INF721 - Deep Learning
L11: Convolutional Neural Networks (Part II)

Prof. Lucas N. Ferreira
Universidade Federal de Viçosa

2024/2

1 Introduction

In the previous lecture, we covered the fundamental concepts of Convolutional Neural Networks (CNNs), including filters, convolutions, padding, stride, and convolutions over volumes. In this lecture, we'll build upon these concepts and explore more advanced CNN architectures and components.

2 Pooling Layers

In addition to convolutions layers, pooling layers make up another important building block in CNNs and they serve two primary purposes:

1. Reducing the dimensions of feature maps
2. Summarizing features present in a region

Unlike convolutional layers, pooling layers:

- Perform predefined computations, so they do not have any learnable parameters
- Are applied independently to each channel in the input volume

2.1 Max Pooling

Max pooling is the most commonly used pooling operation in modern CNNs for computer vistion. It works by:

1. Taking a filter of size $f \times f$
2. Sliding it over the input with stride s
3. Taking the maximum value in each window

For example, the figure below show a max pooling operation in a 5x5 image with a 3×3 filter and stride of 1:

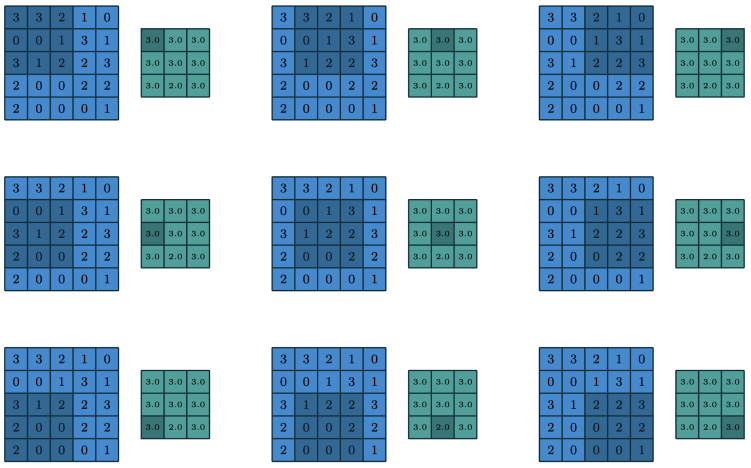

Figure 1: Max pooling operation

2.2 Average Pooling

Average pooling is less frequently used in modern architectures for computer vision problems. It operates similarly to max pooling but takes the average of values in each window instead of the maximum. For example, the figure below shows an average pooling operation in a 5x5 image with a 3×3 filter and stride of 1:

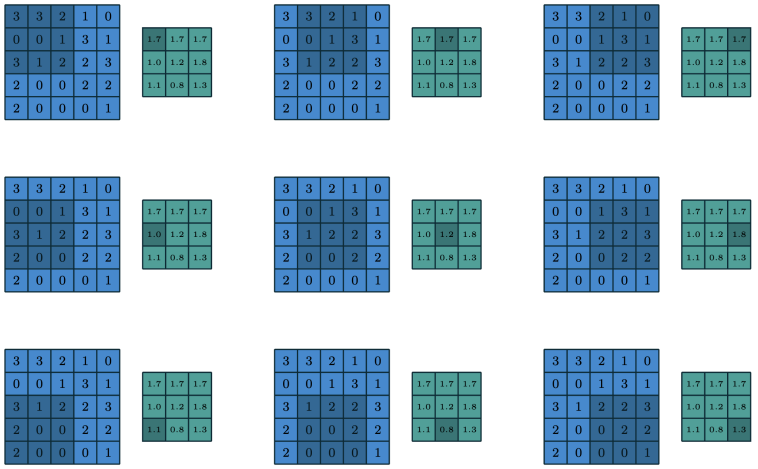

Figure 2: Average pooling operation

2.3 Pooling Over Volumes

When applying pooling to volumes (multi-channel inputs):

- Each channel is pooled independently
- The number of channels remains unchanged
- The spatial dimensions (height and width) are reduced according to the pooling parameters

3 Classic CNN Architectures

3.1 LeNet-5 (1998)

LeNet-5, developed by Yann LeCun et al., was one of the first successful CNN architectures. It was designed for handwritten digit classification.

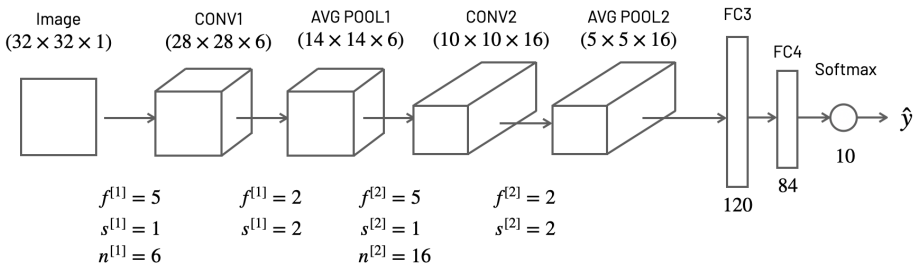

Figure 3: LeNet-5 Architecture

This network has approximately 60,000 parameters used average pooling and tanh activation functions.

3.2 AlexNet (2012)

AlexNet, developed by Alex Krizhevsky et al., won the ImageNet competition in 2012 and is often considered the breakthrough that popularized deep learning in computer vision.

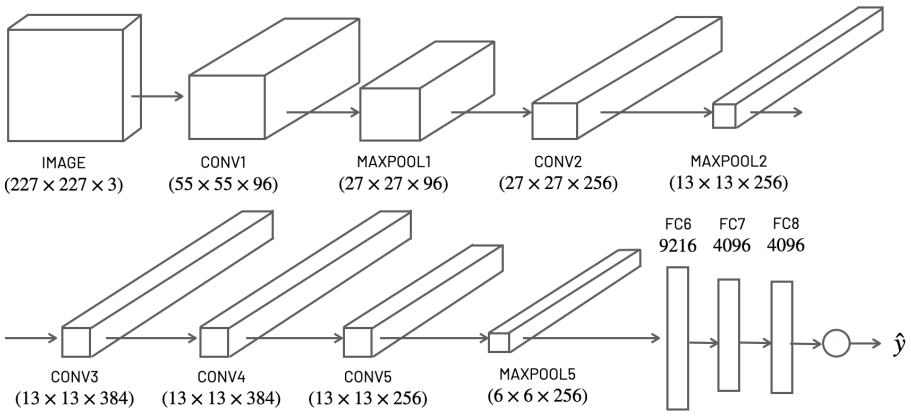

Figure 4: AlexNet Architecture

This network has approximately 60M parameters and introduced several key innovations that are now common in deep learning:

1. Use of ReLU activation functions
2. GPU implementation
3. Dropout regularization

3.3 VGG-16 (2015)

VGG-16 introduced a more systematic approach to CNN architecture design:
Key principles:

1. All convolutional layers have 3×3 filters with stride 1 and padding 1
2. All max pooling layers have 2×2 filters with stride 2
3. Number of filters doubles after each max pooling layer

This network has approximately 138M parameters and is known for its simplicity and uniform architecture.

4 Residual Networks (ResNet)

As networks became deeper, researchers encountered a counterintuitive problem: adding more layers sometimes led to worse performance, even on the training set. This was not due to overfitting but rather to optimization difficulties. As the size of the network increased, the gradient signal became too weak or too strong, making it difficult to train the network effectively. These problems are know as the vanishing gradient and exploding gradient problems, respectively. ResNet introduced residual connections (skip connections) to address this issue.

4.1 Residual Connections

Residual connections are a way to "skip" layers in a network. In a plain network, the output of a layer $l+2$ is given by:

$$z^{[l+1]} = W^{[l+1]}a^{[l]} + b^{[l+1]}$$
$$a^{[l+1]} = g(z^{[l+1]})$$
$$z^{[l+2]} = W^{[l+2]}a^{[l+1]} + b^{[l+2]}$$
$$a^{[l+2]} = g(z^{[l+2]})$$

In a residual network, the output of layer $l+2$ is given by:

$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

, where $a^{[l]}$ is the input to layer $l+2$ and $z^{[l+2]}$ is the output of layer $l+2$. The figure below shows a residual block with two layers:

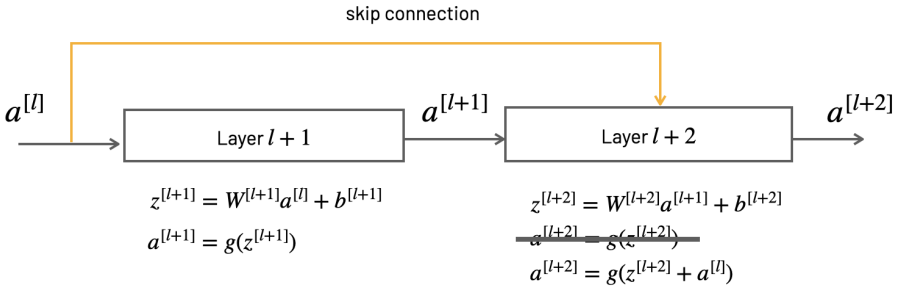

Figure 5: A Residual Block

Residual connections allow a residual block to easily learn the identity function, so adding more blocks does not hurt the performance of the network. This enables the training of very deep networks with hundreds of layers.

4.2 ResNet Architecture

A ResNet architecture is composed of several residual blocks, with each block containing multiple convolutional layers. The figure below shows a generic architecture of a ResNet model:

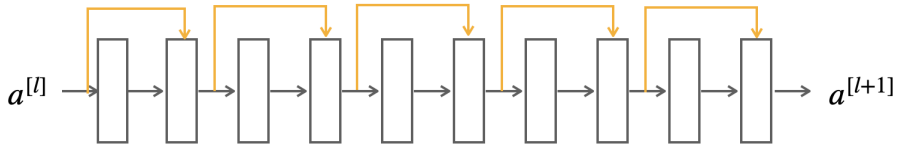

Figure 6: A Generic ResNet Architecture

5 Practical Considerations

5.1 Design Patterns

Common patterns observed in modern CNNs:

1. Spatial dimensions (height x width) generally decrease through the network
2. Number of channels generally increases through the network
3. Pooling layers are used strategically to reduce spatial dimensions
4. Final layers are typically fully connected

5.2 Implementation Tips

1. Start with proven architectures
2. Consider computational resources when choosing architecture size
3. Use appropriate stride values based on input image size
4. Monitor training and validation metrics carefully
5. Consider using pre-trained models when possible

6 Conclusion

Modern CNN architectures have evolved from simple designs like LeNet-5 to complex networks with residual connections. Each innovation has addressed specific challenges:

- LeNet-5: Basic CNN structure
- AlexNet: Scale and practical implementation
- VGG: Systematic design
- ResNet: Very deep networks

The field continues to evolve, with new architectures and techniques being developed regularly.