Question 1

A. The Turing test is a way to determine if a machine is intelligent enough to deceive a human.
A machine pass the Turing test if after having a conversation with a human, that human didn't figure out that he was talking to an AI and not a human.

B. The output of a classification problem is said to be discrete, the input has a specific number of possible answers. regression doesn't have a certain set of labels, the output is continuous, it cannot be classified in a set of classes.

C. Machine learning has 4 basic components:
   1. Dataset: A dataset is the set of data that the machine will use to find its function. It is a set of points (x,y) where x is the input vector and y is the desired output vector for that x.
   2. Machine Learning Model: After inputing a matrix of input, plotting
   3. Objective function: It is also known as the error function, it is a way to measure how accurate the model is. Its a function that measure how far the prediction is from the actual output.
   4. Optimization Algorithm: Series of algorithms that we can use to minimize the objective function. Gradient descent, Adam, Adagrad and RMSProp are examples of optimization algorithm.

D. Supervised learning learns from labeled input and output while unsupervised doesn't need the labels. Unsupervised learning will learn by analyzing and observing the structure of the data. The datasets of unsupervised learning do not contain any output y.
An example of supervised learning would be to input a training set that contains pictures of dogs and cats AND their labels.
An example of unsupervised learning would be to input a "training" set that contains dogs and cats pictures but no label. The algorithm will then cluster the inputs together by analyzing the similarities between pictures.

E. A model that underfit is a model that is not capable to capture complex data. The model is said to have high bias which leads to inaccurate result for both training set and test set. The capacity of an undercutting model will be low, the hypothesis space won't be rich enough in term of function to correctly model the complexity of our sets.

   A model that overfit is a model that will behave as a memory on the training set. This happens when there is too much "freedom" when training the model, leads to unrealistic fitting, a fitting that is over complex. Overfitting models have high variance which lead to bad results on the test set.

F. When training a machine learning model, the learning rate refers to the "steps" the model is going to take when minimizing the cost functions. You need to find the just middle as if the learning rate is too small, you will need many iterations to find the minimum and if your learning rate is too large, you will have drastic updates that leads to a divergent behaviors.

G. Gradient descent and Stochastic Gradient descent are two algorithms that allows us to find the minima. In the classic GD, we update the weight only after we finish an epoch, meaning only after we go through all the data points. This will result in a slow "walk" to the the minima since we are only taking a "step" after processing the whole dataset.
In the other hand, the Stochastic GD will update the weight after each data point or batch is processed. The batches needs to be smaller than the actual data set. This will result in way less iterations to reach the minima.
SGD will converge much faster but it will be less minimized in comparison to the Classic GD.

Question 2 (a) Consider a linear regression problem with the absolute error (or L1 error) function. The error associated with a single training sample with input $\mathbf{x}$ and target value y is given as:

$$J(\mathbf{w}) = |y - \mathbf{w}^T\mathbf{x}| = |y - (w_0x_0 + \ldots + w_ix_i + \ldots + w_nx_n)| \quad (1)$$

You are tasked with developing a gradient-descent learning rule for the above objective function. Your rule should be in the form:

$$w_i \leftarrow w_i - \eta??? \quad (2)$$

(b) Assume you have a problem with a dataset of two data points $\mathbf{x_1}$ = [0.9, 0.8, 0.4, 0.1] and $\mathbf{x_2}$ = [0.5, 0.2, 0.9, 0.8] and targets y_1 = 0.2 and y_2 = 0.9. You aim to train a linear regression model using the absolute error function, with the initial weights $\mathbf{w}$ =[0.5, 0.5, 0.5, 0.5] and a learning rate of 0.1 (assume no bias weight). Assume that the weights are updated after processing each data point, and that your model is trained with two epochs (meaning that your weights are updated 4 times). In each step of the training process, compute the output, the error, and the updated weights. Show your work.

a) let $z = y - w^Tx$; $J(w) = |z|$

$$J(w) = \begin{cases} y - w^Tx & y - w^Tx > 0 \\ -y + w^Tx & -y + w^Tx < 0 \end{cases}$$

$y - w^Tx > 0$

$$\nabla J(w_i) = \frac{d(y - w_ix_i)}{dw_i} = -x_i$$

$-y + w^Tx < 0$

$$\nabla J(w_i) = \frac{d(-y + (w_ix_i))}{dw_i} = x_i$$

$$\nabla J(w_i) = \begin{cases} -x_i & y - w^Tx > 0 \\ x_i & -y + w^Tx < 0 \end{cases}$$

$$w_i \leftarrow w_i - \rho \nabla J(w_i)$$

b) EPOCH 1, itteration 1

$\hat{y}_1 = 0.5(0.9) + 0.5(0.8) + 0.5(0.4) + 0.5(0.1) = 1.1$

$ERR = |y - \hat{y}| = 0.9$

new weight $= w_i - \eta(x_i)$

$$\begin{bmatrix} 0.5 - 0.1(0.9) \\ 0.5 - 0.1(0.8) \\ 0.5 - 0.1(0.4) \\ 0.5 - 0.1(0.1) \end{bmatrix} = \begin{bmatrix} 0.41 \\ 0.42 \\ 0.46 \\ 0.49 \end{bmatrix}$$ New weights

EPOCH 1, itteration 2

$\hat{y}_2 = 0.41(0.5) + 0.42(0.2) + 0.46(0.9) + 0.49(0.8) = 1.095$

$ERR = |y - \hat{y}| = |0.9 - 1.095| = 0.195$

$$\begin{bmatrix} 0.41 - 0.1(0.5) \\ 0.42 - 0.1(0.2) \\ 0.46 - 0.1(0.9) \\ 0.49 - 0.1(0.8) \end{bmatrix} = \begin{bmatrix} 0.36 \\ 0.4 \\ 0.37 \\ 0.41 \end{bmatrix}$$ New weight

EPOCH 2, Itteration 1

$$\hat{Y}_1 = 0.36(0.9) + 0.4(0.8) + 0.37(0.4) + 0.41(0.1) = 0.833$$

$$ERR = |y - \hat{y}| = |0.2 - 0.833| = 0.633$$

$$\begin{bmatrix} 0.36 - 0.1(0.9) \\ 0.4 - 0.1(0.8) \\ 0.37 - 0.1(0.4) \\ 0.41 - 0.1(0.1) \end{bmatrix} = \begin{bmatrix} 0.27 \\ 0.32 \\ 0.33 \\ 0.4 \end{bmatrix}$$ New weight

EPOCH 2, Itteration 2

$$\hat{Y}_2 = 0.27(0.5) + 0.32(0.2) + 0.33(0.9) + 0.4(0.8) = 0.816$$

$$ERR = |y - \hat{y}| = |0.9 - 0.816| = 0.084$$

$$\begin{bmatrix} 0.27 - 0.1(0.5) \\ 0.32 - 0.1(-0.2) \\ 0.33 - 0.1(-0.9) \\ 0.4 - 0.1(-0.8) \end{bmatrix} = \begin{bmatrix} 0.32 \\ 0.34 \\ 0.42 \\ 0.48 \end{bmatrix}$$ New weight

Question 3 Consider a classification problem with three possible classes: A, B, and C. For a batch of 5 sample data points, your neural network produces the following outputs. Each output represents a probability distribution over the 3 classes.

(a) Using categorical cross-entropy, calculate the **average** loss for the batch of five samples. Show your work.

Sample	Output			True Label
	Class A	Class B	Class C	
1	0.7	0.2	0.1	B
2	0.1	0.2	0.7	A
3	0.3	0.6	0.1	B
4	0.8	0.1	0.1	C
5	0.2	0.3	0.5	A

A x
C x
B ✓
A x
C x

Table 1: Table of Predicted Probabilities (Output) and True Labels

$$\begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.7 \\ 0.3 & 0.6 & 0.1 \\ 0.8 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$CCE = -\frac{1}{5}\Big(0\ln(0.7) + 1\ln(0.2) + 0\ln(0.1) + 1\ln(0.1) + 0\ln(0.2) + 0\ln(0.7) + 0\ln(0.3) + 1\ln(0.6) + 0\ln(0.1) + 0\ln(0.8) + 0\ln(0.1) + 1\ln(0.1) + 1\ln(0.2) + 0\ln(0.3) + 0\ln(0.5)\Big)$$

$$= 1.67$$

(b) Assuming the model chooses the class of the highest probability as the output class, what is the classification accuracy of the classification model represented in the table? Show your work.

A	B	C	
0.7	0.2	0.1	B ✗
0.1	0.2	0.7	A ✗
0.3	0.6	0.1	B ✓
0.8	0.1	0.1	C ✗
0.2	0.3	0.5	A ✗

$$ACC = \frac{N_{corr}}{N_{tot}} = \frac{1}{5} = 0.2$$

Question 4 You are given with a dataset of the following eight (x, y) data points:

[(0.5,0.57), (0.98,0.5), (1,0.98), (1.4,0.89), (5,-0.75), (6.1,0), (5.5,-0.45), (6.5,0.2)]

(a) Using Mean Squared Error (MSE) as your criteria, determine which of the following candidate functions best fits the given data: $f_1(x) = e^x$, $f_2(x) = x^2$, $f_3(x) = cos(x)$, and $sin(x)$. (note: for cos(x) and sin(x), the inputs are in radians). Show your work.

(b) Inspect your results visually. Plot the data points and show how they fit the curves of the different candidate functions. Elaborate on your results.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

$$MSE = \frac{1}{8}\left((0.57 - e^{0.5})^2 + (0.5 - e^{0.98})^2 + (0.98 - e^{1})^2 + (0.89 - e^{1.4})^2 + (-0.75 - e^{5})^2 + (0 - e^{6.1})^2 + (-0.45 - e^{5.5})^2 + (0.2 - e^{6.5})^2\right)$$

$$= 90412.45$$

$$MSE = \frac{1}{8}\left((0.57 - 0.5^2)^2 + (0.5 - 0.98^2)^2 + (0.98 - 1^2)^2 + (0.89 - 1.4^2)^2 + (-0.75 - 5^2)^2 + (0 - 6.1^2)^2 + (-0.45 - 5.5^2)^2 + (0.2 - 6.5^2)^2\right)$$

$$= 594.97$$

$$MSE = \frac{1}{8}\left((0.57 - \cos(0.5))^2 + (0.5 - \cos(0.98))^2 + (0.98 - \cos(1))^2 + (0.89 - \cos(1.4))^2 + (-0.75 - \cos(5))^2 + (0 - \cos(6.1))^2 + (-0.45 - \cos(5.5))^2 + (0.2 - \cos(6.5))^2\right)$$

$$= 0.5988$$

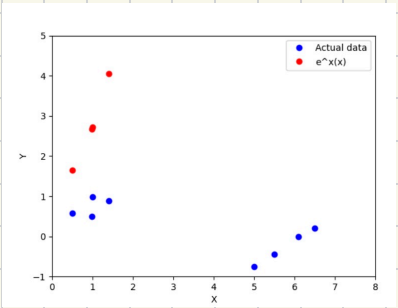
MSE $= \frac{1}{8}\left((0.57-\sin(0.5))^2+(0.5-\sin(0.98))^2+(0.98-(1))^2+\right.$

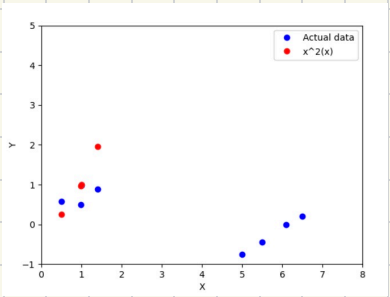
$\left.(0.89-\sin(1.4))^2+(-0.75-\sin(5))^2+(0-\sin(6.1))^2+(-0.45-(5.5))^2+(0.2-(6.5))^2\right)$

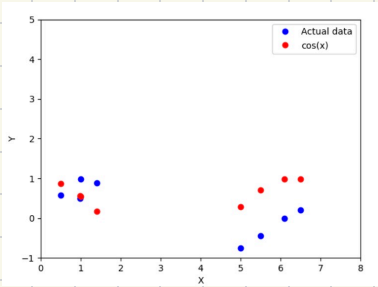
$= 0.03601$

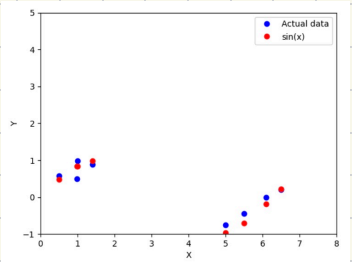
Sin(x) is the function that best fit the data

b)









As seen in part a, the sin function is the one with the lowest MSE
We can Confirm that visually after plotting our 4 functions against the given dataset.
The sin one is the one closest to our points.

Question 5 Consider a min-dataset of two data points $\mathbf{x_1} = [5, 3, 1, 3]$ and $\mathbf{x_2} = [1, 5, 1, 20]$ and targets $y_1 = 1$ and $y_2 = 0$. Your goal is to train a logistic regression model with the binary cross-entropy (BCE) loss function, a learning rate of 0.2, and all initial weights having a value of 0.5. You are to consider biases in your calculations.

(a) Given input $\mathbf{x_i}$, what is the output function of a logistic regression model? What is the BCE loss function? What is the gradient of the BCE function?

(b) Train a logistic regression model using the provided dataset. Use a batch of size 2 to train your model in 2 epochs (i.e. each weight is updated twice). Show your work.

a) OUTPUT: $\sigma(z)$ where $z = w^T x_i$

BCE Loss: $-\frac{1}{N}\sum_{i=1}^{N} y_{i1}\ln(p_{i1}) + (1-y_{i1})\ln(1-p_{i1})$

$\nabla BCE(w) = \frac{1}{N}\sum_{i=1}^{N}[\sigma(w^T x_i) - y_i]x_i$

$\begin{bmatrix} \nabla J(w_0) \\ \nabla J(w_1) \\ \nabla J(w_2) \end{bmatrix}$

b) $\underline{x_1}$

$z_1 = 6.5$

$\hat{y}_1 = \sigma(6.5) = 0.9984988$

$\underline{x_2}$

$z_1 = 14$

$\hat{y}_2 = \sigma(14) = 0.9999992$

$\underline{\nabla J}$

$\frac{1}{2}([0.9984988-1]1 + [0.9999992-0]1) = 0.4992549$

$\frac{1}{2}([0.9984988-1]5 + [0.9999992-0]1) = 0.4962$

$\frac{1}{2}([0.9984988-1]3 + [0.9999992-0]5) = 2.4977746$

$\frac{1}{2}([0.9984988-1]1 + [0.9999992-0]1) = 0.4992549$

$\frac{1}{2}([0.9984988-1]3 + [0.9999992-0]20) = 9.9977402$

$$\begin{matrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{matrix}\begin{bmatrix} 0.5-0.2(0.499249) \\ 0.5-0.2(0.4962) \\ 0.5-0.2(2.497746) \\ 0.5-0.2(0.499249) \\ 0.5-0.2(9.99774402) \end{bmatrix} \quad \begin{bmatrix} 0.4001507 \\ 0.40076 \\ 0.0004508 \\ 0.4001507 \\ -1.49954804 \end{bmatrix}$$

X_1

$z_1 = -1.69319132$

$\hat{y}_1 = 0.1553566$

X_2

$z_1 = -28.78764\,64$

$\hat{y}_2 = 3.145609 \times 10^{-13}$

$$\frac{1}{2}([0.1553566 - 1]1 + [3.145459 - 0]1) = -0.4223$$

$$\frac{1}{2}([0.1553566 - 1]5 + [3.145459 - 0]1) = -2.1116$$

$$\frac{1}{2}([0.1553566 - 1]3 + [3.145459 - 0]5) = -1.2669$$

$$\frac{1}{2}([0.1553566 - 1]1 + [3.145459 - 0]1) = -0.4223$$

$$\frac{1}{2}([0.1553566 - 1]3 + [3.145459 - 0]20) = -1.2669$$

$$
\begin{matrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{matrix}
\begin{bmatrix}
0.4001507 - 0.2(-0.4223) \\
0.40076 - 0.2(-2.1116) \\
0.0004508 - 0.2(-1.2669) \\
0.4001507 - 0.2(-0.4223) \\
-1.49954804 - 0.2(-1.2669)
\end{bmatrix}
=
\begin{bmatrix}
0.4846 \\
0.82308 \\
0.2538308 \\
0.4846102 \\
-1.24616804
\end{bmatrix}
$$