Computational Intelligence & Machine Learning.

Basics :-

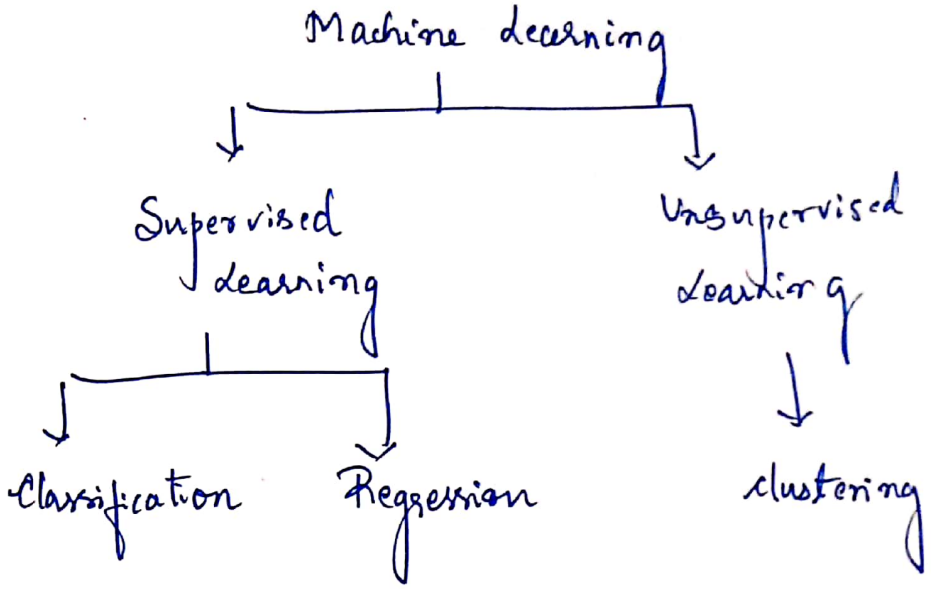
→ The ability of machine to categorize correctly new Examples that differ from those used for training is known as Generalization.

→ Machine learning is done in 2 phases
- Training phase
- Testing phase

* In training phase we train the machine with N variants of a random continuously varying data say 'X'

* In testing phase a new varient of the continously varying quantity x is given to test the prediction of the Machine learning Algorithm.

→ There can be 2 ways of machine learning
- Supervised
- un-Supervised.



Linear Basis function Models :- [Linear Regression]

→ Regression analysis is a form of predictive modeling technique which gives relationship between dependent (target) & Independent variables (predictor)

→ Linear Regression is a statistical Model which gives linear relationship between two or more variables.

Note :- Regression algorithms are used to predict the continuos values such as price, salary, age etc.

→ Simple Linear Model of regression is given as

$$y(x, w) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

where $x = (x_1, \ldots, x_D)^T$

$x \rightarrow$ Independent variable

$y \rightarrow$ dependent variable.

→ To Extend the class of models to Include non-linearity of i/p variables Equation is modified as

$$y(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \underbrace{\phi_j(x)}_{\text{Basis function}}$$

↓ Total M parameters

* w_0 → Bias parameter

$$\cdot y(x, w) = \sum_{j=0}^{M-1} w_j \phi_j(x) = w^T \phi(x)$$

where $w = (w_0 \ldots, w_{M-1})^T$ & $\phi = (\phi_0, \ldots \phi_{M-1})^T$

Note :- Here original variables are vector x, features of x is denoted as $\{\phi_j(x)\}$

→ By using non-linear Basis fn we allow $y(x, w)$ to non-linear fn of i/p vector x.

→ Basis function → polynomial
→ Gaussian
→ Sigmoidal

* polynomial $\phi_j(x) = x^j$

* Gaussian $\phi_j(x) = \exp\left\{-\frac{(x-\mu)^2}{2s^2}\right\}$

* Sigmoidal $\phi_j(x) = \sigma\left(\frac{x - \mu}{s}\right)$

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Note :- Other basis for is Fourier Basis, wavelets.
We can also assume $\phi(x)$ to simply equal to x.
$\phi(x) = x$.

Maximum likelihood & Least Squares :-

* Basics to be known
  - Polynomial curve fitting
  - Gaussian function
  - Bayes theorem.

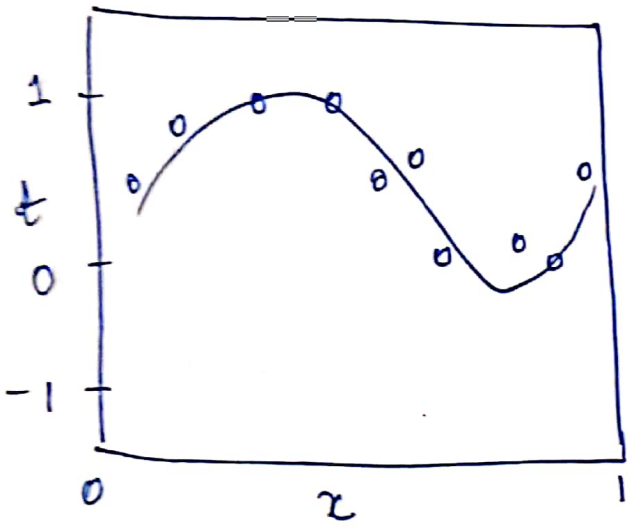
→ Applications in which training data comprises Examples of input vectors along with their corresponding target vectors are known as Supervised learning problems. If the desired output consists of one or more continuous variables, then task is called regression.

→ If training data consists of a set of input vector x without any corresponding target values. This is unsupervised learning where we need to discover groups of similar examples in the data. called as clustering.

→ The technique of reinforcement learning is concerned with the problem of suitable actions to taken in a given situation in order to maximize a reward. Here learning is on a trail & Error process.

Ex:- polynomial curve fitting

→ given a training set with N observations of x $x = (x_1, \ldots, x_N)^T$ with $t = (t_1, \ldots, t_N)^T$



* x is uniformly spaced in range [0, 1]

* t is calculated by taking an Example function $\sin 2\pi x$ with small level of random noise. (g/p)

→ Our goal is to Exploit this training set in order to make predictions of $\hat{t}$ of the target for some new value $\hat{x}$. of the Input variable. & discover the underlying function $\sin(2\pi x)$ through Machine learning Algorithm.

→ Polynomial function is given as

$$y(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$

$$= \sum_{j=0}^{M} w_j x^j$$

* M → order of polynomial
* polynomial co-efficients $w_0, \ldots, w_M$ are denoted as vector w.
* polynomial function $y(x, w)$ is a non-linear function of x, and linear function of w.

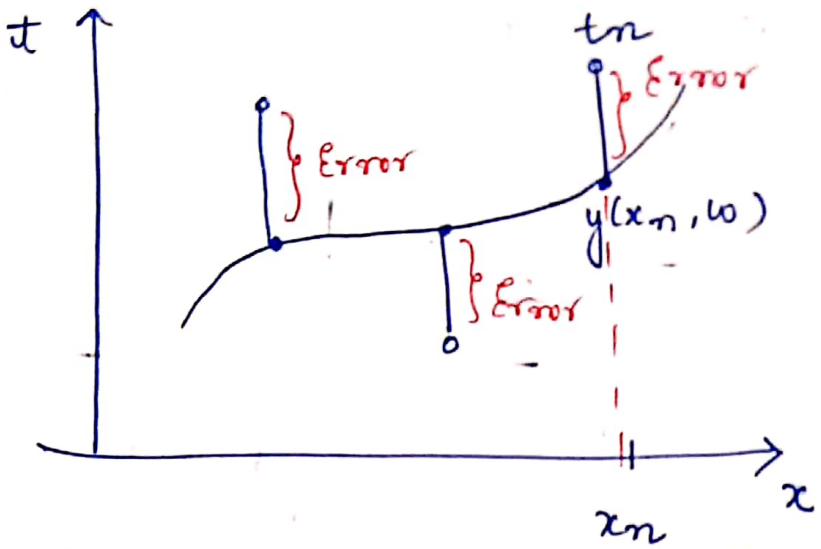
→ The values of co-efficients will be determined by fitting the polynomial to the training data

→ In fitting the polynomial curve to the training data we occur with Error or misfit between $y(x, w)$ for any given w & x.

→ To minimize error widely used Error function for is Square of errors between $y(x_n, w)$ & data point x_n & corresponding target t_n.

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2$$

→ Fig below shows the geometrical interpretation of sum of square Errors.

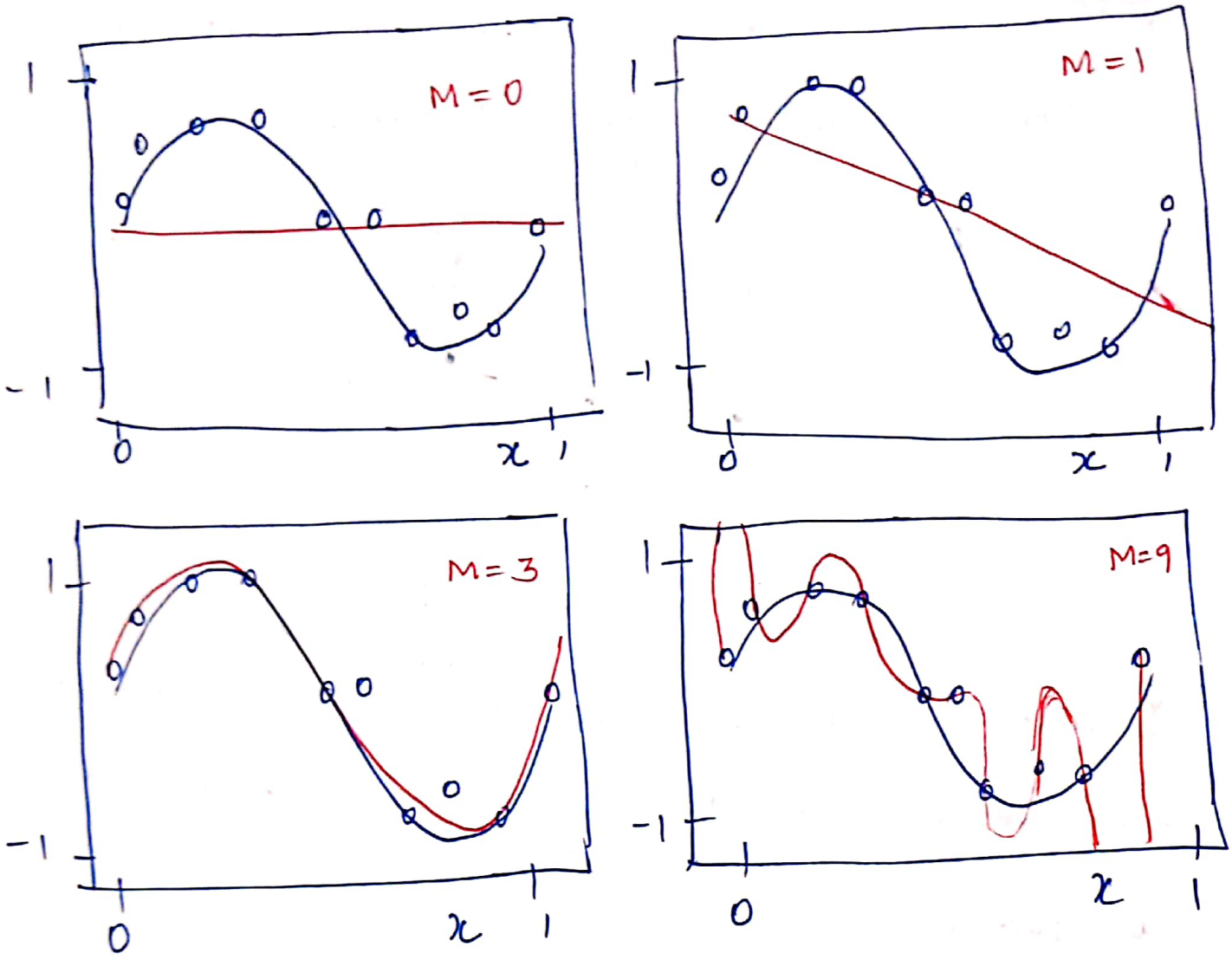


* Here we can solve the curve fitting problem by choosing the value of w $E(w)$ is as small as possible.

* Resulting polynomial with changed w is given as $y(x, w^*)$

→ Other than reducing $E(w)$, there remains the problem of choosing order of M of the polynomial.

→ .



→ we notice that with M = 0 & M = 1 polynomials we get poor fits of data [we need a sine(sin2πx) function fit]

Note :- Blue colour curve → Actual fit needed
Red colour curve → fit got with the order of polynomial.

→ with M = 3 we get Best fit to the function Sin (2πx)

→ with M = 9 we get Excellent fit to the training data.

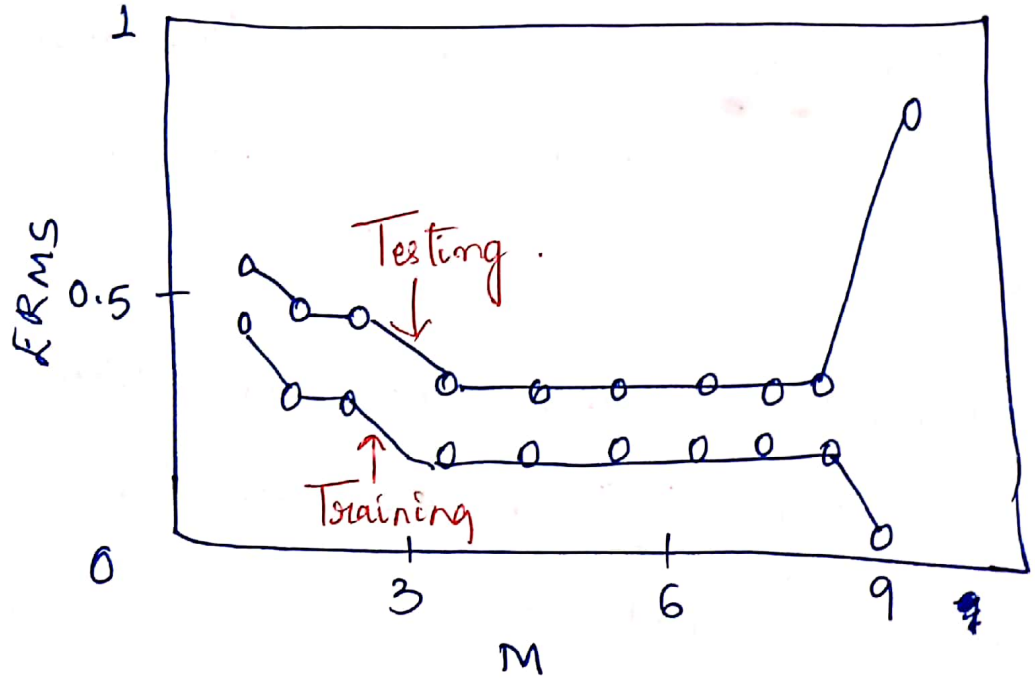
→ In fact polynomial passes Exactly through Each data point & $E(w^*) = 0$; But curve Oscillates Wildely & gives poor representation of the function $\sin(2\pi x)$. This is called over-fitting.

→ It is sometimes more convenient to use root-mean-square of Error function

$$E_{RMS} = \sqrt{2E(w^*)/N}$$

↓

different sizes of data sets.



* We note in above fig that for $3 \leq M \leq 8$ we get small values of E_{RMS} (Error).

* For $M = 9$, training set error goes to zero

but test-set error becomes very large, ∴ giving wild oscillations in $y(x, w^*)$ & the predictions.

→ The over-fitting problem becomes less severe as the size of data-set increases.

→ Over-fitting problem is a general property of maximum likelihood, By adapting Bayesian approch, overfitting problem can be avoided.

note :- In Bayesian Model the effectare number of parameters adapts automatically to the size of the data set.

→ One more approch to control the over-fitting phenomenon is regularization. Regularization involues adding a penalty term to the Error function.

$$\tilde{E}(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2}\|w\|^2$$

where $\|w\|^2 = w^T w = w_0^2 + w_1^2 + \dots + w_M^2$ & the co-efficient λ governs the relative importance of the regularization term.

Probability theory :-

→ Probability theory provides a consistent framework for the quantification & manipulation of uncertainity; When combined with decision theory. will allow us to make optimal predictions given all the Information available to us, even though that information may be incomplete or ambiguos.

Rules of Probability :-

Sum rule :- $P(x) = \sum_{y} P(x, y)$

Product rule :- $P(x, y) = P(y|x)\, P(x)$

note :- $P(x, y)$ → joint probability {Probability of x & y}

$P(y|x)$ → Probability of y given x.

$P(x)$ → Marginal probability / probability of x

→ From product rule, together with the symmetry property $P(x, y) = P(y, x)$

we obtain the following relationship between Conditional probabilities.

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$$

Baye's theorem

* Baye's theorem plays a central role in pattern recognition & machine learning.

* Denominator of Baye's theorem can be written as

$$P(X) = \sum_{Y} P(X, Y) \rightarrow \text{sum rule}$$

or

$$P(X) = \sum_{Y} P(X|Y)\,P(Y)$$

$$\left(\because P(X, Y) = P(Y|X)\,P(X)\right)$$

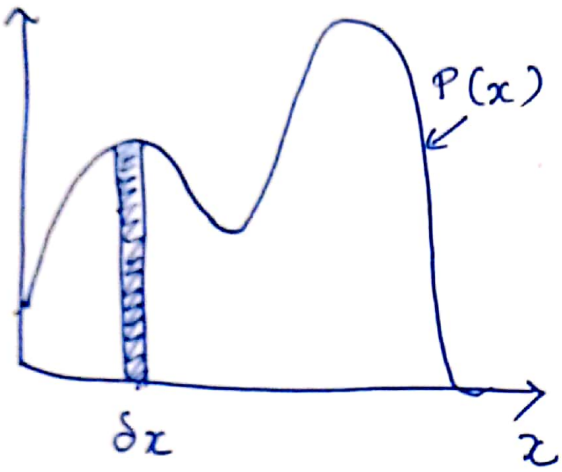
$$\therefore P(X) = \sum_{Y} P(X|Y)\,P(Y)$$

↓ normalization constant-

Probability densities :-

→ we wish to consider probabilities with continu- -os variables rather than just discrete set of Events.

→ If we have to take probability of continuously varying real-valued variable x falling in the interval $(x, x+\delta x)$ is given as $P(x)\,\delta(x)$ for $\delta x \to 0$, then $P(x)$ is called Probability density function over x.



→ The probability that x will lie in an interval (a, b) is given by

$$P(x \in (a,b)) = \int_a^b P(x)\,dx$$

→ Probabilities are nonnegative

$$\therefore P(x) \geq 0$$

$$\int_{-\infty}^{\infty} P(x)\,dx = 1$$

→ $P(x)$ must satisfy these two conditions.

→ Most important operations involves probabilities is that of finding weighted averages of functions

→ The average value of some function $f(x)$ under a probability distribution $p(x)$ is called Expectation of $f(x)$. & is denoted by $\mathbb{E}[f]$.

Discrete Distribution

$$E[f] = \sum_x P(x) f(x)$$

Continuous variables

$$E[f] = \int P(x) f(x) \, dx$$

→ Applying / Re-writing concept of Baye's theorem with respect to linear Basis function model

likelihood

$$y(w|t) = \frac{P(t|w) \; P(w)}{P(t)}$$

↓ (posterior Probability)

Prior Probability Info.

* w → weights associated with input

t → target achieved when right weights are set

→ Baye's theorem is used to convert a prior probability into a posterior probability by incorporating the Evidence provided by the observed data.

$$\text{Posterior} \propto \text{likelihood} \times \text{prior}$$

note :- all of the quantities in Baye's probability theorem are viewed as function of w.

* The denominator is normalization Constant which

Ensures that posterior distribution on d·H·S is a valid probability density & Integrates to one.

→ In Baye's theorem, the likelihood function plays a central role.

→ A widely used Estimator of w is maximum likelihood in which w is set to a value that maximizes likelihood function ~~P(t|w)~~ P(t|w).

The Gaussian Distribution :-

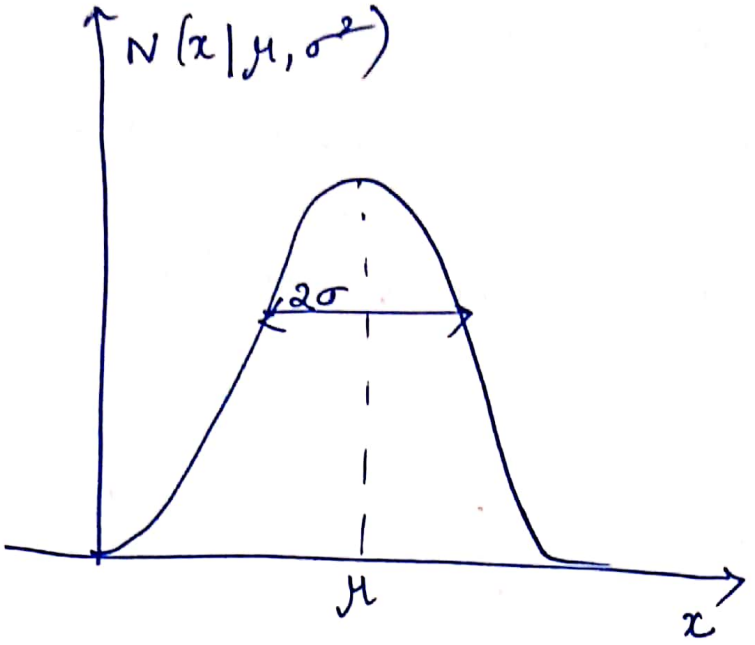
→ Let us assume that the random variable which is continuously varying & is used to train the Machine has a gaussian distribution Instead of polynomial distribution, discussed Earlier

→ For a real-valued variable x, Gaussian distribution is defined as.

$$N(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

↓

Gaussian is also called normal distribution.



$\mu \rightarrow$ mean

$\sigma^2 \rightarrow$ variance

$\sqrt{\sigma^2}$ / $\sqrt{\text{variance}} = \sigma =$ Std deviation

$\frac{1}{\text{Variance}} = \frac{1}{\sigma^2} = \beta$ [precision]

* Above Gaussian distribution satisfies the below Condition

$$N(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} N(x|\mu,\sigma^2)\,dx = 1$$

* Expectation of x under this gaussian distribution is given as

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

μ → average value of x

→ We can obtain or Maximize likelihood function by writing the probability of data set, given μ, σ^2

Gaussian distribution is given as

$N(x|\mu, \sigma^2)$ → taking probability of this distribution wrt μ, σ^2

wrt μ

$$P(x|\mu, \sigma^2) = \prod_{N=1}^{N} N(x_n|\mu, \sigma^2)$$

* It is more convenient to maximize log of likelihood function.

∴ Above Equation becomes

$$\ln p(x|\mu, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi)$$

∴ likelihood fn wrt μ is maximizing the above Expression. we get

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

μ_{ML} → max likelihood

wrt β

$$p(t|x, w, \beta) = N(t|y(x, w), \beta^{-1})$$

$$p(t|x, w, \beta) = \prod_{n=1}^{N} N(t_n|y(x_n, w), \beta^{-1})$$

$$\beta = \frac{1}{\sigma^2} \quad \therefore \beta^{-1} = \sigma^2$$

111y taking log for above Expression we get

$$\ln p(t|x, w, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, w_{ML}) - t_n\}^2$$

→ Now returning back to maximum likelihood & least square concept.

→ Let us consider target t with Gaussian noise model

$$\therefore t = y(x, w) + \epsilon \quad \leftarrow \text{Gaussian noise}$$

$$\downarrow$$

$$y(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x)$$

[linear Basis for model]

→ Taking probability of the above Expression with precision parameter from Gaussian distribution

$$p(t|x, w, \beta) = N(t|y(x, w), \beta^{-1})$$

(normal distribution · inverse variance · precision)

$$\left[\frac{1}{\sigma^2} = \beta \Big/ \frac{1}{\beta^{-1}} = \sigma^2 \right]$$

$\beta = \sigma^2$ (variance)

* Here Expectation of new x to a target t is given as

$$E[t|x] = \int t \cdot p(t|x)\, dt = y(x, w)$$

→ Now consider a data set of inputs

$X = \{x_1, \ldots, x_N\}$ with $t_1 \ldots t_N$

→ We can obtain the Expression for the likelihood function as a function of Adjustable parameter w, β

$$\therefore p(t|x, w, \beta) = \prod_{n=1}^{N} N(t_n | w^T \phi(x_n), \beta^{-1})$$

$$y(x, w) = w^T \phi(x_n)$$

→ To maximize this likelihood function

$$p(w|t) = \frac{p(t|w) \cdot p(w)}{p(t)}$$

(likelihood fn) → Bayes theorem

& keeping notation of LHS dropping x we get :

$$p(t|w, \beta) = \prod_{n=1}^{N} N(t_n | w^T \phi(x_n), \beta^{-1})$$

→ Taking log on both sides to maximize the above function $p(t|w, \beta)$ [refer gaussian distribution] we get

$$\ln p(t|w, \beta) = \sum_{n=1}^{N} \ln N\{t_n | w^T \phi(x_n), \beta^{-1}\}$$

$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(w) \rightarrow (1)$$

likelihood ↓ Error to likelihood

where where $E_D \rightarrow$ Sum of Square Error. for n

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - w^T \phi(x_n)\}^2$$

→ We can maximize the likelihood function under a conditional Gaussian noise distribution by minimizing a sum-of squares Error given by $E_D(w)$.

→ ~~Maxmize~~ Minimize by taking gradient of $E_D(w)$

$$\nabla \ln p(t|w, \beta) = \beta \frac{d}{dw} \left[\frac{1}{2} \sum_{n=1}^{N} \{t_n - w^T \phi(x_n)\}^2 \right]$$

(const) (const)

* Simply differentiate by w

βE_D.

$$0 = \frac{d}{dw} \left[\frac{1}{2} \sum_{n=1}^{N} \{t_n - w^T \phi(x_n)\}^2 \right]$$

Assume this x [const is = 0]

note :- above Expression is w.r.t to

$$\frac{d}{dx} x^2 = 2x \frac{d}{dx} \cdot x$$

$$\therefore \quad 0 = \frac{d}{dw} \sum_{n=1}^{N} \frac{1}{2} 2 (t_n - w^T \phi(x_n)) \cdot \frac{d}{dw} [t_n - w^T \phi(x_n)]$$

$$\therefore 0 = \sum_{n=1}^{N} \alpha \cdot \frac{1}{\alpha} (t_n - w^T \phi(x_n) \cdot [0 - \phi^T(x_n)]$$

$$0 = \sum_{n=1}^{N} (t_n - w^T \phi(x_n) \cdot \phi^T(x_n)$$

$$0 = \sum_{n=1}^{N} t_n \phi(x_n)^T - \sum_{n=1}^{N} w^T \phi(x_n) \cdot \phi^T(x_n)$$

* For making the Equation in generalized form we will write the above Equation as

$$\rightarrow \quad 0 = t \cdot \phi(x_n)^T - w^T \phi(x_n) \phi(x_n)^T$$

$$\rightarrow \quad t \cdot \phi(x_n)^T = w^T \phi(x_n) \phi(x_n)^T$$

Or

$$\rightarrow \quad w \phi(x_n)^T \phi(x_n) = t \cdot \phi(x_n)^T$$

$$\rightarrow \quad \therefore w \phi(x_n)^T \phi(x_n) \cdot \left[\phi^T(x_n) \cdot \phi(x_n)\right]^{-1} = \left[\phi^T(x_n) \cdot \phi(x_n)\right]^{-1} \cdot t \cdot \phi(x_n)^T$$

$$\left[\because A \cdot A^{-1} = 1 \;/\; \phi^T(x) \cdot \phi(x)^{-1} = 1 \right]$$

$$\therefore w \underbrace{\phi(x_n)^T \phi(x_n) \cdot \left[\phi^T(x_n) \cdot \phi(x_n)\right]^{-1}}_{w_{ML}} = \left[\phi^T(x_n) \cdot \phi(x_n)\right]^{-1} \cdot t \cdot \phi(x_n)^T$$

$$\boxed{\therefore \quad w_{ML} = \left[\phi^T \phi\right]^{-1} \cdot \phi^T \cdot t}$$

→ The quantity $w_{ML} = (\phi^T\phi)^{-1}\phi^T t$

↓

normal Equations for least Squares problem

ϕ → N x M matrix called Design matrix

$$\phi = \begin{bmatrix} \phi_0(x_1) & \phi(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_{M-1}(x_2) \\ \vdots & & & \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_{M-1}(x_N) \end{bmatrix}$$

The quantity $\phi = (\phi^T \cdot \phi)^{-1}\phi^T$ is known as Moore – Penrose pseudo Inverse of matrix ϕ. [ϕ is a N x M matrix]

note :- $w_{ML} = [\phi^T\phi]^{-1} \cdot \phi^T \cdot t$

$(M \times N)(N \times M)$ → $M \times M$

$N \times M$ $N \times 1$

$w_{ML} = (M \times 1)$ → Row vector as a

→ Now we can maximize the log likelihood function

$$\ln p(t|w,\beta) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(w)$$

wrt to precision parameter β.

$$\frac{d}{d\beta}\left[\ln p(t|w,\beta)\right] = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(w)$$

$$= \frac{N}{2}\cdot\frac{1}{\beta} - 0 - E_D(w)$$

Equating above Equation to zero in order to minimize it

$$\frac{N}{2\beta} - E_D(w) = 0$$

$$\frac{N}{2\beta} = E_D(w)$$

$$\frac{1}{\beta} = \frac{2}{N}\cdot E_D(w)$$

Subst $E_D(w)$

$$\therefore \frac{1}{\beta_{ML}} = \frac{2}{N}\cdot\frac{1}{2}\sum_{n=1}^{N}\left(t_n - w^T\phi(x)\right)^2$$

$$\therefore \frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - w^T\phi(x)\right)^2$$

∴ we can conclude that to maximize the likelihood function by 2 parameters w, β.

wrt w :-

$$w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t$$

wrt β :-

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{t_n - w_{ML}^T \phi(x_n)\}^2$$

∴ We can conclude that to maximize the likelihood function by 2 parameters w, β.

wrt w :-

$$w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t$$

wrt β :-

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{t_n - w_{ML}^T \phi(x_n)\}^2$$

Sequential learning :-

→ When we have Input data set sufficiently large & we use the complete set or Entire Batch for training the machine [Batch processing] the computationally costly

→ ∴ we can use sequential algorithms as an alternative. which is also known as on-line algorithms.

→ Here data sets of the Input are considered one at a time & parameters are updated Each time.

→ We can obtain a Sequential learning algorithm by applying the technique of stochastic gradient descent / Sequential gradient descent.

→ Error function is given for all data points as Summation function.

$$E = \sum_n E_n$$

→ The stochastic gradient descent algorithm updates the parameter w using

$$w^{(\tau+1)} = w^{\tau} - \eta \nabla E_n$$

τ → Iteration number

η → Learning rate

→ we Initialize value of 'w' to some starting vector $w^{(0)}$

→ For the case of Sum of square of Error function $\left[E(w) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - w^T \phi(x_n)\}^2 \right.$

$$w^{(\tau+1)} = w^{\tau} + \eta (t_n - w^{(\tau)T} \phi_n) \phi_n$$

where $\phi_n = \phi(x_n)$. This is known as least-mean-squares or LMS algorithm.

Regularized least Squares :-

→ earlier concept of adding a regularization term to an Error function in order to control over-fitting to control total Error function & minimize it.

$$E_D(w) + \lambda E_W(w)$$

↓ regularization term

$E_D(w)$ → data dependent Error

$E_W(w)$ → regularization term

→ simple regularizer is given by sum-of-squares of weight vector Elements.

$$E_W(w) = \frac{1}{2} w^T w$$

also $$E(w) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - w^T \phi(x_n)\}^2$$

[Subst done for $E_D(w) + \lambda E_W(w)$]

∴ Total Error fn is

$$\frac{1}{2} \sum_{n=1}^{N} \{t_n - w^T \phi(x_n)\}^2 + \frac{\lambda}{2} w^T w$$

$$E(w) + \lambda E_W(w) \rightarrow (1)$$

→ This choice of regularizer is known in machine learning as weight decay as in Sequential learning algorithms, it Encourages weight-vectors to decay towards zero, unless supported by the data.

→ setting gradient of Eqn (1) wrt to 'w' to zero & solving for 'w' we get

Eqn (1) is

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - w^T\phi(x_n)\}^2 + \frac{\lambda}{2}w^T.w$$

$$\therefore \frac{1}{2}\sum_{n=1}^{N}\{t_n - w^T\phi(x_n)\}^2 + \frac{\lambda}{2}w^2$$

differentiating above Expression wrt 'w'

$$= \frac{1}{2}\cdot 2\cdot\{t_n - w^T\phi\}\phi^T + \frac{\lambda}{2}\cdot 2w$$

$$= t\phi^T - w^T\phi\phi^T + \lambda.w$$

$$= t\phi^T - w^T(\phi\phi^T - (-)\lambda I)$$

$$0 = t\phi^T - w^T(\phi\phi^T + \lambda I)$$

$$\therefore w(\phi^T\phi + \lambda I) = t\phi^T$$

or $$\boxed{w = (\lambda I + \phi^T \phi)^{-1} \phi^T \cdot t}$$

↑ regularization factor

Simple Extension of least Square with regularization factor.

$$w_{ML} = (\phi^T \phi)^{-1} \phi^T \cdot t$$

Multiple outputs :-

→ If we wish to predict more target variables 't' denoted collectively by target vector 't', then

$$y(x, w) = w^T \phi(x)$$

↓

K → dimensional column vector

$w \rightarrow M \times K$ matrix

$\phi(x) \rightarrow M$ dimensional column vector

* Gaussian distribution

$$p(t|x, w, \beta) = N(t | w^T \phi(x), \beta^{-1} I)$$

$$\therefore \ln p(T|X, w, \beta) = \sum_{n=1}^{N} \ln N(t_n | w^T \phi(x_n), \beta^{-1} I)$$

$\{t_1 \cdots t_n\}$ ↑ ↓ $\{x_1 \cdots x_n\}$

$$= \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2} \sum_{n=1}^{N} \| t_n - w^T \phi(x_n) \|^2$$

↑ III term

[III° to $\frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(w)$ derived Earlier, above Expression can also be written as $\frac{N}{2} \ln\left(\frac{\beta}{2\pi}\right) - \beta E_D(w)$ [$\log A - \log B = \log A/B$]]

∴ maximizing above fn wrt w

$$\boxed{w_{ML} = (\phi^T \phi)^{-1} \phi^T \cdot T}$$

T → Target vector

↑ III° to $w_{ML} = (\phi^T \phi)^{-1} \phi^T \cdot t$ ↑ Single target

→ If we Examine result for target variable t_K

$$w_K = (\phi^T \phi)^{-1} \phi^T t_K = \phi\, t_K.$$

The Bias-variance Decomposition :-

→ Use of maximum likelihood or least squares can lead to Severe over-fitting if complex data sets are taken.

→ We can limit the no of basis function in order to avoid over-fitting but this will limit the flexibility of the model to capture important trends in data.

→ Here adding a regularization term (λ) can control over-fitting for models with many parameters.

→ Bayesian treatment of linear regression will avoid the over-fitting problem of maximum likelihood.

→ Note :-> Bias is used to simplify the assumptions made by the model to make the target function easier to approximate.

2) Variance is the amount that the Estimate of the target will change given different training data.

3) Bias - variance tradeoff is the property of a model that the variance of the parameter Estimated across samples by increasing the bias in the Estimated parameters.

⇒ Bias - variance decomposition or tradeoff is the conflict in trying to simultaneously minimize the two sources of Error that prevent supervised learning algorithms from generalizing beyond the training set.

Bias Error - erroneous assumptions in the learning algorithm.

Variance - Error from sensitivity to small fluctuations in the training set.

** Bias - Variance decomposition is a way of analyzing a learning algorithm's Expected generalization Error with respect to three terms bias, variance, irreducible Error. Resulting from noise itself.

→ Assuming $g(x)$ as generalization Error which is taken for predicting a new target t for a new input x

$$E\left[(t - g(x)\right]^2$$

→ Error in Estimation of t, can be written as

$$\mathcal{E}\left[(t - y(x)\right]^2 = \mathcal{E}\left[(t - g(x)\right]^2$$

$$= \mathcal{E}\left[(\underbrace{t - g(x)}_{a} + \underbrace{g(x) - y(x)}_{b})^2\right]$$

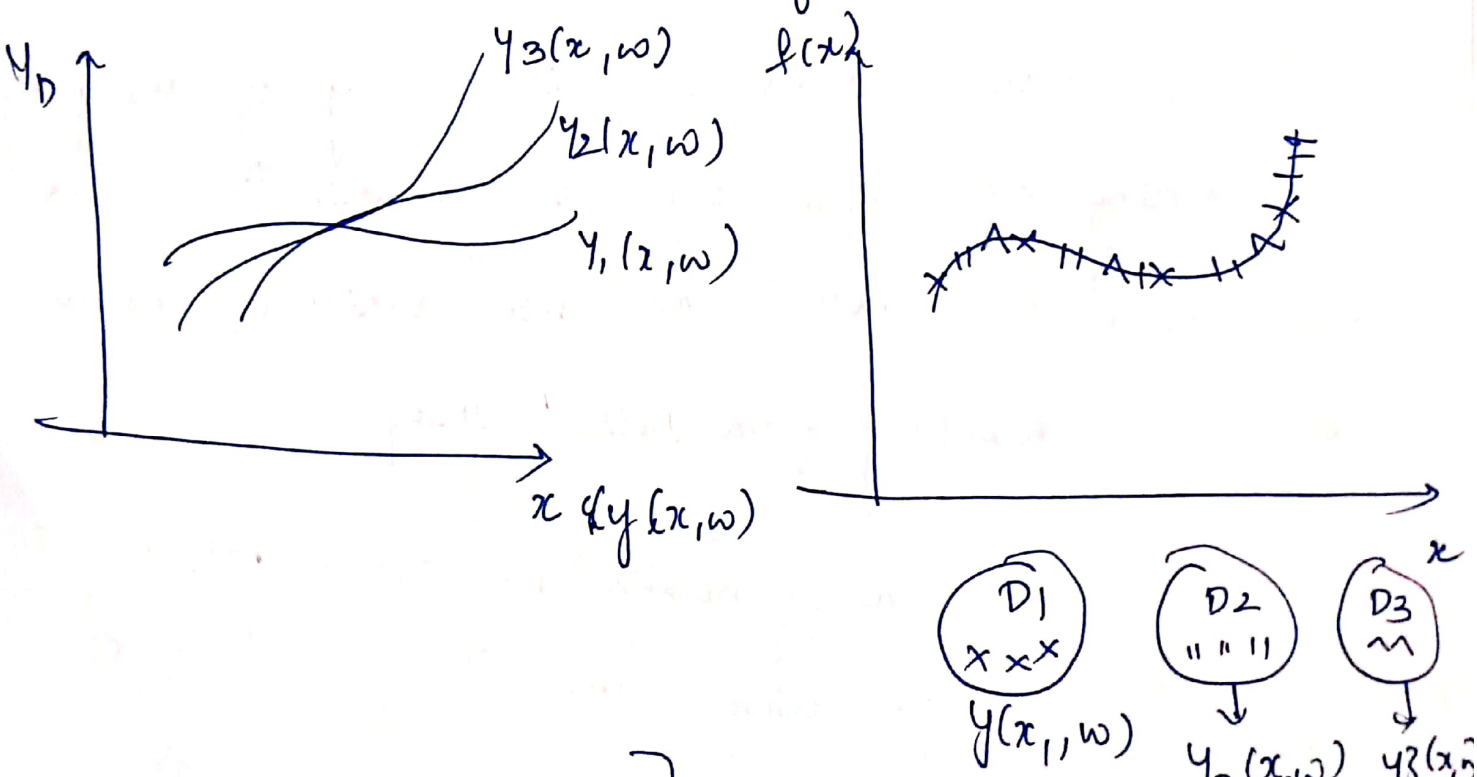
$$= \mathcal{E}\left[(t - g(x))^2\right] + \mathcal{E}\left[(g(x) - y(x))^2\right]$$

$$+ 2 \cdot \mathcal{E}\left[(t - g(x))\, g(x) - y(x)\right]$$

a^2 $\quad b^2$

$2 \quad a \quad b.$

→ Here $Y(x)$ is obtained from data set-



→ $\mathcal{E}\left[\left[(y(x) - g(x)\right]^2\right]$ →(1)

we can write the above Eqn (1) as

→ $\mathcal{E}_D\left[(g(x) - y(x,D))^2\right]$.

$$= E_D\left[\underbrace{(y(x,D) - E_D[y(x,D)]}_{p} + \underbrace{E_D[y(x,D)] - g(x)}_{q}\right]^2$$

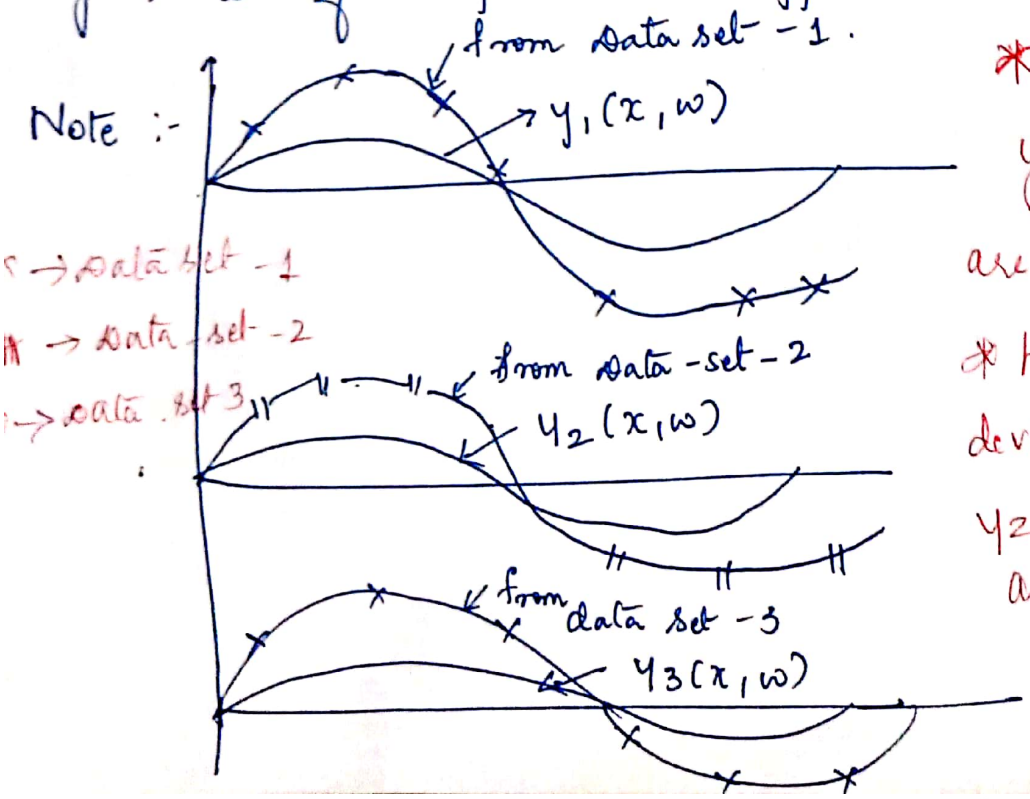
$$= E_D\left[p^2 + q^2 + 2pq\right]$$

$$= \underbrace{E_D\left[(y(x,D) - E_D[y(x,D)]\right]^2}_{\text{Variance}} + \underbrace{E_D\left[E_D[y(x,D) - g(x)]\right]^2}_{\text{Bias}^2} + 2\underbrace{E_D\left[(y(x,D) - E_D[y(x,D)]\left[E_D(y(x,D) - g(x))\right]\right]}_{\text{noise}}$$

$$E^{(\text{loss})} = \text{Variance} + \text{Bias}^2 + \text{Noise}$$

* $g(x)$ is from Expectation

$y(x)$ is from parametric approch.



Note :-

$\rightarrow$ data set - 1

$\rightarrow$ data set - 2

$\rightarrow$ data set 3

* $g(x)$ is reference

$y_1(x,w)$, $y_2(x,w)$ are different solutions

* how far $g(x)$ is deviated from $y_1(x,w)$

$y_2(x,w)$ is known as Bias.

* Estimation within $y_1(x,w)$, $y_2(x,w)$, $y_3(x,w)$ or deviation within $y_1(x,w)$, $y_2(x,w)$, $y_3(x,w)$ is called as Variance.