The CNN Architecture

Agenda

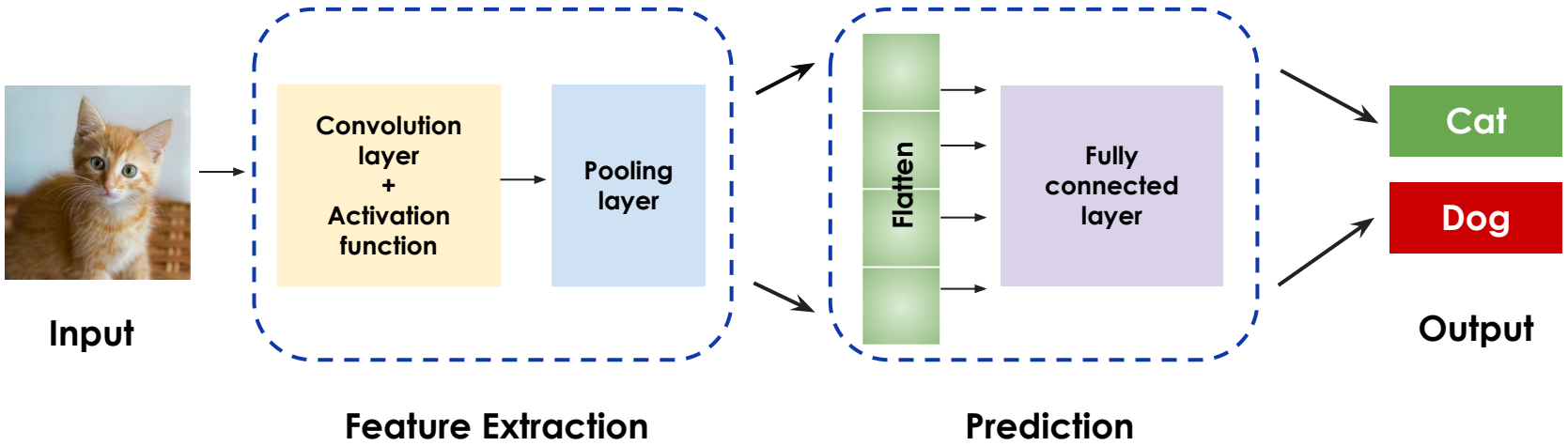
- Architecture and building blocks
- Building a CNN model

Architecture and building blocks

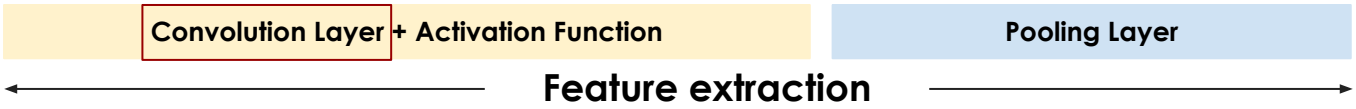
The CNN architecture - Building blocks

The CNN architecture for image classification is comprised of two major parts:

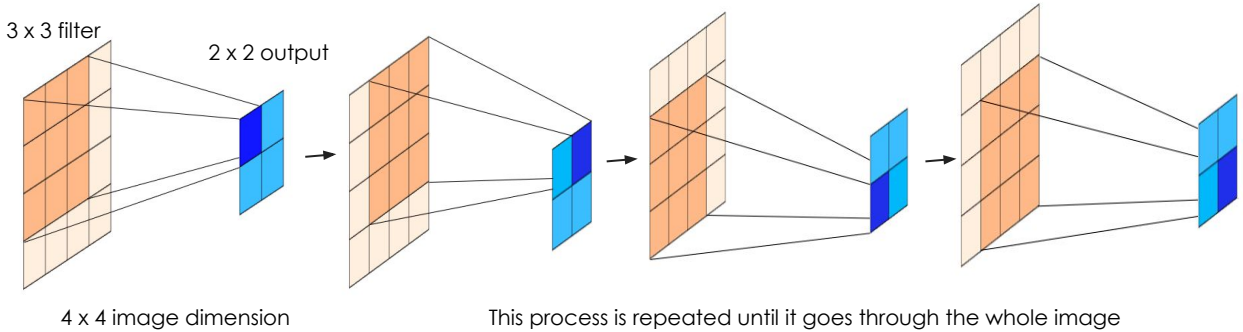
1. **The feature extraction stage** w/ **(a)** Convolution layer **+** Activation function & **(b)** Pooling layer
2. **The prediction stage** w/ **(a)** Flatten layer & **(b)** Fully connected layer



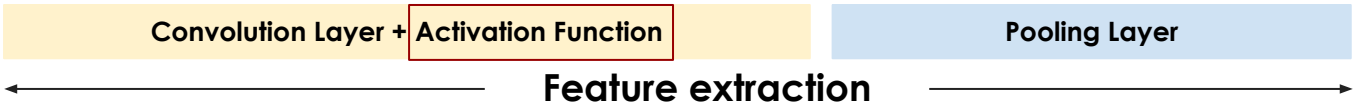
The convolutional layer



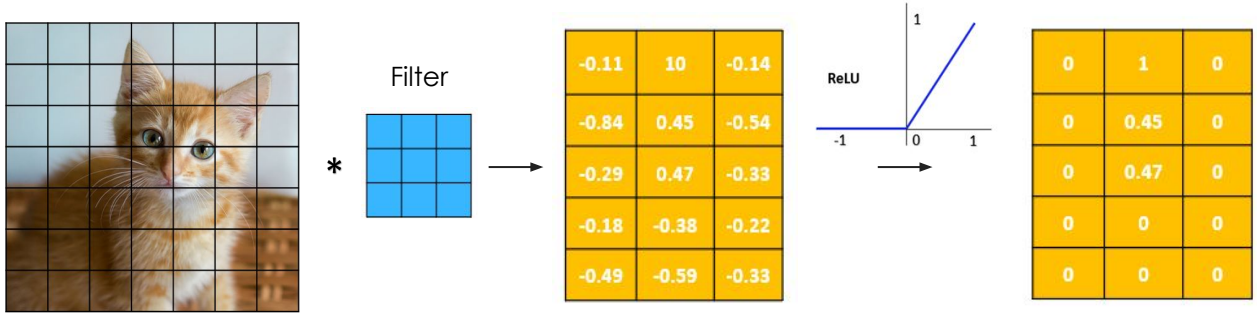
- In the convolutional layer, a filter is applied on an input image to build a feature map.
- Filters can be custom made. In convolutional neural networks though, the values of the filters are learned through training to determine the important features that can categorize the image more accurately. We simply pass the number of filters and their sizes to the CNN, and the model learns the filter values to develop various feature maps that capture the presence of the features detected.



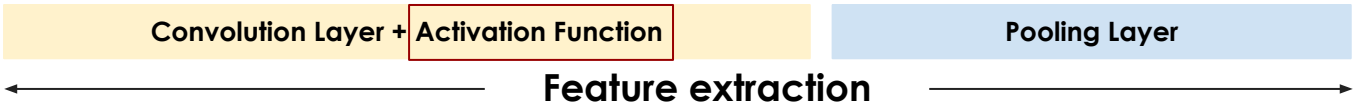
The activation function



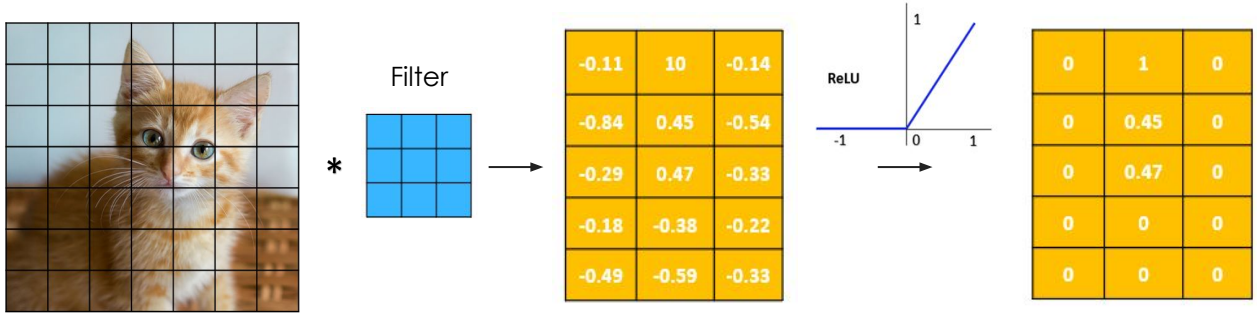
- The next step is the Activation Function. In CNNs, the **ReLU activation function** is used to incorporate a non-linear component into neural networks.
- The ReLU function is used in CNNs because the convolution operation is essentially a linear operation (a sum of element-wise products) between the image and the filter, and ReLU adds the non-linearity required for this operation to be able to detect complex non-linear boundaries in the image.



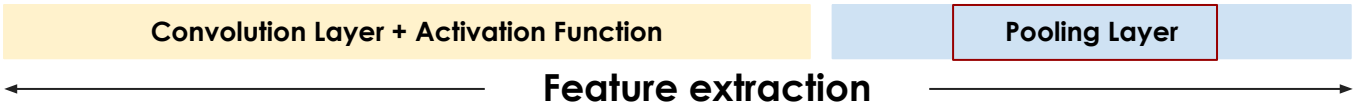
The activation function



- One advantage of using **ReLU (Rectified Linear Unit)** is that if there are any negative pixels, the function will convert it into zero. This is useful for visualizing these image maps, since negative pixels have no meaning, and it makes subsequent computations more efficient since zeros are easy to work with.

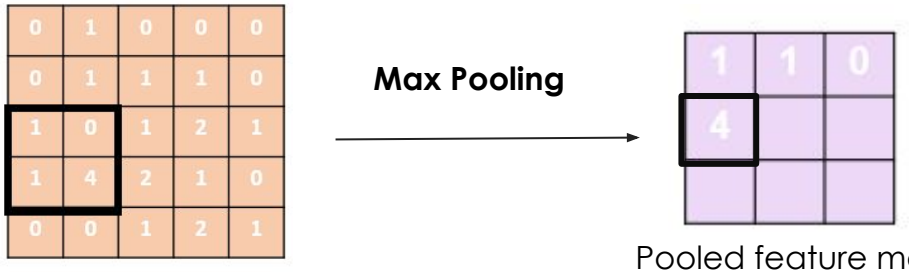


The pooling layer

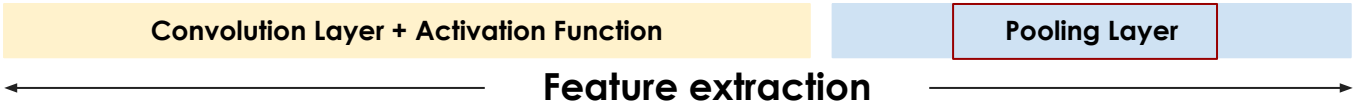


- The next step in the Feature Extraction stage is **Pooling**. The pooling layer helps in removing unwanted features from the image. In doing so, it reduces the size of the image and hence decreases the computational cost of the model.

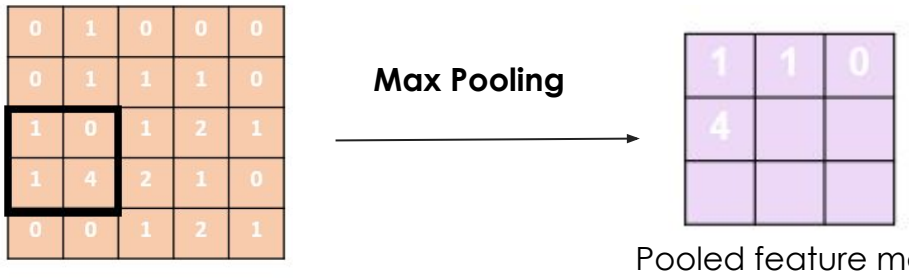
- CNNs can use '**Max Pooling**' to create a pooled feature map using only the maximum values of each patch, and dispose of the unnecessary pixel information.
- An important question here is - **Do we lose information in this process?**
  The answer is **Yes**. The final feature map includes fewer cells and thus less information than the original input image. But, the purpose of this step is to **discard irrelevant features** so that the network can do its job more efficiently.



The pooling layer



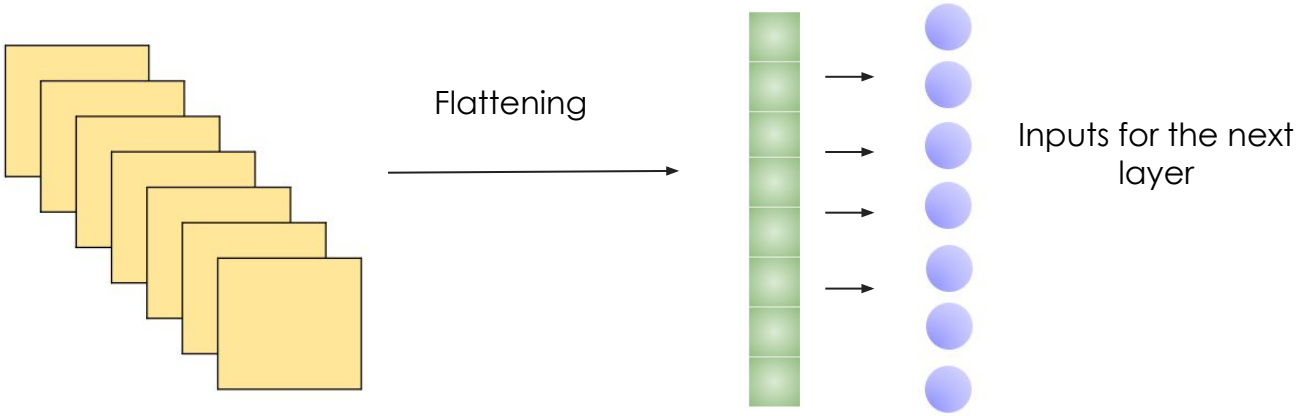
- This process is what's responsible for the "spatial invariance" property of CNNs. Pooling also reduces the size of the images, and hence the number of parameters, which minimises the likelihood of "overfitting", which neural networks are often susceptible to.



The Flatten Layer

- After the Feature Extraction stage, we move to the Prediction stage of the CNN. The first step of the Prediction stage is **the Flatten layer.**

- In this layer, the CNN literally **flattens** the pooled feature map into a column vector, and the output is used in a standard artificial neural network configuration, which are the fully connected layers of the CNN.



The Fully connected layer

- So far, we have learned about the convolution operation, the activation function, pooling and flattening. The final stage of the process is the fully connected layer, where the information from the previous layers is sent to an artificial neural network architecture with Dense (fully connected) layers.

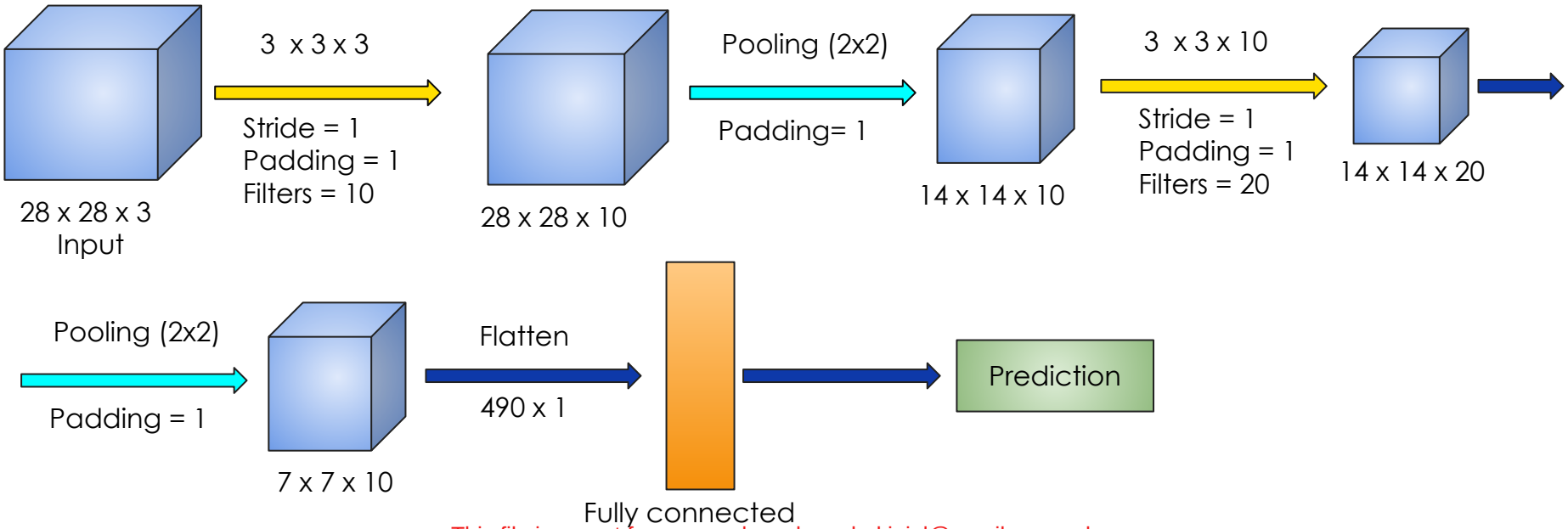
- The aim of this step is to take the input and combine the features into a wider variety of attributes that make the convolutional network more capable of categorising images, which is the sole purpose of building a convolutional neural network.

Building a CNN Model

Let's build a Convolutional Neural Network

Let's say we want to build a simple Convolutional Neural Network (CNN) for a prediction problem.

For example: Let's say we need to classify an image as a dog or a cat.
We can build a simple CNN with convolution and pooling layers to solve this problem.



Summary

To summarize, let's go over what we've learnt about convolutional neural networks in a nutshell:

- We are starting off by applying filters to input images in the form of a convolutional layer.
- We break up the linearity of the convolution operation, using the ReLU activation function.
- We reduce the image dimension using pooling, for isolating the important features and computational efficiency.
- We flatten the feature map and feed it into a fully-connected neural network to generate the final predictions.

Throughout this process, the trainable parameters of the CNN - the weights and filter values, are trained and continuously updated through backpropagation so that the CNN can achieve its best possible performance in image prediction tasks.

Thank You