Introduction

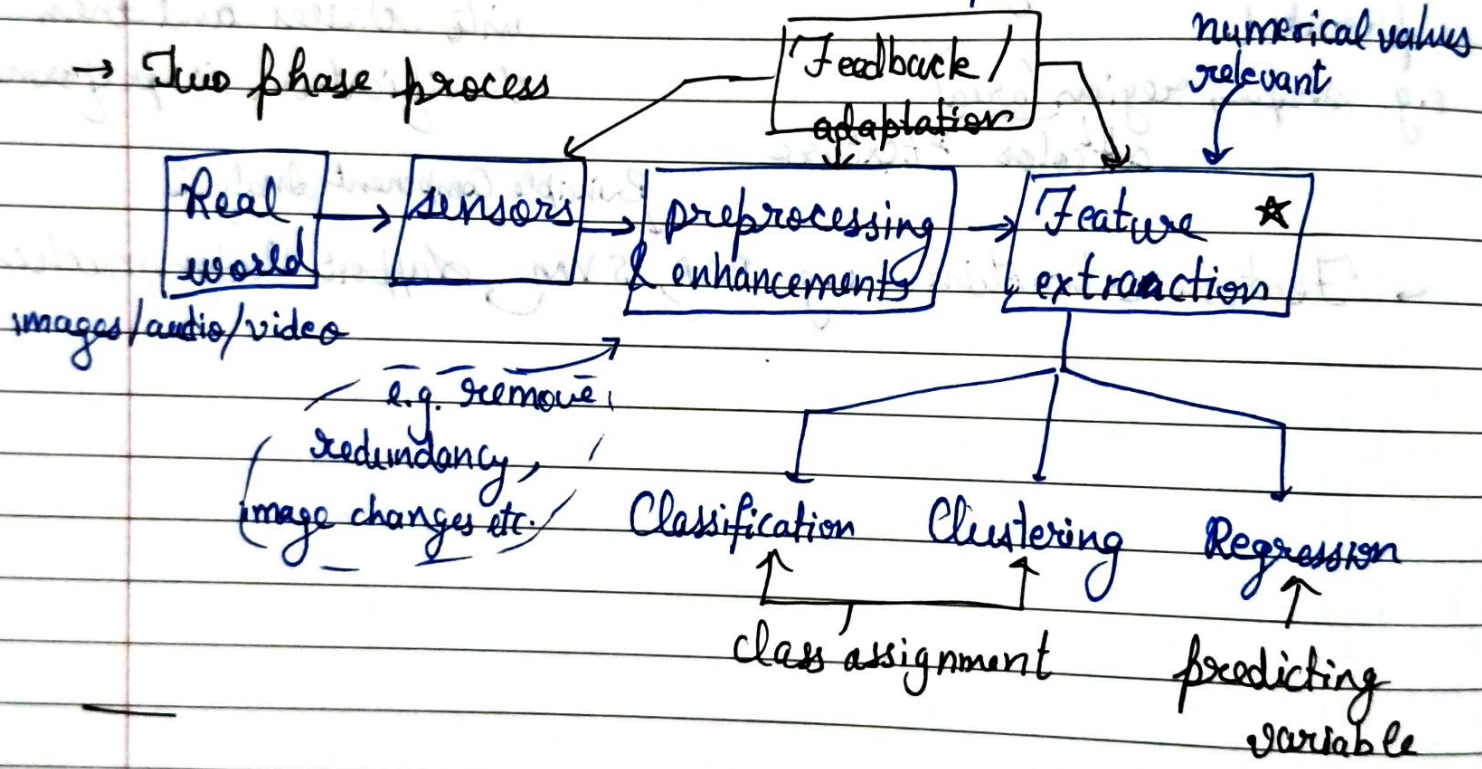
→ Pattern Recognition : identifying and analysing patterns in data is called pattern recognition.

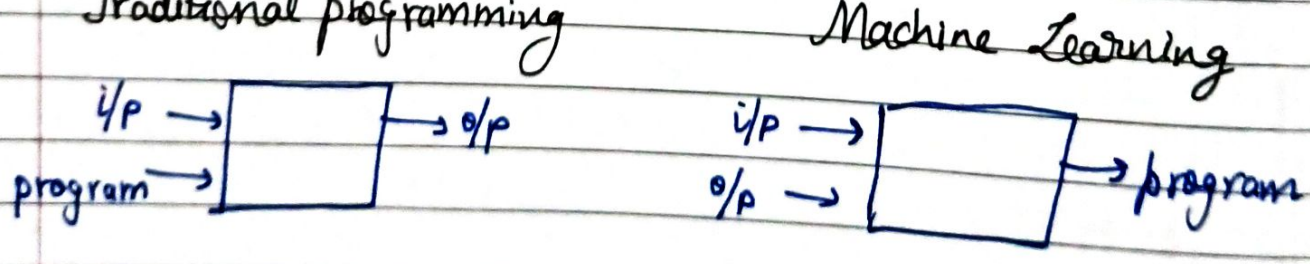
→ Machine Learning is a type of pattern recognition done automatically by machine.

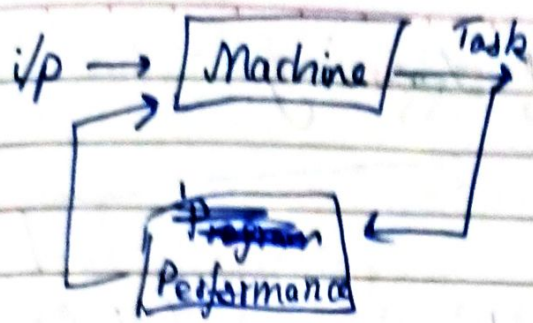
→ Pattern : similarities which are not identical but can help in classification.

→ ~~There are several standard models for PR~~

→ Two phase process







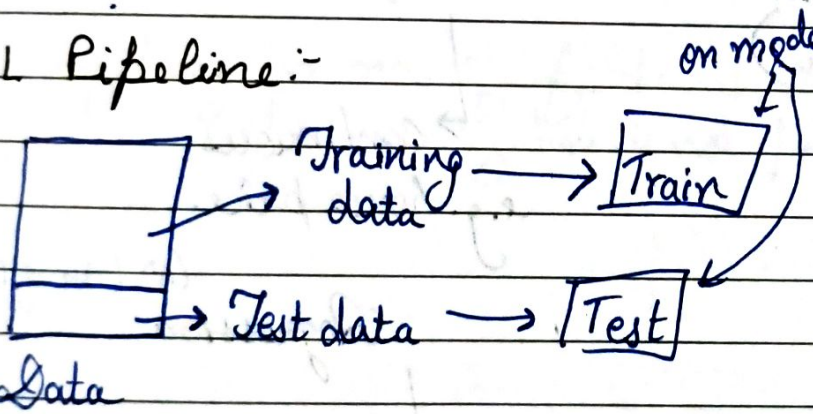
⇒ Advantages
↳(i) We are not explicitly programming
↳(ii) Handle large data
↳(iii) Make human being independent

Feature matrix :-

Features →

Samples ↓		F_1	F_2	F_3	F_4	...
	S_1					
	S_2					
	S_3					
	⋮					

ML Pipeline :-



⇒ Traing data : test data is generally 70:30, 80:20, 90:10

50:50 → 60:40 → 70:40 | The efficiency keeps improving. We stop when saturation is reached.

Types of Learning

- Supervised (Classification)
- Unsupervised (Clustering)
- Reinforcement

→ Supervised Learning :-

↳ A teacher provides labelled training sets to train a classifier.

Supervised learning

- classification (binary / multiclass)
  - ↳ discrete
  - e.g. Yes/No
- regression
  - ↳ continuous
  - e.g. house price

↳ Classification

- ↳ Logistic regression
- ↳ Bayesian classification
- ↳ KNN
- ↳ Decision tree
- ↳ Random forest
- ↳ Neural Network
- ↳ Support Vector Machine

(Not in syllabus)

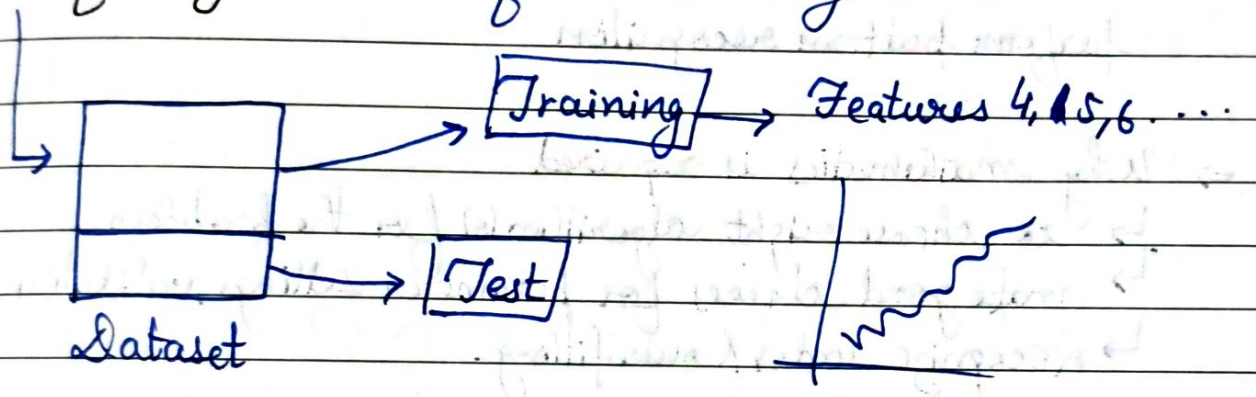
Regression

- ↳ Linear regression
- ↳ KNN regression
- ↳ Decision tree regression
- ↳ SVM regression
- ↳ Radial bands network for regression.

→ Unsupervised learning

↳ Clustering
  ↳ K means clustering
  ↳ ~~K~~medoid K medoids clustering
  ↳ Heirarachical clustering
  ↳ DB scan
  ↳ Fuzzy means clustering

→ Overfitting (Curse of dimensionality)

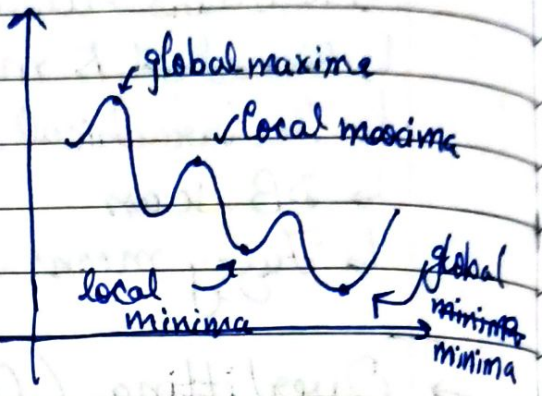


→ Underfitting → no. of features is too less
  ↳ generalization is not possible.

⇒ Optimization

→ Optimization is the process of training the model iteratively to maximize / minimize a value.

→ Common optimization methods are :
- Maximum likelihood
- Expectation maximization
- Gradient descent



∴ We need mathematics in order to perform pattern recognition

⇒ Why mathematics is required
↳ to choose right algorithm(s) for the problem
↳ make good choices for parameter settings, validation strategies
↳ Recognize under & over fitting.

→ Mathematical notation
$|B|$ → cardinality
$\|\vec{v}\|$ → norm
$\mathbb{R}^n$ → dimensionality

→ Probability space : It is a random process / experiment with 3 components (Ω, E, P)

Ω → outcomes (sample space)
E → events
P → distribution

Discrete space	Continuous space
→ $\|\Omega\|$ is finite	→ $\|\Omega\|$ is infinite
→ Σ is used	→ $\int$ is used

⇒ In PR, data must be unbiased / unskewed.

→ pdf : probability density function → continuous
pmf : probability mass function → discrete

→ probability function → maps probability to sample space
↳ area is always 1.

→ cdf : cumulative density function.
↳ probability till a value e.g. $P(x \leq 3)$
$= P(x=1) + P(x=2) + P(x=3)$

⇒ Random variable
↳ It is a function which associates number x with each outcome of a process.

→ Multivariate Probability Distribution
↳ several random processes occur (in parallel or in sequence)
↳ can be described as joint prob. of several random variables.

e.g. $y = a_1x_1 + a_2x_2 + \cdots a_nx_n + c$, we want to find
(i) change in y w.r.t. x_1, change in y w.r.t. x_2...
(ii) relationship b/w $x_1, x_2 \cdots x_n$

→ prob. given one event occurs : $P(X=x|Y=y) = P(X=x, Y=y) / P(Y=y)$

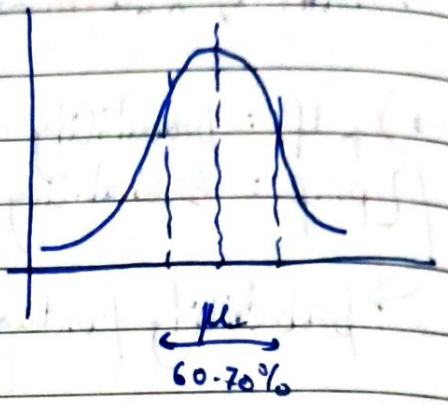
⇒ same concepts of joint, marginal & conditional prob. applies for continuous & multivariate distribution.

before experiment ↓ | after experiment ↓

→ Expected value / mean

⇒ $E(f) = \sum_i p(x_i) \cdot f(x_i)$

⇒ Generally, missing values are filled with the average value



$0.4 \times 2 + 0.6 \times 0.5 = 0.8 + 0.3 = 1.1$

$0.3 \times 3 + 0.9 \times 0.2 = 0.9 + 0.14 = 1.04$

→ Expected value = mean in ideal case

→ Variance = mean value - expected value

⇒ $\text{mean} = \sum_i (x_i)/N$

the spread ⇒ $\text{Var}(X) = E[(X-\mu)^2] = \sum_i (x_i-\mu)^2 \cdot p(x_i)$; $\mu = E(X)$

⇒ Standard deviation = $\sqrt{\text{Var}(X)}$

⇒ Covariance

↳ tendency for x & y to deviate from their means in same (or opposite) directions

★★ $\frac{1}{N}$ → population | $\frac{1}{N-1}$ → sample (a part of population)

→ $\text{cov}(x,y) = \frac{1}{N-1} \sum_{i=1}^{n} (x_i-\mu_x)(y_i-\mu_y)$

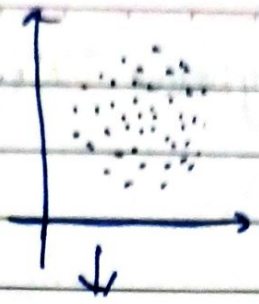
↓

e.g. only a small region

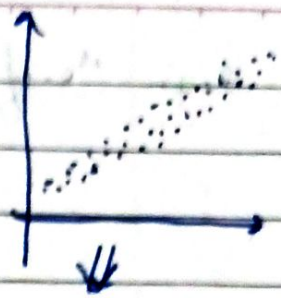
→ $f(x_i) = (x_i - \mu_x)$

$g(y_i) = (y_i - \mu_y)$

$\therefore cov = \frac{1}{N-1} \sum_i f(x_i) g(y_i)$



No covariance



positive covariance

⇒ Correlation

↳ we will only study Pearson's correlation coeff.

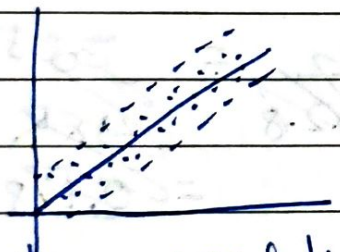
$$corr(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

↳ $-1 \leq corr \leq 1$

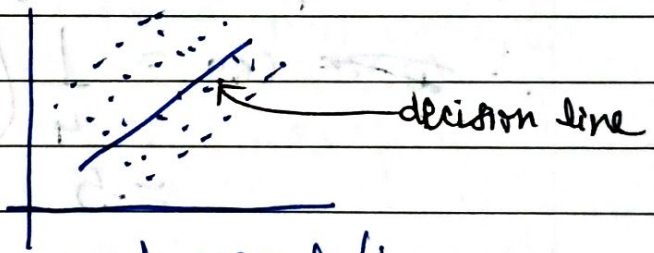
if -1 to 0 : negatively correlated
if at 0 : no relation b/w features
if 0 to 1 : positively correlated

↳ It only reflects linear dependence b/w variables.

↳ we will study techniques to convert non linear to linear relationship



strong correlation



weak correlation

$$\therefore corr(x,y) = \rho(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y} = \frac{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}$$

where $\sigma_x = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{N-1}$, $\sigma_y = \sum_{i=1}^{N} \frac{(y_i - \bar{y})^2}{N-1}$

and $\bar{x} = \sum_{i=1}^{n} x_i / N$, $\bar{y} = \sum_{i=1}^{N} y_i / N$

e.g.

No. of study hours	No. of sleeping hours
2	10
4	9
6	8
8	7
10	6

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$	y_i	$y_i - \bar{y}$	$(y_i - \bar{y})^2$	$(x_i - \bar{x})(y_i - \bar{y})$
2	-4	16	10	2	4	-8
4	-2	4	9	1	1	-2
6	0	0	8	0	0	0
8	2	4	7	-1	1	-2
10	4	16	6	-2	4	-8
30		40	40		10	-20

$\therefore \sigma_x = \frac{40}{4} = 10, \quad \sigma_y = \frac{10}{4} = 2.5$

$\therefore$ Cov $= \frac{1}{4}(-20) \times \frac{1}{10 \times 2.5} = \frac{-5}{25} = -0.2$

$\therefore$ Corr $= \frac{-5}{\sqrt{10} \times 2.5 \sqrt{25}} = \frac{-5}{25} = -1$

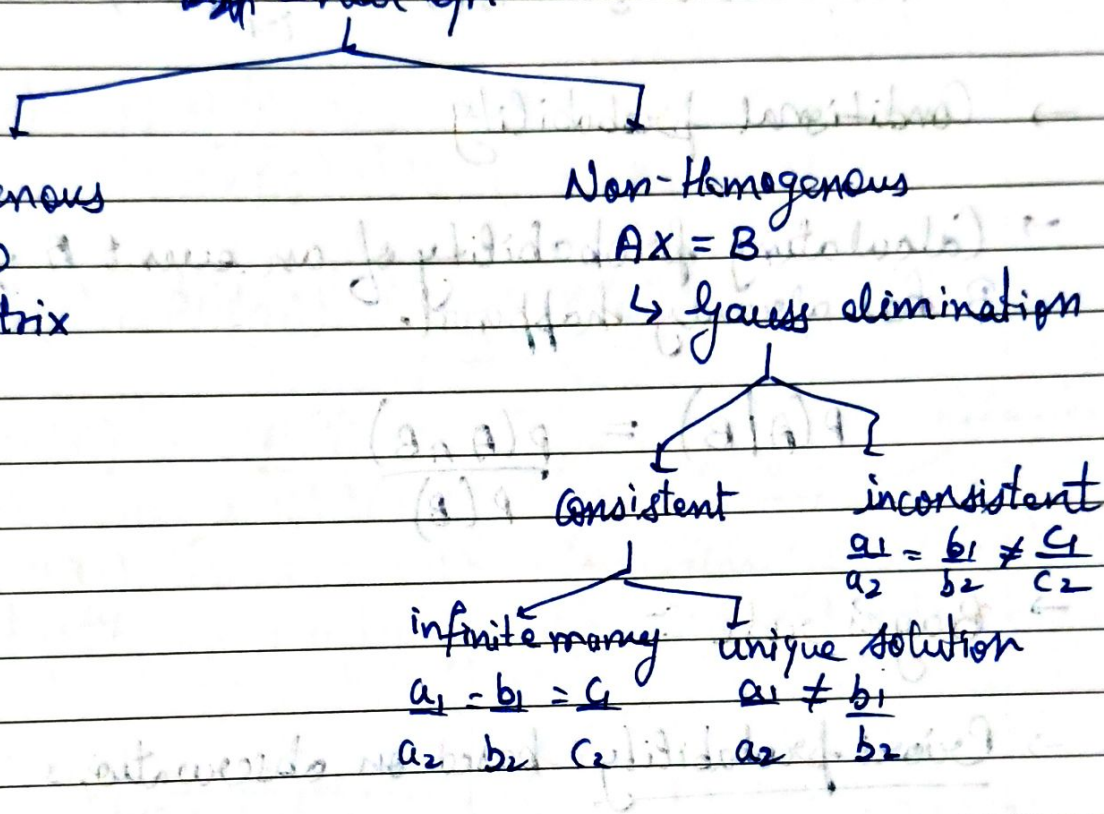
$= -0.2$

→ Applications of linear algebra
- ↳ operations between vectors & matrices
- ↳ coordinate transformations
- ↳ dimensionality reduction
- ↳ linear regression
- ↳ solution of linear system of equations etc.

e.g If two features are highly correlated, we can remove one of them.

→ SVD → Singular value decomposition
↳ it is used to reduce the no. of dimensions

→ Linear regression → relⁿ



Date 2 / 9 / 24

→ Axioms of Probability

① $0 \leq P(A) \leq 1$

② $P(S) = 1$

③ If $A_1, A_2 \ldots A_n$ are mutually exclusive
$P(A_i \cap A_j) = 0$ then
$P(A_1 \cup A_2 \cup A_3 \cdots A_n) = \sum_{i=1}^{n} P(A_i)$

→ Conditional probability

→ Calculating probability of an event A given that event B has already happened.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

→ Bayes rule :-

→ Priori probability : based on observations

→ Decision rule :- rule which the model forms to classify the data.
→ It is never based on priori probability

Raw data : data captured through sensors
↳ our model cannot understand raw data

e.g. Feature vector for salmon & seabass

→ Posterior probability : $P(\omega_i / x_i)$
↑ class ↑ feature vector

how much this feature is likely if output is the given class.

→ $P(x_i/\omega_i)$ ⇒ likelihood probability

⇒ Conditional probability :-

$$P(A/B) = P(A \cap B)/P(B) \quad \text{-(i)}$$

⇒ Bayes rule

$$P(B/A) = P(A \cap B)/P(A) \quad \text{-(ii)}$$

from (i) and (ii), $P(A \cap B) = P(A) \cdot P(B/A) + P(B) \cdot P(A/B)$

$$\therefore P(A/B) = \frac{P(A) \cdot P(B/A)}{P(B)}$$

$P(A/B)$ ↑ posterior probability

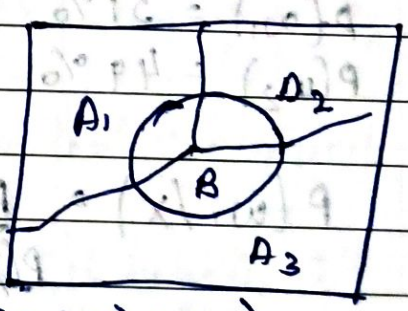
$P(A)$ ← priori

$P(B/A)$ ← likelihood

$P(B)$ ← (marginalization or normalization)

→ Intersection of events

$$P(B) = P(A_1 \cap B) + P(A_2 \cap B) + P(A_3 \cap B)$$



$$= P(B/A_1)P(A_1) + P(B/A_2)P(A_2) + P(B/A_3)P(A_3)$$

$$\therefore \boxed{P(B) = \sum_{i=1}^{n} P(B/A_i) P(A_i)}$$

→ **Bayes Theorem** :-

$$P(A_i/B) = \frac{P(B/A_i) \cdot P(A_i)}{P(B)}$$

where, $P(B) = \sum_{i=1}^{n} P(B/A_i) P(A_i)$

→ In terms of PR,

$$P(\omega_i/x) = \frac{P(x/\omega_i) P(\omega_i)}{\sum_{k=1}^{n} P(x/\omega_k) P(\omega_k)}$$

class ↑ feature

e.g. In a country, there are 51% of adults are male, 9.5% males smoke cigar whereas 1.7% of females smoke cigares. Find probability that person smokes cigar is a male.

Ans. $P(\omega_1) = 51\%$ $\quad P(x/\omega_1) = 9.5\%$

$P(\omega_2) = 49\%$ $\quad P(x/\omega_2) = 1.7\%$

$$P(\omega_1/x) = \frac{P(x/\omega_1) P(\omega_1)}{P(x/\omega_1) P(\omega_1) + P(x/\omega_2) P(\omega_2)}$$

$$= \frac{9.5 \times 51}{9.5 \times 51 + 49 \times 1.7}$$

$$= \frac{484.5}{567.8} = \boxed{0.85}$$

→ **Vector** :-

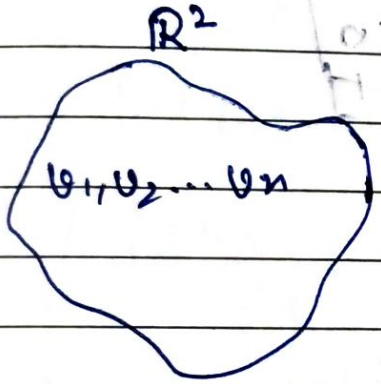
→ Addition, subtraction, dot product, angle b/w 2 vectors, length, magnitude

→ Linear combination of vectors.

Consider a set $S = \{v_1, v_2 \cdots v_n\}$; then, a new vector of same space is a linear combination of vectors from same space

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$$

e.g.



$\alpha_1, \alpha_2, \alpha_3 \cdots \alpha_n \rightarrow$ scalars

→ Basis : dimension of vector space which can span the whole vector space
→ they must be linearly independent i.e. must be orthonormal
↳ orthogonal + unit length

→ Cosine similarity.

→ Span: Let V be a vector space and let S be a subset of V, $S = \{v_1, v_2 \cdots v_n\}$. Then, span of S is the set of all linear combination of vectors $v_1, v_2 \cdots v_n$ denoted by span (v)

→ $\alpha_1 v_1 + \alpha_2 v_2 + \cdots \alpha_n v_n$ belongs to span of v

e.g. A vector $\vec{v} = \begin{bmatrix} 19 \\ 10 \\ -1 \end{bmatrix}$ in span (S) where $S = \left\{ \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \begin{bmatrix} -5 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \\ -4 \end{bmatrix} \right\}$. Find value of α_1, α_2 and α_3.

[Ans: $\begin{bmatrix} 4 \\ -1 \\ 2 \end{bmatrix}$]

Ans $\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} \alpha_1 + \begin{bmatrix} -5 \\ 0 \\ 1 \end{bmatrix} \alpha_2 + \begin{bmatrix} 1 \\ 7 \\ -4 \end{bmatrix} \alpha_3 = \begin{bmatrix} 19 \\ 10 \\ -1 \end{bmatrix}$

$\therefore \begin{bmatrix} 3 & -5 & 1 \\ -1 & 0 & 7 \\ 2 & 1 & -4 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 19 \\ 10 \\ -1 \end{bmatrix}$

$\therefore \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 19 \\ 10 \\ -1 \end{bmatrix} \times \begin{bmatrix} 3 & -5 & 1 \\ -1 & 0 & 7 \\ 2 & 1 & -4 \end{bmatrix}^{-1}$

$\begin{bmatrix} 3 & -5 & 1 \\ -1 & 0 & 7 \\ 2 & 1 & -4 \end{bmatrix}^{-1} = \begin{bmatrix} -7 & 10 & -1 \\ -19 & -14 & -13 \\ -35 & -22 & -5 \end{bmatrix}^T \times \frac{1}{(-72)}$

$\therefore \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 19 \\ 10 \\ -1 \end{bmatrix} \begin{bmatrix} -7 & -19 & -35 \\ 10 & -14 & -22 \\ -1 & -13 & -5 \end{bmatrix} \times \frac{1}{-72} = \begin{bmatrix} 4 \\ -1 \\ 2 \end{bmatrix}$

$\therefore \alpha_1 = 4, \alpha_2 = -1, \alpha_3 = 2$

→ Norm of a vector

(i) $\|x\| \geq 0$ & $\|x\| = 0$ only if $x = 0$

(ii) $\|x+y\| = \|x\| + \|y\|$ → linear transformation

(iii) $\|\alpha x\| = |\alpha| \|x\|$

Pnorm $= \|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$

∴ l_1 norm, $\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$

l_2 norm, $\|x\|_2 = \sqrt{(x_1)^2 + (x_2)^2 + \cdots (x_n)^2}$

~~l_3 norm~~

l_∞ norm, $\|x\|_\infty = \max(|x_1|, |x_2|, \ldots, |x_n|)$

→ ~~Similarity measures~~

~~Supremum~~

① ~~Supremum~~ distance

② Manhattan distance

③ Euclidean distance

④ Minkowski distance

Date 7 / 9 / 24

Bayesian Decision Theory

→ Priori probabilities: $P(\omega_1)$, $P(\omega_2)$ → calculate using N_1/N, N_2/N

→ Likelihood probability: $P(X/\omega_i)$

↳ we will calculate feature vectors for each instance

→ Posterior probability: $P(\omega_i/X) = \frac{P(X/\omega_i)\,P(\omega_i)}{P(X)}$

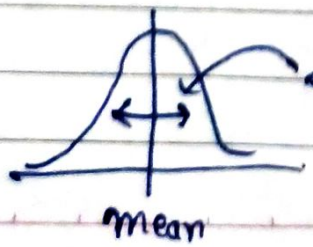
where $P(X) = \sum_{i=1}^{n} P(X/\omega_i)\,P(\omega_i)$

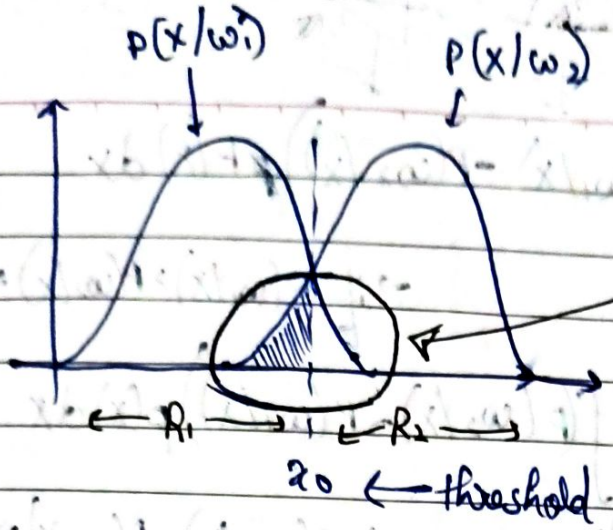
→ Bayes classification: if $P(\omega_1/X) > P(\omega_2/X)$
then new sample is ω_1, else ω_2

$\therefore p(X/\omega_1)\cdot p(\omega_1) > p(X/\omega_2)\cdot p(\omega_2)$

↳ if $p(\omega_1) = p(\omega_2) = 1/2$, $P(X/\omega_1) > P(X/\omega_2) \Rightarrow \omega_1$

⇒ Most of examples in real world follows normal distribution





If $x < x_0$, it will classify in ω_1
$x > x_0$, it will classify in ω_2

$$\therefore \text{probability error, } P_e = \left(\int_{-\infty}^{x_0} P(x/\omega_2)\,dx + \int_{x_0}^{\infty} P(x/\omega_1)\,dx \right) \frac{1}{2}$$

→ How we can minimise this error probability

↳ Bayesian classifier is an optimal classifier.

$$P_e = P(x \in R_2, \omega_1) + P(x \in R_1, \omega_2)$$

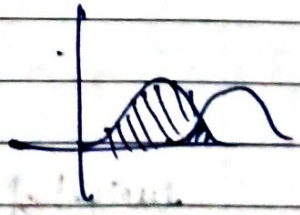
↳ i.e. $P(x \in R_2 \text{ and } x \in \omega_1)$ or $P(x \in R_2 \cap \omega_1)$

From here, $P_e = P(x \in R_2/\omega_1) P(\omega_1) + P(x \in R_1/\omega_2) P(\omega_2)$

↳ i.e. $P(x \in R_2 / x \in \omega_1) \cdot P(x \in \omega_1)$

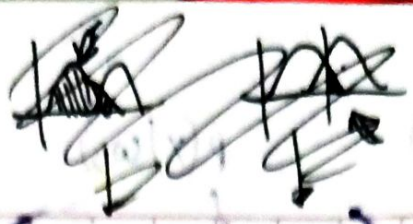
$$\therefore P_e = P(\omega_1) \int_{R_2} P(x/\omega_1)\,dx + P(\omega_2) \int_{R_1} P(x/\omega_2)\,dx$$

[Now, for R_1, $P(\omega_1/x) > P(\omega_2/x)$
R_2 : $P(\omega_2/x) > P(\omega_1/x)$



$$\therefore P_e = \int_{R_1} P(\omega_1/x) P(x)\,dx + \int_{R_2} P(\omega_2/x) P(x)\,dx$$

[Using Bayes rule]



$$\therefore P_e = P(\omega_1) - \int_{R_1} \left(P(\omega_1/x) - P(\omega_2/x)\right) p(x)\,dx$$

→ if $P(\omega_1/x) > P(\omega_2/x) \Rightarrow$ error min

Similarly, $P_e = P(\omega_2) - \int_{R_2} \left(P(\omega_2/x) - P(\omega_1/x)\right) p(x)\,dx$

→ if $P(\omega_2/x) > P(\omega_1/x) \Rightarrow$ error min

Therefore, Bayes decision rule minimize the error.

→ ω = state of nature = class = random variable
→ $p(x)$ = probability density function = evidence
→ $p(x/\omega_i)$ = conditional probability density = likelihood

→ $P(\text{error}) = \min\left(P(\omega_1), P(\omega_2)\right)$ ⇐ prior probability decision rule

---

Risk :-

↳ we also consider $\lambda(\alpha_i/\omega_j)$ i.e. weight function.
where, α_i → action
∴ It tells how much risk is there eg. tumor is benign or malignant

→ $\text{risk} = R = \lambda_{12} P(\omega_1) \int_{R_2} p(x/\omega_1)\,dx + \lambda_{21} P(\omega_2) \int_{R_1} p(x/\omega_2)\,dx$

λ_{12} : factor — weight of class 1 but assigning it to class 2. ; R_2 : original

λ_{21} : factor — weight of class 2 but assigning it to class 1. ; R_1 : original

e.g. if tumor $\omega_1 \rightarrow$ malign
$\omega_2 \rightarrow$ benign
then $\lambda_{12} > \lambda_{21}$

→ In general for m classes,

$$r_k = \sum_{i=1}^{m} \lambda_{ki} \int_{R_i} P(x/\omega_k)\, dx$$ ← belonging to class k but classified to other classes

∴ total risk, $r = \sum_{k=1}^{m} r_k P(\omega_k)$

→ ∴ $x \in R_i$, if $\sum_{k=1}^{m} \lambda_{ki} P(x/\omega_k) P(\omega_k) < \sum_{k=1}^{m} \lambda_{kj} P(x/\omega_k) P(\omega_k)$

$\forall\ j \neq i$

because in that case the risk is minimum.

→ ∴ $l_1 = \lambda_{11} P(x/\omega_1) P(\omega_1) + \lambda_{21} P(x/\omega_2) P(\omega_2)$ } for 2 class
$l_2 = \lambda_{12} P(x/\omega_1) P(\omega_1) + \lambda_{22} P(x/\omega_2) P(\omega_2)$ } classifier

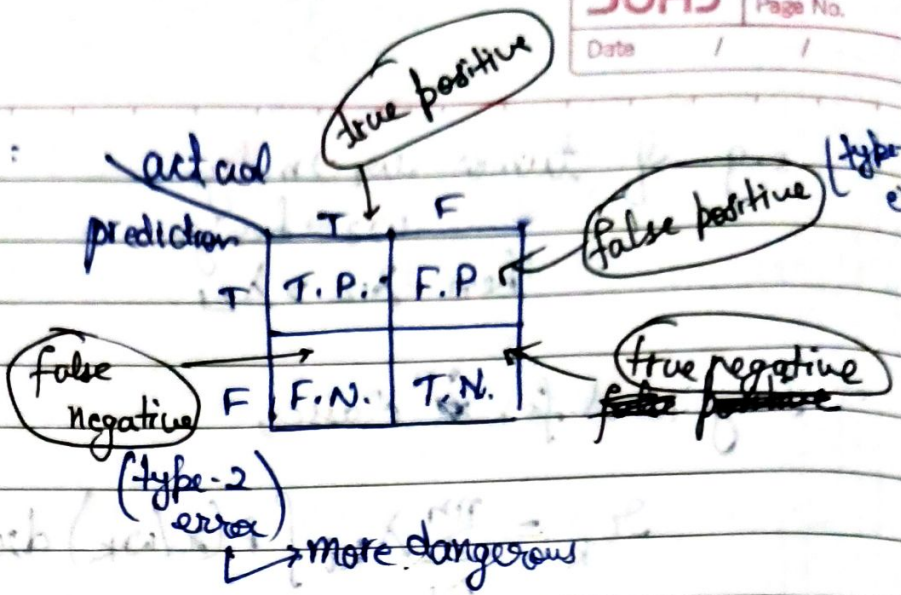
↳ 4 weights needed, where, $\lambda_{11} = \lambda_{22} = 0$

∴ we assign $x \rightarrow \omega_1$ if $l_1 < l_2$ and vice versa

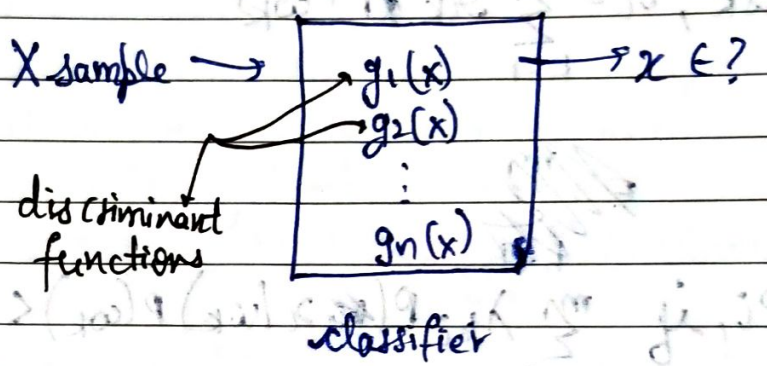
∴ loss matrix, $L = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix}$ ← in our case $\lambda_{12} > \lambda_{21}$

∴ $P(X/\omega_2)\lambda_{21} < P(X/\omega_1)\lambda_{12}$ in our case.

Confusion matrix :



→ Discriminant function



→ It is opposite to risk function.
→ It is monotonically increasing function.

∴ if $g_i(x) > g_j(x)$, $x \in \omega_i$

$$g_i(x) = -R(\alpha_i / x)$$

2015 Page No.
Date 10 / 9 / 24

Discriminant function

→ It is a function which is used to create decision boundary / decision surface.

as we know, risk, $R(\alpha_i/x) = \sum_{i=1}^{c} \lambda(\alpha_i/\omega_i) P(\omega_j/x)$ (c: no. of classes; α: action)

$\approx 1 - P(\omega_i/x)$

→ ∴ discriminant function $g_i(x) = \max_i \{R(\alpha_i/x)\}$

$= -R(\alpha_i/x)$

$= P(\omega_i/x)$

↳ density function.

↳ it can be any F s.t. F(·) is monotonically function then F(gi(x)) is also monotonically incr. function.

→ $g_i(x) = P(\omega_i/x)$

take F = log function

∴ $F(g_i(x)) = \ln P(\omega_i/x)$

now, $P(\omega_i/x) = \frac{P(x/\omega_i) P(\omega_i)}{P(x)}$ ← constant

∴ $F(g_i(x)) = \ln [P(x/\omega_i) P(\omega_i)]$

$= \ln P(x/\omega_i) + \ln P(\omega_i)$ ← Discriminant function

↑ prob. dist. functions

Prob. density function → Parametric ← distribution curve known
→ Non-parametric ← distribution curve unknown.

Discrete PDF :-

① Bernoulli distribution

→ Denoted by $x \sim Ber(p)$ ← known parameter

→ There is one element which can have only 2 outputs

→ Prob. of success = p
Prob. of failure = $1-p$

② Binomial distribution → n independent elements or n identical Bernoulli trials

$$P(X = x) = {}^{n}C_x \, p^x \, q^{n-x}$$

③ Poisson distribution

→ $P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!}$ → when $n \to \infty$, p is small, np = constant (λ)

④ Geometric distribution

↳ keep trying until you get success.

$$P(X = x) = pq^{x-1}$$

e.g. if x = toss of coin & head is success
T → TT → TTT → TTTH.

∴ Success occurs only one time in all the trials.

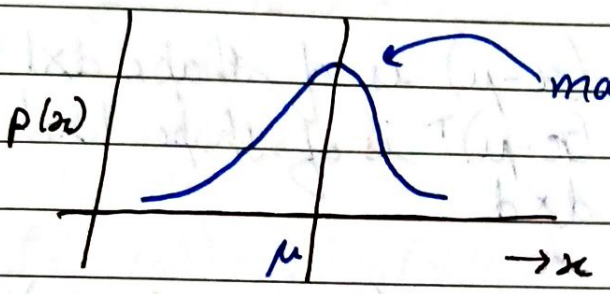
Normal distribution / Gaussian distribution

→ Represented as $X \sim N(\mu, \sigma^2)$

(μ: mean, σ^2: variance)

→ PDF, $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

→ Curve:



max value occurs at mean

→ $\mu = \int_{-\infty}^{\infty} x\, p(x)\, dx$, $\sigma^2 = E[(x-\mu)^2]$

$= \int_{-\infty}^{\infty} (x-\mu)^2 p(x)\, dx$

Multivariate Normal Density

⇒ univariate ⇒ x i.e. there is only one feature

→ But we cannot take only single feature to decide which feature belong to which class.

→ Instead we have feature vector.

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \times e^{\left[-\frac{1}{2}\{(x-\mu)^T \Sigma^{-1} (x-\mu)\right]}$$

where, μ = expected value = $E[x] = \int_{-\infty}^{\infty} x\, p(x)\, dx$

x is d-dimensional feature vector

Σ is covariance matrix

$|\Sigma|$ is determinant of covariance matrix

$\Rightarrow \Sigma = E[(x-\mu)(x-\mu)^T]$

where, $(x-\mu)$ is of shape $d \times 1$

$(x-\mu)^T$ is of shape $1 \times d$

$\therefore \Sigma$ is $d \times d$

$\Rightarrow$ Inner product = scalar = $(x-\mu)^T(x-\mu)$

$\therefore \Sigma = \int_{-\infty}^{\infty} (x-\mu)(x-\mu)^T p(x)\, dx$

$\sigma_{ij} = E[(x_i-\mu_i)(x_j-\mu_j)]$ → covariance

$\sigma_i = E[(x-\mu_i)^2]$ → variance

$\therefore$ Diagonals represent variance and rest are covariance

e.g. Bivariate, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

Lets assume x_1 and x_2 are statistically independent

$\therefore \sigma_1^2$ & σ_2^2 only exist while $\sigma_{12} = \sigma_{21} = 0$

→ $\therefore \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$ & $p(x) = \frac{1}{(2\pi)|\Sigma|^{1/2}} e^{[\quad]}$

$\left(\frac{-1}{2}\left(\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right)\right)$

→ ~~$p(x/\omega_i) =$~~

=

Final pdf

→ $p(x/\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{\left[-\frac{1}{2}\{(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\}\right]}$

$g_i(x) = p(x/\omega_i)$

$\therefore f(g_i(x)) = \ln p(x/\omega_i)$

Final discriminant function $= \frac{-1}{2}\left[(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right] - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i| + \ln p(\omega_i)$

K-NN Classifier

K → value
NN → nearest neighbour

→ This is the oldest method.

→ It is non-parametric i.e. doesn't assume any distribution.

→ Also called lazy learner bcz. it is slow & prepares everything only when new data comes.

e.g.

S.No	Maths	English	Result
1	4	3	Fail
2	6	7	Pass
3	7	8	Pass
4	5	5	Fail
5	8	8	Pass
6	6	8	?

Assume $k = 3$, 6th is the testing data.

Ans

1	4	3	$\sqrt{(4-6)^2+(3-8)^2} = \sqrt{29}$	F
2	6	7	$\sqrt{(6-6)^2+(7-8)^2} = \sqrt{1}$	P
3	7	8	$\sqrt{(7-6)^2+(8-8)^2} = \sqrt{1}$	P
4	5	5	$\sqrt{(5-6)^2+(8-5)^2} = \sqrt{10}$	F
5	8	8	$\sqrt{(8-6)^2+(8-8)^2} = \sqrt{2}$	P
6	6	8		

∴ Result is Pass since it is 3rd closest

Q

Height	Weight	Class	Distance	Distance
167	51	UW	10.8	6.7
182	62	N	8.87	13
176	69	N	9.674	13.41
173	64	N	4.35	7.61
172	65	N	8.52	8.24
174	56	UW	6.66	9.05
169	58	N	4.74	1.41
173	57	N	2.7	7
170	55	N	5.8	2
Test: 170	57	?	← test data	4.85

Ans mean H = 173.44, mean W = 58.67

For k = 3, Nearest values are : N, N, UW

∴ Using k-NN, the class of testing data is N.

Hamming code example

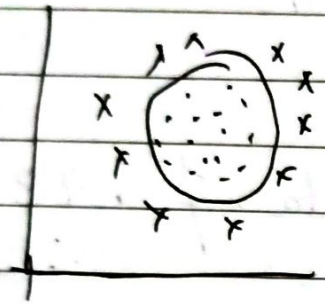
	Pepper	Ginger	Chilly	Likes	Hamming distance
A	T	T	T	F	1+0+0 = 1 ← 2nd
B	T	F	F	T	1+1+1 = 3
C	F	T	T	F	0+0+0 = 0 ← closest
D	F	T	F	T	0+0+1 = 1 ← 2nd
E	T	F	F	T	1+1+1 = 3
Test →	F	T	T	?	

k = 3

if k = 3, nearest are A B C and on average, answer is False

Artificial Neural Network

← Linear boundary can be solved using mathematics

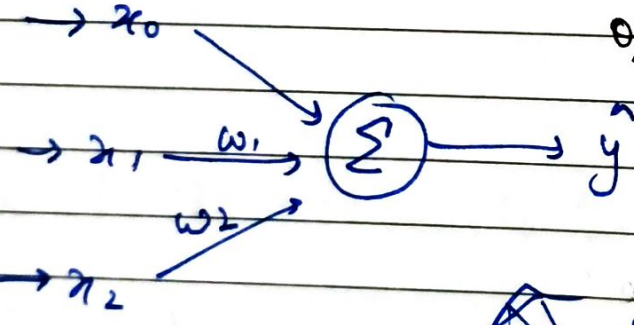


← Non-linear boundary which is very difficult mathematics. ∴ DL is used.

In

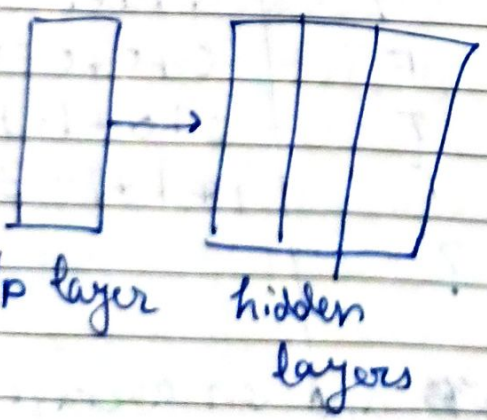
⇒ Deep Learning, the model itself performs feature extraction by applying algo unknown to us

⇒



Neural Network (single layer)

⇒



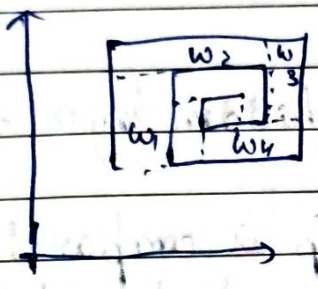
machine + part of human brain

Artificial Neural Network

⇒ Why ?

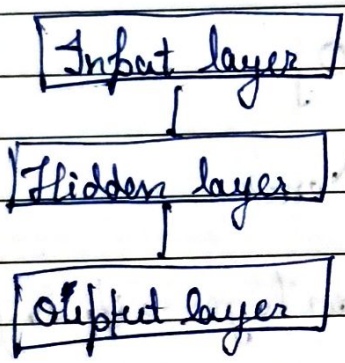
↳ if data is too complex, it is difficult to extract the features.

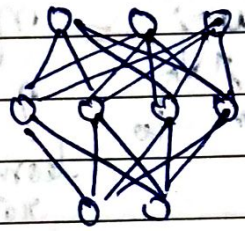
e.g.



we cannot easily determine the decision boundary.

⇒

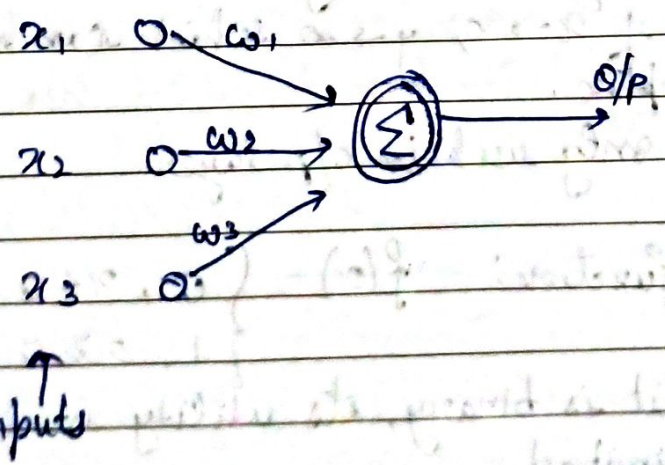




We have linear natured lines which are interconnected

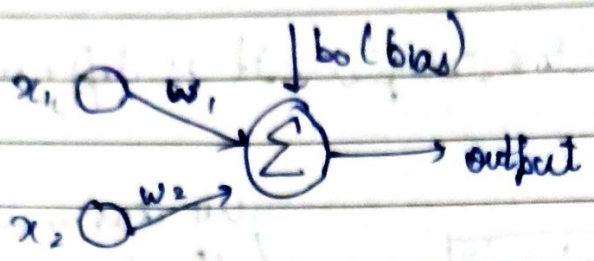
1NN → 0 hidden layer

2NN → 1 hidden layer



$\Sigma = x_1 w_1 + x_2 w_2 + x_3 w_3$

→ We want to convert linear output to non-linear output with the help of activation function.



Forward pass : i/p layer → hidden layer → o/p layer

Backward pass/ propagation : Actual o/p is compared with predicted output & we go backward to update the weights

$$\Delta w_{ij} = \eta \delta_j o_j$$

i → from, j → to, η → learning rate, δ → error

∴ o/p layer → hidden layer → i/p layer

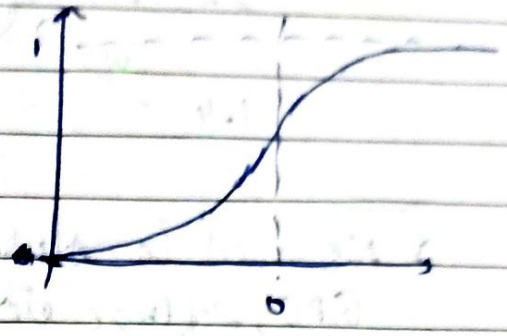
Activation function :-

① Linear activation function: $y = mx + c$
but if $x \to \infty$, $y \to \infty$ which can make the system complex.
∴ it is only used in o/p layer

② Binary step function: $f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$
→ Since it is binary, its utility is also limited

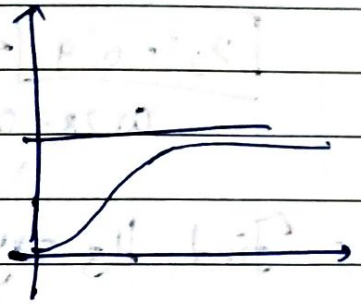
③ Sigmoid function : $A = \frac{1}{1+e^{-x}}$

→ Non-linear activation function
→ as $x \to \infty$, slope becomes constant
∴ we get optimized solution



④ tanh function : It is updated & more efficient than sigmoid

$$F(x) = \tanh(x) = \left(\frac{2}{1+e^{2x}} - 1\right)$$



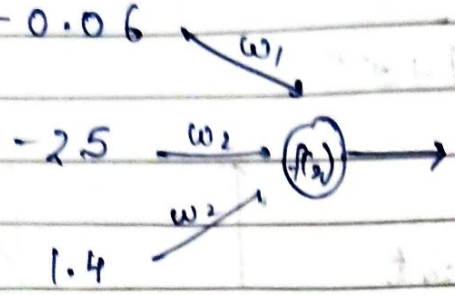
⑤ ReLU (Rectified Linear Unit) : most popular

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Feedforward NN : unidirectional
Recurrent NN : they have directed cycles

→ Perceptron = Neural Network.

→ If data has variation then learning rate is usually kept low so that it fits.

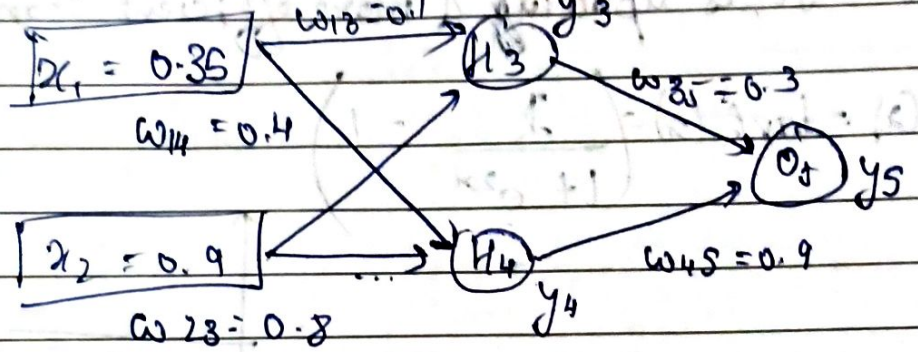
e.g. 

output $= f(y) = \frac{1}{1+e^{-y}}$

where $y = -0.06\,w_1 + (-2.5)w_2 + 1.4w_3$

→ No need to study how training is performed in Neural Network, GPU, softmax etc.

Backprop e.g.



Find $H_3 \to y_3$, $H_4 \to y_4$, $O_5 \to y_5$

$$w_{ji}\,{}_{new} = w_{ji}\,{}_{old} + \Delta w_{ji}$$

where $\Delta w_{ji} = \alpha \delta_j O_j$

target output

$\Rightarrow \delta_j$ (o/p unit) $= y_j(1-y_j)(t_j - y_j)$

e.g. $\delta_5 = y_5(1-y_5)(t_5 - y_5)$

$\Rightarrow \delta_j$ (hidden layer) $= y_j(1-y_j)\sum w_{kj}\delta_j$

e.g. $\delta_3 = y_3(1-y_3)(w_{35}\delta_5)$ ← if O_6 as well then add $w_{36}\delta_6$ as well