CMPUT 366, Winter 2022
Midterm Exam

Total points: **60**

This exam has **6** pages

The midterm exam is closed book. You are not allowed to use any resources or equipment other than this exam, a pen/pencil, and a single-page "cheat sheet".

You are not allowed to use your own scratch paper. Instead, use backsides of the exam pages as scratch paper for your work.

You are not allowed to use any equipment (e.g., phones, computers, tablets, etc.)

You are not allowed to communicate with anybody except the examiners/proctors.

Name: _____

CCID: _____@ualberta.ca

---

1. **(Short Answers)**

   (a) **[2 points]** What are the semantics of a belief network? (Hint: What guarantee does it make about conditional independence?)

   Every node is independent of its non-descendants conditional on only its parents.

   (b) **[2 points]** What is an admissible heuristic?

   A function that takes a node as an argument and returns a weakly positive number that is guaranteed to be smaller than the cost of the cheapest path from its argument to a goal node.

   (c) **[2 points]** For what kind of problem is local search appropriate?

   When we only care about finding a goal state, and we don't care about the path to the goal state.

   (d) **[2 points]** Explain why hill-climbing is incomplete.

   Hill-climbing will halt whenever it reaches a local optimum. However, a problem can have many local optima that are not goal nodes, so the unmodified hill-climbing algorithm is not guaranteed to find a goal node if it exists.

   (e) **[2 points]** Explain why variable elimination can be more efficient than just computing the full joint distribution and then normalizing.

   Because by eliminating variables one at a time, variable elimination can often avoid repeated operations. (Also acceptable to say that it can operate on smaller factor tables)

   (f) **[2 points]** Why is loss on the training dataset not a good estimate for generalization performance?

   Because of overfitting: a sufficiently complex model can learn to optimally predict any training set, but it must do that by learning "noise" in the training set (i.e., regularities that are a result of chance rather than being present in the underlying population).

   (g) **[2 points]** What is selection bias?

   Selection bias is when whether subjects receive treatment is systematically related to their response to the treatment.

   (h) **[2 points]** Consider a computation with inputs x_1, x_2, x_3 and output y. What is computed by backward-mode automatic differentiation for each intermediate value s_i?

   The local derivative $\overline{s}_i = \frac{\partial y}{\partial s_i}$.

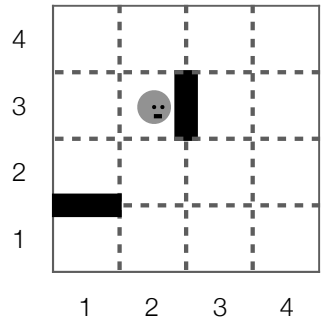
   (i) **[2 points]** Give an expression for the value of a single rectified linear unit (relu), with inputs x_1, x_2, weights w_1, w_2, and offset (bias) b.

   $h(x_1, x_2; w_1, w_2, b) = \max\{0, w_1 x_1 + w_2 x_2 + b\}$ OR $h(\mathbf{x}; \mathbf{w}, b) = \max\{0, \mathbf{w}^T \mathbf{x} + b\}$.

   (j) **[2 points]** Why are convolutional networks more efficient to train than fully-connected feed-forward networks?

   Either of the following: (1) Because they have dramatically fewer parameters than an equivalent fully-connected network, or (2) Because they encode invariances that reduce the number of examples they need to train on.

2. **(Search)**



GridBot is an agent that lives in the grid world pictured above. GridBot can move a single square up, down, left, or right, but not diagonally, and not through either of the two walls. GridBot starts in square $(x, y) = (2, 3)$, as pictured. GridBot's goal is to get to either of the bottom corners $(1, 1)$ or $(4, 1)$ using the smallest number of actions.

(a) **[2 points]** How many *states* does this search problem have?
16

(b) **[2 points]** Let *succ* be the successor function for this search problem. What are the return values for the following function calls:

i. $succ((2, 3)) = \{(2, 4), (1, 3), (2, 2)\}$
ii. $succ((3, 2)) = \{(2, 2), (3, 3), (3, 1), (4, 2)\}$
iii. $succ((1, 1)) = \{(2, 1)\}$
iv. $succ((4, 1)) = \{(3, 1), (4, 2)\}$

(c) **[2 points]** Consider the following *heuristic function*:

$$h((x, y)) = y - 1$$

Is the heuristic function h an *admissible heuristic* for this problem? Why or why not?
Yes. GridBot must move DOWN by at least $y - 1$ steps to get to a bottom corner, so it is always an underestimate.

(d) **[3 points]** Either construct an admissible heuristic function h' for this problem that *dominates* h, or else explain why this is impossible.
It is possible; here is an example:

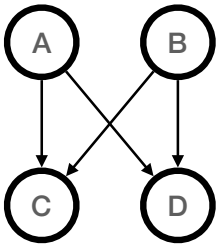
$$h((x, y)) = y - 1 + \min\{x - 1, 4 - x\}$$

(e) **[4 points]** List the **paths** that are removed from the frontier by A^* using the heuristic h from the previous question, **in the order in which they are removed.** You may stop once you remove a path to a goal node.

i. $\langle(2, 3), (2, 2)\rangle$
ii. $\langle(2, 3), (2, 2), (2, 1)\rangle$
iii. $\langle(2, 3), (2, 2), (2, 1), (1, 1)\rangle$

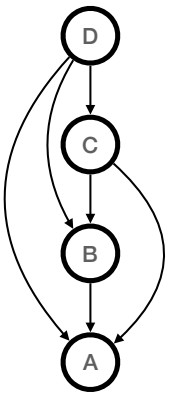
3. **(Uncertainty)**

(a) **[3 points]** What factorization of the joint distribution $P(A, B, C, D)$ does the belief network below represent?



$P(C \mid A, B)P(D \mid A, B)P(A)P(B)$

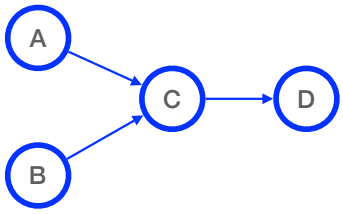
(b) **[3 points]** Is the belief network below also consistent with the same joint distribution? Why or why not? (Hint: What factoring does the belief network below represent?)



Yes. This belief network represents the factoring $P(A \mid B, C, D)P(B \mid C, D)P(C \mid D)P(D)$, which is a valid factoring for *any* joint distribution $P(A, B, C, D)$ by the chain rule.

(c) **[3 points]** Draw a belief network that represents the factoring

$$P(A, B, C, D) = P(D \mid C)P(C \mid A, B)P(A)P(B).$$



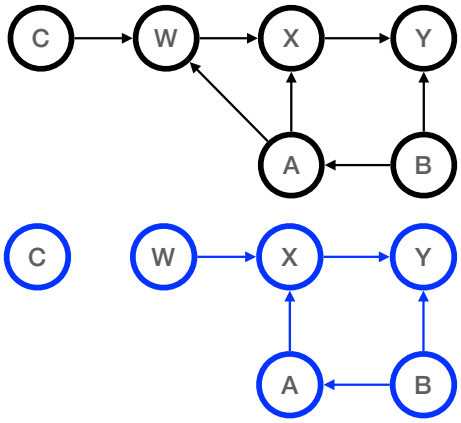
(d) **[2 points]** List the factors that would be constructed as the first step of variable elimination for the belief network from question (3c).
$\{f_1(C, D), f_2(A, B, C), f_3(A), f_4(B)\}$
(note: obviously the names and order of arguments needn't match)

(e) **[3 points]** List the new factor, and the operations used to create it, when the variable elimination algorithm eliminates B from the list of factors in question (3d).
$f_5(A, C) = \sum_B (f_2 \times f_4)$

(f) **[3 points]** Draw the graph for the post-intervention distribution for the query

$$P(Y \mid do(W = true))$$

for the following causal model:



4. **(Neural Networks)**

   (a) **[3 points]** What is the output of a *convolution* operation on the matrix $\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ using kernel $\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$?

   $$\begin{bmatrix} 3 & -1 \\ -2 & 3 \end{bmatrix}$$

   (b) **[3 points]** What is the output of a 2×2 *maximum pooling* operation on the matrix $\mathbf{X} = \begin{bmatrix} 8 & 1 & 2 \\ 3 & 4 & 1 \\ 2 & 3 & 9 \end{bmatrix}$?

   $$\begin{bmatrix} 8 & 4 \\ 4 & 9 \end{bmatrix}$$

   (c) **[4 points]** Consider the following set of training examples (x, y):

   $$\begin{aligned} E = \{&(\texttt{5}, \texttt{0000011111}), \\ &(\texttt{4}, \texttt{00001111}), \\ &(\texttt{3}, \texttt{000111}), \\ &(\texttt{2}, \texttt{0011}), \\ &(\texttt{1}, \texttt{01})\} \end{aligned}$$

   Assume that the inputs and outputs are represented as a single activation; that is, the input `5` represents the value 5 rather than a one-hot vector; the output `01` is a sequence of the value 0 followed by the value 1.

   Can a recurrent neural network that recurs though the **outputs only** achieve a perfect training loss on E with sufficient training and a sufficiently large hidden layer? Why or why not?

   No. A single output value does not provide a sufficient amount of information about the history of outputs to be able to tell how many outputs remain.

   Recurring through the hidden layer, the RNN could keep track of how many 0s are left to output and how many 1s are left to output. However, this is not an option when the only recurrence is through the outputs.