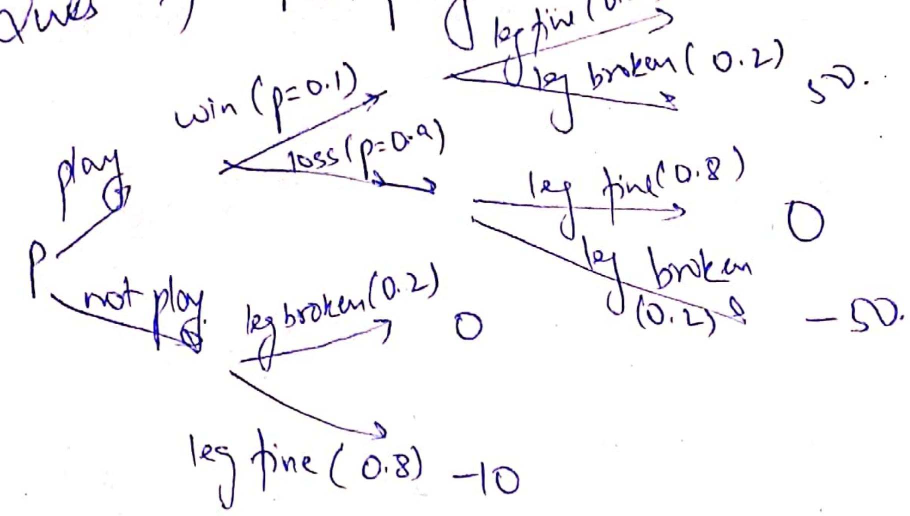
Ques 1) P → player



(2) [Expected utility of playing] = 0.1 (100 × 0.8 + 50 × 0.2)
+ 0.9 (−50 × 0.2 + 0 × 0.8)

= 0.1 (90) + 0.9 (−10)

= 9 − 9 = 0.

∴ Expected utility of playing = 0

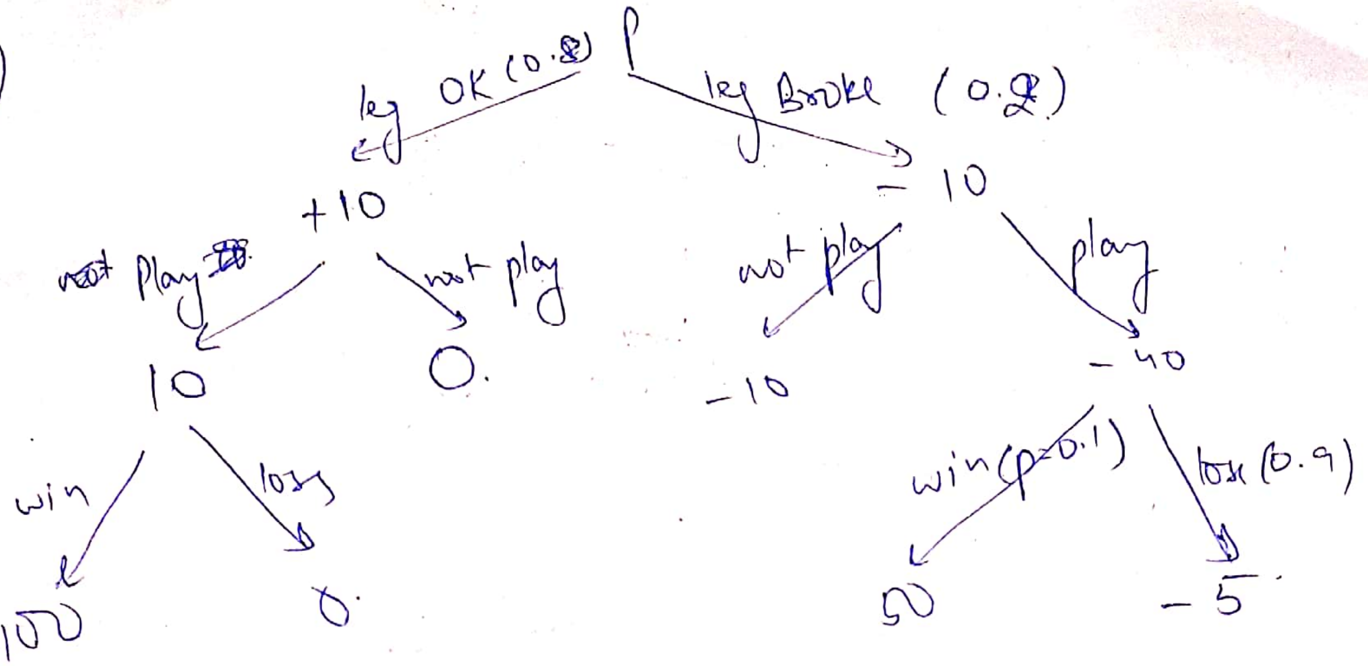
Expected utility of not playing = 0.2 (−10) + 0.(0.8)

= −2.

∴ Expected utility of not playing = −2.

Since 0 > −2, ∴ P should play.

A3)

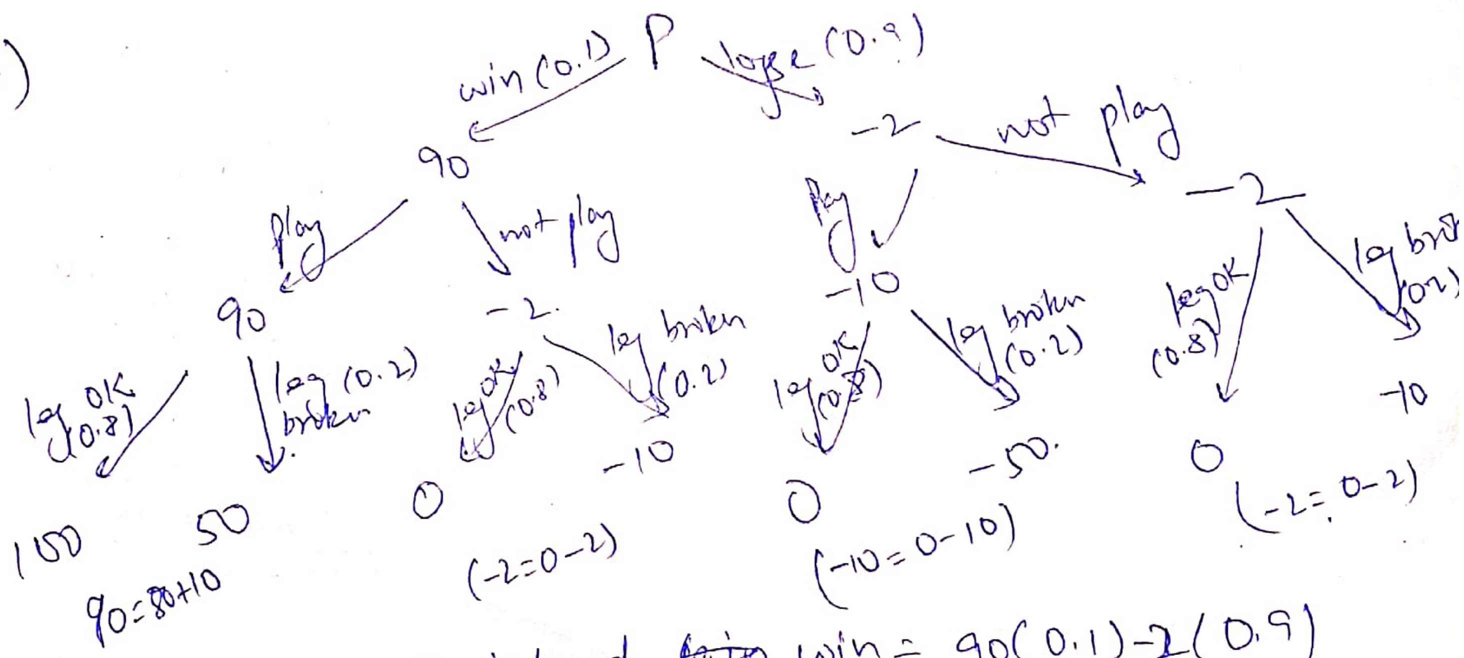


∴ ~~Expected info of~~

Expected value of perfect info of the leg =

$0.2 \times 10 + 0.8 \times 10 = -2 + 8 = 6.$

4)



∴ Expected val of info of ~~fair~~ win = $90(0.1) - 2(0.9)$
$= 9 - 1.8$
$= 7.2.$

(5) To find P[win | broken]
If winning decision branch is inserted after LEG OK & LEG BROKEN, the reqd prob. can be calculated & that can help to see whether win is more likely or loss

Question 2.)

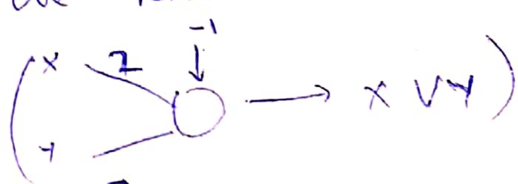
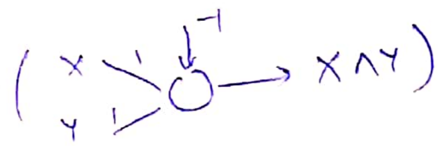
T.P. $f: \{0,1\}^d \rightarrow \{0,1\}$

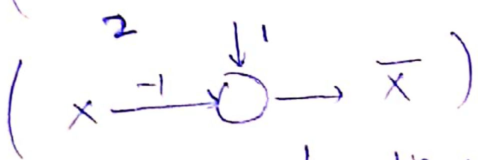
can be represented as a neural network with just 1 hidden layer.

Now this question can be seen like having a black box which takes any number of boolean values as inputs & outputs either 0 or a 1.

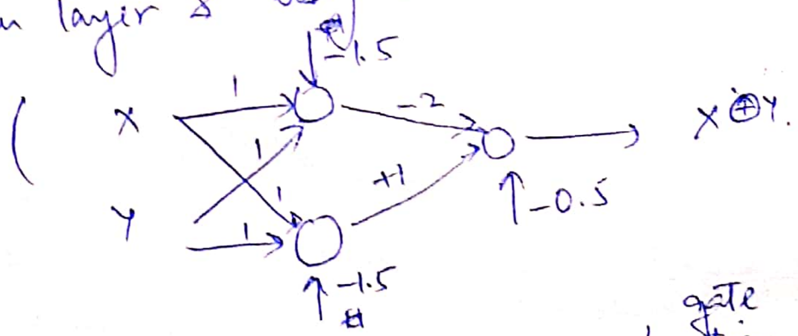
So this question can be modelled as proving that ~~we~~ can we present a ~~single hidden~~ boolean function ~~as a single layer layer perceptron~~ with the help of a single ~~boolean~~ hidden layer.

Now we know than simple gates like

OR  AND 

NOT  can be represented directly as single perceptrons.

When we need functions like XOR, we introduce a hidden layer & use that XOR can be easily implemented.

XOR 

Now we know that any ~~function~~ gate can be created using AND and NOT gates (from ECE 101 (Digital ckts)) (omitting the proof w.r.t.)

∴ we can reduce any expression to a series of and and ~~or gates~~ not gates. Thus we can use these gates to model any boolean logic (part of hidden layers) & their output would be a boolean value either 0 or 1

~~Or~~ This can be illustrated using XOR gate itself on the previous page.

Thus since any boolean function can be represented using AND and OR gates & these & can be implemented using just a single perceptron,

∴ Any boolean function can be implemented using just ~~the~~ a single hidden layer.

Ans 3) 2 input nodes, 1 hidden layer

$Y = \operatorname{argmax}_y (P(Y = y \mid X))$,

$$Y = P(Y = 1 \mid X) = \frac{\exp(\beta_1 X_1 + \beta_2 X_2)}{1 + \exp(\beta_1 X_1 + \beta_2 X_2)}$$

$$Y = P(Y = -1 \mid X) = \frac{1}{1 + \exp(\beta_1 X_1 + \beta_2 X_2)}$$