Learning Vector Quantization solved Numerical

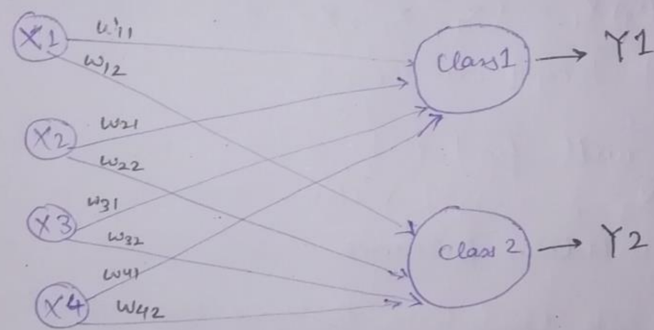
By Devesh Mishra (202404)

e.g. Construct and test on LVQ net with five vectors assigned to 2 classes. The given vectors along with classes are :

Vectors	Assigned class (Cluster)
X1 = [0 0 1 1]	1
X2 = [1 0 0 0]	2
X3 = [0 0 0 1]	2
X4 = [1 1 0 0]	1
X5 = [0 1 1 0]	1

consider learning rate $(\alpha) = 0.1$

solution →



∵ Here every vector has four inputs and no. of output classes are 2.

∴ We draw a neural network with 4 inputs and 2 outputs.

→ Number of output classes given in problem is 2.

∴ consider first two vectors as a initial weight vectors and remaining three as input vectors.

∴ initial weight vectors :- $W_1 = [0\ 0\ 1\ 1]$

$W_2 = [1\ 0\ 0\ 0]$

$\because$ no. of input in each vector are 4 (ie. $i = 1, 2, 3, 4$)
So, to initialize the weight matrix we have to consider 4 rows.

Also there are 2 output classes so, we have to consider two columns (ie. $j = 1, 2$).

$\Rightarrow$ So, weight matrix has 4 rows 2 columns.

$\Rightarrow$ initialized weight matrix $(w_{ij}) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$
or (first weight matrix)

Now, from remaining three vectors, consider first input vector(X_3): $X_i = (x_1, x_2, x_3, x_4) = (0\ 0\ 0\ 1)$
and target cluster $= 2$

So we will find out distance between weight vector 1 and X_3, weight vector 2 and X_3 with the help of Euclidean Distance.

$\rightarrow$ Calculate Euclidean Distance b/w clusters $j = 1, 2$ and first input vector by the formula:

$$\boxed{D_j = \sum_{i=1}^{n} (w_{ij} - x_i)^2}$$

Distance b/w column 1 and X_3 :-

first we calculate distance b/w column 1 means $j = 1$,

$$D_1 = \sum_{i=1}^{n} (w_{i1} - x_i)^2$$

$$D_1 = (0-0)^2 + (0-0)^2 + (1-0)^2 + (1-1)^2 = 1$$

Similarly : $D_2 = (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 = 2$

Compare D_1 and D_2. $\Rightarrow$ $D_1 < D_2$

So winning cluster is $j=1$ (by considering the minimum value). Update the weight of only column $j=1$!

Equations to update the weights are :-

$$W_{ij} \text{ (new)} = W_{ij} \text{ (old)} - \alpha [X_i - W_{ij} \text{ (old)}]$$

(why we are updating because, Given input vector belongs to target cluster 2, winning cluster is 1).

But Target cluster $\neq$ Winning cluster).

$\because$ It is '$\neq$' sign $\therefore$ we put minus in formula.

Suppose Target cluster = Winning cluster, then we will put plus in formula.

$\because$ $\alpha = 0.1$ and $j = 1$

$W_{11} \text{ (new)} = 0 - 0.1 (0-0) = 0$

$W_{21} \text{ (new)} = 0 - 0.1 (0-0) = 0$

$W_{31} \text{ (new)} = 1 - 0.1 (0-1) = 1.1$

$W_{41} \text{ (new)} = 1 - 0.1 (1-1) = 1$

So, new weight matrix $(W_{ij}) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1.1 & 0 \\ 1 & 0 \end{bmatrix}$

Consider, next input vector ie ~~W~~ $X_4 = (1\ 1\ 0\ 0)$

its target cluster = 1.

Similarly we will find out D_j using updated weight matrix $W_{ij} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1.1 & 0 \\ 1 & 0 \end{bmatrix}$

$\therefore\ D_1 = (0-1)^2 + (0-1)^2 + (1.1-0)^2 + (1-0)^2 = 4.21$

$D_2 = (1-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 = 1$

Here ! $D_2 < D_1$ so winning cluster is $j=2$ by considering minimum value. So update column $j=2$ of W_{ij}.

update weight of column 2 as:-

$W_{12(new)} = 1 - 0.1(1-1) = 1$

$W_{22(new)} = 0 - 0.1(1-0) = -0.1$

$W_{32(new)} = 0 - 0.1(0-0) = 0$

$W_{42(new)} = 0 - 0.1(0-0) = 0$

$\Rightarrow\ W_{ij} = \begin{bmatrix} 0 & 1 \\ 0 & -0.1 \\ 1.1 & 0 \\ 1 & 0 \end{bmatrix}$

$\Rightarrow$ Now Consider next input vector $X_5 = (0\ 1\ 1\ 0)$

its target cluster = 1 . .

Again :- $D_1 = (0-0)^2 + (0-1)^2 + (1.1-1)^2 + (1-0)^2 = 2.01$

$D_2 = (1-0)^2 + (-0.1-1)^2 + (0-1)^2 + (0-0)^2 = 3.21$

$\Rightarrow\ D_1 < D_2$ (winning vector = 1 , & Target cluster = 1)

$\therefore \quad w_{ij}(new) = w_{ij}(old) + \alpha\,[x_i - w_{ij}(old)]$

$w_{11}(new) = 0 + 0.1\,(0-0) = 0$

$w_{21}(new) = 0 + 0.1\,(1-0) = 0.1$

$w_{31}(new) = 1.1 + 0.1\,(1-1.1) = 1.09$

$w_{41}(new) = 1 + 0.1\,(0-1) = 0.9$

So final updated weight matrix $(w_{ij}) = \begin{bmatrix} 0 & 1 \\ 0.1 & -0.1 \\ 1.09 & 0 \\ 0.9 & 0 \end{bmatrix}$

In this way we can update the weight of the Learning vector Quantization neural network.