Convolutional Neural Networks

Understanding CNNs from Basics

July 19, 2025

Outline

What is a CNN?

What is a Convolutional Neural Network?

CNNs are a class of deep learning models primarily used for image, video, and spatial data.

- Automatically extract spatial features using filters
- Replaces manual feature extraction
- Efficient in capturing spatial hierarchies

Why Use CNNs?

- Fully connected networks do not scale well for image data
- CNNs preserve spatial locality using convolution
- Fewer parameters, shared weights
- Effective in object detection, segmentation, and classification

Convolution Layer

What is Convolution?

Convolution is a weighted sum of inputs in a local region.

- Applies a small filter (kernel) over the input
- Captures local features like edges, textures
- Parameters are shared across spatial locations

Convolution Math

Let:

- Input: $I \in \mathbb{R}^{H \times W}$
- Kernel: $K \in \mathbb{R}^{k \times k}$

Then, output at position (i, j) is:

$$S(i,j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I(i+m, j+n) \cdot K(m,n)$$

Output Shape Formula

Given:

- Input size: $H_{in} \times W_{in}$
- Kernel size: K
- Padding: P
- Stride: S

Then:

$$H_{out} = \left\lfloor \frac{H_{in} + 2P - K}{S} + 1 \right\rfloor \quad W_{out} = \left\lfloor \frac{W_{in} + 2P - K}{S} + 1 \right\rfloor$$

Example: Output Size

- Input: 32×32
- Kernel: 5×5
- Stride: 1, Padding: 0

$$H_{out} = \frac{32 - 5}{1} + 1 = 28, \quad W_{out} = \frac{32 - 5}{1} + 1 = 28$$

Output size: 28×28

Padding and Stride

What is Padding?

- Adds border around the input
- Keeps output the same size (when needed)
- Common types: 'valid' (no padding), 'same' (pad to preserve size)

What is Stride?

- Number of pixels the filter moves per step
- Stride ¿ 1 reduces output size
- Acts like downsampling

Activation and Pooling

Activation Function

ReLU (Rectified Linear Unit) is most commonly used:

$$f(x) = \max(0, x)$$

- Introduces non-linearity
- Helps prevent vanishing gradients

Pooling Layers

Max Pooling and **Average Pooling** reduce spatial dimensions.

- Aggregates regions to smaller representation
- Reduces computation, controls overfitting

Example:

- Input: 4×4, Pool: 2×2, Stride: 2
- Output: 2×2

CNN Architecture

Typical CNN Architecture

- Input Layer (e.g., image)
- Convolution $\rightarrow$ ReLU $\rightarrow$ Pooling
- Repeat above layers
- Flatten $\rightarrow$ Fully Connected Layer(s)
- Output Layer (Softmax or Sigmoid)

Parameter Sharing and Sparsity

- Each filter scans entire image but uses same weights
- Fewer parameters than fully connected layers
- Efficient training and less overfitting

Use Cases

Applications of CNNs

- Image classification (e.g., CIFAR-10, ImageNet)
- Object detection (YOLO, Faster R-CNN)
- Face recognition and biometrics
- Medical image analysis
- NLP (e.g., text classification with 1D CNNs)

Benefits of CNNs

- Captures spatial features automatically
- Good generalization with less data
- State-of-the-art for image-based tasks

Key Concepts Recap

- Convolution: Local feature detection
- Pooling: Dimensionality reduction
- ReLU: Non-linearity
- CNNs are efficient for structured data like images

Thank you!

Questions?