CSE-425

Final - Section - 01

18101021

Ans. to Ques. No - 01

The design is suitable for pattern A and pattern B classification, but the main issue is the number of training samples, for a neural network to learn much more samples, it would need more samples, failing which is unable to provide the requisite output. Thus, no convolution layers are linear and any composition of linear layers is still linear. Previously, it is showed that the classes are not linearly seperable.

A complete answer includes mention of the whole neural network computing a linear function up to

18101021

the final non-linearity and the data being linearly inseparable. (+0.5) correct answer and partial justification (+0.5) correct justification.

As a result, there will be testing samples where the pattern does not match either A or B, and the neural network may provide a false output due to the lack of training.

18101021

Ans. to the Ques. No-02

In order to re-iterate parameter sharing is occurred when a feature map is created from the result of the convolution between input and a filter data from an unit within a plane in the convolution layer. All units within the layer plane share the same weights, hence it is called parameter/weight sharing. Convolutional layers are the layers where filters are applied to original image. To other feature maps ~~is~~ in a deep CNN.

1	-1	-1
-1	1	-1
-1	-1	-1

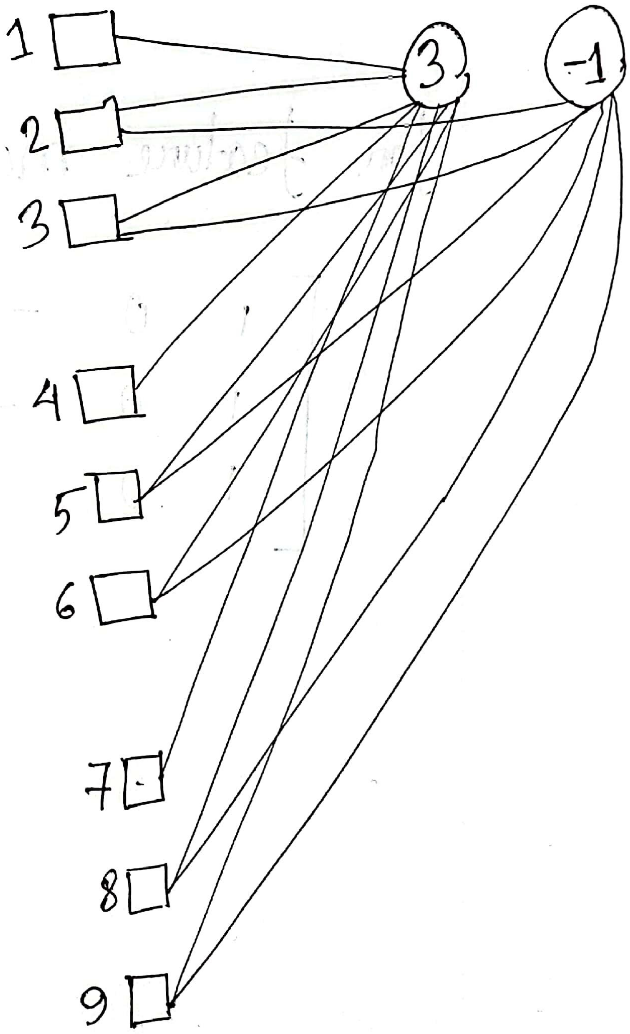
filter

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

6 x 6 image

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

4x4



$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ for black pixel and $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ for white pixel.

for feature maping in feature map-01

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad 3\times 3 \text{ filter 1}$$

for feature mapping in feature map-02

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad 3\times 3 \text{ filter 2}$$

Ans. to the Ques. No-03

The update gate in GRU refers to the preservation of past data for the future whereas the reset gate in GRU refers to the amount of data that will be ~~given~~ forgotten. GRU differs from LSTM in that it was created with a simple architecture. GRU has only two gates: update and reset but LSTM has three gates: forget, input and update. The GRU update gate is a single gate that combines the LSTM's forget and input gates. GRU does not have a cell state link LSTM and unlike LSTM does not have a reset or output gate. In GRU, the values are moved to the next stage, and the amount of old information or new information may be controlled more precisely than with LSTM.

Subsequently, GIU is ~~signifa~~ significantly faster than LSTM when a little amount of data is used than when a large amount of data is used. Because, it uses fewer parameters, fewer gates, has more gradient control capability, is less sophisticated and hence retains less memory, the cite is faster than the RNN. RNNs have feedback loops in its recurrent layer that allows them to keep a lot of memory. The gradient's loss function makes it difficult to train RNN to tackle challenges that take a long time and cause the gradient to worsen. RNN is slower than GRU due to the time necessary for back propagation, giving them a competative edge over RNN.