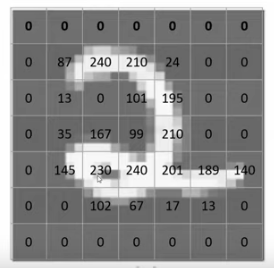
**CNN (Convolutional Neural Network)**

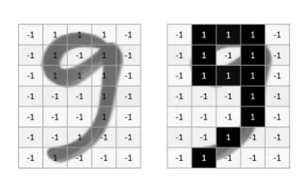
1. CNN image classifications take an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion).
2. Computers see an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.
3. When deep learning CNN models are applied to train and test, each input image will pass through a series of **convolution layers with filters (Kernels),** **Pooling**, **Flatten layer,** **Fully connected layers (FC)** and apply Softmax function to classify an object with probabilistic values between 0 and 1. For multiclass classification, we can use loss as categorical\_crossentropy, sparse\_categorical\_crossentropy.

**Convolution**

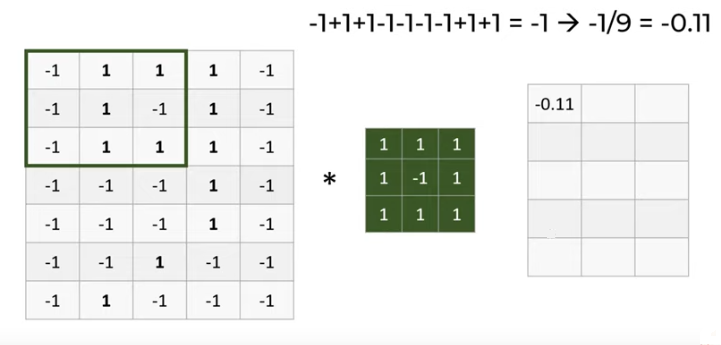
1. Let us suppose this in the input matrix of 5×5 and a filter of matrix 3X3, for those who don’t know what a **filter is a set of weights in a matrix applied on an image or a matrix to obtain the required feature**
2. In convolutional layer, the image pixels (image matrix) are multiplied with filter matrix to generate a feature matrix.
3. Consider the handwritten digit 2 and its pixel representation.

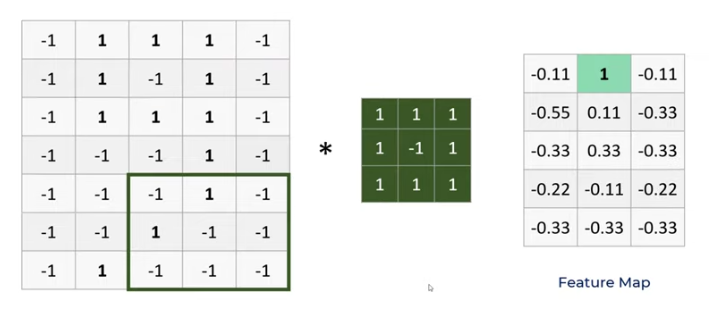
** **

1. For sake of simplicity of mathematical operations, consider image matrix of 9 as follows

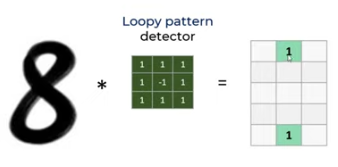
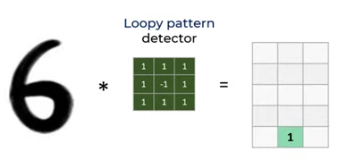
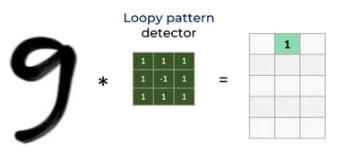


1. Convolution operation is applied using a filter. Let’s assume that a 3\*3 filter is used for this operation. In convolution layer, the image pixels (image matrix) are multiplied with filter (kernels) matrix to generate a feature map.

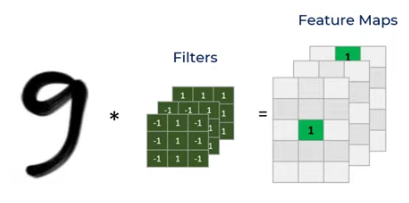




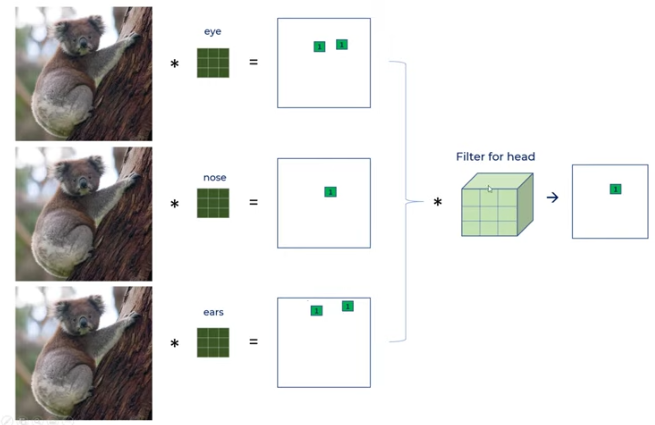
1. This is how result of convolution process detects different patterns in the images using filters.



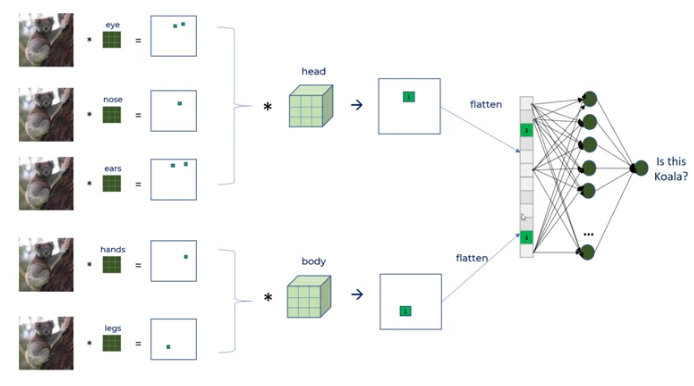
1. f filters are used in each convolution layer to detect different features.



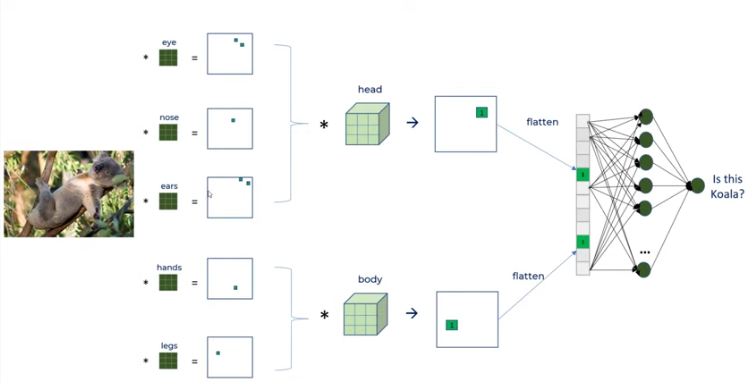
1. Consider another example



1. In case of above example, filters might end up detecting eyes, nose and ears. Then feature map matrix obtained is again subjected to another convolution possibly to detect head of the animal.



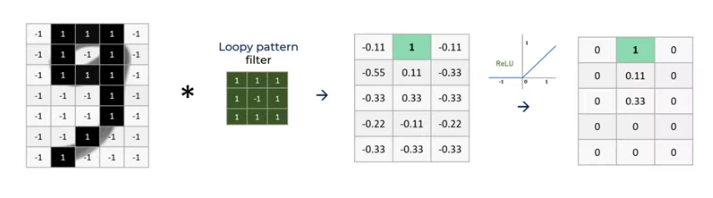
1. After few repeated convolutional operations, the feature map obtained is flattened and then a Dense layer is applied with sigmoid (binary classification) or softmax (multiclass classification) activation function.
2. With different images the flattened matrix will be different.



1. Convolutional layer is responsible for feature extraction.
2. Dense Output layer is responsible for classification.

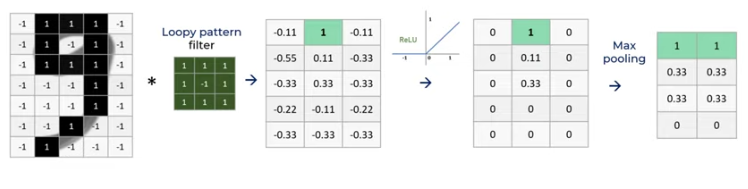
**Activation function is applied to the Feature Map**

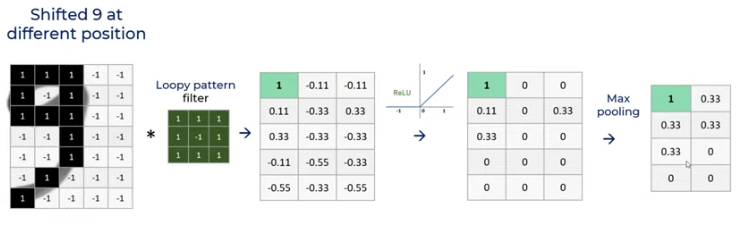
1. On hidden layer (output of convolution layer) activation function is applied.



**Pooling**

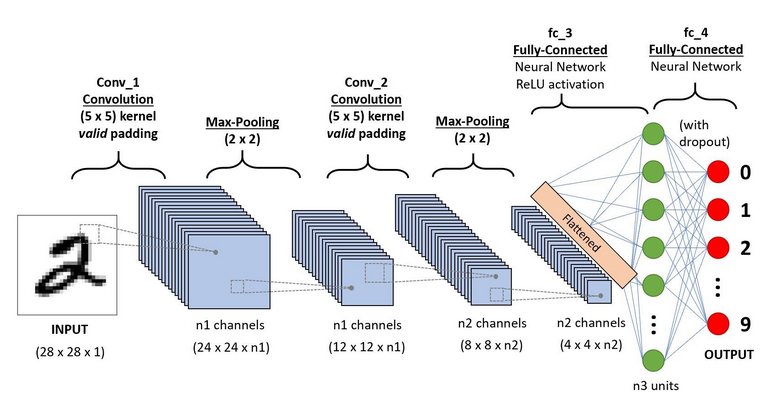
1. Pooling layer is used to reduce the dimension.
2. On the feature map obtained after Convolutional layer, MaxPooling layer is applied to select the maximum value from that window of given stride.





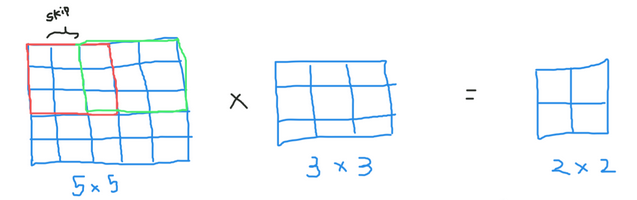
1. Max pooling along with Convolution helps detect position invariant feature detection.
2. Pooling also prevents overfitting as there are less parameters.

**CNN Architecture**



**Stride**

1. Stride is **the number of pixels shifts over the input matrix**. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on



**Input(m,m), Filter(n,n) Stride=p**

**Output = (m-n+1)/p, (m-n+1)/p**

**Padding**

It is applied so that even the corner pixels get their due weightage.

