Unit 5 Multilayer Perceptrons

Convolutional Network

Kiran Bagale

August 2025

Convolutional Networks: Introduction

- Special class of multilayer perceptrons for **pattern classification**
- Neurobiologically motivated by Hubel and Wiesel (1962, 1977)
  - Locally sensitive neurons in visual cortex
  - Orientation-selective neurons
- Designed to recognize **two-dimensional shapes** with high invariance to:
  - Translation and Scaling
  - Skewing and Other forms of distortion

Three Key Structural Constraints

1. **Feature Extraction**
   - Local receptive fields
   - Forces extraction of local features
2. **Feature Mapping**
   - Multiple feature maps per layer
   - Weight sharing within feature maps
3. **Subsampling**
   - Local averaging and resolution reduction
   - Reduces sensitivity to distortions

Constraint 1: Feature Extraction

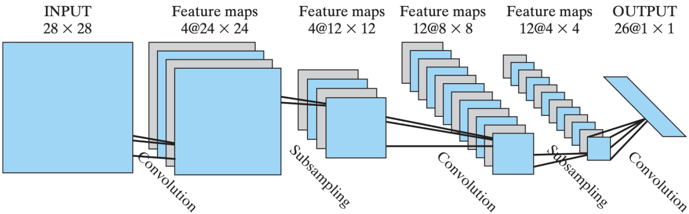

Fig: Convolutional network for image processing such as handwriting recognition. (Reproduced with permission of MIT Press.)

Local Receptive Fields

- Each neuron takes inputs from a **local receptive field**
- Forces extraction of local features
- Once extracted, exact location becomes less important
- Preserves relative position of features

Key Insight

Local feature extraction mimics biological visual processing

Constraint 2: Feature Mapping

Weight Sharing Architecture

- Each layer composed of multiple **feature maps**
- Neurons in same feature map share identical weights
- Implemented through **convolution** with small kernels

Benefits

- **Shift invariance** through convolution operations
- **Parameter reduction** through weight sharing
- Improved generalization capability

Constraint 3: Subsampling

Local Averaging and Subsampling

- Follows each convolutional layer
- Performs local averaging
- Reduces feature map resolution

Effects

- Reduces sensitivity to shifts
- Increases robustness to distortions
- Creates hierarchical feature representation

Example: Handwritten Character Recognition

Network Architecture (Figure 4.23)

- **Input:** 28×28 sensory nodes
- **Task:** Recognize handwritten characters
- **Structure:** Alternating convolution and subsampling layers

Layer Configuration

- Input layer: 28×28 nodes
- 4 hidden layers (alternating conv/subsample)
- Output layer: 26 neurons (26 characters)

Architecture Details: Layers 1-2

Hidden Layer 1: First Convolution

- 4 feature maps of 24×24 neurons
- Receptive field: 5×5 per neuron

Hidden Layer 2: First Subsampling

- 4 feature maps of 12×12 neurons
- Receptive field: 2×2 per neuron
- Trainable coefficient and bias
- Sigmoid activation function

Architecture Details: Layers 3-5

Hidden Layer 3: Second Convolution

- 12 feature maps of 8×8 neurons
- Connections from multiple previous feature maps

Hidden Layer 4: Second Subsampling

- 12 feature maps of 4×4 neurons

Output Layer: Final Convolution

- 26 neurons (one per character)
- Receptive field: 4×4 per neuron

The "Bipyramidal" Effect

Pattern Across Layers

As we progress through the network:

- **Number of feature maps** increases
- **Spatial resolution** decreases

Biological Inspiration

- Inspired by Hubel and Wiesel's findings
- "Simple" cells followed by "complex" cells
- Hierarchical feature processing

Remarkable Parameter Efficiency

Network Statistics

- **Synaptic connections:** $\approx 100,000$
- **Free parameters:** $\approx 2,600$
- **Reduction factor:** $\approx 38:1$

Achieved Through

- **Weight sharing** across feature maps
- Reduced learning machine capacity
- Improved generalization ability

Training

All parameters learned via stochastic backpropagation!

Additional Advantages

Parallel Implementation

- Weight sharing enables parallel processing
- Advantage over fully connected MLPs
- Efficient computational implementation

Automatic Feature Learning

- Network learns to extract features automatically
- No manual feature engineering required
- Supervised learning through backpropagation

Key Lessons from Convolutional Networks

Lesson 1: Power of Constraints

A multilayer perceptron of manageable size can learn complex, high-dimensional, nonlinear mappings by **constraining its design** through incorporation of prior knowledge.

Lesson 2: Learning Effectiveness

Synaptic weights and bias levels can be effectively learned by cycling the simple **backpropagation algorithm** through the training sample.

Bottom Line

Structured constraints + effective learning = powerful pattern recognition