exam

Due December 13, at 11:59pm Eastern

List any collaborations here: Vishal Kumar, Ajay Krishnaswamy

Question 1. In each of the following questions, you will be given a scenario and be asked to consider whether a number of possible algorithms may or may not be well-suited to the task. One sentence should suffice for each. (1 point each)

1.a. A robot must navigate a collapsed mine to bring food and water to some survivors. If it falls down a shaft it will be destroyed. Air breezes are detectable nearby to most shafts, but breezes are sometimes too faint to be detected. Additionally, the rough terrain means that the robot's feet may slip and it may not always move forward at the intended pace. Explain why you should or should not use each of the following to reach the survivors:

i) A*: This is not ideal for the problem as it assumes a static environment, whereas the collapsed mine scenario involves unpredictability such as faint breezes.

ii) A Markov Decision Process: This is partially suitable for the problem, as they are good with randomness like the uncertain terrain, but do not have a fully observable environment as the mine is collapsed.

iii) A Partially-Observable Markov Decision Process: This is very suitable for the problem as they work well with partial observability and uncertainty, which is depicted in this problem.

iv) A Dynamic Bayes Network: This is likely suitable for the problem as they model the varying aspects of the mine, such as air flow and terrain, and handle uncertainties in movement and sensing.

v) A non-Dynamic Bayes Network: This is not very suitable for the problem because they don't account for changes over time, unlike the dynamic nature of this problem.

1.b. In the game of curling, the objective is to slide a stone down a sheet of ice and try to get it as close to the center of a target as possible. You release the stone at a particular velocity, however there are small imperfections in the ice. Explain why you should or should not use each of the following to predict where the stone will stop.

i) A*: This is not ideal for the problem as it is not useful in predicting the final position in curling which is a continuous dynamic game, given that they are path finders in discrete environments.

ii) A Markov Decision Process: This is not ideal for the problem as it cannot predict physical trajectories influenced by continuous variables, given that they are decision makers in stochastic environments.

iii) A Partially-Observable Markov Decision Process: This is not ideal for the problem as it cannot predict the final position in curling, because they focus on decision making under uncertainty and partial information, which does not align with the physics-based motion of the stone.

iv) A Dynamic Bayes Network: This could be somewhat useful in modeling the behavior of the curling stone, as they can incorporate the dynamics of the stone's movement and the uncertain effect of ice imperfections over time.

v) A non-Dynamic Bayes Network: This would not be suitable in modeling the behavior of the curling stone, as they are better suits for static modeling and lack the capability to capture the dynamic interaction between the stone and ice and the varying conditions.

1.c. Suppose you need to assemble a piece of Ikea furniture but lost the instructions. You need to figure out what order to assemble the pieces in, starting with a pile of parts. You have a very powerful computer that doesn't have the internet. Assume that since you will build the furniture according to the instructions generated, you will not have any problem identifying parts or applying each step. Explain why you should or should not use each of the following to create the sequence of steps to build the furniture:

i) A*: This could be useful in the problem as it can efficiently find the shortest sequence of steps to assemble the furniture by evaluating different assembly sequences based on heuristics. However, the effectiveness relies on the quality and defined nature of the heuristics.

ii) Reinforcement Learning: This is not ideal for the problem as it requires trial and error learning, which is impractical considering the physical assembly of the furniture, where mistakes can be costly.

iii) A Dynamic Bayes Network: This is not very useful in the problem, as it is designed to model dynamic systems and temporal relationships, which are less relevant to the sequential process of assembling furniture with a predefined set of parts to assemble.

iv) A Perceptron: This is not suitable as it is primarily used for classification tasks and lacks the structure needed to sequence a set of assembly steps from a pile of furniture parts.

1.d. Suppose you want to classifying rodents found on campus by species. There are 3 types of rodents (rat, shrew, or, mouse) that can be identified by inspecting size, color, tail length, whisker length, and size of front teeth. You must consider that any of these attributes can take on a range of values (e.g., a baby rat could be the size of a full-grown mouse). Explain why you should or should not use each of the following:

i) A*: This is not suitable for the problem as it is a path finding algorithm, not a classifying algorithm, and does not possess the capability to analyze and classify data based on multiple attributes.

ii) A Markov Decision Process: This is not suitable for the problem as it is a decision-making algorithm, not a classifying algorithm, and does not possess the capability to analyze and classify data based on multiple attributes.

iii) A Dynamic Bayes Network: This is not ideal for the problem as it is not suited for analyzing static characteristics (like size and color), but rather suited for dynamic, time-dependent variables.

iv) A Perceptron: This is definitely suitable for the problem as it fundamentally used for classification purposes. It can learn to classify rodents into different species based on input features such as size, color, tail length, whisker length, and front teeth size, even when these features vary within a species.

Question 2. (2 points) Suppose an autonomous car is driving down a road that is passing through a forest. The car strikes a pedestrian crossing a street. The pedestrian was dressed in a Halloween costume that made them look like an *Ent* (a walking, talking tree from The Lord of the Rings). It was raining at the time. The autonomous car uses only camera sensors. The car makes decisions using deep reinforcement learning. You have been called in to help investigate the crash.

You remember from CS 3600 that your professor told you there were four potential causes of errors. Explain how each type of error could have caused the crash using specific details from the scene.

Sensor error: The unusual costume combined with poor weather conditions likely caused the sensors to fail in correctly identifying the pedestrian as a human figure crossing the street.

Effector error: If the car's braking system didn't respond promptly or effectively due to the wet conditions, this could have been a contributing factor, regardless of the sensor input.

Model error: The car's AI might not have been trained to recognize a person in an Ent costume as a pedestrian, especially in conjunction with the rain, leading to a misinterpretation of the scene and hence, the crash.

Wrong objective function: The car's decision-making system might have been programmed to prioritize maintaining traffic flow over cautious driving in conditions of poor visibility or unusual obstacles, leading it to not slow down or stop for what is perceived as a non-human obstacle, which is the Ent costume.

Question 3. (1 point). Consider the issue of prejudicial bias in machine learning models.

3.a. Explain how an imbalance of data for a particular feature can lead to prejudicial bias.

An imbalance of data for a particular feature can lead to prejudicial bias when the training data overrepresents certain types or categories of data while underrepresenting others. For example, if a dataset used for a training facial recognition system contains images of predominantly one racial group, the model may become biased towards recognizing faces from that group more accurately. This bias emerges from the model's learning being skewed by the disproportionate representation by certain features in the training data, limiting its ability to generalize effectively across diverse datasets.

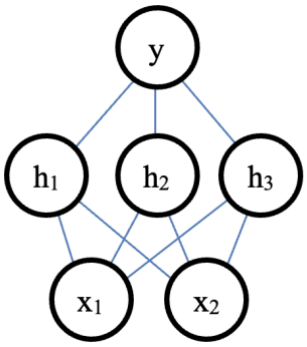
3.b. Suppose you have restricted the model's access to sensitive features but your red team discovers the potential for prejudicial bias still exists due to secondary features that cannot be removed. What technique might you use, and why might it work?

When sensitive features are restricted and prejudicial bias is still a concern, a technique like "Learning Fair Representations" can be effective as it focuses on both group and individual fairness. It ensures that the model's decisions are based on relevant and non-discriminatory attributes rather than sensitive or secondary features that might introduce bias. This is achieved through an optimization process that seeks a data representation which encodes the data effectively for decision-making while simultaneously obfuscating any information about membership in protected groups.

Question 4. Neural networks. Recall that the Rectified Linear Unit (ReLU) activation function is defined as:

$$ReLU(x) = \max(x, 0)$$

Consider the following neural network:



The activation function for all nodes are ReLUs. Neurons h_1, h_2, h_3, and y have a bias of -1. All weights, including the bias weights, are initialized to 0.1.

4.a. (1 point) x_1 and x_2 are input nodes. Suppose they are given the following values from one data point:

$x_1 = 3.0$
$x_2 = 2.0$

Compute the output activation of node *y*. (show work for partial credit)

For all h nodes, A(h) = ReLU(0.1 * 3 + 0.1 * 2 – 0.1*1) = ReLU(0.4) = 0.4
For y, A(y) = ReLU(0.1 * 0.4 + 0.1 * 0.4 + 0.1 * 0.4 – 0.1*1) = ReLU(0.02) = 0.02

output of $y = 0.02$

4.b. (1 point) Suppose the target value should have been **0.0**. Using the loss function

$$L(target, output) = \frac{1}{2}(target - output)^2$$

and output of y that you computed from part 4.a, compute w_{x_1,h_1}, which is the weight between nodes x_1 and h_1, after back propagation. Use a learning rate of $\alpha = 0.5$.

(Hint: you can verify your answer by using updated values for the weights in another forward pass to see if the output of *y* is closer to the target)

∂L/∂w_{x1,h1} = ∂L/∂y * ∂y/∂h1 * ∂h1/∂w_{x1,h1}

L (target, output) = 0.5 * (target – output)^2

∂L/∂y = –(target – output) = –(0 – 0.02) = 0.02

∂y/∂h1 = w_{y,h1} = 0.1

∂h1/∂w_{x1,h1} = x1 = 3.0

∂L/∂w_{x1,h1} = ∂L/∂y * ∂y/∂h1 * ∂h1/∂w_{x1,h1} = 0.02 * 0.1 * 3.0 = 0.006

w' = w – α * ∂L/∂w = 0.1 – 0.5 * 0.006 = 0.097

w_{x_1,h_1} = 0.097

4.c. (1 point) Suppose at some point in the future of the neural network training the weights are:
$w_{x_1,h_1} = w_{x_2,h_1} = w_{x_1,h_2} = w_{x_2,h_3} = w_{x_1,h_3} = w_{x_2,h_3} = w_{bias,h_1} = w_{bias,h_2} = w_{bias,h_3} = 0.08$
$w_{h_1,y} = w_{h_2,y} = w_{h_3,y} = w_{bias,y} = 0.09$
Run a forward pass on the neural network using the following data:
x_1 = 0.6
x_2 = 0.6
Compute the output activation of node *y*. (show work for partial credit)

For all h nodes, A(h) = ReLU(0.08 * 0.6 + 0.08 * 0.6 - 0.08 * 1) = ReLU(0.016) = 0.016
For y, A(y) = ReLU(0.09 * 0.016 + 0.09 * 0.016 + 0.09 * 0.016 – 0.09 * 1) = ReLU(-0.08568) = 0

output of *y* = 0.0

4.d. (1 point) Suppose the true target value is 1.0 for the data point in 4.c. Explain what will happen to the weights after the error from the output of y is backpropagated. (Hint: computing the back-propagation step may help though we don't require you to give us the new weights). Why is it different than what happened in 4.b.?

In this scenario, when the target value is 1.0 but the network predicts 0, an error is generated, leading to a non-zero loss. During backpropagation, this loss is used to calculate gradients, which informs how the weights should be adjusted to reduce the error. The weights between the input layer and the h layer, as well as between the h layer and output layer, will be updated in a way that increases the likelihood of the network predicting values closer to the true target in future forward passes. This process is iterative and continues throughout the training of the network. The difference from 4b arises because previously there was no error, which meant no changes

were made to the weights; now, with a target of 1.0, the network has a clear signal to change the weights to reduce the loss.

4.e. (1 point) It's not good for a network to be sensitive to the choice of a bias value. Instead of fiddling with the bias value, your professor, who is a genius,[*] has invented a new activation function called the Markified Linear Unit (MaLU), which is defined as:

$$g(x) = \begin{cases} x, & x > 0 \\ 0.001x, & x \leq 0 \end{cases}$$

Explain why the Markified Linear Unit will work better on the neural network above.

The MaLU activation function offers a nuanced approach to handling negative inputs compared to the traditional ReLU activation function. By providing a small, non-zero output for negative inputs, MaLU maintains a gradient through which learning can occur even when the input to a neuron is not positive. This feature mitigates the risk of neurons "dying" (a state in which neurons stop learning entirely due to a lack of gradient when the input is negative. Therefore, MaLU could help in keeping the neural network adaptable, ensuring all neurons have the chance to adjust and contribute to the model's learning process. This is particularly beneficial in deep networks with complex patterns where maintaining the ability to learn from all features is crucial.

[*] Allegedly, it has not been confirmed.

Question 5. In this question we look into modifications to the basic gradient descent technique.

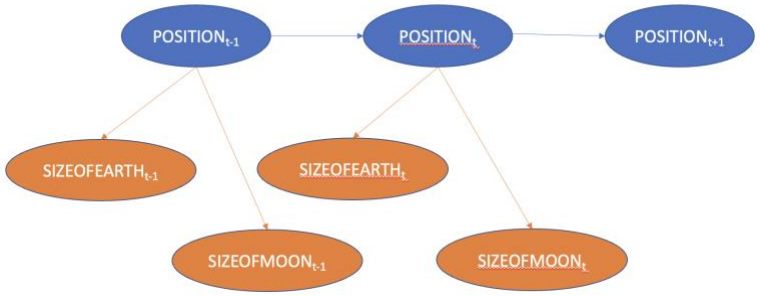
5.a. (1 point) Some more advanced gradient descent algorithms introduce the idea of "momentum" where a fraction of the delta of a weight from the previous iteration of back-propagation is added to the current delta of a weight from the current iteration of back-propagation. How does it help with the problem of local minima?

Introducing momentum in gradient descent algorithms helps mitigate the problem of local minima by adding a fraction of the previous weight update to the current update. This process can be visualized as a ball rolling down a hill; as it goes, it builds up momentum, which allows it to roll through small dips without getting stuck. In the context of gradient descent, this momentum term helps the algorithm to continue past local minima and not settle for sub-optimal solutions, especially in complex landscapes with many such minima.

5.b. (1 point) Another advanced option for back-propagation is to use a decaying learning rate. The learning rate (alpha) starts relatively high and gets smaller after every epoch. What problem that we discussed in class does a decaying learning rate help with?

Using a decaying learning rate addresses the problem of overshooting the minimum. At the beginning of training, when the weights are far from optimal values, a larger learning rate helps in making faster progress. However, as the algorithm approaches the minimum, a high learning rate can lead to overshooting the optimal values, causing the model to miss the minimum and oscillate around it. By gradually reducing the learning rate, the updates become finer as the model gets closer to the minimum, allowing for more precise convergence to the optimal weights.

Question 6. (1 point) Suppose you are an astronaut on the way to the moon. Unfortunately, your radar has gone out and you have lost contact with Mission Control on Earth. You want to be able to estimate your current position and predict when you will get to the lunar orbital insertion point so you can fire your thrusters and enter orbit. You figure your current position is related to the observable size of the Earth and the Moon from your position (as you get farther from Earth it looks smaller; as you get closer to the Moon it looks bigger). Your observations aren't perfect, but you remember from CS 3600 that a Dynamic Bayesian Network can be used to estimate unobservable features—like your position in space—from imperfect observations. This is the Dynamic Bayesian Network you come up with, along with the conditional probability tables (not shown):



You decide to implement a particle filter to estimate your future position. This DBN is a bit different from the ones studied in class. What step in the particle filtering algorithm has to change to account for this network, and how must it be changed? You do not have to derive any equations.

For this DBN, the step that needs to be modified is the prediction step. When propagating each particle from time t-1 to t, we need to use transition probabilities that depend on both the previous position (position_(t-1)) and the observed sizes (sizeofearth_(t-1) and sizeofmoon_(t-1)). This means that the new state for each particle is sampled from the probability distribution:

P(position_(t)| position_(t-1), sizeofearth_(t-1), sizeofmoon_(t-1))

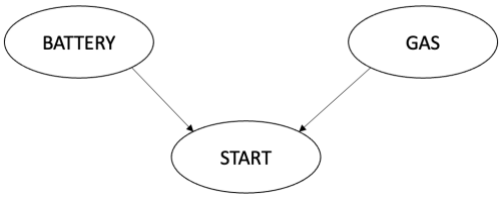
This change is necessary because the position is not solely determined by its previous state but is also directly influenced by the size of the Earth and Moon as seen by the astronaut. The particles must reflect these relationships to accurately predict future positions.

Question 7. (1 point) Someone develops an AI system that takes an image of you and makes it look more professional so that you can add it your LinkedIn account or résumé. For example, it will make your clothes look more expensive, remove blemishes and dirt from your face, make your hair look professionally and expensively groomed, and change the background to look like you are sitting in a well-furnished, private office. But this service needs a very big server and costs with high energy and maintenance costs. Thus, the service costs $500 for a set of 5 pictures. What principle from our class discussion on societal implications must be considered, and why.

The principle of fairness must be considered when evaluating the societal implications of such an AI service. This service, due to its high cost, inherently creates a barrier to access, potentially leading to unwarranted prejudicial bias. Since only those who can afford the $500 fee for the set of five pictures can benefit from the service's ability to enhance professional appearance, it could disproportionately favor wealthier individuals, enhancing their chances of being perceived more positively on professional platforms like LinkedIn or in resumes. This could lead to an unfair advantage in job searching and networking opportunities, thereby affecting different socio-economic groups unequally. Access to such services could reinforce or exacerbate existing societal biases where individuals from less affluent backgrounds are already ate a disadvantage, thus further growing socio-economic divides.

Question 8: Consider the following Bayesian network that models components of a car.

- BATTERY: The car's battery is working (true or false)
- GAS: The car's gas tank has gas (true or false)
- START: The engine will start (true or false).



The engine will not start when the car is out of gas or if the battery is not working. If the car has gas and the battery is working there is an 80% chance that the car will start. History suggests that the battery works 70% of the time and that the car has gas 60% of the time.

8.a. (1 pt.) Fill out the conditional probability tables:

BATTERY	P(BATTERY)
T	0.7
F	0.3

GAS	P(GAS)
T	0.6
F	0.4

BATTERY	GAS	P(START=T | BATTERY, GAS)	P(START=F | BATTERY, GAS)
T	T	0.8	0.2
T	F	0.0	1.0
F	T	0.0	1.0
F	F	0.0	1.0

8.b. (2 pts.) Suppose you know that the car doesn't start. Use the definition of independence to prove or show that BATTERY and GAS not independent on each other when the car is observed to not start. Recall that dependence means that when one variable changes its value, then the distribution over the values of the other variable changes. Show your work to receive any partial credit.

Hint: Variables A and B are independent if P(A, B) == P(A)P(B). But we have an extra variable, C with a *given* value that must be factored in.

Here, we need to show that $P(A, B \mid C) \neq P(A \mid C) * P(B \mid C)$, so we have to check $P(\text{BATTERY}= T, \text{GAS}= T \mid \text{START} = F) \neq P(\text{BATTERY} = T \mid \text{START} = F) * P(\text{GAS} = T \mid \text{START} = F)$

$P(\text{START} = F) = 0.4 * 0.3 + 0.6 * 0.3 + 0.4 * 0.7 + 0.7 * 0.6 * 0.2 = 0.664$

$P(\text{START} = F \mid \text{BATTERY} = T) = P(\text{START} = F \mid \text{BATTERY} = T, \text{GAS} = T) * P(\text{BATTERY} = T, \text{GAS} = T) + P(\text{START} = F \mid \text{BATTERY} = T, \text{GAS} = F) * P(\text{BATTERY} = T, \text{GAS} = F)$

$P(\text{START} = F \mid \text{GAS} = T) = P(\text{START} = F \mid \text{GAS} = T, \text{BATTERY} = T) * P(\text{GAS} = T, \text{BATTERY} = T) + P(\text{START} = F \mid \text{GAS} = T, \text{BATTERY} = F) * P(\text{GAS} = T, \text{BATTERY} = F)$

$P(\text{START} = F \mid \text{BATTERY} = T) = 0.2 * 0.7 * 0.6 + 1 * 0.7 * 0.4 = 0.364$

$P(\text{START} = F \mid \text{GAS} = T) = 0.2 * 0.7 * 0.6 + 1 * 0.3 * 0.6 = 0.264$

P(BATTERY = T | START = F) = [P(START = F | BATTERY = T) * P(BATTERY = T)] / P(START = F)

P(GAS = T | START = F) = [P(START = F | GAS = T) * P(GAS = T)] / P(START = F)

P(BATTERY = T | START = F) = (0.364 * 0.7) / 0.664 = 0.384

P(GAS = T | START = F) = (0.264 * 0.6) / 0.664 = 0.239

P(GAS = T, BATTERY = T | START = F) = [P(START = F | GAS = T, BATTERY = T) * P(GAS = T, BATTERY = T)] / P(START = F)

P(GAS = T, BATTERY = T | START = F) = (0.2 * 0.7 * 0.6) / 0.664 = 0.127

$0.384 * 0.239 = 0.092 \neq 0.127$

Thus, P(BATTERY= T, GAS= T | START = F) ≠ P(BATTERY = T | START = F) * P(GAS = T | START = F)

This means that knowing whether the battery is working affects the probability of the car having gas when it is observed that the car does not start, and vice versa. Therefore, BATTERY and GAS are not independent given the car does not start.

8.c. (1 pt.) Given your answer to 3.b and the fact that the car doesn't start, explain how you could use a voltmeter on the battery to predict whether the gas tank is empty. Draw a new Bayesian Network that illustrates how the test works. [Hint: your network should now have 4 nodes, and the new node should be an emission model]

Using a voltmeter to test the battery provides us with additional information about the state of the battery. This affects the probability distribution over the state of the gas tank. When we observe that the car does not start, it could be due to a dead battery, an empty gas tank, or both. If we then use a voltmeter and find out the battery is working, this would increase the probability that the gas tank is empty because the starting problem must be attributed to something else (since the battery is not the issue). Conversely, if the voltmeter shows the battery is not working, it does not necessarily mean the gas tank is full; however, it would not change the probability that the gas tank is empty.