Convolutional Neural Networks (CNNs)

Theory, Mathematics, and Architecture Design

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería
Universidad Anáhuac México

January 21, 2026

Agenda

Biological Inspiration

CNNs are biologically inspired by the visual cortex organization (Hubel & Wiesel, 1959).

- Local Receptive Fields: Neurons respond only to a small sub region of the visual field.
- Hierarchy:
- **V1 Area:** Detects simple edges and orientations.
- **Higher Areas:** Detect shapes, textures, and objects.

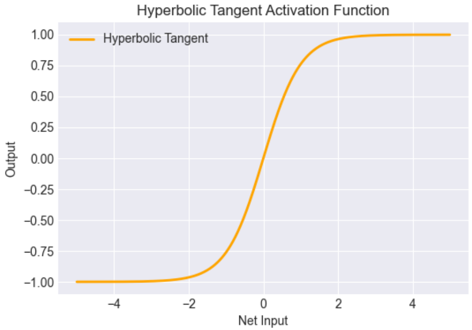


Figure 1: Hierarchical processing in the brain vs. Deep Learning.

Why not Standard MLPs?

Imagine an image of 224×224 pixels (RGB).

The Curse of Dimensionality

If we flatten this image into a vector:

$$\text{Input Vector} = 224 \times 224 \times 3 = 150,528 \text{ inputs}$$

If the first hidden layer has 1,000 neurons (Dense Layer):

$$\text{Weights} = 150,528 \times 1,000 \approx 150 \text{ Million Parameters!}$$

Problem: Huge computational cost and massive overfitting.
Solution: Local connectivity and Parameter Sharing (CNNs).

Agenda

The Mathematical Definition

In Deep Learning, a "convolution" is technically a **sliding dot product** (cross-correlation).
We pass a filter (kernel) K over an image I:

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \tag{1}$$

- **Element-wise multiplication:** Between the kernel weights and the image patch.
- **Summation:** Aggregating the result into a single pixel in the Feature Map.

Feature Maps & Channels

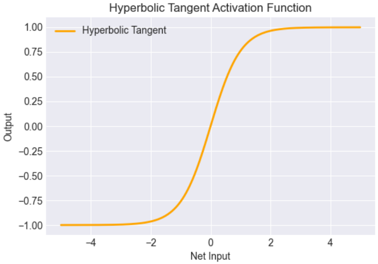
Input Tensor ($H \times W \times C_{in}$)

An image usually has 3 channels (RGB).

Output Tensor ($H' \times W' \times K$)

If we apply K distinct filters (e.g., 32 filters):

- We obtain 32 discrete 2D maps.
- These stack together to form the output depth (Channels).
- **Analogy:** Each channel represents a specific "feature" (one for horizontal lines, one for



[Placeholder: Animation of a kernel sliding over a matrix]

Figure 2: The sliding window

Agenda

Spatial Dimensions Control

How do we calculate the output size of a layer? This is the fundamental arithmetic equation for CNN design.

$$O = \left\lfloor \frac{W - F + 2P}{S} \right\rfloor + 1 \tag{2}$$

Where:

- W: Input Volume Size (Width/Height)
- F: Filter /Kernel Size (e.g., 3×3)
- S: **Stride** (Step size of the slide)
- P: **Padding** (Zeros added around the border)

Padding Options

1. **Valid Padding ($P = 0$):**
   - No padding.
   - The image shrinks after convolution.
   - Losing edge information.

2. **Same Padding ($P > 0$):**
   - Padding added to preserve dimensions.
   - Output Size = Input Size (if stride=1).
   - Allows for very deep networks without vanishing spatial size.

MATLAB syntax: `"Padding", "same"` vs `"Padding", 0`

Agenda

The Non-Linearity (ReLU)

Convolution is a *linear* operation (multiplication + sum). To learn complex patterns, we need non-linearity.

$$f(x) = \max(0, x) \tag{3}$$

Rectified Linear Unit (ReLU):

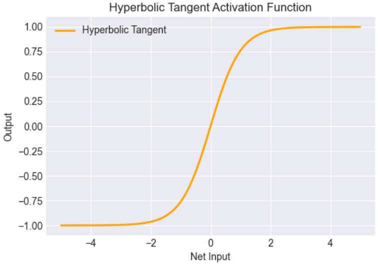
- Introduces sparsity.
- Avoids Vanishing Gradient problem (common in Sigmoid).
- Extremely computationally efficient.

Pooling Layers (Subsampling)

Purpose: Progressively reduce the spatial size (S-dimension) to reduce parameters and computation.

- **Max Pooling:** Takes the maximum value in the window. Extracts the most prominent features.
- **Average Pooling:** Takes the average. Smoothes features.

Key Property: *Translation Invariance*. Small shifts in the image don't change the outcome.



Placeholder: 2x2 Max Pooling on a 4x4 matrix

Flattening & Fully Connected

Transition to Classification

After extracting spatial features with Convolution+Pooling, we get a 3D volume (e.g., $7 \times 7 \times 512$).

1. **Flatten:** Unroll the volume into a 1D vector (size 25,088).
2. **Fully Connected (FC):** Standard MLP layers to combine features and perform classification logic.
3. **Softmax:** Probability distribution for the output classes.

Agenda

Defining a CNN in MATLAB

Using the `Layer Graph` approach (standard for modern image classification):

Listing 1: Standard VGG-Style Block

```
inputSize = [28 28 1]; % Grayscale image (S-S-C)

layers = [
    imageInputLayer(inputSize, "Name", "imageinput")

    % Convolutional Block 1
    convolution2dLayer(3, 8, "Padding", "same", "Name", "conv_1")
    batchNormalizationLayer("Name", "batchnorm_1")
    reluLayer("Name", "relu_1")

    % Downsampling
    maxPooling2dLayer(2, "Stride", 2, "Name", "maxpool_1")

    % Classification Head
    fullyConnectedLayer(10, "Name", "fc") % 10 classes
```

Understanding Data Formats: SSCB

When training CNNs, MATLAB (since R2023a) usually expects tensors in the format **SSCB**.

- **S (Spatial):** Height (H).
- **S (Spatial):** Width (W).
- **C (Channels):** Depth (Colors or Feature Maps).
- **B (Batch):** Number of images processed in parallel.

$$Tensor = 224 \times 224 \times 3 \times 32$$

Agenda

Summary

- **Convolution** exploits spatial hierarchies and shared weights, making image processing efficient.
- **ReLU** provides the necessary non-linearity for deep learning.
- **Pooling** reduces dimensionality and grants translation invariance.
- **Architecture:** It is a game of managing Dimensions (S) and Channels (C) usually decreasing spatial size while increasing channel depth deep in the network.