⇒ Rate the OP :-

0 ⟶ 10

(Easy) (Difficult).

⇒

0 5 ⟶ 10

Easy moderate Difficult.

1. How a loss function is different from metric function? Write 3 metric functions with there working.

loss ⇓ Used to Update weights.

metric. ⇓ just to evaluate model.

mae, mse, accuracy, binary_ categorical_ A—, .com-:

2. What is maxpooling operation in Convnets? Explain.

⇓

Down Sample your image

60X60 ⇒ 30X30

(2,2).

↳ Reduces requirement of more time for training-:

3. Calculate categorical cross-entropy loss for y_true = [[0, 1, 0], [1, 0, 0], [0, 0, 1]], y_pred = [[0.1, 0.8, 0.1], [0.1, 0.1, 0.8], [0.1, 0.1, 0.8]]. *Hint: ln(0.8) = −0.2231, ln(0.1) = −2.3025.*

$$\text{loss} = -\sum y\text{-true} \ln y\text{-pred}$$

$$= -(1 \times \ln(0.8) + 1 \times \ln(0.1) + 1 \times \ln(0.8))$$

$$= -(-0.2231 - 2.3025 - 0.2231)$$

$$= 2.7487$$

0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0

Image Data

1	0	-1
1	0	-1
1	0	-1

Kernel

4. Do one convolution operation with the given kernel on the above image data. Given: **Stride = 2**

-3	3	0
-3	3	0
-3	3	0

training

testing Data

5. Define these terms: a) Temporal Leak b) Redundancy in data c) Optimisation d) Generalisation e) Normalisation of data.

mean = 0
SD = 1

t in seconds.

t = 0, 1, 2, 3, ...

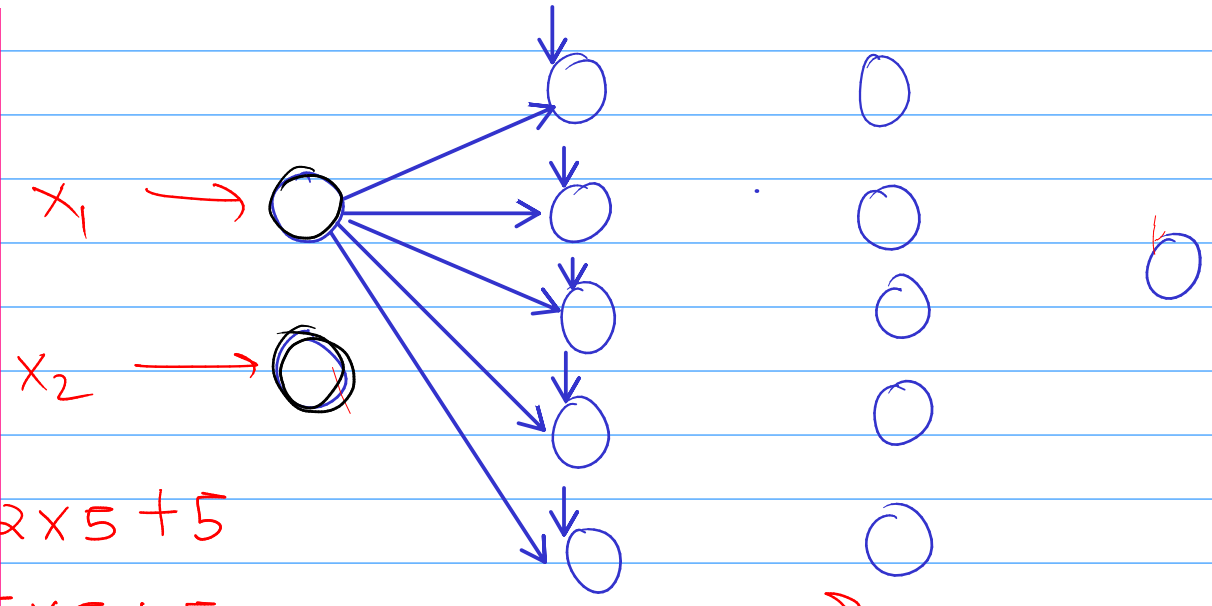
1, 3, 0, 2

1. Make a custom loss function which returns $(y\_true - y\_pred)^4$ as loss. Also write the compile statement for the model. *Hint : tf.pow(), tf.subtract()*

```
def custom_loss(y_true, y_pred):
    sub = tf.subtract(y_true, y_pred)
    return tf.pow(sub, 4)

model.compile(optimiser='rmsprop',
              loss = custom_loss,
              metrics=['mae'])
```

2. A dense neural network is formed in 2 – 5 – 5 – 1 fashion. Calculate the total number of trainable parameters (weights).



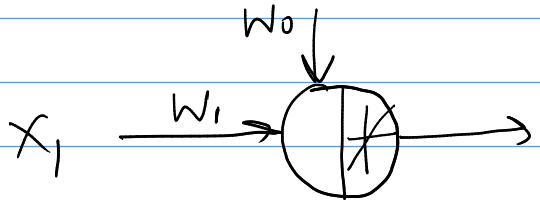
$2 \times 5 + 5$

$+ 5 \times 5 + 5$

$+ 1 \times 5 + 1$

51

3. A loss function depends upon two weights w0 and w1 as loss = w0 + w1. Find out the direction in which this loss can be minimised. *Hint: Gradient Descent*



$y = w_1 x_1 + w_0$

$0 - (w_1 x_1 + w_0)$

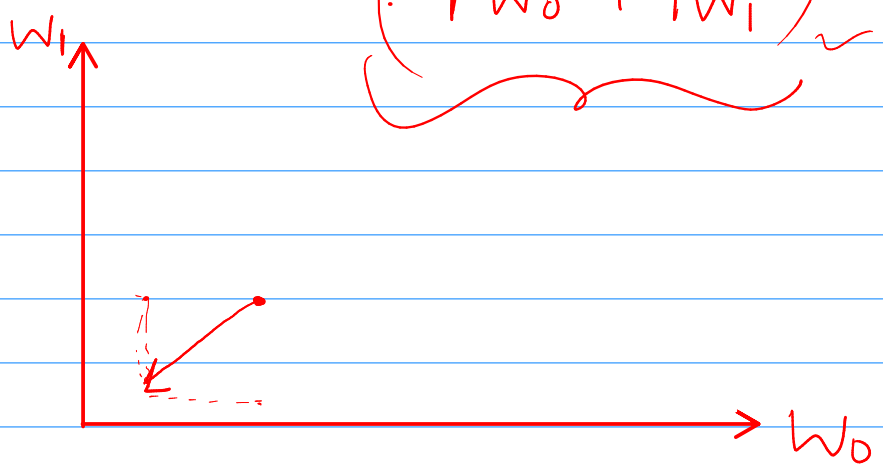
$x_1 = 1$

$loss = -(w_0 + w_1)$

direction

$\rightarrow -\left(\frac{\partial}{\partial w_0}\hat{w}_0 + \frac{\partial}{\partial w_1}\hat{w}_1 + \cdots\right) loss$

$\rightarrow -(1\hat{w}_0 + 1\hat{w}_1)$



$$w' = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

4. WiFi network strength was measured by changing distance from router (D), router antenna length (L) and manufacturer (M) (Nokia = 0, TP-Link = 1, D-Link = 2) and a table was prepared. Write a code/pseudo-code (Neural Network) for predicting Wifi network strength at some set of values (D', L', M').

D	L	M	WNS
1	10cm	0	−2dB
2	-	2	-
			⋮

Regression or Classification

Binary

Multi-Class.