ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DAY – 10

Date: Jul 04, 2025

Convolutional Neural Network (CNN) in Machine Learning

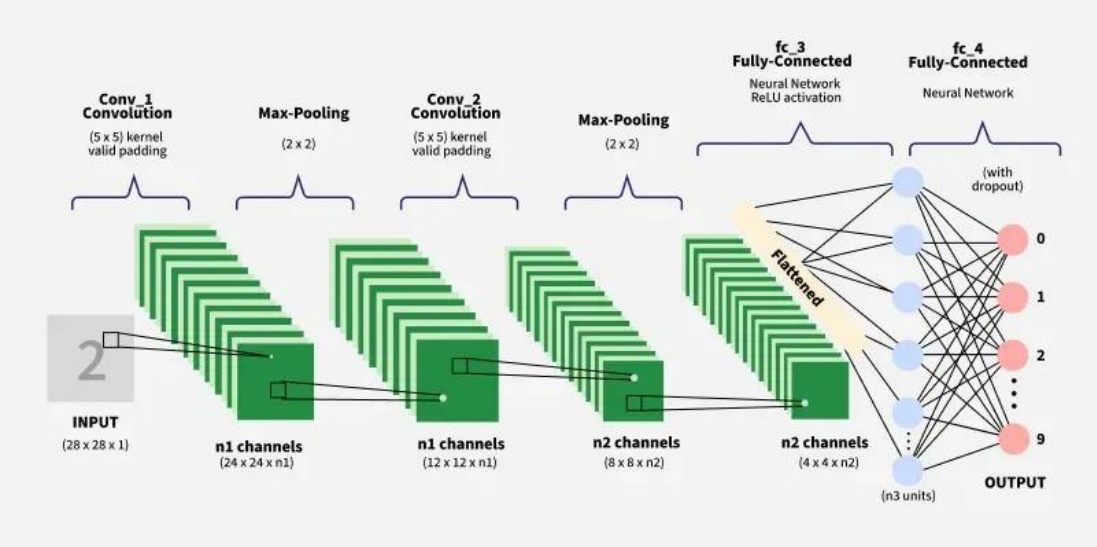
Convolutional Neural Networks (CNNs) are deep learning models designed to process data with a grid-like topology such as images. They are the foundation for most modern computer vision applications to detect features within visual data.

Key Components of a Convolutional Neural Network

- **Convolutional Layers**: These layers apply convolutional operations to input images using filters or kernels to detect features such as edges, textures and more complex patterns. Convolutional operations help preserve the spatial relationships between pixels.
- **Pooling Layers:** They downsample the spatial dimensions of the input, reducing the computational complexity and the number of parameters in the network. Max pooling is a common pooling operation where we select a maximum value from a group of neighboring pixels.
- **Activation Functions:** They introduce non-linearity to the model by allowing it to learn more complex relationships in the data.
- **Fully Connected Layers:** These layers are responsible for making predictions based on the high-level features learned by the previous layers. They connect every neuron in one layer to every neuron in the next layer.

How CNNs Work?

1. **Input Image:** CNN receives an input image which is preprocessed to ensure uniformity in size and format.
2. **Convolutional Layers:** Filters are applied to the input image to extract features like edges, textures and shapes.
3. **Pooling Layers:** The feature maps generated by the convolutional layers are downsampled to reduce dimensionality.
4. **Fully Connected Layers:** The downsampled feature maps are passed through fully connected layers to produce the final output, such as a classification label.
5. **Output:** The CNN outputs a prediction, such as the class of the image.



Different Types of CNN Models

1. LeNet

LeNet developed by Yann LeCun and his colleagues in the late 1990s was one of the first successful CNNs designed for handwritten digit recognition. It laid the foundation for modern CNNs and achieved high accuracy on the MNIST dataset which contains 70,000 images of handwritten digits (0-9).

2. AlexNet

AlexNet is a CNN architecture that was developed by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton in 2012. It was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) a major image recognition competition. It consists of several layers of convolutional and pooling layers followed by fully connected layers.

3. Resnet

ResNets (Residual Networks) are designed for image recognition and processing tasks. They are renowned for their ability to train very deep networks without overfitting making them highly effective for complex tasks.

4. GoogleNet

GoogleNet also known as InceptionNet is renowned for achieving high accuracy in image classification while using fewer parameters and computational resources compared to other state-of-the-art CNNs.

5. VGG

VGGs are developed by the Visual Geometry Group at Oxford, it uses small 3x3 convolutional filters stacked in multiple layers, creating a deep and uniform structure. Popular variants like

VGG-16 and VGG-19 achieved state-of-the-art performance on the ImageNet dataset demonstrating the power of depth in CNNs.

Applications of CNN

- **Image classification:** CNNs are the state-of-the-art models for image classification. They can be used to classify images into different categories such as cats and dogs.
- **Object detection:** It can be used to detect objects in images such as people, cars and buildings. They can also be used to localize objects in images which means that they can identify the location of an object in an image.
- **Image segmentation:** It can be used to segment images which means that they can identify and label different objects in an image. This is useful for applications such as medical imaging and robotics.
- **Video analysis:** It can be used to analyze videos such as tracking objects in a video or detecting events in a video. This is useful for applications such as video surveillance and traffic monitoring.

K-Nearest Neighbor(KNN) Algorithm

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm generally used for classification but can also be used for regression tasks. It works by finding the "k" closest data points (neighbors) to a given input and makesa predictions based on the majority class (for classification) or the average value (for regression). Since KNN makes no assumptions about the underlying data distribution it makes it a non-parametric and instance-based learning method.

What is 'K' in K Nearest Neighbour?

In the k-Nearest Neighbours algorithm k is just a number that tells the algorithm how many nearby points or neighbors to look at when it makes a decision.

Example: Imagine you're deciding which fruit it is based on its shape and size. You compare it to fruits you already know.

If k = 3, the algorithm looks at the 3 closest fruits to the new one.

If 2 of those 3 fruits are apples and 1 is a banana, the algorithm says the new fruit is an apple because most of its neighbors are apples.

Statistical Methods for Selecting k

- **Cross-Validation:** Cross-Validation is a good way to find the best value of k is by using k-fold cross-validation. This means dividing the dataset into k parts. The model is trained on some of these parts and tested on the remaining ones. This process is repeated for each part. The k value that gives the highest average accuracy during these tests is usually the best one to use.

- **Elbow Method:** In Elbow Method we draw a graph showing the error rate or accuracy for different k values. As k increases the error usually drops at first. But after a certain point error stops decreasing quickly. The point where the curve changes direction and looks like an "elbow" is usually the best choice for k.
- **Odd Values for k:** It's a good idea to use an odd number for k especially in classification problems. This helps avoid ties when deciding which class is the most common among the neighbors.

Working of KNN algorithm

The K-Nearest Neighbors (KNN) algorithm operates on the principle of similarity where it predicts the label or value of a new data point by considering the labels or values of its K nearest neighbors in the training dataset.

Step 1: Selecting the optimal value of K

K represents the number of nearest neighbors that needs to be considered while making prediction.

Step 2: Calculating distance

To measure the similarity between target and training data points Euclidean distance is used. Distance is calculated between data points in the dataset and target point.

Step 3: Finding Nearest Neighbors

The k data points with the smallest distances to the target point are nearest neighbors.

Step 4: Voting for Classification or Taking Average for Regression

- When you want to classify a data point into a category like spam or not spam, the KNN algorithm looks at the K closest points in the dataset. These closest points are called neighbors. The algorithm then looks at which category the neighbors belong to and picks the one that appears the most. This is called majority voting.
- In regression, the algorithm still looks for the K closest points. But instead of voting for a class in classification, it takes the average of the values of those K neighbors. This average is the predicted value for the new point for the algorithm.

It shows how a test point is classified based on its nearest neighbors. As the test point moves the algorithm identifies the closest 'k' data points i.e. 5 in this case and assigns test point the majority class label that is grey label class here.

Applications of KNN

- **Recommendation Systems:** Suggests items like movies or products by finding users with similar preferences.

- **Spam Detection:** Identifies spam emails by comparing new emails to known spam and non-spam examples.
- **Customer Segmentation:** Groups customers by comparing their shopping behavior to others.
- **Speech Recognition:** Matches spoken words to known patterns to convert them into text.

LAMP

LAMP refers to a software stack whose initial letters consist of the following individual components:

- Linux operating system
- Apache Web Server
- MySQL database system
- PHP Scripting Language

Install LAMP on Ubuntu

Step 1: Install Apache

- Update the package index:

```
sudo apt update
```

- Install Apache:

```
sudo apt install apache2
```

Step 2: Install MySQL

- Install MySQL server:

```
sudo apt install mysql-server
```

Step 3: Install PHP

- Install PHP and required modules:

```
sudo apt install php libapache2-mod-php php-mysql
```

Step 4: Test PHP Processing

- Create a test PHP file:

```
sudo nano /var/www/html/info.php
```