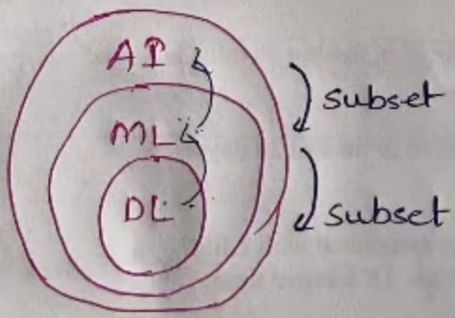
What is machine learning?

field of study that gives computers a capability to learn without being explicitly programmed

example: online shopping amazon

→ machine adapts to the user based on data



* Well posed learning problem:

An agent solves a problem or task T, performance P and gain some experience E

If P is measured at T it can improve E.

(learning by experience)

→ Examples:

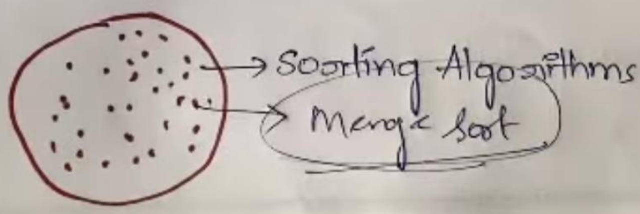
1. Playing checkers Problem
2. Hand written recognition Problem
3. Robo ~~lea~~ driving learning Problem

Problem	Task (T)	Performance (P)	experience (E)
1. playing checkers learning	playing against opponents to win game	make perfect moves to win game	It plays itself to improve
2. Hand writing recognition learning	classifying the images & text	Better classification	A database of home work text
3. Robot driving learning Problem	Drive the car in a 4 lane highway	source to dest. the avg distance travelled (long & safe)	Images, Vehicles on Road

* PERSPECTIVES OF MACHINE LEARNING:

Perspective of machine learning involves searching very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by learner

ex:



* ISSUES IN MACHINE LEARNING:

1. What algorithms should be used?
2. which algorithms perform best for which types of problems?
3. How much training data is sufficient? and testing data
4. What kind of methods should be used?
5. What methods should be used to reduce learning overhead
6. for which type of data which methods should be used?

* PERSPECTIVES OF MACHINE LEARNING:

* DESIGNING A LEARNING SYSTEM:

To get a successful learning system, it should be designed
- for a proper design, several steps should be followed

↓

Why? - for perfect & efficient system

* 4 steps

1. choosing the training experience
2. choosing the target function
3. choosing a representation for target function
4. choosing a learning algorithm for approximating the target function

To get a successful learning system, it should be designed

- for a proper design, several steps should be followed

Why? - for perfect & efficient system

* 4 Steps

1. choosing the training experience
2. choosing the target function
3. choosing a representation for target function
4. choosing a learning algorithm for approximating the target function

- final design is obtained.

STEP 2 : CHOOSING THE TARGET FUNCTION :

What type of knowledge is learnt and how it is used by the Performance system.

Example. checkers game

→ While moving diagonally, set of all possible moves is called legal moves
↓
one move
↓

- Travel only in forward dir
- only one move per chance
- only in diagonal direction
- Jump over opponent

Target function ⇒ V(b)

Example. checkers game

→ While moving diagonally, set of all possible moves is called legal moves
↓
one move
↓
target move

f(x)

g(x) h(x)

- Travel only in forward dir
- only one move per chance
- only in diagonal direction
- Jump over opponent

Target function ⇒ V(b)

Board state ⇒ b

Legal moves set ⇒ B

1. b is final board state that is won, then $V(b) = 100$
2. b is final board state that is lost, then $V(b) = -100$
3. b is final board state that is draw, then $V(b) = 0$.
4. If b is not final state then $V(b) = V(b')$

$b' \rightarrow$ best final state

STEP 1: CHOOSING A TRAINING EXPERIENCE:

In choosing a training experience, 3 attributes are taken

1. Type of feedback
2. degree
3. distribution of examples.

1) Whether the training experience provides direct or indirect feedback regarding the choices made by Performance system

Direct feedback

Indirect feedback

3. distribution of examples.

1) Whether the training experience provides direct or indirect feedback regarding the choices made by Performance system

Task → P → good → good E

Direct feedback - move

Indirect feedback

training

Di Indi

Examples: Checkers game, learning driving.

→ You want to go with direct / Indirect

Its your choice

---

2) Degree to which learner will control the sequence of training.

Example: learning driving.

with trainers help

with trainers Partial help

completely on your own

3) How well it represents the distribution of examples over which the performance of final system is measured

- more possible combinations
- more situations, more examples

Example: learning driving

✓ with trainers help — ✓ with trainers Partial help — completely on your own

3) How well it represents the distribution of examples over which the performance of final system is measured

- more possible combinations
- more situations, more examples
- learning over distributed range of examples.

Example: learning driving

STEP:4: CHOOSING A LEARNING ALGORITHM FOR APPROXIMATING THE TARGET FUNCTION

To learn a target function (f) we need a set of training examples

(describe a particular board state and training value)

(b) — Vtrain(b)

Ordered pair = (b, Vtrain(b))

(training example representation)

Example: black won the game

(i.e $X2=0$, which means no red

$$V_{train}(b) = +100$$

$$b = (X1=3, X2=0, X3=1, X4=0, X5=0, X6=0)$$

$$\langle (x_1=3, x_2=0, x_3=1, x_4=0, x_5=0, x_6=0) +100 \rangle$$

We need to do 2 steps in this phase

1. Estimating training values:

In every step, we consider successor

(depending on the next step of opponent)

$$V_{train}(b) \leftarrow \hat{V}(\text{successor}(b))$$

represents the next board state

(estimating that this move will help/destroy opponent)

$\hat{V} \rightarrow$ represents approximation

2. Adjusting the weights:

$V^\wedge \rightarrow$ represents approximation

2. Adjusting the weights:

There are some alg. to find weights of linear functions

Here, we are using LMS (least mean square)

(used to minimise the error)

Error = $(V(train(b) - V^\wedge(b))^\wedge 2$

If error=0, no need to change weights

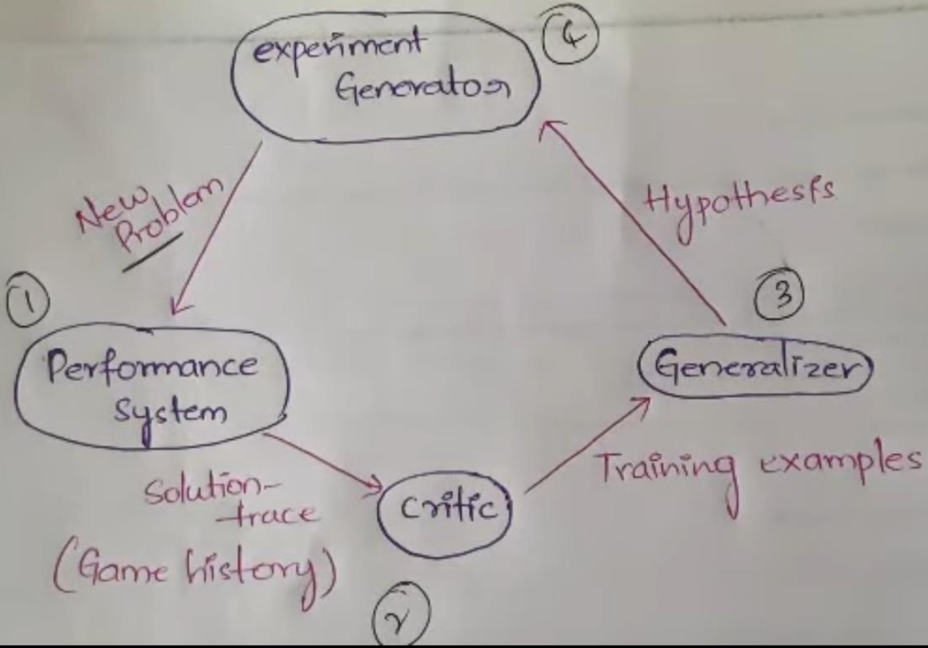
If error is positive, each weight is increased in propo

If error is negative, each weight is decreased in pro

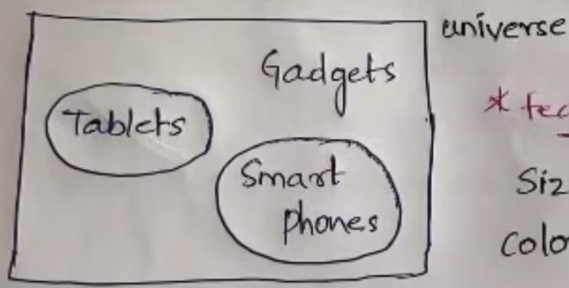
*FINAL DESIGN:

- has 4 different modules

1. Performance System
2. Critic
3. Generalizer
4. Experiment Generator



* CONCEPT LEARNING:



* features (Binary valued attributes)

Size:

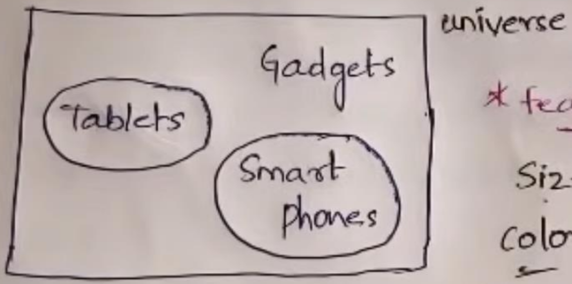
color:

Screentype:

shape:

concept = $\langle x_1, x_2, x_3, x_4 \rangle$

* CONCEPT LEARNING:



* features (Binary valued attributes)

Size: large, Small

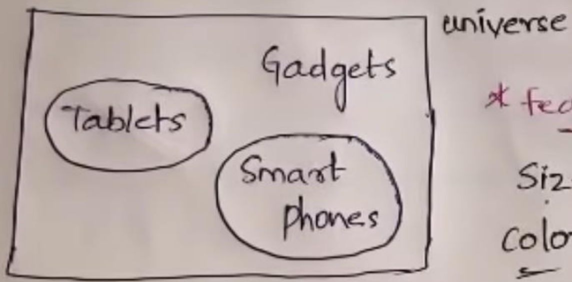
color: Black, Blue

Screentype: Flat, Folded

shape: Square, Rectangle

.

* CONCEPT LEARNING:



* features (Binary valued attributes)

Size: large, Small → x_1

color: Black, Blue → x_2

Screentype: Flat, Folded → x_3

shape: Square, Rectangle → x_4

Screentype: Flat, Folded $\rightarrow x_3$
Shape: Square, Rectangle $\rightarrow x_4$

concept = $\langle x_1, x_2, x_3, x_4 \rangle \rightarrow$ 4.

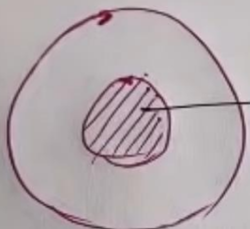
Tablet = < large, black, Flat, Square >

Smartphone = < Small, Blue, Folded, Rectangle >

No. of possible instances = 2^d $2^4 = 16$

d = no. of features.

$\boxed{\text{Total possible concepts} = 2^{2^d}}$ 2^{2^4} 2^{16}



$\rightarrow$ target concept / hypothesis space

$\langle \phi, \phi, \phi, \phi \rangle \rightarrow$ Reject all
(most specific hypothesis)

$\langle ?, ?, ?, ? \rangle \rightarrow$ accept all
(most general hypothesis)

(most specific hypothesis)

$\langle ?, ?, ?, ? \rangle \rightarrow$ accept all

(most general hypothesis)

Main Goal →

finds all concepts/hypothesis that are consistent

Example:

2000's

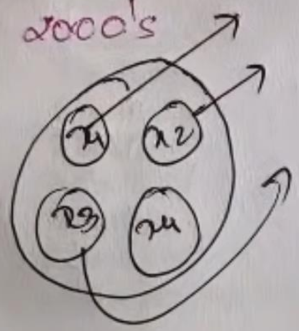
2020's

Main Goal →

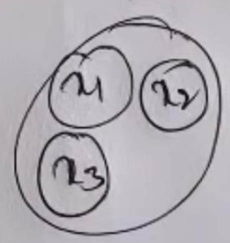
finds all concepts/hypothesis that are consistent

Example:

2000's



2020's



* CONCEPT LEARNING AS SEARCH:

Main goal of this search is to find the hypothesis that best fits the training examples.

Examples: Enjoysport learning task

→ 6 attributes

(Sky, Air temperature, humidity, wind, water and forecast)

Sky → 3 values – Rainy, cloudy and sunny

Remaining attributes – only 2 values

Different instances possible –

* CONCEPT LEARNING AS SEARCH:

Main goal of this search is to find the hypothesis that best fits the training examples.

Examples: Enjoysport learning task

→ 6 attributes

(1 Sky, 2 Air temperature, 3 humidity, 4 wind, 5 water and 6 forecast)

Sky → 3 values – Rainy, cloudy and sunny

Remaining attributes – only 2 values

→ 6 attributes

(Sky, Air temperature, humidity, wind, water and forecast)

Sky → 3 values – Rainy, cloudy and sunny

Remaining attributes – only 2 values

∴ Different instances possible =

Syntactically distinct hypothesis =

(Additionally, 2 more values – ? and ϕ)

Semantically distinct hypothesis =

(Null – taken as common)

joysport learning task

→ 6 attributes

(Sky, Air temperature, humidity, wind, water and forecast)

Sky → 3 values – Rainy, cloudy and sunny

Remaining attributes – only 2 values

∴ Different instances possible = $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$

Syntactically distinct hypothesis =

(Additionally, 2 more values – ? and ϕ)

<ϕ, Rainy, cloudy, sunny, ?

Semantically distinct hypothesis =

(Null – taken as common)

Enjoysport learning task

→ 6 attributes

(Sky, Air temperature, humidity, wind, water and forecast)

↓

3 values – Rainy, cloudy and sunny

Remaining attributes – only 2 values

∴ Different instances possible = $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$

Syntactically distinct hypothesis = $5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$

(Additionally, 2 more values – ? and ϕ)

<ϕ, Rainy, cloudy, sunny, ?>

Semantically distinct hypothesis =

(Null – taken as common)

forecast)

↓

3 values – Rainy, cloudy and sunny

Remaining attributes – only 2 values

∴ Different instances possible = $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$

Syntactically distinct hypothesis = $5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$

(Additionally, 2 more values – ? and ϕ)

<ϕ, Rainy, cloudy, sunny, ?>

Semantically distinct hypothesis = $1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3)$

$= 973$

(Null – taken as common

(ϕ) only 1 more value)

– After finding all the syntactically & semantically

Different instances possible $= 3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$

Syntactically distinct hypothesis $= 5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$

(Additionally, 2 more values - ? and ϕ)

$\langle \phi$, Rainy, cloudy, sunny, ? $\rangle$

Semantically distinct hypothesis $= 1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3)$

(Null - taken as common)

(ϕ) only 1 more value)

$= 973$

- After finding all the syntactically & semantically distinct hypothesis,

we search the best match from all these

(i.e much closer to our learning problem)

* FIND S ALGORITHM:

(finding a maximally specific hypothesis)

- this algorithm considers only positive examples

*** Representations:**

most specific hypothesis $\Rightarrow \phi$

most general hypothesis $\Rightarrow$?

$\Rightarrow$ Algorithm:

Step1: Initialise with most specific hypothesis (ϕ)

$h_0 = \langle \phi, \phi, \phi, \phi, \phi \rangle \Rightarrow$ 5 attributes

Step2: for each +ve sample,

for each attribute,

if(value = hypothesis value) $\Rightarrow$ Ignore

* FIND (S) ALGORITHM: ϕ

(finding a maximally specific hypothesis)

- this algorithm considers only positive examples   Yes ⇒ +ve
                                                     No ⇒ −ve

*** Representations:**

most specific hypothesis ⇒ ϕ ✓

most general hypothesis ⇒ ? ✓

⇒ **Algorithm:**

Step1: Initialise with most specific hypothesis (ϕ)

$h_0 = <\phi, \phi, \phi, \phi, \phi>$ ⇒ 5 attributes

Step2: for each +ve sample,

for each attribute,

if(value = hypothesis value) ⇒ Ignore

* Representations: No ⇒ −ve

most specific hypothesis ⇒ ϕ ✓

most general hypothesis ⇒ ? ✓

⇒ Algorithm:

Step1: Initialise with most specific hypothesis (ϕ)

$h_0 = <\phi, \phi, \phi, \phi, \phi>$ → 5 attributes

5 ϕ

Step2: for each +ve sample,

for each attribute,

if(value = hypothesis value) ⇒ Ignore

else

Replace with the most general hypothesis (?)

Origin	manufacturer	color	Year	Type	class

if(value = hypothesis value) ⇒ Ignore
else
Replace with the most general hypothesis (?)

origin	manufacturer	color	Year	Type	class
JP	HO	Blue	1980	eco	+ (Yes)
JP	TO	Green	1970	spo	− (No)
JP	TO	Blue	1990	eco	+
USA	AU	Red	1980	eco	−
JP	HO	White	1980	eco	+
JP	TO	Green	1980	eco	+
JP	HO	Red	1980	eco	−

origin
1. JP −
2. JP
3. JP −
4. USA
5. JP
6. JP
7. JP

* Disadvan...
- consider...
- h6 may...

$h_1 = \langle 'JP', 'HO, Blue, 1980, eco \rangle$

$h_2 = h_1$

$h_3 = \langle JP, ?, Blue, ?, eco \rangle$

$h_4 = h_3$

$h_5 = \langle JP, ?, ?, ?, eco \rangle$

$\boxed{h_6 = \langle JP, ?, ?, ?, eco \rangle}$ → most general

Origin	manufacturer	color	Year	Type	class
1. JP	HO	Blue	1980	eco	+ (Yes)
2. JP	TO	Green	1970	spo	- (No)
3. JP	TO	Blue	1990	eco	+
4. USA	AU	Red	1980	eco	-
5. JP	HO	White	1980	eco	+
6. JP	TO	Green	1980	eco	+
7. JP	HO	Red	1980	eco	-

* Disadvantage:
- considers only +ve values
- h6 may not be sole hypothesis that fits the complete data

* VERSION SPACE:

Subset of hypothesis H consistent with the training examples

$$VS_{H,D} = \{h \in H \mid consistent(h, D)\}$$

H = hypothesis
D = training examples

Consistent

$h(x) = c(x)$

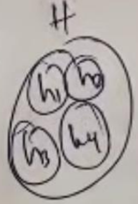
* VERSION SPACE:

Subset of hypothesis (H) consistent with the training examples (D).

$VS_{H,D} = \{ h \in H \mid consistent(h, D) \}$

H = hypothesis

D = training examples



consistent

$h(x) = c(x)$

* Algorithm to obtain Version space:

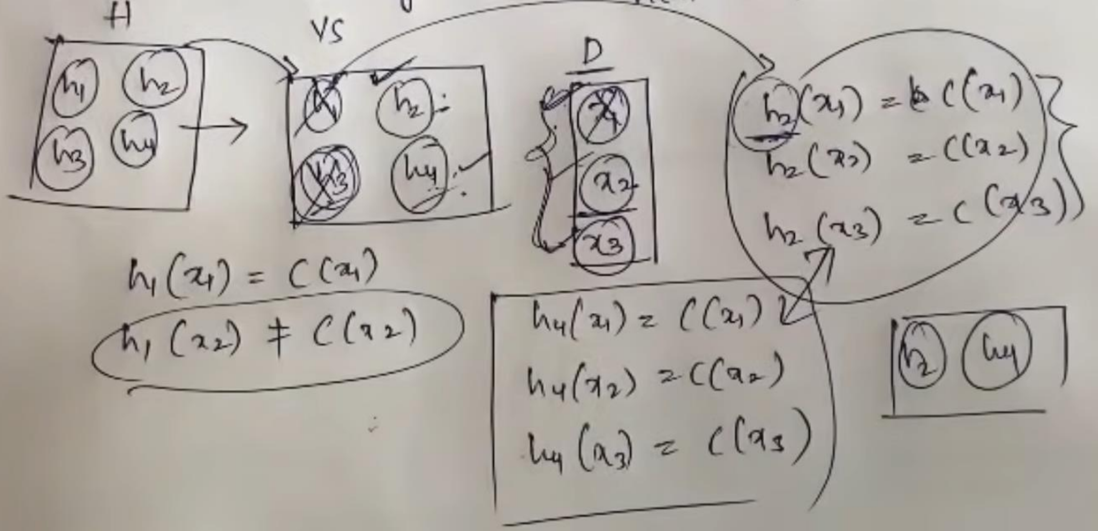
(List then eliminate algorithm) → 3

1. Version Space ← list containing every hypothesis in H
2. from this step, we keep on removing inconsistent hypothesis from version space

for each training example, $\langle x, c(x) \rangle$ remove any hypothesis that is $h(x) \neq c(x)$ $h(x) = c(x)$

3. o/p the list of hypothesis into version space after checking for all training examples.

for all training examples. $h(x) = c(x)$



* CANDIDATE ELIMINATION ALGORITHM:

→ uses the concept of version space

→ considers both +ve and −ve values (Yes and No)

→ both specific and general hypothesis

- for positive samples, move from specific to general
- for negative samples, move from general to specific

$S = \{\phi\phi\phi\phi\phi\}$ + ↓

$G = \{?????\}$ − ↑

Algorithm:

1. Initialise both general and specific hypothesis (S and G)

$S = \{\phi, \phi, \phi, \phi, \ldots \phi\}$ depends on no. of attributes.

$G = \{?, ?, ? \ldots ?\}$

2. for each example,

if example is positive

make specific to general

else example is -ve

make general to specific

Dataset (EnjoySport)

Sky	Temperature	Humidity	Wind	Water	Forecast	Enjoy
Sunny	Warm	Normal	Strong	Warm	Same	Yes (+)
Sunny	Warm	High	Strong	Warm	Same	Yes (+)
Rainy	Cold	High	Strong	Warm	Change	No (-)
Sunny	Warm	High	Strong	Cool	Change	Yes (+

make general to specific

Example: EnjoySport

$S_0 = \{\phi\phi\phi\phi\phi\phi\}$ $G_0 = \{??????\}$

1) +ve,

$S_1 = \{$'sunny', 'warm', 'normal', 'strong', 'warm', 'same'

$G_1 = \{??????\}$

+ve,

(specific to general)

= {'sunny', 'warm' ? 'strong', 'warm', 'same'}

1) +ve,

$S_1 = \{$'sunny', 'warm', 'normal', 'strong', 'warm', 'same'$\}$

$G_1 = \{??????\}$

2) +ve,

(specific to general)

$S_2 = \{$'sunny', 'warm', ?, 'strong', 'warm', 'same'$\}$

$G_2 = \{??????\}$

3) $S_3 = \{$'sunny', 'warm' ? 'strong', 'warm', 'same'$\}$

$G_3 = \{$<'sunny' ?????>, <?, 'warm' ????>, <????? 'same'>$\}$

4) $S_4 = \{$'sunny', 'warm', ?, 'strong', ? ?>

... <?, 'warm' ? ????>}

(specific to general)

S_2 = {'sunny', 'warm', ?, 'strong', 'warm', 'same'}

G_2 = {? ? ? ? ? ? ?}

3) S_3 = {'sunny', 'warm' ? 'strong', 'warm', 'same'}

G_3 = {<'sunny' ? ? ? ? ? >, <?, 'warm' ? ? ? ? >, <? ? ? ? ? 'same'>}

4) S_4 = {'sunny', 'warm', ?, 'strong', ? ?>

G_4 = {<'sunny' ? ? ? ? ? >, <? 'warm' ? ? ? ? >}

S_4 and G_4 ⇒ final hypothesis

*CANDIDATE ELIMINATION ALGORITHM:

→ uses the concept of version space

→ considers both +ve and -ve values (Yes and No)

→ both specific and general hypothesis

- for positive samples, move from specific to general
- for negative samples, move from general to specific

S = {ϕ ϕ ϕ ϕ ϕ} + ↓

G = {? ? ? ? ? ?} − ↑

* Algorithm:

1. Initialise both general and specific hypothesis (S and G)

$$S = \{\phi, \phi, \phi, \phi, \ldots \phi\}$$

$$G = \{?, ?, ? \ldots ?\}$$

depends on no. of attributes.

2. for each example,

    if example is positive

        make specific to general

    else example is −ve

        make general to specific

make general to specific

Example: EnjoySport

$S_0 = \{\phi\phi\phi\phi\phi\phi\}$ $G_0 = \{??????\}$

1) +ve,

$S_1 = \{$'sunny', 'warm', 'normal', 'strong', 'warm', 'same'

$G_1 = \{??????\}$

+ve,

(specific to general)

= {'sunny', 'warm' ? 'strong', 'warm', 'same'}

Dataset (EnjoySport)

Sky	Temperature	Humidity	Wind	Water	Forecast	Enjoy
unny	Warm	Normal	Strong	Warm	Same	Yes (+)
unny	Warm	High	Strong	Warm	Same	Yes (+)
iny	cold	High	Strong	Warm	change	No (-)
nny	Warm	High	Strong	cool	change	Yes

Dataset (EnjoySport)

Sky	Temperature	Humidity	Wind	Water	Forecast	Enjoy
unny	Warm	Normal	Strong	Warm	Same	Yes (+)
unny	Warm	High	Strong	Warm	Same	Yes (+)
iny	cold	High	Strong	Warm	change	No (−)
nny	Warm	High	Strong	cool	change	Yes

make specific to general

else example is −ve

make general to specific

Example: EnjoySport

$S_0 = \{\phi\phi\phi\phi\phi\phi\}$ $G_0 = \{??????\}$

1) +ve,

$S_1 = \{$'sunny', 'warm', 'normal', 'strong', 'warm', 'same'$\}$

$G_1 = \{??????\}$

2) +ve,

(specific to general)

$S_2 = \{$'sunny', 'warm', ?, 'strong', 'warm', 'same'$\}$

$G_2 = \{??????\}$

1) +ve,

S_1={'sunny','warm','normal','strong','warm','same'}

G_1={? ? ? ? ? ?}

2) +ve,

(specific to general)

S_2={'sunny','warm',?,'strong','warm','same'}

G_2= {? ? ? ? ? ?}

3) S_3 = {'sunny', 'warm' ? 'strong','warm','same'}

G_3= { <'sunny' ? ? ? ? ?>, < ?, 'warm' ? ? ? ?>, <? ? ? ? ? 'same'>}

4) S_4= {'sunny','warm', ?, 'strong', ? ?>

G_1={? ? ? ? ? ?}

2) +ve,

(specific to general)

S_2={'sunny','warm',?,'strong','warm','same'}

G_2= {? ? ? ? ? ?}

3) S_3 = {'sunny', 'warm' ? 'strong','warm','same'}

G_3= { <'sunny' ? ? ? ? ?>, < ?, 'warm' ? ? ? ?>, <? ? ? ? ? 'same'>}

4) S_4= {'sunny','warm', ?, 'strong', ? ?>

G_4={<'sunny' ? ? ? ? ?>, <? 'warm' ? ? ? ?>}

S_4 and G_4 ⇒ final hypothesis

* INDUCTIVE BIAS:

→ Remarks on CE and VS Algorithms:

1) Will the CE algorithm gives us correct hypothesis?
2) What training example should the learner request next?

→ Inductive Learning:

From examples we derive rules

→ Deductive Learning:

Already existing rules are applied to our examples the CE algorithm gives us correct hypothesis?

2) What training example should the learner request next?

→ Inductive Learning:

From examples we derive rules

→ Deductive Learning:

Already existing rules are applied to our examples.

* Biased hypothesis space:

does not consider all types of training examples.

Solution → include all hypothesis.

Already existing rules are applied to our examples.

* Biased hypothesis space:

does not consider all types of training examples.

Solution → include all hypothesis.

ex: Sunny $\wedge$ warm $\wedge$ normal $\wedge$ strong $\wedge$ cool $\wedge$ change $\Rightarrow$ Yes.

* Unbiased hypothesis Space:

Providing a hypothesis capable of representing set of all examples

Possible Instances: $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$.

Target concepts: 2^{96} (huge)

does not consider all types of training examples.

Solution → include all hypothesis.

ex: Sunny ∧ warm ∧ normal ∧ strong ∧ cool ∧ change ⇒ Yes.

and

* Unbiased hypothesis Space:

Providing a hypothesis capable of representing set of all examples

Possible Instances: $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$.

Target concepts: 2^{96} (huge)

(Practically, not possible)

all examples

Possible Instances: $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$.

Target concepts: 2^{96} (huge)

(Practically, not possible)

* Idea of Inductive Bias:

learner generalizes beyond the observed traing examples to infer new examples

'$>$' $\rightarrow$ inductively inferred from

$x > y \Rightarrow y$ is inductively inferred from x.

Example:

Learning algorithm $\rightarrow$ L.

training data (D_c) $= \{x, c(x)\}$

New instance $= x_i$

Represented as $L(x_i, D_c)$

$$(D_c \wedge x_i) > L(x_i, D_c)$$

* DECISION TREE LEARNING:

→ used in: tree structured classification & Regression

Dataset → Algorithm → classifies the data
(decision tree algorithm)

* 2 types of Nodes:

1. Decision Node
2. Leaf Node

Example: Loan System

(decides if the loan should be approved (Rejected)

↓ D

employed

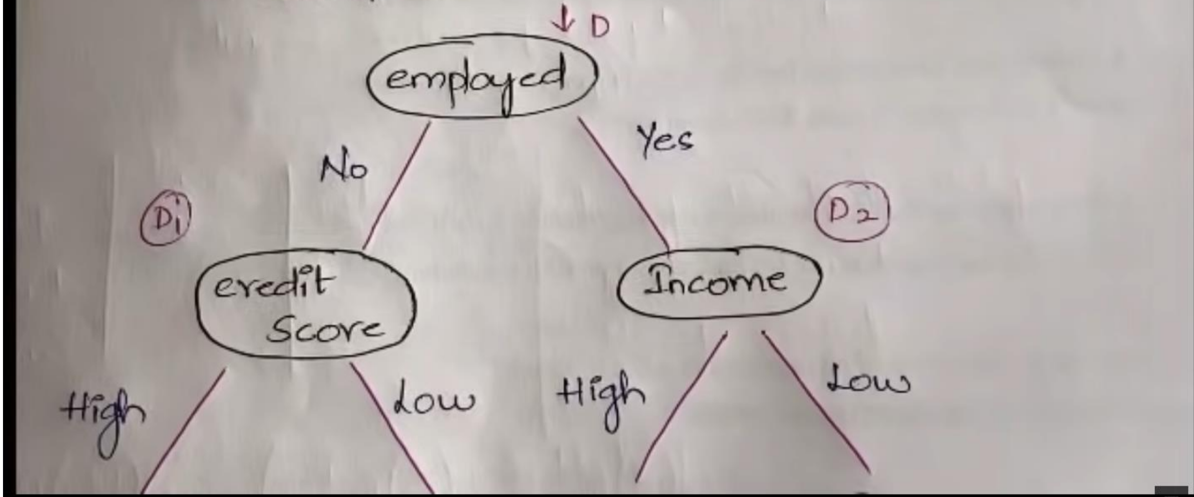
No — D_1

Yes — D_2

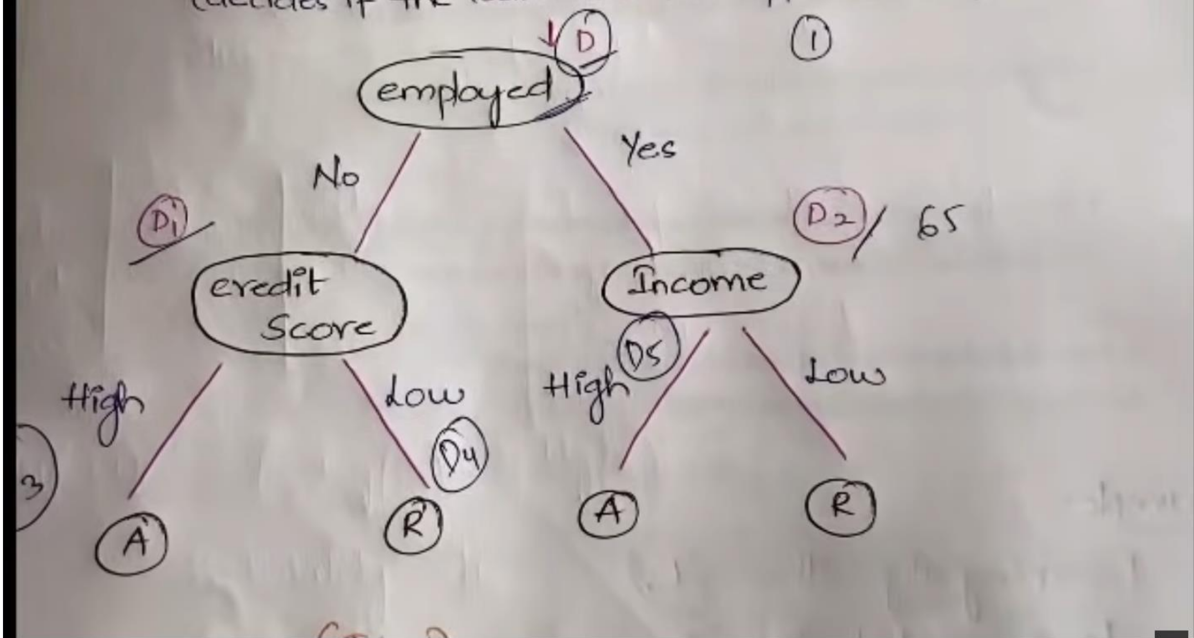
2. Leaf Node

Example: Loan System

(decides if the loan should be approved (Rejected)



(decides if the loan should be app



→ Algorithm (ID3)

1. In the given dataset, choose a target attribute
2. calculate Information gain of target attribute

$$IG = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

3. for remaining attributes, find entropy.

Entropy = IG × Probability

$$E(A) = \sum \frac{P_i+N_i}{P+N}\, I(P_i N_i)$$

1. In the given dataset, choose a target attribute
2. calculate Information gain of target attribute

$$IG = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

3. for remaining attributes, find entropy.

Entropy = IG × Probability

$$E(A) = \sum \frac{P_i+N_i}{P+N}\, I(P_i N_i)$$

4. Calculate Gain = IG − E(A)

Age	Competetion	Type	Profit
old	Yes	s/w	Down
old	No	s/w	Down
old	No	h/w	Down
mid	Yes	s/w	Down
mid	Yes	h/w	Down
mid	No	h/w	Up
mid	No	s/w	Up
new	Yes	s/w	Up
new	No	h/w	Up
new	No	s/w	Up

Step:1 Target Attribute = Profit

(2) Information gain:

$$IG = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

$\Rightarrow P = \text{count}(\text{down}) = 5$

$N = \text{count}(\text{up}) = 5$

$$= \frac{-5}{10}\log_2\left(\frac{5}{10}\right) - \frac{5}{10}\log_2\left(\frac{5}{10}\right)$$

$$= -\left(\frac{1}{2}\log_2(2^{-1}) + \frac{1}{2}\log_2(2^{-1})\right)$$

$$= -\left(\frac{1}{2}\times -1 \log_2 2 + \frac{1}{2}\times -1\log_2 2\right)$$

$$= -\left(\frac{1}{2}\times -1 + \frac{1}{2}\times -1\right)$$

old	No	s/w	Down
old	No	h/w	Down
mid	Yes	s/w	Down
mid	Yes	h/w	Down
mid	No	h/w	Up
mid	No	s/w	Up
new	Yes	s/w	Up
new	No	h/w	Up
new	No	s/w	Up

$IG = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$

$\Rightarrow P = \text{count(down)} = 5$

$N = \text{count(up)} = 5$.

$= \frac{-5}{10}\log_2\left(\frac{5}{10}\right) - \frac{5}{10}\log_2\left(\frac{5}{10}\right)$

$= -\left(\frac{1}{2}\log_2(2^{-1}) + \frac{1}{2}\log_2(2^{-1})\right)$

$= -\left(\frac{1}{2}\times -1\log_2 2 + \frac{1}{2}\times -1\log_2 2\right)$

$= -\left(\frac{1}{2}\times -1 + \frac{1}{2}\times -1\right)$

$= -(-1) = 1$

$IG = 1$

Step:1 Target Attribute = Profit

$\log_n a^m = m\log a$

---

③ Calculate entropy for remaining attributes.

$E(A) = \Sigma \frac{P_i + N_i}{P+N} I(P_i N_i)$ [IG × Probability]

(Age)

(1) Prepare a table for each attribute

rows - values of undertaken attribute (old, mid, new)

columns - values of target attribute (down, up)

	down	up
old	3	0
mid	2	2

Entropy = IG × Probability

P = down count

③ Calculate entropy for remaining attribu

$$E(A) = \Sigma \frac{P_i + N_i}{P+N} I(P_i N_i) \quad [IG \times Probability]$$

(Age)

(1) Prepare a table for each attribute

rows - values of undertaken attribute
(old, mid, new)

columns - values of target attribute
(down, up)

	down	up
old	3	0
mid	2	2
new	0	3.

Entropy = IG × Probability

P = down count

N = up count.

columns - values of target attribute
(down, up)

	down	up
old	3	0
mid	2	2
new	0	3

Entropy = IG × Probability

P = down count

N = up count

$$IG(old) = -\left(\frac{3}{3}\log\left(\frac{3}{3}\right) + \frac{0}{3}\log\left(\frac{0}{3}\right)\right) = 0$$

Probability = 3/10

$$Entropy(old) = 0 \times \frac{3}{10} = 0$$

$$IG(mid) = -\left(\frac{2}{4}\log\left(\frac{2}{4}\right) + \frac{2}{4}\log\left(\frac{2}{4}\right)\right)$$

$$= -\left(\frac{1}{2}\log\left(\frac{1}{2}\right) + \frac{1}{2}\log\left(\frac{1}{2}\right)\right)$$

$$= -\left(\frac{-1}{2} + \frac{-1}{2}\right) = -(-1) = 1$$

Probability = 4/10

$$Entropy(mid) = 1 \times \frac{4}{10} = 0.4$$

$$IG(new) = -\left(\frac{0}{3}\log\left(\frac{0}{3}\right) + \frac{3}{3}\log\left(\frac{3}{3}\right)\right) = 0$$

Probability = 3/10

$$Entropy(new) = 0 \times \frac{3}{10} = 0$$

$$Entropy(Age) = E(O) + E(m) + E(N) = 0 + 0.4 + 0 = 0.4$$

(4) $Gain = IG - E(A) = 1 - 0.4 = 0.6$

Step.

In the same way, calculate Gain of other attributes

Gain(Competetion) = 0.124

Gain (Type) = 0.

Gain (Age) = 0.6.

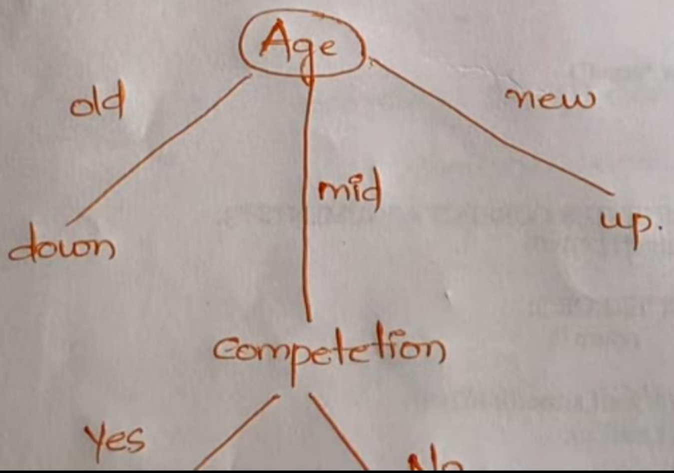
Highest gain → root node

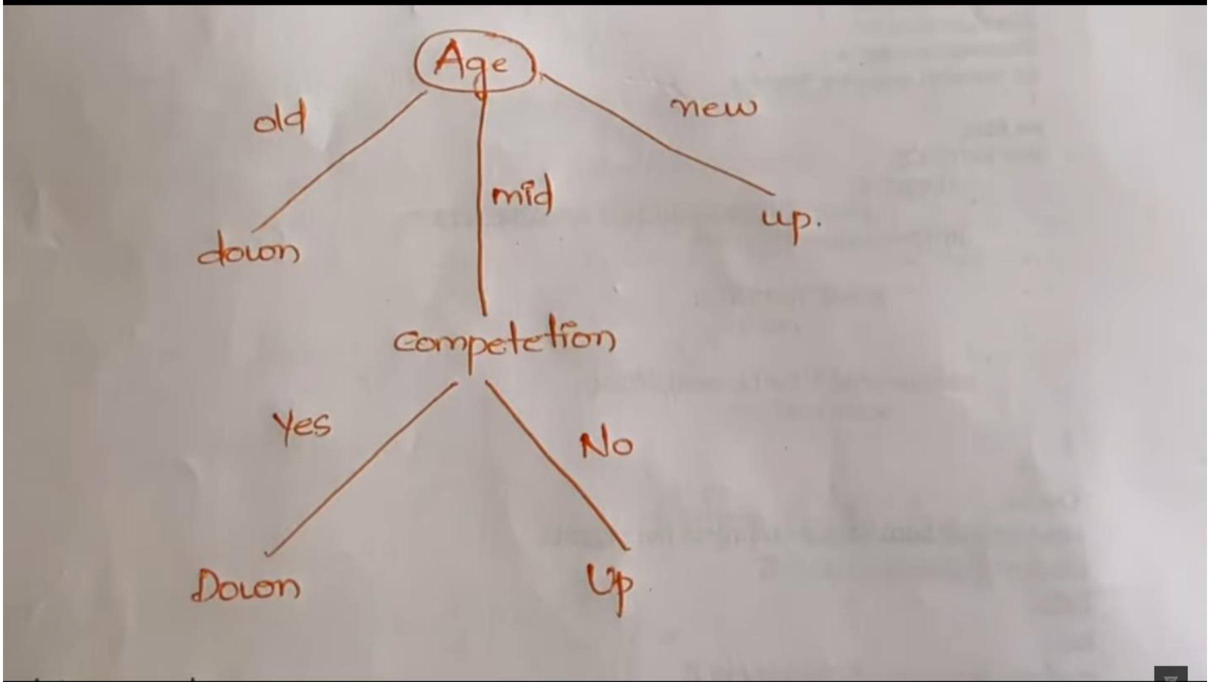
(Age)

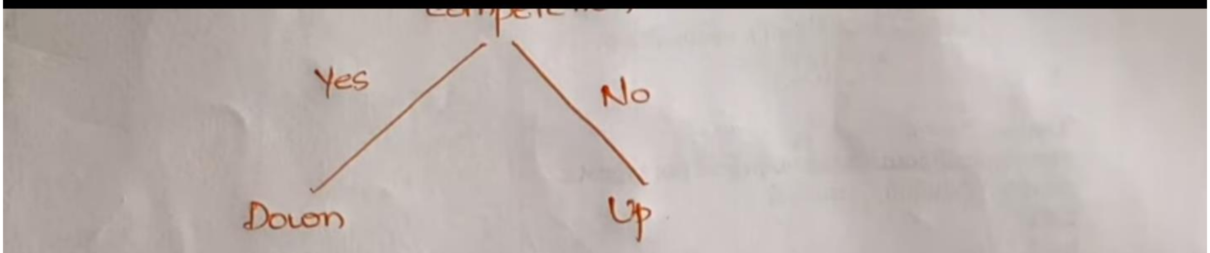
Gain (Age) = 0.6.

Highest gain → root node

(Age)







old → all down

mid → Some down and some up.

new → all up.

Why competetion, → next highest gain

Why not other attribute (Type)

In the same way, calculate Gain of other attributes

Gain(Competition) = 0.124

Gain(Type) = 0.

Gain(Age) = 0.6

Highest gain → root node

(Age)

* BACK PROPAGATION ALGORITHM:

(Backward propagation of error)

When error occurs, we go in backward direction

i.e output → hidden → input layer.

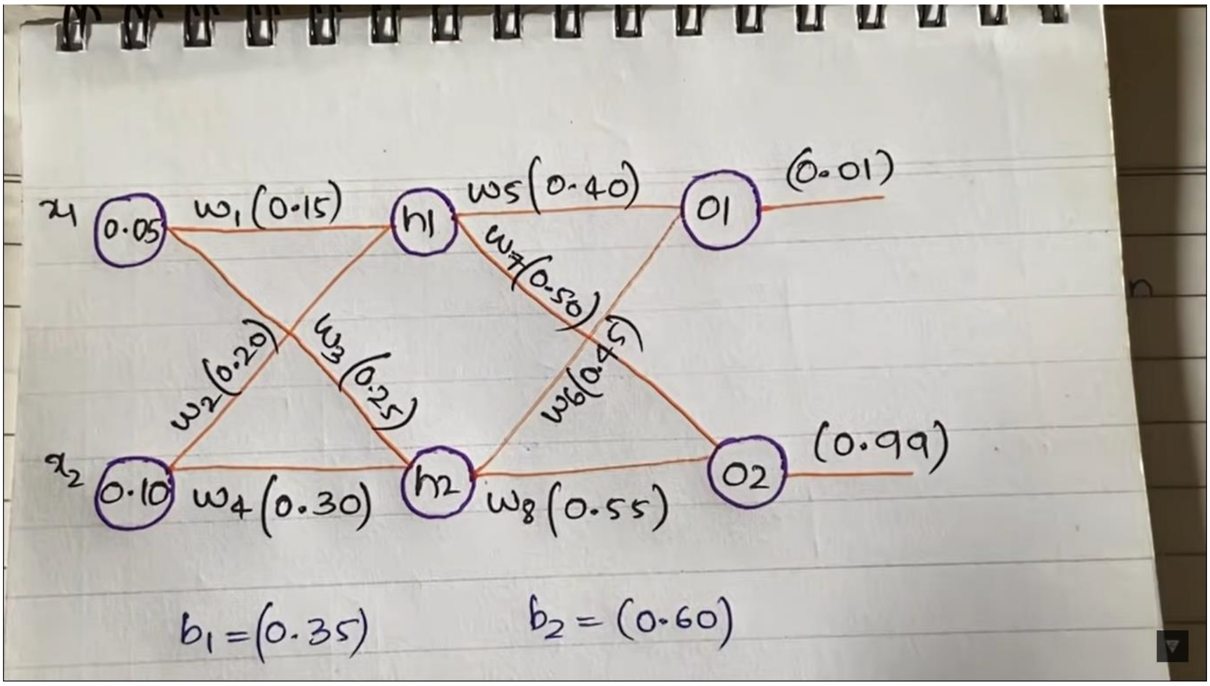
Part 1: Calculate forward Propagation Error.

1) Calculate h1 (in and out)

$$h_1(in) = w_1 x_1 + w_2 x_2 + b_1$$

$$= 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35$$

$$= 0.377$$



* BACK PROPAGATION ALGORITHM:

(Backward propagation of error)

When error occurs, we go in backward direction

i.e output → hidden → input layer.

Part1: Calculate forward Propagation Error.

1) calculate h1 (in and out)

$$h_1(in) = w_1 x_1 + w_2 x_2 + b_1$$

$$= 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35$$

$$= 0.377$$

(Backward propagation of error)

When error occurs, we go in backward direction

i.e output → hidden → input layer.

Part1: Calculate forward Propagation Error.

1) calculate h1 (in and out)

$$h_1(in) = w_1 x_1 + w_2 x_2 + b_1$$

$$= 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35$$

$$= 0.377$$

$$h_1(out) = 1 / \left(1 + e^{-h_1(in)}\right)$$

$$= \frac{1}{1 + e^{-(0.377)}} = 0.5932$$

2) Calculate h_2 (in and out)

$h_2(in) = x_1 w_3 + x_2 w_4 + b_1$

$= 0.05 \times 0.25 + 0.10 \times 0.3 + 0.35$ $\quad h_2(out) = 0.596$

$= 0.0125 + 0.03 + 0.35 = 0.3925$

3) Calculate O_1 (in and out)

$O_1(in) = h_1(out) \times w_5 + h_2(out) \times w_6 + b_2$

$= 0.593 \times 0.4 + 0.596 \times 0.45 + 0.6$

$= 1.105$

$O_1(out) = 1/(1+e^{-O_1(in)}) = 0.7513$

3) Calculate O_1 (in and out)

$O_1(in) = h_1(out) \times w_5 + h_2(out) \times w_6 + b_2$

$= 0.593 \times 0.4 + 0.596 \times 0.45 + 0.6$

$= 1.105$

$O_1(out) = 1/(1+e^{-O_1(in)}) = 0.7513$

4) Calculate O_2 (in and out)

$O_2(in) = h_1(out) \times w_7 + h_2(out) \times w_8 + b_2$

$= 0.5932 \times 0.50 + 0.5968 \times 0.55 + 0.6$

$= 1.22484$

$$O_2(out) = 1 / \left(1 + e^{-O_2(in)}\right)$$
$$= 0.7729$$

5) calculate E_{Total}

$$E_{total} = \Sigma \frac{1}{2}(target - o(p))^2$$
$$= E_{o1} + E_{o2}$$
$$= \frac{1}{2}(0.01 - 0.7513)^2 + \frac{1}{2}(0.99 - 0.7729)^2$$
$$= \frac{1}{2}(-0.7413)^2 + \frac{1}{2}(0.2171)^2$$

Part: 2: calculating backward propagation error
(output layer → hidden layer)

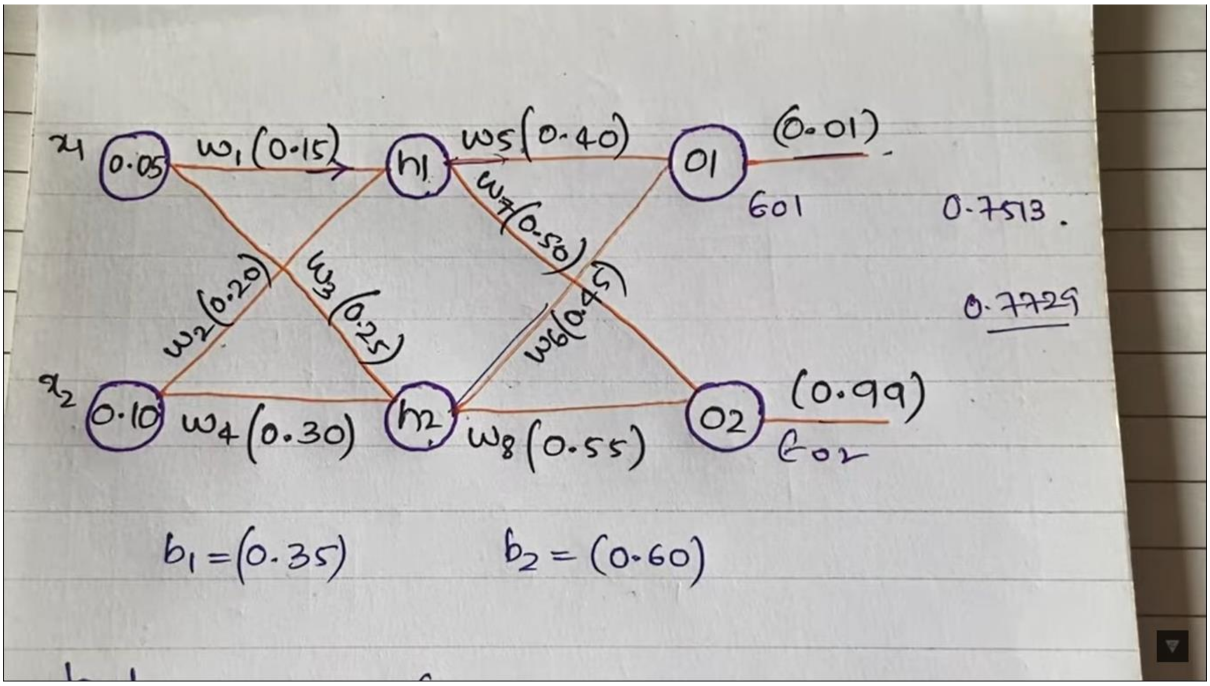
w_5, w_6, w_7 and w_8.

first letus adjust w_5

$$w_5^* = w_5 - \eta \frac{\partial E_{total}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial total}{\partial out_{o1}} = out_{o1} - target_{o1} = 0.751365 - 0.01$$



Part: 2: Calculating backward propagation error
(output layer → hidden layer)

w_5, w_6, w_7 and w_8.

first letus adjust w_5 0.6

$$w_5^* = w_5 - \eta \frac{\partial E_{total}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial total}{\partial out_{o1}} = out_{o1} - target_{o1} = 0.751365 - 0.01$$

$$= 0.741365$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial total}{\partial out_{o1}} = out_{o1} - target_{o1} = 0.751365 - 0.01$$
$$= 0.7413565$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1})$$
$$= 0.751365(1 - 0.751365) = 0.186815602$$

$$\frac{\partial net_{o1}}{\partial w_5} = out_{h1} = 0.59326992$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.7413565 \times 0.186815602 \times 0.59326992$$
$$= 0.08216704$$

$$w_5^* = w_5 - \eta \frac{\partial E_{total}}{\partial w_5}$$

$$= 0.4 - 0.6 \times 0.08216704 = 0.350699776.$$

Part:3: calculating backward propagation of error

Part:3: calculating backward propagation of error (hidden → input layer)

(w_1, w_2, w_3, w_4)

first lets adjust w_1

$$w_1^* = w_1 - \eta \frac{\partial E_{total}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out(h_1)}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out(h_1)} = \frac{\partial E_{o1}}{\partial out_{h_1}} + \frac{\partial E_{o2}}{\partial out_{h_1}}$$

$$\frac{\partial E_{total}}{\partial out(h_1)} = \frac{\partial E_{o1}}{\partial out_{h_1}} + \frac{\partial E_{o2}}{\partial out_{h_1}}$$

$$\frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial out_{h_1}}$$

$$\frac{\partial E_{o1}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}}$$

$$\frac{\partial E_{o2}}{\partial net_{o2}} \times \frac{\partial net_{o2}}{\partial out_{h_1}}$$

$$\frac{\partial E_{o2}}{\partial out_{o2}} \times \frac{\partial out_{o2}}{\partial net_{o2}}$$

$$\frac{\partial E_{o2}}{\partial out_{o2}} = (out_{o2} - target_{o2})$$

$$\frac{\partial\, \ldots}{\partial out(h_1)} \qquad \frac{\ldots}{\partial out h_1} + \frac{\ldots}{\partial out h_1}$$

$$\frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial out h_1} \qquad\qquad \frac{\partial E_{O2}}{\partial net_{O2}} \times \frac{\partial net_{O2}}{\partial out h_1}$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$\frac{\partial E_{O1}}{\partial out_{O1}} \times \frac{\partial out_{O1}}{\partial net_{O1}} \qquad\qquad \frac{\partial E_{O2}}{\partial out_{O2}} \times \frac{\partial out_{O2}}{\partial net_{O2}}$$

$$\frac{\partial E_{O2}}{\partial out_{O2}} = (out_{O2} - target_{O2})$$

$$= 0.772928465 - 0.99$$

$$= -0.217071535$$

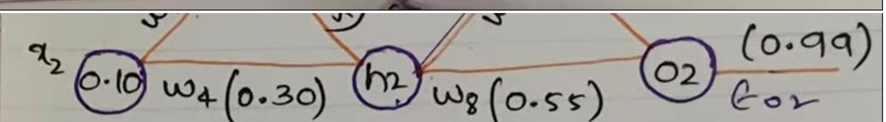
$\frac{\partial}{\partial out(h_1)}$

$$\frac{\partial E_{O1}}{\partial net_{O1}} \times \frac{\partial neto}{\partial outh}$$

$$\downarrow$$

$$\frac{\partial E_{O1}}{\partial out_{O1}} \times \frac{\partial out_{O1}}{\partial net_{O1}}$$

$\frac{\partial E_{O2}}{\partial out}$



$b_1 = (0.35) \qquad b_2 = (0.60)$

$$\frac{\partial out_{O_2}}{\partial net_{O_2}} = out_{O_2}(1 - out_{O_2})$$

$$= 0.7729(1 - 0.7729)$$

$$= 0.175510052$$

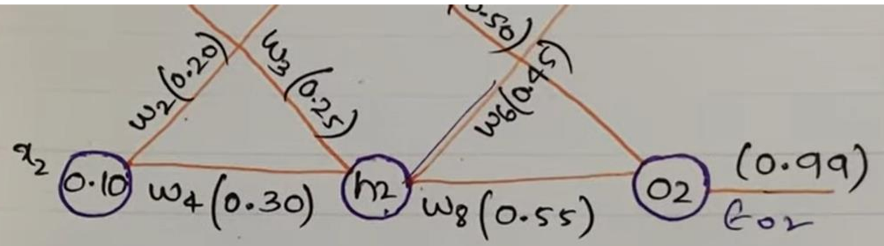
$\frac{\partial net_{O1}}{\partial out h_1}$ = on O1 from h1 $\Rightarrow w_5$

$$= 0.4$$

$\frac{dE_{total}}{dout(h_1)}$

$\frac{dE_{o1}}{dnet_{o1}} \times \frac{dn}{do}$

$\frac{dE_{o1}}{dout_{o1}} \times \frac{dou}{dn}$



$b_1 = (0.35)$ $b_2 = (0.60)$

$$\frac{dout_{o2}}{dnet_{o2}} = out_{o2}(1 - out_{o2})$$
$$= 0.7729(1 - 0.7729)$$
$$= 0.175510052$$

$b_1 = (0.35)$ $b_2 = (0.60)$

$$\frac{dout_{o2}}{dnet_{o2}} = out_{o2}(1 - out_{o2})$$
$$= 0.7729(1 - 0.7729)$$
$$= 0.175510052$$

$\frac{dnet_{o1}}{douth_1}$ = on O1 from h1 $\Rightarrow w_5$

$= 0.4$

$\frac{dnet_{o2}}{}$ = on O2 from h1 $\Rightarrow w_7$

$$\frac{\partial E_{total}}{\partial out(h_1)} =$$

$$\frac{\partial net_{o1}}{\partial out_{h_1}} = \text{on } o1 \text{ from } h1 \Rightarrow w_5 = 0.4$$

$$\frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial ne}{\partial ou}$$

$$\frac{\partial net_{o2}}{\partial out_{h_1}} = \text{on } O_2 \text{ from } h1 \Rightarrow w_7 = 0.50$$

$$\frac{\partial E_{o1}}{\partial out_{o1}} \times \frac{\partial ou}{\partial n}$$

$$0.13849856 \times 0.4 + (-0.0380982 \times 0.50)$$
$$= 0.055399425 + (-0.019049119)$$
$$= 0.036350306.$$

$$w_1^* = w_1 - \eta \frac{\partial E_{total}}{\partial w_1}.$$

$$\frac{\partial E_{total}}{\partial w_1} = \boxed{\frac{\partial E_{total}}{\partial out_{h_1}}} \times \frac{\partial out(h_1)}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out(h_1)} = \frac{\partial E_{o1}}{\partial out_{h_1}} + \frac{\partial E_{o2}}{\partial out_{h_1}} \quad = 0.036350306.$$

$$\frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial out_{h_1}} \qquad \frac{\partial E_{o2}}{\partial net_{o2}} \times \frac{\partial net_{o2}}{\partial out_{h_1}}$$

$$\frac{\partial E_{o1}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \qquad \frac{\partial E_{o2}}{\partial out_{o2}} \times \frac{\partial out_{o2}}{\partial net_{o2}}$$

$$w_1^* = w_1 - \eta \frac{\partial E_{total}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out(h1)}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1}$$

$$\frac{\partial out_{h1}}{\partial net_{h_1}} = out_{h1}(1 - out_{h1})$$

$$= 0.241300709$$

$$\frac{\partial net_{h1}}{\partial w_1} = \frac{\partial}{\partial w_1}(w_1 x_1 + \underbrace{w_2 x_2}_{0} + \underbrace{b_1}_{0})$$

$$= x_1 = 0.05$$

$$= 0.241300709$$

$$\frac{\partial net_{h1}}{\partial w_1} = \frac{\partial}{\partial w_1}(w_1 x_1 + \underbrace{w_2 x_2}_{0} + \underbrace{b_1}_{0})$$

$$= x_1 = 0.05$$

$$\frac{\partial E_{total}}{\partial w_1} = 0.036350306 \times 0.241300709 \times 0.05$$

$$= 0.00438568$$

$$w_1^* = w_1 - \eta \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.6 \times 0.00438568$$

$$= 0.1497368592$$