20-9-2021 Tuesday

week#4 lecture #4

First stretch the image 3 dimensional img to 1 Dimensional array

0	0
255	0

R

255	255
0	0

G

0	0
0	0

B

G stretched to 1D array

0	0	255	0	255	255	0	0	0	0	0	0

R G B

Now, element wise subtraction of images from test set & Training set images

$|Tr1 - T|$

e.g

10	0	5	6	5

0	10	0	10	0

=

10	-10	5	-4	5

Abs =

10	10	5	4	5

$10 + 10 + 5 + 4 + 5 = 34$

L_1 $|P_1 - P_2|$

L_2 distance $(P_1 - P_2)^2$ Penalizes those pixels that have

Large distance

for example In L_1, distance of 2 Pixels is 10, 4 Separately while in L_2 distance, same is 100, 16 difference in first is 6, while in L_2 is 84.

HyperParameters:

Value of K is odd bcz at even we get ties.

K can be 1, 3, 5, 7, 9, ...

we put these values for K & check accuracy on each value & Pick best accuracy for that dataset

Accuracy = $\frac{\text{No. of Correctly Predicted}}{\text{No. of total Images/Total Predicted Images}}$

e.g

C	C	C
P	D	D
S	S	S

Predicted Labels

C	S	C
D	S	D
D	S	C

Actual labels

Accuracy = $\frac{5}{9}$ = 55%

For K=1 in Training set, Accuracy = 100%

So, we don't use training set to check accuracy.

we divide train set into 2 sets

a) Train set → used for Predictions

b) Validation Set → used to check Accuracy.

Also K can't be greater than no. of imgs in the train set

Test set → unseen data

Problems:

① Computationaly expensive (Slow in testing for large data set)

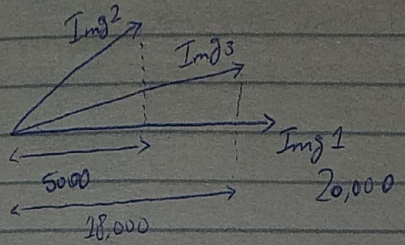
② Curse of Dimensionality.

Linear Classifier:

we try to learn generic img of all the classes we want to predict.

Then take difference from new unseen img from these generic img & whoever is closest will be that image class.

In this we use dot product as our distance metric



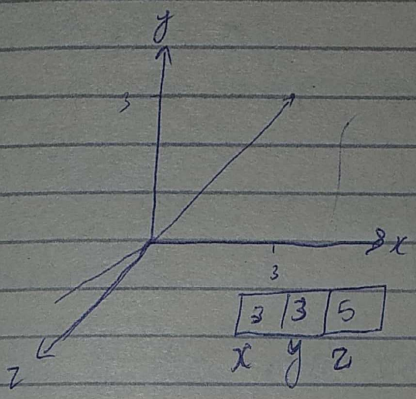
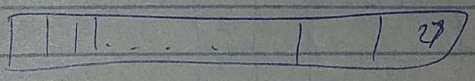
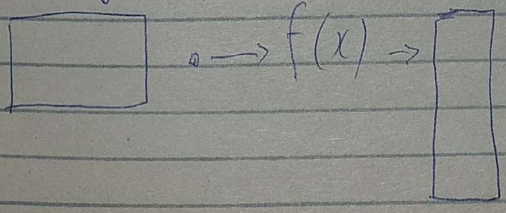


Image can be represented as vector

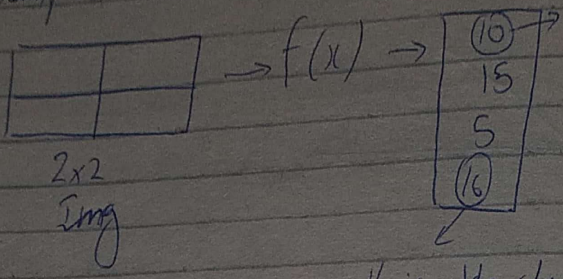
like  27 dimensional vector

$2 \times 2 \times 3 = 12$ dimensional Point

More value of dot product means this img class is more closer to test img



for example



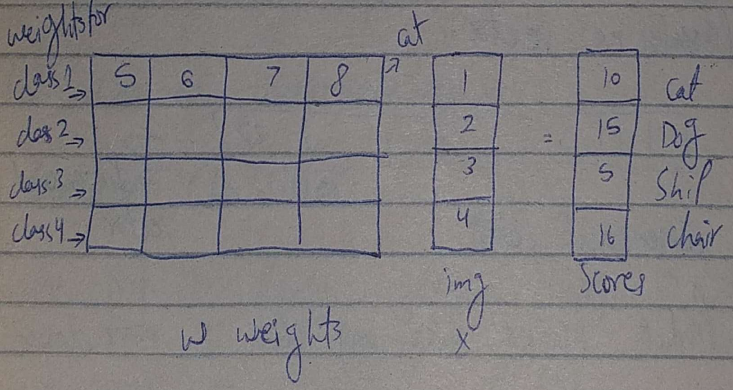
16 is the class label we will give to this test class

W X $=$

(4×4) (4×1)

img stretched as vector form

4×1



1	2
3	4

5	6
7	8

weights for class 1

generic Img for 1st class

D

less imp ↑ more imp ↑

0	-2	5	4

argmax(wx)
index of max value in column vector is the class Predicted

$y = mx + c$ if $c = 0$, line passes through 0

x_1	x_2	x_3	x_4

$m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4$

bias is added to learn more specific Accurate images