1. Taxonomy of AI [2 points]: For questions a and b, your answers should be ~2-4 sentences.

(a) How is deep learning different from: (1) machine learning, and (2) artificial intelligence?

Machine learning essentially is a branch of AI that focuses on machines learning patterns from data to make decisions and predictions.

Deep learning is a subset of Machine Learning that uses neural networks that have many layers (which is why it's called "deep") to learn from large amounts of data.

Artificial Intelligence is a more broad term that focuses on building systems that show human-like intelligence, taking reasoning into account along with problem solving and decision making.

(b) Give examples (not discussed in class/readings) of the following:

i. An artificial intelligence algorithm which is not a machine learning Algorithm.

An example could be a rule based system. This system works based on rules and logic given in the algorithm to solve problems, rather than learning patterns from data like in machine learning algorithms.
An example is a chess playing AI. In certain chess apps, you can play with "the computer" which has an algorithm that has been given a set of rules about piece movement and strategy.

ii. A machine learning algorithm which is not a deep learning algo-Rithm.

An example is the Support Vector Machines (SVMs) which can be used for classification tasks, particularly when dealing with high-dimensional data. It does not use does not use the layered structure which are used in deep learning algorithms.
An example would be a text classification algorithm. If you have files like law documents, newspapers and research articles, it can separate those into their respective categories.

2. Datasets and Tasks [3 points]: For questions a, b, and c, your answers should be ~2-4 sentences.

(a) Why are datasets used for deep learning often split into train and test Sets?

Datasets for deep learning are split into train and test data sets to make sure that the model can generalize effectively to new data. The train set is used to teach the model by adjusting its parameters, while the test set is used to figure out how well the model performs on data it hasn't seen before. Dividing the data helps prevent overfitting, where the model memorizes the training data rather than learning general patterns.

(b) Should there be any overlap of data between the training and test Datasets?

No, there should be no overlap between the training and test datasets. Overlapping data would let the model memorize the test data and could give false results based on how well it can work with new data.

(c) How are classification and regression different?

Classification and regression are both types of supervised learning but are different in the way they output the results.
Classification is used to predict a label or category, such as classifying emails as "spam" or "not spam."
Regression is used for predicting a continuous value, for example, forecasting house prices or stock prices.

3. Artificial Neurons:

(a) Learning a Perceptron [5 points]: Show the mathematical steps of learning a Perceptron model over one epoch, using the training data shown in Table 1. This is NOT a programming exercise. For full credit, you must include the mathematical steps used to derive the model parameters (bias and weights) and a table showing the model parameters after each training update. Recall that the perceptron model is of the form:

$$\hat{Y} = sign(\mathbf{w}^T\mathbf{x} + \mathbf{b})$$

where sign(◦) returns the sign of the input, i.e. sign(x) = x/|x| for x / = 0. For training, use a learning rate of 0.25, and the following model parameter initializations:

$$\mathbf{w}^T = [1, 1]$$

$$\mathbf{b} = -1$$

Each training sample should be processed one at a time, in the order given by Table 1 (i.e. no random sampling).

(b) Visualizing Model Testing [7 points]: Provide a 2-D scatter plot of the testing dataset from Table 1. Plot X1 along the "x-axis" and X2 along the "y-axis". Use two distinct markers (e.g. × and ◦) to indicate the labels (Y). Then, using the result from part (a), overlay the model's decision boundary as a solid line. Recall that the decision boundary is the set of x values which satisfy:

$$\mathbf{w}^T\mathbf{x} + \mathbf{b} = 0$$

Finally, report the model's predictions on the test data. You may hand-draw the plot and submit a picture in your PDF.

	Sample	X_1	X_2	Y
Training	1	1	1	-1
	2	-1	1	1
	3	-1	-1	1
	4	1	-1	-1
Test	1	0.5	2	1
	2	2	2	-1
	3	-0.5	0	1
	4	-0.5	-2	-1

Table 1: Training and test datasets.

3)

a) We use $w_j + \eta(\text{target}^{(i)} - \text{output}^{(i)})x_j^{(i)}$

Training Sample 1:

$$\hat{y} = \text{sign}(w^T x + b)$$
$$= \text{sign}\left([1\ \ 1]\begin{bmatrix}1\\1\end{bmatrix} - 1\right)$$
$$= \text{sign}(2-1) = \text{sign}(1) = 1$$

$$w_j^i = w_j^i + \eta(Y - \hat{Y})x_j^i$$
$$w_1 = 1 + 0.25(-1-1) * 1 = 0.5$$
$$w_2 = 1 + 0.25(-1-1) \times 1 = 0.5$$
$$b = w_0 = -1 + 0.25(-1-1) * 1 = -1.5$$

Training Sample 2:

$$\hat{y} = \text{sign}(w^T x + b)$$
$$= \text{sign}\left([0.5,\ 0.5]\begin{bmatrix}-1\\1\end{bmatrix} - 1.5\right)$$
$$= \text{sign}(-0.5+0.5-1.5)$$

$w_j = w_j + \eta(y - \hat{y})x_j$

$w_1 = 0.5 + 0.25(1-(-1)) \times 1 = 0$
$w_2 = 0.5 + 0.25(1-(-1) \times 1 = 1$
$b = w_0 = -1.5 + 0.25(1-(-1)) \times 1 = -1$

Training Sample 3:

$\text{sign}([0\ 1]\begin{bmatrix} -1 \\ -1 \end{bmatrix} - 1)$

$= \text{sign}(0 - 1 - 1)$
$= \text{sign}(-2) = -1$

$w_1 = 0 + 0.25(1-(-1)) \times (-1)) = -0.5$
$w_2 = 1 + 0.25(1-(-1)) \times (-1)) = 0.5$
$b = w_0 = -1 + 0.25(1-(-1)) \times (1)) = -0.5$

Training Sample 4:

$\text{sign}([-0.5, 0.5]\begin{bmatrix} 1 \\ -1 \end{bmatrix} - 0.5)$

$= \text{sign}(-0.5 - 0.5 - 0.5)$
$= \text{sign}(-1.5)$
$= -1$

$w_1 = -0.5 + 0.25(-1-(-1)) \times 1 = 0.5$

$w_2 = 0.5 + 0.25(-1-(-1)) \times (-1) = 0.5$

$b = w_0 = -0.5 + 0.25(-1-(-1)) \times 1$

$= -0.5$

Sample	x_1	x_2	y	$w_0(0)$	w_1	w_2	$\hat{y}$
1	1	1	-1	-1	1	1	1
2	-1	1	1	-1.5	0.5	0.5	-1
3	-1	-1	1	-1	0	1	-1
4	1	-1	-1	-0.5	-0.5	0.5	-1

$\therefore$ $w^T = [-0.5, 0.5]$ and $b = -0.5$

for decision boundary, we use

$$w^T x + b = 0$$

$w^T = [-0.5, 0.5]$

$b = -0.5$

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Therefore,

$$[-0.5, 0.5] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (-0.5) = 0$$

Dividing by (-0.5) on both sides,

$$x_1 - x_2 + 1 = 0$$

$$x_1 - x_2 = -1$$

when $x_1 = 0$

$$0 - x_2 = -1$$

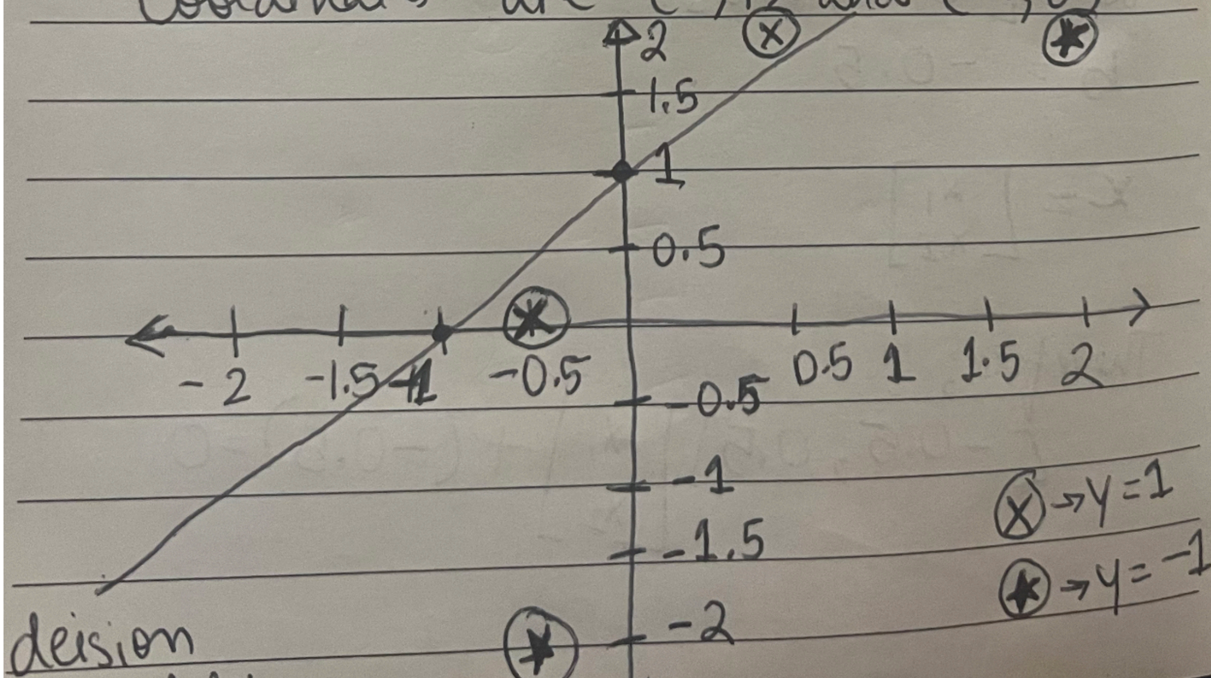
$$x_2 = 1$$

when $x_2 = 0$,

$$x_1 = 0 + -1$$

$$x_1 = -1$$

Coordinates are $(0,1)$ and $(-1,0)$



(c) Model Evaluation [4 points]: Evaluate the prediction results for the final model on the test data by computing the confusion matrix, precision, recall, and accuracy.

c)

		Actual	
		1	-1
Predicted	1	1	0
	-1	1	2

$$\text{Accuracy} = \frac{1+2}{1+0+1+2} = \frac{3}{4}$$

$$\text{Precision} = \frac{1}{1+0} = 1$$

$$\text{Recall} = \frac{1}{1+1} = \frac{1}{2}$$

(d) Model Analysis [4 points]: Discuss what insights are gained about the model's performance by examining each of the four evaluation results: confusion matrix, accuracy, precision, and recall.

From the table we can see tha all labels were predicted correctly except for 1, which leads to a 75% accuracy. But because the model did not predict all labels correctly that were actually 1, we can say that the recall is 50%. Also the model has a precision of 1 which means that all the predictions that are 1 are correct. What we can derive from the model is that it might incorrectly label samples of 1 as -1 but it will be conservative and always correctly predict labels that are -1 as -1.

(e) Extra Credit [2.5 points]: Analysis of Learning Trajectory Provide a 2-D scatter plot of the entire dataset (train and test combined), using two distinct markers to indicate the labels. Next, overlay the model's final/initial (i.e. after/before training the perceptron in part a) decision boundaries with solid/dotted lines, respectively. Then, conjecture a possible "optimal" solution for w and b which results in a model that correctly predicts all labels. Note that there are many possible such solutions. Overlay your chosen decision boundary as a dashed line on your plot. Finally, compare the model's initial and final decision boundaries to your chosen solution. Did the training process move the model's decision boundary closer to an "optimal" solution?

The training process definitely did move the model's decision boundary closer to an "optimal" solution. The final decision boundary is more closely related to my "optimal" solution than it is related to the initial decision boundary.

(as seen in picture below)

e)

	Sample	x_1	x_2	y
Training	1	1	1	-1
	2	-1	1	1
	3	-1	-1	1
	4	1	-1	-1
Test	1	0.5	2	1
	2	2	2	-1
	3	-0.5	0	1
	4	-0.5	-2	-1

final:

$x_1 - x_2 + 1 = 0$

Initial:

$x_1 + x_2 - 1 = 0$