

COMPUTER VISION

Project-2

Rakesh Kumar Mahato

RKM190000

Network Model:

0	Input image (28x28x1)
1	Maxpool (4x4)
2	Convolution Layer Filter # = 16, Size = 3x3 Padding = same
3	ReLU Activation
4	Convolution Layer Filter # = 32, Size = 3x3 Padding = same
5	ReLU Activation
6	Maxpool (2x2)
7	Flatten
8	Dense Layer (512 nodes)
9	Dense Layer (10 nodes)
10	Softmax

Experiments:

		Training Accuracy	Testing Accuracy
1	Initial Test	0.7161	0.7303
2	Epoch increased 5 to 10	0.7525	0.7461
3	Epoch increased 10 to 20	0.7695	0.7722
4	Epoch increased 20 to 50	0.8578	0.7631
5	Epoch = 20, # Filter =10	0.7768	0.7751
6	Epoch = 20, # Filter =16	0.8069	0.78
7	Epoch = 20, # Filter =16, filter size 3x3 to 2x2	0.7795	0.7686
8	Epoch = 20, # Filter =16, filter size 3x3, maxpool after conv removed	0.8631	0.7875
9	Epoch = 20, # Filter =16, filter size 3x3, maxpool after conv removed Another Convolution layer added (#filter = 32)	0.9245	0.7873
10	Epoch = 20, # Filter =17, filter size 3x3, Maxpool added after 2nd Convolution Layer	0.8595	0.8009
11	Tried to increase epoch to 50	0.9685	0.7801
12	Added 1 more convolution layer before maxpool (#filter = 64)	0.9351	0.7925

Result:

Best experiment was # 10

```
tf.keras.layers.MaxPool2D(4, 4, input_shape=(28, 28, 1)),  
tf.keras.layers.Conv2D(16, (3, 3), padding='same', activation=tf.nn.relu),  
tf.keras.layers.Conv2D(32, (3, 3), padding='same', activation=tf.nn.relu),  
tf.keras.layers.MaxPool2D(2, 2),  
tf.keras.layers.Flatten(),  
tf.keras.layers.Dense(512, activation=tf.nn.relu),  
tf.keras.layers.Dense(10, activation=tf.nn.softmax)
```