Kahbod Aeini - 98101209

1. a) First we need to discuss how the problem must be modelled.
Inscriptions can easily be represented by a 9 length binary vector where each element indicates the corresponding pixel whether gleams or not.

By the recent modelling of the problem we reach the following table as our training dataset.

Letter	Pi_0	Pi_1	Pi_2	Pi_3	Pi_4	Pi_5	Pi_6	Pi_7	Pi_8
A	1	0	0	0	1	1	0	0	0
A	0	1	1	0	1	1	1	0	0
A	0	1	0	0	1	1	0	0	0
B	0	0	1	0	1	0	0	1	0
B	0	0	1	1	0	0	1	1	0
B	1	1	0	1	0	0	1	1	0
?	0	1	0	0	1	1	1	1	0

Now we elicit the below conditional probability table from the dataset.

$P(Pi_0 = 1 \mid A)$	1/3		$P(Pi_0 = 1 \mid B)$	1/3
$P(Pi_1 = 1 \mid A)$	2/3		$P(Pi_1 = 1 \mid B)$	1/3
$P(Pi_2 = 1 \mid A)$	1/3		$P(Pi_2 = 1 \mid B)$	2/3
$P(Pi_3 = 1 \mid A)$	0/3		$P(Pi_3 = 1 \mid B)$	2/3
$P(Pi_4 = 1 \mid A)$	3/3		$P(Pi_4 = 1 \mid B)$	1/3
$P(Pi_5 = 1 \mid A)$	3/3		$P(Pi_5 = 1 \mid B)$	0/3
$P(Pi_6 = 1 \mid A)$	1/3		$P(Pi_6 = 1 \mid B)$	2/3
$P(Pi_7 = 1 \mid A)$	0/3		$P(Pi_7 = 1 \mid B)$	3/3
$P(Pi_8 = 1 \mid A)$	0/3		$P(Pi_8 = 1 \mid B)$	0/3

Since in the table there exists a couple of zero and one probability in the dataset, for preventing overfitting and cancelling possible cases, Laplace1 smoothing needs to be done.

So after Laplace1 method, conditional probability table updates to the below table.

$P(Pi_0 = 1 \mid A)$	2/5		$P(Pi_0 = 1 \mid B)$	2/5
$P(Pi_1 = 1 \mid A)$	3/5		$P(Pi_1 = 1 \mid B)$	2/5
$P(Pi_2 = 1 \mid A)$	2/5		$P(Pi_2 = 1 \mid B)$	3/5
$P(Pi_3 = 1 \mid A)$	1/5		$P(Pi_3 = 1 \mid B)$	3/5
$P(Pi_4 = 1 \mid A)$	4/5		$P(Pi_4 = 1 \mid B)$	2/5
$P(Pi_5 = 1 \mid A)$	4/5		$P(Pi_5 = 1 \mid B)$	1/5
$P(Pi_6 = 1 \mid A)$	2/5		$P(Pi_6 = 1 \mid B)$	3/5
$P(Pi_7 = 1 \mid A)$	1/5		$P(Pi_7 = 1 \mid B)$	4/5
$P(Pi_8 = 1 \mid A)$	1/5		$P(Pi_8 = 1 \mid B)$	1/5

Now we calculate the probability of $P(A \mid Pi\ =\ [0, 1, 0, 0, 1, 1, 1, 1, 0])$ and $P(B \mid Pi\ =\ [0, 1, 0, 0, 1, 1, 1, 1, 0])$ using the calculated probability table.

$P(A \mid Pi\ =\ [0, 1, 0, 0, 1, 1, 1, 1, 0]) = P(A) \times P(Pi_0 = 0 \mid A) \times P(Pi_1 = 1 \mid A) \times P(Pi_2 = 0 \mid A)$
$\times\ P(Pi_3 = 0 \mid A) \times P(Pi_4 = 1 \mid A) \times P(Pi_5 = 1 \mid A) \times P(Pi_6 = 1 \mid A) \times P(Pi_7 = 1 \mid A)$
$\times\ P(Pi_8 = 0 \mid A) = \frac{1}{2} \times \frac{3}{5} \times \frac{3}{5} \times \frac{3}{5} \times \frac{4}{5} \times \frac{4}{5} \times \frac{4}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{4}{5} = \frac{13{,}824}{3{,}906{,}250}$

$P(B \mid Pi\ =\ [0, 1, 0, 0, 1, 1, 1, 1, 0]) = P(B) \times P(Pi_0 = 0 \mid B) \times P(Pi_1 = 1 \mid B) \times P(Pi_2 = 0 \mid B)$
$\times\ P(Pi_3 = 0 \mid B) \times P(Pi_4 = 1 \mid B) \times P(Pi_5 = 1 \mid B) \times P(Pi_6 = 1 \mid B) \times P(Pi_7 = 1 \mid B)$
$\times\ P(Pi_8 = 0 \mid B) = \frac{1}{2} \times \frac{3}{5} \times \frac{2}{5} \times \frac{3}{5} \times \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{3}{5} \times \frac{4}{5} \times \frac{4}{5} = \frac{5{,}184}{3{,}906{,}250}$

Accordingly the letter has a greater chance to be an A than B. So we classify the letter as A.

b) In the decision tree we separate nodes based on a feature in each level. The main crisis in generating a proper decision tree is to choose which feature must be the separator on each level. This subproblem usually is solved by the amount of difference a feature can make on each level of the tree, which can be elected more efficiently and optimum by methods like Conditional Entropy and Information Gain.

2. Note x^j as jth node in the graph problem and each node is represented in an d dimension feature space (R^d), y^j as the class of the jth node, w^k the hyperplane calculated in the kth step, w^* an 'Oracle' weight vector that correctly separates the training examples.
The quantity γ is introduced in the assumption as a place-holder for the minimum value of the $y^i(x^j.w^*)$. We also assume R as $R \in R^d$ where for $i \in \{1,\ 2,\ .\ .\ .,\ n\} \Rightarrow ||x^i|| \leq R$.

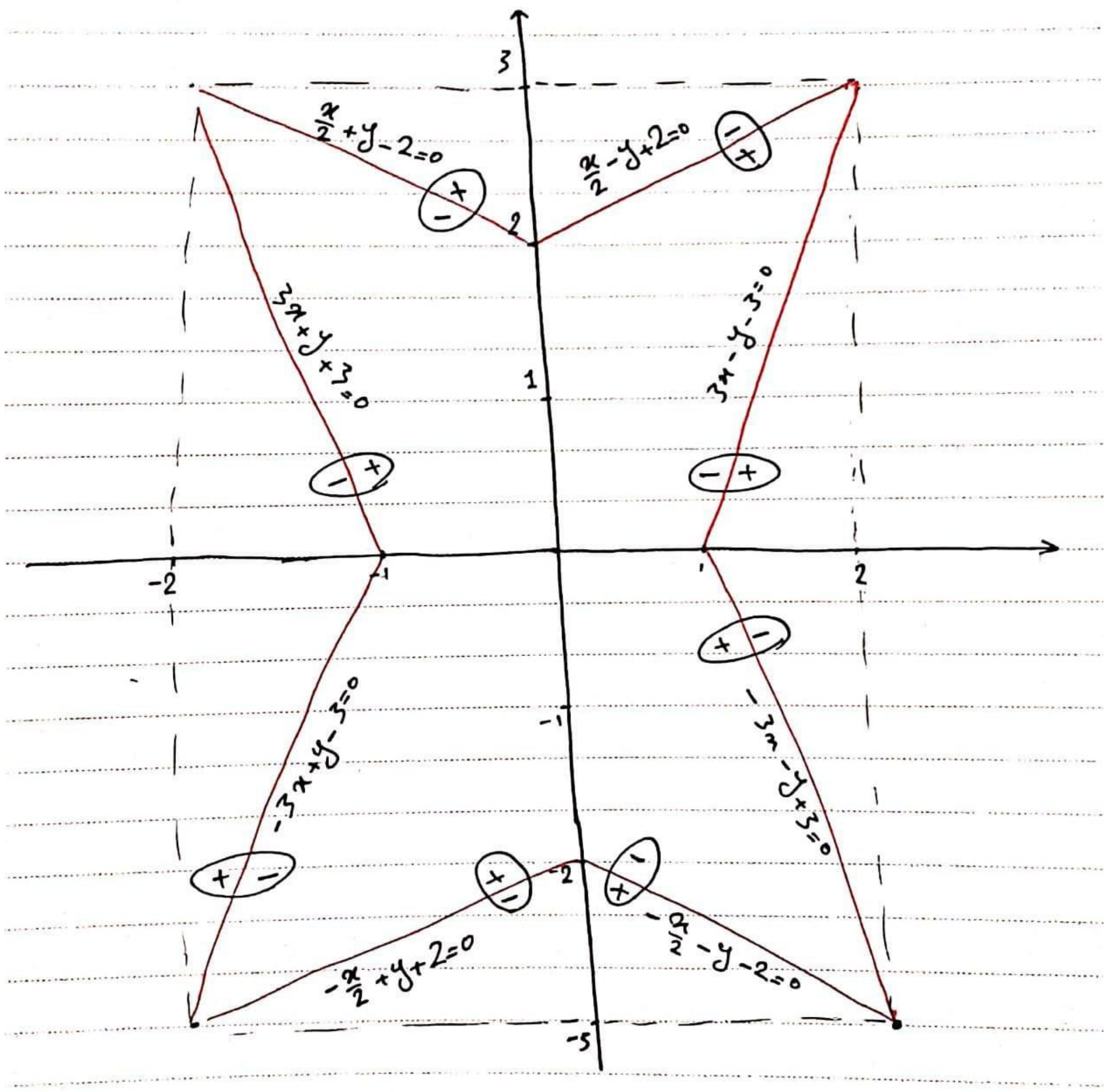
Note that $w^1 = 0$, and for $k \geq 1$, note that if x^j is the misclassified point during iteration k, we have
$w^{k+1}.w^* = (w^k + y^j x^j).w^* = w^k.w^* + y^j(x^j.w^*) > w^k.w^* + \gamma$

It follows by induction that $w^{k+1}.w^* > k\gamma$. Since $w^{k+1}.w^* \leq ||w^{k+1}||.||w^*|| = ||w^{k+1}||$, we get $||w^{k+1}|| > k\gamma$. **<u>1</u>**

Now we try to obtain an upper bound. We argue that
$||w^{k+1}||^2 = ||w^k + y^j x^j||^2 = ||w^k||^2 + ||y^j x^j||^2 + 2(w^k.x^j)y^j = ||w^k||^2 + ||x^j||^2 + 2(w^k.x^j)y^j$
$\leq ||w^k||^2 + ||x^j||^2 \leq ||w^k||^2 + R^2$
from which it follows by induction that $||w^{k+1}||^2 \leq kR^2$ **<u>2</u>**

Together **<u>1</u>** and **<u>2</u>** yield $k^2\gamma^2 < ||w^{k+1}||^2 \leq kR^2$ which implies that $k < \frac{R^2}{\gamma^2}$.
So we proved that the number of iterations has an upper bound that means the perceptron algorithm always converges to the answer with a finite number of steps.

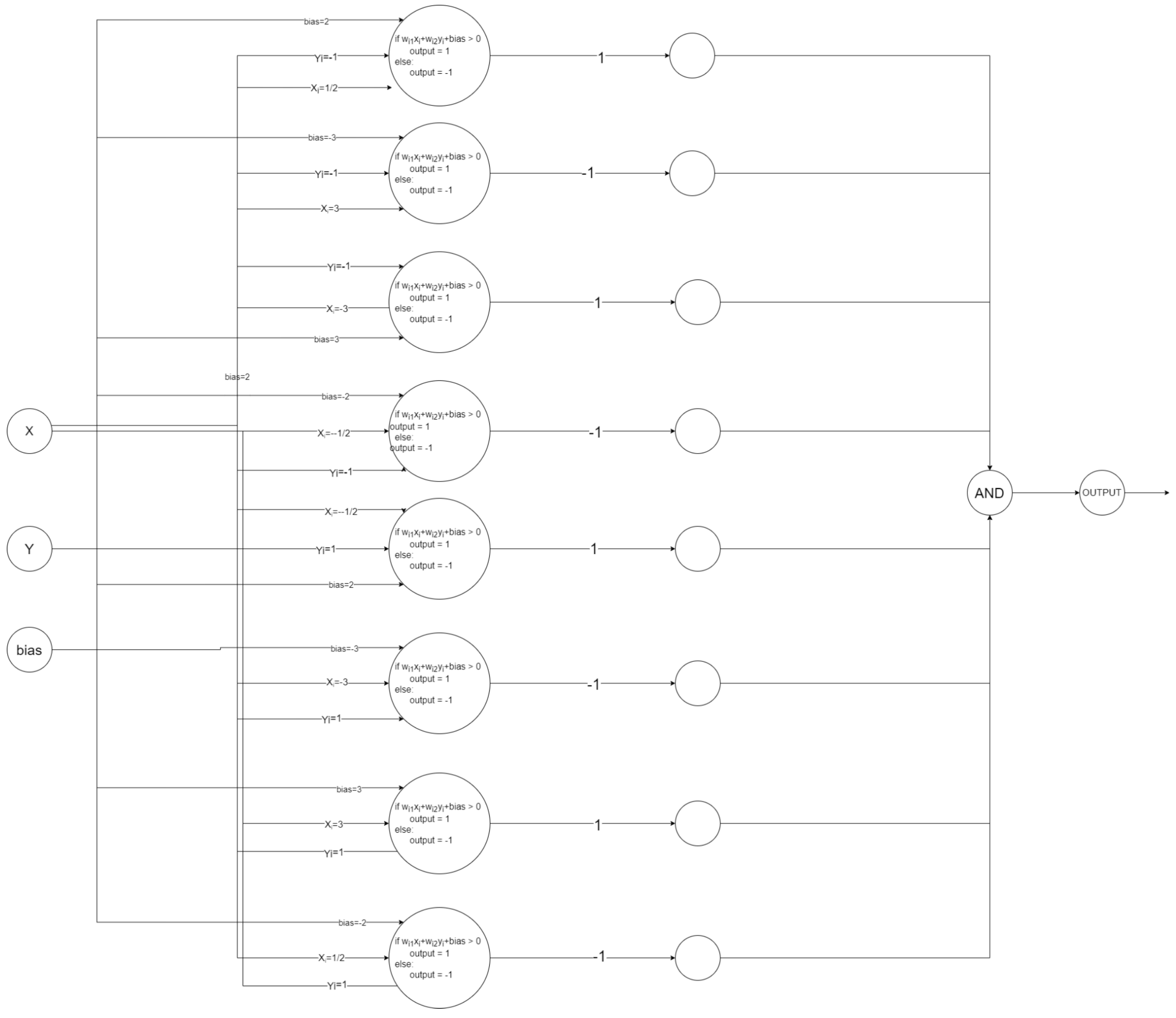
3. First we discuss how a point would be in the given polygon. Consider the below schema.



Here we calculated the linear equation of each edge. And the signs on each side of the edges indicate the sign of the equation's output based on the given x and y in the areas separated by the edge.

So clearly only points (e.g. p(x, y)) which have the whole 8 equations output signs inside the polygon can settle inside the polygon.

Now we easily model the problem and scenario checking approach using a neural network.


X
Y
bias
bias=2
Yi=-1
Xi=1/2
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
1
bias=-3
Yi=-1
Xi=3
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
-1
Yi=-1
Xi=-3
bias=3
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
1
bias=2
bias=-2
Xi=--1/2
Yi=-1
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
-1
Xi=--1/2
Yi=1
bias=2
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
1
bias=-3
Xi=-3
Yi=1
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
-1
bias=3
Xi=3
Yi=1
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
1
bias=-2
Xi=1/2
Yi=1
if wi1xi+wi2yi+bias > 0 output = 1 else: output = -1
-1
AND
OUTPUT

Clearly if the output equals 1, p(x, y) settles inside the polygon and if it is -1, the point is outside the polygon.