Introduction to CNN
Introduction to CNN

Agenda
Introduction to CNN

Agenda

How do we recognize digit? How does the CNN?
Convolutions for feature extraction
Convolutional Neural Network (CNN)
Terminology
CNN explainer
How do we recognize the digit?
How do we recognize the digit?

By summarized patterns from experience
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit
How do we recognize the digit?

By summarized patterns from experience.

Patterns: the features shared by all samples of one digit.
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit
How do we recognize the digit?

By summarized patterns from experience
Patterns: the features shared by all samples of one digit
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
How do we recognize the digit?

By summarized patterns from experience.

Patterns: the features shared by all samples of one digit.

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
“3” — the “connection between up and bottom”

![Image showing digit recognition with highlighted features](image-url)
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
“3” — the “connection between up and bottom”
How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
“3” — the “connection between up and bottom”

features
Patterns

Input Layer

feature 1

feature 2

feature 3
Patterns

feature 1

feature 2

feature 3
Patterns

feature 1

feature 2

feature 3
Patterns

feature 1

feature 2

feature 3
It is a “7”
Neural Network is an algorithm used to recognize patterns in data. Convolutional Neural Network uses convolutions to extract image features and recognize feature patterns.
Neural Network is an algorithm used to recognize patterns in data.
Convolutional Neural Network uses convolutions to extract image features and recognize feature patterns.

Extract image features -> find feature patterns for each category
Extract Features
Convolution

https://setosa.io/ev/image-kernels/
Extract Features

Convolution

https://setosa.io/ev/image-kernels/
Extract Features
Convolution

- blur
- sobel
- outline
- emboss
- sharpen
Extract Features

Convolution

Edges
Extract Features
Convolution

Edges
Corners
Extract Features

Convolution

blur
sobel
outline
emboss
sharpen

Edges
Corners
Lines
Extract Features

Convolution

Blur
Sobel
Outline
Emboss
Sharpen

Edges
Corners
Lines
Circles
Extract Features

Convolution

- blur
- sobel
- outline
- emboss
- sharpen

Edges
Corners
Lines
Circles
...
Extract Features
Convolution

Edges
Corners
Lines
Circles
...

Basic Features

blur
sobel
outline
emboss
sharpen
Extract Features
Convolution for high-level feature
Extract Features
Convolution for high-level feature

Columns
Extract Features
Convolution for high-level feature

Columns
Statues
Extract Features
Convolution for high-level feature
Extract Features
Convolution for high-level feature

Columns
Statues
...
Complex Features
Features in different levels
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network

- sobel ➔
- blur ➔
- outline ➔
- sobel ➔
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network

- sobel
- blur
- outline
- sobel

+
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map

Convolutional Neural Network
Feature Map
Convolutional Neural Network
Feature Map
Convolutional Neural Network

Different processed versions of the original images
Training CNN

Customized Convolution Filter (kernel)

Change the kernel values

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.7</td>
<td>-1.2</td>
<td>0.7</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>-1.9</td>
</tr>
</tbody>
</table>

[Image of customized convolution filter (kernel)]
**Training CNN**

**Customized Convolution Filter (kernel)**

Change the kernel values

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>-1.2</td>
<td>0.7</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>-1.9</td>
</tr>
</tbody>
</table>

(custom)
Training CNN
Customized Convolution Filter (kernel)
Training CNN
Customized Convolution Filter (kernel)
Training CNN
Customized Convolution Filter (kernel)

Adjust the weights according to the recognition performance
Training CNN

Customized Convolution Filter (kernel)

Adjust the weights according to the recognition performance

Feature map changes until weights adjusting finished
Convolution Kernels (Parameters to be learned)

Convolutional Layer 1
Input Image (28x28 pixels)
Filter-Weights (5x5 pixels)
(16 channels)
(14x14 pixels)

Convolutional Layer 2
Filter-Weights (5x5 pixels)
16 of these ...
(16 channels)
(36 channels)
(7x7 pixels)

Fully-Connected Layer
Output Layer
Class

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>
Convolution Kernels (Parameters to be learned)

Features

Input Image (28x28 pixels) → Convolutional Layer 1 → Convolutional Layer 2 → Fully-Connected Layer → Output Layer

Filter-Weights (5x5 pixels) → (14x14 pixels) → Filter-Weights (5x5 pixels) → (7x7 pixels) → Fully-Connected Layer

(16 channels) → (16 of these ...) → (36 channels) → (128 features) → (10 features)

Class

0
1
2
3
4
5
6
7
8
9
Convolution Kernels
(Parameters to be learned)

Features

Feature map

Convolutional Layer 1

Filter-Weights
(5x5 pixels)

Input Image
(28x28 pixels)

16 of these...
(16 channels)

Convolional Layer 2

Filter-Weights
(5x5 pixels)

(14x14 pixels)

(7x7 pixels)

(36 channels)

Fully-Connected
Layer

Output
Layer

Class

0
1
2
3
4
5
6
7
8
9

(128 features)
(10 features)
Convolution Kernels
(Parameters to be learned)

Features

Feature map

Feature Pattern recognition

Convolutional Layer 1
- Input Image (28x28 pixels)
- Filter-Weights (5x5 pixels)
- (14x14 pixels)
- (16 channels)

Convolutional Layer 2
- Filter-Weights (5x5 pixels)
- (7x7 pixels)
- (36 channels)

Fully Connected Layer
- (128 features)
- (10 features)

Output Layer
- Class
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9
Convolution Kernels (Parameters to be learned)

Features

Feature map

Feature Pattern recognition

Convolutional Layer 1

Input Image (28x28 pixels)

Filter-Weights (5x5 pixels)

Feature map (14x14 pixels)

(16 channels)

Convolutional Layer 2

Filter-Weights (5x5 pixels)

(7x7 pixels)

Features (7777777)

16 of these...

(36 channels)

Fully Connected Layer

Output Layer

Class

0
1
2
3
4
5
6
7
8
9

(128 features)

(10 features)
Convolution Kernels (Parameters to be learned)

Features

Feature map

Feature Pattern recognition

Recognition Performance

Input Image (28x28 pixels)

Convolutional Layer 1

Filter-Weights (5x5 pixels)

(14x14 pixels)

(16 channels)

Convolutional Layer 2

Filter-Weights (5x5 pixels)

(7x7 pixels)

(36 channels)

Fully Connected Layer

Output Layer

(128 features)

(10 features)

Class

0

1

2

3

4

5

6

7

8

9

(77777777)
Convolutional Kernels
(Parameters to be learned)

Feature map

Feature Pattern recognition

Recognition Performance

Input Image
(28x28 pixels)

Convolutional Layer 1

Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)

Convolutional Layer 2

Filter-Weights
(5x5 pixels)
(7x7 pixels)
(36 channels)

Fully Connected Layer

Output Layer

(128 features) (10 features)

Class
0
1
2
3
4
5
6
7
8
9
Convolution Kernels (Parameters to be learned)

Features

Feature map

Feature Pattern recognition

Recognition Performance

Training

Model
- Num of layers (stages)
- Num of channels (filters) in each layer

Convolutional Layer 1
- Input Image (28x28 pixels)
- Filter-Weights (5x5 pixels)
  - (14x14 pixels)
  - (16 channels)

Convolutional Layer 2
- Filter-Weights (5x5 pixels)
  - (16 of these ...)
  - (128 features)

Fully Connected Layer
- (36 channels)

Output Layer
- (10 features)
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Class 7
- Recognition Performance

Model
- Num of layers (stages)
- Num of channels (filters) in each layer
https://poloclub.github.io/cnn-explainer/#article-convolution
Other operations

Pooling Layer (reduce the resolution)

Flatten Layer (Convert 2D data to 1D for neural nets)
Other operations
Other operations
Other operations

Pooling Layer
reduce the resolution

max pooling

average pooling
Other operations

Pooling Layer
reduce the resolution
Other operations

Pooling Layer
- Reduce the resolution

Flatten Layer
- Convert 2D data to 1D for neural nets
Other operations

Pooling Layer
- Reduce the resolution

Flatten Layer
- Convert 2D data to 1D for neural nets
Other operations

Pooling Layer
- Reduce the resolution

Flatten Layer
- Convert 2D data to 1D for neural nets

Fully-Connected Layer
- Neural Nets for pattern recognition
Other operations

ReLu Activation

max(pixel_value, 0)
Add non-linearity