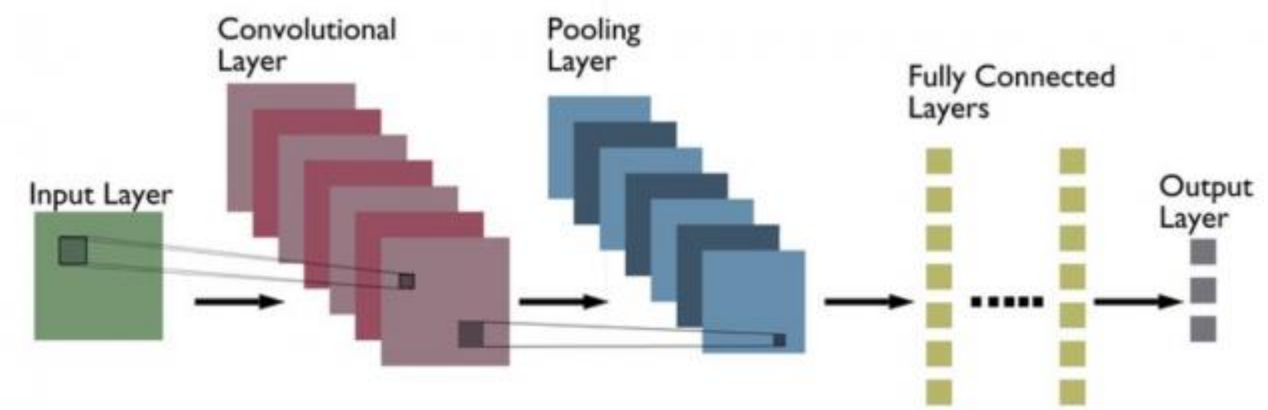
CNN **Architectures**

Convolutional Neural Networks, commonly referred to as CNNs, are a specialized kind of neural network architecture that is designed to process data with a grid-like topology. This makes them particularly well-suited for **dealing with spatial and temporal data**, like **images** and **videos**, that maintain a high degree of correlation between adjacent elements.

CNNs are similar to other neural networks, but they have an added layer of complexity due to the fact that they use a **series of convolutional layers**. Convolutional layers perform a **mathematical operation** called **convolution**, a sort of **specialized matrix multiplication**, on the input data. The convolution operation helps to preserve the spatial relationship between pixels by learning image features using small squares of input data. . The picture below represents a typical CNN architecture



The LeNet architecture consists of two convolutional layers, two subsampling layers (max pooling layer), and three fully connected layers. Overall, the LeNet architecture demonstrates the power of convolutional neural networks for image recognition tasks and how it paved the way for many subsequent developments in deep learning algorithms.

Image source

The convolutional layer

Image source

Let's consider an input image with height *H*, width *W*, and *C* channels (for example, an RGB image has 3 channels). We also have a set of *K* learnable filters (kernels), where each filter has a height, width, and channels. The channels are usually much bigger than the input number of channels, and the height and width of the filter are significantly lower than the spatial resolution of the input.

The 2D convolution operation takes each filter and slides it over the input image. With each position, it performs an element-wise multiplication between the filter and the corresponding input image pixels and then sums the resulting values to produce a single output value. This operation is equivalent to a matrix multiplication operation with a weight of the matrix constructed from the repeated values from the kernel, making it equivalent to a fully connected layer with weight sharing. It is also important to note that an input image's pixel values are not precisely connected to the output layer in partly connected layers.

To compute the output feature map for all filters, we need to perform the convolution operation K times, once for each filter. The output feature map is then obtained by stacking the feature maps along the depth axis. It's worth noting that the size of the output feature map is determined by the size of the input image, the size of the filter, the stride, and the padding.

The Cambrian Explosion

Image source

The Cambrian Explosion refers to a period of rapid diversification of life on Earth that occurred approximately 541 million years ago. For the past several decades, deep learning underwent significant growth and progress, resulting in a huge zoo of deep learning models capable of solving a wide range of tasks that were previously out of reach for autonomous agents.

In particular, the development of deep neural networks has led to breakthroughs in a wide range of applications and machine learning algorithms, including image recognition, speech recognition, natural language processing, and game playing. The success of the LeNet on MNIST data set acted as a trigger for this huge explosion that lasted for several decades, but there were other factors catalyzing the process, such as:

- **Increased availability of data:** With the proliferation of digital devices and the growth of the internet, there has been a massive increase in the amount of training data that is available for machine learning models.
- **Advances in hardware:** The development of graphical processing units (GPUs) and other specialized hardware has made it possible to train deep neural networks more efficiently than was previously possible.
- **Development of new algorithms:** Researchers have developed new techniques for training deep neural networks, mostly the modifications of a first-order method - gradient descent which allows faster convergence and sometimes helps to find better local minima.
- **Open-source software:** The availability of open-source software frameworks like TensorFlow and PyTorch has made it easier for researchers and developers to experiment with deep learning techniques.