24/07/2023

Q. How do we learn?

Q. Give examples of machine learning.

27/07/2023

Q. What is the differce between Database Management and Big-Data Analytics?

Big-Data

- Structured (Student and faculty data in institute server)
- Unstructured (Traffic data) (Medical, CCTV data)

Share-Market data has velocity.

Medical and Share-Market are real-time data.

Trustworthy data is very reliable.

- Reducing volume of memory is a big challenge.

Q. What is ML?

=> Learning is any process by which a system improves performance from experience.

- <u>Steps in ML</u>
  (i) Data Gathering
  (ii) Data Pre-processing
  (iii) Feature Engineering
  (iv) Algorithm Selection & Training
  (v) Making Predictions

Q. Differentiate between ML and Data Science.

- 'Learning' is used when –
  (i) Human experience is low
  (ii) Minor changes are not discernible by humans (eg. speech Recognition)

- Every ML algorithm has 3 parts :
  (i) Representation
  (ii) Evaluation
  (iii) Optimization

- Inductive learning has 3 parts :
  (i) classification (Discrete)
  (ii) Regression (continuous)
  (iii) Probabilistic Estimation

In regression analysis,

$$y = ax + b$$

$a \rightarrow$ Regression Co-efficient

Eg. : crop yield = $w_0 + w_1$(amount of fertilizer) $+ w_2$ (amount of water)

- Assume a model :

$$Y = w_0 + w_1 X + s$$

where s is the error term.

$$y = \hat{w}_0 + \hat{w}_1 x$$

$\hat{w}_0$ and $\hat{w}_1$ are estimated.

Let, $\hat{y}_i = \hat{w}_0 + \hat{w}_1 x_i$ be the prediction of Y based on the i^{th} value of X.

$e_i = y_i - \hat{y}_i$ represent the i^{th} 'residual'

Residual Sum of Squares (RSS) is defined as -

$$\sum_{i=1}^{m} e_i^2 = e_1^2 + e_2^2 + \dots + e_m^2$$

Hastie
Tibshirani
Friedman
The Elements of Statistical Learning

31/07/2023

Loss Function:

$L_n()$, also known as cost function.

$$L_n(t_n, h(x_n; w_0, w_1)) = \{t_n - h(x_n; w_0, w_1)\}^2$$

Average loss,

$$L = \frac{1}{N} \sum L_n()$$

$$= \frac{1}{N} \sum_{i=1}^{n} \{t_n - h(x_n; w_0, w_1)\}^2$$

$$= \frac{1}{N} \sum_{i=1}^{n} \{t_n - (w_0 + w_1 x_i)\}^2$$

- Gradient Descent Algorithm

There are two types based on how the training data is processed.

1. Stochastic Gradient Descent
2. Batch Gradient Descent

z-score is the co-efficient divided by its standard error.

Q. What is the effect if we increase the degree of the regression polynomial?

=> If the curve oscillates too much, there is overfitting.

The root mean square error is $\sqrt{2E(W^*)/N}$.

- overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data due to detailing of large training data.

- If it performs well on test set → Good model
  If it performs well only on traing set → overfitting
  If it cannot perform → underfitting

- Proper feature selection has to be done.
  We need some sort of feature selection in which predictors with no relationship with the dependent variable are not influential in the final model.

03/08/2023

- Bias Vs variance

weight of some predictors that do not have much predicting power, lead to high-variance low bias model.

The optimal Bias and variance is chosen to get a good model. For this, we need 'Regularization'.

- validation : using a second dataset, often known as 'validation set' we evaluate our model. In overfitted cases, Training loss will be low but validation loss will be high.

K-fold cross validation split the data into K equal blocks or subsets.

Each time, one of the K subsets is used as a validation set, while others are used for training.

Average over K trials is taken as loss.

If K = N (no. of observations)

Average squared validation loss is L^{CV}

$$L^{CV} = \frac{1}{N} \sum_{n} \{t_n - h(x_n; w_0, w_1)\}^2$$

- In regularization, instead of reducing the number of parameters, we keep all the parameters but reduce the weights of non-predictive parameters.
- There are two types of regularization -
  - (i) L_1 or Lasso Regularization
  - (ii) L_2 or Ridge Regularization

- Lasso (L_1) Regularization

The penalty is the sum of the absolute values of weights.

$$\text{Loss} = \text{Min}\left\{\sum_{i=1}^{n}(t_i - w_i x_i)^2 + \lambda\sum_{i=1}^{n}|w_i|\right\}$$

$$W_{t+1} = W_t - \frac{d(\text{loss})}{dW}$$

λ is the Regularization Parameter.

- Ridge (L_2) Regularization

The penalty is the sum of squares of weights.

$$\text{Loss} = \text{Min}\left\{\sum_{i=1}^{n}(t_i - w_i x_i)^2 + \lambda\sum_{i=1}^{n}w_i^2\right\}$$

- Selecting a good value of λ is critical to our need for finding the correct co-efficients. It is useful to automatically penalizee featwtes that make model too complex. This will decrease the importance to higher terms and will bring the model towards less complexity.

◎ Linear System

A Linear System of N equations with n unknowns, writing in linear form:

$$X_{N\times n}\, W_{n\times 1} = Y_{N\times 1}$$

If $N = m$, so X is a square matrix and assuming samples are linearly independent,

$$W = X^{-1} Y$$

If $N < m$,

we have more feature than data points.
we cannot find all solutions.
we have to increase sample size.

If $N > m$,

we have more data than features.

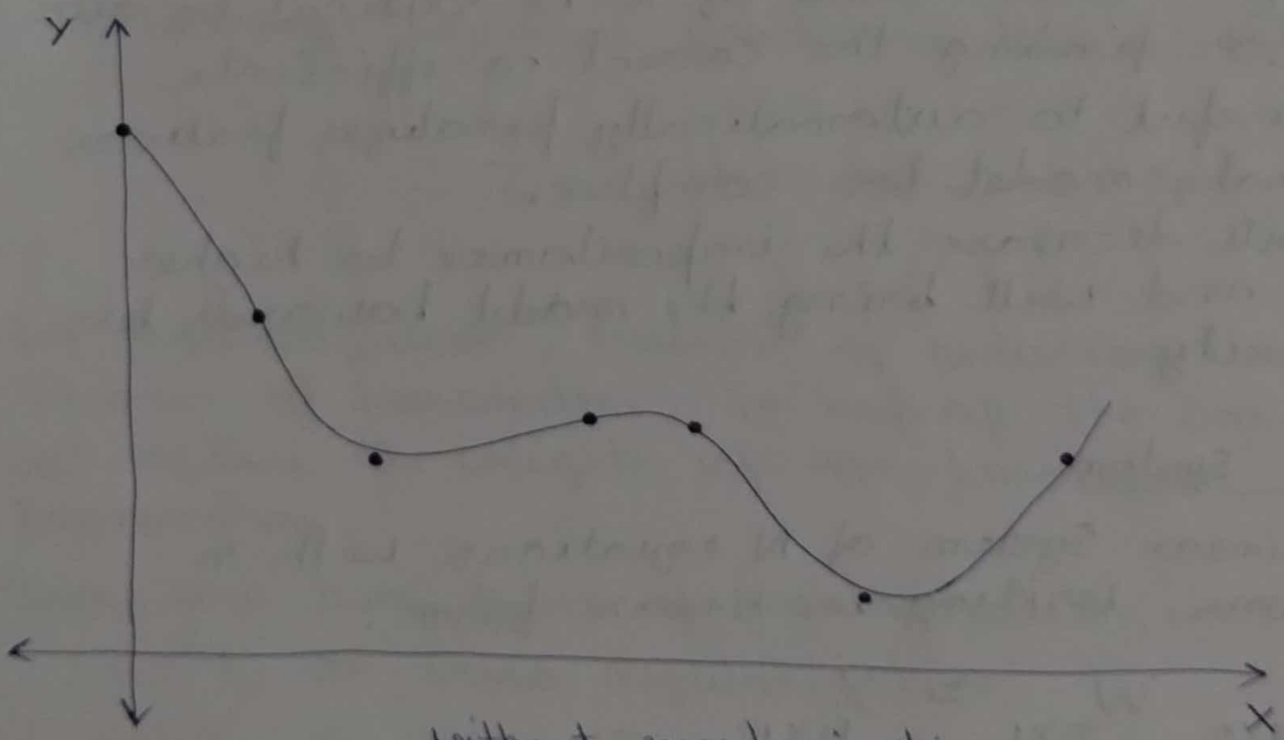
$$XW = Y$$

$$\text{or, } X^T X W = X^T Y$$

$$\text{or, } W = (X^T X)^{-1} X^T Y$$

$$\therefore W = X^T Y$$

optimization of least square loss function



without regularization

14/08/2023

- SVM was introduced by Vladimir Vapnik in 1995.
- Its main target is to find hyperplane and margin.
- We solve high dimensional problems using SVM since there is less overfitting.

- Types:

  (i) Linearly separable known as Linear SVM

  (ii) Linearly inseparable known as Non-linear SVM.

- Objects closest to the ~~[illegible]~~ hyperplane are called support vectors.

- The original objects are transformed, using a set of mathematical functions, known as kernels.

- We are given l training examples $\{x_i, y_i\}$; $i = 1..l$, where each example has d inputs ($x_i \in R^d$), and a class label with one of two values ($y_i \in \{-1, 1\}$).

  All hyperplanes in R^d are parameterized by a vector (w) and a constant (b), expressed using the equation

$$w \cdot x + b = 0$$

$$f(w, x, b) = \text{sign}(wx + b)$$

- An SVM hyperplane is an n-dimensional generalisation of a straight line in 2D.

$$WX + b = 0$$

$$W = [w_1 \ w_2 \ \dots \ w_m]$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

There will be many solutions and hence a lot of hyperplanes.

We have to select the best ones.

Many solutions can be formed by scaling (up or down) or changing angles. So, there can be an infinite number of hyperplanes.

- For a good classifier, choose one of the infinite number of hyperplanes, so that it performs better not only on training data but as well as test data.
- Larger the margin, lower is the error.
- The perceptron algorithm will definitely converge if the data is linearly separable.
- We want the hyperplane that maximizes the geometric distance to the closest data points.

In SVM literature, this marging is popularly written as $\mu(W, b)$.

- In general, there is a trade-off between the margin and the number of mistakes on the training data.

- A linear SVM can be refitted to learn a hyperplane that is tolerable to a small number of training errors.

Slack variables ξ_i can be added to allow misclassification of difficult or noisy examples. This is soft-margin classification.

Minimize

$$\frac{1}{2} W \cdot W + \lambda \sum_{k=1}^{R} \xi_k$$ subject to the

constraint $$y_i (W^T x_i + b) \geq 1 - \xi_i$$

and $$\xi_i \geq 0 \quad \forall i$$

Parameter λ can be viewed as a way to control overfitting.

- Consider a problem:

  optimise $f(x)$ subject to the constraint $h(x) = 0$.

If $f(x)$ is non-linear and $h(x)$ is linear, this is called a 'Convex optimisation Problem'.

- Take a Lagrange Multiplier α.

$$L(\underline{x}, \alpha) = f(x) - \alpha h(x)$$

$\underset{x}{\text{Min}} \ \underset{\alpha}{\text{Max}} \ L(x, \alpha)$ is an optimal solution, subject to $\alpha \geq 0$.

- For a case of binary classification starting with a training data of 8 tuples;
using quadratic programing, we can solve the KKT constraints. ~~to solve~~

- original problem

Find w and b such that

$$\Phi(w) = \frac{1}{2} W^T W \text{ is to be minimized}$$

subject to the constraints

- To convert from non-linear to linear, map the input space to feature space (higher D).

- Hyperplane

linear : $w_1 x_1 + w_2 x_2 + w_3 x_3 + c = 0$

non-linear : $w_1 x_1^2 + w_2 x_2^2 + w_3 x_1 x_2 + w_4 x_3^2 + w_5 x_1 x_3 + w_6 x_2 x_3 + c = 0$

- The original input space (non-linearly separable data) can always be mapped to some higher dimensional feature space where the training set is linearly separable.

$$\Phi : X \rightarrow \varphi(X)$$

The 3-D input vector $X(x_1, x_2, x_3)$ can be mapped into a 6-D space $Z(z_1, z_2, z_3, z_4, z_5, z_6)$ using following mappings:

$$z_1 = \varphi_1(X) = x_1^2$$
$$z_2 = \varphi_2(X) = x_2^2$$
$$z_3 = \varphi_3(X) = x_3^2$$
$$z_4 = \varphi_4(X) = x_1 x_2$$
$$z_5 = \varphi_5(X) = x_1 x_3$$
$$z_6 = \varphi_6(X) = x_2 x_3$$

So, now it becomes linearly separable.

$$w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 + w_6 z_6 + c = 0$$

- Effects of transform

(i) works for small datasets; it fails realistically sized problems.

(ii) cost of mapping

For n-D input, $\exists \quad N_H = \frac{(N+d-1)!}{d!\,(N-1)!}$

different monomials comprising a fature space of dimentionality N_H. Here d is the maximum degree of modomial

(iii) Computational cost increase in higher dimentional problem.

- Choosing Transformation: Kernel

we define a mapping $z = \varphi(x)$ that transforms the d-dimentional input vector x, to a new vector z.

Eg:

In R^2, $\quad w_1 x_1^2 + w_2\sqrt{2}\, x_1 x_2 + w_3 x_2^2 = 0$

Now, $z_1 = x_1^2$

$z_2 = \sqrt{2}\, x_1 x_2$

$z_3 = x_2^2$

Ellipse to a plane transformation.

$WX + b = 0$ in R^2 has been transformed to
$WZ + b' = 0$ in R^3.

When we transform a non-linear function to a linear one in higher dimention, then there will be an 'Inner Product'.

- Kernel Trick

If X_i and X_j are tuples, then $X_i \cdot X_j$ is regarded as a measure of similarity b/w X_i and X_j.

Similarly, $\varphi(X_i) \cdot \varphi(X_j)$ is also regarded as a similarity measure b/w $\varphi(X_i)$ and $\varphi(X_j)$.

Then, we have to find the correlation b/w $X_i \cdot X_j$ and $\varphi(X_i) \cdot \varphi(X_j)$.

In the previous example,

$X_i = [x_{i1}, x_{i2}]$ and $X_j = [x_{j1}, x_{j2}]$ are any two vectors in R^2.

$\varphi(X_i) = [x_{i1}^2, \sqrt{2}\, x_{i1} x_{i2}, x_{i2}]$ and
$\varphi(X_j) = [x_{j1}^2, \sqrt{2}\, x_{j1} x_{j2}, x_{j2}]$ are two transformed version of X_i and X_j in R^3.

Now, $\varphi(X_i) \cdot \varphi(X_j) = (x_{i1} x_{j1} + x_{i2} x_{j2})^2$

$$= (X_i \cdot X_j)^2$$

So, there exist a correlation b/w them.

$K : R^2 \rightarrow R^3$

$$K(X_i, X_j) = X_i \cdot X_j \Rightarrow \varphi(X_i) \cdot \varphi(X_j)$$

Considering data-points,

$$K(x_i, x_j) = x_i^T x_j$$

After transformation,

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$

If it is polynomial;

$$K(x_i, x_j) = (1 + x_i^T x_j)^p$$

If it is gaussian,

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

If it is laplacian,

$$K(x, y) = e^{-\lambda\|x - y\|}$$

If it is Mahalanobis;

$$K(x, y) = e^{-(x-y)^T A (x-y)}$$

If it is sigmoid,

$$K(x; y) = \tanh(\beta_0 x^T y + \beta_1)$$

- A kernel function $K(x_i, x_j)$ is a real valued function defined on R such that there is another function $\varphi : X \to Z$ such that $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$. Symbolically, we write,

$\varphi: R^m \rightarrow R^n : K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$

where $n > m$

- Properties of SVM

(i) Flexibility in choosing similarity function
(ii) Sparseness of solution when dealing with large datasets
(iii) Ability to handle large feature spaces
(iv) overfitting can be controlled by soft margin approach

- weakness

(i) Sensetive to noise
(ii) Binary classification possible, no multi-class

- Challenges

(i) choosing kernel
(ii) choosing kernel parmater
(iii) optimization criterion (Hard Vs Soft)

21/08/2023

- If we reduce the number of states in a sequential circuit, we can reduce hardware costs.

- In ML, we use likelyhood functions. That's why probability is needed.

- Degree distribution in real life network is less than the random network due to high degree of cohesiveness.

- $p(D|w)$ is evaluated for the observed data set D as a function of the parameter vector 'w', it is called the likelihood function.

  It expresses how probable the observed dataset is, for different settings of the parameter vector w. In ML literature, the negative logarithm of the likelihood function is called an error function.

- posterior $\propto$ likelihood

- 10 student data:

  115 122 130 127 149 160 152

  138 149 180

  Identify likelihood function.

24/08/2023

- serial computers requires billions of cycles to perform some tasks but the brain takes less than a second.

  eg. :- Face Recognition

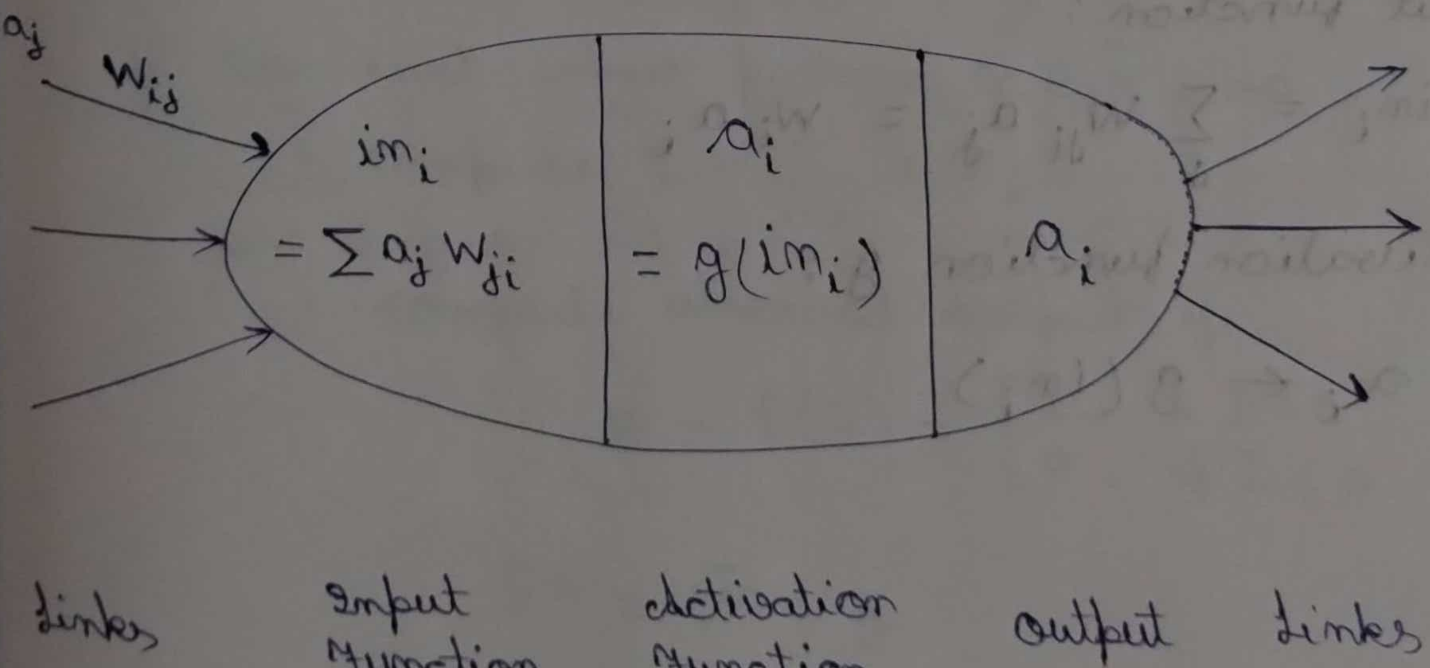
- A basic unit of a neural network is neuron. It takes inputs, does some math with them, and produces one output.

$$in_i = \sum a_j W_{ji}$$

$$a_i = g(in_i)$$

'a_j' is input and 'a_i' is the output.

- Each element of a Neural Network is a node and this node is called 'unit'.
- Units are connected by 'links'. Each link has a numeric 'weight'.



- How NN learns a task?

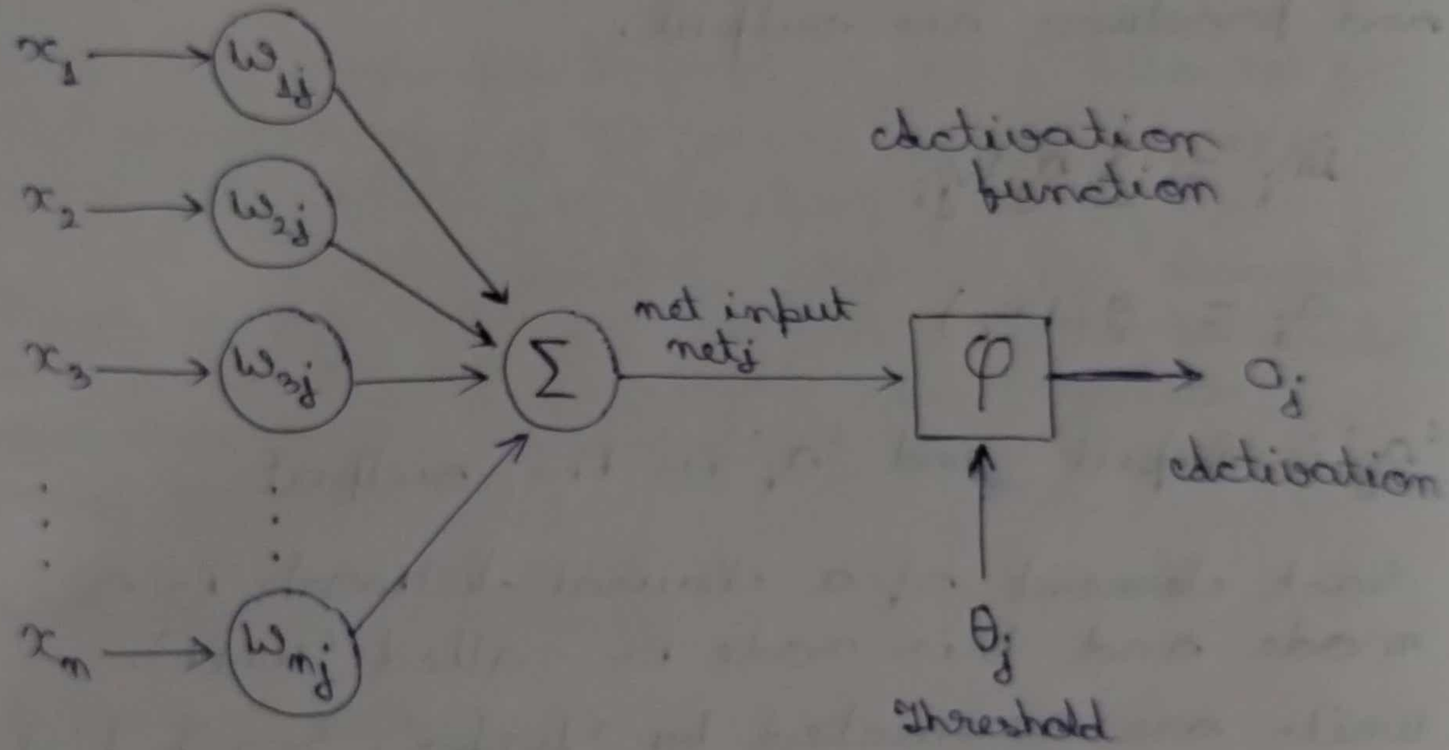
⇒ (i) Initializing the weights

(ii) use of a learning algorithm

(iii) Set of training examples

(iv) encode the examples as inputs

(v) Convert output into meaningful results



Input function:

$$in_i = \sum_j W_{ji} a_j = w_i a_i$$

activation function g:

$$a_i \leftarrow g(in_i)$$

ANN
zuraed } Book

31/08/2023

Summing junction has a bias called b_K.

- Feed Forward Neural Networks

· Multilayer perceptrons
· Deep feed-forward networks
· Vector of input features : x
vector of predicted values : $\hat{y}$
Neural activation : $y = A\left(\sum_{i=1}^{n} x_i w_{iy} + b_y\right)$

A → Some activation function

· A single neuron in such a neural network is called 'Perceptron'.

Training of single layer feed-forward NN —

1. Initialize $\bar{W} = w_0, w_1, \ldots, w_n$ to some random weights.
2. For each input pattern $x \in \bar{X}$, do
 · compute $I = \sum_{i=0}^{n} w_i x_i$
 · compute observed output y

$$y = f(I) = \begin{cases} 1, & \text{if } I > 0 \\ 0, & \text{if } I \leq 0 \end{cases}$$

$$\bar{Y}^0 = \bar{Y}^0 + y$$

3. If the desired output $\bar{Y}$ matches the observed

output $\bar{y}^o$, then output $\bar{W}$ and exit.

4. Otherwise update the weight matrix $\bar{W}$ as follows :

    - For each output $y \in \bar{Y}^o$, do
        - if the observed out y is 1 instead of 0, then $w_i = w_i - a x_i$ $\forall i$, $i = 0(1)m$
        - else, if the observed output y is 0 instead of 1, then $w_i = w_i + a x_i$ $\forall i$, $i = 0(1)m$

5. Go to Step 2

This is based on supervised learning technique.

ADALINE : Adaptive Linear Network Element is also an alternative term to perceptron.

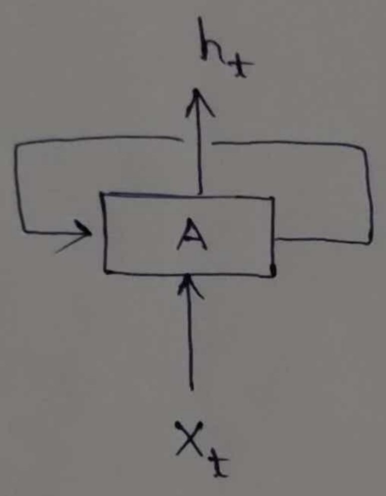
- All neurals in a particular layer have the same transfer function.

- In a matrix, W_{jk} represent the connecting weights between j^{th} neuron in the hidden layer and k^{th} neuron in the output layer

- Back Propagation Algorithm

The above discussion comprises how to calculate values of different parameters in a l-m-n multiple layer FFNN.
The principle is based on error-correction with 'Steepest-descent method'.

- Recurrent Neural Network



Dividing it into time frames, we can visualise it as sequential network