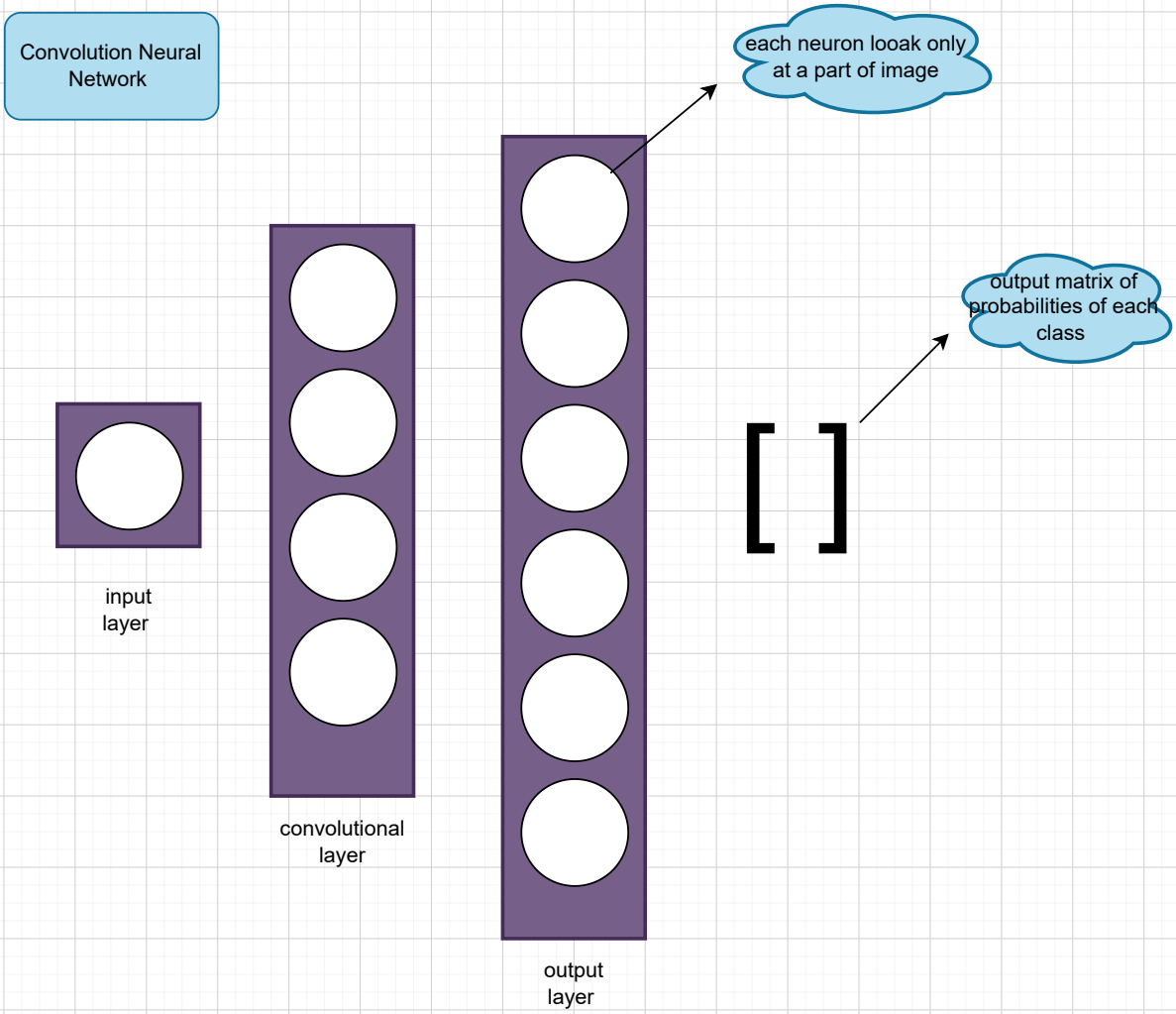
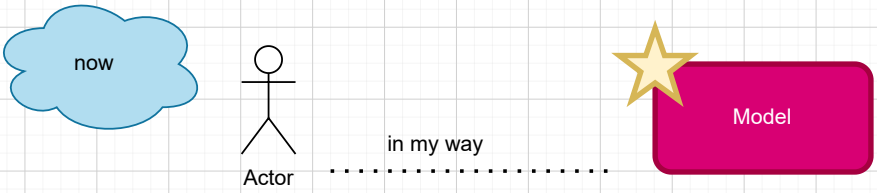


Pooling layers

it is a good way to think that it maintain feature need as it is &minimize the others

Pooling layers effectively reduce the spatial dimensions of feature maps while retaining the most significant information. Max pooling retains the highest values from each region, preserving prominent features, whereas average pooling captures the average value, providing a smoother representation. This helps in reducing computational complexity and improving the model's ability to generalize from training data.



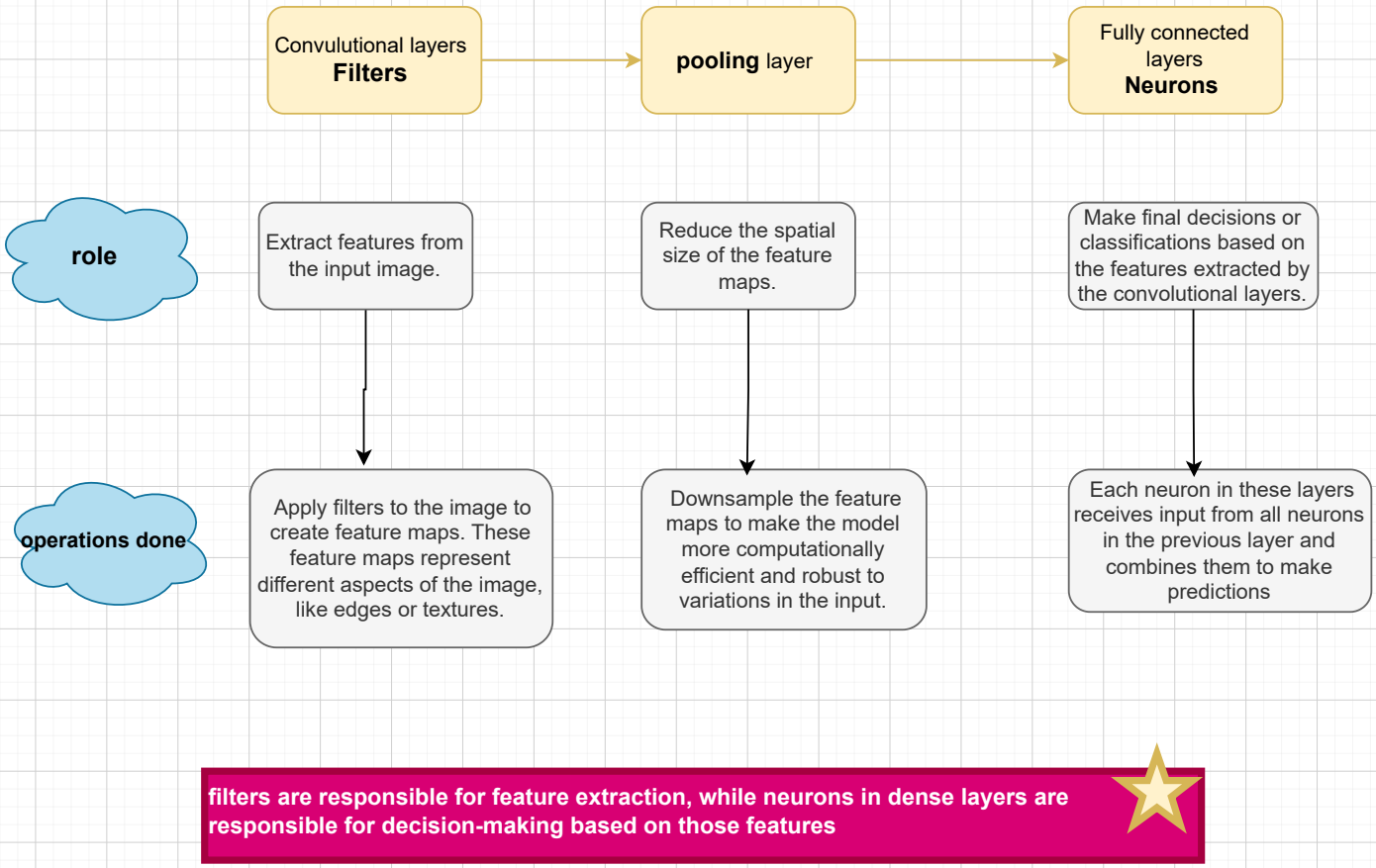
Parameters

filters

- **Purpose**: Filters (or kernels) are used in convolutional layers of a Convolutional Neural Network (CNN). Their job is to scan over the input image and extract features like edges, textures, or patterns.
- **Function**: A filter is a small matrix (e.g., 3x3 or 5x5) that moves across the image to perform convolution operations. This involves multiplying the filter values with the image pixel values it covers, summing the results, and creating a new value in the output feature map.
- **Example**: If you have a 5x5 filter and you apply it to a 32x32 image, the filter moves across the image to detect features. The result is a feature map that highlights where certain features are present in the image.

Neurons

- **Purpose**: Neurons are the basic units in fully connected (dense) layers and in the layers of a neural network. They process inputs and produce outputs based on activation functions.
- **Function**: In a dense layer, each neuron receives input from every neuron in the previous layer. The neuron applies a weight to each input, adds a bias, and then passes the result through an activation function to produce an output. These outputs are then used as inputs for the next layer.
- **Example**: If you have a dense layer with 500 neurons, each neuron receives input from all neurons in the previous layer, processes it, and passes it to the next layer. This allows the network to make complex decisions based on the features extracted by the convolutional layers.



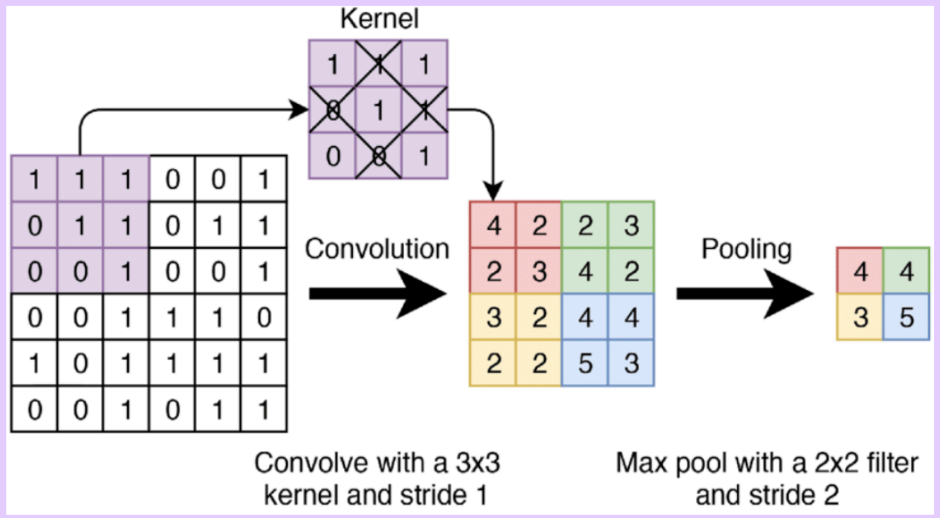
filters are responsible for feature extraction, while neurons in dense layers are responsible for decision-making based on those features

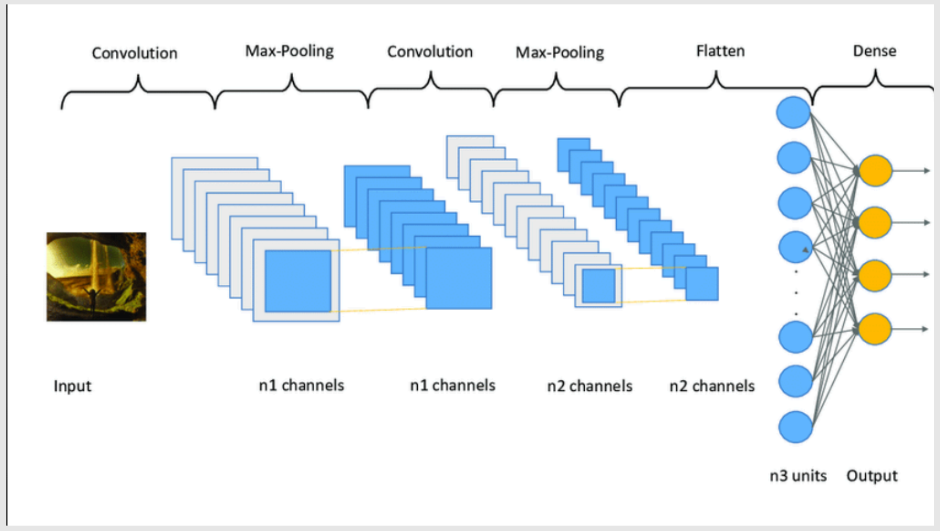
`Dropout`: Adds a dropout layer with a dropout rate of 50%.

- `0.5`: During training, 50% of the neurons in this layer will be randomly set to zero. This helps **prevent overfitting** by ensuring that the model does not rely too heavily on any specific neurons.

`Conv2D` stands for "2D Convolution". It's a layer used in Convolutional Neural Networks (CNNs) to process and **extract features from 2D input data**, like images.

`Flatten()`: Converts the 2D feature maps into a 1D vector. This is necessary before passing the data to **dense** (fully connected) layers.





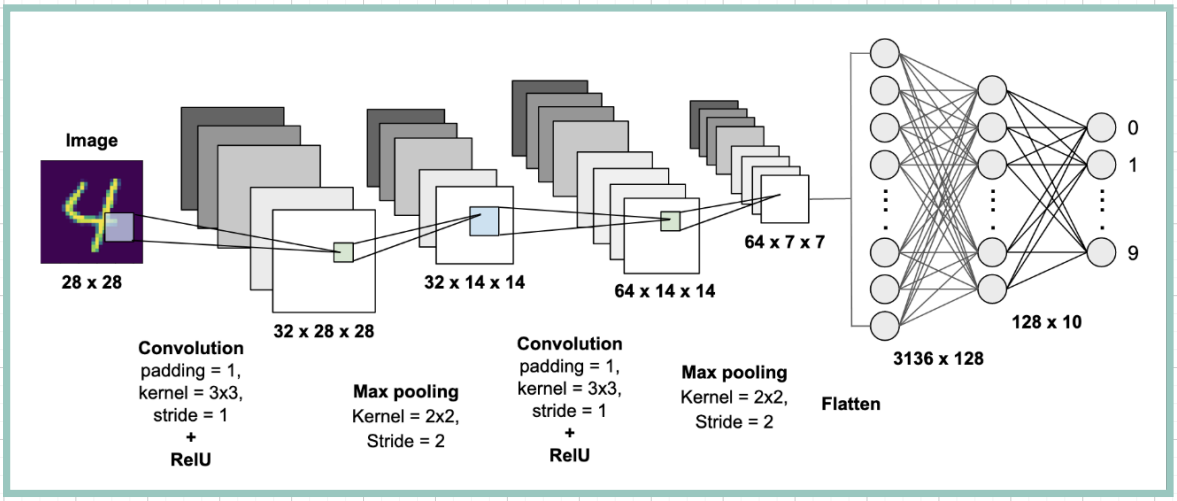


Image
28 x 28
32 x 28 x 28
32 x 14 x 14
64 x 14 x 14
64 x 7 x 7
3136 x 128
128 x 10
0
1
9
Convolution
padding = 1,
kernel = 3x3,
stride = 1
+
RelU
Max pooling
Kernel = 2x2,
Stride = 2
Convolution
padding = 1,
kernel = 3x3,
stride = 1
+
RelU
Max pooling
Kernel = 2x2,
Stride = 2
Flatten