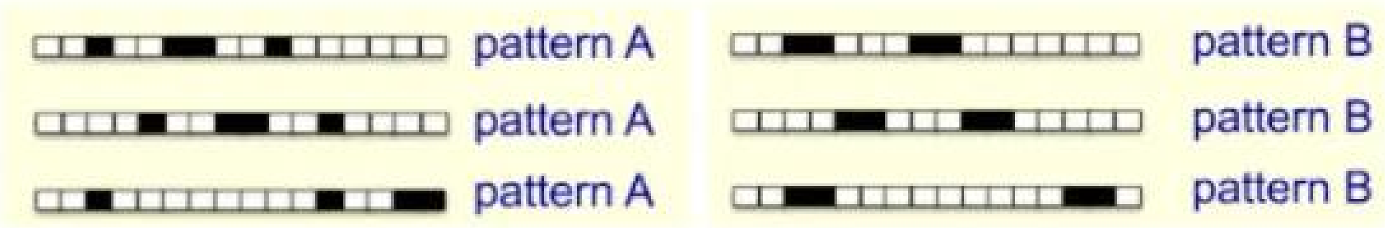
Question 1



Consider the above classification of patterns . The training set consists of patterns A and B in all possible translations. Consider a neural network that consists of a 1D convolution layer with a linear activation function, followed by a linear layer with a logistic output. Can such an architecture perfectly classify all of the training examples? Why or why not?

Add file

Question 2

Write the core difference of RNN ,Peephole-LSTM and GRU? What is the basic difference of LSTM forget gate and GRU's reset gate. Show the matrix workflow of the LSTM.

Add file

16201070

Ans: to the: Qus; No: 2

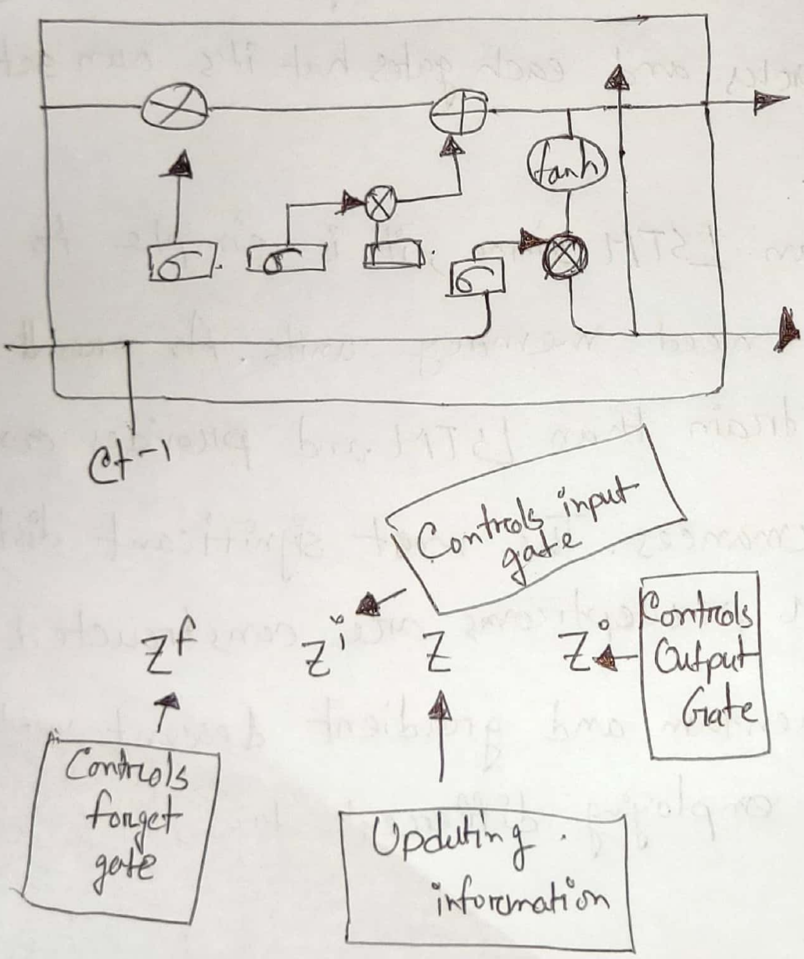
When compared to RNN, the workflow of GRU is quite similar, with the only variation being the activities performed inside the unite itself. Let's have a look at the ~~stu~~ structure of it. It's impor- ~~thant~~ ~~to~~ tant to remember that gates are nothing than neural networks and each gates hat it's own set weights and biases.

GRU is better than LSTM since, it is simple to alter and does not need memory units. As aresult it is quicker to train than LSTM and provides com- parable performances. The most significant distinc tion is that LSTM perceptrons are constructed utilizing the momentum and gradient descent metho rather than just employing different types of percep tions.

16201070

The main difference between the GRU and The LSTM is that the GRU has two portals (Reset and update) while LSTM has three gates that are input, output, forget.

Matrix Workflow of LSTM:



$$\boxed{z} = \boxed{W}\begin{bmatrix} x^t \\ h^t \end{bmatrix}$$

$$\boxed{z^i} = \sigma\left(\boxed{W^i}\begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$

$$\boxed{z^f} = \sigma\left(\boxed{W^f}\begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$

$$\boxed{z^o} = \sigma\left(\boxed{W^o}\begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$

Information flow of LSTM.

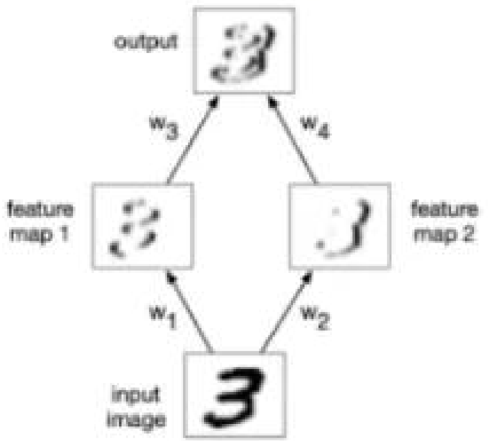
Question 3

If you have a 64*64 binary image at input in a CNN network with 7 filters(size of 5*5) stride of 2 and no padding of 0 and apply 3 sets of Conv and max pool(size of 2*2) what will be the number of nodes in the flattening layer? Show each steps after conv and max pool layers happen.

Add file

Question 4

Explain the shared concept of CNN? You will design a convolutional network to detect vertical boundaries in an image. The architecture of the network is as shown below.



The ReLU activation function is applied to the first convolution layer. The output layer uses the linear activation function. Design two convolution filters for the rst layer, of size 3 x 3. One of them should detect dark/light boundaries, and the other should detect light/dark boundaries.

Add file

16201070

Ans! to the! Qus! No! 3

$$\text{Image size } \overset{w}{64} \times \overset{h}{64}$$

no padding of 0 means $P=0$;

filters size $F = 7$ 5×5

stride $= 2$ max pooling (2×2)

We know that, After convolutional layer we, find

$$\left[\frac{W+2P-F}{S}\right]+1$$

and $$\left[\frac{H+2P-F}{S}\right]+1.$$

Set 1,

$$\frac{64+(2\times0)-5}{2}+1 = 30;$$

$$H = \frac{64+2\times0-5}{2}+1 = 30$$

first tensor $= 7$ layers of 30×30 Conv tensor

max pooling 7 layers $\left(\frac{30}{2}\right) = (15 \times 15)$

Set 2,

$$\left[\frac{15+(2\times0)-5}{2}\right]+1 = 6.$$

$$\left[\frac{15+(2\times0)-5}{2}\right]+1 = 6.$$

Second tensor 7 layers of 6×6 Conv tensor

16201070

max pooling $\left(\frac{6}{2}\right) = (3 \times 3)$

7 layers of 3X3.

Ans: to the : Qu. No: 4 16201070

Shared Concept of CNN :

To reiterate parameter sharing occurs when a future map is generated from the result of the convolution between a filter and input data from a unit within a plane ~~to~~ in the conv layer. All units within this layer plane share the same weights ; hence it is called weight / parameter sharing. Convolutional layers are the layers where filters are applied to original image, or to other feature maps in a deep CNN.

16201070

Two convolutional filters:

$$W_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Scanned with CamScanner