NAME: _____

AI2 Midterm Fall 2025 Answers

Each question is worth 10 points.

1. A robot moves on a grid where 20% of cells are dangerous. Its camera sensor detects danger correctly 90% of the time but falsely signals danger in 10% of safe cells. If the camera sensor signals danger, what is the probability the cell is actually dangerous?

```
P(D) = 0.2; P(-D) = 0.8
P(C|D) = 0.9; P(-C|D) = 0.1
P(C|-D) = 0.1; P(-C|-D) = 0.9

P(D|C) = P(C|D)P(D) / P(C)
       = P(C|D)P(D) / (P(C|D)P(D) + P(C|-D)P(-D))
       = (0.9 * 0.2) / ( 0.9 * 0.2 + 0.1 * 0.8)
       = 0.6923
```

2. Continuing from last question, the robot also has IR camera sensor. The IR camera sensor detects danger correctly 80% of the time but falsely signals danger in 20% of safe cells. If the camera *and* IR camera sensors both signal danger, what is the probability the cell is actually dangerous?

```
P(D) = 0.2; P(-D) = 0.8
P(C|D) = 0.9; P(-C|D) = 0.1
P(C|-D) = 0.1; P(-C|-D) = 0.9
P(IR|D) = 0.8; P(-IR|D) = 0.2
P(IR|-D) = 0.2; P(-IR|-D) = 0.8

P(D|C,IR) = P(C,IR|D) P(D) / P(C,IR)

Cannot be solved, since C and IR may not be independent.
We are not given P(C,IR|D).
```

3. Continuing from last question, the robot also has a tactile sensor that is independent of both camera sensors. The tactile sensor detects danger correctly 60% of the time but falsely signals danger in 30% of safe cells. If the camera *and* tactile sensors both signal danger, what is the probability the cell is actually dangerous?

```
P(D) = 0.2; P(-D) = 0.8
P(C|D) = 0.9; P(-C|D) = 0.1
P(C|-D) = 0.1; P(-C|-D) = 0.9
P(T|D) = 0.6; P(-T|D) = 0.4
P(T|-D) = 0.3; P(-T|-D) = 0.7

P(D|C,T) = P(C,T|D) P(D) / P(C,T)
    --since tactile sensor is ‘‘independent of both camera sensors’’
    --we can assume P(C,T) = P(C)P(T)
    = P(C|D)P(T|D)P(D) / P(C)P(T)
    = P(C|D)P(T|D)P(D) / (P(C|D)P(T|D)P(D) + P(C|-D)P(T|-D)P(-D))
    = (0.9*0.6*0.2) / (0.9*0.6*0.2 + 0.1*0.3*0.8)
    = 0.8182
```

4. Consider a 2-layer neural network with 2 inputs, 2 neurons in the hidden layer, and 1 output neuron. The network uses the ReLU activation function, defined as $\text{ReLU}(x) = \max(0, x)$. The network weights and biases are given as:

$$\text{Hidden layer weights: } W_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\text{Output layer weights: } W_2 = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad b_2 = 3$$

The input to the network is:

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Calculate the final output of the network. Show work (calculate the output of the hidden layer, apply ReLU, etc.)

```
A = w1 * x = [1*1+2*2; 3*1+4*2] = [5; 11]
B = A + b1 = [5+1; 11+2] = [6; 13]
C = relu(B) = [6; 13]
D = w2 * C = [1*6+2*13] = [6+26] = [32]
E = D + b2 = [32 + 3] = [35]
F = relu(E) = [35]
```

or

```
import torch
w1 = torch.tensor([[1.0,2.0],[3.0,4.0]])
w1.requires_grad = True
w2 = torch.tensor([1.0,2.0])
w2.requires_grad = True
b1 = torch.tensor([1.0,2.0])
b1.requires_grad = True
b2 = torch.tensor([3.0])
b2.requires_grad = True
x = torch.tensor([1.0, 2.0])
o = (w2.matmul( (w1.matmul(x) + b1).relu() ) + b2).relu()
35
```

5. Suppose we wish to minimize the output of the network. What would the gradients be for W_1, b_1, W_2, b_2?

```
A[0] = w1[0][0]*1 + w1[0][1]*2 = 1*1 + 2*2 = 5
A[1] = w1[1][0]*1 + w1[1][1]*2 = 3*1 + 4*2 = 11
B[0] = A[0] + b1[0] = 5 + 1 = 6
B[1] = A[1] + b1[1] = 11 + 2 = 13
D[0] = w2[0]*B[0]   = 1 * 6 = 6
D[1] = w2[1]*B[1]   = 2 * 13 = 26
E = D[0] + D[1] + b2[0] = 32 + 3 = 35

E.grad = 1
b2[0].grad = 1*E.grad = 1
D[0].grad = 1*E.grad = 1
D[1].grad = 1*D.grad = 1
w2[0].grad = B[0] * D[0].grad = 6
w2[1].grad = B[1] * D[1].grad = 13
B[0].grad = w2[0] * D[0].grad = 1
B[1].grad = w2[1] * D[1].grad = 2
b1[0].grad = 1 * B[0].grad = 1
b1[1].grad = 1 * B[1].grad = 2
A[0].grad = 1 * B[0].grad = 1
A[1].grad = 1 * B[1].grad = 2
w1[0][0].grad = x[0] * A[0].grad = 1 * 1 = 1
w1[0][1].grad = x[1] * A[0].grad = 2 * 1 = 2
w1[1][0].grad = x[0] * A[1].grad = 1 * 2 = 2
w1[1][1].grad = x[1] * A[1].grad = 2 * 2 = 4
```

or

```
o.backward()
w1.grad
tensor([[1., 2.],
        [2., 4.]])
w2.grad
tensor([ 6., 13.])
b1.grad
tensor([1., 2.])
b2.grad
tensor([1.])
```

6. From the below table, which attribute is most relevant for identifying fraudulent transactions. Explain your reasoning (Hint: you do not need to calculate information gain).

Transaction	Amount > $1000	International	Unusual Time	Fraud?
T1	Yes	No	No	No
T2	Yes	Yes	Yes	Yes
T3	No	No	Yes	No
T4	Yes	Yes	No	Yes
T5	No	Yes	Yes	No
T6	Yes	No	Yes	Yes

`The Amount > 1000 is the most relevant.`

7. Continuing from last question, draw a decision tree to identify fraudulent transactions.

```
[Amount>1000] -no-> [No]
              -yes->[International] -yes-> [Yes]
              -no->[Unusual Time] ->yes->[Yes]
                                  ->no->[No]
```

8. You are building a checkers playing program. You need to explore possible moves. You have a choice between depth-first and breadth-first search methods. Which one is appropriate. Explain why.

`Depth first upto depth-N, calculating minimax values.`

9. We train a model to predict fradulent transactions based on 10000 different features collected from everywhere. After training, the model achieves an accuracy of 98% on the training set. We put the model into production, and discover it performs poorly at actually predicting fradulent transactions. What went wrong? Provide three possible explanations.

1. It's possible the model memorized the training data.
2. It's possible that the training data and reality have different class probabilities. e.g. suppose training data has 50% fraud and 50% non-fraud training instances, but in reality, only 5% of transactions are fradulent.
3. It's possible that the model and data are OK, but reality of fradulent transactions has changed---making the model out of date.

10. Suppose we attempt to build a language model using a conditional probability table, where the probability of each word depends on all previous words:

$$P(\text{word}_n \mid \text{word}_1, \text{word}_2, \ldots, \text{word}_{n-1})$$

Assume the model is limited to a context window of 10,000 words and a vocabulary of 10,000 distinct words.

Explain the main problems and limitations that would arise with this design. Consider both computational and data requirements. Would this approach work in 20 years when computers are 1000000x faster and have 1000000x more memory? Why or why not?

There are several issues:
1. storage requirements are impossible (e.g. where would we store this probability table?)
2. compute requirements are impossible (e.g. how would we marginalize out the probability, even if we could store it somehow).
3. training data: there is not enough data in the universe to fill up that probability table.
4. it is unlikely to generalize well, except to recall exact sequences of words that fit the entire context exactly.