HOMEWORK 04

Courses `CSC14003`: Intro to Artificial Intelligence
18CLC6, FIT - HCMUS.
`09/09/2020`

This is a team homework for 2 students, but for some reason - can not form a team, so I do it as a individual assignment :

- `18127231`: Đoàn Đình Toàn (GitHub: [@t3bol90](https://github.com/t3bol90))

Problem:

Problem 1. (3.0pts)

A robot in a lumber yard learns to discriminate Oak wood from Pine wood. It learns a decision tree classifier from the examples shown aside:

No.	Density	Grain	Hardness	Class
1	Heavy	Small	Hard	Oak
2	Heavy	Large	Hard	Oak
3	Heavy	Small	Hard	Oak
4	Light	Large	Soft	Oak
5	Light	Large	Hard	Pine
6	Heavy	Small	Soft	Pine
7	Heavy	Large	Soft	Pine
8	Heavy	Small	Soft	Pine

Table 1.1

a. (2.0pts) Build a decision tree using the ID3 Decision Tree Induction algorithm.

b. (1.0pt) Classify these new examples as Oak or Pine using the decision tree above.

[Density=Light, Grain=Small, Hardness=Hard]
[Density=Light, Grain=Small, Hardness=Soft]

Problem 2. (2.0pts)

Consider the following training dataset, in which Transportationis the target attribute. Show calculations to choose an attribute for the root node of the ID3 decision tree

Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Table 2.1

Problem 3.(2.0pts):

Consider the data setshown aside. A and B are numerical attributes and Z is a Boolean classification

A	B	Z
1	2	T
2	1	F
3	2	T
1	1	F

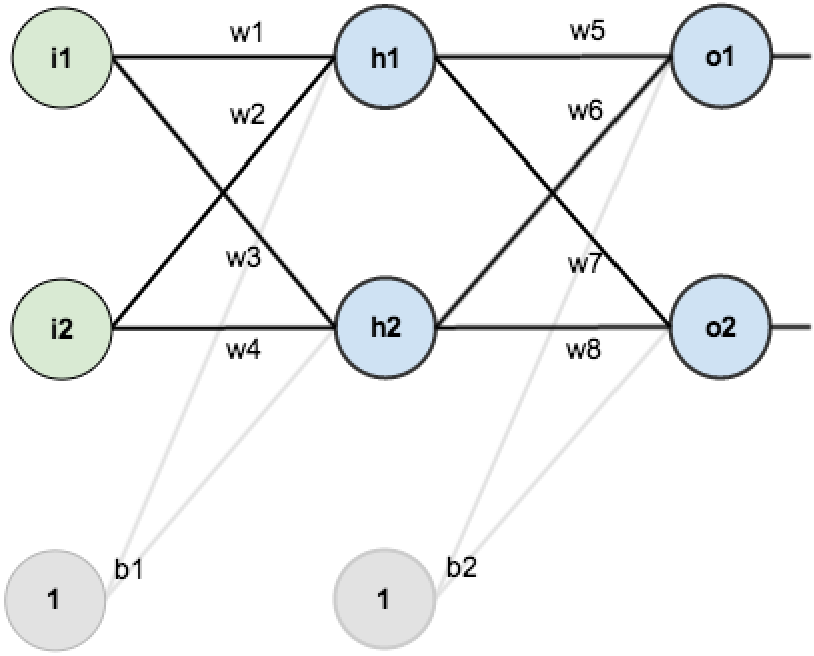
Table 3.1

a. Let P be the perceptron that has two input neurons for the two atttributes, A and B, and one bias of constant value 1. The correponding weights are `wA = 2`, `wB= 1`, and `wo= -4.5` (for bias). What is the total error after training one epoch?

b. Provide your analysis to find a set of weights and a threshold that categorizes all this data correctly.

Problem 4. (3.0pts):

Given a neural network with two inputs, two hidden neurons, two output neurons, as shown below. Additionally, in the hidden and output layers, each of which will include a bias that has a constant output value of 1.



- **Learning rate 0.5**
- **Input values: i1=0.05 i2=0.10**
- **Target values: t1=0.01 t2=0.99**
- **Bias values: b1=0.35 b2=0.60**
- **Initial weight:**

w1	**0.15**
w2	**0.20**
w3	**0.25**
w4	**0.30**
w5	**0.40**
w6	**0.45**
w7	**0.50**
w8	**0.55**

Present all calculations required to perform the backpropagation once (i.e. one forward pass and one backward pass) on the given neural network in the following cases

a. Ignore all biases.

b. Take into account all biases.

Answers:

Problem 1:

a.

With the Entropy formula:

$$H(S) = \Sigma_{x \in X} - p(x) \log_2 p(x)$$

and the Information gain:

$$IG(S, A) = H(S) - \Sigma_{t \in T} p(t) H(t) = H(S) - H(S|A)$$

`ID3` algorithms pseudo code[^(1)]:

```
ID3 (Examples, Target_Attribute, Attributes)
    Create a root node for the tree
    If all examples are positive, Return the single-node tree Root, with label =
+.
    If all examples are negative, Return the single-node tree Root, with label =
-.
    If number of predicting attributes is empty, then Return the single node tree
Root,
    with label = most common value of the target attribute in the examples.
    Otherwise Begin
        A ← The Attribute that best classifies examples.
        Decision Tree attribute for Root = A.
        For each possible value, vi, of A,
            Add a new tree branch below Root, corresponding to the test A = vi.
            Let Examples(vi) be the subset of examples that have the value vi for
A
            If Examples(vi) is empty
                Then below this new branch add a leaf node with label = most
common target value in the examples
            Else below this new branch add the subtree ID3 (Examples(vi),
Target_Attribute, Attributes – {A})
    End
    Return Root
```

Then we run ID3 algorithm on problem's data set:

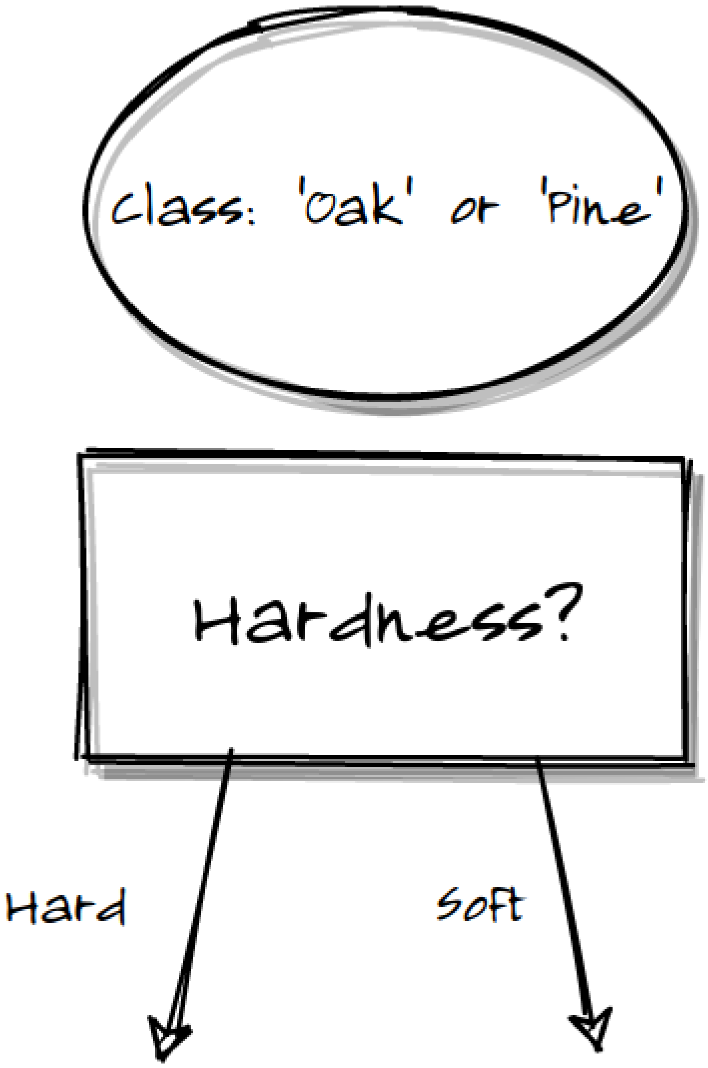
In iteration-1:

$$H(S) = -p(\text{class} = \text{'Oak'}) \times \log_2(p(\text{class} = \text{'Oak'})) \\ - p(\text{class} = \text{'Pine'}) \times \log_2(p(\text{class} = \text{'Pine'})) = 0.5 + 0.5 = 1$$

Features	IG(Features, S)
Density	1 - 0.5 - 0.5 = 0
Grain	1 - 0.5 -0.5 = 0
Hardness	1 - 0.31127812445913283 - 0.5 = 0.18872187554086717

$\longrightarrow$ Choose 'Hardness' as root.

Then split 'Hardness' out off data set.



In iteration-2:

$$H(\text{Harness}) = 0.8112781244591328$$

- Hard hardness on decision:

Density	Grain	Hardness	Class
Heavy	Small	Hard	Oak
Heavy	Large	Hard	Oak
Heavy	Small	Hard	Oak
Light	Large	Hard	Pine

Features	IG(Features, Hardness)
Density	0.8112781244591328 - 0.0 = 0.8112781244591328
Grain	0.8112781244591328 - 0.5 = 0.31127812445913283

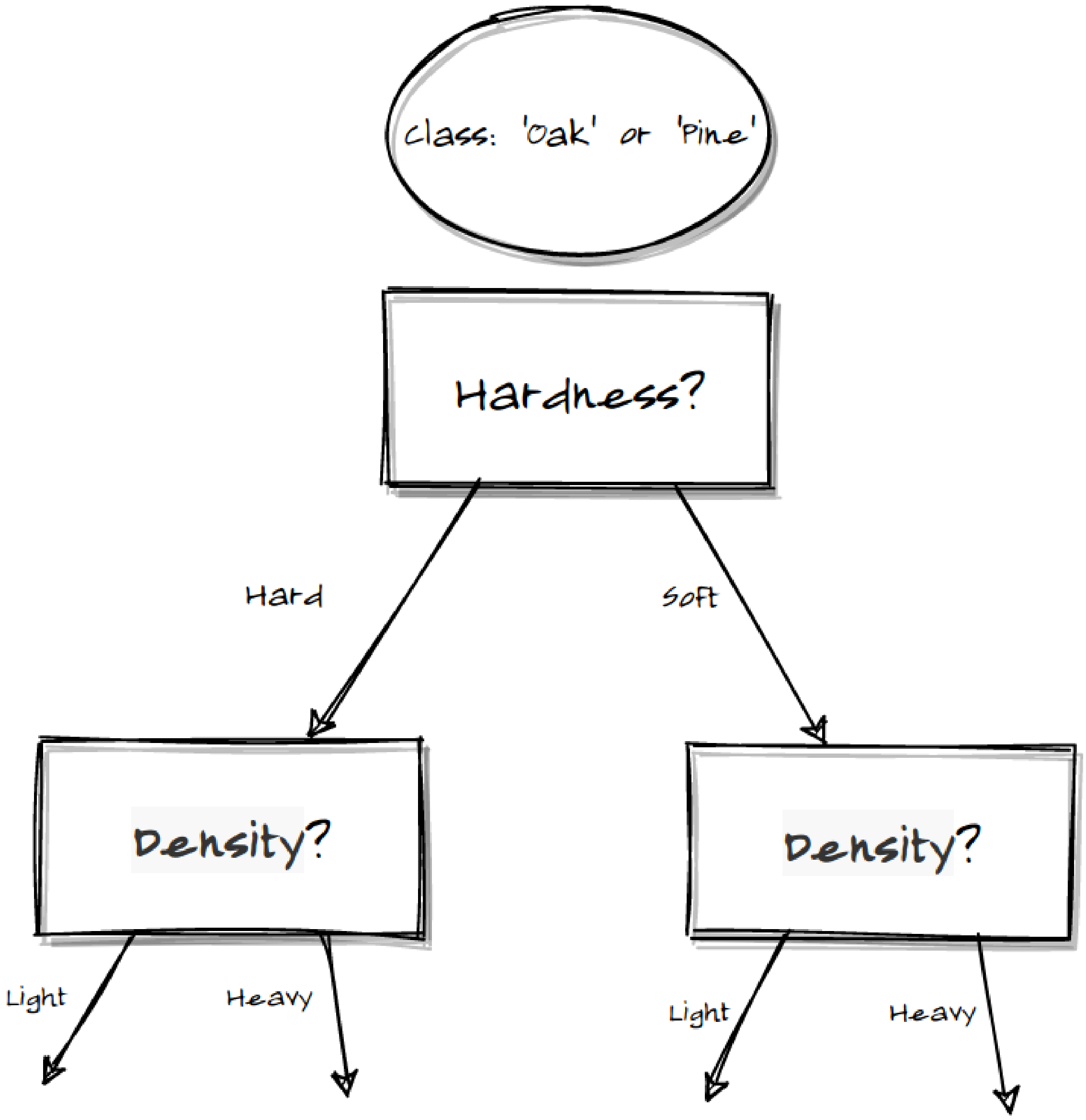
$\rightarrow$ choose Density as attribute.

- Soft hardness on decision:

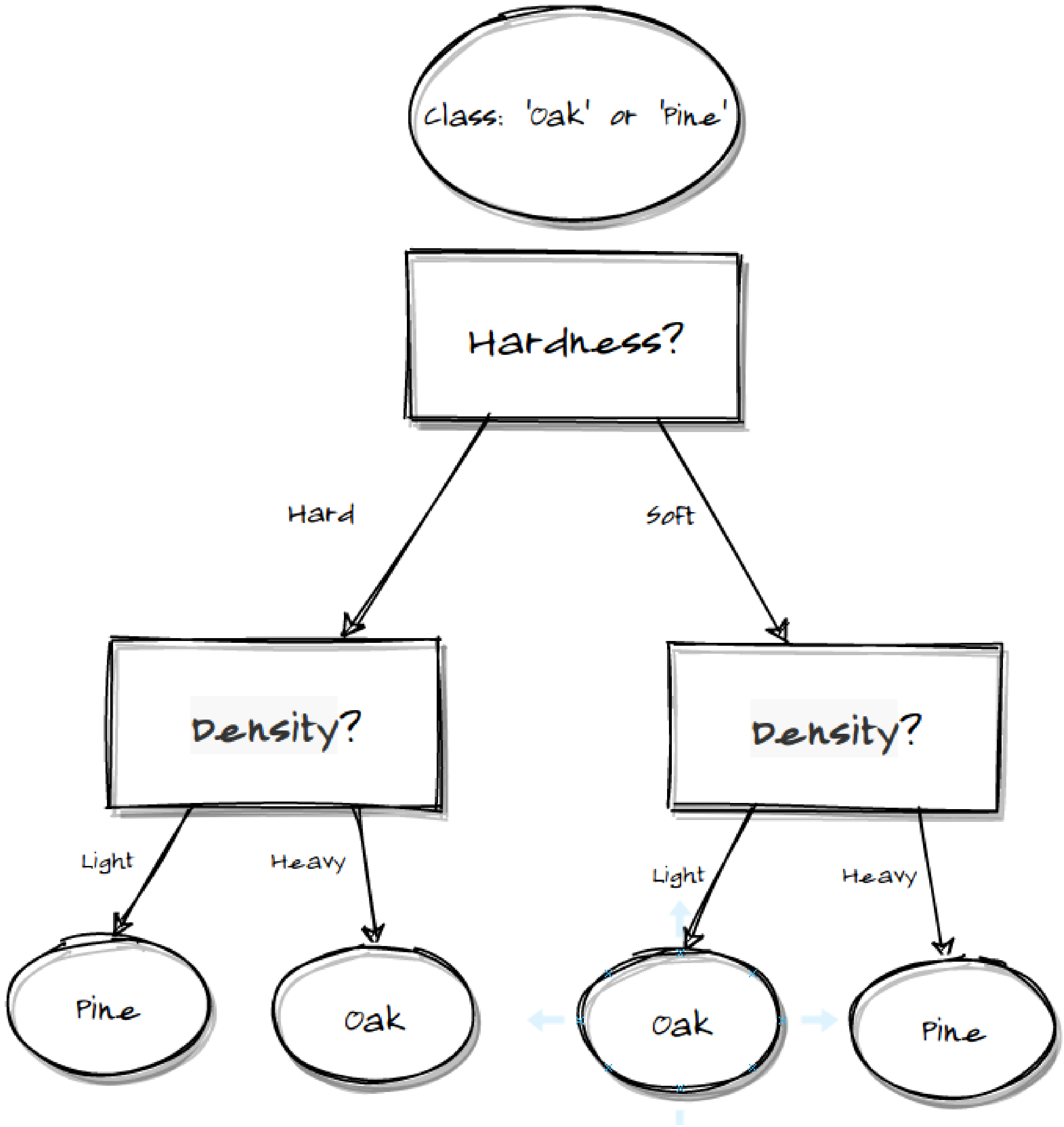
Density	Grain	Hardness	Class
Light	Large	Soft	Oak
Heavy	Small	Soft	Pine
Heavy	Large	Soft	Pine
Heavy	Small	Soft	Pine

Features	IG(Features, Hardness)
Density	0.8112781244591328 - 0.0 = 0.8112781244591328
Grain	0.8112781244591328 - 0.5 = 0.31127812445913283

$\rightarrow$ choose Density as attribute.



At the end of iteration, we can predict the Class of tuple because there is no multiple result on each branch.



Finish build `ID3 Decision Tree`.

b.

Examples	Result
[Density=Light, Grain=Small, Hardness=Hard]	**Pine**
[Density=Light, Grain=Small, Hardness=Soft]	**Oak**

Problem 2:

We have this table from `Table 2.1`

Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

With 4 features: [Gender, Car Ownership, Income Level, Transportation] we can calculate Entropy and Information Gain in the table below:

$$\begin{aligned} H(S) &= p(\text{Transportation} = \text{Train}) \times \log_2(p(\text{Transportation} = \text{Train})) \\ &+ p(\text{Transportation} = \text{Bus}) \times \log_2(p(\text{Transportation} = \text{Bus})) \\ &+ p(\text{Transportation} = \text{Car}) \times \log_2(p(\text{Transportation} = \text{Car})) \\ &= 0.5210896782498619 + 0.5287712379549449 + 0.5210896782498619 = 1.5709505944546684 \end{aligned}$$

Splitting on Gender:

- On Male Gender (`p=5/10 = 0.5`):

$$\begin{aligned} p(\text{Transportation} = \text{Train}|\text{Gender} = \text{Male}) = 1/5 = 0.2, \\ \implies \log_2(p(\text{Transportation} = \text{Train}|\text{Gender} = \text{Male})) = -2.321928094887362, \\ \implies p(\text{Transportation} = \text{Train}|\text{Gender} = \text{Male}) \times \log_2(p(\text{Transportation} = \text{Train}|\text{Gender} = \text{Male})) = -0.46438561897747244 \end{aligned}$$

$$\begin{aligned} p(\text{Transportation} = \text{Bus}|\text{Gender} = \text{Male}) = 3/5 = 0.6, \\ \iff \log_2(p(\text{Transportation} = \text{Bus}|\text{Gender} = \text{Male})) = -0.7369655941662062, \\ \iff p(\text{Transportation} = \text{Bus}|\text{Gender} = \text{Male}) \times \log_2(p(\text{Transportation} = \text{Bus}|\text{Gender} = \text{Male})) = -0.44217935649972373 \end{aligned}$$

$$\begin{aligned} p(\text{Transportation} = \text{Car}|\text{Gender} = \text{Male}) = 1/5 = 0.2, \\ \implies \log_2(p(\text{Transportation} = \text{Car}|\text{Gender} = \text{Male})) = -2.321928094887362, \\ \implies p(\text{Transportation} = \text{Car}|\text{Gender} = \text{Male}) \times \log_2(p(\text{Transportation} = \text{Car}|\text{Gender} = \text{Male})) = -0.46438561897747244 \end{aligned}$$

$$\implies H(S|Gender = Male) = 1.3709505944546687$$

- On Female Gender (`p=5/10 = 0.5`):

  In the same way with these step above:

$$\begin{aligned} p(\text{Transportation} = \text{Train}|\text{Gender} = \text{Female}) \times \log_2(p(\text{Transportation} = \text{Train}|\text{Gender} = \text{Female})) &= -0.5287712379549449 \\ p(\text{Transportation} = \text{Bus}|\text{Gender} = \text{Female}) \times \log_2(p(\text{Transportation} = \text{Bus}|\text{Gender} = \text{Female})) &= -0.46438561897747244 \\ p(\text{Transportation} = \text{Car}|\text{Gender} = \text{Female}) \times \log_2(p(\text{Transportation} = \text{Car}|\text{Gender} = \text{Female})) &= -0.5287712379549449 \end{aligned}$$

$$\implies H(S|Gender = Male) = 1.5219280948873621$$

$$\begin{aligned} \implies H(\text{S}|\text{Gender}) \quad &= 0.5 * 1.3709505944546687 + 0.5 * 1.5219280948873621 \quad &= 1.4464393446710155 \\ \implies IG(\text{S,Gender}) \quad &= 1.5709505944546684 - 1.4464393446710155 \quad &= 0.12451124978365291 \end{aligned}$$

Splitting on Car Ownership:

$$\begin{aligned}\implies H(\text{S}|\text{Car OwnerShip}) &= 1.036452797660028 \\ \implies IG(\text{S,Car OwnerShip}) &= 1.5709505944546684 - 1.036452797660028 = 0.5344977967946405\end{aligned}$$

Splitting on Travel Cost:

$$\begin{aligned}\implies H(\text{S}|\text{Travel Cost}) &= 0.36096404744368116 \\ \implies IG(\text{S,Travel Cost}) &= 1.5709505944546684 - 0.36096404744368116 = 1.2099865470109874\end{aligned}$$

Splitting on Income Level:

$$\begin{aligned}\implies H(\text{S}|\text{Income Level}) &= 0.8754887502163469 \\ \implies IG(\text{S,Income Level}) &= 1.5709505944546684 - 0.8754887502163469 = 0.6954618442383216\end{aligned}$$

The maximum `IG` attribute is `Travel Cost`, then is selected as splitting attribute aka root node of the `ID3` decision tree.

Problem 3:

a.

In first epoch:

$$a = hardlim(n) = hardlim(W_p + b)$$

The `perceptron` output follow the rule that $a_i \geq 0$ will be `T` and $a_i < 0$ will label at `F`.

The error function for `perceptron` is `L1 error`:

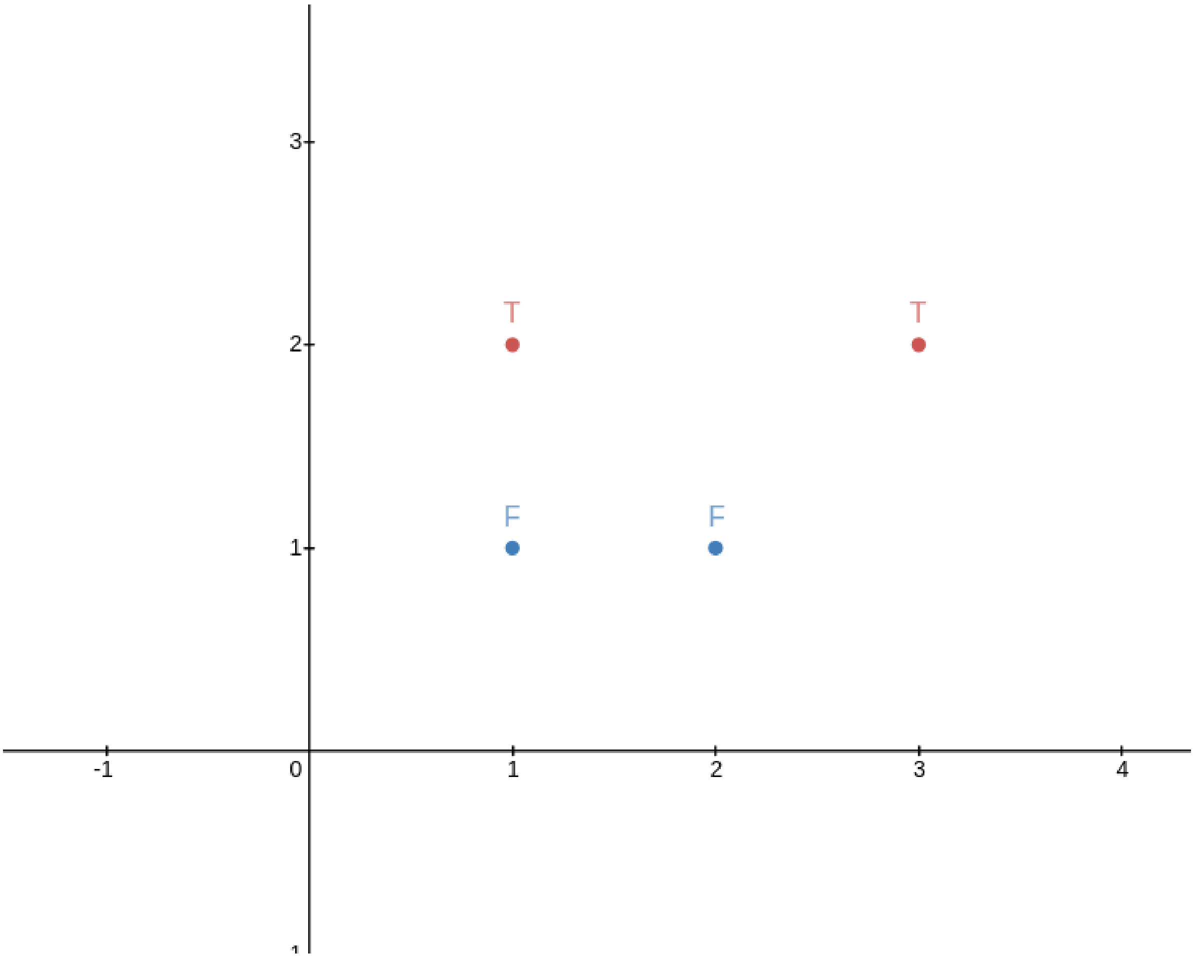
$$\text{L1}_{\text{error}} = \Sigma_{i=1}^{n} |y_{\text{true}} - y_{\text{predict}}|$$

A	B	Z	a	e
1	2	T	$2.1 + 1.2 - 4.5 = -0.5$	0.5
2	1	F	$2.2 + 1.1 - 4.5 = 0.5$	0.5
3	2	T	$3.2 + 2.1 - 4.5 = 2.5$	0
1	1	F	$1.2 + 1.1 - 4.5 = -1.5$	0

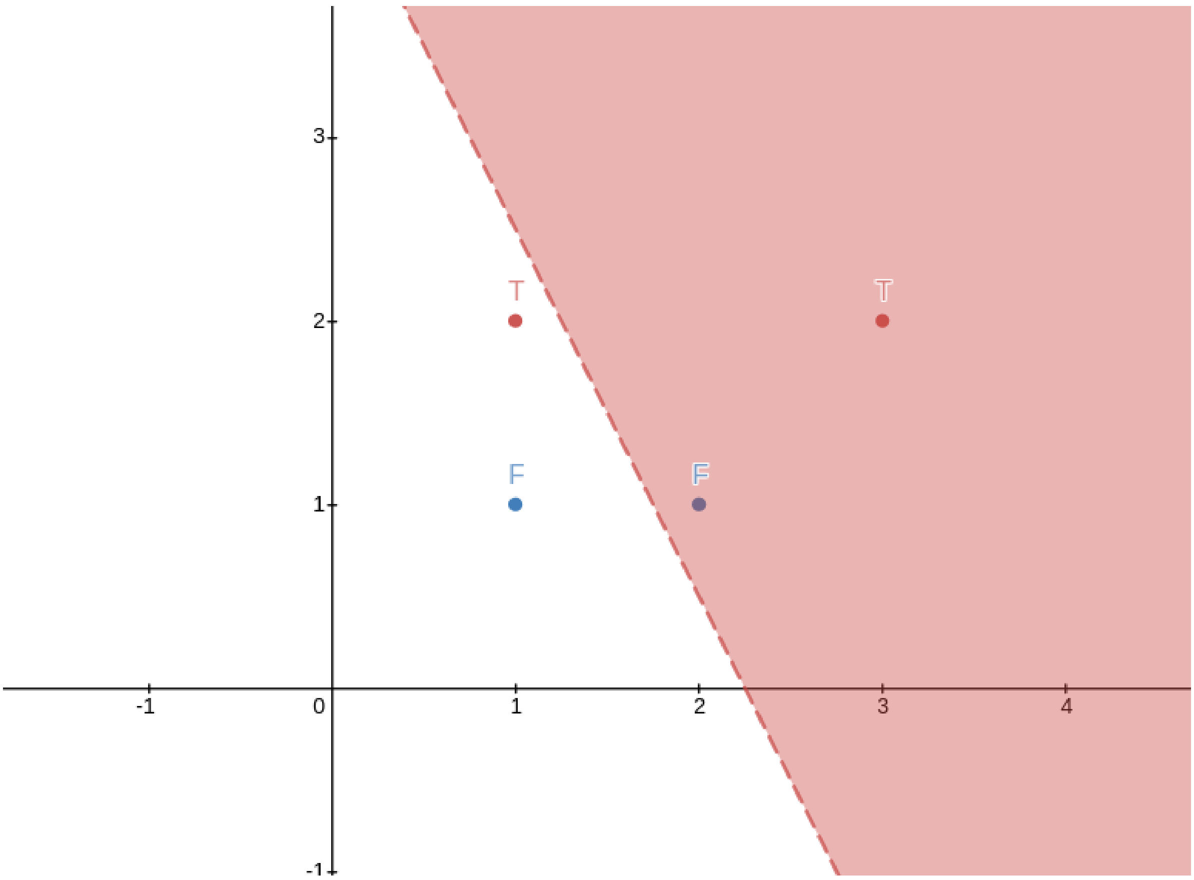
Total error in first epoch: $0.5 + 0.5 = 1$.

b.

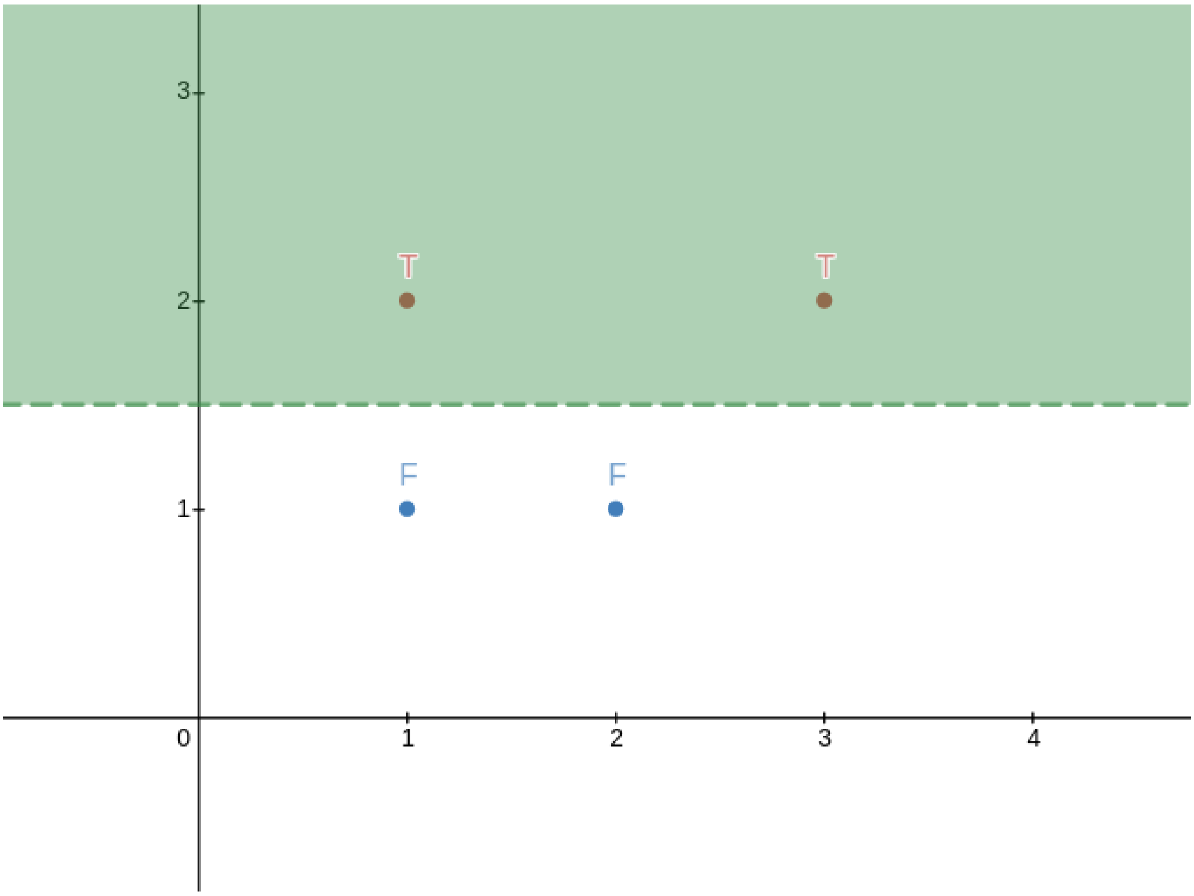
After plot the data to graph:



Our init weight:



We need a better line which split data set into 2 clusters. Let start with
$w_A = 0, w_b = 1, w_0 = -1.5$

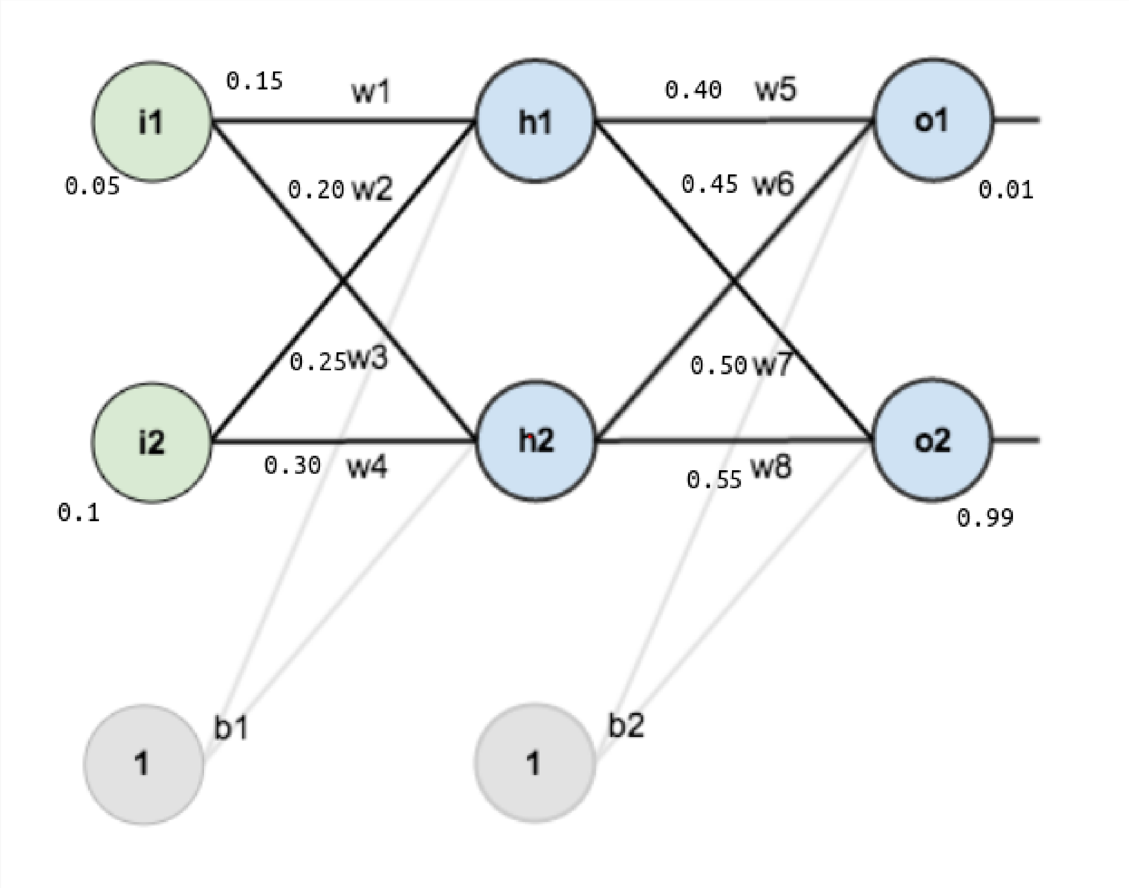


It's will easy to converge.

Problem 4:

a.

The forward pass:



Neuron net: ignore all bias

Variable	Value
$\mathrm{h1_{in}} = \mathrm{w1.\,i1} + \mathrm{w2.\,i2}$	$= 0.027500$
$\mathrm{h1_{out}} = \frac{1}{1+e^{-\mathrm{h1_{in}}}} = \frac{1}{1+e^{-0.0275}}$	$= 0.506875$
$\mathrm{h2_{in}} = \mathrm{w3.i1} + \mathrm{w4.i2}$	$= 0.042500$
$\mathrm{h2_{out}}$	$= 0.510623$
$\mathrm{o1_{in}} = \mathrm{h1.w5} + \mathrm{h2.w6}$	$= 0.432530$
$\mathrm{o1_{out}}$	$= 0.606478$
$\mathrm{o2_{in}} = \mathrm{h1.w7} + \mathrm{h2.w8}$	$= 0.534280$
$\mathrm{o2_{out}}$	$= 0.630481$

Calculate total error:

$$\mathrm{E_{total}} = (1/2)\Sigma(\text{target - output})^2$$

Variable	Value
E_{o1}	$= 0.177893$
E_{o2}	$= 0.064627$
E_{total}	$= 0.242163$

The backward pass:

- **Output layer:**

Using backward propagation of errors methods, to update w_i, we calculate $\frac{\partial E_{total}}{\partial wi}$

In output layer:

$$\frac{\partial E_{total}}{\partial wi} = \frac{\partial E_{total}}{\partial o_{out}} . \frac{\partial o_{out}}{\partial o_{in}} . \frac{\partial o_{in}}{\partial wi} = -(\text{target}_o - \text{out}_o) . o_{out} (1 - o_{out}) . h_{out}$$

Variable	Value
$\frac{\partial E_{total}}{\partial w5}$	0.072157
$\frac{\partial E_{total}}{\partial w6}$	0.072691
$\frac{\partial E_{total}}{\partial w7}$	-0.042455
$\frac{\partial E_{total}}{\partial w8}$	-0.042769

The update process:

$$wi_{new} = wi - \eta . \frac{\partial E_{total}}{\partial wi} = wi - 0.5 . \frac{\partial E_{total}}{\partial wi}$$

Variable	Value
$w5_{new}$	0.363922
$w6_{new}$	0.413655
$w7_{new}$	0.521228
$w8_{new}$	0.528616

- **Hiden layer:**

$$\frac{\partial E_{total}}{\partial wi} = (\Sigma_o \frac{\partial E_{total}}{\partial o_{out}} . \frac{\partial o_{out}}{\partial o_{in}} . \frac{\partial o_{in}}{\partial wi}) . \frac{\partial h_{out}}{\partial h_{in}} . \frac{\partial h_{in}}{\partial wi}$$
$$\text{With } \delta_o \text{ as } [-(\text{target}_o - \text{out}_o) . o_{out} (1 - o_{out}) . h_{out}]$$
$$\frac{\partial E_{total}}{\partial wi} = (\Sigma_o \delta_o . w_{ho}) . h_{out} . (1 - h_{out}) . \text{inp}$$

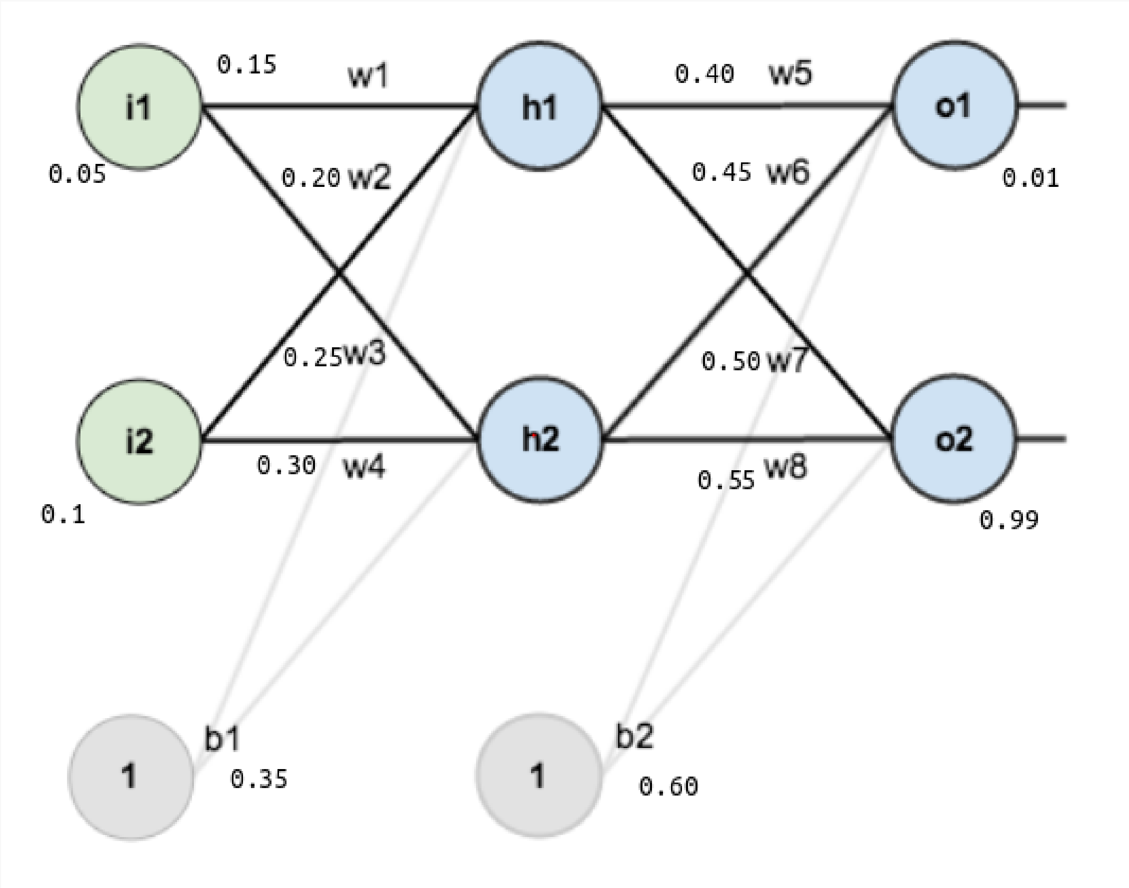
So:

Variable	Value
$\mathrm{w1_{new}} = \mathrm{w1} - 0,5.0,010024$	0.144988
$\mathrm{w2_{new}} = w2 - 0,5.0,020048$	0.189976
$\mathrm{w3_{new}} = \mathrm{w3} - 0,5.(-0,001065)$	0.250532
$\mathrm{w4_{new}} = \mathrm{w4} - 0,5.(0.002130)$	0.301065

Done :>.

b.

Using the same formula in a, but with the bias:



Neuron net: with bias

The forward pass:

Variable	Value
$\mathrm{h1_{in}} = \mathrm{w1.\,i1} + \mathrm{w2.\,i2} + \mathrm{b1.1}$	$= 0.375500$
$\mathrm{h1_{out}} = \frac{1}{1+e^{-\mathrm{h1_{in}}}} = \frac{1}{1+e^{-0.3755}}$	$= 0.593270$
$\mathrm{h2_{in}} = \mathrm{w3.i1} + \mathrm{w4.i2} + \mathrm{b1.1}$	$= 0.392500$
$\mathrm{h2_{out}}$	$= 0.596884$
$\mathrm{o1_{in}} = \mathrm{h1.w5} + \mathrm{h2.w6} + \mathrm{b2.1}$	$= 1.105906$
$\mathrm{o1_{out}}$	$= 0.751365$
$\mathrm{o2_{in}} = \mathrm{h1.w7} + \mathrm{h2.w8} + \mathrm{b2.1}$	$= 1.224921$
$\mathrm{o2_{out}}$	$= 0.772928$

Calculate total error:

Use same formula with a.

Variable	Value
$\mathrm{E_{o1}}$	$= 0.274811$
$\mathrm{E_{o2}}$	$= 0.023560$
$\mathrm{E_{total}}$	$= 0.298371$

The backward pass:

- **Output layer:**

Use same formula with a.

Variable	Value
$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w5}}$	0.082167
$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w6}}$	0.082668
$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w7}}$	-0.022603
$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w8}}$	-0.022740

Variable	Value
$\mathrm{w5_{new}}$	0.358916
$\mathrm{w6_{new}}$	0.408666
$\mathrm{w7_{new}}$	0.511301
$\mathrm{w8_{new}}$	0.561370

- **Hiden layer:**

Use same formula with a.

Variable	Value
$\mathrm{w1}_{\mathrm{new}} = \mathrm{w1} - 0,5.0,0004857$	0.149780
$\mathrm{w2}_{\mathrm{new}} = w2 - 0,5.0,000877$	0.189976
$\mathrm{w3}_{\mathrm{new}} = \mathrm{w3} - 0,5.(0.000498)$	0.249751
$\mathrm{w4}_{\mathrm{new}} = \mathrm{w4} - 0,5.(0,000994)$	0.299502

Done :>

Have a Great Day