

2020-2021学年秋季学期

自然语言处理

Natural Language Processing



授课教师：胡玥

助 教： 于静

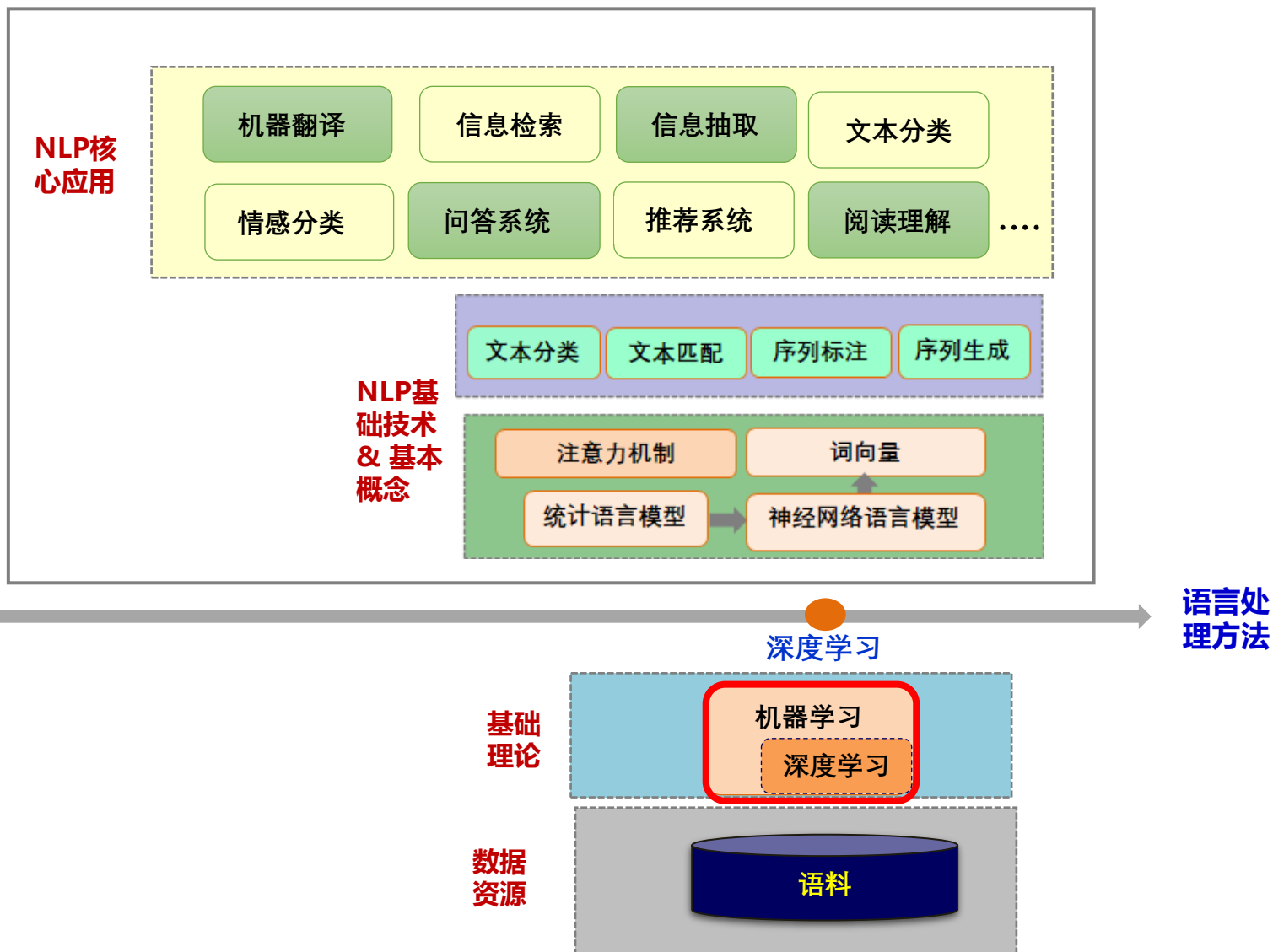
自然语言处理
Natural Language Processing

第 5 章 卷积神经网络

授课教师：胡玥

授课时间：2020.9

基于深度学习的自然语言处理课程内容



概 要

本章主要内容：

介绍卷积神经网络（CNN）的基本概念，模型结构和模型应用并简要的介绍模型参数的学习问题。

本章教学目的：

掌握卷积神经网络（CNN）的相关知识。

内 容 提 要

5.1 概述

5.2 卷积神经网络结构

5.3 卷积神经网络学习

5.4 卷积神经网络应用

5.1 概述

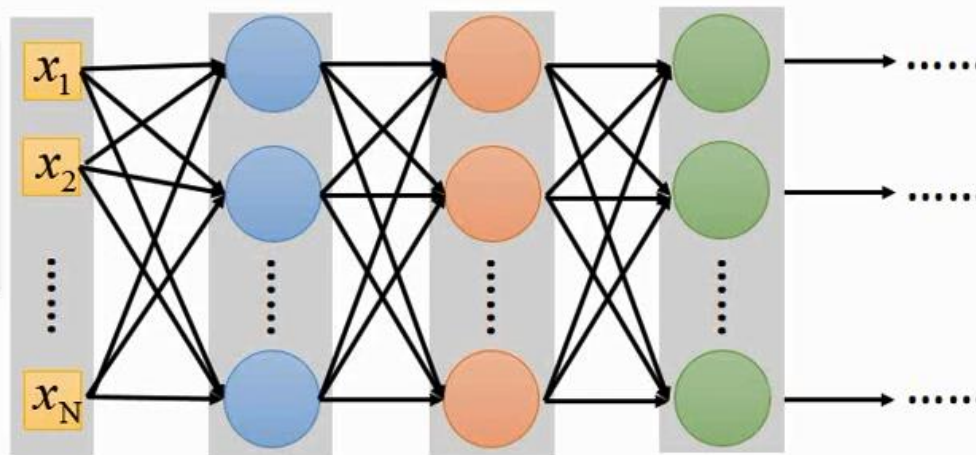
1. 问题引入:

Why CNN for Image ?

[Zeiler, M. D., *ECCV 2014*]

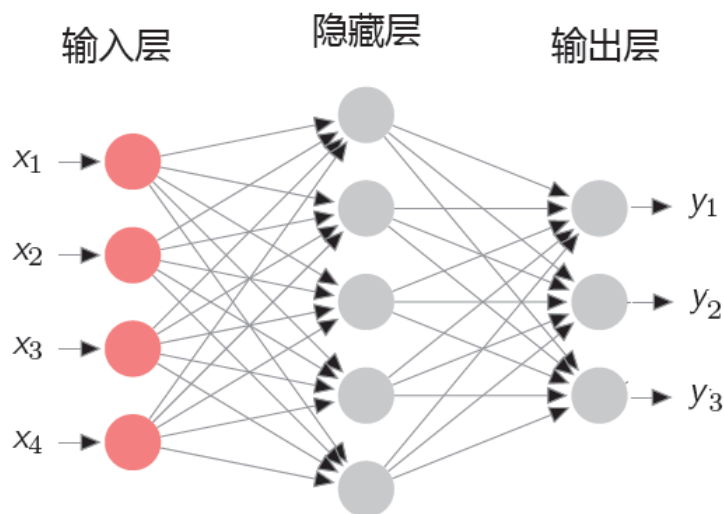
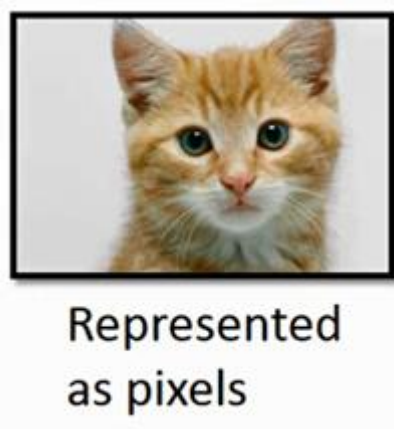


Represented
as pixels



5.1 概述

前馈神经网络



在全连接前馈神经网络中，如果第 l 层有 n^l 个神经元，第 $l-1$ 层有 $n^{(l-1)}$ 个神经元，连接边有 $n^{(l)} * n^{(l-1)}$ 个，也就是权重矩阵有 $n^{(l)} * n^{(l-1)}$ 个参数。当 m 和 n 都很大时，权重矩阵的参数非常多，训练的效率会非常低。

设图像 10×10 ；第一隐藏层 1024 个神经元, 该层全连接参数 **102400**

解决方法：



卷积神经网络

5.1 概述

卷积神经网络 (Convolutional Neural Networks, CNN)

是一种前馈神经网络。卷积神经网络是受生物学上**感受野** (Receptive Field) 的机制而提出的。

感受野 (Receptive Field) 的机制

感受野主要是指听觉系统、本体感觉系统和视觉系统中神经元的一些性质。比如在视觉神经系统中，一个神经元的感受野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。

5.1 概述

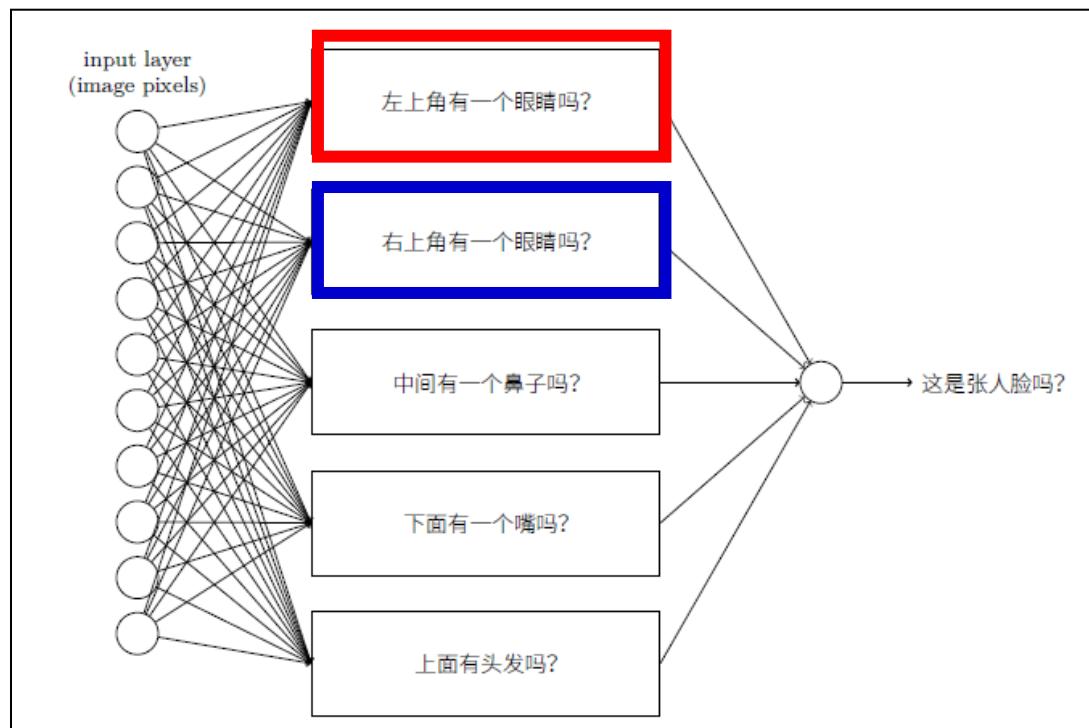
如何识别？



5.1 概述



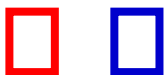
5.1 概述



CNN 事实上只能获取局部信息，是通过层叠来增大感受野

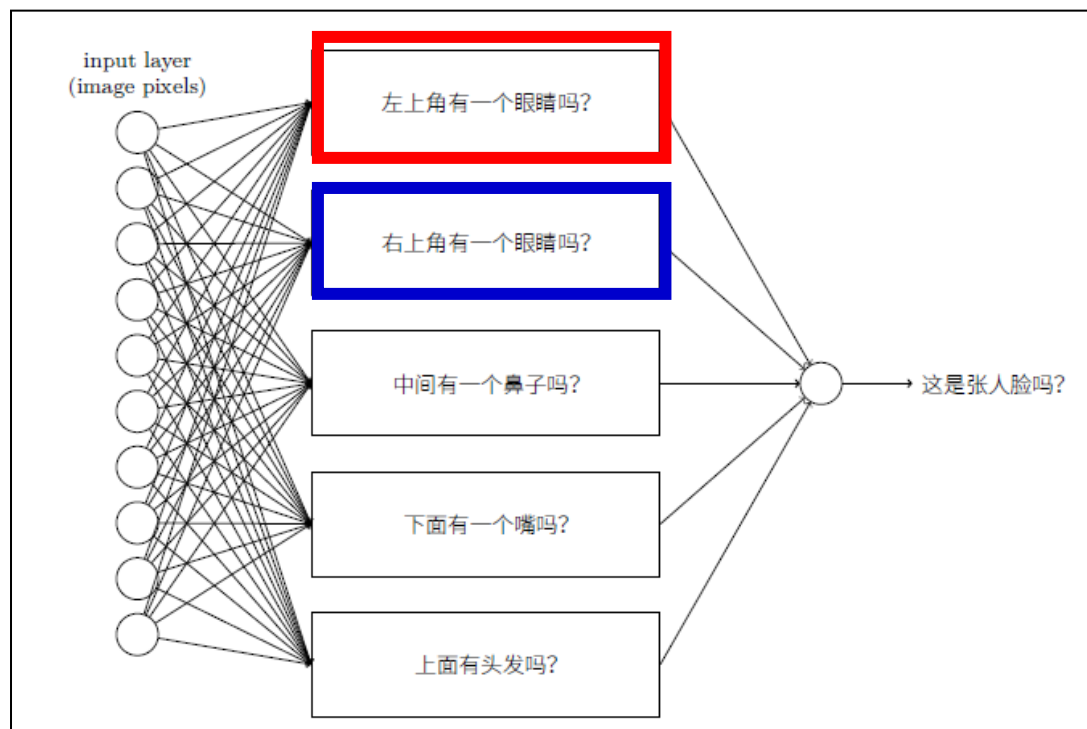
5.1 概述

卷积核 (filter)



原始图片

特征图谱 (Feature Map)



运算: 卷积运算

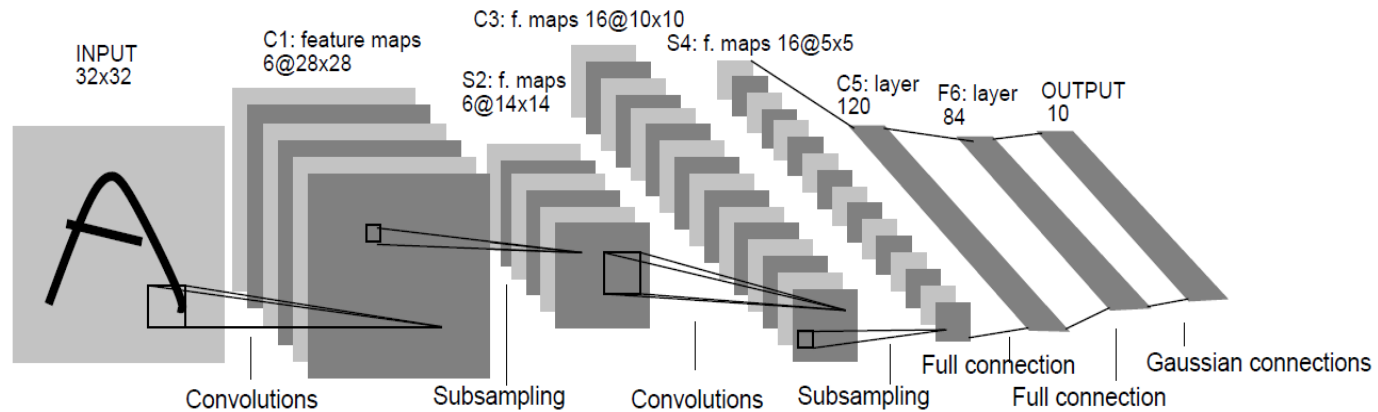
2. 卷积运算:

卷积运算: 见 附录

5.1 概述

3. 卷积网:

最早的CNN: LeNet-5



Yann LeCun: Gradient-Based Learning Applied to Document Recognition, 1998

内 容 提 要

5.1 概述

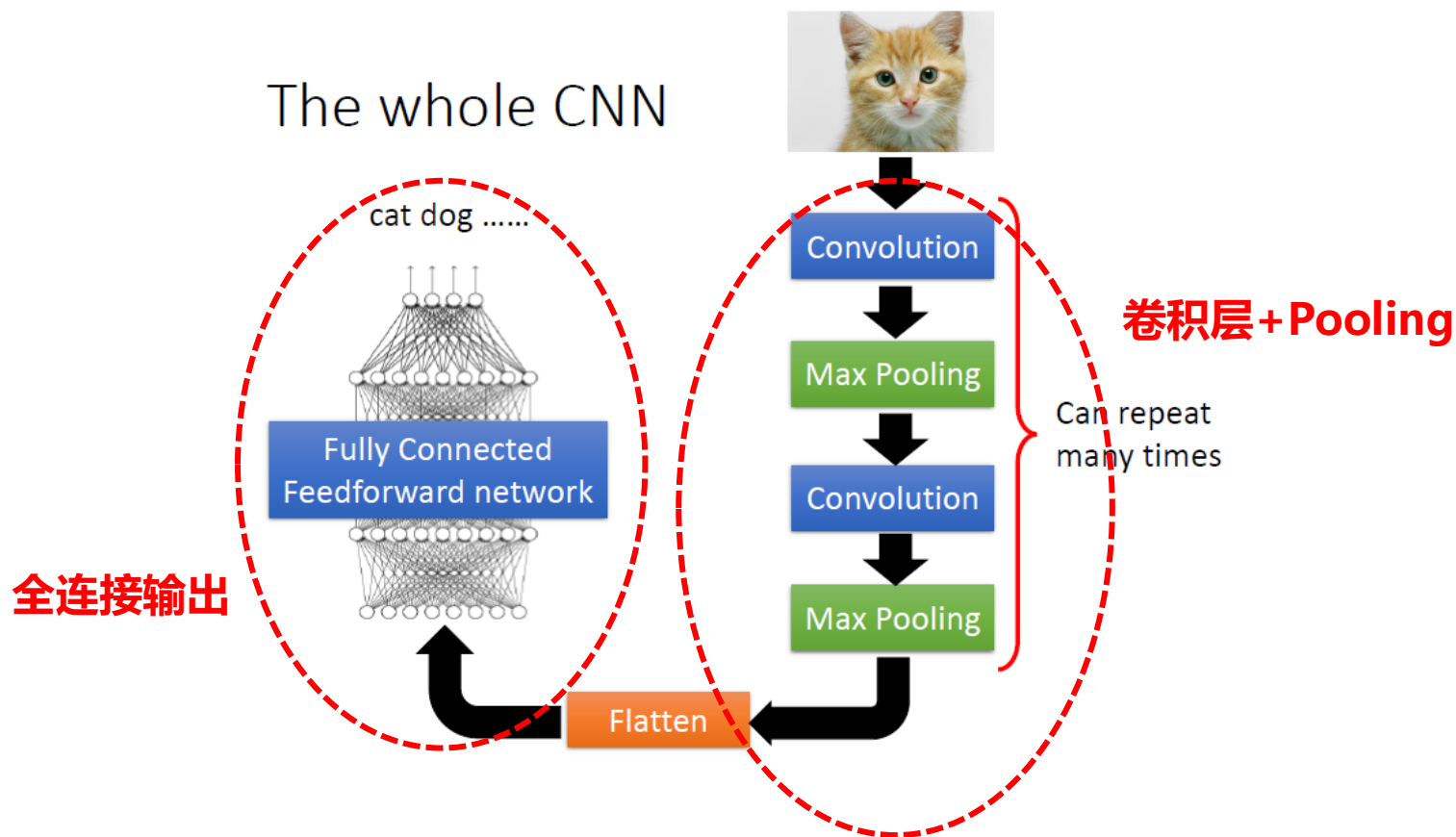
5.2 卷积神经网络结构

5.3 卷积神经网络学习

5.4 卷积神经网络应用

5.1 卷积神经网络结构

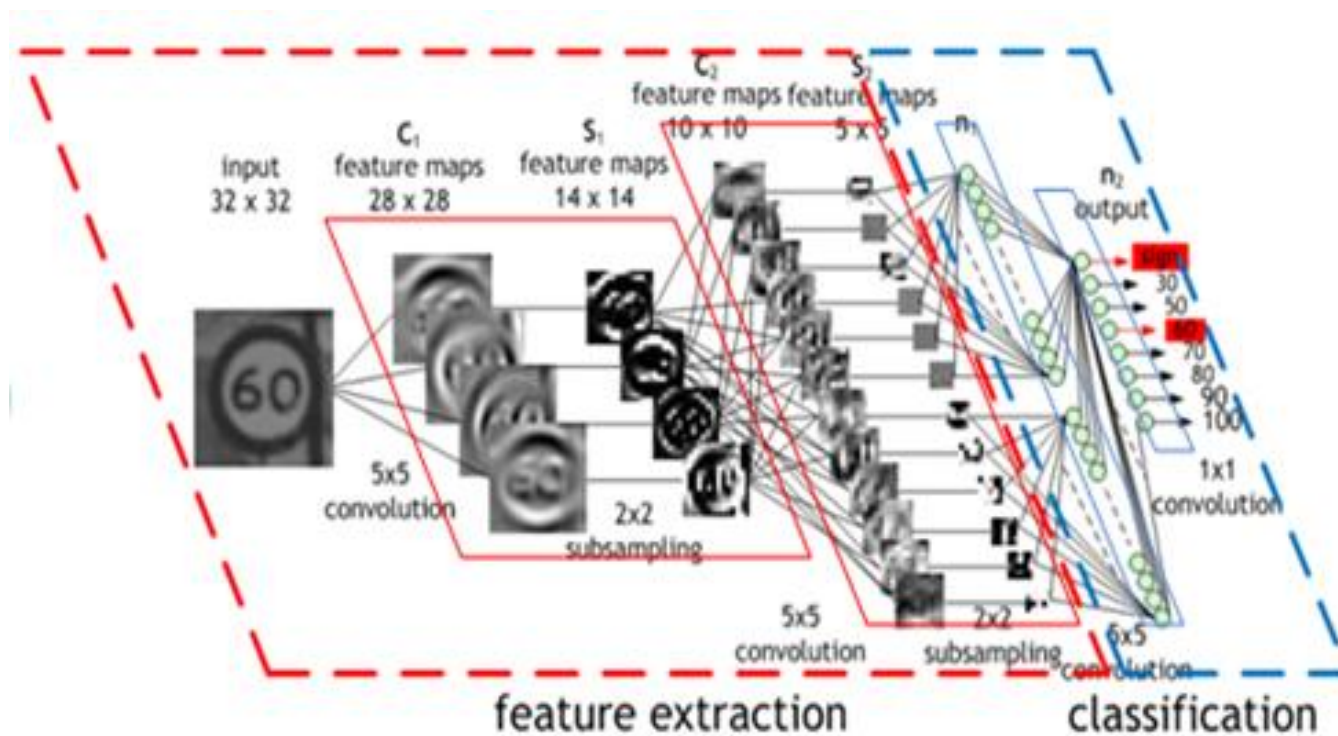
图像识别



卷积网络是由卷积层、子采样层、全连接层交叉堆叠而成

5.1 卷积神经网络结构

手写识别

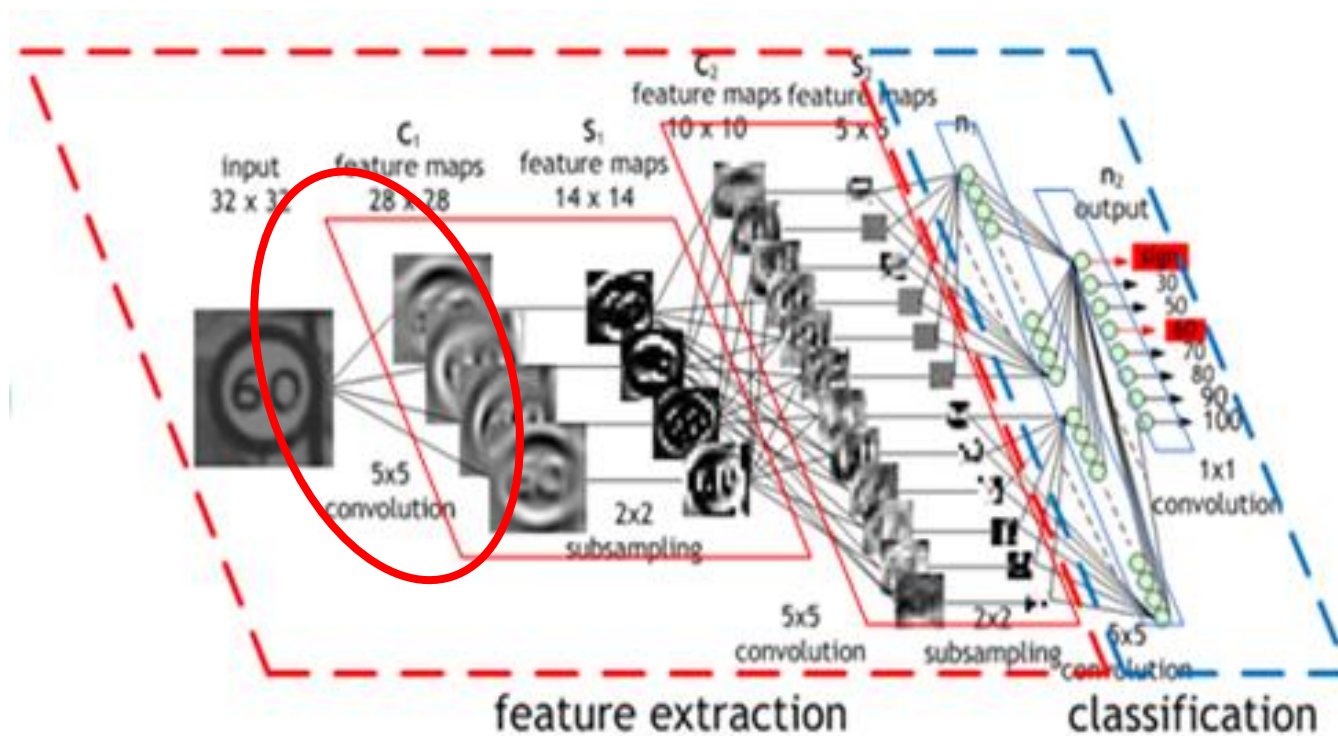


卷积层+Pooling

全连接输出

5.1 卷积神经网络结构

卷积层



卷积层+Pooling

全连接输出

5.1 卷积神经网络结构

卷积层:

输入图像:

1	0	0	0	0	1
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	1	0	1	0

6 x 6 image

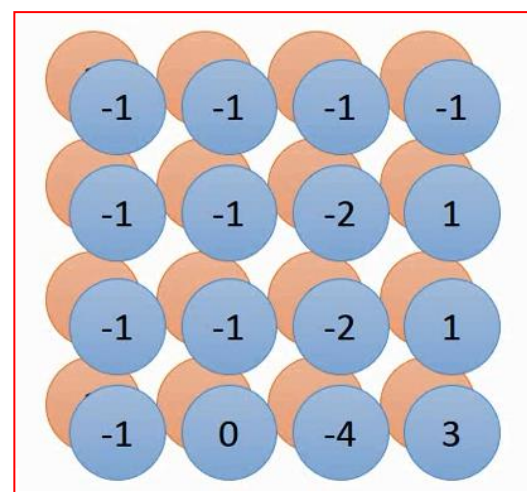
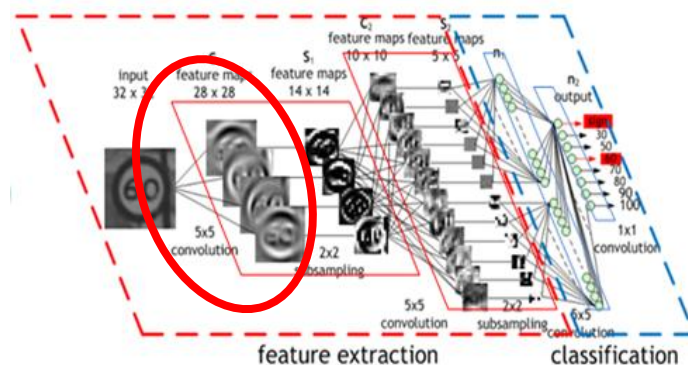
卷积核:

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



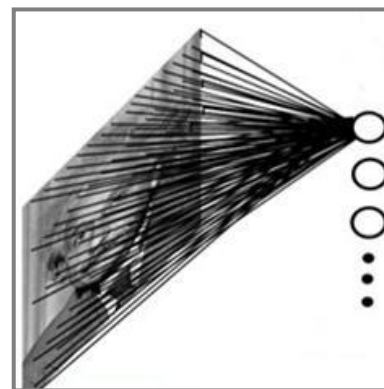
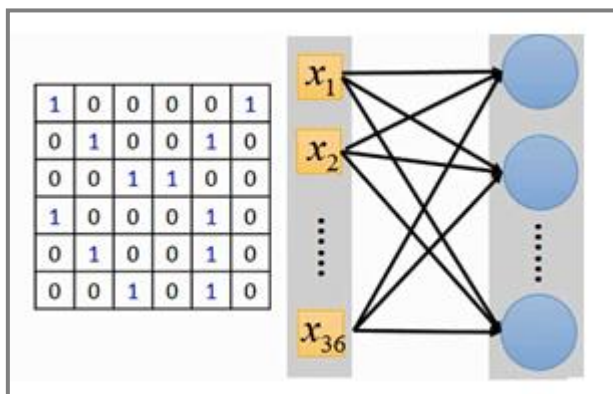
Filter1的特征图谱

Filter2的特征图谱

5.1 卷积神经网络结构

卷积连接特点:

全连接:

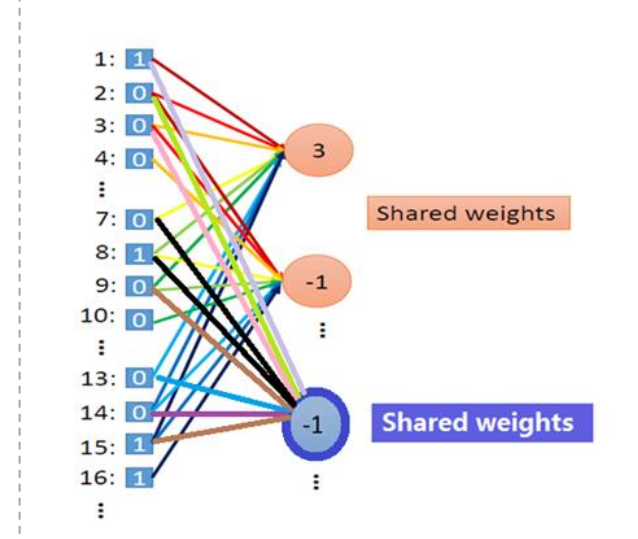
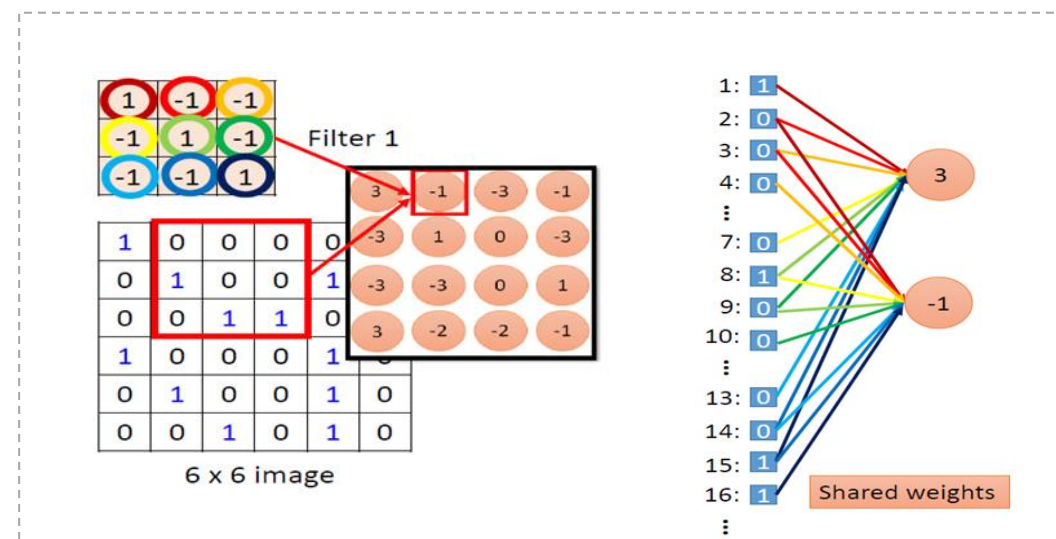
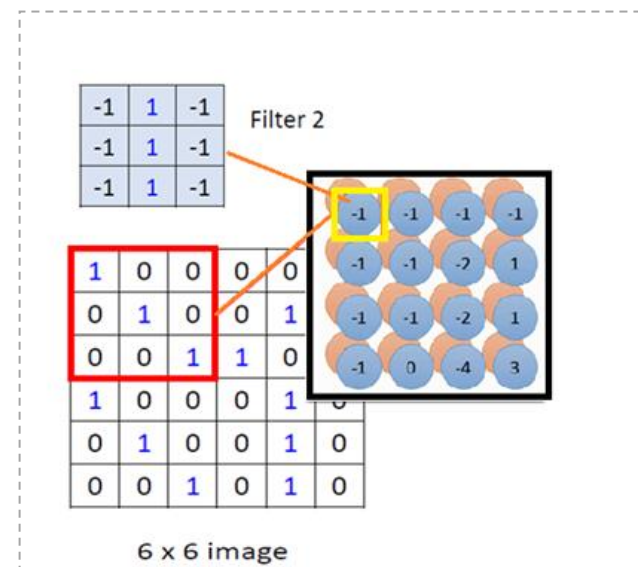
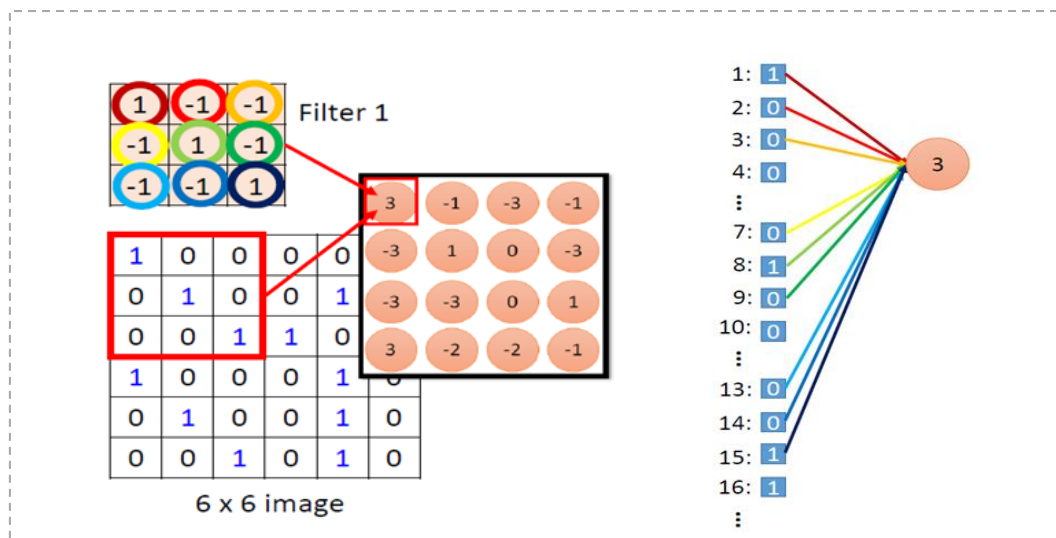


如：图像 10×10 ，第一层神经元 1024

全连接参数 102400

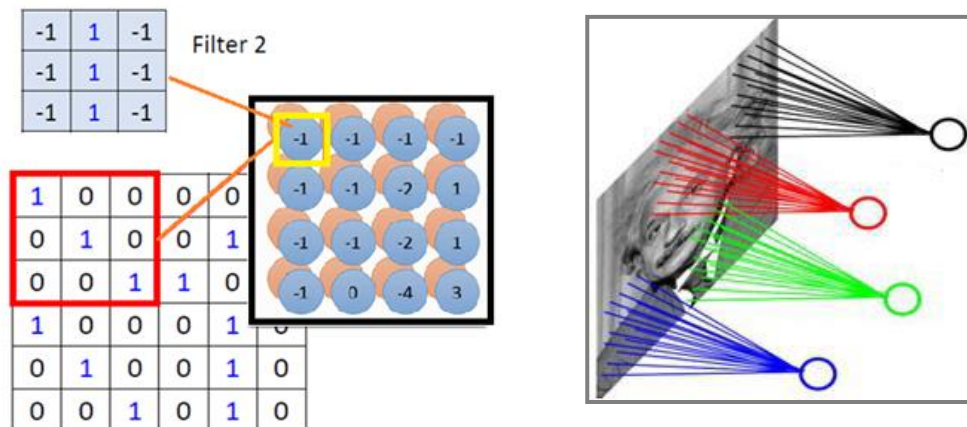
5.1 卷积神经网络结构

卷积连接: **特点: 1 局部连接, 2 权重共享**



5.1 卷积神经网络结构

卷积连接:



如：图像 10×10 ，第一层神经元 1024

- 如 卷积层1024个神经元是用 16 个 3×3 滤波器卷积得到
即：卷积层神经元 $16 \times (8 \times 8) = 1024$

卷积层：参数 $9 \times 16 = 144$

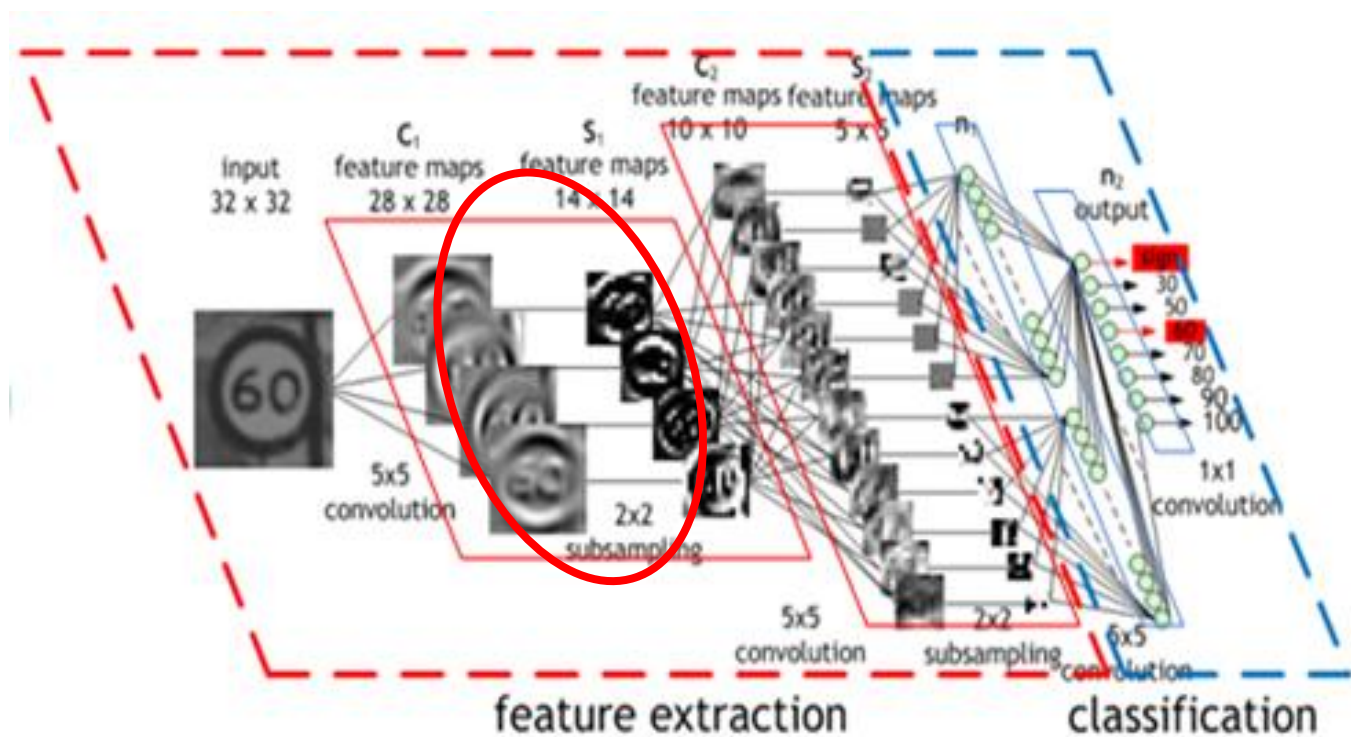
- 如 卷积层1053个神经元是用 13 个 2×2 滤波器卷积得到
即：卷积层神经元 $13 \times (9 \times 9) = 1053$

卷积层：参数 $4 \times 13 = 52$

卷积层连接可以减少参数的个数

5.1 卷积神经网络结构

池化层 (Pooling)



卷积层+Pooling

全连接输出

5.1 卷积神经网络结构

卷积层问题:

输入图像:

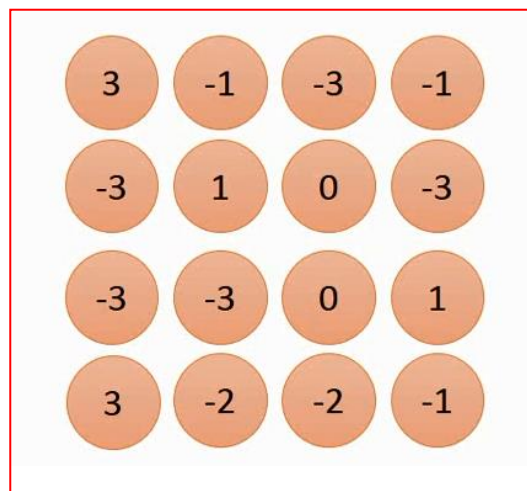
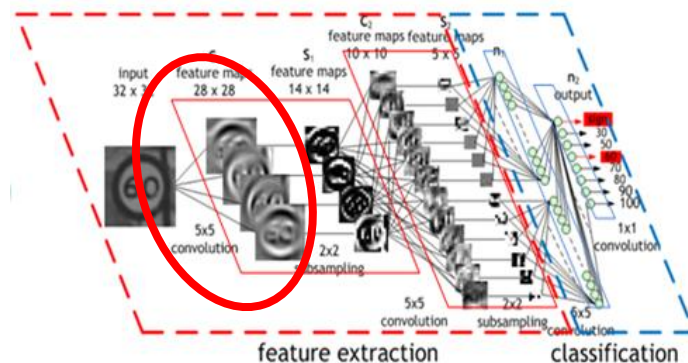
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

卷积核:

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1



Filter1的特征图谱

卷积层虽然可以显著减少连接的个数,但是每一个特征映射的神经元个数并没有显著减少。

解决方法?

5.1 卷积神经网络结构

- Subsampling the pixels will not change the object



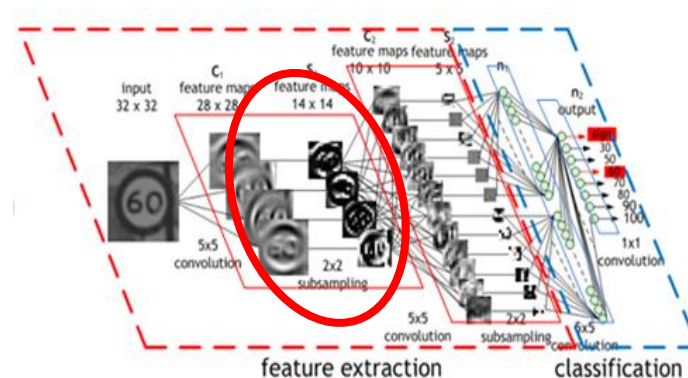
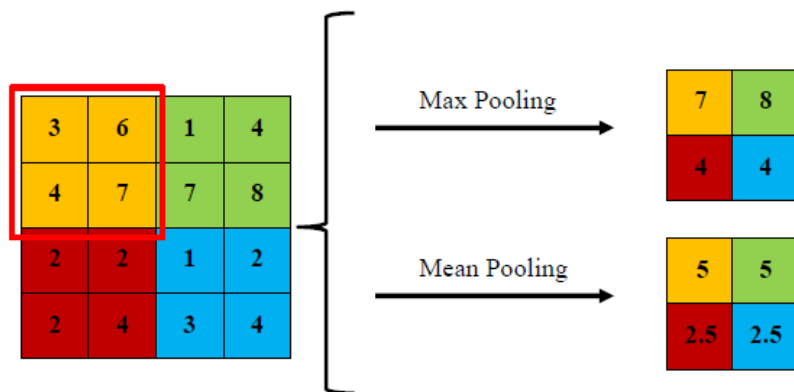
We can subsample the pixels to make image smaller

➡ Less parameters for the network to process the image

5.1 卷积神经网络结构

池化层 (Pooling)

池化：一种采样操作，用于减少模型参数并保留有效信息避免过拟合，提高训练速度



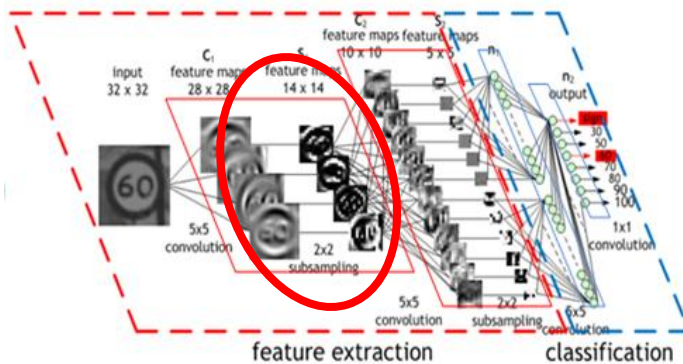
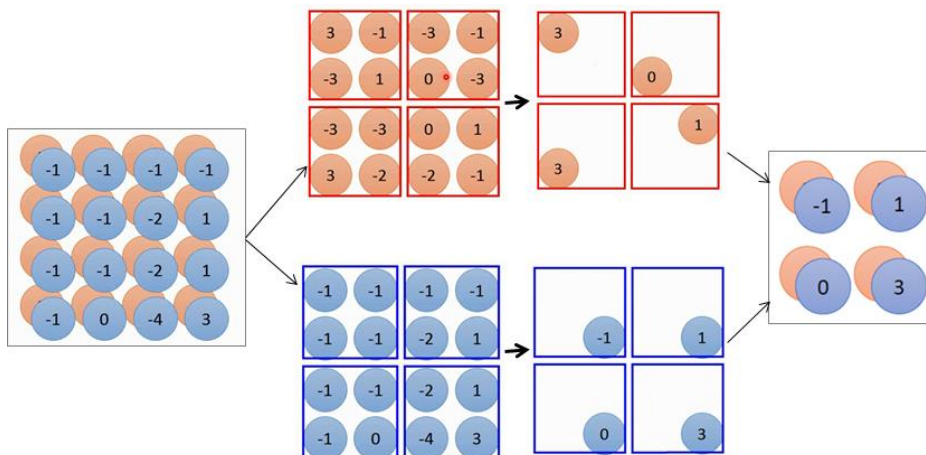
常用的池化方法

Max Pooling

Mean Pooling

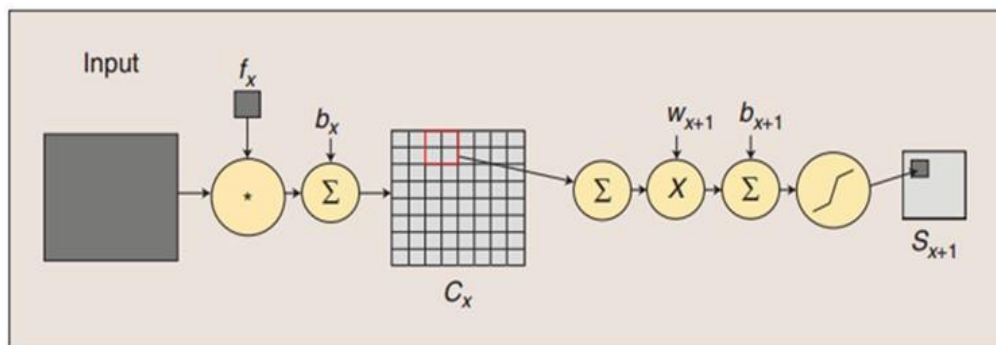
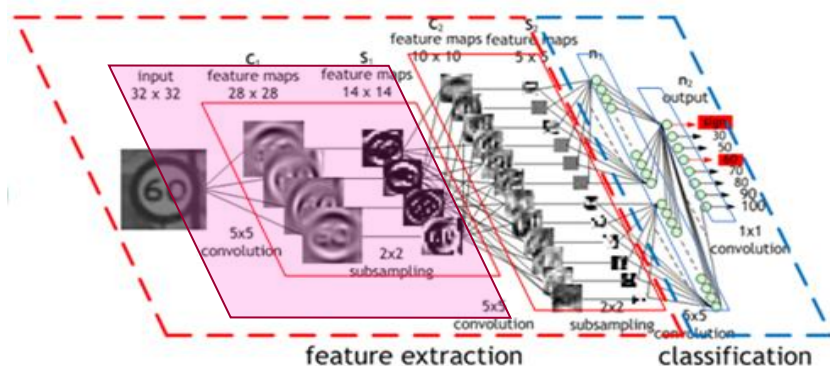
5.1 卷积神经网络结构

Max Pooling



5.1 卷积神经网络结构

卷积层+Pooling层 信息传播:



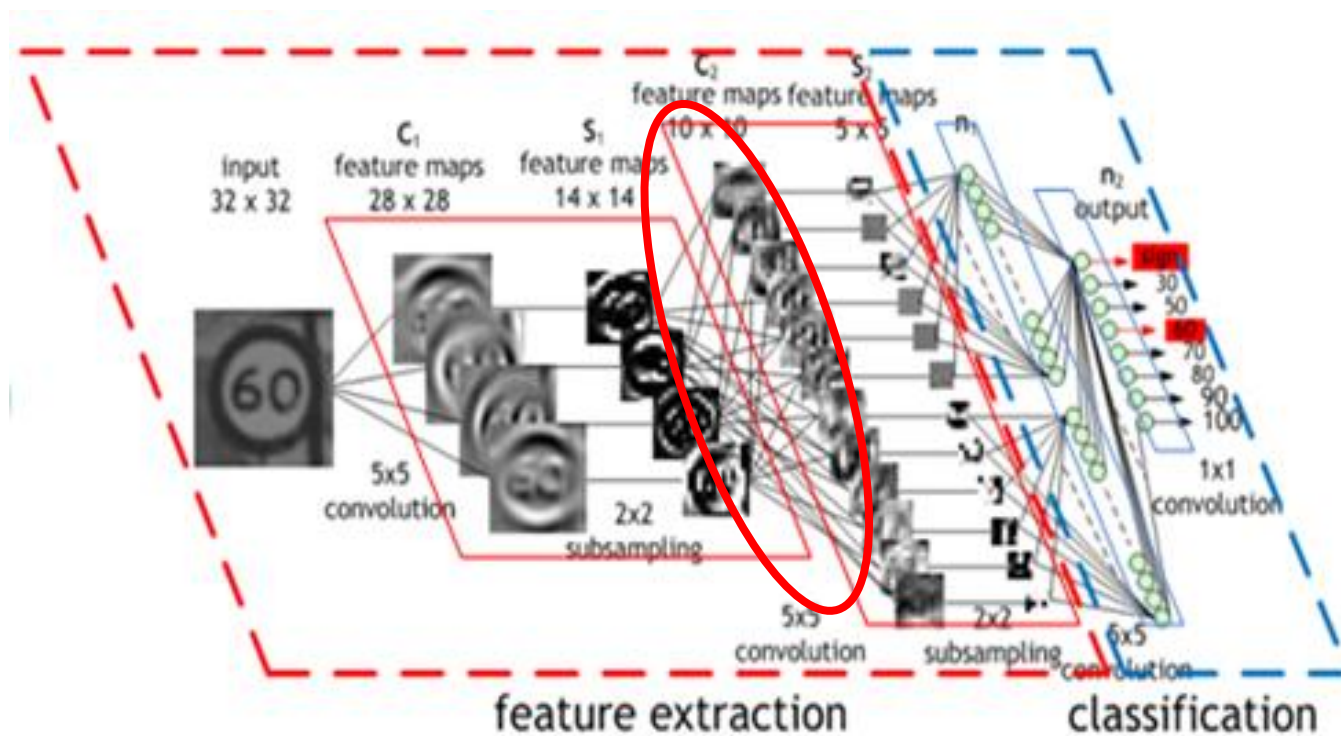
卷积和子采样过程:

- **卷积过程包括:** 用一个可训练的滤波器 f_x 去卷积一个输入的图像、然后加一个偏置 b_x , 得到卷积层 C_x 。
- **子采样过程包括:** 每邻域四个像素pooling变为一个像素, 然后通过标量 w_{x+1} 加权, 再增加偏置 b_{x+1} , 然后通过一个sigmoid激活函数, 产生一个大概缩小四倍的特征映射图 S_{x+1} 。

网络训练时卷积层和池化层作为一个整体

5.1 卷积神经网络结构

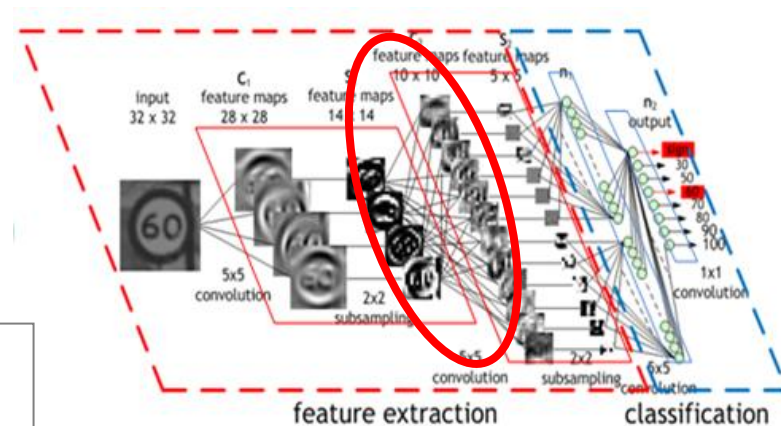
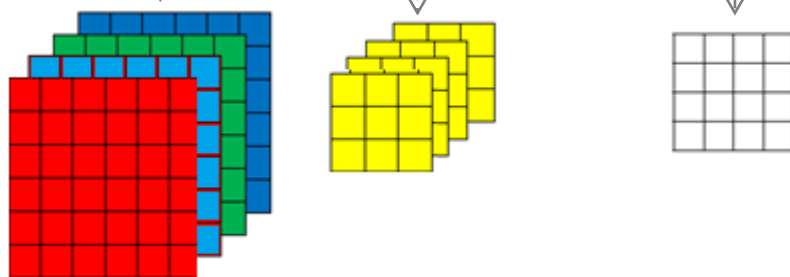
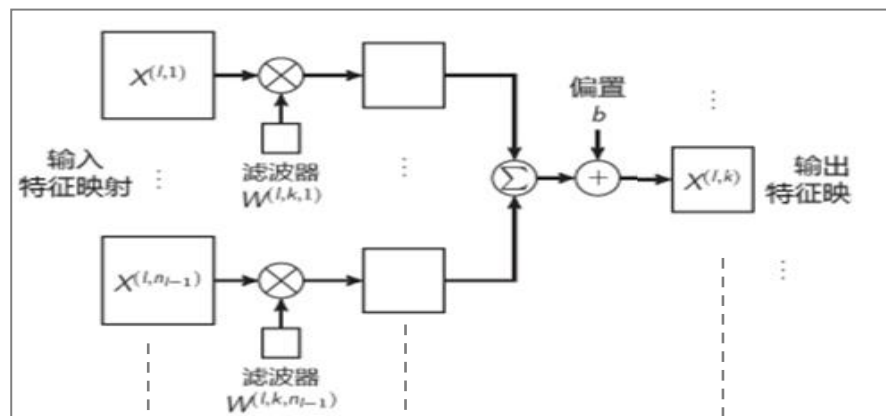
多通道卷积



卷积层+Pooling

全连接输出

5.1 卷积神经网络结构



5.1 卷积神经网络结构

例：输入为特征图谱 3 维，2个 filter(每个3维)

输入 (3 维)

2个 filter(每个3维)

输出 (特征映射2)

Input (zero-padding) (7x7x3)
 $\times [:, :, 0]$

0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	2	1	1	2	1	0
0	1	1	2	2	0	0
0	2	2	1	0	0	0
0	2	1	2	1	1	0
0	0	0	0	0	0	0

$\times [:, :, 1]$

0	0	0	0	0	0	0
0	1	0	0	1	2	0
0	0	1	0	0	2	0
0	2	1	2	2	1	0
0	1	0	2	1	1	0
0	2	1	1	0	2	0
0	0	0	0	0	0	0

$\times [:, :, 2]$

0	0	0	0	0	0	0
0	0	2	2	2	1	0
0	0	1	1	1	0	0
0	0	0	1	0	1	0
0	2	0	1	1	0	0
0	2	2	2	0	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)
 $w0 [:, :, 0]$

1	1	1
0	-1	0
0	-1	1

$w0 [:, :, 1]$

0	-1	-1
1	-1	-1
1	1	-1

$w0 [:, :, 2]$

0	1	0
-1	-1	-1
-1	-1	0

Bias b0 (1x1x1)
 $b0 [:, :, 0]$

1

Filter W1 (3x3x3)
 $w1 [:, :, 0]$

0	1	1
-1	-1	-1
0	0	1

$w1 [:, :, 1]$

-1	1	0
1	0	0
1	-1	-1

$w1 [:, :, 2]$

0	0	-1
-1	-1	0
-1	-1	0

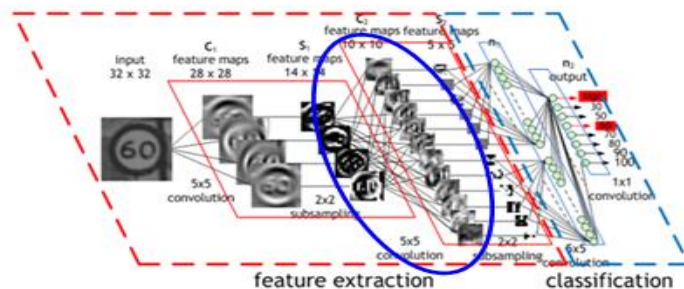
Bias b1 (1x1x1)
 $b1 [:, :, 0]$

0

Output (3x3x2)
 $o [:, :, 0]$

-5		

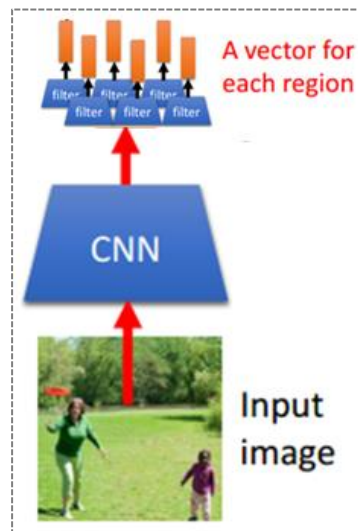
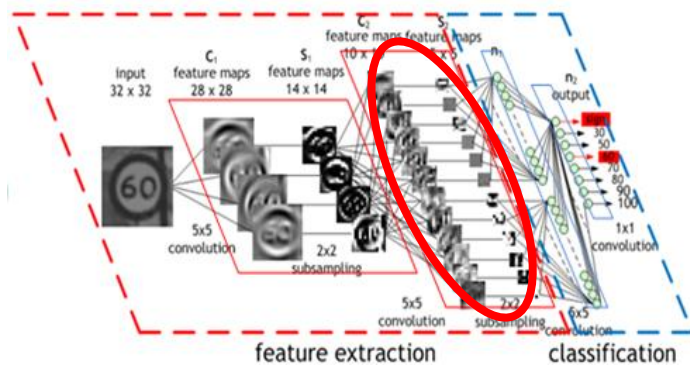
$o [:, :, 1]$



5.1 卷积神经网络结构

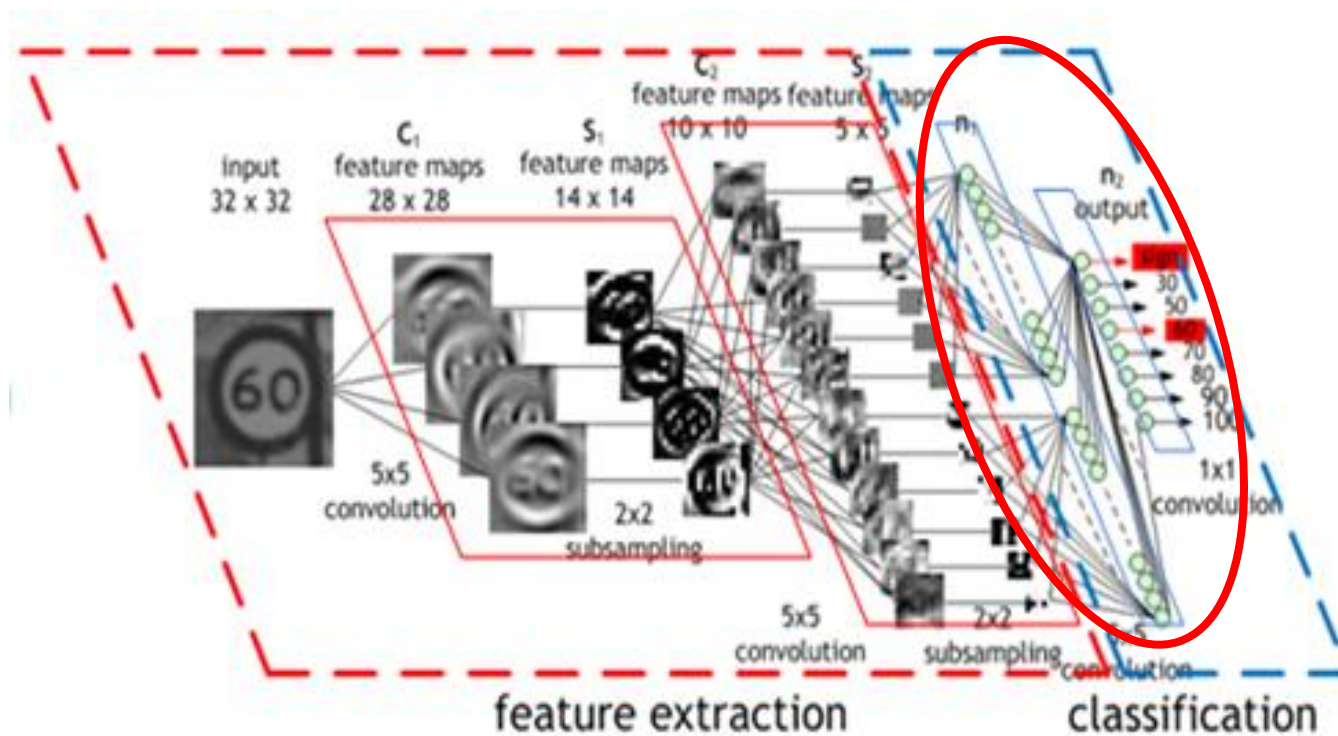
最后的池化层 (Pooling)

最后的池化层是提取出来图像的不同的特征，用这些特征可以进行整体图像的分类，也可以分别用这些特征完成不同的任务



5.1 卷积神经网络结构

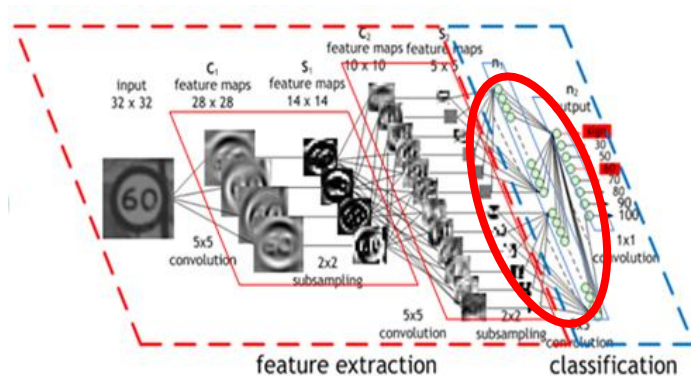
全连接层



卷积层+Pooling

全连接输出

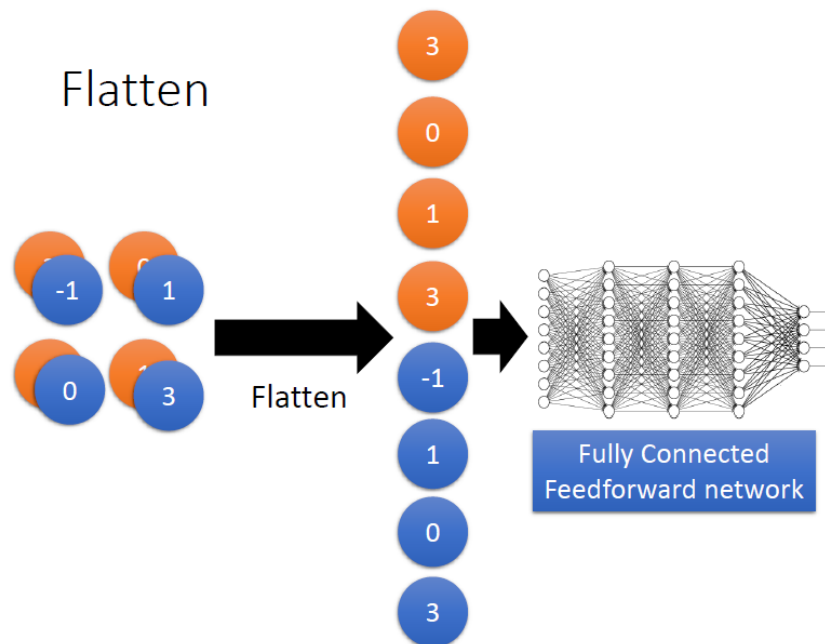
5.1 卷积神经网络结构



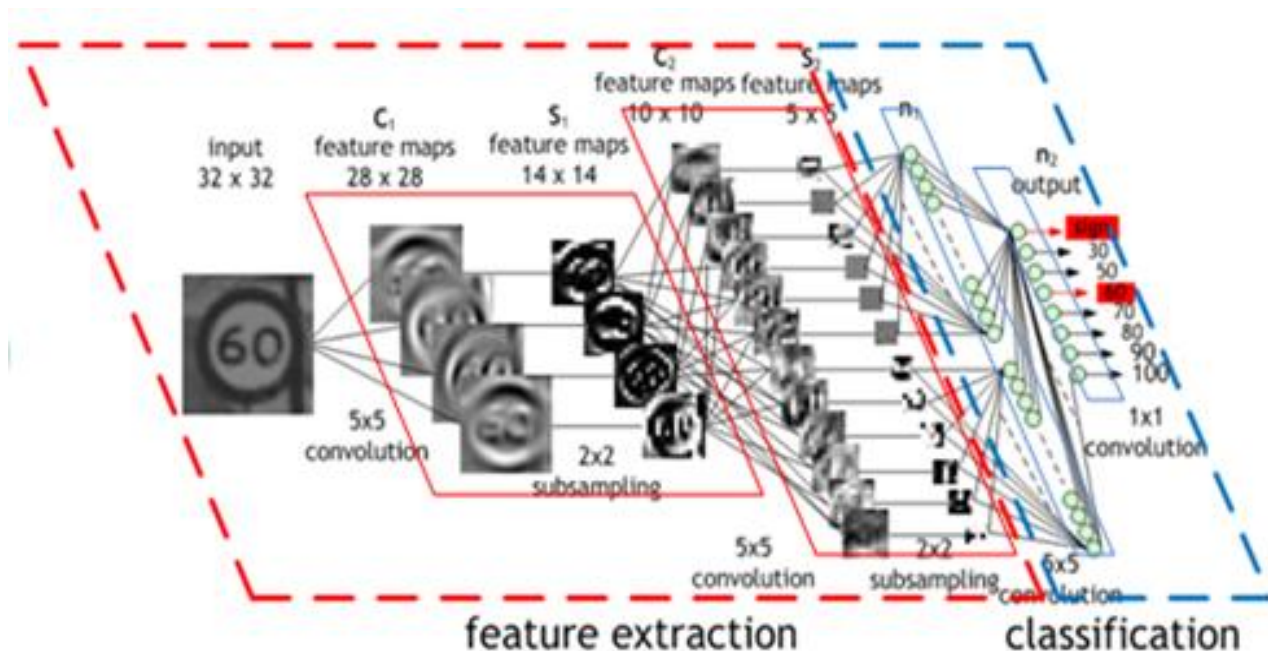
方法:

将最后池化层的单元“平化”

然后组成全连接输入网



5.1 卷积神经网络结构



CNN网络结构:

输入: X (根据具体需要)

输出: Y (根据具体需要设定)

参数: 各卷积层中各过滤器值和偏置; 卷积层到池化层权重和偏置;
全连接输出网各连接权重和偏置

5.1 卷积神经网络结构

卷积神经网络有三个结构上的特性：

- 1 局部连接
- 2 权重共享
- 3 空间或时间上的次采样

这些特性使得卷积神经网络具有一定程度上的
平移、缩放和扭曲不变性

内 容 提 要

5.1 概述

5.2 卷积神经网络结构

5.3 卷积神经网络学习

5.4 卷积神经网络应用

5.2 卷积神经网络学习

CNN-Mnist涉及的变量

超参数

迭代轮数
学习率
batch大小
卷积filter大小
卷积步长
特征图个数
池化大小

权重变量

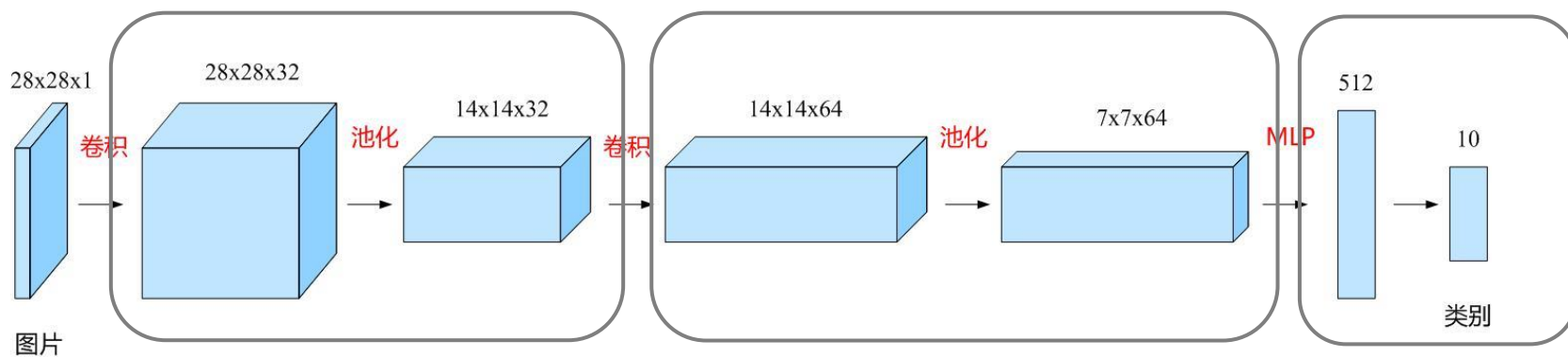
卷积filter的权值
卷积filter的偏置
全连接的权值

状态变量

输入图片数据
输入图片对应类别

5.2 卷积神经网络学习

CNN-Mnist框架



内 容 提 要

5.1 概述

5.2 卷积神经网络结构

5.3 卷积神经网络学习

5.4 卷积神经网络应用

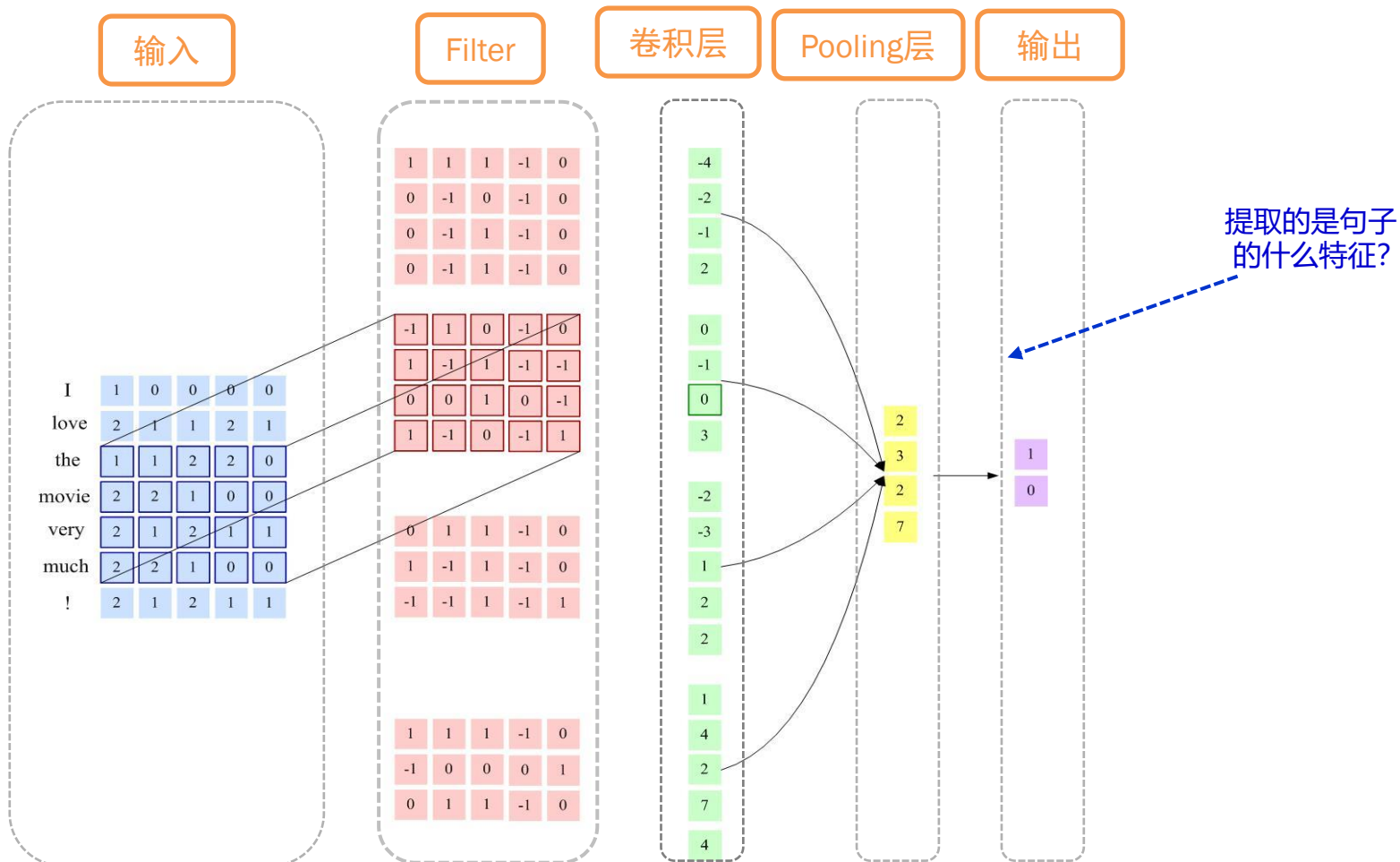
5.3 卷积神经网络应用

CNN在NLP中应用:

- 各种分类任务：文本分析、情感分析、实体关系抽取等等
- 用于其它任务的特征提取

5.3 卷积神经网络应用

例： 情感分类

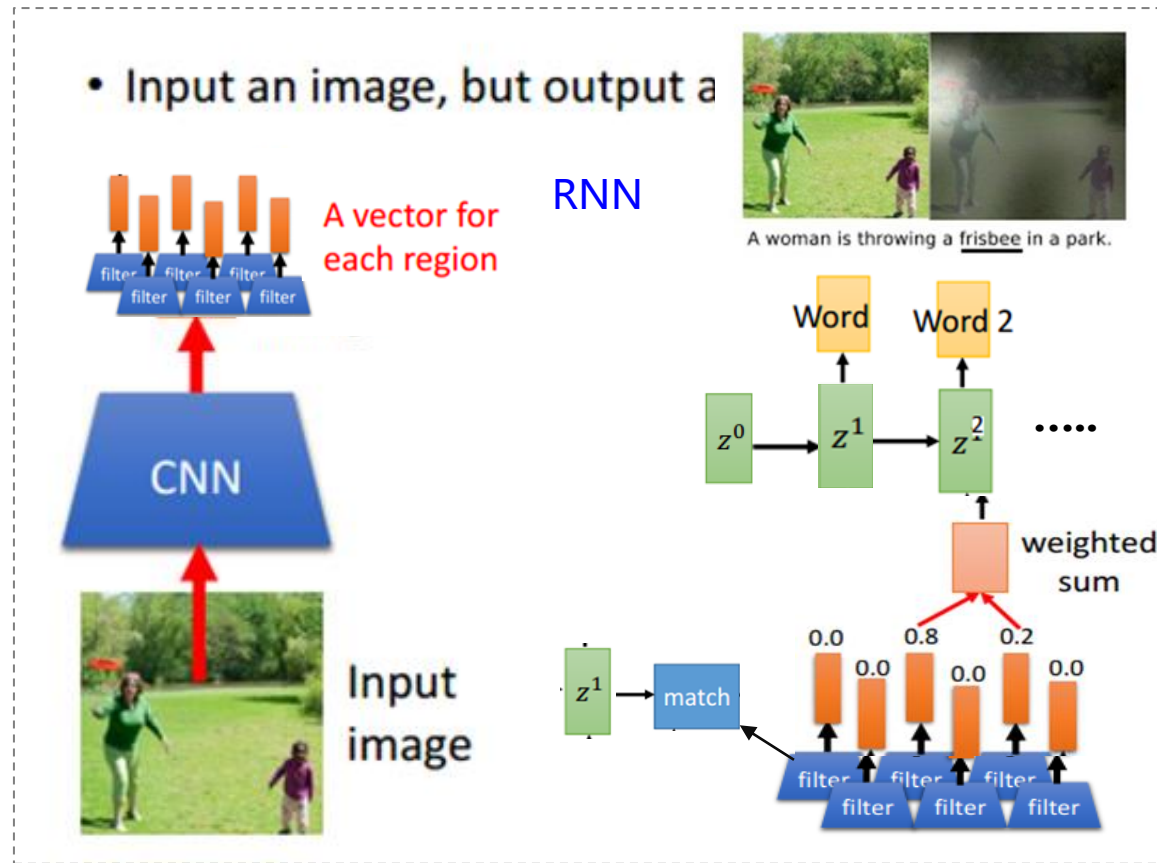


paper: <http://arxiv.org/pdf/1510.03820v4.pdf>

code: <https://github.com/dennybritz/cnn-text-classification-tf>

5.3 卷积神经网络应用

例： 图片标题生成



参考文献:

李宏毅课程

http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML16.html

邱锡鹏, 《神经网络与深度学习》讲义

刘鹏飞, 卷积神经网络和递归神经网络实践

在此表示感谢!

谢谢各位！



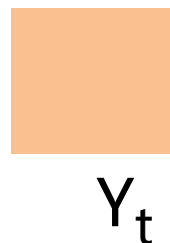
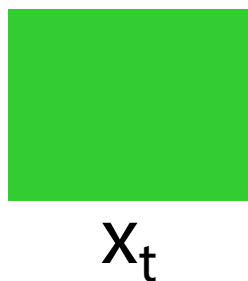
Q&A

附录： 卷积运算

卷积：

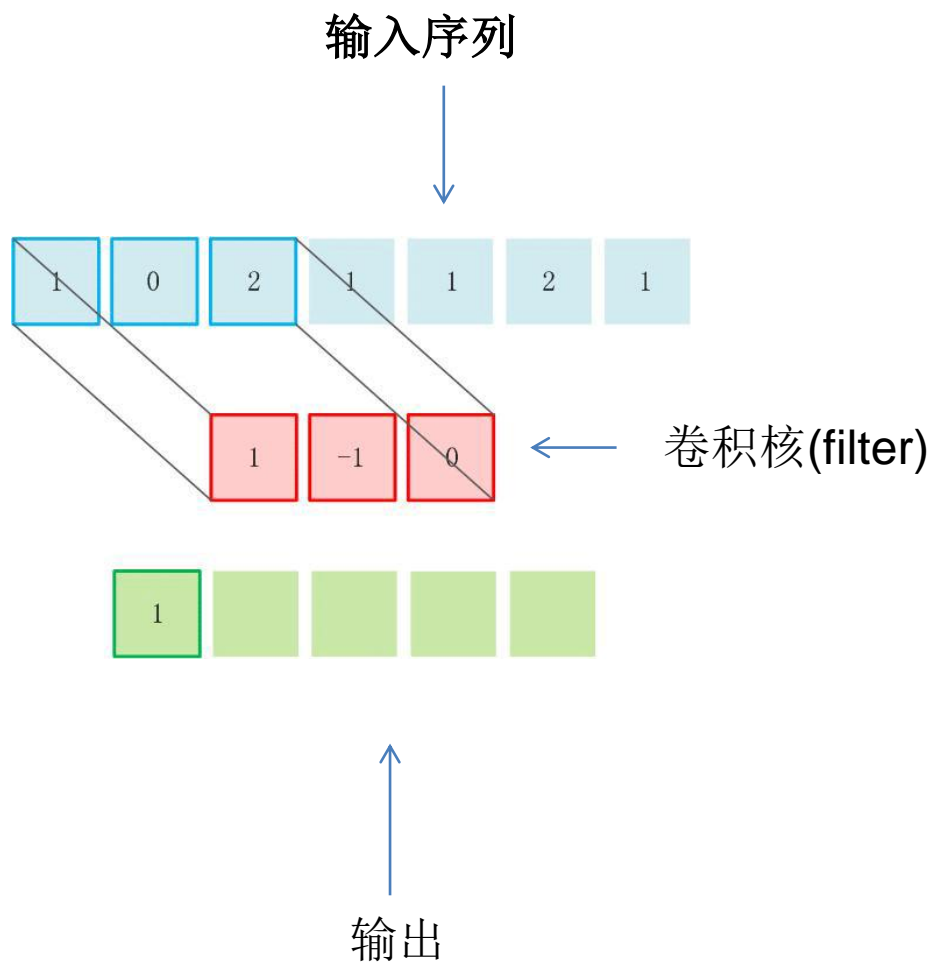
是分析数学中一种重要的运算（这里只考虑离散序列的情况）。一维卷积经常用在信号处理中。给定一个输入信号序列 $x_t, t = 1, \dots, n$, 和滤波器 $f_t, t = 1, \dots, m$, 一般情况下滤波器的长度 m 远小于信号序列长度 n 。卷积的输出为：

$$y_t = \sum_{k=1}^m f_k \cdot x_{t-k+1}$$



本质上是一种加权求和运算

一维卷积示例



通过计算中心像素点以及相邻像素点的加权和来构成特征映射，实现空间特征的提取

- **卷积核大小**：输入中每次提取特征区域大小
- **卷积核中参数**：卷积运算中的一组共享参数，该参数随机化初值，然后根据误差函数通过反向传播梯度下降进行迭代优化，通过优化求出的参数才能实现特征提取的作用

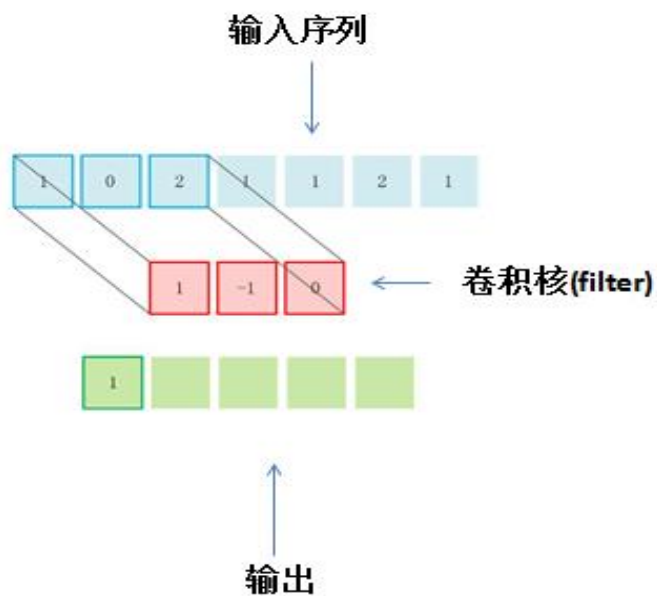
一维卷积

当输入长度为 n 时，与输出长度与下列参数有关：

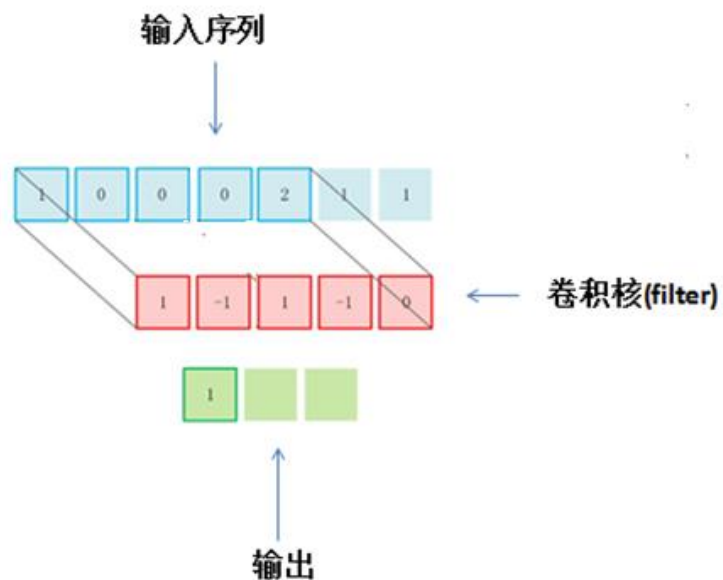
1. 卷积filter大小(Filter size = f)

例：

卷积核为 3



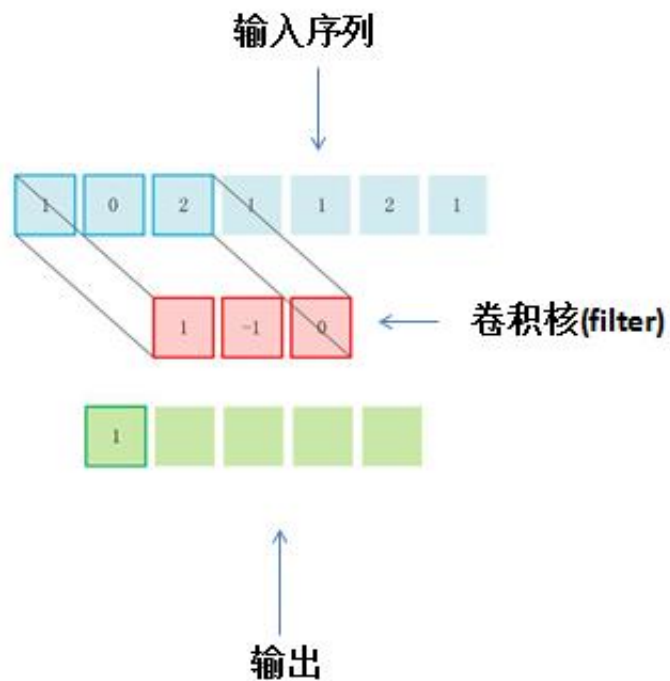
卷积核为 5



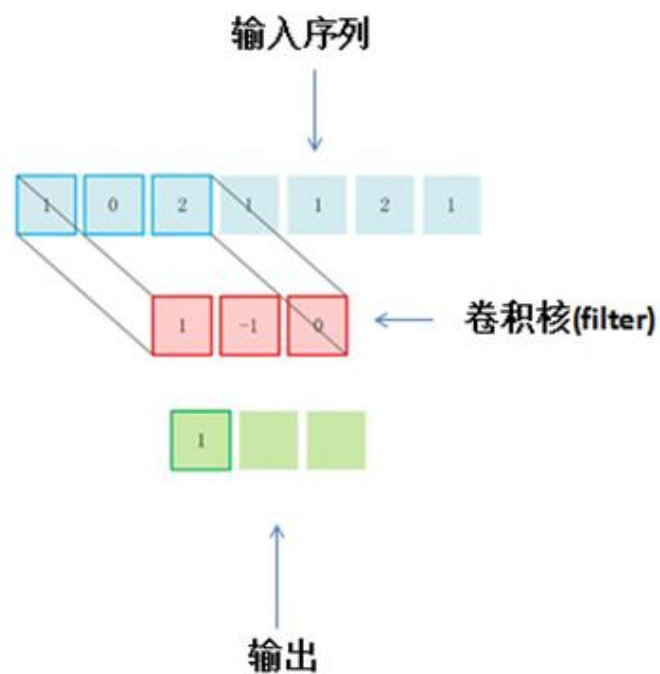
一维卷积

2. 步长(Stride size = s)

例: 步长为 1



步长为 2



一维卷积

3. 填充 (Padding size = p)

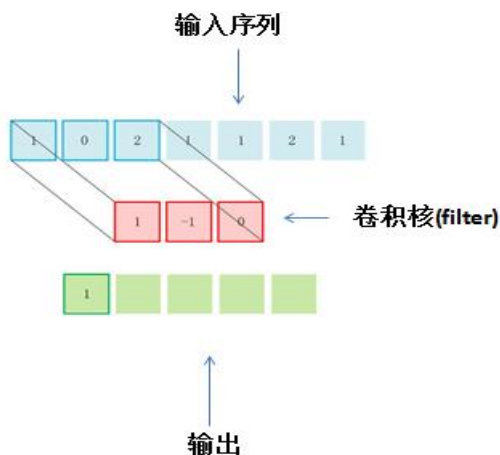
卷积的结果按输出长度不同可以分为两类：

窄卷积：输出长度 $(n - f)/s + 1$ ，不补零。

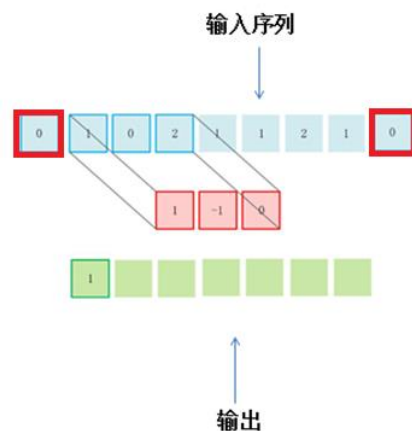
等长卷积：输出长度 n ，对于不在 $[1, n]$ 范围之外的 x_t 用零补齐

在这里除了特别声明，我们一般说的卷积默认为**窄卷积**。

例： 不填充 (窄卷积)



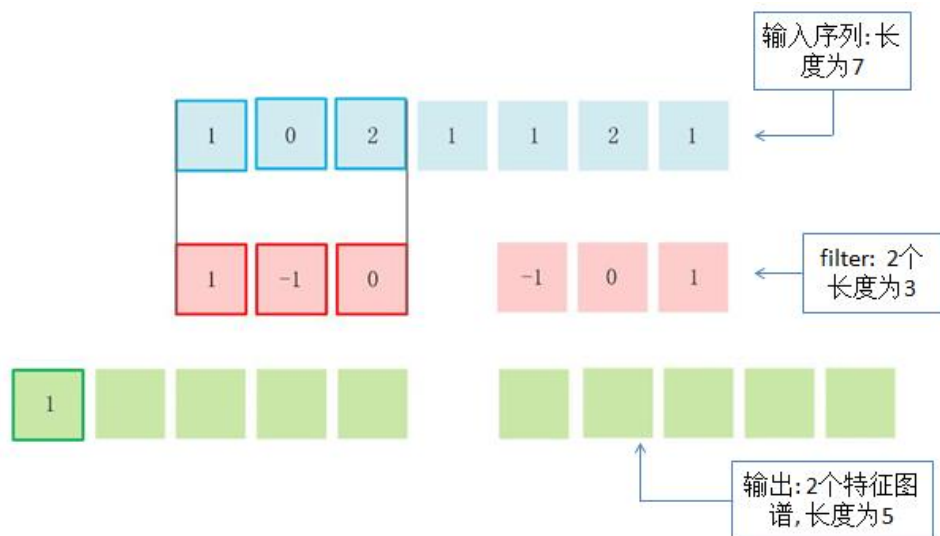
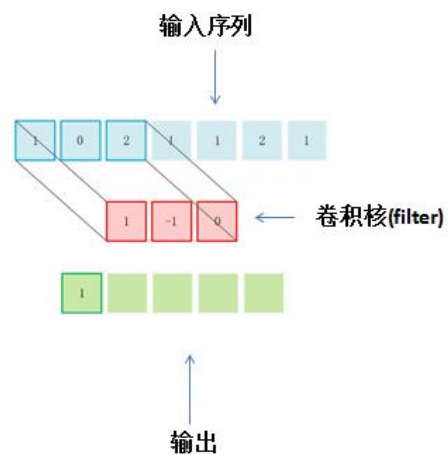
填充 (等长卷积)



$$\text{特征图谱输出长度} = (n + 2p - f) / s + 1$$

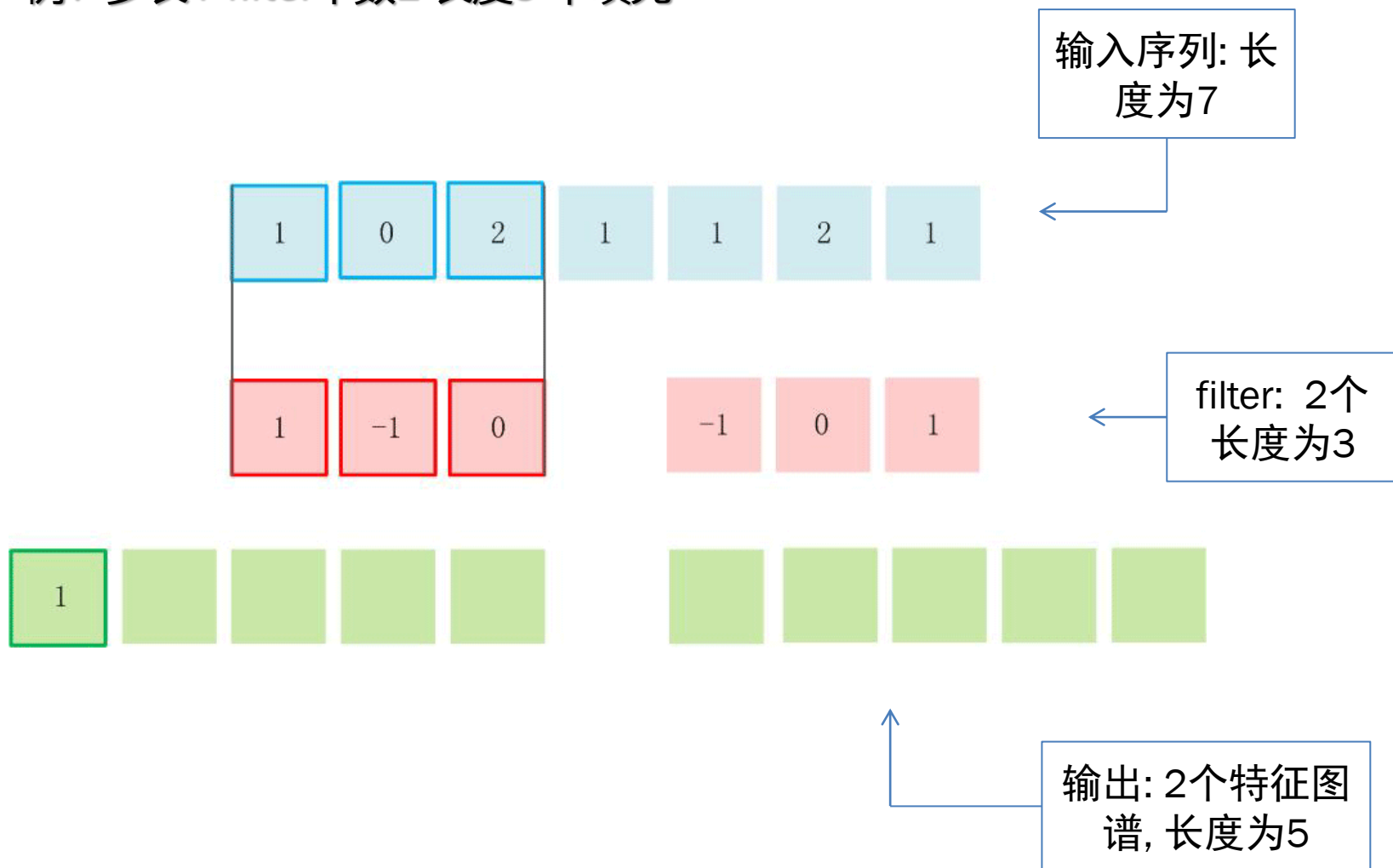
一维卷积

4. 卷积filter个数



一维卷积

例：步长1 filter个数2 长度3 不填充



二维卷积

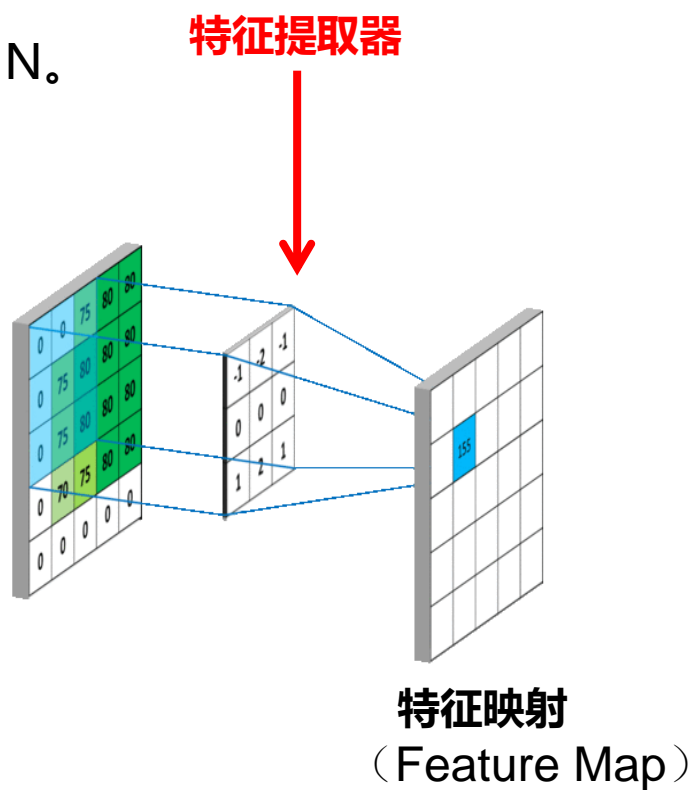
二维卷积经常用在图像处理中。给定一个图像

x_{ij} , $1 \leq i \leq M$, $1 \leq j \leq N$, 和滤波器 f_{ij} ,

$1 \leq i \leq m$, $1 \leq j \leq n$, 一般 $m \ll M$; $n \ll N$ 。

卷积的输出为：

$$y_{ij} = \sum_{u=1}^m \sum_{v=1}^n f_{uv} \cdot x_{i-u+1, j-v+1}.$$



二维卷积

例：步长1 filter个数1 3*3 不填充

Input (zero-padding) (5x5)

$x[:, :]$

1	0	0	0	0
2	1	1	2	1
1	1	2	2	0
2	2	1	0	0
2	1	2	1	1

Filter W (3x3)

$w[:, :]$

1	1	1
0	-1	0
0	-1	1

Output (3x3)

$o[:, :]$

1		

步长 = 1

二维卷积

例：步长2 filter个数1 3*3 填充

Input (zero-padding) (7x7)

$x[:, :]$

0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	2	1	1	2	1	0
0	1	1	2	2	0	0
0	2	2	1	0	0	0
0	2	1	2	1	1	0
0	0	0	0	0	0	0

Filter W (3x3)

$w[:, :]$

1	1	1
0	-1	0
0	-1	1

Output (3x3)

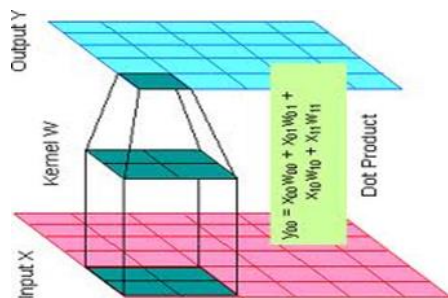
$o[:, :]$

-2		

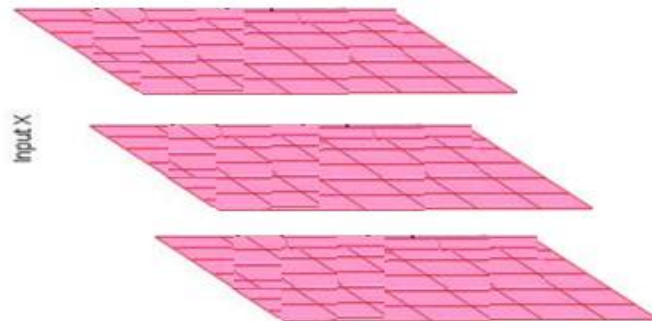
步长 = 2

多通道卷积

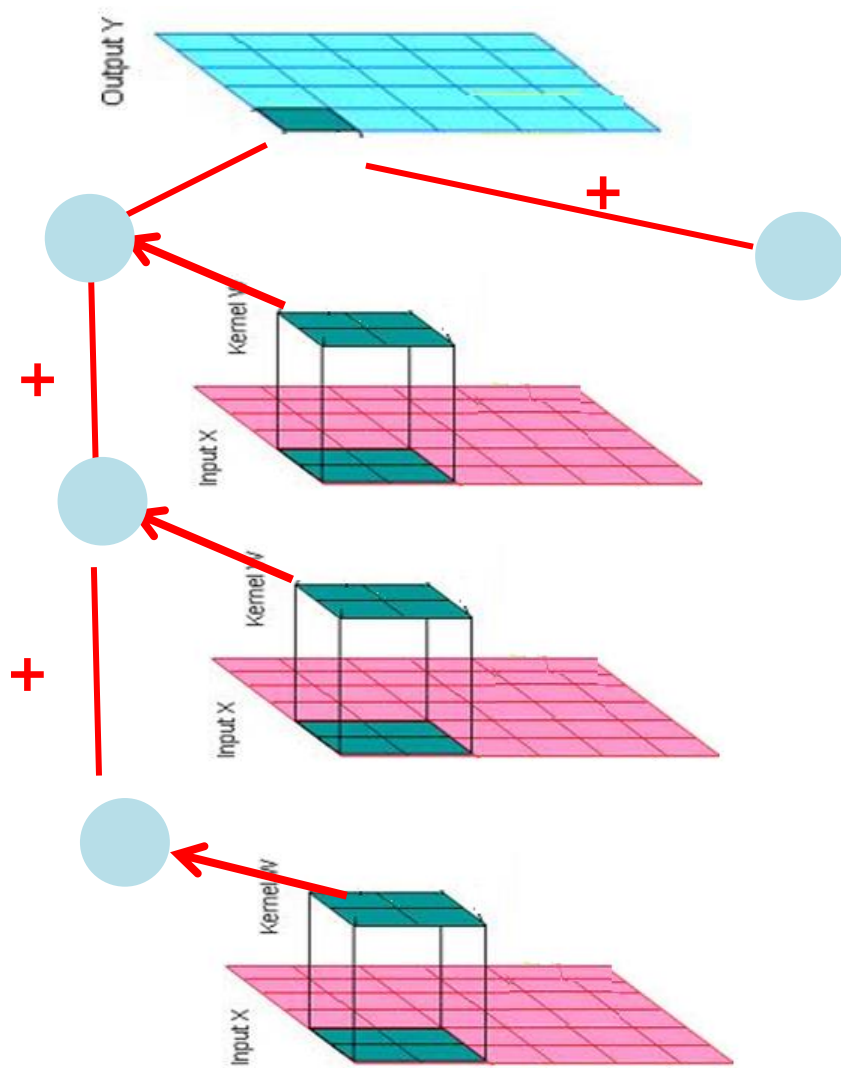
二维卷积



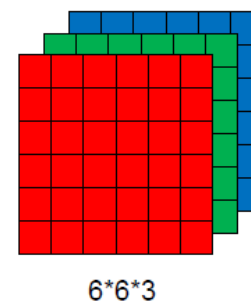
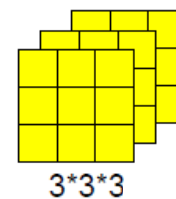
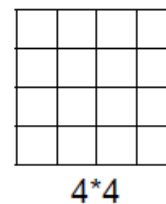
三维 (3通道) 卷积



三维 (3通道) 卷积



偏置



4.3 梯度下降法

问题引入:

知识以参数的形式记忆在模型中

如何学习参数 (确定模型的参数) ?

如: $y = ax + b$ 如何确定 a, b ?

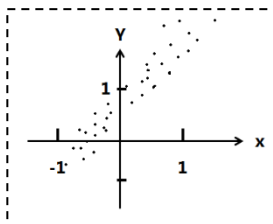
通过训练数据学习参数

有监督训练 给定实例 $(x^i; \hat{y}^i)$

如: $(1,3), (2,5), (3,7) \dots$

有: $a = 2; b = 1$

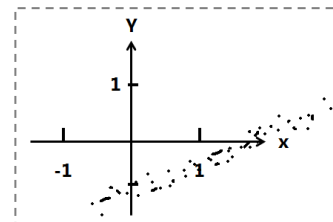
模型: $y = 2x + 1$



如: $(2,0), (4,1), (6,2) \dots$

有: $a = 1/2; b = -1$

模型: $y = (1/2)x - 1$



4.4 反向传播算法

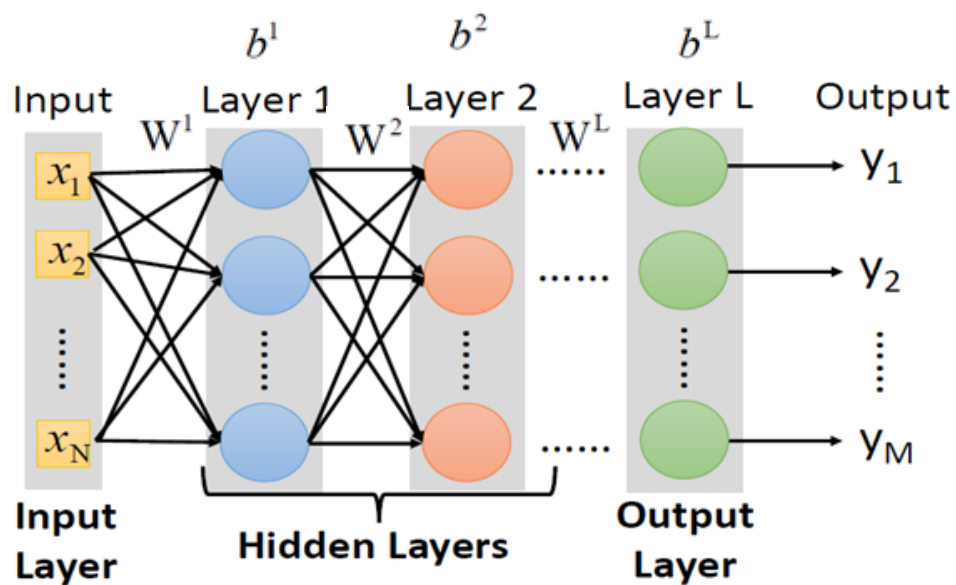
前馈神经网络：

有监督训练

给定实例 (x^i, \hat{y}^i)

如何求 θ ?

输入： x^i



模型输出： y^i

$$y = f(x) = \sigma(W^L \dots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \dots + b^L)$$

$$\theta = \{W^1, b^1, W^2, b^2, \dots, W^L, b^L\}$$

三维卷积

输入特征映射为 3，输出特征映射为2

Input (zero-padding) (7x7x3)

$x[:, :, 0]$

0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	2	1	1	2	1	0
0	1	1	2	2	0	0
0	2	2	1	0	0	0
0	2	1	2	1	1	0
0	0	0	0	0	0	0

$x[:, :, 1]$

0	0	0	0	0	0	0
0	1	0	0	1	2	0
0	0	1	0	0	2	0
0	2	1	2	2	1	0
0	1	0	2	1	1	0
0	2	1	1	0	2	0
0	0	0	0	0	0	0

$x[:, :, 2]$

0	0	0	0	0	0	0
0	0	2	2	2	1	0
0	0	1	1	1	0	0
0	0	0	1	0	1	0
0	2	0	1	1	0	0
0	2	2	2	0	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

$w0[:, :, 0]$

1	1	1
0	-1	0
0	-1	1

$w0[:, :, 1]$

0	-1	-1
1	-1	-1
1	1	-1

$w0[:, :, 2]$

0	1	0
-1	-1	-1
-1	-1	0

Bias b0 (1x1x1)

$b0[:, :, 0]$

1

Filter W1 (3x3x3)

$w1[:, :, 0]$

0	1	1
-1	-1	-1
0	0	1

$w1[:, :, 1]$

-1	1	0
1	0	0
1	-1	-1

$w1[:, :, 2]$

0	0	-1
-1	-1	0
-1	-1	0

Bias b1 (1x1x1)

$b1[:, :, 0]$

0

Output (3x3x2)

$o[:, :, 0]$

-5		

$o[:, :, 1]$
