Day76_CNN_Introduction

August 29, 2025

1 Introduction to Convolutional Neural Networks - CNN

Welcome back to my Deep Learning documentation!

In this notebook, we begin our journey into **Convolutional Neural Networks (CNNs)**, one of the most powerful models in deep learning for **images, videos, and computer vision tasks**.

2 What is CNN?

- CNN = **Convolution Layer + Neural Network**
- Unlike traditional ANNs, CNNs are designed to handle **spatial data** (images/videos).

Example:

- A CNN can look at a emoji and classify it as **Happy**.
- A CNN can look at a emoji and classify it as **Sad**.

Key idea: CNNs automatically detect patterns like edges, shapes, and colors in images, which then combine to recognize complex objects.

3 Data for CNNs

3.1 Image Representation

- Every image is made of **pixels**.
- Each pixel has an intensity value between **0 and 255**.
  - 0 → black / no intensity.
  - 255 → maximum brightness of a color.

Example:

- A pure black pixel = 0.
- A pure white pixel = 255.
- A gray pixel = somewhere between (e.g., 128).

3.2 Channels in Images

- **Grayscale (2D)**: Black & White images, only one channel.
- **RGB (3D)**: Color images, 3 channels (Red, Green, Blue).

Example:

- A grayscale photo of a handwritten digit (MNIST dataset).
- A color image of a cat/dog has 3 channels (R, G, B).

4 Prerequisites for CNN

Before learning CNNs, two areas are important:

4.1 Image Processing

- Deals with **enhancing or transforming images**.
- Tasks: resizing, filtering, edge detection, noise removal.

Example: Instagram filters = Image Processing.

4.2 Computer Vision

- Deals with making computers **understand the content of images**.
- Tasks: object detection, face recognition, image classification.

Example: Facebook automatically tagging people in photos = Computer Vision.

5 Architecture of CNN

A standard CNN has **4 key layers**:

1. Convolution
2. Max Pooling
3. Flatten
4. Fully Connected (ANN)

Each plays a specific role in extracting features and making predictions.

6 Convolution Layer

6.1 What is Convolution?

- Convolution = **mathematical operation** that combines two pieces of information:

1. **Input Image** (matrix of pixels).
2. **Filter / Kernel (Feature Detector)** → a small matrix like 3×3 or 5×5.

Output = **Feature Map** (convolved image).

6.2 Real-Life Analogy

Imagine looking at a photo through a **magnifying glass with a pattern**.

- If the pattern is horizontal lines → the glass highlights horizontal edges.
- If the pattern is vertical lines → the glass highlights vertical edges.

That's how convolution detects features like edges, curves, and textures.

6.3 Striding

- Stride = how many steps the filter moves across the image.
- Stride = 1 → moves one pixel at a time (more detail, larger output).
- Stride = 2 → skips more pixels (smaller output, faster computation).

6.4 Padding

- Problem: Each convolution reduces image size.
- Solution: **Padding** adds an extra border of 0s around the image.
- Helps preserve edge information.

Example:

- Input 6×6 image → Convolution → 4×4 output.
- With padding → size maintained (e.g., 6×6 stays 6×6).

7 Max Pooling Layer

7.1 What is Pooling?

- Pooling = reduces the size of the feature map.
- Default = **2×2 Max Pooling** → picks the maximum value from each 2×2 block.

7.2 Why Pooling?

1. Reduces computational load.
2. Keeps the most important features.

3. Makes the model more robust (ignores minor variations).

7.3 Example

Feature map section:

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

After 2×2 max pooling → **4** (the maximum).

Think of it like **taking the strongest signal** and discarding the weaker ones.

8 Flatten Layer

- Takes the pooled feature maps (2D) and converts them into a **1D vector**.
- This long vector acts as the **input to the fully connected ANN layer**.

Example:

- Pooled maps: 5×5 → Flatten → 25 values in a single row.

9 Fully Connected Layer (ANN)

- Works just like a regular **Artificial Neural Network**.
- The flattened vector is input.
- Outputs predictions (e.g., cat, dog, happy, sad).

Example:

- Input photo → CNN layers extract features → ANN outputs:
  - Cat = 0.85 probability
  - Dog = 0.10
  - Other = 0.05

Prediction = **Cat**

10 ANN vs CNN (Key Difference)

- In **ANN**:
  - Training adjusts **weights** of connections.
- In **CNN**:
  - Training adjusts **filter values (kernels)**.

This is why CNNs are much better at handling images — they learn **filters** that detect edges, colors, and shapes automatically.

11 Summary

- CNNs are specialized for **images & videos**.
- They use pixels (0–255 values) as input.
- Main layers:
  1. Convolution → feature extraction
  2. Max Pooling → dimensionality reduction
  3. Flatten → converts into 1D vector
  4. Fully Connected Layer → prediction
- **ANN updates weights**; **CNN updates filters**.

12 Conclusion

In this notebook, we:
- Learned why CNNs are important for images and videos.
- Understood image basics (pixels, grayscale vs RGB).
- Covered CNN layers: Convolution, Pooling, Flatten, Fully Connected.
- Compared ANN and CNN training.