The Receptive Field in CNNs

Jonathan Kobold
5.3.2020

Laboratoire IBISC
Université Paris Saclay, Université Evry Val d’Essonne
Convolutional neural networks (CNNs) are currently the number one tool for pattern recognition

- **Spatial patterns**
  - Object detection from images
  - Semantic segmentation
  - Image classification
- **Sound patterns**
  - Natural Language Processing (NLP)
  - Music recommender systems
  - Automatic subtitles
  - Voice control
- **Temporal patterns**
  - Process control
  - Automatic translation
CNNs

Left: Resnet (Kaiming et al. 2015)
Right: U-Net (Ronneberger et al. 2015)
Operations in a CNN

- Convolutions
  - Standard Convolutions
  - Dilated Convolutions
  - Strided Convolutions
- Pooling Operations
  - Maximum Pooling
  - Average Pooling
- Regularisation Operations
  - Dropout
  - Batch Normalisation
- Activation Functions
Discrete Convolution

Kernel size: 3  Dilation rate: 1  Stride: 1

\[ y_3 = x_3 w_1 + x_4 w_2 + x_5 w_3 \]
Discrete Convolution

Kernel size: 3  Dilation rate: 1  Stride: 1

\[ y_1 = x_1w_1 + x_2w_2 + x_3w_3 \]
Discrete Convolution

Kernel size: 3  Dilation rate: 1  Stride: 1

\[ y_2 = x_2 w_1 + x_3 w_2 + x_4 w_3 \]
Discrete Convolution

Kernel size: 3  Dilation rate: 1  Stride: 1

\[ y_3 = x_3 w_1 + x_4 w_2 + x_5 w_3 \]
Discrete Convolution

Kernel size: 3      Dilation rate: 1      Stride: 1

\[ y_3 = x_3 w_1 + x_4 w_2 + x_5 w_3 \]
Discrete Convolution

Kernel size: 5  Dilation rate: 1  Stride: 1

\[ y_1 = x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4 + x_5 w_5 \]
Discrete Convolution

Kernel size: 5      Dilation rate: 1      Stride: 1

\[ y_1 = x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4 + x_5 w_5 \]
Convolution
Dilated Convolution

Kernel size: 3  Dilation rate: 2  Stride: 1

\[ y_1 = x_1 w_1 + x_3 w_2 + x_5 w_3 \]
Dilated Convolution

Kernel size: 3  Dilation rate: 2  Stride: 1

\[ y_1 = x_1 w_1 + x_3 w_2 + x_5 w_3 \]
Kernel size: 3  Dilation rate: 3  Stride: 1

\[ y_1 = x_1 w_1 + x_4 w_2 + x_7 w_3 \]
Dilated Convolution

Kernel size: 3  Dilation rate: 3  Stride: 1

\[ y_1 = x_1 w_1 + x_3 w_2 + x_5 w_3 \]
Maximum Pooling

Kernel size: 3    Dilation rate: 1    Stride: 1

\[ y_1 = \max\{x_1, x_2, x_3\} \]
Maximum Pooling

Kernel size: 3  Dilation rate: 1  Stride: 1

\[ y_3 = x_3w_1 + x_4w_2 + x_5w_3 \]
Maximum Pooling

Kernel size: 2  Dilation rate: 1  Stride: 2

\[ y_1 = \max\{x_1, x_2, x_3\} \]
Maximum Pooling

Kernel size: 2  Dilation rate: 1  Stride: 2

\[ y_3 = x_3 w_1 + x_4 w_2 + x_5 w_3 \]
Operations in a CNN

- Sliding windows which perform:
  - Weighted Sum
  - Maximum
  - Average

- Point-wise Operations
  - Dropout
  - Batch Normalisation
  - Activation Functions
A simple CNN

Output Neuron

Input Image
A simple CNN
A simple CNN
A simple CNN
A simple CNN
A simple CNN
A simple CNN
Receptive Field
The **receptive field** is the number of pixels in the input which impact the value of the output of a CNN.
Why is the Receptive Field important?

Example: Cyclist and pedestrian detection for autonomous driving
Why is the Receptive Field important?

Receptive Field: $100 \times 100$ pixels
Why is the Receptive Field important?

Cyclist and pedestrian are not distinguishable with a $100 \times 100$ receptive field. A detection of heads is possible!

Pedestrian  
Cyclist
Why is the Receptive Field important?

Receptive Field: $324 \times 444$ pixels
Why is the Receptive Field important?

Cyclist and pedestrian are distinguishable with a $324 \times 444$ receptive field!
Considerations for choosing the receptive field size:

- Decides the size of the input’s context available to the model
- Too small receptive fields make detection tasks impossible
- Too large receptive fields may lead to target objects disappearing because they are too small

→ The right size depends on the application

Choose the receptive field in the same order of magnitude as the objects to detect! Within this range larger receptive field sizes lead to larger models.
Methods for Extending the Receptive Field

Stacking convolution layers (Stack)
Methods for Extending the Receptive Field

Stacking convolution layers (Stack)
Resnet (Kaiming et al. 2015)

VGG-19

- Output size: 224
  - 3x3 conv, 64
  - Pool, /2
  - 3x3 conv, 128
  - Pool, /2
  - 3x3 conv, 256
  - Pool, /2
  - 3x3 conv, 256
  - Pool, /2
  - 3x3 conv, 256

34-layer plain

- Output size: 112
  - 7x7 conv, 64, /2
  - Pool, /2
  - 3x3 conv, 64
  - 3x3 conv, 64
  - 3x3 conv, 64
  - 3x3 conv, 64

34-layer residual

- Output size: 56
  - 7x7 conv, 64, /2
  - Pool, /2
  - 3x3 conv, 64
  - 3x3 conv, 64
  - 3x3 conv, 64
  - 3x3 conv, 64
  - 3x3 conv, 64
Methods for Extending the Receptive Field

Alternating convolution and maxpool layers (Pyramid)
Methods for Extending the Receptive Field

Alternating convolution and maxpool layers (Pyramid)

- conv 3x
- maxpool 2x, stride 2
- conv 3x
- maxpool 2x, stride 2
- conv 3x
- maxpool 2x, stride 2
- conv 3x
- maxpool 2x, stride 2
- conv 3x
- input
Methods for Extending the Receptive Field

A single convolution with large kernel (Single)

```
conv 18x
input
```
Receptive field: $18 \times 18$

<table>
<thead>
<tr>
<th>Method</th>
<th>Stack</th>
<th>Pyramid</th>
<th>Single</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>76</td>
<td>27</td>
<td>324</td>
</tr>
</tbody>
</table>
Methods for Extending the Receptive Field

A trous spatial pyramid pooling (ASPP) (Chen et al. 2017)
Methods for Extending the Receptive Field

Transfer block (Transfer) (Kobold 2019)

![Diagram of the Transfer block](image-url)
Methods for Extending the Receptive Field

Sequential methods:

- Stacks
- Pyramids

→ Rigid structures, no tuning possible

Parallel methods:

- ASPP
- Transfer Block

→ Tunable, number of parameters can be adapted to the actual problem

→ Scale well with parallel (GPU) computing architectures
Effective Receptive Field

conv
pool
conv
pool
conv

conv
pool
conv

conv
pool
conv

conv
pool
conv

conv
pool
conv

conv
pool
conv

conv
pool
conv
Effective Receptive Field

conv
pool
conv
pool
conv
Effective Receptive Field

conv
pool
conv
pool
conv
Effective Receptive Field

Stack

Pyramid
Effective receptive fields gained from simulations. Networks with 20 layers (Luo et al. 2016)
Effective Receptive Field  Example: Mushroom Classification

Original  Stack  Pyramid
Effective Receptive Field

Resnet (Kaiming et al. 2015)    Inception (Szegedy et al. 2014)
The effective receptive field (EFR) describes the impact of the input pixels on the output.

\[ ERF_i = \frac{\partial \text{Output}}{\partial x_i} \]
The ERF

- follows a Gaussian distribution
- cannot be modified in sequential models
- can have its standard deviation modified in parallel model
- is a better estimate of a model's capacity than the receptive field alone
Summary

- CNNs
- Receptive Field
  - Importance
  - Stack
  - Pyramid
  - Single
  - ASPP
  - Transfer Block
- Effective Receptive Field
Thank You for Your Attention