

TensorFlow初识

Who am i

- 1, 人工智能学院院长
- 2, 从事机器学习行业, 深度学习行业, 大数据行业。
- 3, 比利时鲁汶大学电子信息系统专业, 世界五百强Barco大数据研发, 新浪微博算法专家
- 4, 老师出自BAT不如让学生走进BAT
- 5, 用接地气的讲述让同学们透彻理解。

TensorFlow概要

- 由Goole Brain开源，设计初衷是加速机器学习的研究
- 2015年11月在GitHub上开源
- 2016年4月分布式版本
- 2017年发布了1.0版本，趋于稳定
- Google希望让这个优秀的工具得到更多的应用，从整体上提高深度学习的效率

Google大量成功项目

- Android系统
- Chromium浏览器
- Go编程语言
- JavaScript引擎V8
- Protobuf数据交换框架
- Bazel编译工具
- OCR工具Tesseract

TensorFlow相关链接

- TensorFlow官方网址: www.tensorflow.org
- GitHub网址: github.com/tensorflow/tensorflow
- 模型仓库网址: github.com/tensorflow/models

支持语言

- Python
- C++
- Go
- Java
- 后端使用C++、CUDA

TensorFlow

- TensorFlow实现的算法可以在众多异构的系统上方便地移植，比如Android手机、iphone、
- 普通的CPU服务器、大规模GPU集群
- 除了执行深度学习算法，TensorFlow还可以用来实现很多其他算法，包括线性回归、逻辑回归、随机森林等
- TensorFlow建立的大规模深度学习模型应用场景也非常广，包括语音识别、自然语言处理、计算机视觉、机器人控制、信息抽取、药物研发、分子活动预测

在Google的应用

- 为了研究超大规模的深度神经网络，Google在2011年启动了Google Brain项目
- 比如Google Search中的搜索结果排序
- Google Photos中的图片标注
- Google Translate中的自然语言处理，都依赖建立的深度学习模型
- 2016年已经有超过2000个项目使用了TensorFlow建立的深度学习模型

核心概念

- TensorFlow中的计算可以表示为一个有向图（Directed Graph）
- 或者称计算图（Computation Graph）
- 其中每一个运算操作（operation）将作为一个节点（node）
- 计算图描述了数据的计算流程，也负责维护和更新状态
- 用户通过python，c++，go，Java语言设计这个这个数据计算的有向图
- 计算图中每一个节点可以有任意多个输入和任意多个输出
- 每一个节点描述了一种运算操作，节点可以算是运算操作的实例化（instance）
- 计算图中的边里面流动（flow）的数据被称为张量（tensor），故得名TensorFlow

代码流程

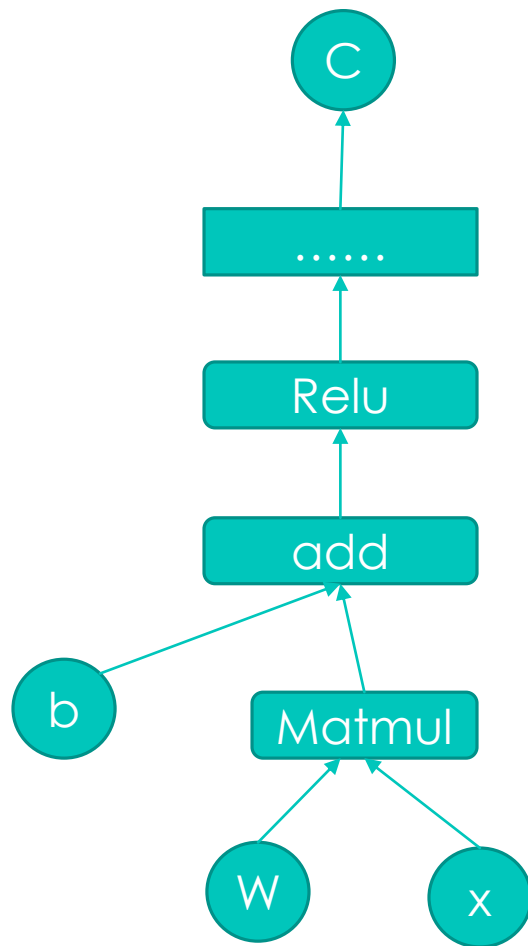
- `Import tensorflow as tf`
- `b = tf.Variable(tf.zeros([100]))`
- `W = tf.Variable(tf.random_uniform([784,100], -1, 1))`
- `x = tf.placeholder(name="x")`
- `relu = tf.nn.relu(tf.matmul(W, x) + b)`
- `Cost = [...]`
- `Sess = tf.Session()`
- `for step in range(0, 10):`
 - `input = ...construct 100-D input array...`
 - `result = sess.run(cost, feed_dict={x: input})`
 - `print(step, result)`

安装tensorflow

- `pip install tensorflow==1.1.0`

```
C:\Program Files\Anaconda3\Scripts>pip install tensorflow==1.1.0  
Collecting tensorflow==1.1.0  
  Downloading tensorflow-1.1.0-cp35-cp35m-win_amd64.whl (19.4MB)  
    63% |███████████| 12.2MB 216kB/s eta 0
```

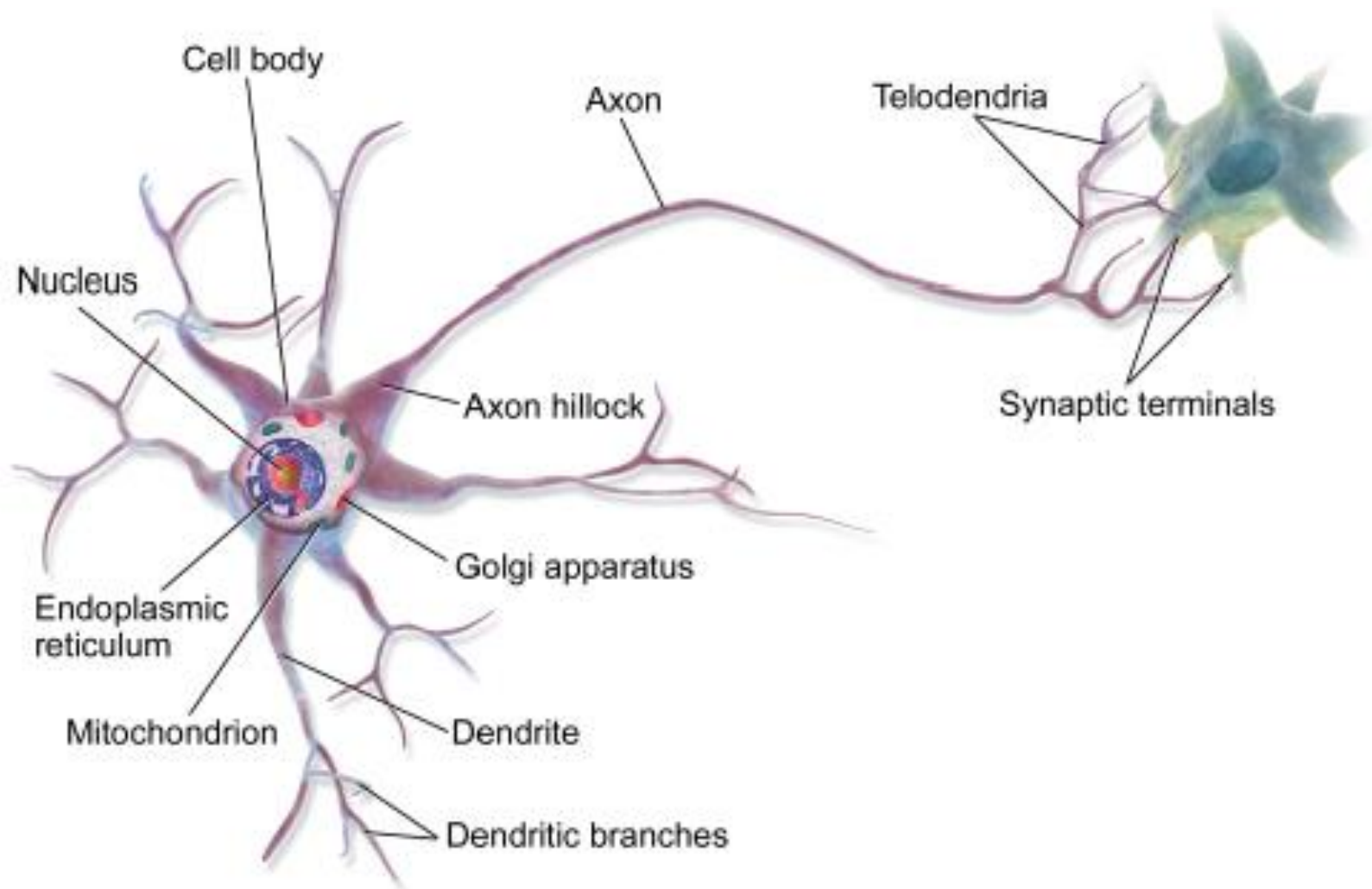
有向无环图DAG



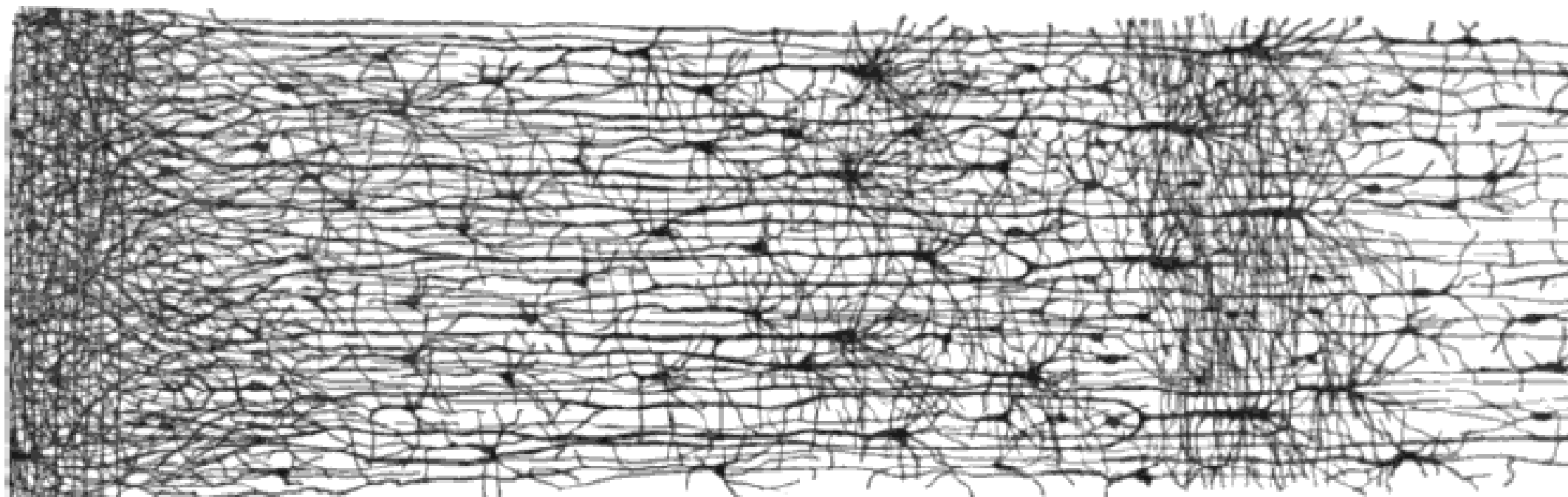
其他深度学习框架

- 深度学习研究的热潮持续高涨，各种开源深度学习框架也层出不穷，其中包括TensorFlow、Caffe、Keras、CNTK、Torch7、MXNet、Leaf、Theano、DeepLearning4j、Lasagne、Neon
- TensorFlow杀出重围，在关注度、用户数上都占据绝对优势，大有一统江湖之势
- TensorFlow在Star数量，Fork数量，Contributor数量这三个数据上都完胜其他对手，主要是Google在业界的号召力确实强大，Google强大的人工智能研发水平，让大家对Google的深度学习框架充满信息

神经元

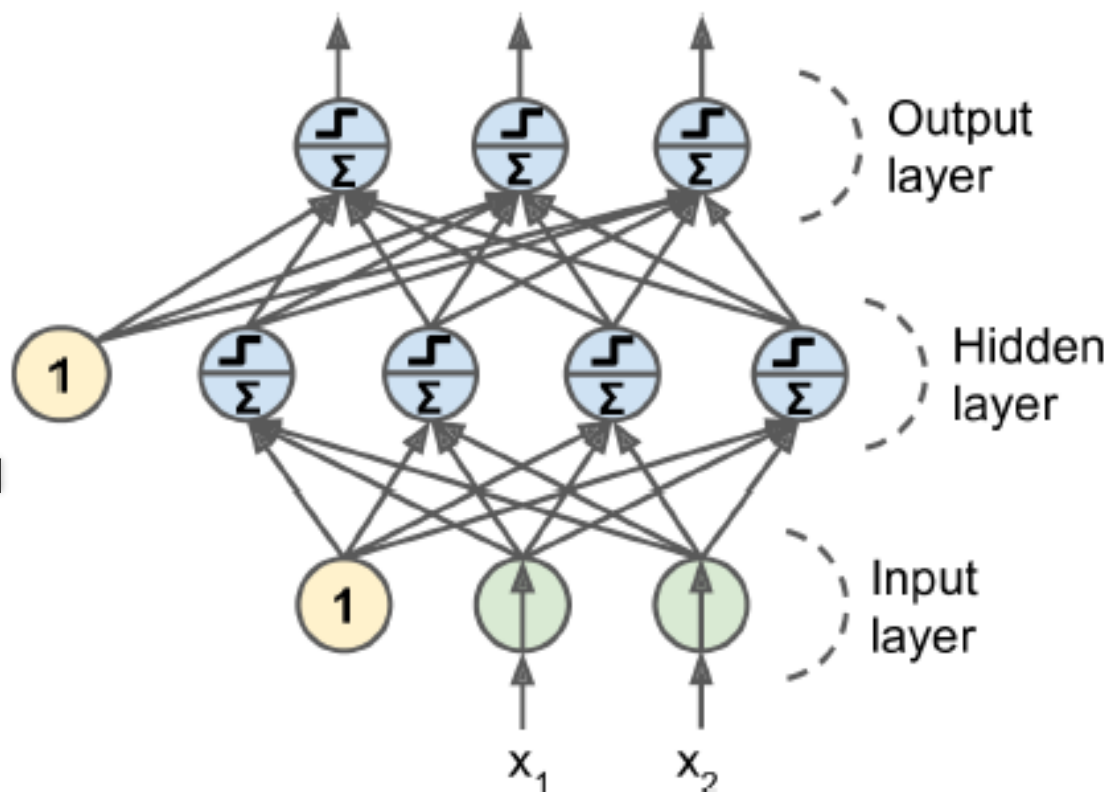


多层神经元

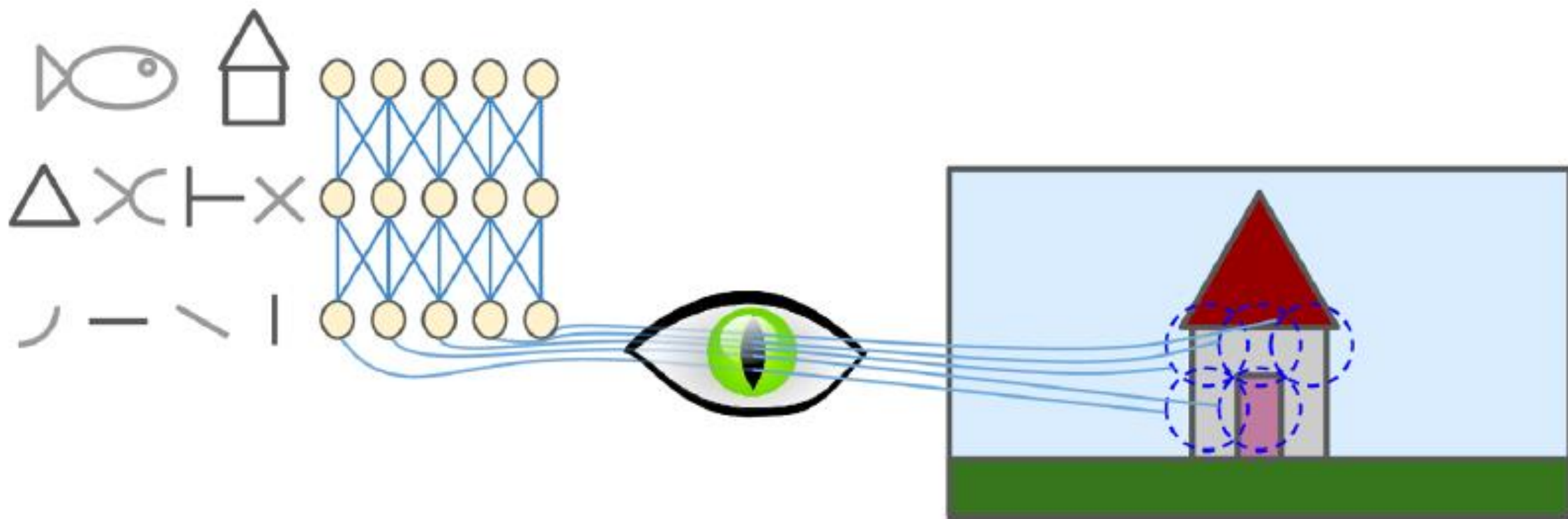


Multi-Layer Perceptron

- 多层感知机
- 第一层输入层，最后输出层，中间隐藏层
- 除了输出层，每层神经元包括bias都是全连接到下一层
- ANN人工神经网络有两个或两个以上隐藏层，称为DNN
- 深度神经网络



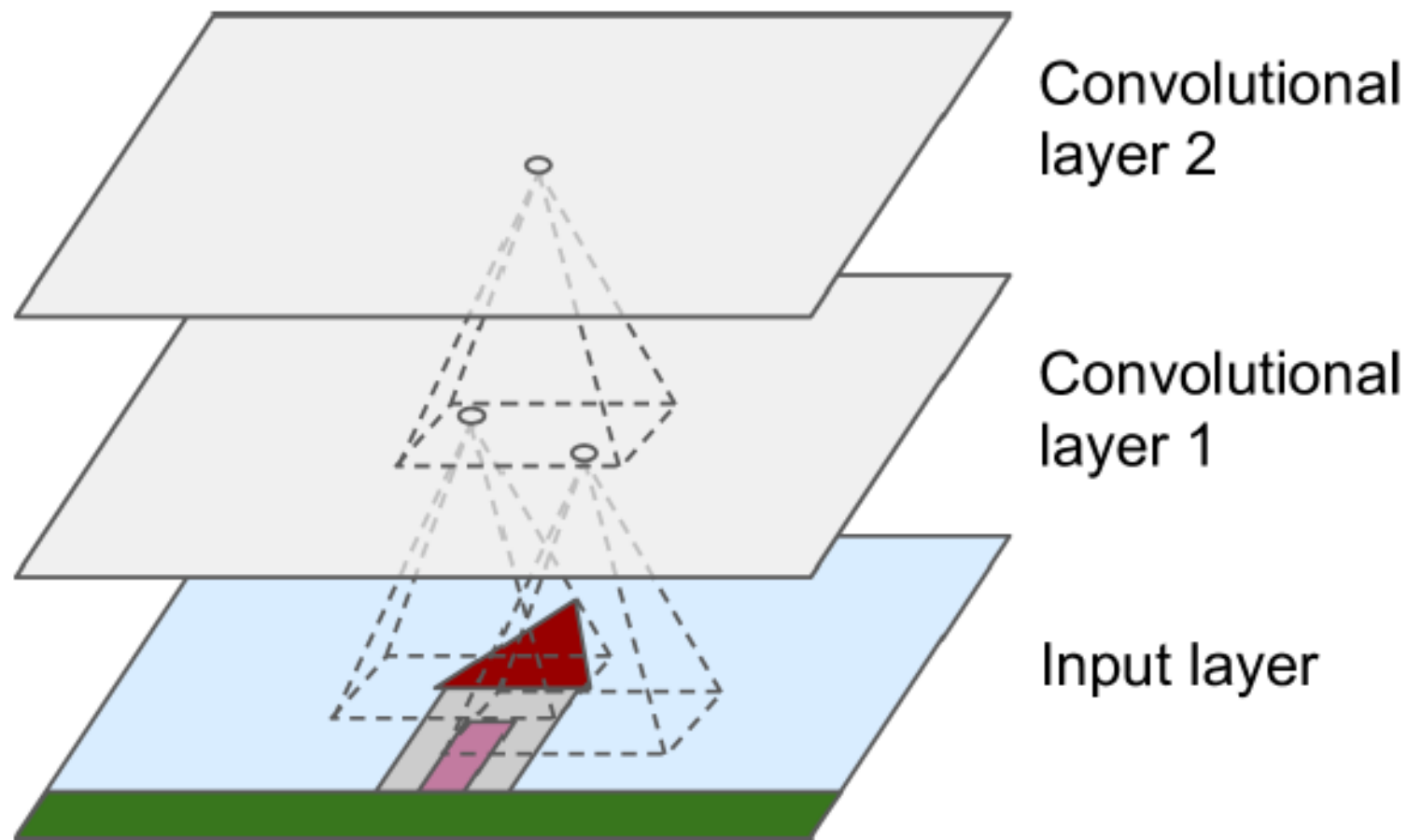
感受野



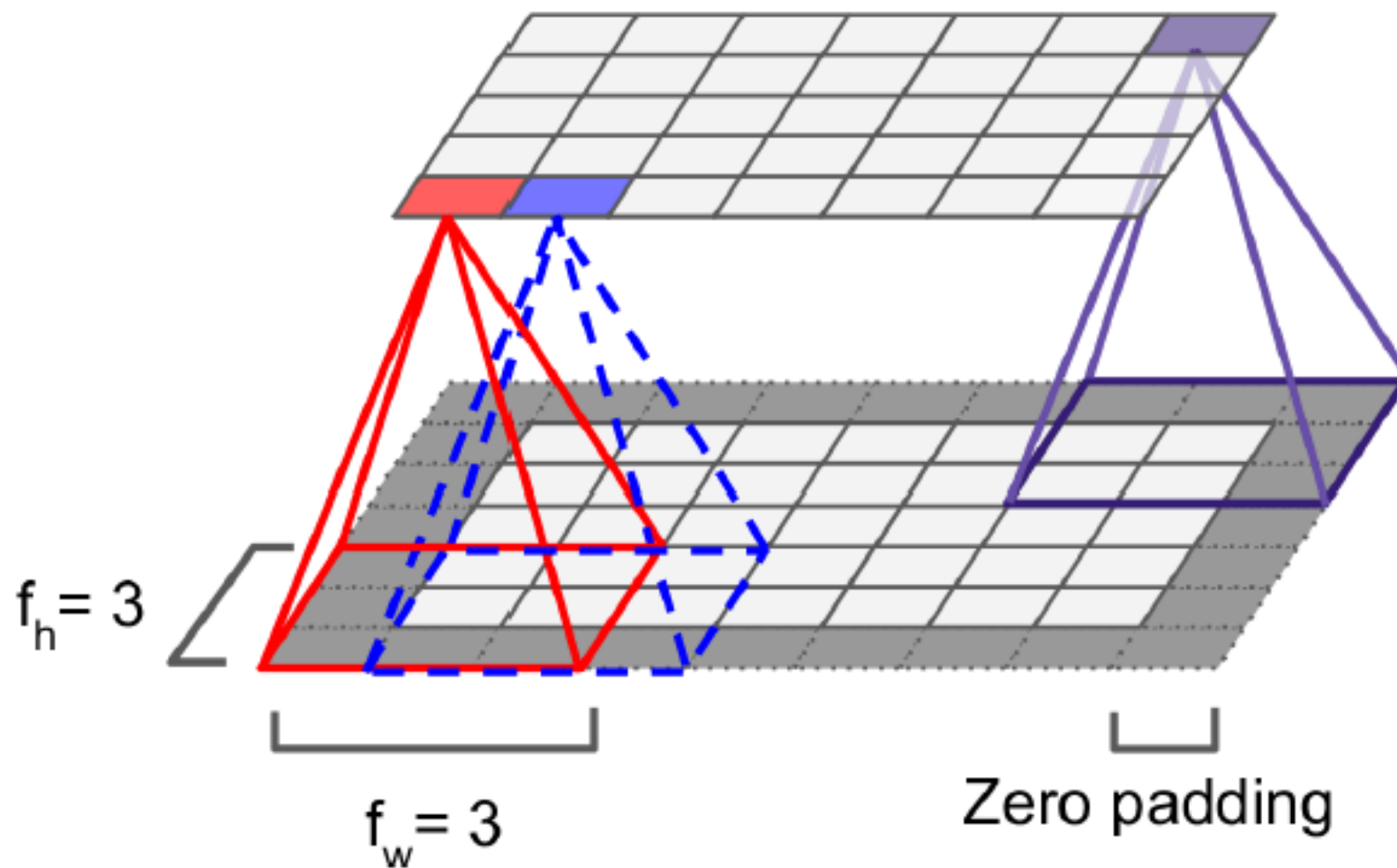
卷积层

- CNN里面最重要的构建单元就是卷积层
- 神经元在第一个卷积层不是连接输入图片的每一个像素，只是连接它们感受野的像素，以此类推，第二个卷积层的每一个神经元仅连接位于第一个卷积层的一个小方块的神经元
- 以前我们做MNIST的时候，把图像变成1D的，现在直接用2D

卷积层



Zero Padding



卷积的计算

- 假设有一个 5×5 的图像，使用一个 3×3 的filter进行卷积，想得到一个 3×3 的Feature Map

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

image 5×5

1	0	1
0	1	0
1	0	1

bias=0

filter 3×3

4		

feature map 3×3

卷积的计算

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

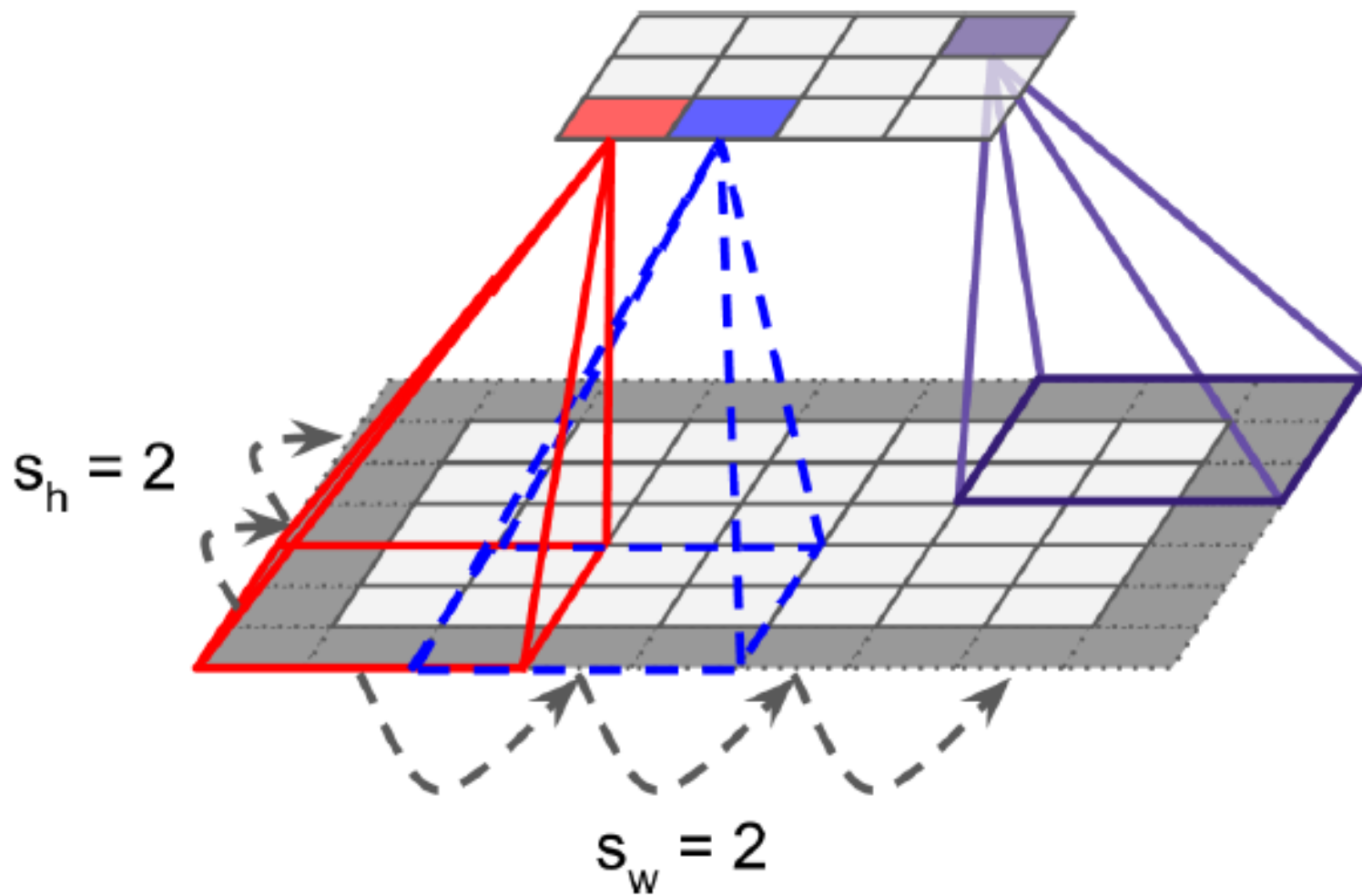
Image

4		

Convolved
Feature

stride

○ 步伐



Stride等于2

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

image 5*5

1	0	1
0	1	0
1	0	1

bias=0

filter 3*3

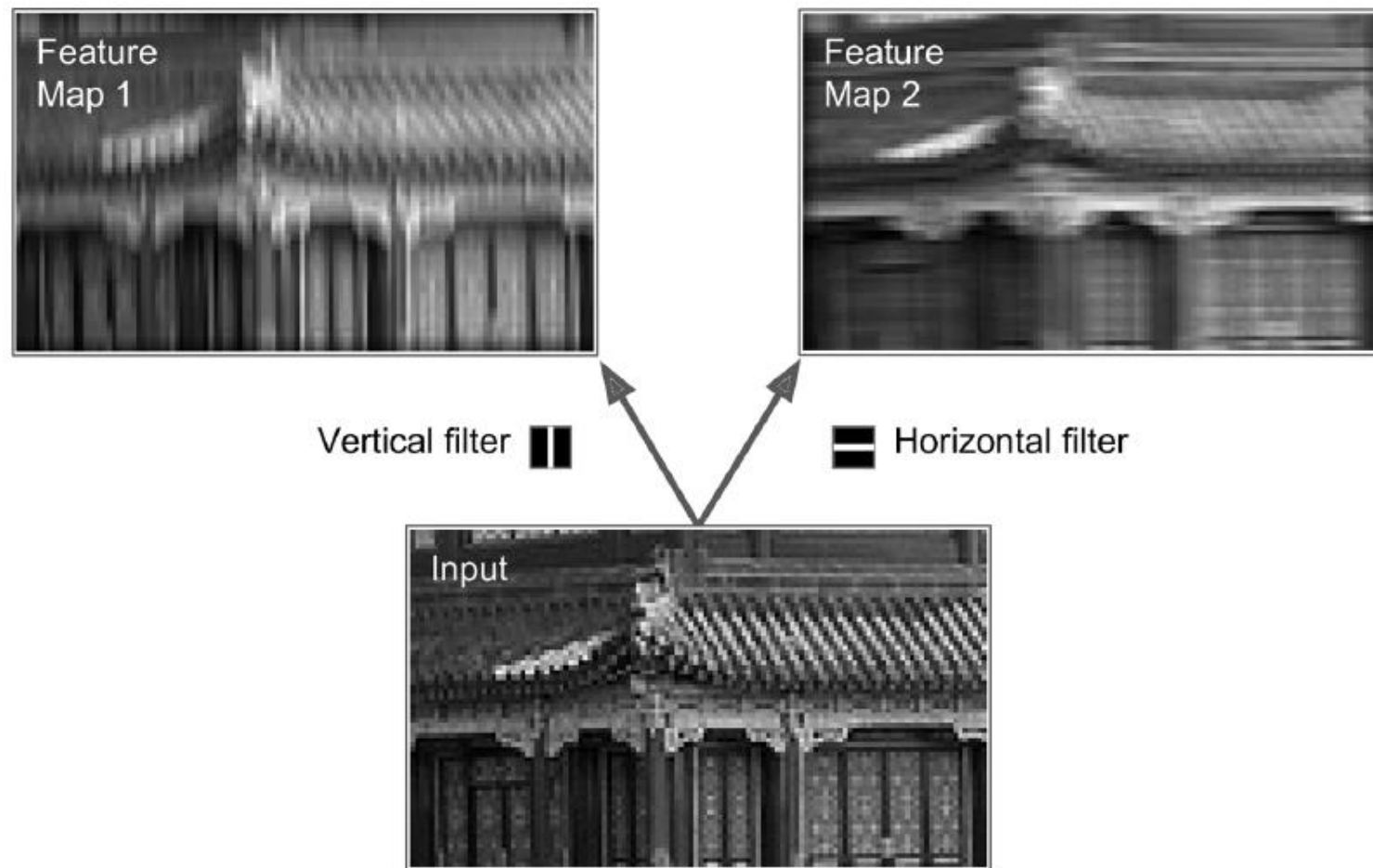
4	4
2	4

feature map 2*2

Filter卷积核

- Convolution kernels
- Vertical line filter 中间列为1，周围列为0
- Horizontal line filter 中间行为1，周围行为0
- 7*7 matrix

Feature Map

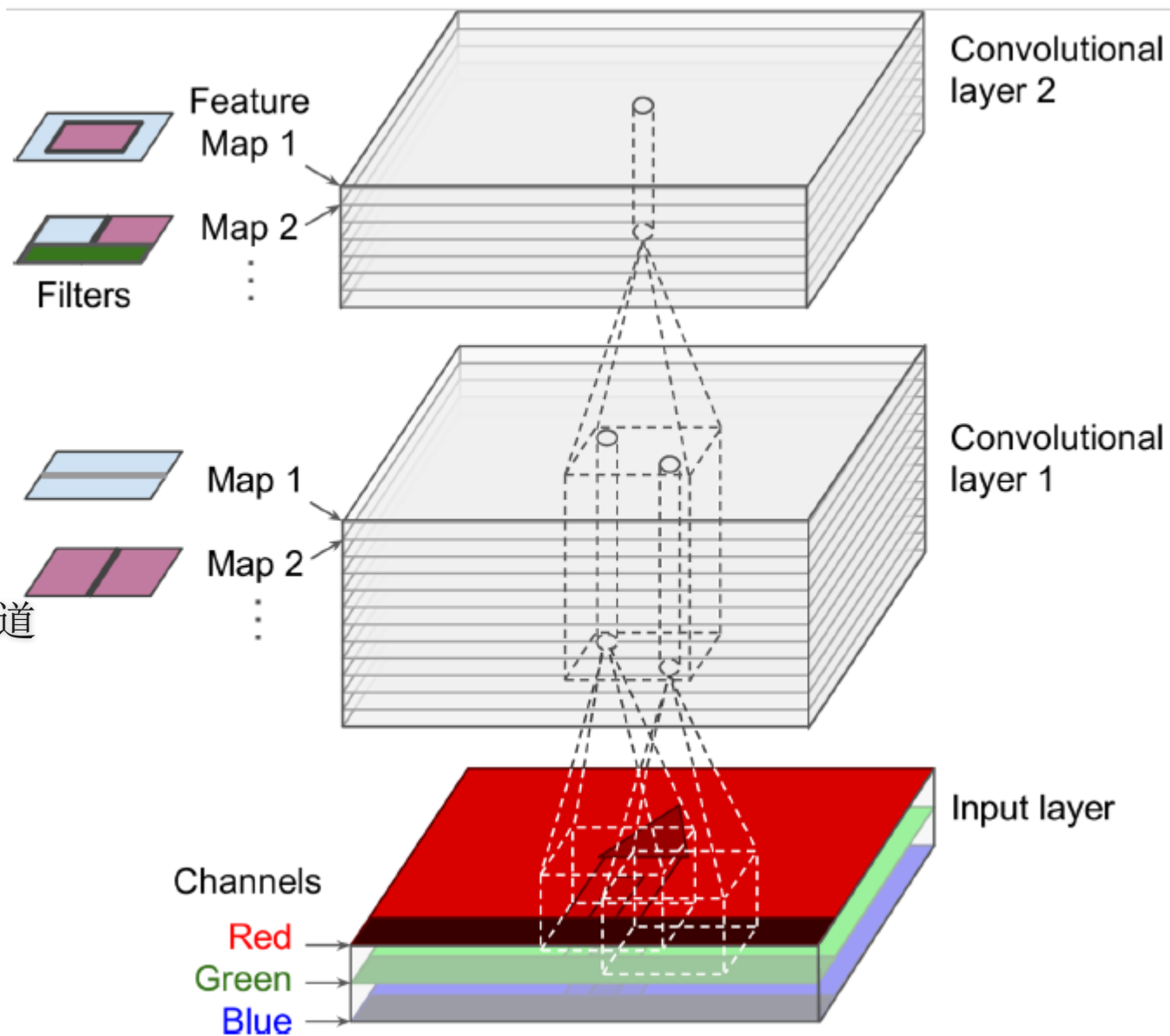


3D

- 在一个特征图里面，所有的神经元共享一样的参数（weights bias），权值共享
- 不同的特征图有不同的参数

3D

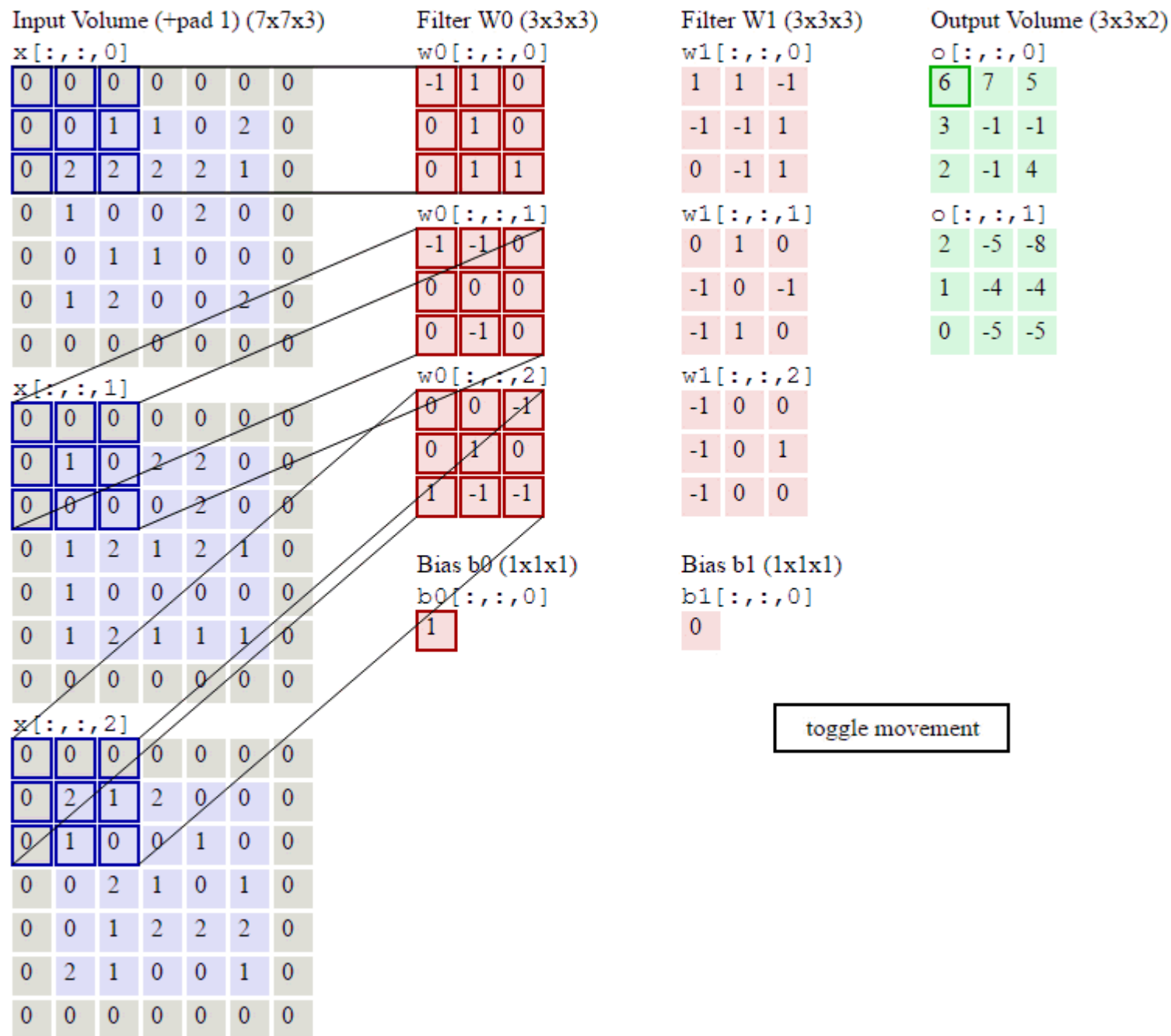
- RGB
- 灰度图一个通道
- 卫星图像



3D

$$a_{i,j} = f(\sum_{d=0}^{D-1} \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} w_{d,m,n} x_{d,i+m,j+n} + w_b)$$

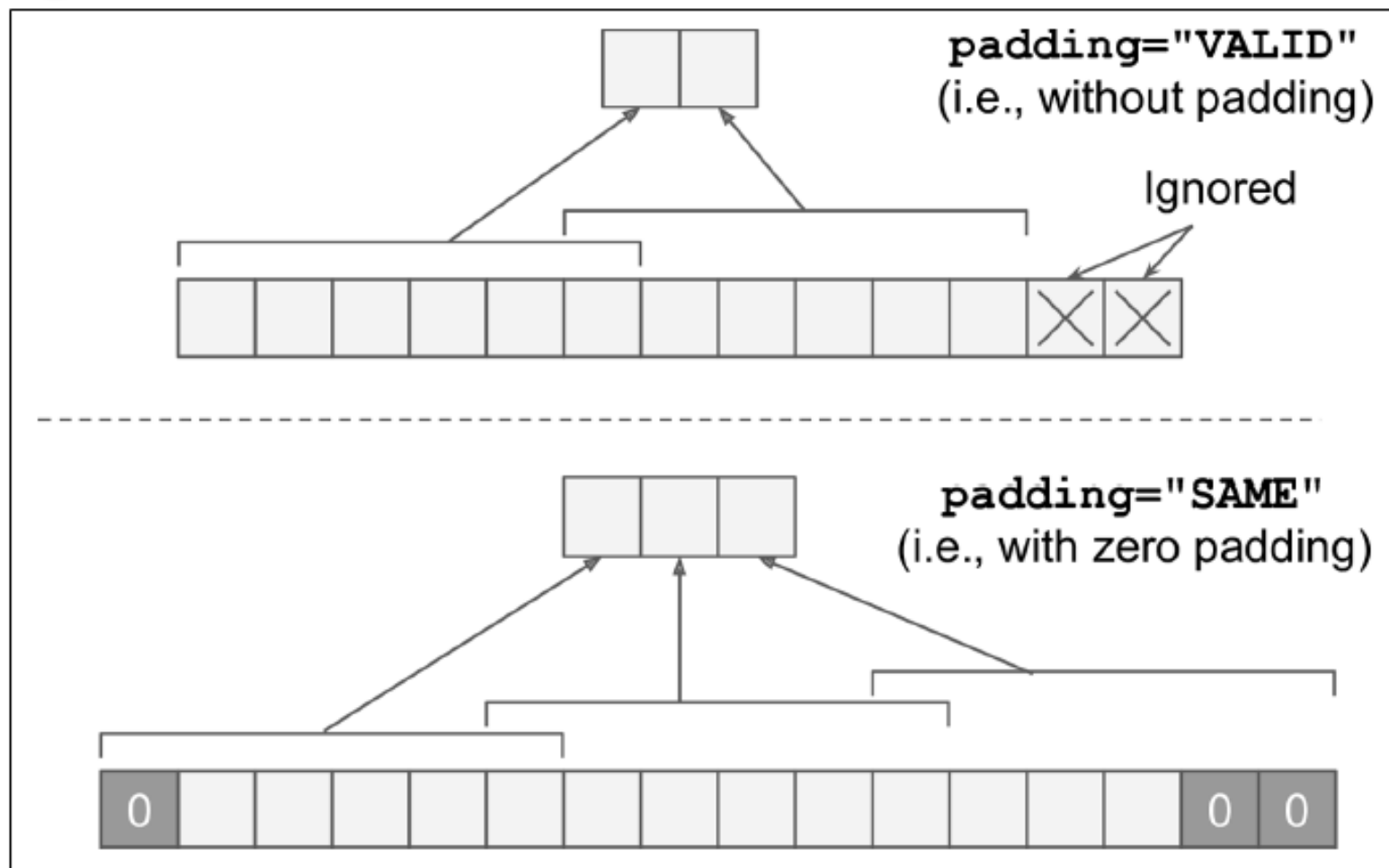
- D是深度；F是filter的大小
- (宽度或高度，两者相同)；
- $w_{d,m,n}$ 表示filter的第m行第n列权重；
- $a_{d,i,j}$ 表示图像的第d层第i行第j列像素；



Padding模式

- VALID
- 不适用zero padding，有可能会忽略图片右侧或底下，这个得看stride的设置
- SAME
- 必要会加zero padding，这种情况下，输出神经元个数等于输入神经元个数除以步长 $\text{ceil}(13/5)=3$

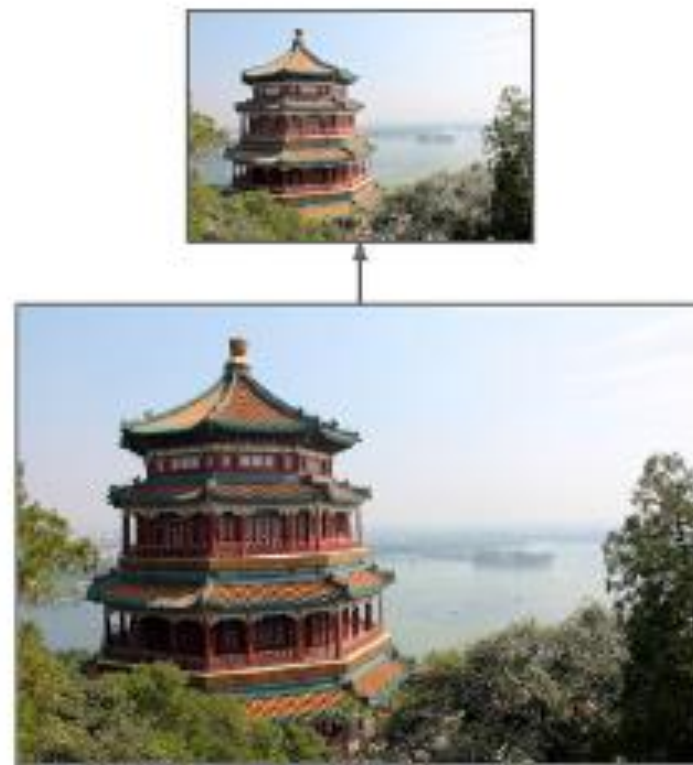
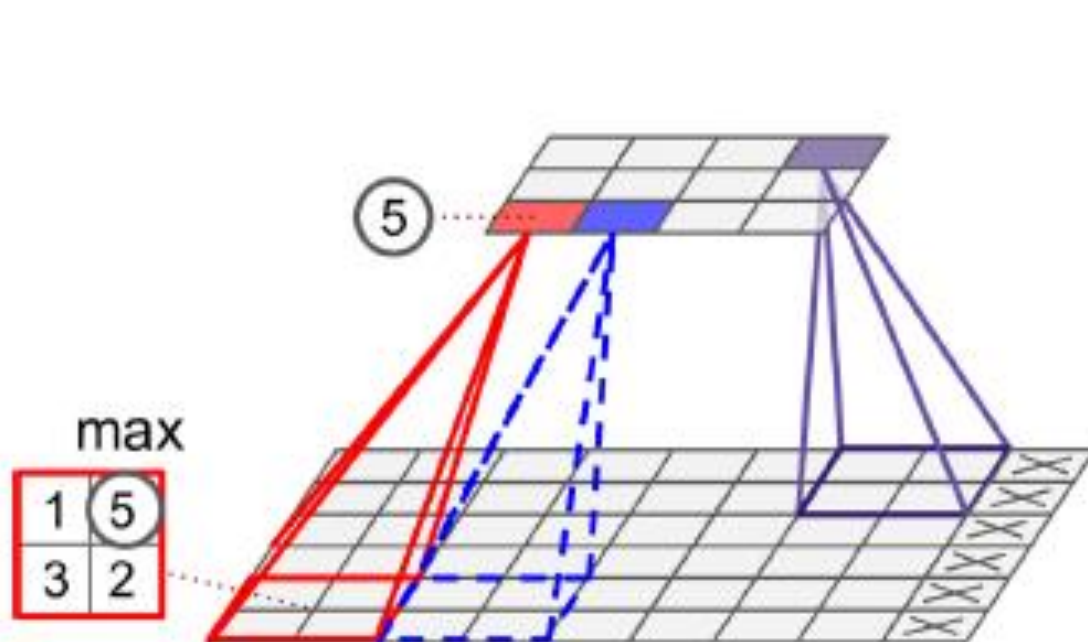
Padding模式



池化Pooling

- 目标就是降采样subsample, shrink, 减少计算负荷, 内存使用, 参数数量 (也可防止过拟合)
- 减少输入图片大小也使得神经网络可以经受一点图片平移, 不受位置的影响
- 正如卷积神经网络一样, 在池化层中的每个神经元被连接到上面一层输出的神经元, 只对应一小块感受野的区域。我们必须定义大小, 步长, padding类型
- 池化神经元没有权重值, 它只是聚合输入根据取最大或者是求均值
- 2*2的池化核, 步长为2, 没有填充, 只有最大值往下传递, 其他输入被丢弃掉了

池化Pooling



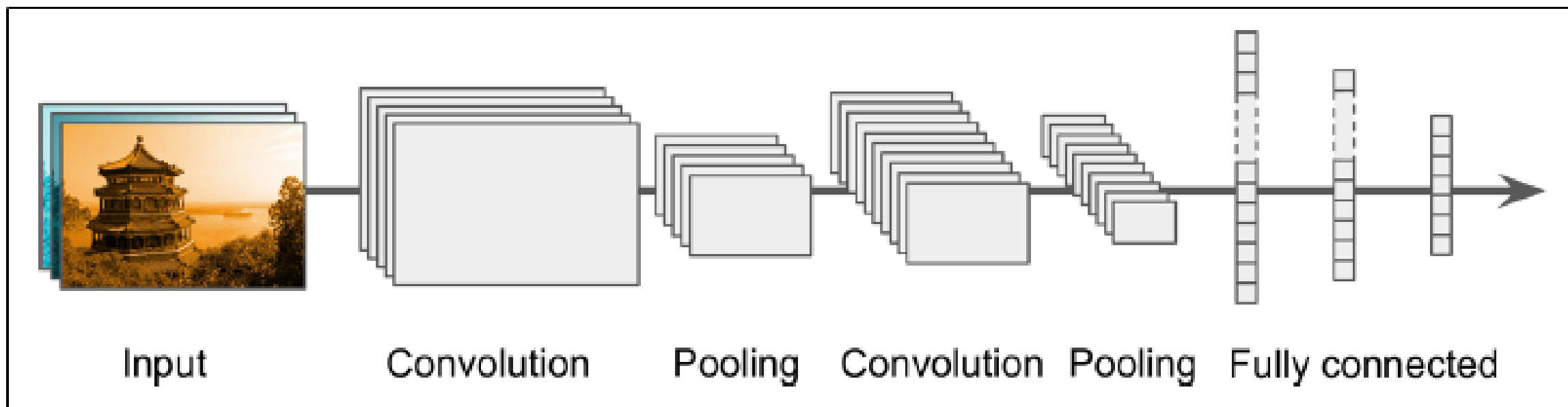
池化Pooling

- 长和宽两倍小，面积4倍小，丢掉75%的输入值
- 一般情况下，池化层工作于每一个独立的输入通道，所以输出的深度和输入的深度相同

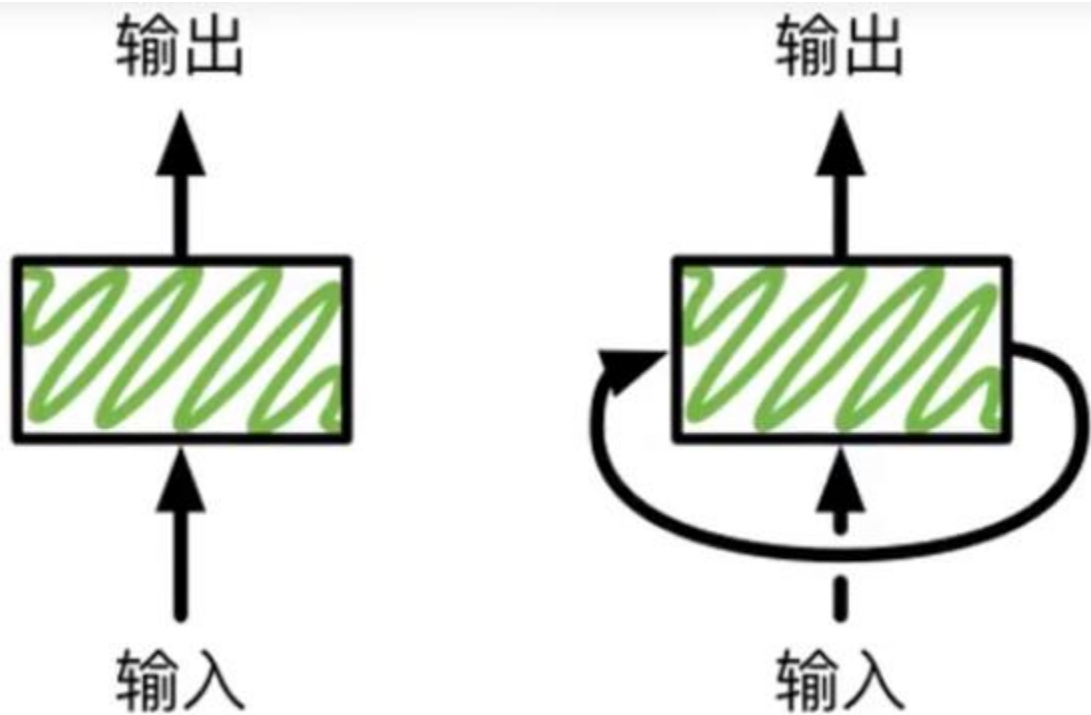
CNN架构

- 典型的CNN架构堆列一些卷积层
- 一般一个卷积层后跟ReLU层，然后是一个池化层，然后另一些个卷积层+ReLU层，然后另一个池化层，通过网络传递的图片越来越小，但是也越来越深，例如更多的特征图！
- 最后常规的前向反馈神经网络被添加，由一些全连接的层+ReLU层组成，最后是输出层预测，例如一个softmax层输出预测的类概率
- 一个常见的误区是使用卷积核过大，你可以使用和 9×9 的核同样效果的两个 3×3 的核，好处是会有更少的参数需要被计算，还可以在中间多加一个非线性激活函数ReLU，来提供复杂程度

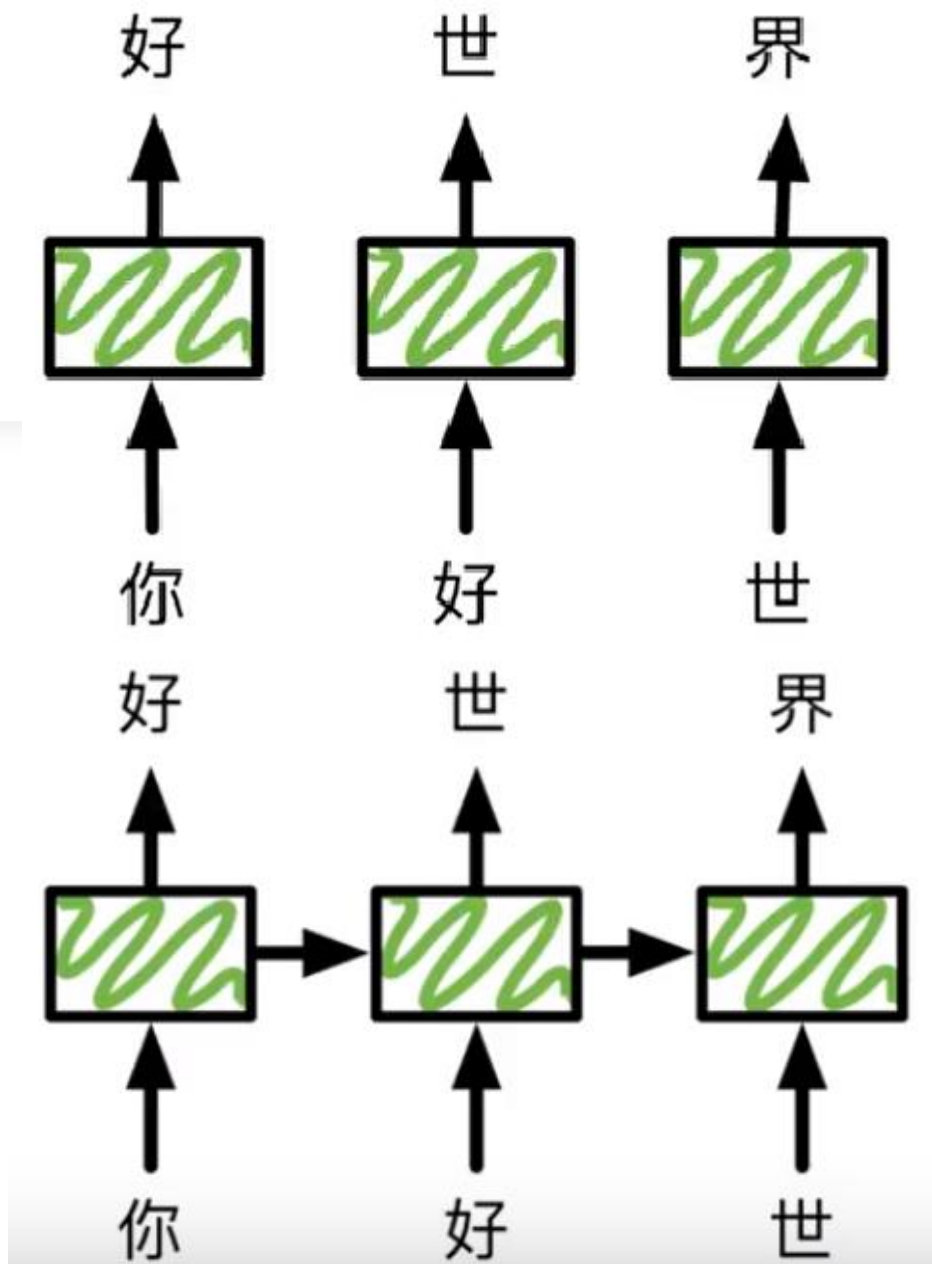
CNN架构



RNN架构



隐藏状态



谢谢大家

- 群号 644090648
- SXT人工智能交流群5



SXT人工智能交流群5

扫一扫二维码，加入群聊。

尚学堂·实战程序员