Convoluted Convolutional Neural Networks

Helen Ngo

06 December 2016

What is a convolutional neural network?

Overview [edit]

Convolutional neural networks model animal visual perception, and can be applied to visual recognition tasks.

Image recognition [edit]

Convolutional neural networks (CNNs) consist of multiple layers of receptive fields. These are small[clarification needed] neuron collections which process portions of the input image. The outputs of these collections are then tiled[how?] so that their input regions overlap, to obtain a better[how?] representation of the original image; this is repeated for every such layer. Tiling allows CNNs to tolerate translation of the input image.[8]

Convolutional networks may include local or global pooling layers[clarification needed], which combine the outputs of neuron clusters.[9][10] They also consist of various combinations of convolutional and fully connected layers, with pointwise nonlinearity applied at the end of or after each layer.[11] A convolution operation on small regions of input is introduced[how?] to reduce the number of free parameters and improve generalization[how?]. One major advantage of convolutional networks is the use of shared weight in convolutional layers, which means that the same filter (weights bank) is used for each pixel in the layer; this both reduces memory footprint and improves performance[how?].[3]

Caffe: Image Processing

Caffe Demos

The Caffe neural network library makes implementing state-of-the-art computer vision systems easy.

Classification

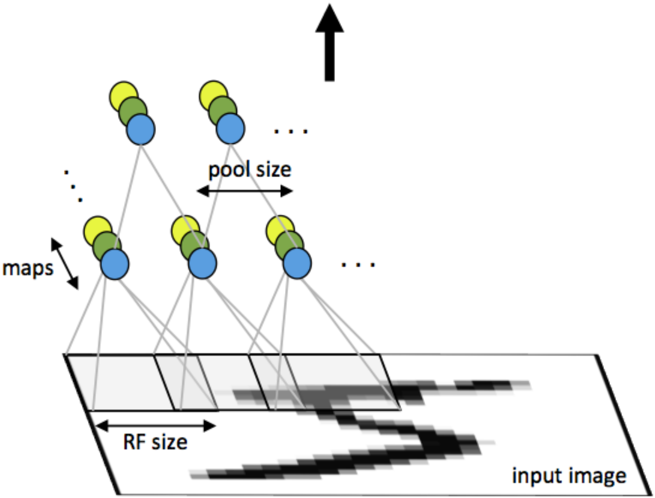
Click for a Quick Example

Maximally accurate | Maximally specific

bag	1.19015
mailbag	0.90366
container	0.53076
backpack	0.43018
covering	0.40157

CNN took 0.071 seconds.

Convolutional Neural Network



- Input: $w \times h \times d$ image

- Input: $w \times h \times d$ image
  1. d is the depth or number of channels

- Input: $w \times h \times d$ image
    1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of

- Input: $w \times h \times d$ image
  1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of
  1. convolutional layers

- Input: $w \times h \times d$ image
    1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of
    1. convolutional layers
    2. fully connected layers

- Input: $w \times h \times d$ image
  1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of
  1. convolutional layers
  2. fully connected layers
- Convolutional layers

- Input: $w \times h \times d$ image
    1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of
    1. convolutional layers
    2. fully connected layers
- Convolutional layers
    1. $k \times k$ patch where $k < w \leq h$

- Input: $w \times h \times d$ image
    1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of
    1. convolutional layers
    2. fully connected layers
- Convolutional layers
    1. $k \times k$ patch where $k < w \leq h$
    2. $k \times k \times d$

- Input: $w \times h \times d$ image
    1. d is the depth or number of channels
- A Convolutional Neural Network (CNN) is comprised of
    1. convolutional layers
    2. fully connected layers
- Convolutional layers
    1. $k \times k$ patch where $k < w \leq h$
    2. $k \times k \times d$
    3. m filters

Convolutional Layers

```
for w in 1..W
  for h in 1..H
    for x in 1..K
      for y in 1..K
        for m in 1..M
          for d in 1..D
            output(w, h, m) += input(w+x, h+y, d) * filter(m, x, y, d)
          end
        end
      end
    end
  end
end
```

Mathematical Convolution

- $$w \times h \times d \times k \times k \times m$$
  $$\therefore$$
  $$\text{6D}$$

Mathematical Convolution

- $$w \times h \times d \times k \times k \times m$$
  $$\therefore$$
  $$\text{6D}$$
- Convolution: a function derived from two given functions by integration that expresses how the shape of one is modified by the other.

Mathematical Convolution

- The convolution $f * g$ of two functions $f(t)$ and $g(t)$ is the function defined by

$$(f * g)(t) = \int_0^t f(t-u)g(u)du.$$

Mathematical Convolution

- The convolution $f * g$ of two functions $f(t)$ and $g(t)$ is the function defined by

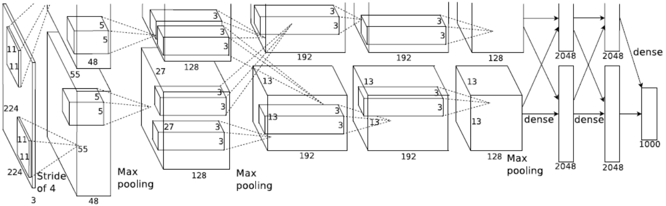
$$(f * g)(t) = \int_0^t f(t-u)g(u)du.$$

- Expresses the amount of overlap of one function g as it is shifted over another function f

Mathematical Convolution

- The convolution $f * g$ of two functions $f(t)$ and $g(t)$ is the function defined by

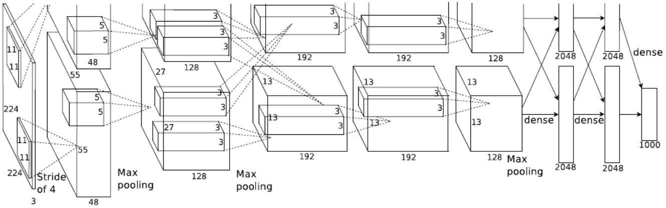
$$(f * g)(t) = \int_0^t f(t-u)g(u)du.$$

- Expresses the amount of overlap of one function g as it is shifted over another function f
- 

Mathematical Convolution

- The convolution $f * m$ of two functions image $f(D)$ and filters $m(D)$ is the function defined by

$$(f * m)(D) = \int_0^D f(D-u)m(D)dD.$$

- 

Unrolling Convolutional Layers

1. Lay out the patches in a $(W * H, K * K * D)$ matrix.

Unrolling Convolutional Layers

1. Lay out the patches in a $(W * H, K * K * D)$ matrix.
2. Consider the filters to be in a $(M, K * K * D)$ kernel matrix.

Unrolling Convolutional Layers

1. Lay out the patches in a $(W * H, K * K * D)$ matrix.
2. Consider the filters to be in a $(M, K * K * D)$ kernel matrix.
3. The output of the two matrices can be rolled back into 2D matrices equivalent to the original 2D layout.

Matrix Multiplication Methods

We have a large matrix-matrix product.

1. Naive Implementations

Matrix Multiplication Methods

We have a large matrix-matrix product.

1. Naive Implementations
2. Basic Linear Algebra Subroutines (BLAS)

Matrix Multiplication Methods

We have a large matrix-matrix product.

1. Naive Implementations
2. Basic Linear Algebra Subroutines (BLAS)
   - "If you have a fast version of BLAS, you will also get high performance on all programs that call BLAS."

Matrix Multiplication Methods

We have a large matrix-matrix product.

1. Naive Implementations
2. Basic Linear Algebra Subroutines (BLAS)
   - "If you have a fast version of BLAS, you will also get high performance on all programs that call BLAS."
3. Convolution using GPU

Matrix Multiplication Methods

We have a large matrix-matrix product.

1. Naive Implementations
2. Basic Linear Algebra Subroutines (BLAS)
   - "If you have a fast version of BLAS, you will also get high performance on all programs that call BLAS."
3. Convolution using GPU
   - ability to access memory in a 2D manner

Matrix Multiplication Methods

We have a large matrix-matrix product.

1. Naive Implementations
2. Basic Linear Algebra Subroutines (BLAS)
   - "If you have a fast version of BLAS, you will also get high performance on all programs that call BLAS."
3. Convolution using GPU
   - ability to access memory in a 2D manner
   - independent arithmetic logic units (ALUs)

Sources

Jia, Yangqing. "Convolution in Caffe: A Memo." GitHub. BERKELEY VISION AND LEARNING CENTER, 19 June 2015. Web. 15 Nov. 2016.

Kumar Chellapilla, Sidd Puri, Patrice Simard. High Performance Convolutional Neural Net- works for Document Processing. Guy Lorette. Tenth International Workshop on Frontiers in Handwriting Recognition, Oct 2006, La Baule (France), Suvisoft, 2006. ¡inria-00112631¿

"Unsupervised Feature Learning and Deep Learning Tutorial." Unsupervised Feature Learning and Deep Learning Tutorial. Standford University, n.d. Web. 06 Dec. 2016.