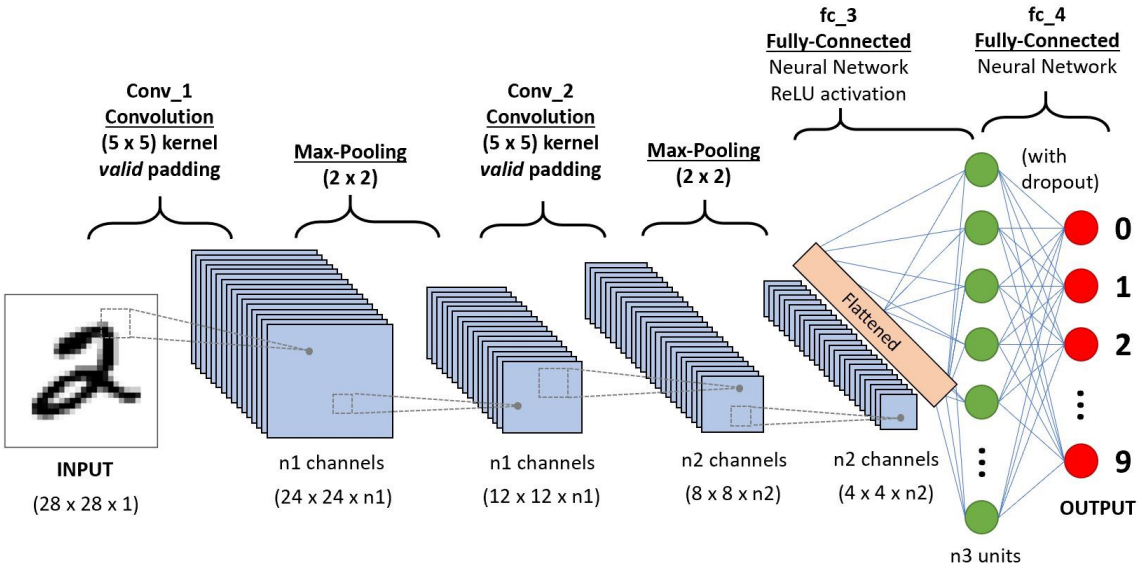
Deep Neural Network

A type of deep learning algorithm CNNs are specifically designed for tasks that involve analyzing visual imagery, such as image classification, object detection, and image segmentation.



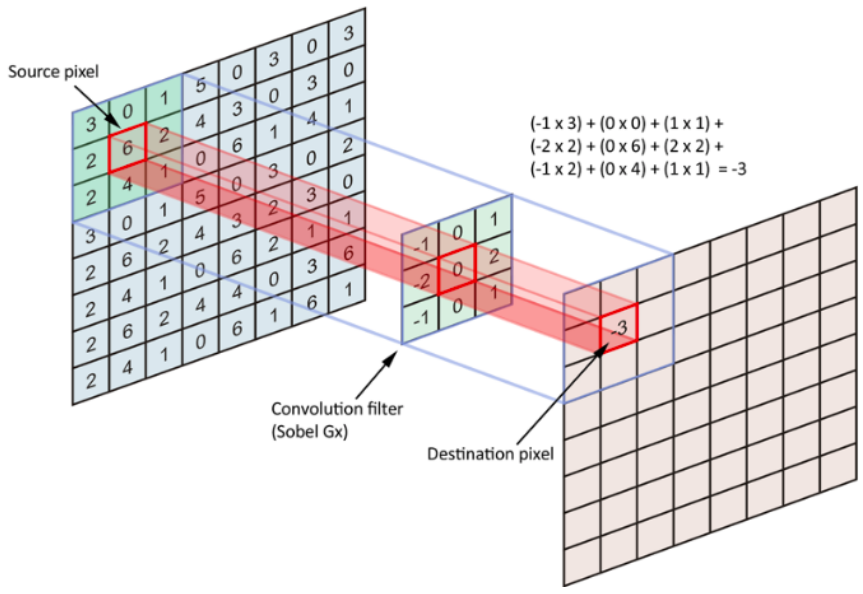
How Do CNNs Work?

CNNs typically consist of the following layers:

1. Convolutional Layer:

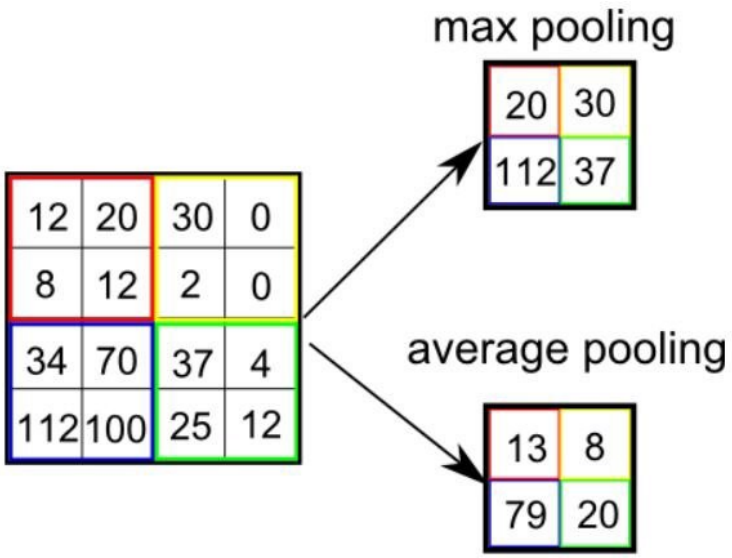
- Filters, also known as kernels, are small matrices of weights that slide across the input data (like an image) to extract specific features. They are the fundamental building blocks of CNNs.
    - **Why are Filters Used?**
        - Filters serve two primary purposes in CNNs:
            - **Feature Extraction:**
                - Filters are designed to detect specific patterns or features in the input data.

- Applies filters (kernels) to the input image to extract features.
- Each filter detects specific patterns in the image.
- The output of this layer is a feature map, which highlights the presence of detected features.



2. Pooling Layer:

- Reduces the dimensionality of the feature maps, making the network more efficient and less prone to overfitting.
- Common pooling techniques include max pooling and average pooling.

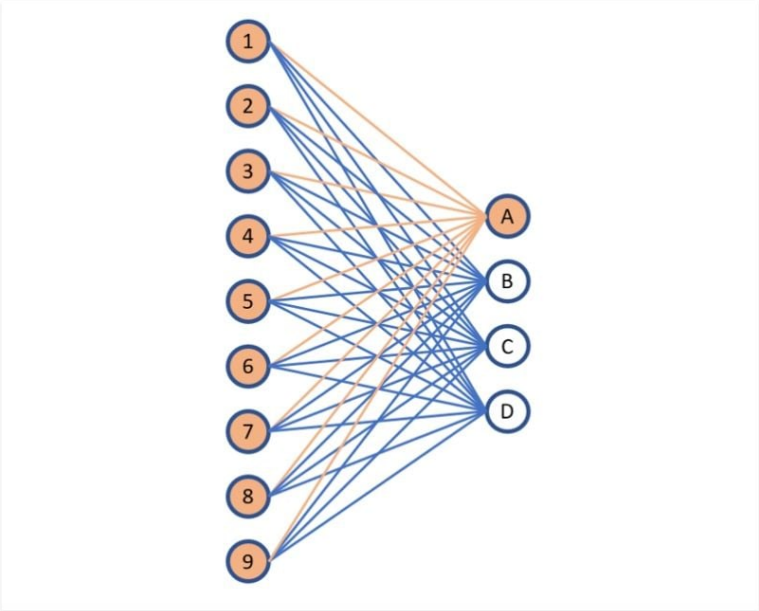


3. Fully Connected Layer:

- flatten is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image.

- Similar to traditional neural networks, this layer connects all neurons from the previous layer to every neuron in the current layer.

- It performs classification or regression tasks based on the learned features.



Advantages of CNNs:

1. Good at detecting patterns and features in images, videos, and audio signals.
2. Robust to translation, rotation, and scaling invariance.
3. End-to-end training, no need for manual feature extraction.
4. Can handle large amounts of data and achieve high accuracy.

Disadvantages of CNNs:

1. Computationally expensive to train and require a lot of memory.
2. Can be prone to overfitting if not enough data or proper regularization is used.
3. Requires large amounts of labeled data.
4. Interpretability is limited, it’s hard to understand what the network has learned.