what is svm classifi...

A support vector machine (SVM) is a type of supervised learning algorithm used in machine learning to solve classification and regression tasks. SVMs are particularly good at solving binary classification problems, which require classifying the elements of a data set into two groups.

https://www.techtarget.com › whatis

What is a Support Vector Machine (SVM)? | Definition from TechTarget

See also


Random forest


k-nearest neighbors.

People also ask

What is a SVM classifier in

Tools/steps/Processes of Data sc (It consists of many steps) → mining

i) Data collection
ii) Store Data
iii) Process Data (Data imputation is a step of Pre Processing the Data)
Pre
Data removal is under Pre Processing the Data)
Data normalization — we try to convert all values in the same range. As machine learning model can't predict properly if there is a huge difference in ranges. So that we need data normalization.
It is a data transformation process that aligns data values to a common scale or distribution of values so that.
General Expression Data

	S1	S2	S3	S4	------	S100
G1	1100	12.00	10			
G2	0-1	0-1	0-12			
:						
G1000						

Predictive model

data – The iris data are a data frame of 150 measurements of iris's petal & sepal length & width, with 50 measurements for each species of 'setosa', 'versicolor' & 'virginica'.

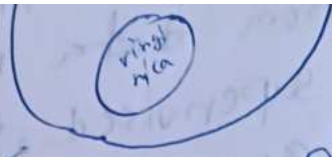
Unsupervised learning – It is a type of m.l that learns from data without human supervision. Unlike supervised learning, unsupervised m.l models are given unlabeled data & allowed to discover patterns & insights without any explicit guidance / instruction.

Supervised learning – It is a category of m.l that uses labeled datasets to train algorithms to predict outcomes & recognize patterns. Unlike unsupervised l., s.l. algos are given labeled training to learn the relationship b/w the i/p & o/ps.

Reinforcement learning – It is a m.l. training method based on rewarding desired behaviours & punishing undesired ones. In general, a reinforcement learning agent- the entity being trained- is able to perceive & interpret its environment, take actions & learn through trial & error.

2) Unsupervised

G
objective - To find & group the similar & dissimilar (similar group of datas is dissimilar from other groups of datas)



80 Training dataset (120)

test d's (30)

[Ratio of splitting
70 - 30
Tr Test
80 - 20]

Classification algos - KNN
SVM
RF
DT
Logistic Regression

21/23 and 30

We will train our model by using these classification algo & training data sets.

Then using the C.M ~~the~~ we will do chow our model is working with the test data sets.

The C. algo will predict the class label species & we will compare with the actual species. with our predicted species. Based on that we'll judge how our classi. algo works (whether it is good for our model or not)

A classification algo is a supervised learning technique that uses training data to categorize new observations

Regression - Here we try to predict numeric values instead of predicting class labels. It is a ml technique that identifies relat-ionships b/w independent & dependent variables to predic an outcome. It involv training algo to identify patterns in the distributio of each data point.

Eg Boston housing ds

attributes					
No. of rooms	length	bedrooms	width	hight	Price
Independent variable (x)					target variable (y)

there is a relationship b/w this two.

We'll try to find out the mapping b/w y & x (or i/p variables)

$[y = f(x)]$

Our regression model will try to find out this particular func (i.e how these i/p variables varies the price) & when there will be a new house it will do the same. i,e it continiously try to find out the neumeric values.

Data preprocessing - i) Remove noisy data (outliers) [eg. height 6.11, forget to give '.', so suppose then the height - 611 ft this is outlier]

(process of transforming raw data into a clean data set)

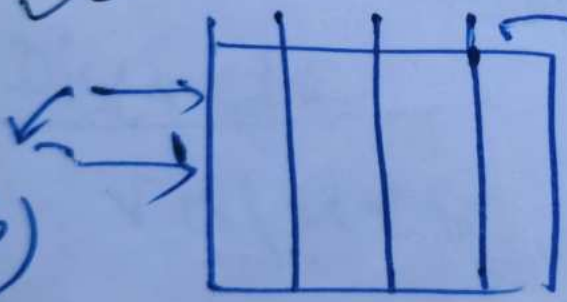
ii) Data imputation (missing value imputation) [we try to estimate the values of the missing ... can apply in various statistical

Process of transforming raw data into a clean data set)

[eg - v
sv

ii) Data imputation (mis
[we try to estimate ta
tuples, we can ap
computation]

iii) Dimensionality reduct



(s)

When p >>> s the m.l model fails to give us the desired result. this problem is called curse of dimentionality. this can be solved by dimentionality reduction or F.E or F.S. [we will remove the very less imp attribute / feature] that will not give any impact to the target attribute we will try to find out the less imp feature & will remove it, this is called feature selection (F.S). There are many algos for this. we'll try to approximate/n (the original no. of features) attributes into n' (another no. of features) no. of attributes to get desired result [9,10,000 ~ 100], this called feature execution (F.E). There is an algo for it that is called principal component analysis (PCA).

iv) Data transformation (data normalization)
In ds, diff. attributes have diff. ranges. If on this we apply the process it will not give us the desired result. So we have to take them all in same range.
It has 2 types - i) min-max
ii) Standard scaler (z-score)
i) try to transform the data in b/w (0,1)
ii) try to in b/w (-1,1)

i) min-max

x_i = orig
x_i' = upd

$$x_i' = \frac{x_i -}{x_{ma}}$$

ii) Standard

$$x_i' = x_i -$$

formula for z-score norma

we have to to the other

Types of

Numeric (

Continuous

Interval scale

Ratio scale

Continuous -
the range values

Discrete values

(36) Features >> Samples → ~~Bad d.s.~~ Bad result
(Cols) (rows)

(37) Features <<< Samples → Good data set as there are many features

x_i = original value
x_i' = updated value / transformed value

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

[Eg - $\frac{F_i}{x_1}$
$x_i \leftarrow \begin{matrix} x_2 \\ \vdots \\ x_n \end{matrix}$ } from here
min value = x_{min}
max " = x_{max}]

iii) Standard scalar (z-score)

$$x_i' = \frac{x_i - \mu}{\sigma}$$

[μ = mean
σ = Standard deviation]

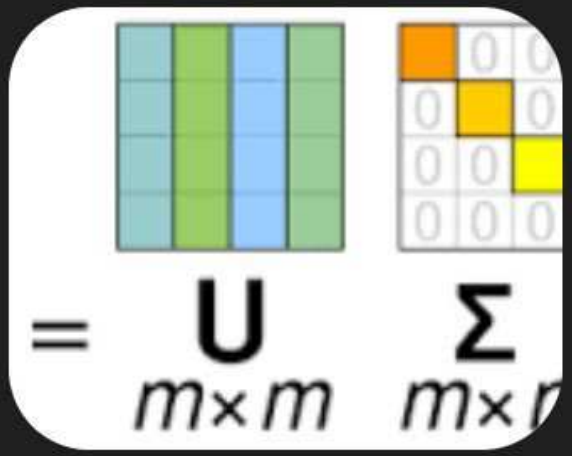
formula for z-score normalization

we have to perform this step before proceeding to the other ~~sets~~ steps.

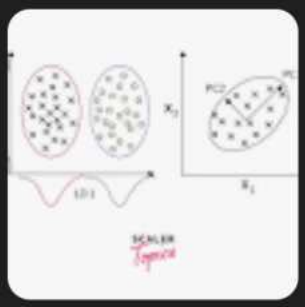
TYPES of (Data Attributes) Features

- Numeric (either int or floating point values)
  - Continuous
  - Discrete
- Categorical (it has certain string values with some certain range — Age(17, 70) ...

Listen



Singular Value Decomposition (SVD) is a matrix factorization technique that's used in machine learning (ML) for dimensionality reduction, data compression, and noise reduction. It's a fundamental part of many ML algorithms.



By fragmenting the training set into smaller training sets, the original problem is broken down into smaller problems, which is known as feature decomposition. Feature Decomposition is an effective strategy for changing the representation of classification problems. 4 May 2023

https://www.scaler.com › topics › fe...

Feature Decomposition in Machine Learning - Scaler Topics

Overview

Listen

Factor analysis is a statistical technique used in machine learning to reduce the number of variables in a dataset by modeling them as a smaller number of unobserved factors:

Explanation

Search Labs | AI Overview

+3

Listen

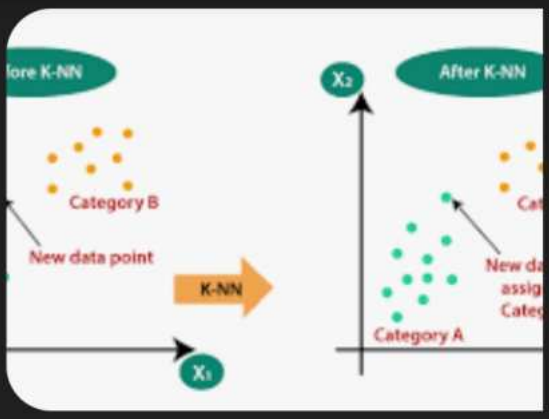
In machine learning, the Naive Bayes classifier is a supervised learning algorithm that uses probability to classify data into categories:

Overview

+3

Listen

A random forest (RF) is a machine learning algorithm that combines multiple decision trees to solve classification and regression problems:



বাংলায় In English

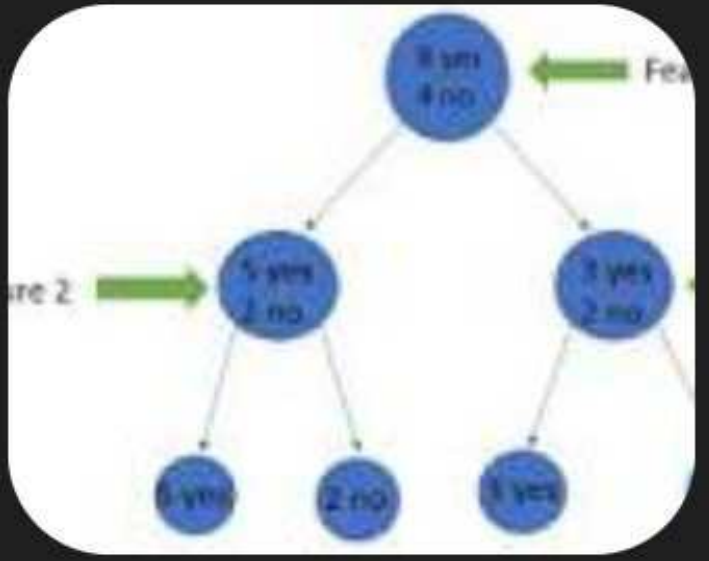
kNN, or the k-nearest neighbor algorithm, is a machine learning algorithm that uses proximity to compare one data point with a set of data it was trained on and has memorized to make predictions.

https://www.elastic.co › what-is

What is k-Nearest Neighbor (kNN)? - Elastic

About featured snippets Feedback

Listen



In machine learning (ML), a decision tree is a supervised learning algorithm that uses a tree-like structure to perform classification and regression tasks. Decision trees are often

TP = 2 (a)

FN = 1 (b)

FP = 1 (c)

TN = 1 (d)

(1) Accuracy = $\frac{TP + TN}{TP + FN + FP + TN}$

Accuracy = $\frac{a + d}{a + b + c + d}$

$= \frac{2 + 1}{5} \rightarrow \frac{3}{5} = 60\%$

limitation of accuracy ~~major~~ measure

pg 23

10,000

Actual class label

9990 (N) | 10 (P)

Predicted class label (P) → 10,000

N → 0

Imbalanced dataset

1) For imbalanced d.s. accuracy measu fails.

TP → 9990

FN → 0

FP → 10

Accuracy = $\frac{9990 + 0}{9990 + 10 + 0}$

$= \frac{9990}{10,000}$

That is very high accuracy, but classifier is very bad as it is not being able to identify a obj that belongs to negative class.

Remedy: (Some alternative measures)

① False positive rate (FPR) [metric]

$$= \frac{c}{c+d} = \frac{FP}{FP+TN}$$

③ Imp True positive rate (TPR) [metric] (Imp)

$$= \frac{a}{a+b} = \frac{TP}{TP+FN}$$

→ whenever we have high accuracy

→ It is also known as Sensitivity or recall measure

④ True negative rate (TNR)

$$= \frac{d}{c+d} = \frac{TN}{FP+TN}$$

→ It is also known as specificity

⑤ For Precision (P)

$$\frac{a}{a+c} = \frac{TP}{TP+FP}$$

⑥ F_1 - Score (F measure)

$$= \frac{2*r*P}{r+P} = \frac{2a}{2a+b+c}$$

$$= \frac{2*TP}{2*TP+FN+FP}$$

[r = recall, P = Precision]

Eg- 8.3, y, G, M → y (class label)

7.07, y, V.G, G → N
2.2, N, A, G → N

} Prediction of the class label of unknown students based upon known student data set / decision tree

Imp
Efficiency of classifiers

Imp
① Accuracy

test → class label
Actual label, Predicted label

100
70 / Y 30 / N

Confusion matrix

Actual class label	Predicted class label: yes	Predicted class label: no
yes	True positive (a)	False negative (b)
no	False positive (c)	True negative (d)

← Job offers

→ falsly categorized as no/neg.

→ No of instances

→ Falsly Predicted as yes

Linear K.F - It is a basic K.F used in ml that when the data is linearly separable, that is, it can be separated using a single line. It's mostly used when there are a large no. of features in a particular dataset.

Polynomial K.F - It is a K.F. that's often used in SVMs to learn non-linear models. It works by mapping i/p data points into a higher-dimensional (entional) feature space using polynomial functions of the original features.

RBF K.F. - It is also known as kernel func, is applied to the distance to calculate every neuron's weight (influence). It's used to measure the similarity b/w 2 data points & is based on the euclidean distance b/w them.

Convolution K.F - Convolution kernels, aka filters, are small matrices (matrix in plural) used for the convolution operation. These kernels slide across the i/p data performing element wise mul. with the corresponding pixels & producing a feature map that highlights specific patterns (features in the i/p.)

Rep. of G

K(G,

[where

NP-hard

that if

one qui

problem

class of

consider

as the

(not de

Ex

Prob

① Consi

of ran

Poissi

for th

② Use

i/p clas

test

50

2 5

-s class;
ies. [(Here
file to
class the

banana
binary form
in app. | org ;

... b/w ~~the~~ patterns. The usability of this func are used in all classifiers & clustering algos./methods.

Types of kernel func

i) **Linear K.F** — It is rep. as,

$$K(x, x') = \langle \varphi(x) \cdot \varphi(x') \rangle$$

[$\varphi(x)$ means linear pattern of x
$x' \rightarrow$ transpose of x]

eg $\langle (2\ 3\ 4)\ \begin{pmatrix}2\\3\\4\end{pmatrix} \rangle = 16$

$\langle\rangle \rightarrow$ inner product b/w 2 vectors (v_1, v_2)

ii) Polynomial K.F. –

$$K(x,x') = \langle \phi(x)\, \phi'(x') \rangle = \langle x, x' \rangle^d$$

[$\phi(x)$ = Polynomial Pattern of x] like
eg $\phi(x) = \cos(x^2 + c)$

$d=2 \rightarrow$ binomial
$=3 \rightarrow$ trinomial
$= \ldots$

iii) Pedial basis K.F. –

$$K(x,x') = \langle \phi(x)\, \phi(x') \rangle$$
$$= \exp\left(\frac{1}{2}\|x - x'\|^2\right)$$
$$= \exp\left(\frac{2}{2} x^T x - \frac{1}{2}\|x\|^2 - \frac{1}{2}\|x'\|^2\right)$$
$$= \exp(x^T x') * \exp\left(-\frac{1}{2}\|x\|^2\right) * \exp\left(-\frac{1}{2}\|x'\|^2\right)$$

where, $\rightarrow [\|x\|^2 = \|(2,3,4)\|^2 = 2^2 + 3^2 + 4^2] \Rightarrow$ NP-complete timing

The time

- Complexity of this KF is NP-complete.

$\|(7,9,0)\|^2 = 7^2 + 9^2 + 0^2$
$x = (7,9,0)$

iv) Graph K.F.

This is very complex K.F because the data

the similarity requires both of edges, so finds the soln

v) Convolution K

$K(x,x') = ($

Convolution

[• How to find
Find inc
matrix

eg

$A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$

Listen

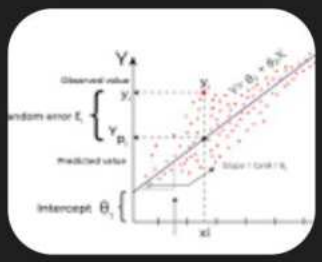
In machine learning (ML), learning rate is a hyperparameter that controls how much a model's parameters change during each iteration of the optimization algorithm:

Learning	Effect

what is linear regre...

Search Labs | AI Overview +4

Listen



In machine learning (ML), linear regression is a statistical technique that models the relationship between a dependent variable and one or more independent variables:

What it does

Linear regression finds the best-fitting linear equation to describe the correlation between the variables. It uses least squares to fit a line to the data, minimizing the sum of the squares of the residuals.

How it works

Linear regression uses a labeled dataset to map data points to linear functions. It can then be used to predict the value of the dependent

Search Labs | AI Overview

+3

Listen

Logistic regression is a supervised machine learning algorithm that uses the logistic function to predict the probability of an observation falling into a specific category. It's a statistical method that's often used for bin-

1st row is known as 1st principle Compon
2nd " " " 2nd principle compone

∴ Reduced data set = only the 1st row (1st PC)
(2×10)

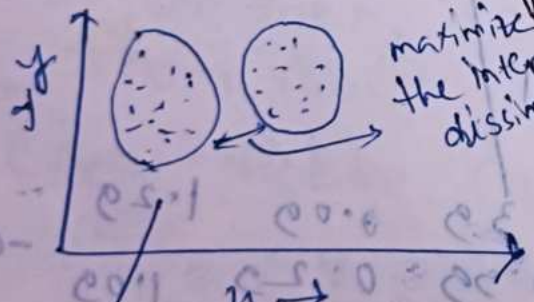
$$= \begin{pmatrix} 0.828 & -1.778 & 0.099 & \dots & -1.223 \\ -0.175 & 0.1429 & 0.3446 & \dots & -0.163 \end{pmatrix}$$

CW

ML

10/3/24

Clustering ← unsupervised learning



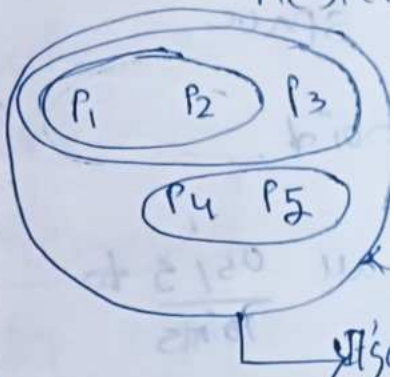
groups → clusters → modules

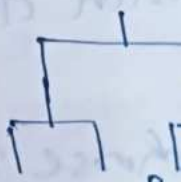
Obj: To find a set of groups of objs/clusters so that the objs in a cluster are similar among themselves

n → obj of clustering

minimize the intra-clusters distance / dissimilarity

Hierarchical





Agglomerative
Divisive

Features
(rows)
features >>> samples → then
features

as possible & objs belonging to other clusters are as dissimilar as possible.

Clustering algos

1) Partitional Clustering
2) Hierarchical "
3) Fuzzy Clustering
4) Density-based "

-0·8 | 0·31 | -0·71
-0·8 | -0·31 | -1·01

sh that which clus-ter is of which flower.

Supervised learning - Here we basically train our model, & if there is a unknown obj in the model it can correctly predict that in what catagory it falls into.

Reinforcement learning - Semisupervised learning. Here step by step we try to improve our ml model to get accurate result. If the model is not giving more accurate result then

Penalize it otherwise give it reward (save it).
Eg- self driving

Iris data - The iris data are a data frame of 150 measurements of iris's petal & sepal length & width, with 50 measurements for each species of 'setosa', 'versicolor' & 'virginica'.

Unsupervised learning - It is a type of m.l that learns from data without human supervision. Unlike supervised learning, unsupervised m.l models are given unlabeled data & allowed to discover patterns & insights

ing. Many. There is an refinement of fuzzy system, fuzzy inference system, fuzzy expert system.
(Automation)

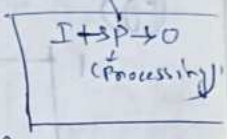
The block diagram of fuzzy inference system

(How to solve fuzzy prob. with inf sys.)

A FIS has 4 components —

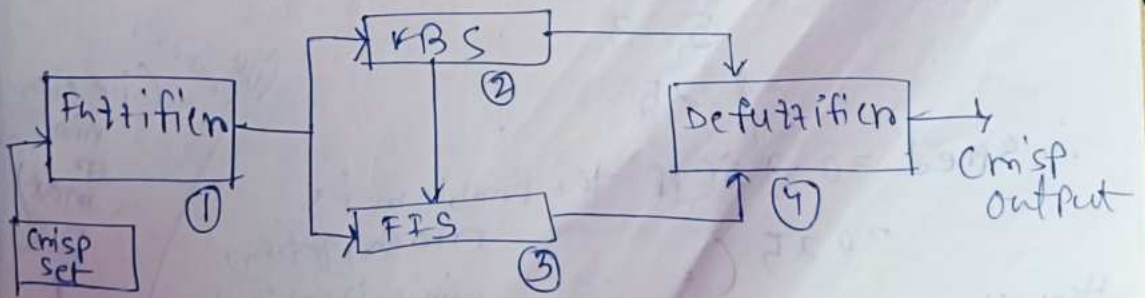
i) Fuzzyifier — This is the mf or m grade

ii) Knowledge Base - This may contain the set of rules.



iii) Fuzzy inference engine - It performs the reasoning of fuzzifiers with incorporating the knowledge based rules with the predefined fuzzy set operations. (OR, AND, NOT).

iv) Defuzzifier - Convert the outcome into FIS real value outcome. (Soft → Hard).



Batch - A batch is a subset of the training data processed together before updating the model's weights. Instead of updating after each data point, the model updates after processing a batch, improving computational efficiency.

loss function - It measures how well a neural n/w performs a task by calculating the diff. b/w the n/w's predictions & the actual values. (Aka Cost func). ③

19/9/24

Ans

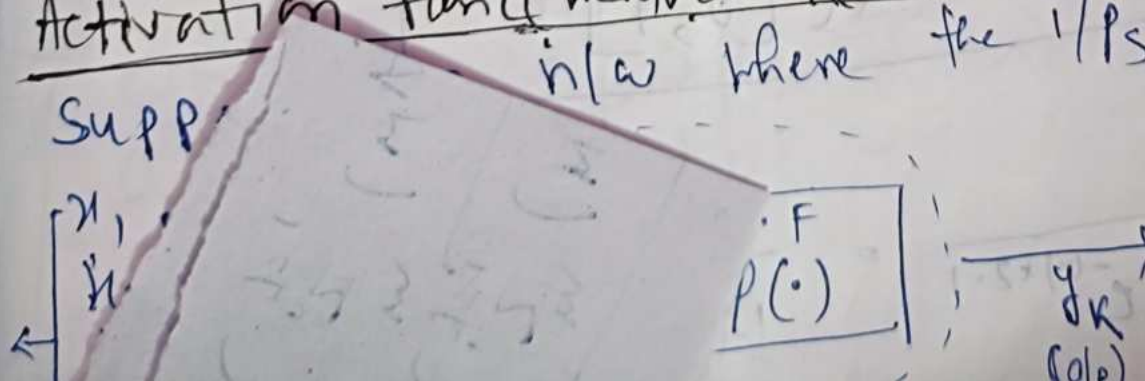
ANN

Activation func (mathematical formulation)

Incremental

n/w where the i/ps are

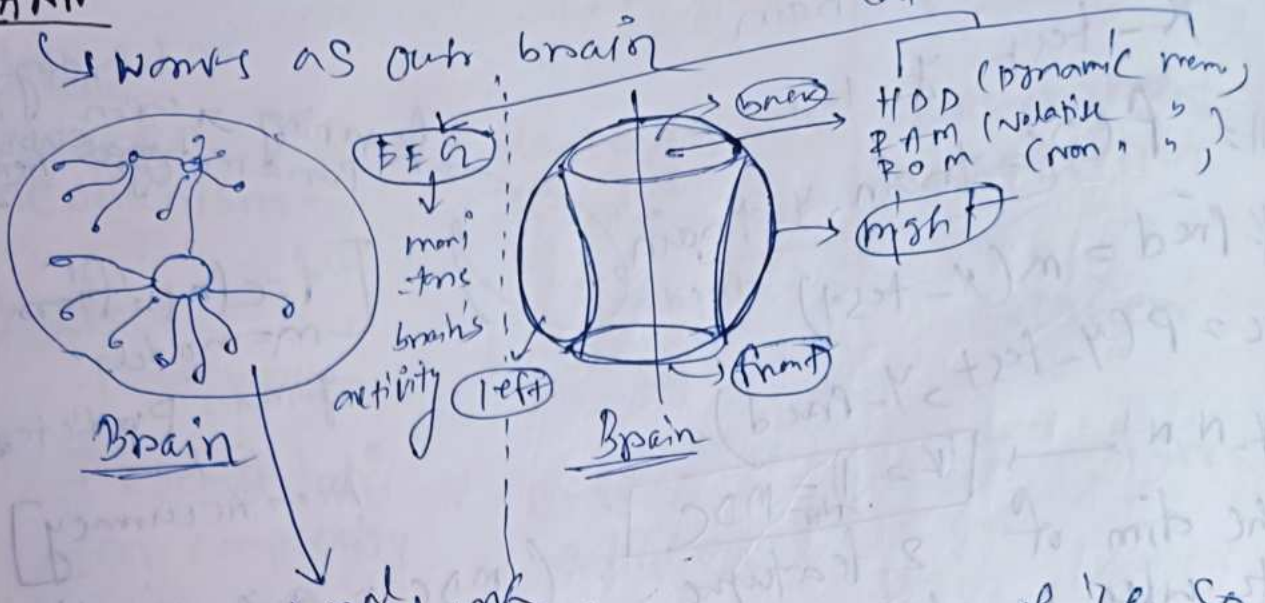
Supp



$x_k \rightarrow \varphi(u_k) = u_k s(u_k)$

ANN

↳ Works as our brain



neural network

- we are manipulating data by our own choice so it is called ANN & considered as soft Computing (History - Read one)

~~ANN~~ ANN

① It is defined based on an architecture that is composed of some layers—

i) Input layer
ii) Hidden layer or layers

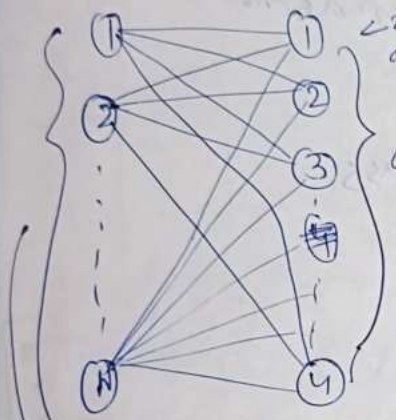
tuness >> samples → ~~Bad d.S.~~ Bad result
(now)

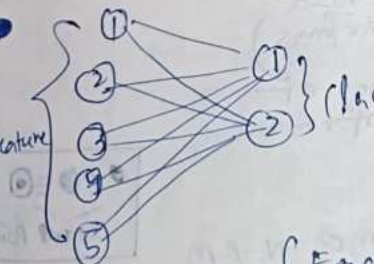
iii) output layers

(more hidden layers → more tc → less performance)

- I/P, O/P → fixed
- Hidden layer/layers — 1 or more
- For any NN there is req. of atleast 2 hidden layers; if more complex → more H.L (Hidden layers)

A general architecture of ANN with I/P (N features), 1 O/P (C, m-class) is given by —





No. of connections → N × m
→ 5 × 2
= 10

(Each feature is a node) → i.e each node is a feature

(Feature ∝ nodes)

↓

5-2 NN

eg = 15-2, 1000-2 (1000 * 2) — —
(15 * 2)

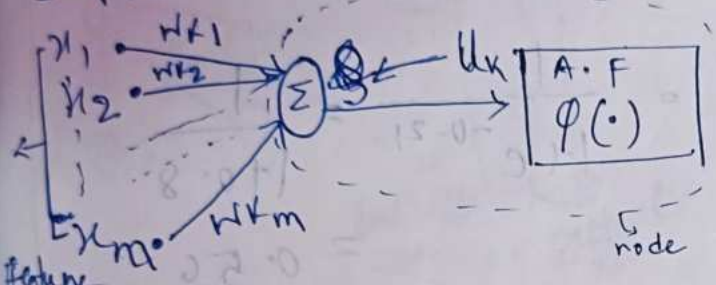
It measures how well ... a task by calculating the n/w's Predictions & the actual values. (AKa Cost func). ③

19/9/24

Ann

Activation func (mathematical formulation)

Suppose a n/w where the I/Ps are



$x_k \rightarrow \varphi(u_k) = u_k$ (say)

& u_k be the processing node of a vector.

$$u_k = \sum_{j=1}^{m} w_{kj} * x_j$$

$\varphi(u_k)$

$y_k = \varphi(u_k) = \varphi(u_k + b)$

It will not produce much diff. than u_k but will stop comm

$y_k = \varphi\left(\sum_{j=1}^{m} (w_{kj} * x_j) + b\right)$

~~bias will be added to every $w_{kj} * x_j$ (i.e., $u_k = w_k x + b$)~~

e-g: ↳ some comm language

bias → b

[• Bias(b) is always constant (0 to 1 → range)

↓ Activation func contains some comm, to stop it we use bias. → bias

[eg. bias
$y(n) = \tan x$
$y(n+2) = \tan(n+2)$] → the diff. will not be much, it will be very very small

⊙ With the diff. formulation of $\varphi(x)$ the diff. activation funcs are defined —

i) Threshold func

$$\varphi(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & x < 0 \end{cases}$$

ii) Sigmoid activation func (A.F.)

$$\varphi(x) = \frac{1}{1+e^{-ax}}$$, a = learning parameters, $a \in [0,1]$.

eg.

$$\varphi(x) = \frac{1}{1+e^{-0.1 \times 2.1}} = \frac{1}{1+e^{-0.21}} = \frac{1}{1+0.8} = 0.56$$

let $x = 2.1$

$\therefore \varphi(x) = 0.55$

$\therefore y_r = \varphi(x) = 0.55$

let 1 set at O/P, $\therefore 1 - 0.55$ (error)

[• Forward & backward prop. → w change + tuning]

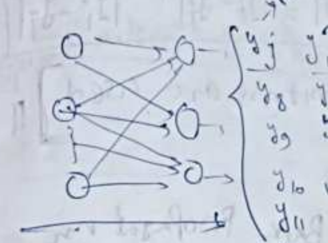
iii) tangent hyperbolic func

$\varphi(x) = \tanh(x)$

[tan(x) v/s tanh(...
Range ($-\infty$ to ∞) | Range with a...

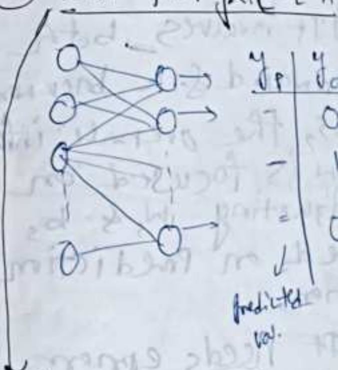
misleníous topic

① F.F n/w



FF → feed form...

② Back propagation n...



The nature of er...

be the i-th actual ...
j-th level, so th...
distance metr...

i) $\|.\|_1$ (Norm1)

ii) $\|.\|_2$ (Norm2)

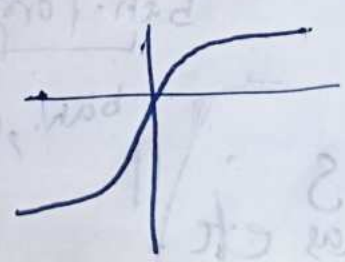
whereas, $\|.\|_1$ m...

used in nns, Particul
hidden layers, autoencoders, &
nns(RNNs). (It is also used
layers of feedforward nns).]

need

tan vs tanh

tan	tanh
1) unbounded (−∞ to ∞)	1) Bounded (−1 to 1)
2) Periodic	2) Non-Periodic



ReLU formula is : f(x) = max(0,x)

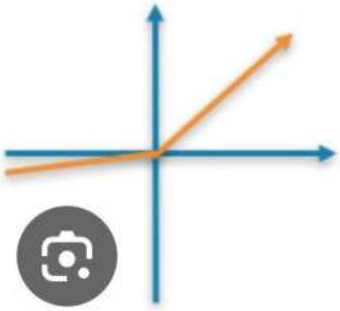
Both the ReLU function and its derivative are monotonic. If the function receives any negative input, it returns 0; however, if the function receives any positive value x, it returns that value.

https://www.deepchecks.com › rec...

What is Rectified Linear Unit (ReLU)? Function and Importance

People also ask

Leaky ReLU Activation Function



$$leakyrelu(z) = \begin{cases} 0.01z & for\ z < 0 \\ z & for\ z \geq 0 \end{cases}$$

Range: -infinity to infinity

another.

Overfitting & Underfitting - [They are 2 types of errors that can occur when training a machine learning model:] ||

Overfitting - A model that performs well on training data but poorly on test data. [This can happen when a model is too complex, overfits a single data set, or learns too many details from the training data.] ||

Underfitting - A model that performs poorly on training data & is unable to generalize to new data. [This can happen when a model is too simple or doesn't train long enough on enough data points.] ||

Activation function - It determines which neurons should be activated as info. passes through the n/w. [Activation fxns are imp. as they allow neural n/ws to learn complex patterns & relationships in data.] ||

Eg - Sigmoid, tanh, ReLU, Softmax, ELU etc.

Effects - i) these introduce non-linearity to a neural n/w, enabling it to learn complex patterns.

ii) Fxns like ReLU or sigmoid determine how

(10) UCI Database
(11) Iris.csv
(11) Heart disease

m
v
s
k

(Same as mvks code, do same)

10/11/24

18/12/24

ML viva

(1) INT_MAX → Macro that rep. the max int. val
INT_MIN → macro that rep. the min int val.

(2) Variance → $s^2 = \frac{\sum_{i}^{n}(x_i - \bar{x})^2}{n-1}$ (sample)

Kurtosis → $Kurt = \frac{\mu_4}{\sigma^4}$

$$Kurt = n * \frac{\sum_{i}^{n}(y_i - \bar{y})^4}{\sum_{i}^{n}(y_i - \bar{y}^2)^2}$$

Skewness → $Skew = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^3}{(n-1) * s^3}$

Where s = std

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

④ Divergence – It's a func that takes 2 probability distributions as input, & returns a no. that measures how much they differ.

⑤ Dataframe – It organizes data in table like format with rows & cols., similar to spreadsheet, where each col rep. diff. features & each row rep. a data point/ obj.

⑤ Numpy – Free, open-source python library that provides support for large, multi-dimentional arrays & mathematical funcs. (eg. linear algebra, trigonometric operations, stats operations, selection, sorting etc.)

⑥ Pandas – Popular, open source library that is specifically designed for data manipulation & analysis. It provides tables, matrices etc.

⑦ matplotlib – It's a popular data visualization library in py. It provides tools to create stats, plots, histograms etc. (plotting lib)

⑧ Scipy – scientific python. It is an open-source py lib. used for scientific & technical computing like skewness, kurtosis, normalization

10/11/24

18/12/24

max int val.

min int val.

ple)

symbolize New beg

OK

etc.

9) np.random.seed(42) → for reproducibility

10) col wise mean, var → data.mean()
data.var()

Kurtosis, skewness → stats.kurtosis()
stats.skew()

10) legend() → It is used to describe elems from a particular area of a graph.

11) subplots() → to fix many plots into 1 plot

12) axs [x, y] → used to plot/hist of subplots
(axs → 2D array; x → At x-axis; y → At y-axis)

13) <u>How to load a d.s.?</u>

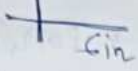
pd.read_csv('File-path')

14) map(float, input("enter val:").split()) → maps strings into float
(numeric strings)

split strs into chars, & then convert it into float/int.

15) tuple(x) → convert/put those int/float vals in tuple (tuple is immutable)

16) List is mutable (Editable)

17) figsize = (6, 4) → 

18) zorder = 5 → It ensures that the points are plotted on the top of the curve.

19) np.random.uniform() → For uniform distribution of vals it is used, i.e values should be alike like 10, 3, 7, 2, 0, 5, 12 etc not, 0, 1000, 12, 55, 9999 etc.

20) Zip(Params-A, Params-B) → It is used to combine eles from multiple iterables into single iterable of tuples.

21) enumerate() → It is used to loop over an iterable & automatically provide an index for each of them.

22) range() vs enumerate()

It is used when you only need a sequence of nos.

It is used when you need both index & the value of each item in an iterable.

23) np.zeros((R.shape[0], S.shape[1])) → np.zeros is used to create a new array filled with 0s.

takes 5mi

takes jar 3

24) plt.imshow() → It is used to display the matrix as an image

cmap = 'viridis' → popular color map

aspect = aspectratio = $\frac{\text{pixels in width}}{\text{pixels in height}}$

25) rand() = random()

26) zip(population, fitness_scores) → pairs each chromo with it's fitness value

27) sorted(zip(popus, fitness), key = lambda x: x[1], reverse = True)

↓

Sorts these pairs by fitness val in descending order.

28) format(x, '05b') → It converts int (decimal) → Binary

int

29) int(binary_str, 2) → Binary → Decimal (int)

30) selected_index = random.choices(range(len(population)), weights = selection_prob, k=1)[0]

↓

randomly selects 1 chromo based on its selection Probability

31) append() v/s extend([1,2])

used when you want to add single ele

Used when you want to add

LONG BOOK

to a list

multiple elements from an iterable to the list

(32) df.loc[df['Fitness'].idxmax()]
↓
To identify the best solution after all generations.
To Identify the individual with highest fitness score. (Best individual)

(33) Probability density (= PDF)
(density = True) - It's a way to show how likely it is for a value to occur, with higher vals. (meaning more likely), & the total area under the curve always adding up to 1.

(34) PMF or Probability mass func - It gives the prob. of a given no. of events, happening in a fixed interval of time or space.

(35) PDF or Probability density func - It shows how likely diff. vals of a continuous random variable are .... (same as prob. density)

(36) Features >> samples → ~~Bad d.s.~~ Bad result
(cols) (rows)

(37) Features >>> samples → ~~Bad~~ Good dataset as there are many features