(1)

22/05/2020

Data mining and Warehousing

Nithin T. Sanil
Roll No: 30
S_8 CSE

1) Hyper planes can correctly classify all the given data tuples. we have the hyperplane with very large margin at classifying future data tuples then the hyperplane with the smaller margin. This is why, SVM searches the hyperplane with the great margin that can be defined as maximum marginal hyperplane (MMH). On this the association gives much seperation between the classes. The distance from MMH to nearest training tuple of the classes. The MMH in the new space corresponds to a non-linear hyper surface in orginal surface is called MMH

3) Back propagation :-

This learns by iteratively processing database of training tuples, This compares the network prediction for each tuple with the actual value we required. The modification is made in the backward direction that is from output layer through each layer down to first hidden layer hence the name back propagation. The target values may be known class label of training tuples

Algorithm of Back propagation

Inputs:-

→ D: data set consisting of the training tuples and their ~~targ~~ target value

→ L: This has the learning rate

→ Network: a multilayer feed-forward network

Steps:

- step1: initialize all weights and bias in network
- step2: while ~~end condition~~ exit condition is not satisfied
- step3: { for all training tuple x in D // give the input
- step4: for each input layer unit j {
- step5: Output (j) = Input (j)

  // Output of a input unit is assigning its actual value
- step6: for each hidden or output layer unit {
- step7: $I_j = \sum_{i=1}^{n} w_{ij} O_i + O_j$; // compute the net input with respect to previous layer
- step8: $\text{Output}(j) = \frac{1}{1+e^{-I_j}}$ // compute each output
- step9: for each unit j in output layer
- step10: Error (j) = Output (j) (1 - Output(j)) ($T - O_j$)
- step11: For each j in hidden layer last to first hidden layer

③ step 12 : $error(j) = Output(j)(1 - output(j)) \sum_{k=1}^{n} Err_k\, \omega_{jk}$ // Computing the errors

step 13 : For each weight ω_{ij} in the network.

step 14 : $\Delta\omega_{ij} = (l) Err_j\, Output(i)$ // weight

step 15 : $\omega_{ij} = \omega_{ij} + \Delta\omega_{ij}$

step 16 : for each bias O_j in network

step 17 : $\Delta O_j = (l)(Err_j)$

step 18 : $O_j = O_j + \Delta O_j$

step 19 : Compute the rate

step 20 : stop

2) Given Condition

classification based on {Pepper = false, ginger = true, chilly = true}

	Pepper	Ginger	chilly	like
A	T	T	T	F
B	T	F	F	T
C	F	T	T	F
D	F	T	F	T

T: True F: False

(4) Here on the above table we taking A, B, C, D as the given data accordingly their likes are given.

By using hamilthonian distance.

$$D_n = \sum_{i=1}^{3} |x_i + y_i|$$

~~Consider A~~

Here their are 3 cases they are pepper, ginger and chilly

So $\sum$ varies from 1 to 3

ie $\sum_{i=1}^{3} |x_i + y_i|$

Consider A,

$$A = \sum_{j=1}^{3} |1+0+0| = 1$$

Consider B,

$$B = \sum_{j=1}^{3} |1+1+1| = 3$$

Consider C

$$C = \sum_{j=1}^{3} |0+0+0| = 0$$

Consider D

$$D = \sum_{j=1}^{3} |0+0+1| = 1$$

~~Both are~~
~~Both are~~ $\to$ 1
Both are true = 0
Both are different = 1

Comparing A, B, C and D ~~A~~ A-D ~~a~~ have closer value ~~and~~ ie 1 and more close to '1' is 0 ie A-D have close ~~a~~ C than B

Scanned with CamScanner

A:- false

D: True

C: false

Among this false has majority So we can say People won't like it