Ans to the Question No 1

Here I have created a CNN model where I implemented three different types of designs.

1st observation, with the optimizer=Adam

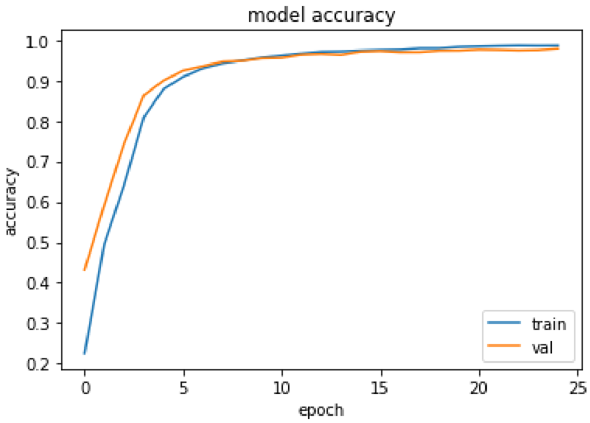
epochs=25,
batch size=500 and
the optimizer=Adam.

But the good performance is shown for regular CNN and inverted CNN, but hour-glass CNN is not performed well.

Regular CNN:

Train Accuracy= 0.9785

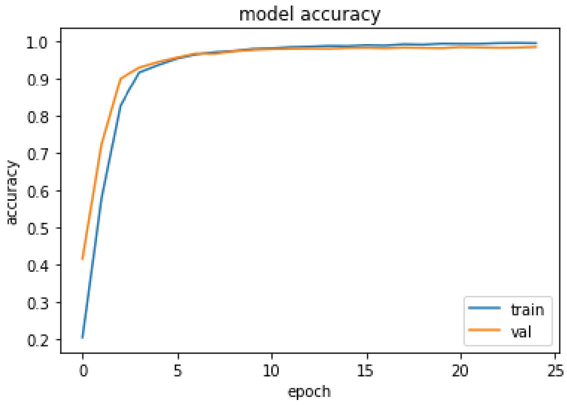
Test Accuracy=0.9800999760627747



Inverted CNN:

Train Accuracy=0.9835

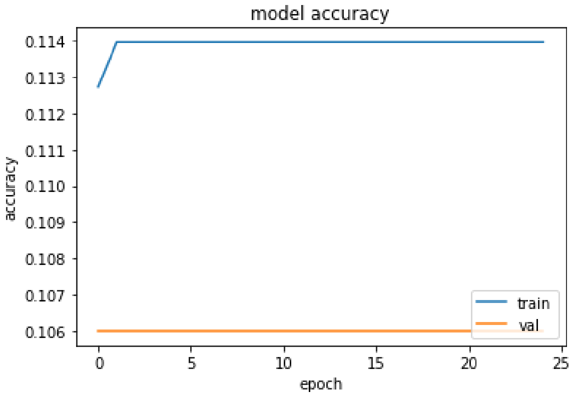
Test Accuracy=0.9843000173568726



Hour Glass CNN:

Train Accuracy=0.1060

Test Accuracy=0.11349999904632568



We can see here that the accuracy is very poor for Hourglass when the optimizer is Adam.

2nd Observation with RMSProp:

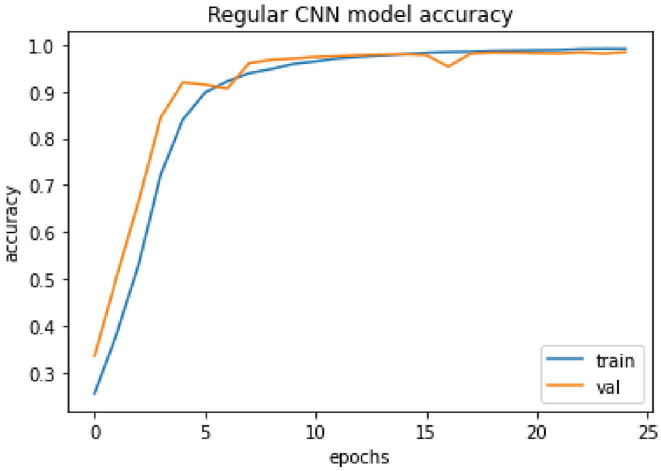
epochs=25,
batch size=500 and
the optimizer=RMSProp.

But the good performance is shown for regular CNN and inverted CNN, but hour-glass CNN is not performed well.

Regular CNN:

Train Accuracy=0.9845

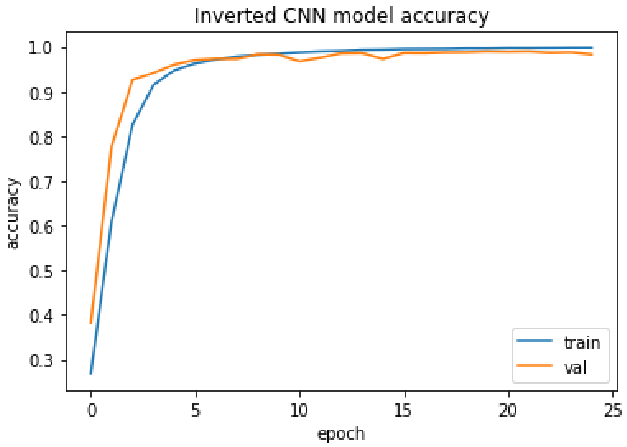
Test Accuracy=0.98580002784729



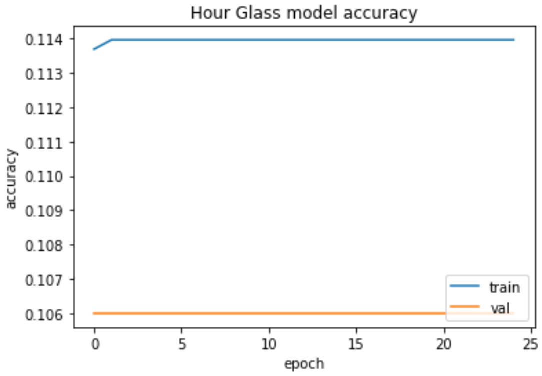
Inverted CNN:

Train Accuracy=0.9867

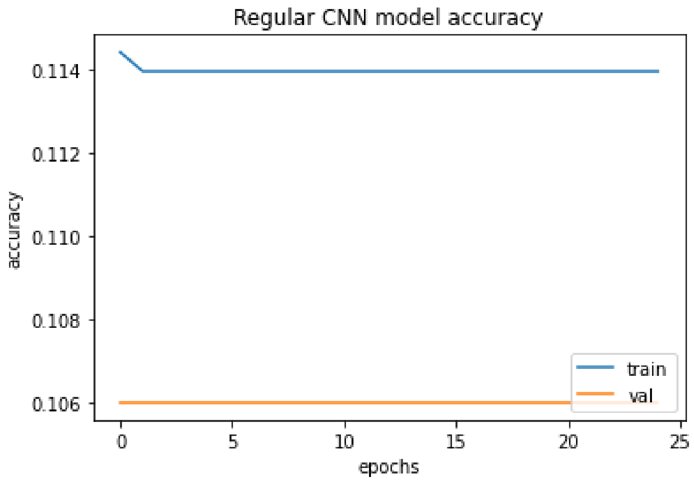
Test Accuracy=0.9866999983787537



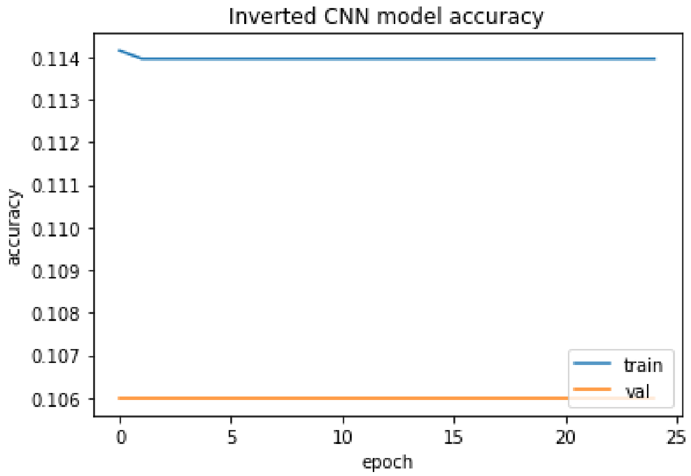
Hourglass CNN:
Train Accuracy=0.1135
Test Accuracy=0.11349999904632568



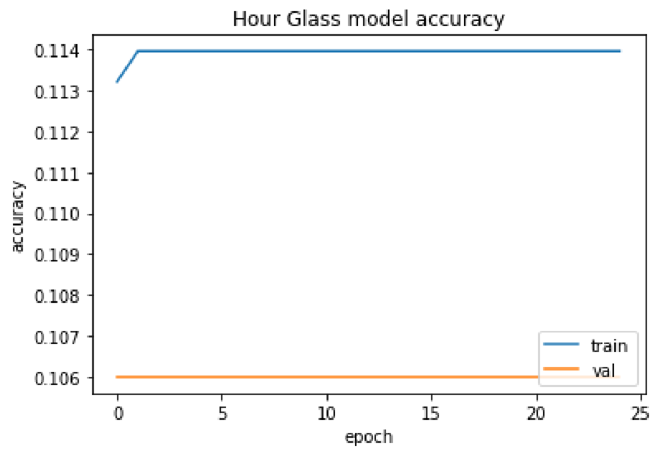
3rd Observation with SGD:
epochs=25,
batch size=500 and
the optimizer=RMSProp.
But the good performance is shown for regular CNN and inverted CNN, but hour-glass CNN is not performed well.
Regular CNN:
Train Accuracy=0.1140
Test Accuracy=0.11349999904632568



Inverted CNN:
Train Accuracy=0.1140
Test Accuracy=0.11349999904632568



Hourglass CNN:
Train Accuracy=0.1140
Test Accuracy=0.11349999904632568

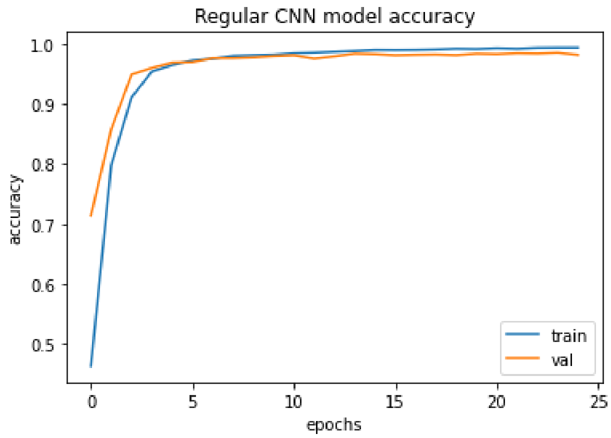


Here in SGD all the designs perform poorly for batch size 500, Lr=.001.

2. Changing Batch Size=128
Optimizer=Adam

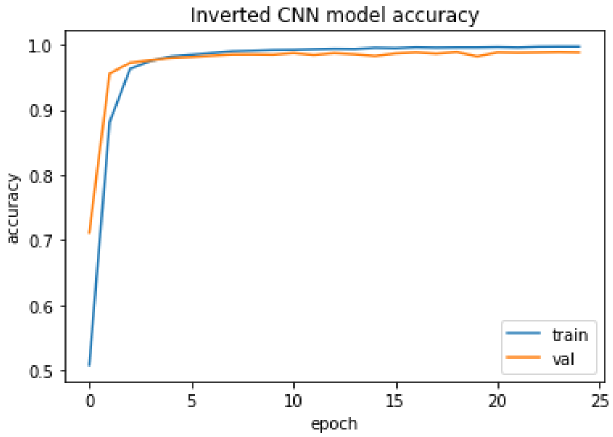
<u>For Regular CNN:</u>

Train accuracy: 0.9943
Test Accuracy: 0.9853000044822693



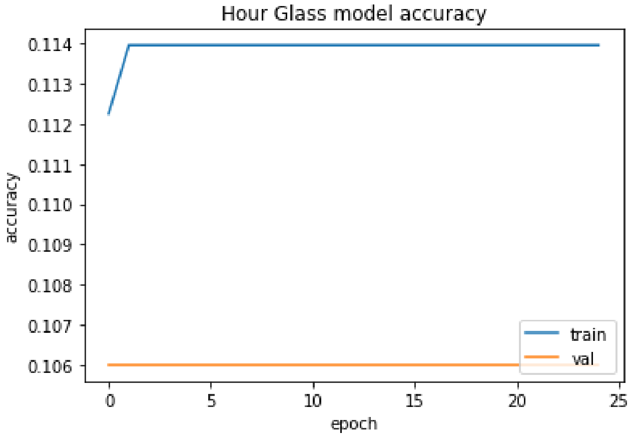
<u>For Inverted CNN:</u>

Train accuracy: 0.9971
Test Accuracy: 0.9890000224113464



<u>For Hourglass CNN:</u>

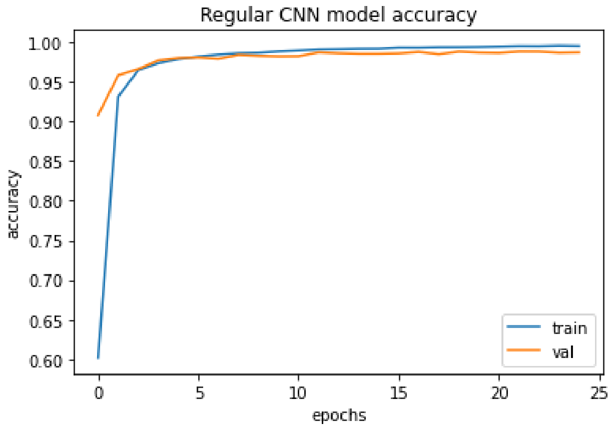
Train accuracy: 0.1140
Test Accuracy: 0.11349999904632568



Changing Batch Size=64
Optimizer=Adam

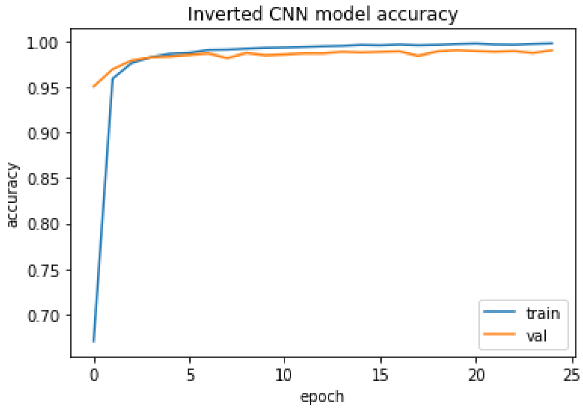
<u>For Regular CNN:</u>

Train accuracy: 0.9951
Test Accuracy: 0.9890000224113464



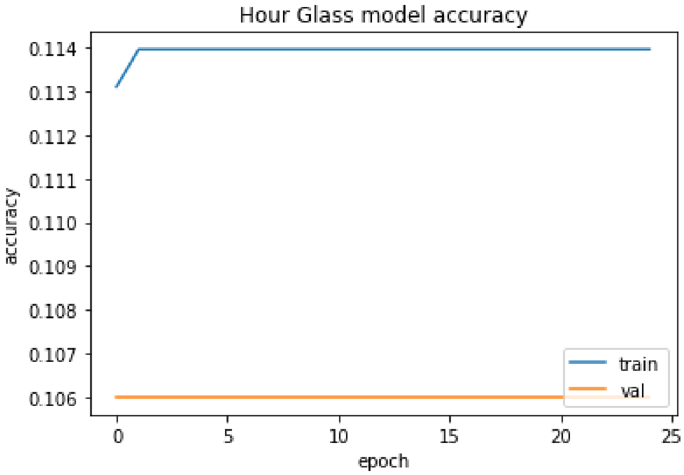
For Inverted CNN:

Train accuracy: 0.9979
Test Accuracy: 0.9902999997138977



For Hourglass CNN:

Train accuracy: 0.1140
Test Accuracy: 0.11349999904632568



Here we can see as we are decreasing the batch size, the performance of the regular CNN and Inverted CNN improved, for the batch size =64, inverted RNN performed about 99% of accuracy for both the training and test data.

Ans to the Question No 2

Scenario1: Changes in Batch Size

Batch Size=64, epochs=25, learning rate=.001
Training accuracy: 0.8172
Test accuracy: 0.6039

Batch Size=128, epochs=25, learning rate=.001
Training accuracy: 0.8992
Test accuracy: 0.5812

Batch Size=256, epochs=25, learning rate=.001
Training accuracy: 0.9515
Test accuracy: 0.5729

Scenario 2: Changes in learning rate

Batch Size=64, epochs=25, learning rate=.001
Training accuracy: 0.8172
Test accuracy: 0.6039

Batch Size=64, epochs=25, learning rate=.005
Training accuracy: 0.0984
Test accuracy: 0.1000

Batch Size=64, epochs=25, learning rate=.01
Training accuracy: 0.0994

Test accuracy: 0.1000

So we can see that the less the learning rate is, the accuracy is better, but when I was trying to increase the learning rate it showed bad results. Similarly, I tried 3 different batch sizes as 64, 128, 256. But when I was increasing the batch size it was doing well on training data but the performance on test data was decreasing.

Building Feed Forward Network

a) The Feed forward Network took more time compared to the LeNet model. It shows poor accuracy. The accuracy for FFN is only 0.2555 for the training set and for the testing set it is only 0.2554999887943268.

b) The number of parameters for Lenet is: 697,046
On the other hand the number of parameters for Feed forward Network: 119,292,650
As we can see the Feed forward Network took more time and parameters, it is not worthy.

Ans. to the Q. No-03

① From the given matrix input (X) we can see the dimension is $6 \times 6 \times 1$. The dimension of kernel is 3×3,

∴ Each filter has $3 \times 3 \times 1 + 1 = 10$ parameters.

For 1 filter here there are 10 params.

② Given that,

$$X = \begin{bmatrix} 7 & 5 & 0 & 0 & 3 & 2 \\ 6 & 4 & 5 & 1 & 4 & 8 \\ 9 & 0 & 2 & 2 & 5 & 4 \\ 6 & 3 & 4 & 7 & 9 & 8 \\ 5 & 7 & 5 & 6 & 9 & 0 \\ 7 & 9 & 0 & 8 & 2 & 3 \end{bmatrix}, \quad f = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Taking stride $= 1$.

$$\begin{bmatrix} 7 & 5 & 0 \\ 6 & 4 & 5 \\ 9 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{aligned} &7 \times 1 + 5.0 + 0(-1) + 6.2 + \\ = &\ 4.0 + 5(-2) + 9.1 + 0.0 \\ &+ 2(-1) = 16 \end{aligned}$$

$$\begin{bmatrix} 5 & 0 & 0 \\ 4 & 5 & 1 \\ 0 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{aligned} &5.1 + 0.0 + 0(-1) + 4.2 \\ = &+ 5.0 + 1(-2) + 0.1 + \\ &2.0 + 2(-1) = 9 \end{aligned}$$

$$\begin{bmatrix} 0 & 0 & 3 \\ 5 & 1 & 4 \\ 2 & 2 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{aligned} &0.1 + 0.0 + 3(-1) + 5.2 \\ = &+ 1.0 + 4(-2) + 2.1 \\ &+ 2.0 + 5(-1) = -4 \end{aligned}$$

$$\begin{bmatrix} 0 & 3 & 2 \\ 1 & 4 & 8 \\ 2 & 5 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{aligned} &0.1 + 3.0 + 2(-1) + 1.2 \\ = &+ 4.0 + 8(-2) + 2.1 + \\ &5.0 + 4(-1) = -18 \end{aligned}$$

$$\begin{bmatrix} 6 & 4 & 5 \\ 9 & 0 & 2 \\ 6 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{aligned} &6.1 + 4.0 + 5(-1) + 9.2 \\ = &+ 0.0 + 2(-2) + 6.1 + \\ &3.0 + 4(-1) = 17 \end{aligned}$$

$$\begin{bmatrix} 4 & 5 & 1 \\ 0 & 2 & 2 \\ 3 & 4 & 7 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{aligned} &4.1 + 5.0 + 1(-1) + \\ = &\ 0.2 + 2.0 + 2(-2) \\ &+ 3.1 + 4.0 + 7(-1) = -5 \end{aligned}$$

$$\begin{bmatrix}5&1&4\\2&2&5\\4&7&9\end{bmatrix}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix} = \begin{aligned}&5\cdot1+1\cdot0+4(-1)\\&+2\cdot2+2\cdot0+5(-2)\\&+4\cdot1+7\cdot0+9(-1)\end{aligned}$$

$$= -10.$$

$$\begin{bmatrix}1&4&8\\2&5&4\\7&9&8\end{bmatrix}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix} = \begin{aligned}&1\cdot1+4\cdot0+8(-1)+2\cdot2\\&+5\cdot0+4(-2)+7\cdot1+\\&9\cdot0+8(-1) = -12\end{aligned}$$

$$\begin{bmatrix}9&0&2\\6&3&4\\5&7&5\end{bmatrix}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix} = \begin{aligned}&9\cdot1+0\cdot0+2(-1)+6\cdot2\\&+3\cdot0+4(-2)+5\cdot1+\\&7\cdot0+5(-1) = 11\end{aligned}$$

$$\begin{bmatrix}0&2&2\\3&4&7\\7&5&6\end{bmatrix}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix} = \begin{aligned}&0\cdot1+2\cdot0+2(-1)+3\cdot2+4\cdot0\\&+7(-2)+7\cdot1+5\cdot0+\\&6(-1) = -9\end{aligned}$$

$$\begin{bmatrix}2&2&5\\4&7&9\\5&6&9\end{bmatrix}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix} = \begin{aligned}&2\cdot1+2\cdot0+5(-1)+4\cdot2\\&+7\cdot0+9(-2)+5\cdot1+\\&6\cdot0+9(-1) = -17\end{aligned}$$

$$\begin{bmatrix} 2 & 5 & 4 \\ 7 & 9 & 8 \\ 6 & 9 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = 2\cdot 1 + 5\cdot 0 + 4(-1) + 7\cdot 2 + 9\cdot 0 + 8(-2) + 6\cdot 1 + 9\cdot 0 + 0(-1) = 2$$

$$\begin{bmatrix} 6 & 3 & 4 \\ 5 & 7 & 5 \\ 7 & 9 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = 6\cdot 1 + 3\cdot 0 + 4(-1) + 5\cdot 2 + 7\cdot 0 + 5(-2) + 7\cdot 1 + 9\cdot 0 + 0(-1) = 9$$

$$\begin{bmatrix} 3 & 4 & 7 \\ 7 & 5 & 6 \\ 9 & 0 & 8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = 3\cdot 1 + 4\cdot 0 + 7(-1) + 7\cdot 2 + 5\cdot 0 + 6(-2) + 9\cdot 1 + 0\cdot 0 + 8(-1) = -1$$

$$\begin{bmatrix} 4 & 7 & 9 \\ 5 & 6 & 9 \\ 0 & 8 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = 4\cdot 1 + 7\cdot 0 + 9(-1) + 5\cdot 2 + 6\cdot 0 + 9(-2) + 0\cdot 1 + 8\cdot 0 + 2(-1) = -15$$

$$\begin{bmatrix} 7 & 9 & 8 \\ 6 & 9 & 0 \\ 8 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = 7\cdot 1 + 9\cdot 0 + 8(-1) + 6\cdot 2 + 9\cdot 0 + 0(-2) + 8\cdot 1 + 2\cdot 0 + 3(-1) = 16$$

The output activation map

16	9	-4	-18
17	-5	-10	-12
11	-9	-17	2
9	-1	-15	16

③ Max pooling operation :-

For pooling size 2 × 2, with stride 2

16	9
17	-5

Here max value = 17

-4	-18
-10	-12

max value = -4

11	-9
9	-1

max value = 11

and

-17	2
-15	16

max value = 16

After applying the max pooling operation assigning the maximum value we get the ~~ab~~ output below –

17	-4
11	16