Chapter 3

Network Visualization

In this section we introduce three ways to visualize the outputs in a neural network. Such visualization especially in a CNN network helps to look at what sort of learning is happening and what portion is the network looking to produce the output. The three visualizations that we are performing here are -

1. **Visualizing Intermediate Layer Activations** - This visualization helps to take a look at the different images from Convolution layers filters,and see how different filters in different layers activate different parts of the image.
   In deep networks we can observe how initial layers tend to visualize general patterns like edges, texture, background etc, while layer at the end of the network visualize more training data specific feature.

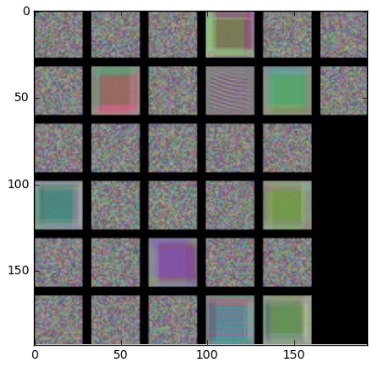
2. **Visualizing Convnet Filters** - In this visualizations ,we observe how different filters are learned along the network.We can observe filters as images by running Gradient Descent on the value of a convnet maximizing the response of a specific filter, starting from a blank input image.

3. **Visualizing Heatmaps of class activations** -In this visulization we produce heatmaps of class activations over input images.A class activation map is a basically a 2D grid of scores for a particular output class for each location in the image.
   CAM can be performed in many ways,here we have implemented Grad-CAM(Gradient-weighted Class Activation Mapping) that uses class-specific gradient information flowing into the final convolutional layer of a CNN and produces a localization map of the important regions in the image.

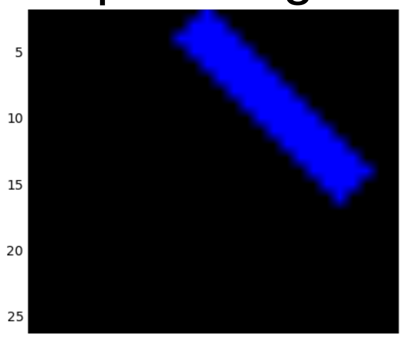
Base model over Line data

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 22, 22, 32)        4736
_________________________________________________________________
batch_normalization_2 (Batch (None, 22, 22, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 11, 11, 32)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 3872)              0
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              3965952
_________________________________________________________________
dense_4 (Dense)              (None, 96)                98400
=================================================================
Total params: 4,069,216
Trainable params: 4,069,152
Non-trainable params: 64
```

Convolutional Filters(conv2d_2)

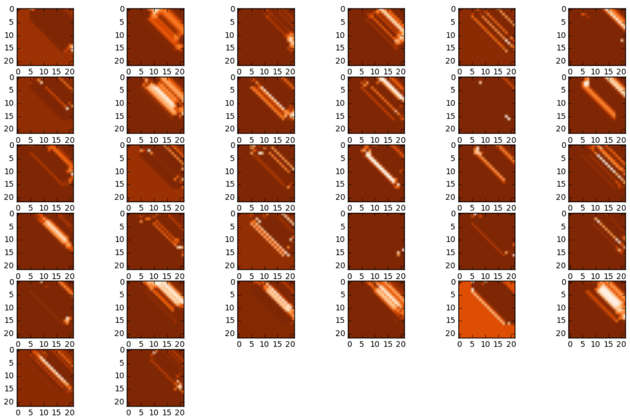


Input Image

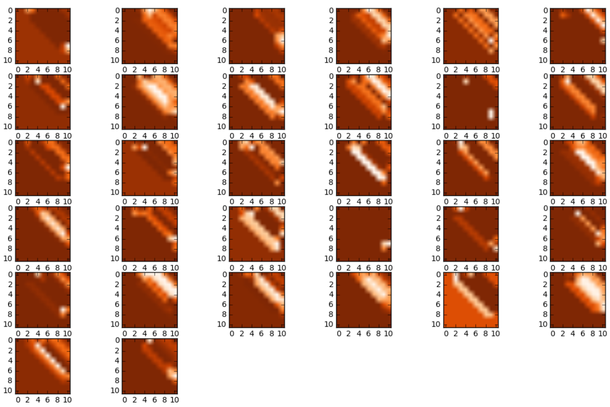


Intermediate Layer Activation

Conv2D(conv2d_2)

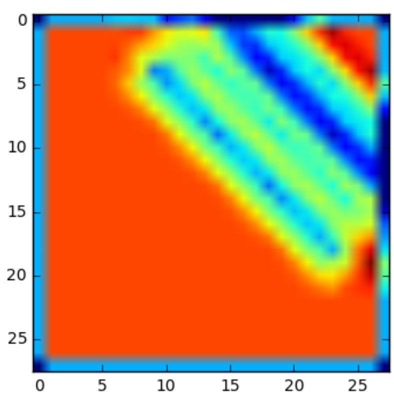


MaxPool2D

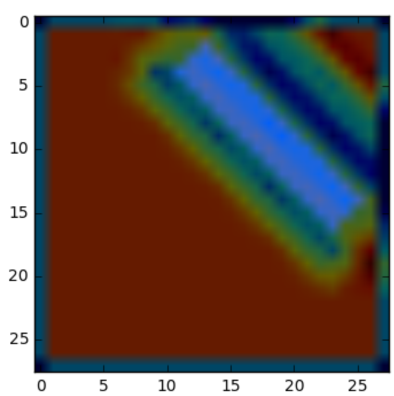


Class Activation Heatmap

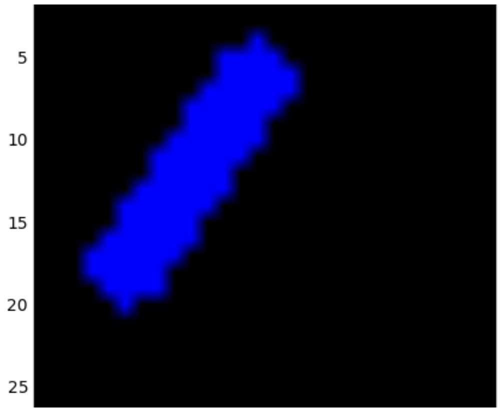
Heatmap



Heatmap over Input Image

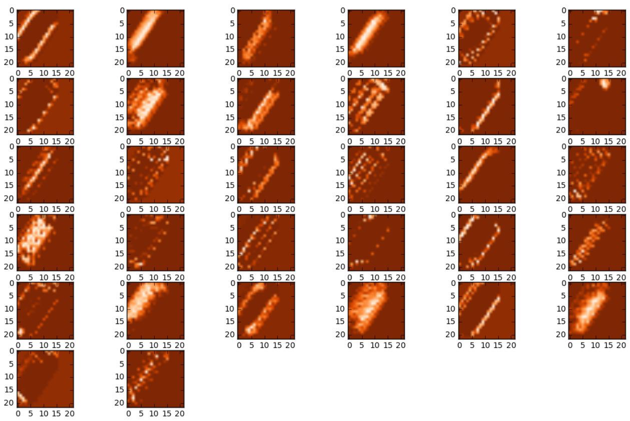


Input Image

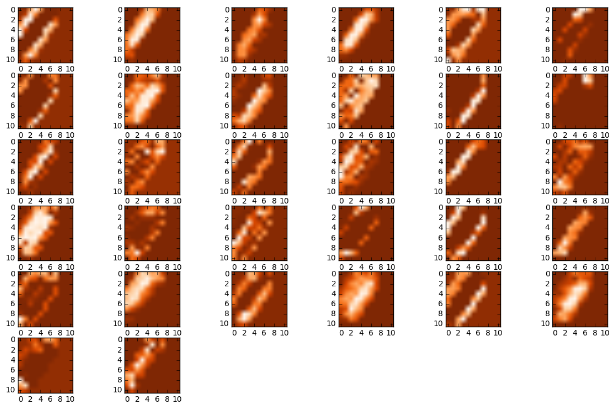


Intermediate Layer Activation

Conv2D(conv_2d)

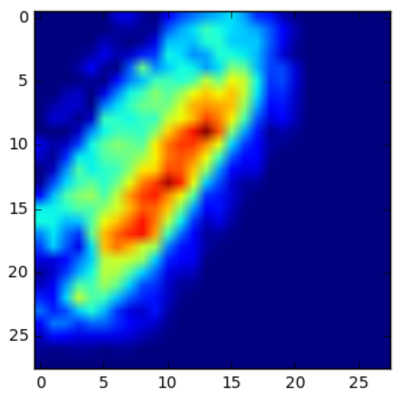


MaxPool2D

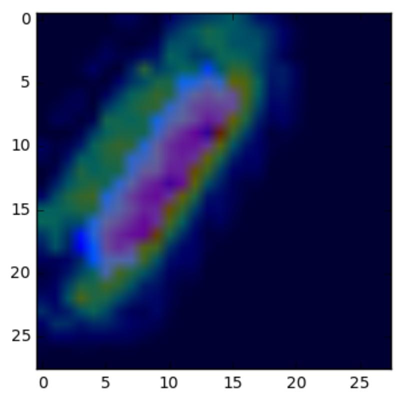


Class Activation Heatmap

Heatmap



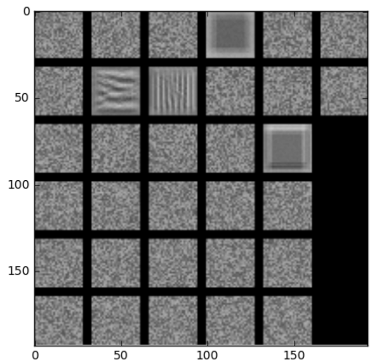
Heatmap over Input Image



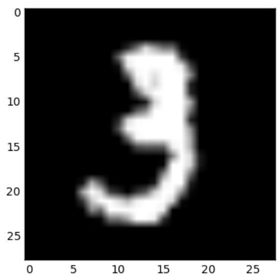
Base Model over MNIST data

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 22, 22, 32)        1600
_________________________________________________________________
batch_normalization_2 (Batch (None, 22, 22, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 11, 11, 32)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 3872)              0
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              3965952
_________________________________________________________________
dense_4 (Dense)              (None, 10)                10250
=================================================================
Total params: 3,977,930
Trainable params: 3,977,866
Non-trainable params: 64
```

Convolutional Filters(conv2d_2)

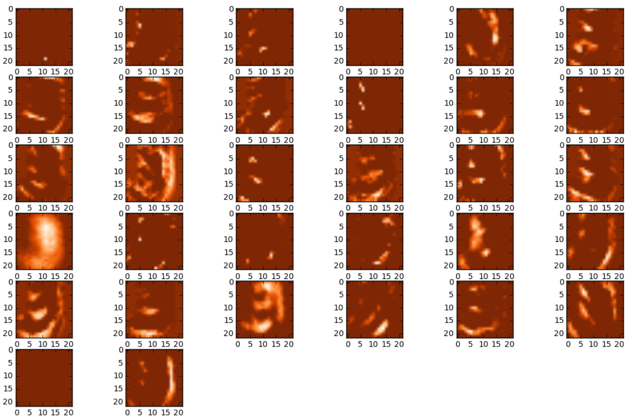


Input Image

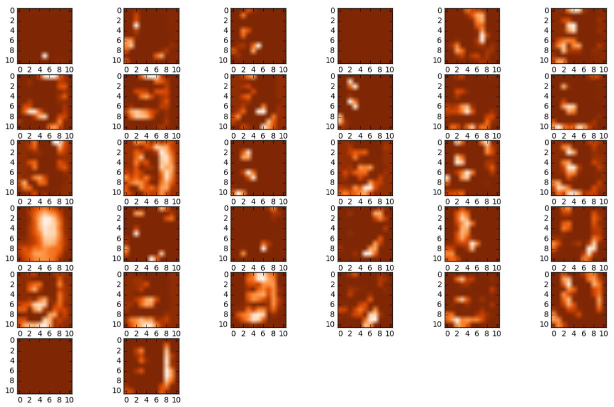


Intermediate Layer Activation

Conv2D(conv2d_2)

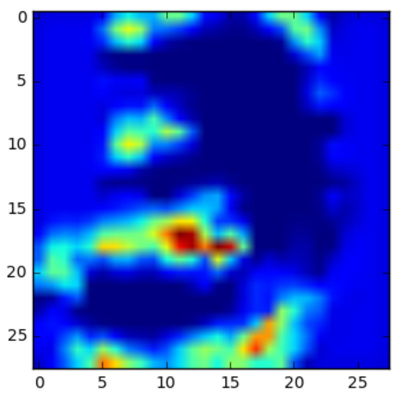


MaxPool2D

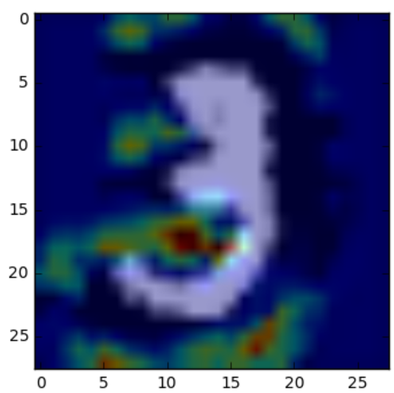


Class Activation Heatmap

Heatmap



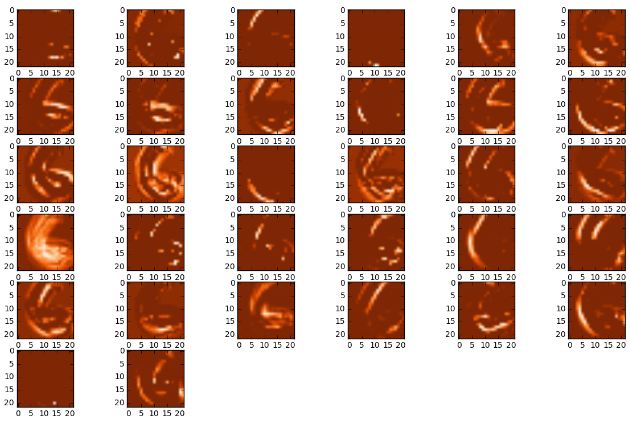
Heatmap over Input Image



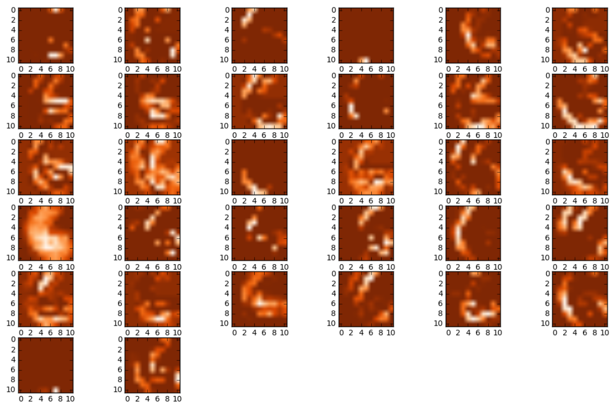
Input Image

Intermediate Layer Activation

Conv2D(conv2d_2)

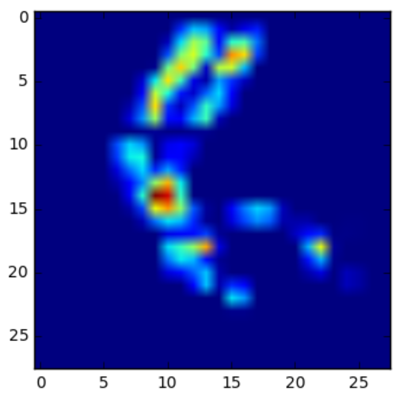


MaxPool2D

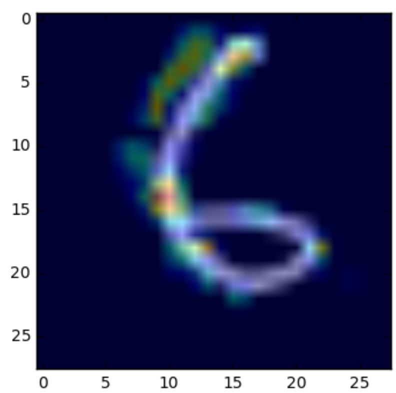


Class Activation Heatmap

Heatmap



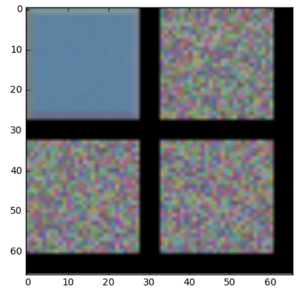
Heatmap over Input Image



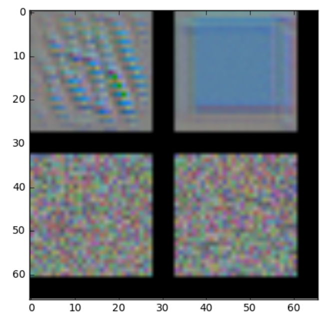
Question 2 - Variation 1 mode

Layer (type)	Output Shape	Param #	Connected to
Input_Image (InputLayer)	(None, 28, 28, 3)	0	
conv2d_1 (Conv2D)	(None, 26, 26, 4)	112	Input_Image[0][0]
conv2d_2 (Conv2D)	(None, 22, 22, 4)	404	conv2d_1[0][0]
flatten_1 (Flatten)	(None, 1936)	0	conv2d_2[0][0]
dense_1 (Dense)	(None, 64)	123968	flatten_1[0][0]
dense_2 (Dense)	(None, 64)	123968	flatten_1[0][0]
dense_3 (Dense)	(None, 64)	123968	flatten_1[0][0]
dense_4 (Dense)	(None, 64)	123968	flatten_1[0][0]

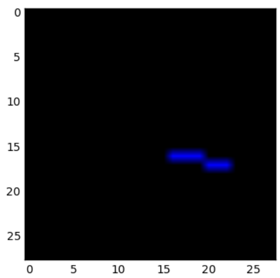
Convolutional Filters(conv2d_1)



Convolutional Filters(conv2d_2)

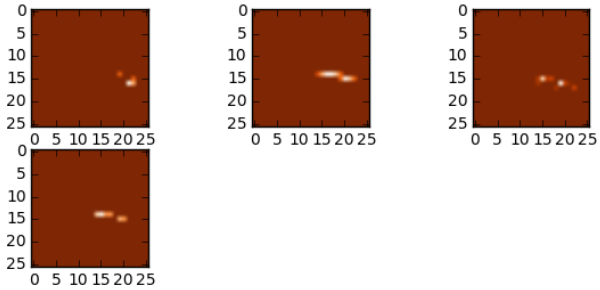


Input Image

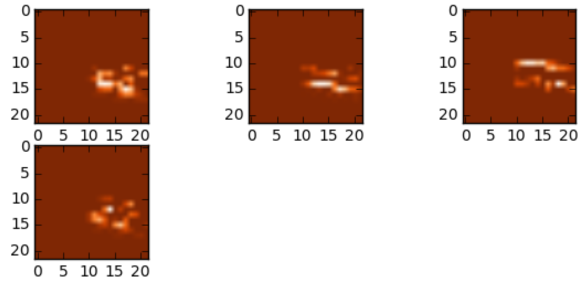


Intermediate Layer Activation

Conv2D(conv2d_1)

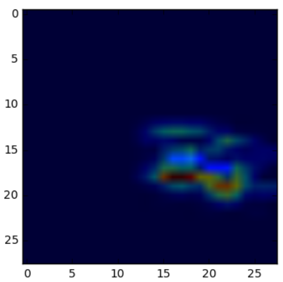


Conv2D(conv2d_2)

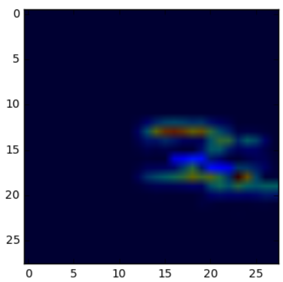


Class Activation Heatmap

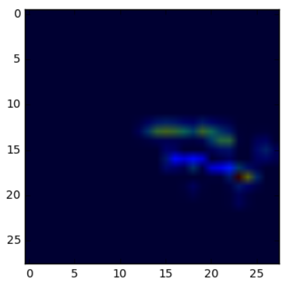
Length



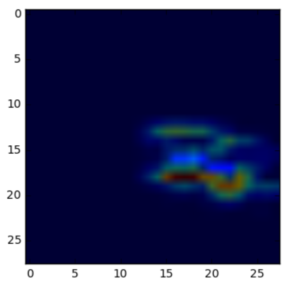
Width



Color



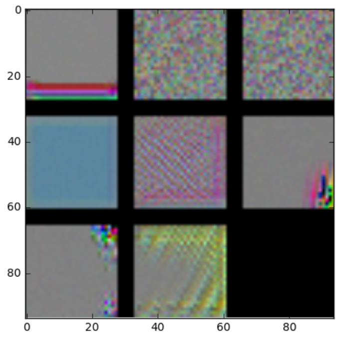
Angle



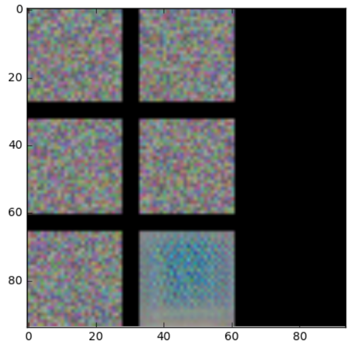
Question 2 - Variation 2 model

```
Layer (type)                     Output Shape         Param #     Connected to
====================================================================================
Input_Image (InputLayer)         (None, 28, 28, 3)    0
conv2d_1 (Conv2D)                (None, 28, 28, 8)    1184        Input_Image[0][0]
max_pooling2d_1 (MaxPooling2D)   (None, 14, 14, 8)    0           conv2d_1[0][0]
conv2d_2 (Conv2D)                (None, 10, 10, 6)    1206        max_pooling2d_1[0][0]
max_pooling2d_2 (MaxPooling2D)   (None, 9, 9, 6)      0           conv2d_2[0][0]
flatten_1 (Flatten)              (None, 486)          0           max_pooling2d_2[0][0]
dense_1 (Dense)                  (None, 128)          62336       flatten_1[0][0]
dense_5 (Dense)                  (None, 256)          124672      flatten_1[0][0]
```

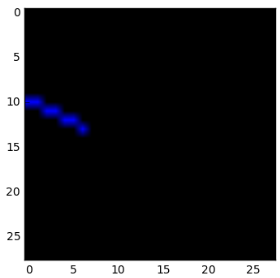
Convolutional Filters(conv2d_1)



Convolutional Filters(conv2d_2)

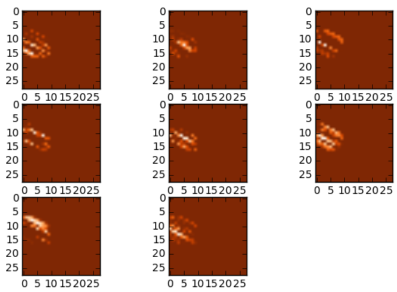


Input Image

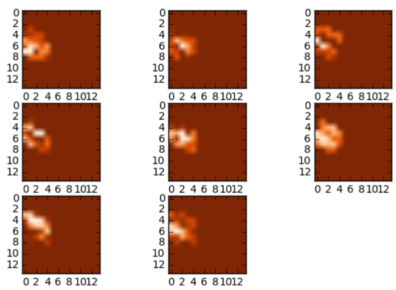


Intermediate Layer Activation

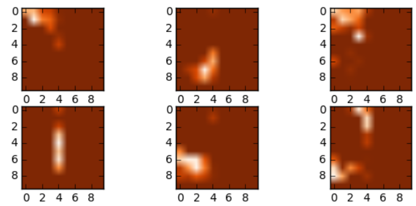
Conv2D(conv2d_1)



MaxPool2D

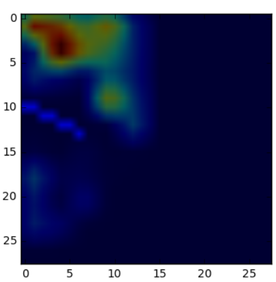


Conv2D(conv2d_2)

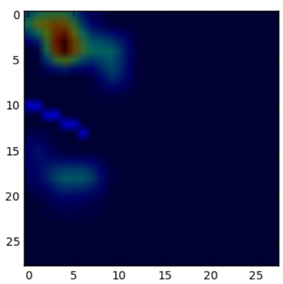


Class Activation Heatmap

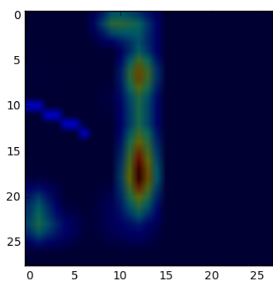
Length



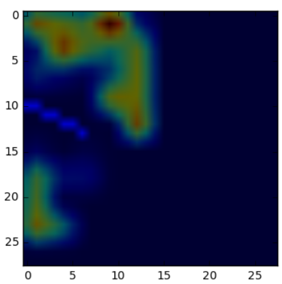
Width



Color



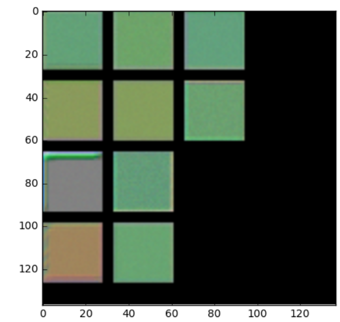
Angle



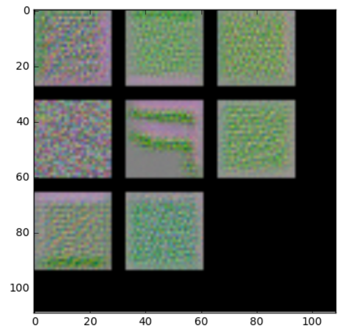
Question 2 - Variation 3 model

Layer (type)	Output Shape	Param #	Connected to
Input_Image (InputLayer)	(None, 28, 28, 3)	0	
conv2d_1 (Conv2D)	(None, 28, 28, 10)	760	Input_Image[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 10)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 12, 12, 8)	728	max_pooling2d_1[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 8)	0	conv2d_2[0][0]
flatten_1 (Flatten)	(None, 288)	0	max_pooling2d_2[0][0]
dense_1 (Dense)	(None, 128)	36992	flatten_1[0][0]
dense_3 (Dense)	(None, 64)	18496	flatten_1[0][0]

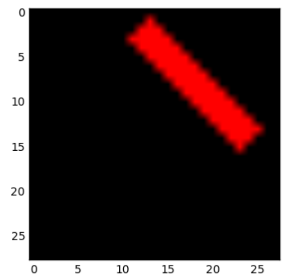
Convolutional Filters(conv2d_1)



Convolutional Filters(conv2d_2)

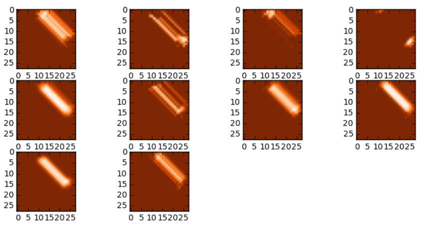


Input Image

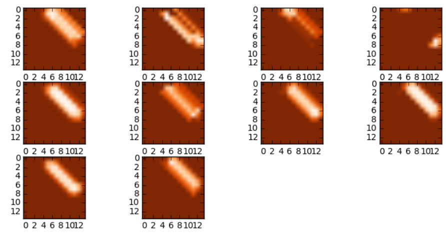


Intermediate Layer Activation

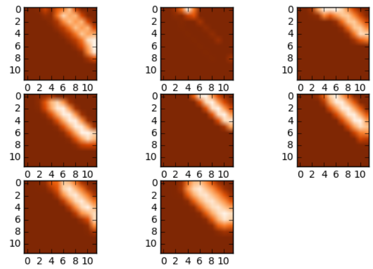
Conv2D(conv2d_1)



MaxPool2D

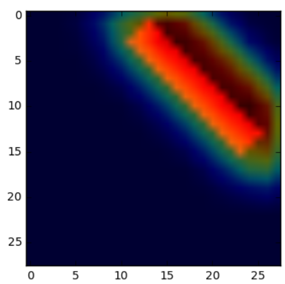


Conv2D(conv2d_2)

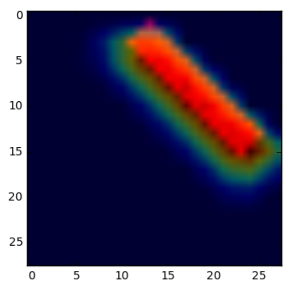


Class Activation Heatmap

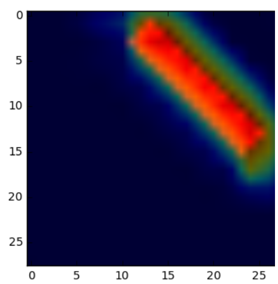
Length



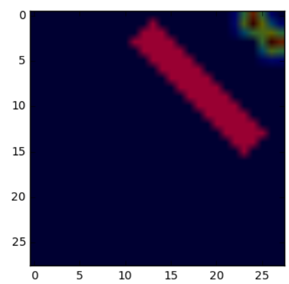
Width



Color



Angle

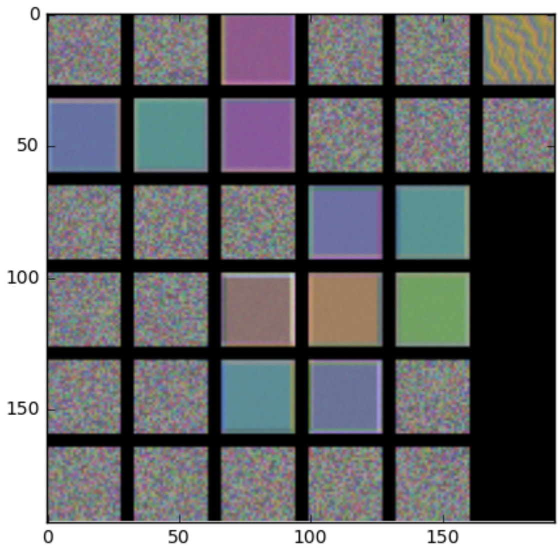


Model : Deep Model on Line Dataset

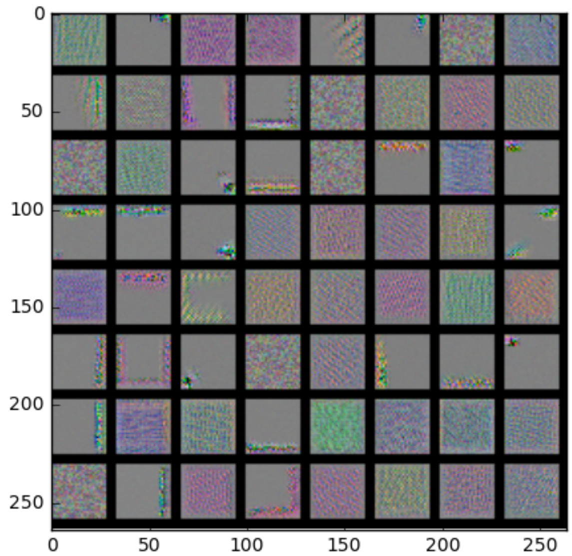
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 26, 26, 32)        896
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 6, 6, 128)         73856
_________________________________________________________________
batch_normalization_4 (Batch (None, 6, 6, 128)         512
_________________________________________________________________
flatten_4 (Flatten)          (None, 4608)              0
_________________________________________________________________
dense_7 (Dense)              (None, 1024)              4719616
_________________________________________________________________
dense_8 (Dense)              (None, 96)                98400
=================================================================
```

Convolutional Filters

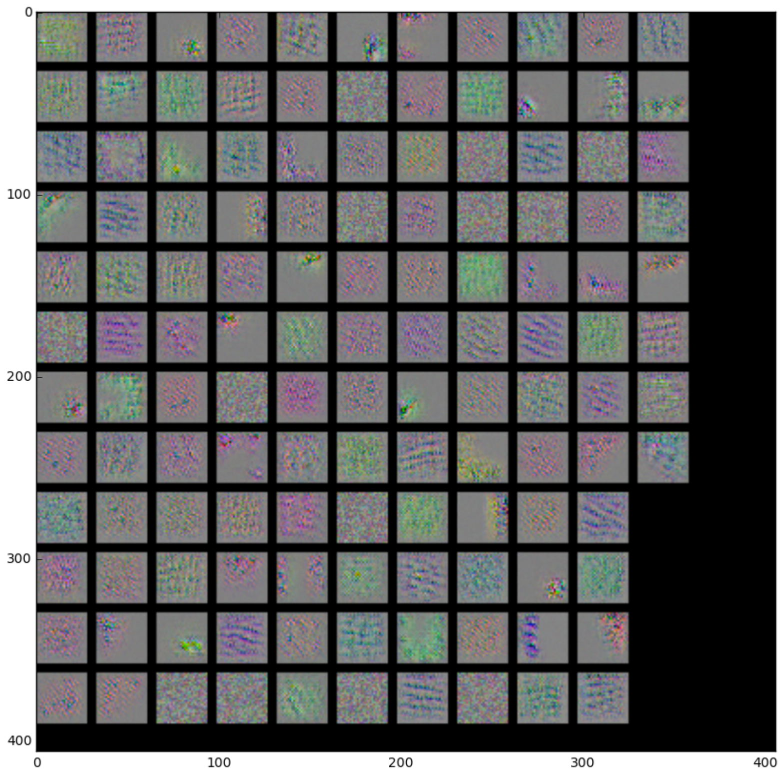
Conv2D(conv2d_8)



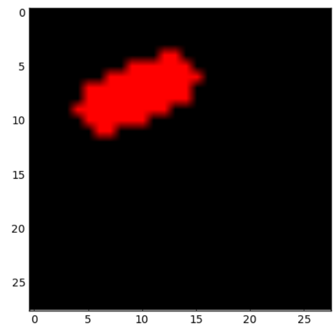
Conv2D(conv2d_9)



Conv2D(conv2d_10)

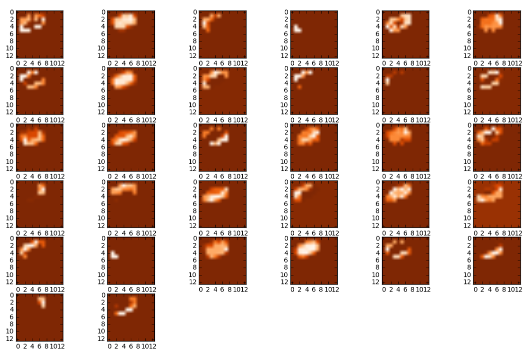


Input Image

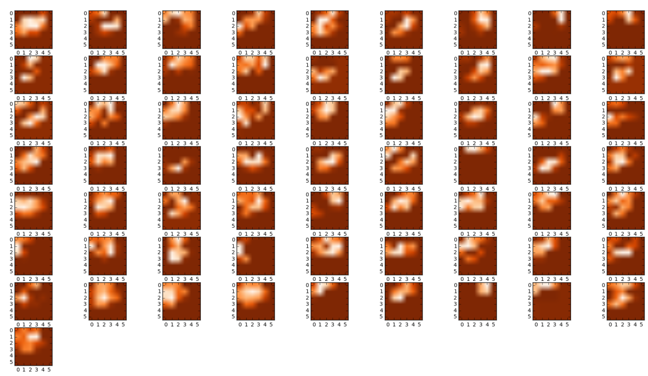


Intermediate Layer Activation

Conv2D(conv2d_8)

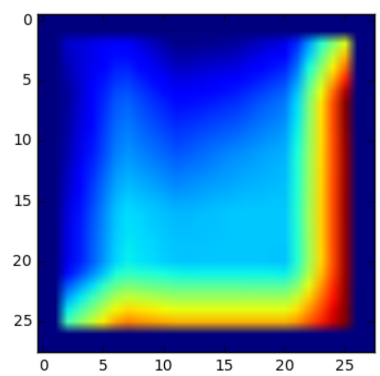


Conv2D(conv2d_9)

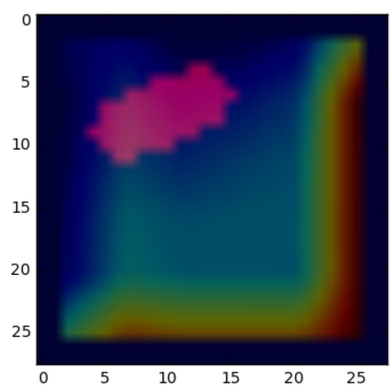


Class Activation Heatmap

Heatmap



Heatmap over Input Image



3.1 Inferences

- Visualization of intermediate layer activation and class activation heatmap shows in most model that the convolutional layers are paying good attention in and around the region which is important for classification.
- Since base model or the model provided in the question is a very shallow network we are only able to visualize what happens initially in convolution and max pooling layer.
- Max pool layer effect on convolutional layer can be clearly seen.
- We observe in convoutional filters some intricate patterns that are learned for the model.
- For images involving 3 channels ,we also observe varied colored filters that are required in classification.
- In variation 3 for model of question 2 we can observe two consecutive conv filter layers and how second conv filters combine small details from conv filter of previous layer.
- We also observe for class activation heatmap for different classes in Question 2 models and how they look at slightly different places to get the best output.
- For some images heatmap doesn't completely overlap with exact input shape,although following the pattern.This is due to resizing heatmap image to overlap original image.
- For obtaining heatmap for some input images, the gradient of class output w.r.t convolutional layer leads to all zero values. Proper normalizing need to be done in those cases.
- Heatmap in case of variation model 3 in question 2,shows bettern visualization as compared to first and second variation, which is in coherence with better accuracy results for variation 3 model.
- More detail learning for mnsit data images can be observed for intermediate layer activation as compared to line dataset ,as in line datset the shapes are very simple.