

# 卷积神经网络 (CNN)

代启国

大连民族大学  
计算机科学与技术系

2018 年 12 月 4 日

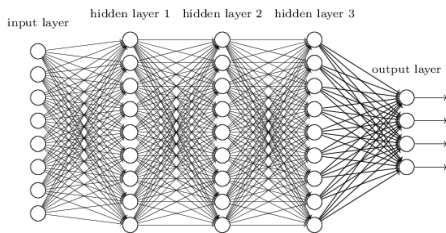
- 卷积神经网络原理及结构
- 卷积神经网络训练方法
- 卷积神经网络的实现

# 教学目标

- 理解卷积神经网络原理及结构
  - 卷积层
  - 池化层
- 理解卷积神经网络训练过程
- 学会利用 Keras 框架实现和训练卷积神经网络

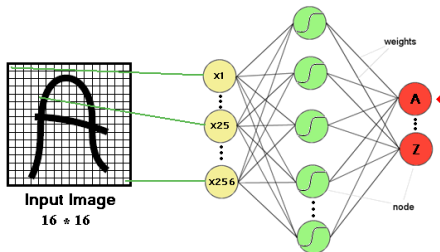
## 深度神经网络

- 输入层: 一维信号



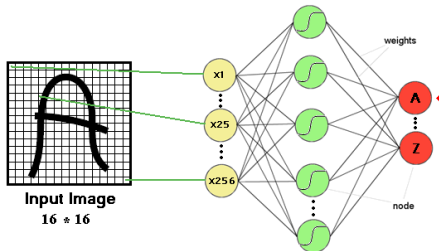
## 深度神经网络

- 输入层: 一维信号



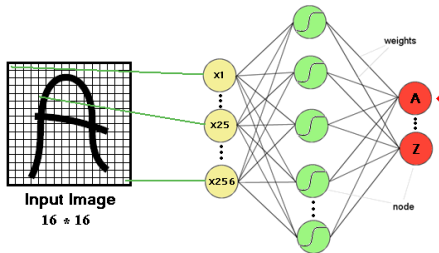
## 深度神经网络

- 输入层: 一维信号
  - 没有考虑图像像素之间的距离和空间关系



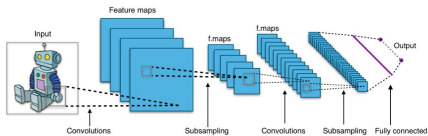
## 深度神经网络

- 输入层: 一维信号
  - 没有考虑图像像素之间的距离和空间关系



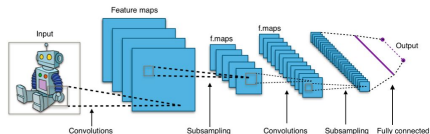
如何能够更加有效地利用图像的空间结构信息？

# 卷积神经网络



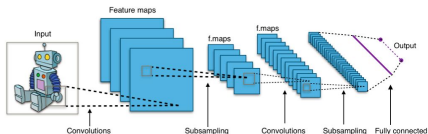


# 卷积神经网络



- 卷积神经网络
  - (Convolutional Neural Network, CNN)
  - 一种十分重要的深度学习方法
- 适合于图像等二维数据的分类
  - 微软团队在 ImageNet 2012 分类数据集中的错误率降至 4.94%，已经低于人眼识别能力 (约 5.1%)

# 卷积神经网络



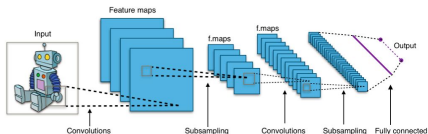
- 卷积神经网络

- (Convolutional Neural Network, **CNN**)
- 一种十分重要的深度学习方法
- 适合于图像等二维数据的分类
  - 微软团队在 ImageNet 2012 分类数据集中的错误率降至 4.94%，已经低于人眼识别能力 (约 5.1%)



- LeCun 出生在法国，曾在多伦多大学跟随深度学习鼻祖 Geoffrey Hinton 进行博士后研究
- 20 世纪 80 年代末，Yann LeCun 就作为贝尔实验室的研究员提出了卷积网络技术
- 上世纪末本世纪初，当神经网络失宠时 Yann LeCun 是少数几名一直坚持的科学家之一
- Facebook 人工智能研究部门负责人

# 卷积神经网络



- 卷积神经网络
  - (Convolutional Neural Network, **CNN**)
  - 一种十分重要的深度学习方法
- 适合于图像等二维数据的分类
  - 微软团队在 ImageNet 2012 分类数据集中的错误率降至 4.94%，已经低于人眼识别能力 (约 5.1%)

论文被拒千百遍,团队不受待见.Yann LeCun为何仍待深度学习如初恋?...

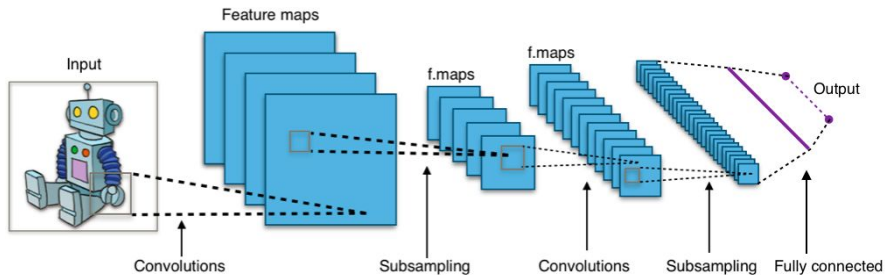


2017年4月25日 - Yann LeCun是当前人工智能领域的三驾马车之一, 可是你能想象神经网络在20年前曾有被完全边缘化的经历吗?  
<https://www.leiphone.com/news/...> - 百度快照



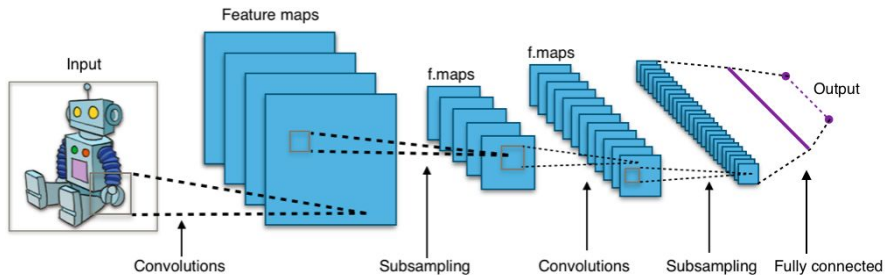
- LeCun 出生在法国, 曾在多伦多大学跟随深度学习鼻祖 Geoffrey Hinton 进行博士后研究
- 20 世纪 80 年代末, Yann LeCun 就作为贝尔实验室的研究员提出了卷积网络技术
- 上世纪末本世纪初, 当神经网络失宠时 Yann LeCun 是少数几名一直坚持的科学家之一
- Facebook 人工智能研究部门负责人

# CNN 网络构造



## 卷积神经网络

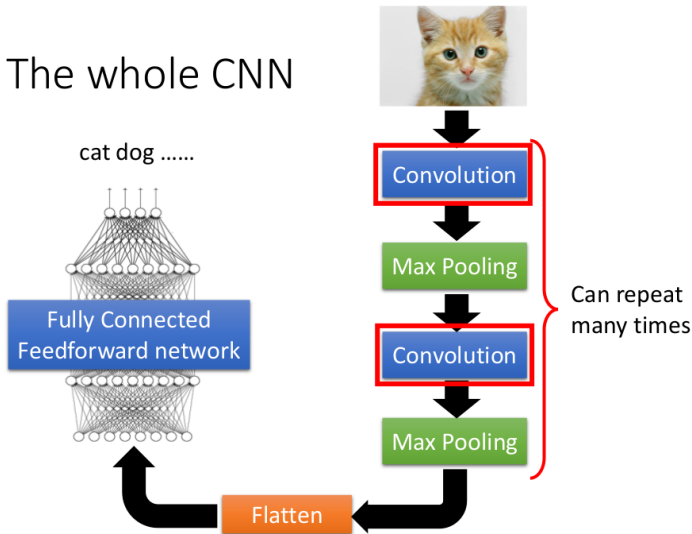
# CNN 网络构造



## 卷积神经网络

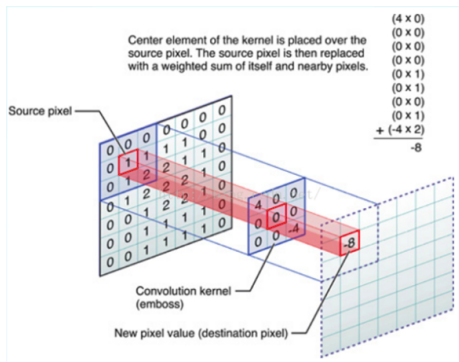
- 卷积层 (Convolution layer)
- 池化层 (Pooling layer), 也称降采样层 (Sub-sampling)
- 全连通层 (Fully connected layer)

## The whole CNN



# CNN 网络构造——卷积层

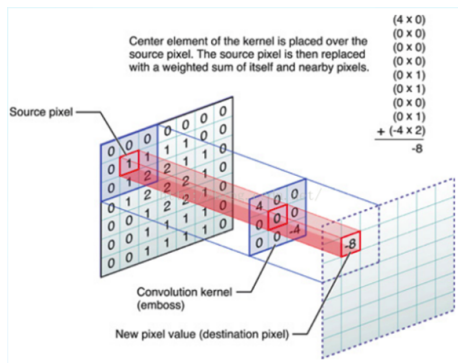
## 如何进行卷积？



# CNN 网络构造——卷积层

## 如何进行卷积？

- 1 将卷积核中心点，与输入图像中某一像素（C）对准

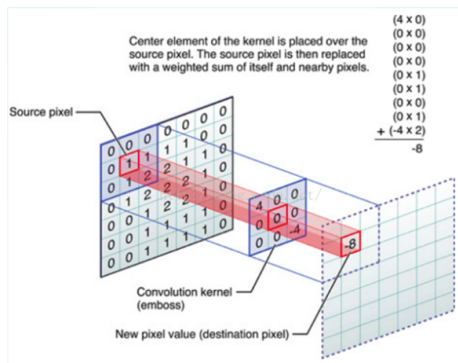




# CNN 网络构造——卷积层

## 如何进行卷积？

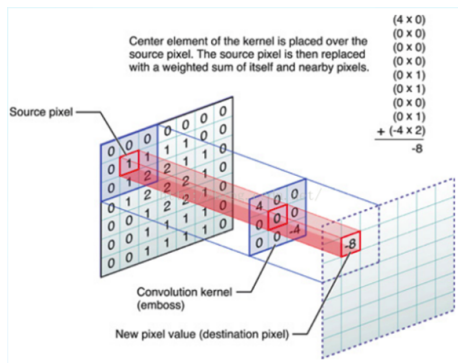
- 1 将卷积核中心点，与输入图像中某一像素（C）对准
- 2 计算“输入图像中 C 及其邻域像素”与“卷积核对应位置元素”乘积之和



# CNN 网络构造——卷积层

## 如何进行卷积？

- 1 将卷积核中心点，与输入图像中某一像素（C）对准
- 2 计算“输入图像中 C 及其邻域像素”与“卷积核对应位置元素”乘积之和
- 3 将该值作为卷积后图像对应位置的像素值



# CNN 网络构造——卷积层

## 如何进行卷积？

- 1 将卷积核中心点，与输入图像中某一像素（C）对准
- 2 计算“输入图像中C及其邻域”与“卷积核对应位置元素”乘积之和
- 3 将该值作为卷积后图像对应位置的像素值

|                        |                        |                        |   |   |
|------------------------|------------------------|------------------------|---|---|
| 1<br><small>x1</small> | 1<br><small>x0</small> | 1<br><small>x1</small> | 0 | 0 |
| 0<br><small>x0</small> | 1<br><small>x1</small> | 1<br><small>x0</small> | 1 | 0 |
| 0<br><small>x1</small> | 0<br><small>x0</small> | 1<br><small>x1</small> | 1 | 1 |
| 0                      | 0                      | 1                      | 1 | 0 |
| 0                      | 1                      | 1                      | 0 | 0 |

Image

|   |  |  |
|---|--|--|
| 4 |  |  |
|   |  |  |
|   |  |  |

Convolved  
Feature

# CNN 网络构造——卷积层

## 如何进行卷积？

- 1 将卷积核中心点，与输入图像中某一像素（C）对准
- 2 计算“输入图像中C 及其邻域”与“卷积核对应位置元素”乘积之和
- 3 将该值作为卷积后图像对应位置的像素值

|   |                 |                 |                 |   |
|---|-----------------|-----------------|-----------------|---|
| 1 | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 0 <sub>x1</sub> | 0 |
| 0 | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 0 |
| 0 | 0 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 1 |
| 0 | 0               | 1               | 1               | 0 |
| 0 | 1               | 1               | 0               | 0 |

Image

|   |   |  |
|---|---|--|
| 4 | 3 |  |
|   |   |  |
|   |   |  |

Convolved  
Feature

# CNN 网络构造——卷积层

## 如何进行卷积？

- 1 将卷积核中心点，与输入图像中某一像素（C）对准
- 2 计算“输入图像中C及其邻域”与“卷积核对应位置元素”乘积之和
- 3 将该值作为卷积后图像对应位置的像素值

|   |   |                 |                 |                 |
|---|---|-----------------|-----------------|-----------------|
| 1 | 1 | 1               | 0               | 0               |
| 0 | 1 | 1               | 1               | 0               |
| 0 | 0 | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> |
| 0 | 0 | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 0 <sub>x0</sub> |
| 0 | 1 | 1 <sub>x1</sub> | 0 <sub>x0</sub> | 0 <sub>x1</sub> |

Image

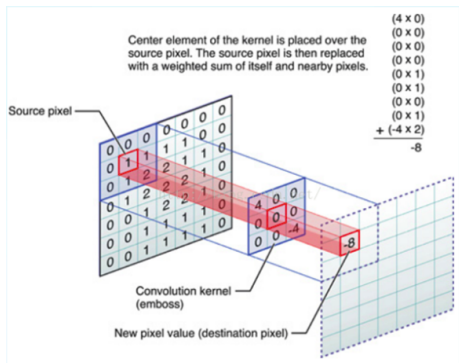
|   |   |   |
|---|---|---|
| 4 | 3 | 4 |
| 2 | 4 | 3 |
| 2 | 3 | 4 |

Convolved  
Feature

# CNN 网络构造——卷积层

## 如何进行卷积？

- 1 将卷积核中心点，与输入图像中某一像素（C）对准
- 2 计算“输入图像中C及其邻域”与“卷积核对应位置元素”乘积之和
- 3 将该值作为卷积后图像对应位置的像素值



## 卷积操作 == 神经元

- 原始图像：输入
- 卷积核：加权参数
- 卷积后的值：输出

## CNN – Zero Padding

|    |    |    |
|----|----|----|
| 1  | -1 | -1 |
| -1 | 1  | -1 |
| -1 | -1 | 1  |

Filter 1

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 |   |   |   |   |   |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |   |
|   | 0 | 0 | 1 | 1 | 0 | 0 |   |
|   | 1 | 0 | 0 | 0 | 1 | 0 |   |
|   | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
|   | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
|   |   |   |   |   |   |   | 0 |
|   |   |   |   |   |   |   | 0 |
|   |   |   |   |   |   |   | 0 |

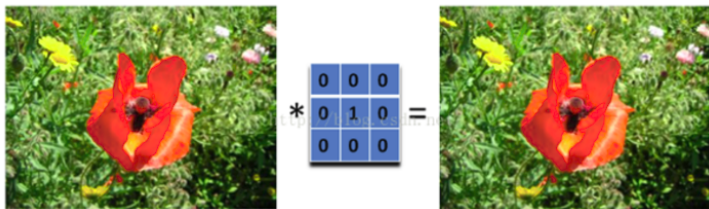
6 x 6 image

You will get another 6 x 6 images in this way



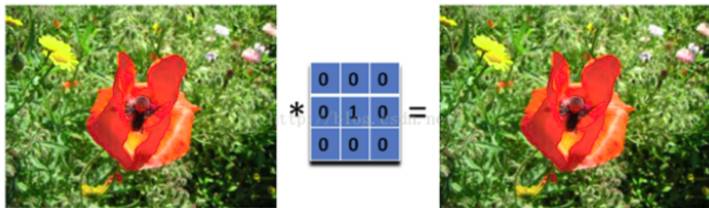
Zero padding

# CNN 网络构造——卷积层





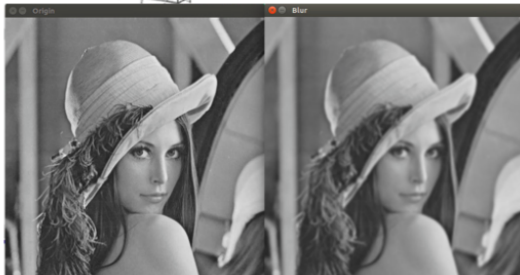
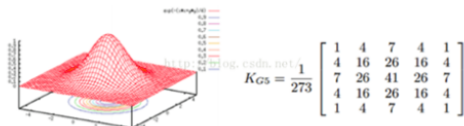
# CNN 网络构造——卷积层



图像没有变化

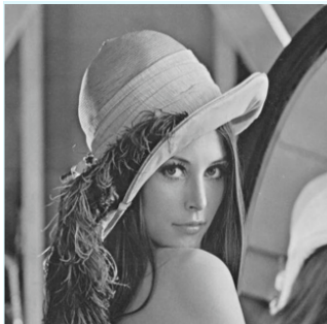
# CNN 网络构造——卷积层

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}; G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



对图像进行高斯模糊

# CNN 网络构造——卷积层



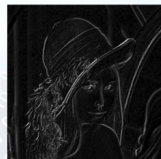
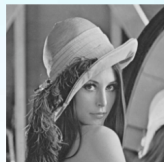
## ➤ 图像导数

$$I_x = I * D_x \quad I_y = I * D_y$$

## ➤ Sobel滤波器

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad D_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$D = 0.5 * D_x + 0.5 * D_y$$



# CNN 网络构造——卷积层

卷积可用于识别图像中特定模式

|   |   |   |    |    |    |   |
|---|---|---|----|----|----|---|
| 0 | 0 | 0 | 0  | 0  | 30 | 0 |
| 0 | 0 | 0 | 0  | 30 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 0  | 0  | 0  | 0 |

Pixel representation of filter



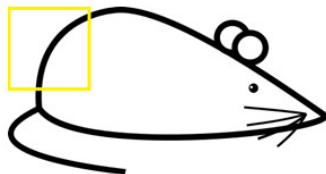
Visualization of a curve detector filter

用于检测特定曲线形状的结构（模式）

# CNN 网络构造——卷积层



Original image

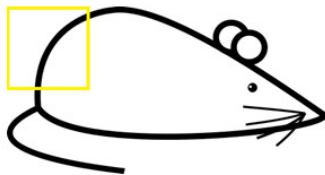


Visualization of the filter on the image

# CNN 网络构造——卷积层



Original image



Visualization of the filter on the image



Visualization of the receptive field

|   |   |   |    |    |    |    |
|---|---|---|----|----|----|----|
| 0 | 0 | 0 | 0  | 0  | 0  | 30 |
| 0 | 0 | 0 | 0  | 50 | 50 | 50 |
| 0 | 0 | 0 | 20 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |

Pixel representation of the receptive field

\*

|   |   |   |    |   |    |   |
|---|---|---|----|---|----|---|
| 0 | 0 | 0 | 0  | 0 | 30 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0 | 0  | 0 |
| 0 | 0 | 0 | 0  | 0 | 0  | 0 |

Pixel representation of filter

Multiplication and Summation =  $(50 \times 30) + (50 \times 30) + (50 \times 30) + (20 \times 30) + (50 \times 30) = 6600$  (A large number!)

- 卷积后的值: 6600 ( “信号强” )

# CNN 网络构造——卷积层



Visualization of the receptive field

|   |   |   |    |    |    |    |
|---|---|---|----|----|----|----|
| 0 | 0 | 0 | 0  | 0  | 0  | 30 |
| 0 | 0 | 0 | 0  | 50 | 50 | 50 |
| 0 | 0 | 0 | 20 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |

Pixel representation of the receptive field

\*

|   |   |   |    |    |    |   |
|---|---|---|----|----|----|---|
| 0 | 0 | 0 | 0  | 0  | 30 | 0 |
| 0 | 0 | 0 | 0  | 30 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |

Pixel representation of filter

Multiplication and Summation =  $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$  (A large number!)

- 卷积后的值: 6600 ( “信号强” )



Visualization of the filter on the image

|    |    |    |    |   |   |   |
|----|----|----|----|---|---|---|
| 0  | 0  | 0  | 0  | 0 | 0 | 0 |
| 0  | 40 | 0  | 0  | 0 | 0 | 0 |
| 40 | 0  | 40 | 0  | 0 | 0 | 0 |
| 40 | 20 | 0  | 0  | 0 | 0 | 0 |
| 0  | 50 | 0  | 0  | 0 | 0 | 0 |
| 0  | 0  | 50 | 0  | 0 | 0 | 0 |
| 25 | 25 | 0  | 50 | 0 | 0 | 0 |

Pixel representation of receptive field

\*

|   |   |   |    |    |    |   |
|---|---|---|----|----|----|---|
| 0 | 0 | 0 | 0  | 0  | 30 | 0 |
| 0 | 0 | 0 | 0  | 30 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |

Pixel representation of filter

Multiplication and Summation = 0

- 卷积后的值: 0 ( “信号低” )



# CNN 网络构造——卷积层



Visualization of the receptive field

|   |   |   |    |    |    |    |
|---|---|---|----|----|----|----|
| 0 | 0 | 0 | 0  | 0  | 0  | 30 |
| 0 | 0 | 0 | 0  | 50 | 50 | 50 |
| 0 | 0 | 0 | 20 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |
| 0 | 0 | 0 | 50 | 50 | 0  | 0  |

Pixel representation of the receptive field

\*

|   |   |   |    |    |    |   |
|---|---|---|----|----|----|---|
| 0 | 0 | 0 | 0  | 0  | 30 | 0 |
| 0 | 0 | 0 | 0  | 30 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 0  | 0  | 0  | 0 |

Pixel representation of filter

Multiplication and Summation =  $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$  (A large number!)

- 卷积后的值: 6600 ( “信号强” )



Visualization of the filter on the image

|    |    |    |    |   |   |   |
|----|----|----|----|---|---|---|
| 0  | 0  | 0  | 0  | 0 | 0 | 0 |
| 0  | 40 | 0  | 0  | 0 | 0 | 0 |
| 40 | 0  | 40 | 0  | 0 | 0 | 0 |
| 40 | 20 | 0  | 0  | 0 | 0 | 0 |
| 0  | 50 | 0  | 0  | 0 | 0 | 0 |
| 0  | 0  | 50 | 0  | 0 | 0 | 0 |
| 25 | 25 | 0  | 50 | 0 | 0 | 0 |

Pixel representation of receptive field

\*

|   |   |   |    |    |    |   |
|---|---|---|----|----|----|---|
| 0 | 0 | 0 | 0  | 0  | 30 | 0 |
| 0 | 0 | 0 | 0  | 30 | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 30 | 0  | 0  | 0 |
| 0 | 0 | 0 | 0  | 0  | 0  | 0 |

Pixel representation of filter

Multiplication and Summation = 0

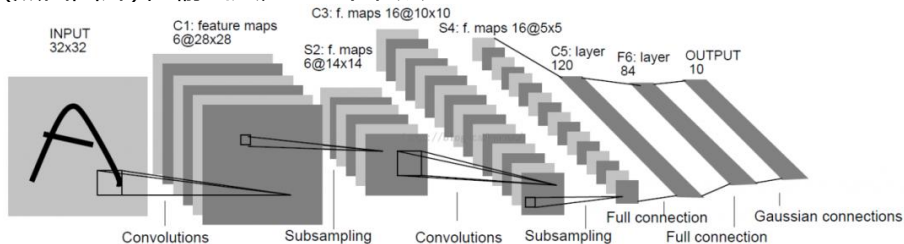
- 卷积后的值: 0 ( “信号低” )

设计不同的卷积核，检测图像中不同的图像模式

# CNN 网络构造——卷积层

## 详解卷积层

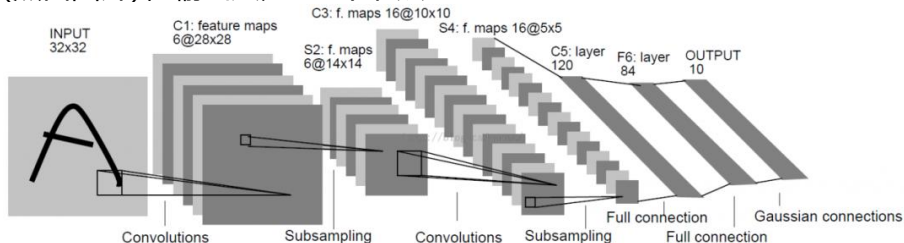
以 LeNet5 网络结构为例，用于识别手写数字，输入层为单通道图片（黑白图片），输出层为 10 个节点



# CNN 网络构造——卷积层

## 详解卷积层

以 LeNet5 网络结构为例，用于识别手写数字，输入层为单通道图片（黑白图片），输出层为 10 个节点

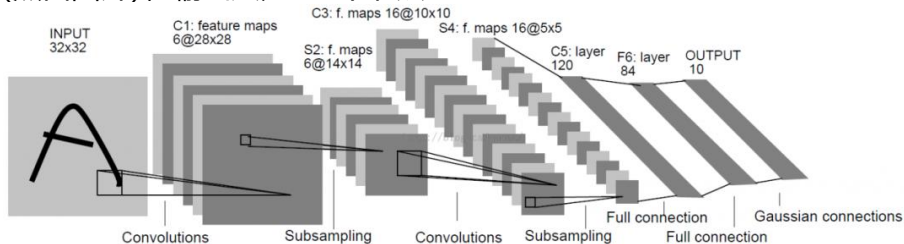


- C1、C3、C5 是卷积层

# CNN 网络构造——卷积层

## 详解卷积层

以 LeNet5 网络结构为例，用于识别手写数字，输入层为单通道图片（黑白图片），输出层为 10 个节点

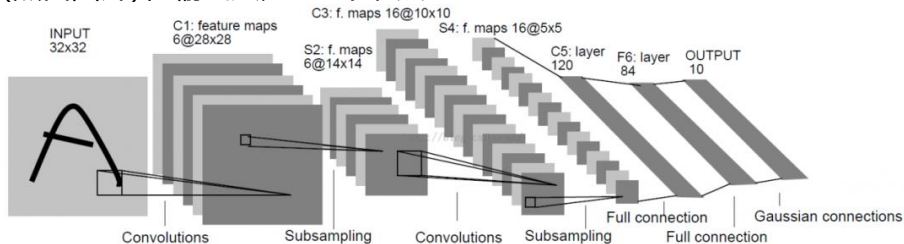


- C1、C3、C5 是卷积层
  - C1: 由 6 个不同卷积核得到 6 个特征图 (Feature map)

# CNN 网络构造——卷积层

## 详解卷积层

以 LeNet5 网络结构为例，用于识别手写数字，输入层为单通道图片（黑白图片），输出层为 10 个节点

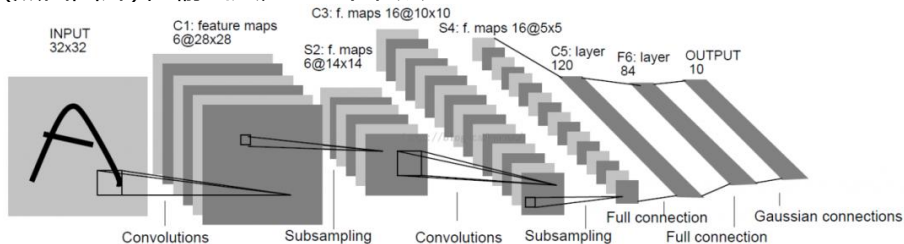


- C1、C3、C5 是卷积层
  - C1: 由 6 个不同卷积核得到 6 个特征图 (Feature map)
  - C3: 由 16 个不同卷积核得到的 16 个特征图

# CNN 网络构造——卷积层

## 详解卷积层

以 LeNet5 网络结构为例，用于识别手写数字，输入层为单通道图片（黑白图片），输出层为 10 个节点



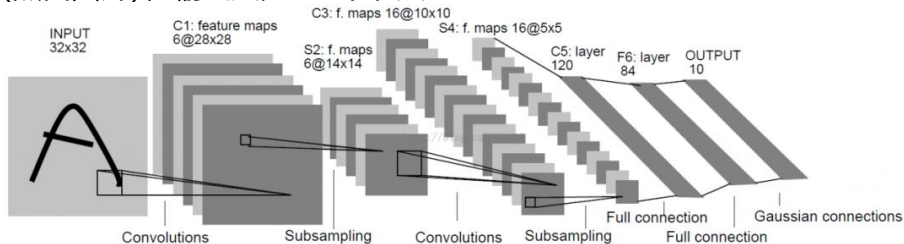
- C1、C3、C5 是卷积层

- C1: 由 6 个不同卷积核得到 6 个特征图 (Feature map)
- C3: 由 16 个不同卷积核得到的 16 个特征图
- C5: 由 12 个不同卷积核得到的 120 个特征图

# CNN 网络构造——卷积层

## 详解卷积层

以 LeNet5 网络结构为例，用于识别手写数字，输入层为单通道图片（黑白图片），输出层为 10 个节点



- C1、C3、C5 是卷积层
  - C1: 由 6 个不同卷积核得到 6 个特征图 (Feature map)
  - C3: 由 16 个不同卷积核得到的 16 个特征图
  - C5: 由 12 个不同卷积核得到的 120 个特征图
- 每个卷积层的不同卷积核对应不同特征

# CNN 网络构造——池化层

- Subsampling the pixels will not change the object



We can subsample the pixels to make image smaller

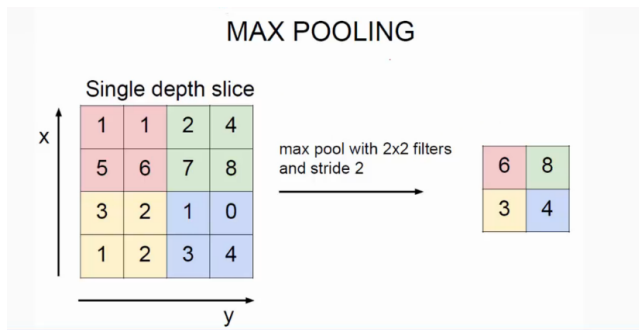
➡ Less parameters for the network to process the image

- 池化层 (降采样) 通常位于卷积层之后
- 池化的目的：旨在降低模型的规模，加速训练



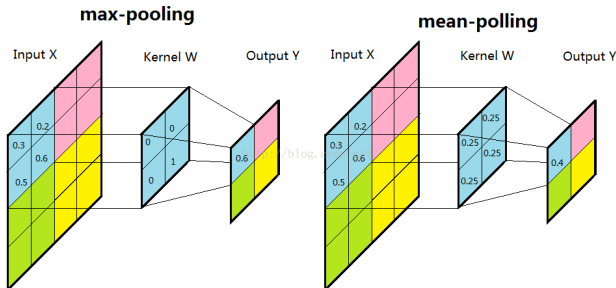
# CNN 网络构造——池化层

- 池化层 (降采样) 通常位于卷积层之后
- 池化的目的：旨在降低模型的规模，加速训练
- 常用的池化方法主要包括最大池化法和均值池化法



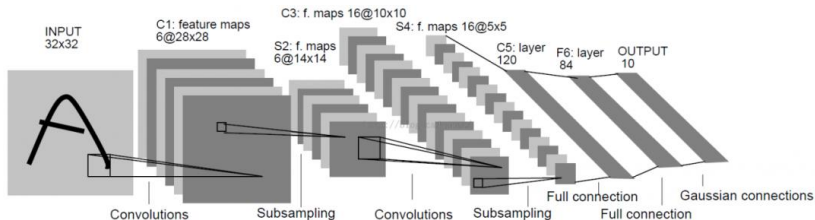
# CNN 网络构造——池化层

- 池化层 (降采样) 通常位于卷积层之后
- 池化的目的：旨在降低模型的规模，加速训练
- 常用的池化方法主要包括**最大池化法** 和**均值池化法**

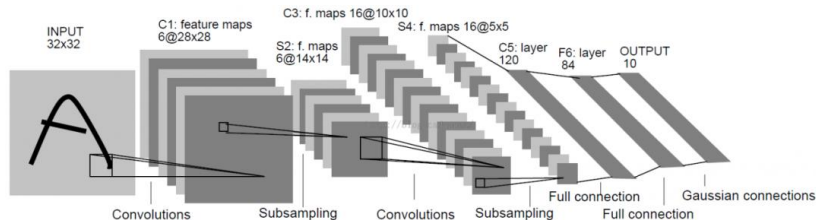


# CNN 网络构造——池化层

- 池化层 (降采样) 通常位于卷积层之后
- 池化的目的：旨在降低模型的规模，加速训练
- 常用的池化方法主要包括最大池化法和均值池化法

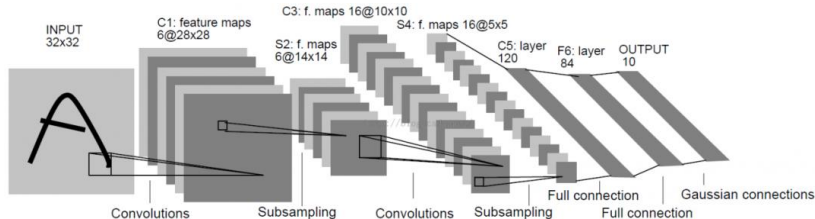


# CNN 网络构造——池化层



- S2、S4 层为池化层（降采样）
- 面积分别是 C1、C3 层中的  $1/4$

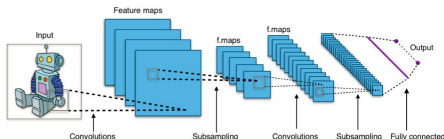
# CNN 网络构造——全连通层



- F6 为全连通层
- 包含 84 个节点, 与 C5 层 120 个节点全连通, 类似于普通神经网络

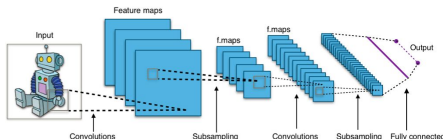
# CNN 网络若干特性

- 局部连接 (local connections)
  - 每个神经元只与上一层的部分神经元相连，只感知局部，而不是整幅图像
- 权值共享 (shared weights)
  - 用一个相同的卷积核去卷积整幅图像，相当于对图像做一个全图滤波
- 降采样 (sub-sampling)



# CNN 网络若干特性

- 局部连接 (local connections)
  - 每个神经元只与上一层的部分神经元相连，只感知局部，而不是整幅图像
- 权值共享 (shared weights)
  - 用一个相同的卷积核去卷积整幅图像，相当于对图像做一个全图滤波
- 降采样 (sub-sampling)



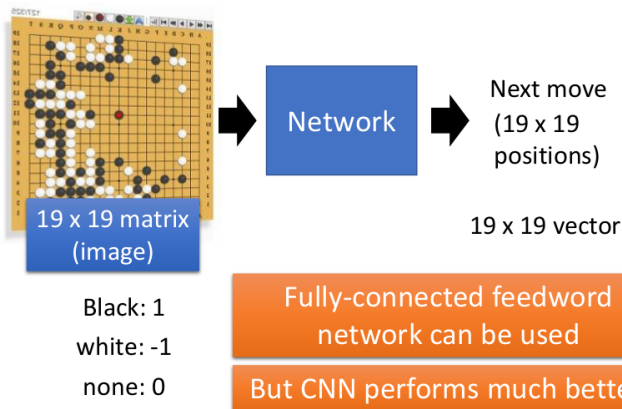
- 仿射不变性 (平移、旋转、缩放)
  - CNN 能够在浅层学到比较基础的特征，比如点、线、边缘，
  - 高层特征是这些基础浅层特征的组合
  - 即使发生仿射变化，底层基础特征依然不变

- 训练的目的：估计各层中不同卷积核及相关参数、全连通层的权值等
- 训练采用“反向传播”
  - 前向传播、损失计算、反向传播、权重更新
  - 损失计算——均方误差 (Mean squared error)

$$E(f; D) = \frac{1}{m} \sum_{i=1}^m (f(x_i) - y_i)^2$$



## Playing Go



## Why CNN for playing Go?

- Some patterns are much smaller than the whole image

Alpha Go uses 5 x 5 for first layer



- The same patterns appear in different regions.

