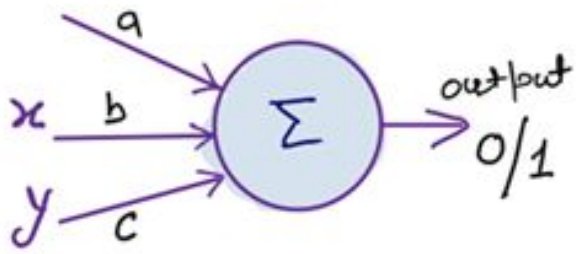
AIML- Cohort 1 -- C6
Compri MakeupExam
Max Marks : 30

Q1. Explain the following concepts in machine learning
[2*3=6Marks]
A. Xavier initialization.
B. Style GAN
C. Re-parameterization and its application

Answer:
Explanation using
A: Standard Definition
B: Standard Definition
C: Standard Definition

Q2.

Device weights a, b, c for the following neuron

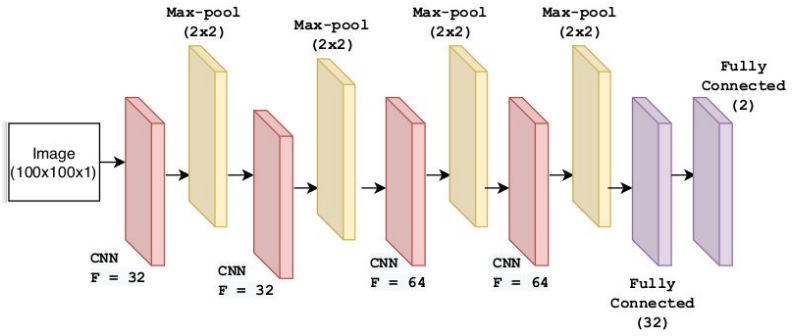


Such that it produces the following output.

X	Y	Output
0	0	1
0	1	1
1	0	0
1	1	0

Answer:
Equations are $a > 0$; $a+c>0$; $a+b<0$; $a+b+c<0$
Solution is: $a=1$; $c=1$; $b=-3$;

Q3. A user has collected a lot of 100x100 size images. Some of them contain images of mountains and other of airplanes. The issue is that the number of imges is large and the user keeps adding more and more images. He did not want to manually label image as mountain or airplanes. He has devised a neural network to do the task. All activation functions to be RelU except at the final layer, where it is softmax. Kernel size for the CNN is throughout kept 5x5. Network architecture is given below (Note: CNN and Max pool layers are shown in pink and yellow respectively. Blue ones are fully connected layers).
Determine the number of trainable parameters for this architecture? Assume bias term is also need to be added in a CNN and image size reduces after an convolution operation.



Answer:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (5, 5), activation='relu',input_shape=(100, 100, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(32, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(2))
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_16 (Conv2D)           (None, 96, 96, 32)        832
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 48, 48, 32)        0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 44, 44, 32)        25632
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 22, 22, 32)        0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 18, 18, 64)        51264
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 9, 9, 64)          0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 5, 5, 64)          102464
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 2, 2, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                8224
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 66
=================================================================
```

Total params: 188,482
Trainable params: 188,482
Non-trainable params: 0

Q4. Answer the below questions: **[1+2 = 3Marks]**

A. Explain weight decay.

B. How it leads to the regularization?

Answer:

A: Standard Definition

B: Standard Definition

Q5. Answer the following briefly. [1*3 = 3Marks]
A. What is the philosophical difference between GoogleLeNet and ResNet.
B. What was the biggest challenge in training a deeper network such as ResNet ?
C. How it was resolved in ResNet?

Answer:

A: GoogleLeNet aims for reducing computations and seeing things at multiple scales. However ResNet targets a simple architecture that is very deep and expects that the extralayers would automatically behave as identity mapping.
B: Biggest challenge that the standard solution of identity mappoint is just a single point in infinite space. So the chances of hitting the solution is very low.
C: ResNet applies skip connections to resolve the same.

Q6. Consider a random variable x, taking value in the range [1,6]. Consider two distributions, P and Q, as shown in the table below and answer the following questions: **[5+1 = 6Marks]**

	P(x)	Q(x)
x=1	0	0.3
x=2	0	0.1
x=3	0.2	0.2
x=4	0.3	0.1
x=5	0.1	0.3
x=6	0.4	0

A. Compute JS-Divergence between two distributions P and Q.
B. Without computation, comment on the possible value of KL-divergence of P and Q

Answer:
A: JSD is as above

x	P	Q	M=(P+Q)/2	log(P/M)	log(Q/M)	P*log(P/M)	Q*log(Q/M)
1	0	0.3	0.15	0	0.301029996	0	0.0903089987
2	0	0.1	0.05	0	0.301029996	0	0.0301029996
3	0.2	0.2	0.2	0	0	0	0
4	0.3	0.1	0.2	0.176091259	-0.301029996	0.0528273777	-0.030103
5	0.1	0.3	0.2	-0.301029996	0.176091259	-0.030103	0.0528273777
6	0.4	0	0.2	0.301029996	0	0.1204119983	0
						0.1431363764	**0.1431363764**
							0.14313638

NOTE: here log is at base 10. if someone takes log on another base (2 or e) they also will be awarded full marks if there are no calculation mistakes.

B: LK-divergence does not exist as neither P to Q not Q to P fully covers. So it is not possible to compute KLD.