28-9-2021 Tuesday

Week#5 lecture #6

SVM classifier:

① Score function

we input an image into a function $f(x)$ which gives scores for each class & max score is predicted class

② Hinge loss function

$$L_i = \sum_j \max(0, s_j - s_y + 1)$$

example:

10
20
18 ← True class
19

loss $= \max(0, 18-10+1) + \max(0, 18-20+1) + \max(0, 18-19+1)$

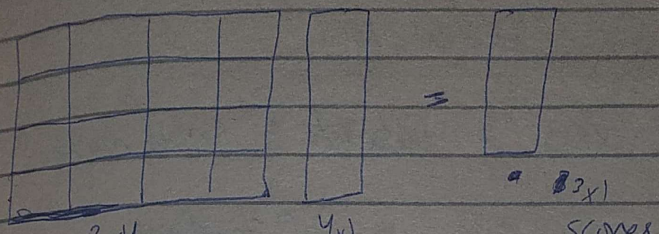
$= 9 + 0 + 0$

$= 9$ $= \frac{1}{4} \times 9$ $= 50-60\%$

loss = max(0, 10 - 18 + 1) + max(0, 20 - 68 + 1) + max(0, 19 - 18 + 1)

= 0 + 3 + 2

= 5



How to calculate weights W?

① Random weights every time (Naive Approach):-

Initialize W & calculate loss, after N no. of iterations, pick least loss w, this w is the best that we found in these iterations. W initialized after each iteration. we calculate loss for each image in train set.

best_loss → float

best w

while iter in range(10,000):

W = random initialization

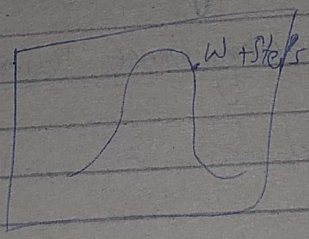
L = loss (X, W)

if L < best_loss:

best W = W

If weights are less & Data set is small, maybe it is good for a greater no. of iterations

② Increase / Decrease steps in Weights W:-



random W = random initialization

loop

Delta = 0.0001 x random W] same dimensions as W

DW = W + Delta [0.0001 / step size

loss = loss (DW, X)

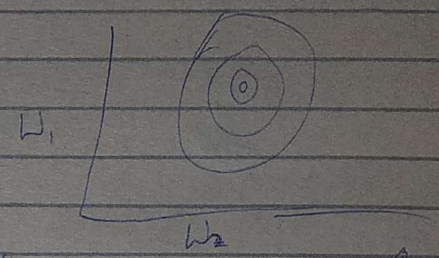
if L < best_loss:

~~Best loss = Loss~~

B random W = DW

Now, we are just adding some noise in initialize random w.

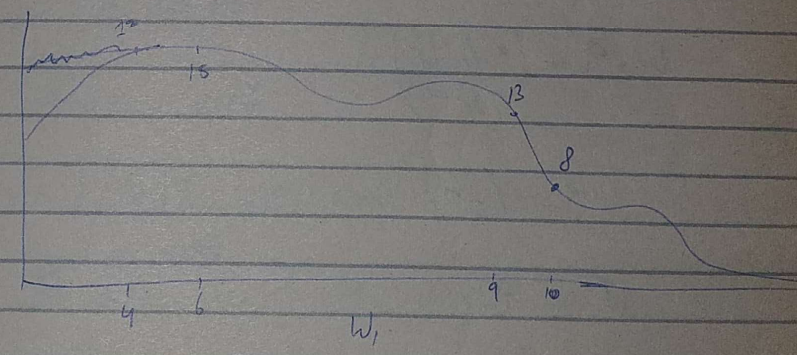
No.s in initialize w are also given small value.



only one global maxima or minima.

Loss function graph

③ Gradient Decent for Loss:-



Slope = $\frac{15-12}{6-4} = \frac{3}{2} = 1.5$

1 unit in x
1.5 unit change in y.

$\frac{f(x+h)-f(x)}{h}$ = [illegible] → 1.5

$\frac{8-13}{10-9} = -5$

We encourage decrease in loss so we increase value of W by 1.

$$w = w - \frac{dL}{dw}$$ ← Gradient Decent

- ve: loss is decreasing
+ ve: loss is Increasing

Function example.

$$w = 9 - \left(\frac{8-13}{10-9}\right)$$

$$= 9 - (-5)$$

$$w = 14$$

$$w = 4 - \left(\frac{15-12}{5-4}\right)$$

$$4 - \left(\frac{3}{1}\right)$$

$$w = 1$$

$W = W - \alpha \frac{dL}{dW}$

α ↓ Jump / Learning rate

Avoid Local Maxima / Minima Problem

if α is greater, step size is great — if it is too great, we can get into this problem.

$h \to$ approaches to 0

$W = 3072$

$L = \text{Loss}(x)$ // $f(x) = L$ for whole training set

for $dW = W$

for W in W

$dW = W + h$

$L = f(W+h) = \text{Loss}(W+h)$

we have to learn every weights in weights 3072

20	19	15	→	20	19	15.01
				$W+h$		

so we again calculate $f(W+h) = L$

$\frac{f(W+h) - f(x)}{h} = \alpha \frac{dL}{dW}$

we also do for

20	19	15	→	20	19.01	15
W						$W+h$

Mathematical Gradient Decent are slow

④ Analytical Gradient Decent:

$f(x) = x^2$

$\frac{df}{dx} = 2x$

$x^n = nx^{n-1}$

$\frac{f(x_1) - f(x)}{1}$

$= \frac{(x+1)^2 - x^2}{1}$

$= \frac{x^2 + 2x + 1 - x^2}{1}$

$= 2x + 1$ → ignore constant

$= 2x$

a_0	a_1	a_2	x_0			True class
b_0	b_1	b_2	x_1	= S =	S_2	
c_0	c_1	c_2	x_2		S_3	

$L(S) =$

$L = \max(0, S_2 - S_1 + 1) + \max(0, S_3 - S_1 + 1)$

$$L = \max\left(0, (b_0x_0 + b_1x_1 + b_2x_2) - (a_0x_0 + a_1x_1 + a_2x_2) + 1\right)$$

$$+ \max\left(0, (c_0x_0 + c_1x_1 + c_2x_2) - (a_0x_0 + a_1x_1 + a_2x_2) + 1\right)$$

$$\frac{dL}{da_0} = 2\max(0, -x_0)$$

$$\frac{dL}{db_0} = \max(0, x_0)$$

$$a_0 = a_0 + \alpha \frac{dL}{da_0}$$

$$a_0 = a_0 + \alpha\left(2\max(0, -x_0)\right)$$

$$b_0 = b_0 - \alpha\left(\max(0, x_0)\right)$$

$$\frac{dL}{da_0} = x_0 \times [S_2 - S_1 + 1 > 0]$$

$$\frac{dL}{db_0} = -x_0[S_2 - S_1 + 1 > 0] - x_0[S_3 - S_1 + 1 > 0]$$

225	S_1
20	S_2
21	S_3

$20 - 225 + 1 > 0$

Vectorized Code

$$\frac{1}{\Sigma 2}\begin{bmatrix}1\\-2\\0.5\end{bmatrix} = \begin{bmatrix}e^{1}\\e^{-2}\\e^{0.5}\end{bmatrix} =$$

→ all +ve values & probabilities

Turn these into positive values keeping their rank same

Probabilities of Each Class & Normalization of Scores

To get probabilities of each class,

$$\begin{bmatrix} e^{1.0} \\ e^{-2.0} \\ e^{0.5} \end{bmatrix} = \begin{bmatrix} e^{1.0}/(e^{1.0}+e^{-2.0}+e^{0.5}) \\ e^{-2.0}/(e^{1.0}+e^{-2.0}+e^{0.5}) \\ e^{0.5}/(e^{1.0}+e^{-2.0}+e^{0.5}) \end{bmatrix}$$

↓

overall sum of these will be 1.

They will be probabilities of each class

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \rightarrow \begin{bmatrix} e^{s_1} \\ e^{s_2} \\ e^{s_3} \end{bmatrix} \rightarrow \begin{bmatrix} e^{s_1}/(e^{s_1}+e^{s_2}+e^{s_3}) \\ e^{s_2}/(e^{s_1}+e^{s_2}+e^{s_3}) \\ e^{s_3}/(e^{s_1}+e^{s_2}+e^{s_3}) \end{bmatrix}$$

values turned to +ve

values turned to probabilities

example:-

$$\begin{bmatrix} 1 \\ -2 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 2.71 \\ 0.135 \\ 1.648 \end{bmatrix} \rightarrow \begin{bmatrix} 0.6 \\ 0.03 \\ 0.37 \end{bmatrix}$$

Soft max classifier.