



# 卷积神经网络

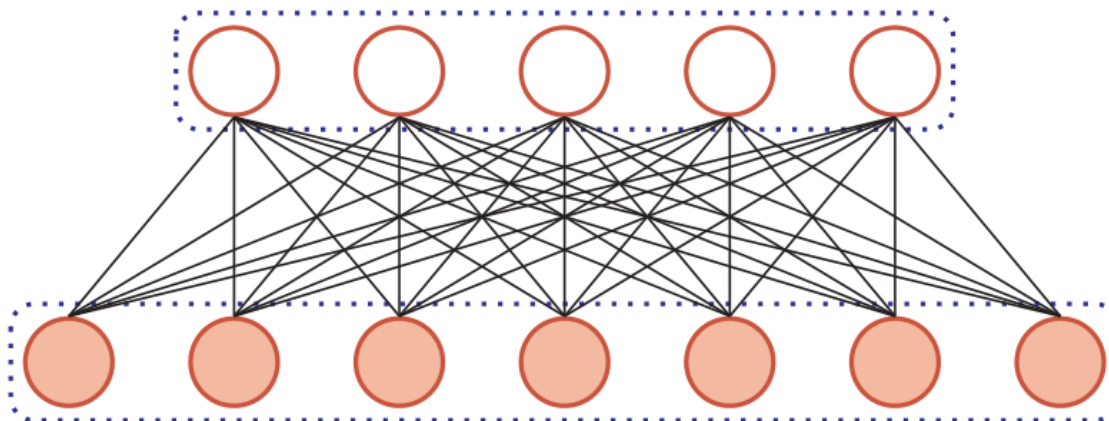
# Convolutional Neural Network

陈昱夫



# 全连接神经网络的缺点

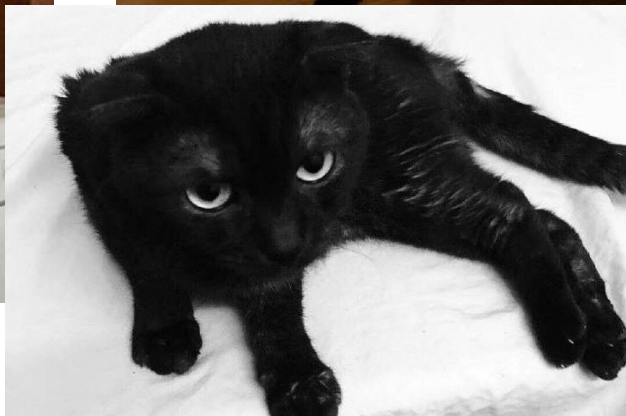
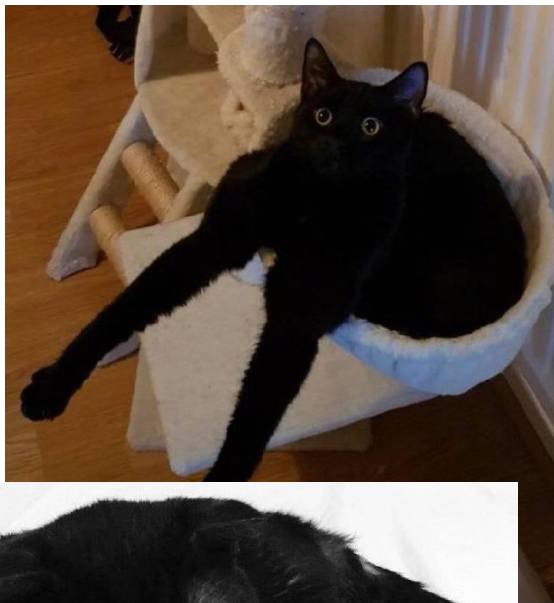
- 权重矩阵的参数非常多





# 全连接神经网络的缺点

- 无法提取局部不变性特征
  - 物体的大小、平移、旋转等





# 卷积神经网络 (CNN)

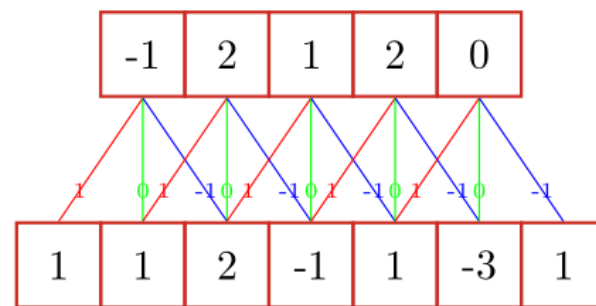
- 基于生物学上感受野 (Receptive Field) 的机制
  - 在视觉神经系统中，一个神经元的感受野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。
- 卷积神经网络有三个结构上的特性
  - 局部连接
  - 权重共享
  - 空间上的下采样



# 卷积

- 一维卷积经常用在信号处理中，用于计算信号的延迟累积。
- 假设一个信号发生器每个时刻 $t$ 产生一个信号 $x^t$ ， $k$ 个时间后的信息衰减率为 $w^k$ ，则当前信号输出 $y_t$ 为：

$$y_t = \sum_{k=1}^m w_k \times x_{t-k+1}$$



Filter: [-1,0,1]



# 卷积

- 二维卷积经常用在图像处理中。因为图像通常为一个二维结构，所以需要将一维卷积进行扩展。

$$y_{i,j} = \sum_{u=1}^m \sum_{v=1}^n w_{u,v} \times x_{i-u+1,j-v+1}$$

1	1	1	1	1
-1	0	-3	0	1
2	1	1	-1	0
0	-1	1	2	1
1	2	1	1	1

$\otimes$

1	0	0
0	0	0
0	0	-1

$=$

0	-2	-1
2	2	4
-1	0	0

*Note: In the original image, the 3x3 kernel and the resulting 3x3 output are highlighted with blue borders. The diagram also includes small annotations like 'x-1', 'x0', and 'x1' indicating the sliding window and weights.*



# 卷积

- 实际上二维卷积已经广泛地被用在图像处理中，可以起到特征提取的作用。
- 能不能不通过先验经验构建滤波器，而是让模型自动学习如何提取特征？



原始图像

高斯平滑

$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

=



拉普拉斯算子

$$\otimes \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} =$$



Robinson算子

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

=



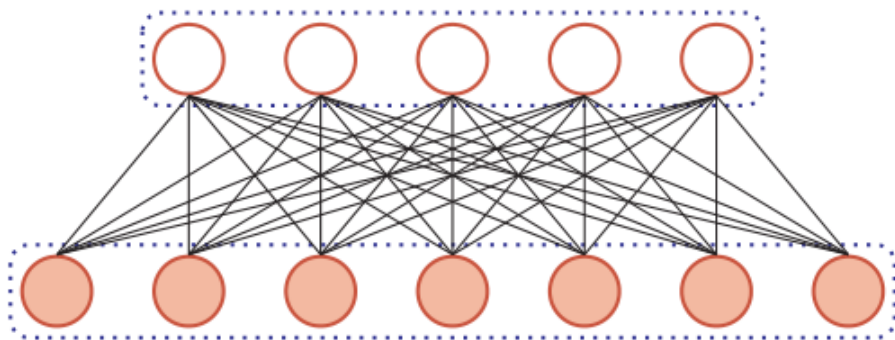
滤波器

输出图像

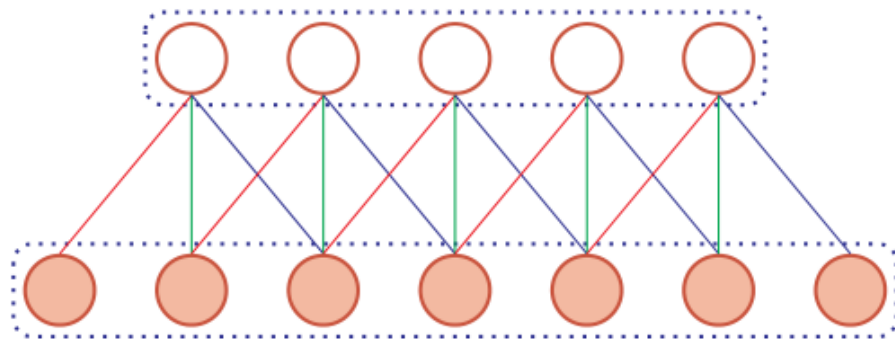


# 卷积层

- 用卷积层代替原来的全连接层。
  - 局部连接
  - 权重共享



(a) 全连接层



(b) 卷积层



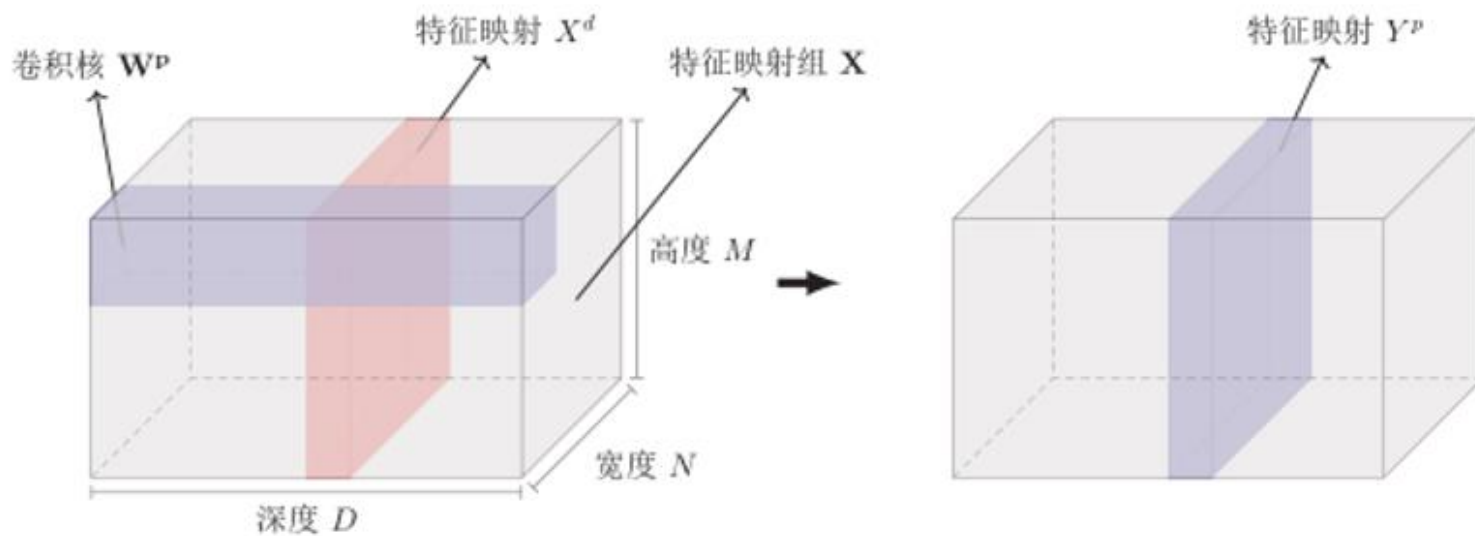


# 卷积层

- 卷积层的结构。

- 输入：D个特征输入  $M \times N \times D$
- 输出：P个特征映射  $M' \times N' \times P$
- 卷积核：D × P个二维卷积核  $D \times P \times m \times n$

大小  $M \times N$ ，D为深度，输入层的D可以理解为图像的通道（比如RGB），卷积层到卷积层之间的深度可以理解为上一层提取的特征数量

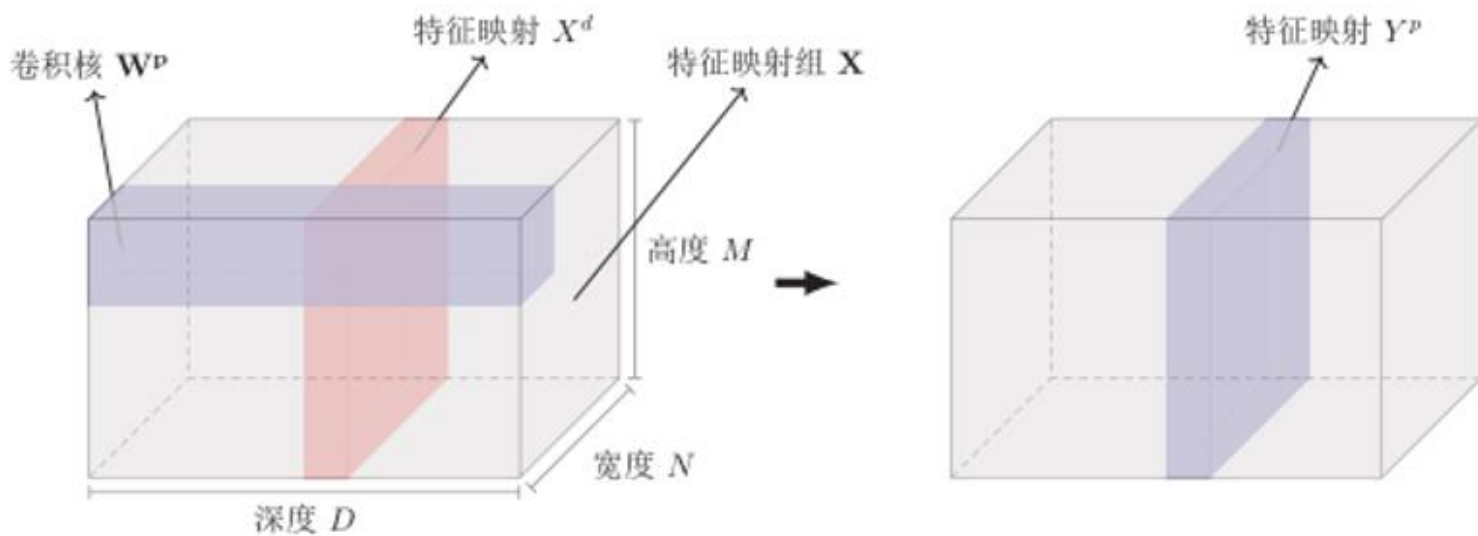




# 卷积层

- $Y^p$ 的计算流程：先用卷积核 $W^{p,1}, W^{p,2}, W^{p,3}, \dots, W^{p,D}$ 分别对输入特征 $X^1, X^2, X^3, \dots, X^D$ 卷积，然后将卷积的结果相加，并加上一个标量偏置 $b^p$ 得到 $Z^p$ ，最后通过某个激活函数得到输出 $Y^p$

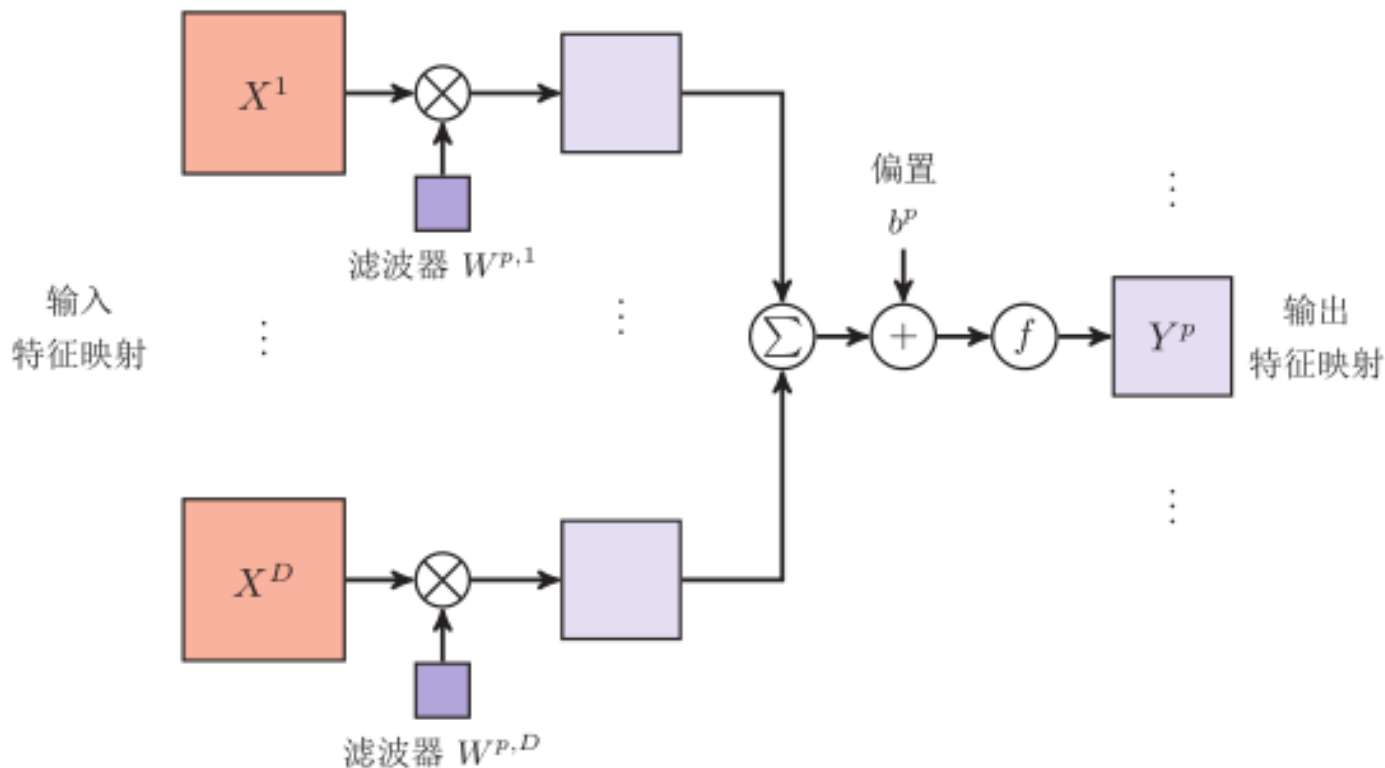
$$Y^p = f(Z^p) = f\left(\sum_{d=1}^D W^{p,d} \otimes X^d + b^p\right)$$





# 卷积层

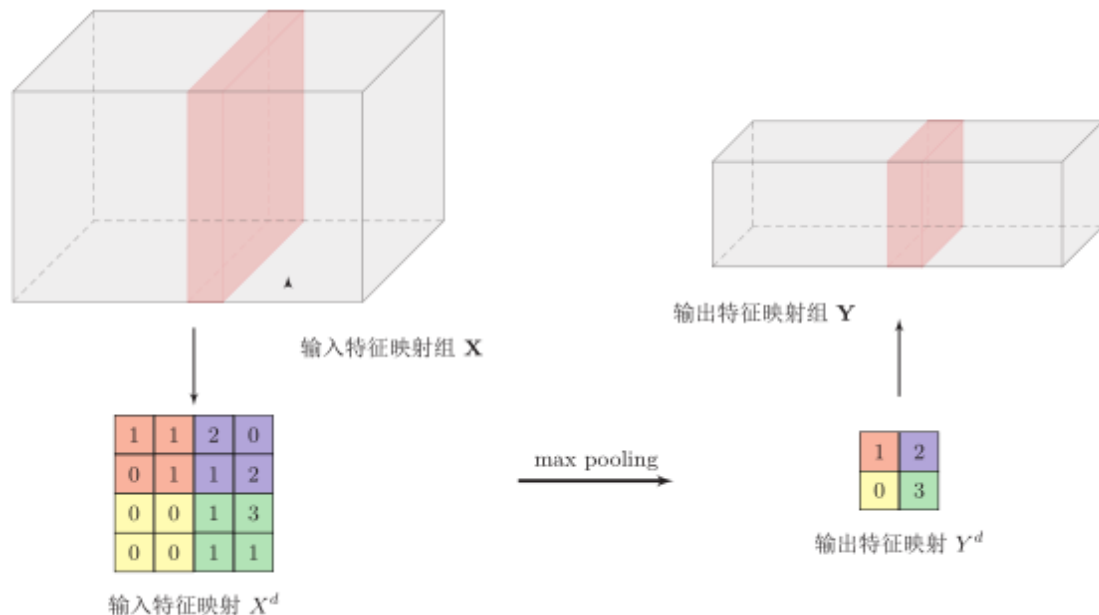
- $Y^p$ 的计算流程：先用卷积核 $W^{p,1}, W^{p,2}, W^{p,3}, \dots, W^{p,D}$ 分别对输入特征 $X^1, X^2, X^3, \dots, X^D$ 卷积，然后将卷积的结果相加，并加上一个标量偏置 $b^p$ 得到 $Z^p$ ，最后通过某个激活函数得到输出 $Y^p$





# 池化层

- 卷积层虽然可以显著减少网络中参数的数量，但特征映射输出的维度仍然很高。
- 池化层对输入特征映射组进行下采样，进一步的筛选特征。

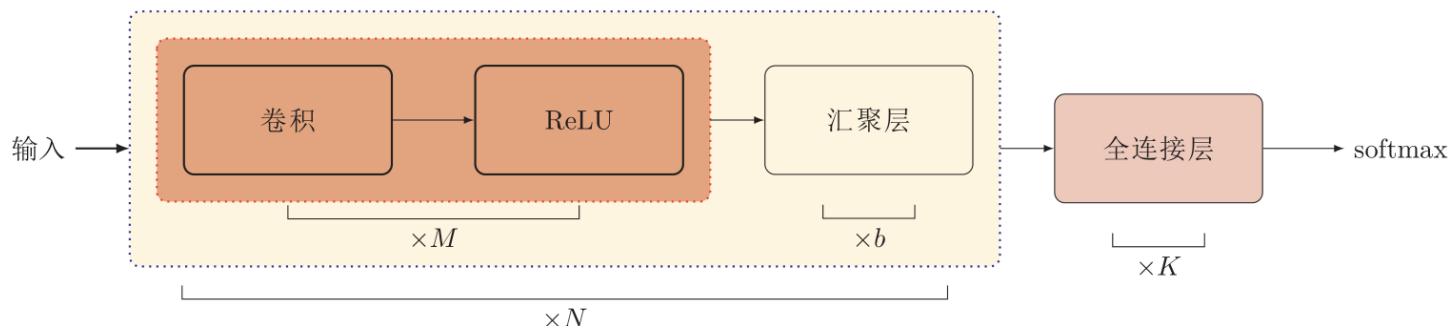


- 最常用到的池化类型是最大池化（max pooling），大小与步长均设置为2，效果相当于原来的特征高度和宽度缩减一半。



# 卷积神经网络结构

- 由卷积层、池化层、全连接层交叉堆叠而成。
- 典型结构

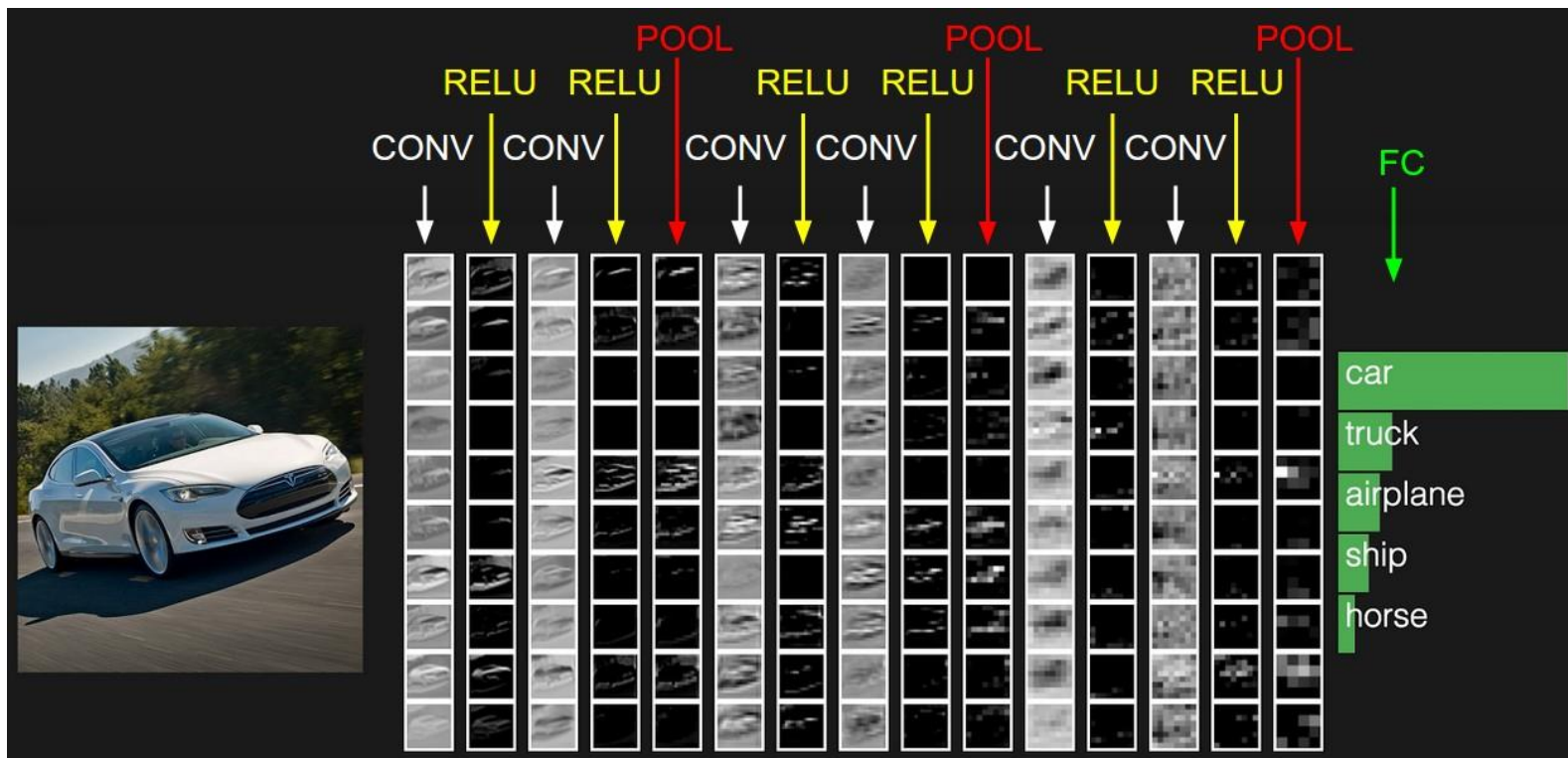


- 定义一个卷积块为连续M个卷积层和b个池化层（M通常设置为2~5，b为0或1）。
- 一个卷积网络中首先堆叠N个连续的卷积块（N的取值区间比较大，比如1~100或者更大）
- 最后再连接K个全连接层（K一般为0~2，相当于最后接一个分类器）



# 卷积神经网络结构

- 由卷积层、池化层、全连接层交叉堆叠而成。
- 典型结构

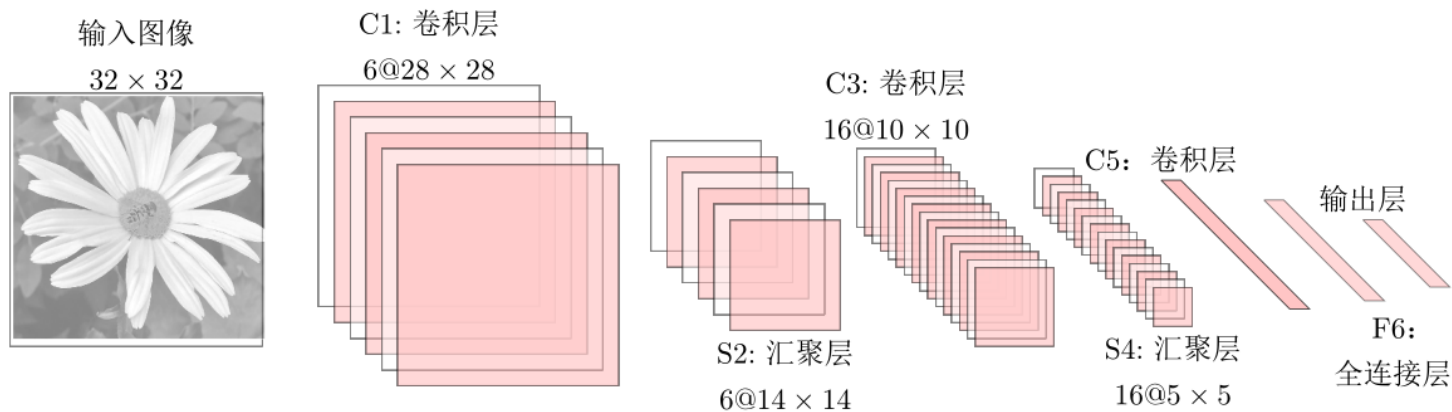




# 卷积神经网络应用

- LeNet-5

- 基于 LeNet-5 的手写数字识别系统在 90 年代被美国很多银行使用，用来识别支票上面的手写数字。





# 卷积神经网络应用

- AlphaGo

- 输入是19x19分辨率的棋局“图像”，输出是361维概率分布向量。其中，概率最大的那个位置就是最佳落子位置。

The input to the policy network is a  $19 \times 19 \times 48$  image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a  $23 \times 23$  image, then convolves  $k$  filters of kernel size  $5 \times 5$  with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a  $21 \times 21$  image, then convolves  $k$  filters of kernel size  $3 \times 3$  with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size  $1 \times 1$  with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used  $k = 192$  filters; [Fig. 2b](#) and [Extended Data Table 3](#) additionally show the results of training with  $k = 128, 256$  and  $384$  filters.

**policy network:**

[19x19x48] Input

CONV1: 192 5x5 filters , stride 1, pad 2 => [19x19x192]

CONV2..12: 192 3x3 filters, stride 1, pad 1 => [19x19x192]

CONV: 1 1x1 filter, stride 1, pad 0 => [19x19] (*probability map of promising moves*)

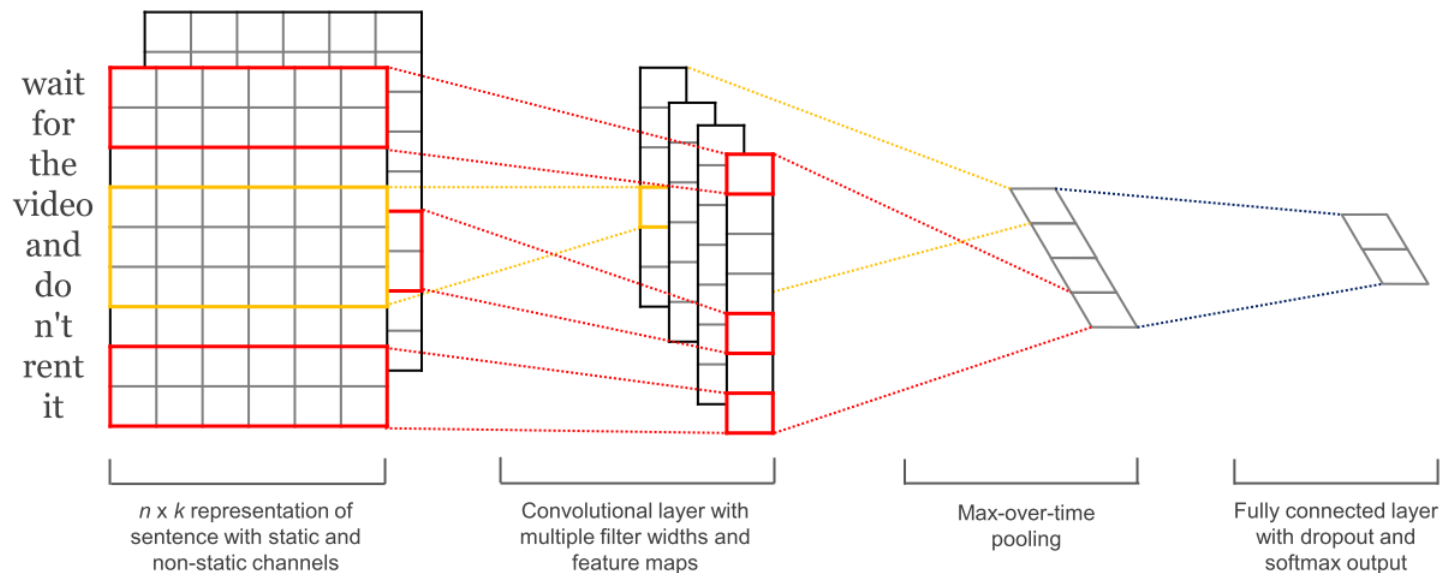




# 卷积神经网络应用

- Sentence Classification

- 输入是某个句子所有单词的词嵌入向量组成的矩阵。





# 期中project

- 包含两个任务
  - 实现CNN模型
  - 实现RNN模型
- 两份数据集
  - 图像识别数据集: CIFAR-10
  - 自然语言数据集: STS Benchmark
- 该Project分组完成, 每组2人



# 期中project

- 报告评分:

- 每组同学共同完成一份报告
- 参考论文要有参考文献, 参考的代码要标记来源
- RNN模型和CNN模型的实现各占50分
- 报告提交DDL: 11月7日晚11:00

评分项	说明	分值
实验原理	总结两种模型的原理	20
网络结构	画出自己模型的网络结构示意图	10
结果分析	展示并分析不同结构下的实验结果	40
创新	可以借鉴现有方法, 但需总结原理	20
排版	整体美观性	10
组员分工	总结组员各自做了什么工作	0



# 期中project

- PPT展示:

- 小组成员共同完成验收, 时间为5~10分钟, 超时扣分
- 通过PPT来展示期中project小组成员完成的工作
- PPT展示会占据一定的期中project分数
- 验收时间: 11月8日实验课

评分项	说明
实现思路	总结这段时间内实现进度
网络结构	介绍自己模型的网络结构
结果分析	展示并分析实验结果
创新	介绍有哪些创新与尝试