Convolutional Neural Networks (CNNs)

Suvinava Basak

December 19, 2024

Abstract

Artificial intelligence has been making tremendous strides in closing the gap between human and computer capabilities. Both amateurs and researchers work on many facets of the area to achieve incredible results. The field of computer vision is one of several such fields. The goal of this field is to make it possible for machines to see and perceive the world similarly to humans. They will then be able to use this knowledge for a variety of tasks, including Natural Language Processing, Media Recreation, Image and Video Recognition, Image Analysis and Classification, Recommendation Systems, and more. Deep Learning's contributions to computer vision have been developed and refined throughout time, primarily using an algorithm, **Convolutional Neural Network**.

In this book, we will dig deeper into the realm of CNN and explore many things, starting from the **historical perspective** to the building blocks of a CNN along with their **mathematical foundations**. We will also look at different CNN architectures, e.g., **AlexNet**, **VGGNet**, **GoogleNet**, etc. and some of the real-life applications. Finally, we end this book with some of the current research and future trends of CNNs.

Contents

1 Introduction

Convolutional Neural Networks (CNNs) are a type of deep learning model **designed to recognize patterns in visual data**, such as images and videos. Unlike traditional neural networks, CNNs exploit the spatial and hierarchical patterns present in the data to efficiently learn features. CNNs consist of layers that perform convolutions, mathematical operations that filter and extract features such as edges, textures, and shapes from images. These networks typically have three main types of layers: **convolutional layers**, **pooling layers** (to reduce the image size), and **fully connected layers** (for final classification). CNNs are particularly powerful for tasks like image recognition, object detection, and facial recognition, as they automatically learn relevant features from the raw input data.

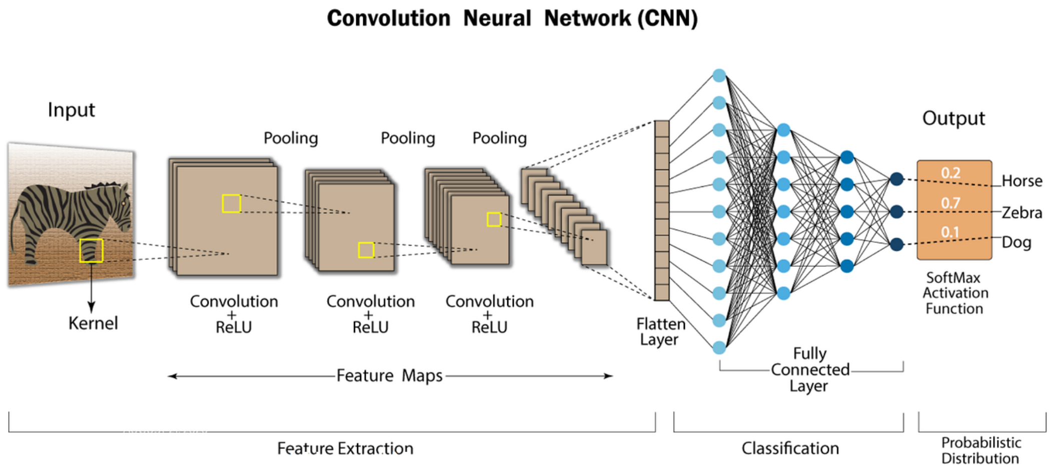

Figure 1.1: A typical Convolutional Neural Network

1.1 Key Characteristics of CNNs

- **Convolutions**: A mathematical operation that extracts patterns such as edges and textures.
- **Hierarchical Feature Learning**: CNNs learn low-level features (e.g., edges) in initial layers and complex patterns (e.g., objects) in deeper layers.
- **Parameter Sharing**: Convolutions reduce the number of parameters, making them computationally efficient.

- **Translation Invariance**: Recognize patterns regardless of their position in the input.

For instance, CNNs are capable of classifying an image of a cat even if the cat appears in different parts of the image.

1.2 Why CNNs are important in Machine Learing

A **Convolutional Neural Network (ConvNet/CNN)** can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. **The pre-processing required in a ConvNet is much lower as compared to other classification algorithms**. While in primitive methods, filters are hand-engineered; with enough training, ConvNets have the ability to learn these filters/characteristics.

CNNs have revolutionized the field of computer vision and beyond by enabling machines to:

- **Understand visual content**: Perform image classification, object detection, and segmentation.
- **Process structured data efficiently**: Learn features directly from raw input, reducing the need for manual feature engineering.
- **Achieve human-level accuracy**: Outperform traditional methods in tasks like face recognition, medical imaging, and natural language processing.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

1.3 Why ConvNets over Feed-Forward Neural Nets?

An image is nothing but a matrix of pixel values. So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes?

For extremely basic binary images, this method might show an average precision score while predicting classes but would fail miserably for complex images having pixel dependencies.

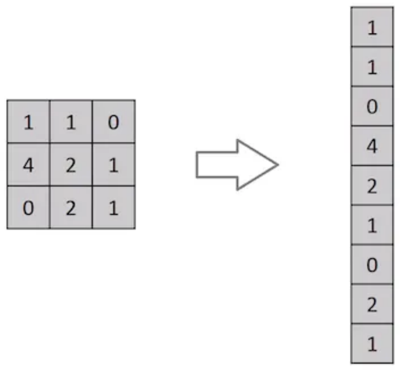

Figure 1.2: Flattening of a 3x3 image matrix into a 9x1 vector

A ConvNet can **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights.

Spatial Dependency means a pixel's value is influenced by nearby pixel's value in image. This is because generally they all belong to same color because they are from same object. **Temporal dependency** comes in videos. When a frame changes to next, if there is not a lot of movement in objects, pixel's values remain same.

Aspect	**Feed – Forward Network (ANNs)**	**Convolutional Neural Network (CNNs)**
Input Representation	Flat, requires feature extraction	Structured (e.g., images as grids of pixels)
Architecture	Fully connected layers	Convolutional + pooling layers + fully connected layers
Parameter Count	High (independent weights for each connection)	Low (shared weights through convolutions)
Feature Learning	Manual or less efficient	Automatic and hierarchical
Applications	Generic machine learning	Specialized for spatial and structured data

Table 1.1: Difference between ANNs and CNNs

1.4 Visual Representation of CNN workflow

The below image illustrates a simple CNN architecture to classify handwritten digits from MNIST dataset. The digits are passed as input images of shape 28×28 to the network and output is 10 classes (from 0 to 9) to classify the image in one of them.

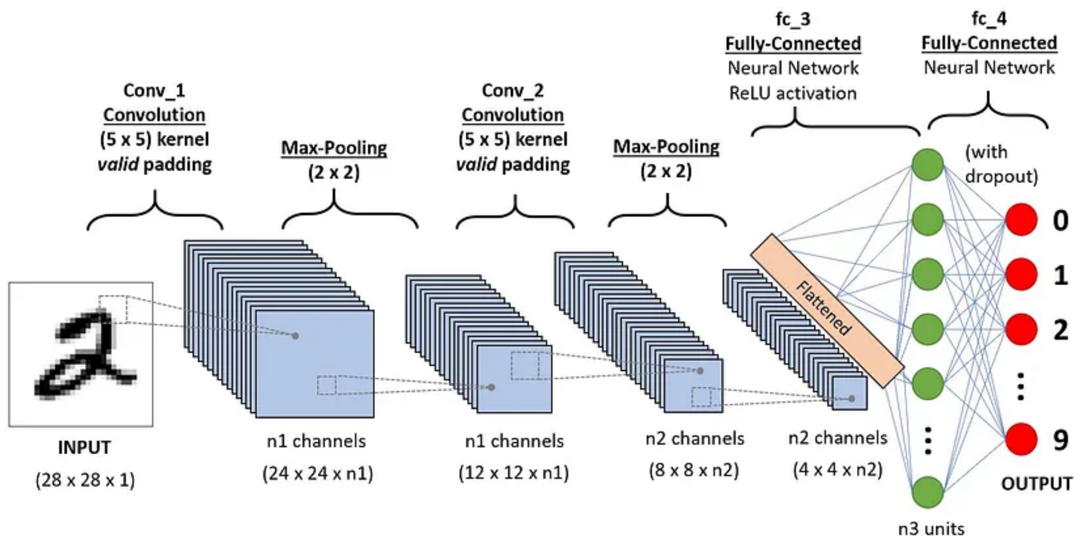

Figure 1.3: A CNN sequence to classify handwritten digits

The input image is passed through a sequence of layers (convolutions, pooling, and fully connected layers) to classify it into one of the 10 categories.

2 Historical Perspective of CNNs

2.1 Origin of CNNs

Convolutional Neural Networks (CNNs) trace their origins to early work in neural networks and pattern recognition. The concept of using convolutions for image analysis emerged from the broader study of artificial neural networks.

Early Pioneers and Inspirations

- **1970s-1980s: Neocognitron and Hebbian Learning**
  The Neocognitron, proposed by Kunihiko Fukushima in 1980, is often considered one of the first models that prefigured CNNs. It introduced the idea of hierarchical layers that could extract visual features, similar to the function of the human visual system. However, the Neocognitron lacked a mechanism for training via backpropagation.[1]

- **1986: Backpropagation and Neural Networks**
  The introduction of backpropagation by David Rumelhart, Geoffrey Hinton, and Ronald Williams in 1986 was a significant advancement. It provided a practical way to train multilayer networks by calculating gradients and adjusting weights, forming the foundation for modern CNN training.[2]

2.2 Key Milestones in CNN Development

2.2.1 LeNet (1998) - The First Major CNN

In 1998, **Yann LeCun** and his team introduced **LeNet-5**, a CNN architecture that successfully tackled the problem of digit recognition, particularly for the **MNIST dataset**. LeNet's architecture demonstrated the potential of CNNs in recognizing visual patterns.[3] [4]

Key components of LeNet-5:

- **Convolutional Layers**: Extracting low-level features such as edges.

- **Pooling Layers**: Reducing the size of the feature maps while retaining essential information.
- **Fully Connected Layers**: Final decision-making layers for classification.

2.2.2 AlexNet (2012) - A Breakthrough in Deep Learning

The introduction of **AlexNet** by **Alex Krizhevsky**, **Ilya Sutskever**, and **Geoffrey Hinton** in 2012 marked a breakthrough in deep learning, achieving a significant performance improvement in the **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**. This success was largely due to the deep architecture and the use of GPUs for training, which accelerated the computation process.[5]

Key features in AlexNet:

- **ReLU Activation**: Allowed faster convergence compared to traditional activation functions like sigmoid.
- **GPU Utilization**: Enabled the training of a large model on a massive dataset.

2.2.3 VGGNet (2014) - Deeper Architectures

VGGNet, developed by the **Visual Geometry Group (VGG)** at Oxford, demonstrated that deeper CNN architectures could provide better results. With 16 to 19 layers, VGGNet showed that increasing depth helped improve the model's ability to recognize more complex patterns.[6]

Key features of VGGNet:

- **Uniform 3x3 Convolutional Filters**: Simplified the architecture while maintaining performance.
- **Increased Depth**: Leveraging more layers enabled the model to capture more detailed features.

2.2.4 GoogleNet, InceptionV1 (2014)

GoogleNet (InceptionV1), introduced in 2014 by researchers at Google, is a groundbreaking convolutional neural network (CNN) architecture designed for efficient, large-scale image recognition tasks. Its key innovation is the **Inception module**, which performs multi-scale convolutions (1x1, 3x3, and 5x5) in parallel, allowing the network to capture features at various scales while maintaining computational efficiency. To reduce the model's complexity, 1x1 convolutions are used for dimensionality reduction before applying larger convolutions. With 22 layers, GoogleNet was deeper than previous architectures like AlexNet, yet more computationally efficient. This architecture won the

2014 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) with state-of-the-art accuracy, paving the way for further advancements in deep learning.[7]

Key features of GoogleNet:

- Introduced the **Inception module**, which performs multi-scale convolutions in parallel (1x1, 3x3, 5x5).
- Used 1x1 convolutions to reduce dimensionality and computational cost.
- Much deeper (22 layers) but computationally efficient compared to VGGNet.

2.2.5 ResNet (2015) - Solving Vanishing Gradient Problem

The introduction of **ResNet** by **Kaiming He** and colleagues in 2015 solved the problem of training very deep networks. By using **skip connections**, ResNet alleviated the vanishing gradient problem, allowing for networks with up to 152 layers.[8]

Key features of ResNet:

- **Residual Learning**: Skip connections help in passing gradients efficiently during backpropagation.
- **Batch Normalization**: Improved the stability of the training process.

2.2.6 MobileNet (2017)

MobileNet is a family of lightweight convolutional neural network (CNN) architectures designed by Google for efficient image classification on mobile and embedded devices with limited computational resources. It is optimized to achieve a good trade-off between accuracy and computational cost, making it suitable for real-time applications on devices with constraints such as smartphones, drones, and IoT devices.[9]

Key features of MobileNet:

- **Depthwise Separable Convolutions**: Reduces computation by splitting convolutions into depthwise and pointwise operations.
- **Width Multiplier**: Scales the number of channels in each layer to adjust model size and performance.
- **Resolution Multiplier**: Adjusts the input image resolution for balancing accuracy and computational cost.
- **Efficient for Real-Time Applications**: Optimized for mobile devices, suitable for tasks like image classification and object detection.

2.2.7 EfficientNet (2019)

EfficientNet is a family of convolutional neural networks introduced by **Mingxing Tan** and **Quoc V. Le** in 2019, designed to achieve state-of-the-art performance while optimizing efficiency. It uses a novel **compound scaling method** that uniformly scales a network's depth, width, and input resolution in a balanced way to maximize accuracy and efficiency. Unlike previous models that scale only one dimension (e.g., depth or width), EfficientNet finds the best combination of all three. The base model, **EfficientNet-B0**, is built using a lightweight architecture with **MBConv blocks** (inspired by MobileNet), and larger variants (B1–B7) scale up for better accuracy while maintaining computational efficiency. EfficientNet, through proper scaling, achieves higher accuracy on benchmarks like ImageNet with significantly fewer parameters and FLOPs compared to models like ResNet and Inception, making it ideal for both resource-constrained devices and high-performance applications.[10]

Key features of EfficientNet:

- Introduced a **compound scaling method** to optimize the trade-off between network depth, width, and resolution.
- Achieved state-of-the-art performance on ImageNet with fewer parameters and FLOPs compared to ResNet and Inception-based models.
- Includes variants from EfficientNet-B0 to B7, each scaling up computational cost and accuracy.

2.3 CNNs in the Modern Era

The Rise of Transfer Learning and Pretrained Models
With the advancements in computational power and large-scale datasets, CNNs have continued to improve. Pretrained models such as **Inception**[7] and **EfficientNet**[10] have played a major role in enabling transfer learning, where models trained on large datasets are adapted for specific tasks.

Transfer Learning
Transfer learning allows practitioners to leverage models trained on large datasets (like ImageNet) and fine-tune them for new, domain-specific tasks. This approach saves time and computational resources while achieving high performance.

2.4 Relevance of CNNs to Artificial Neural Networks (ANNs)

CNNs are an extension of **traditional Artificial Neural Networks (ANNs)**, which generally rely on fully connected layers for classification. Unlike ANNs, CNNs use shared weights, convolutions, and pooling layers to process spatial information more efficiently.

Key differences from ANN:

- **Weight Sharing**: In CNNs, filters (kernels) are shared across the entire input, making them computationally more efficient compared to fully connected layers in ANNs.[11]
- **Hierarchical Feature Learning**: CNNs learn low-level features in the initial layers and combine them into more abstract concepts in deeper layers. This hierarchical feature learning is not a core principle of traditional ANNs.

3 Building Blocks of CNNs

To understand how CNNs work, it's crucial to break down their architecture into core components. These blocks — **Convolutional Layers**, **Pooling Layers**, **Activation Functions**, **Fully Connected Layers**, and **Dropout Layers** — work together to extract and classify patterns in data.

3.1 Convolutional Layers: The Core Component

The convolutional layer is the most fundamental building block of CNNs. It is responsible for detecting spatial features (e.g., edges, textures) by applying a **kernel** or **filter** to the input data.

3.1.1 How Convolution Works

Bibliography

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proceedings of the IEEE Conference on Neural Networks, 1990*, pp. 396–404, 1990.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, pp. 1097–1105, 2012.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1409.1556.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1–9, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, vol. abs/1704.04861, 2017.

[10] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[11] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

Further references

Apart from the above references, several YouTube videos (by Brandon Rohrer [1] [2], Serrano.Academy, 3Blue1Brown, deeplizard, The Julia Programmina Language, MIT, Standford University) are relevant to know more about Convolutional Neural Networks in detail.

There are also some blogs on `medium.com` which can serve as a good starting point to delve deeper into the realm of CNN:

- *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way* by Sumit Saha
- *An Introduction to different Types of Convolutions in Deep Learning* by Paul-Louis Pröve
- *Gentle Dive into Math Behind Convolutional Neural Networks* by Piotr Skalski
- *Intuitively Understanding Convolutions for Deep Learning* by Irhum Shafkat

Happy Learning!