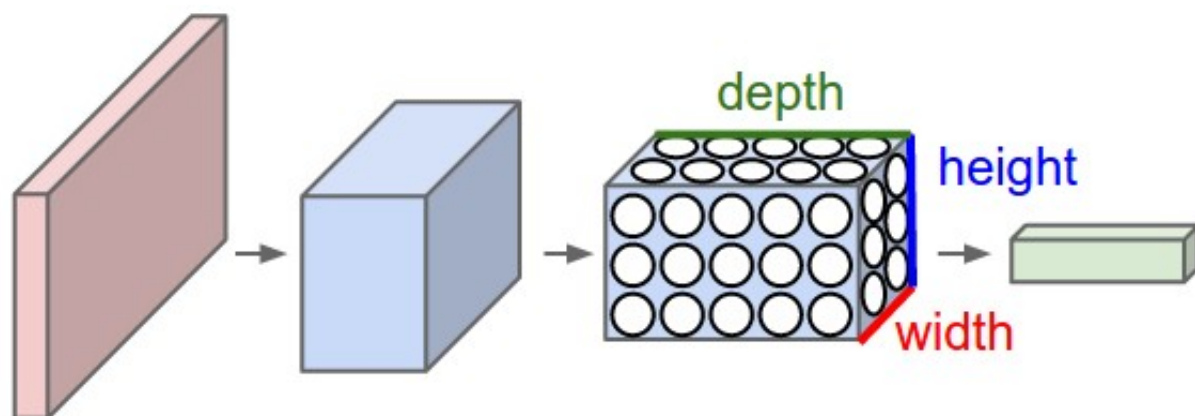
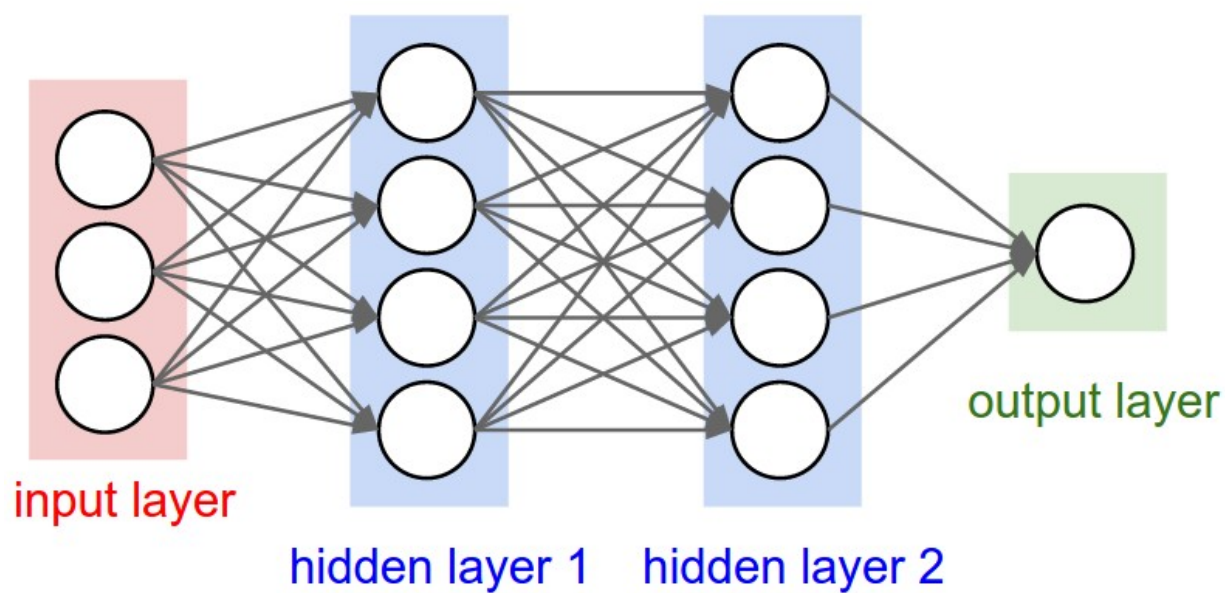


# 神经网络



神经网络的基本结构

## 卷积神经网络

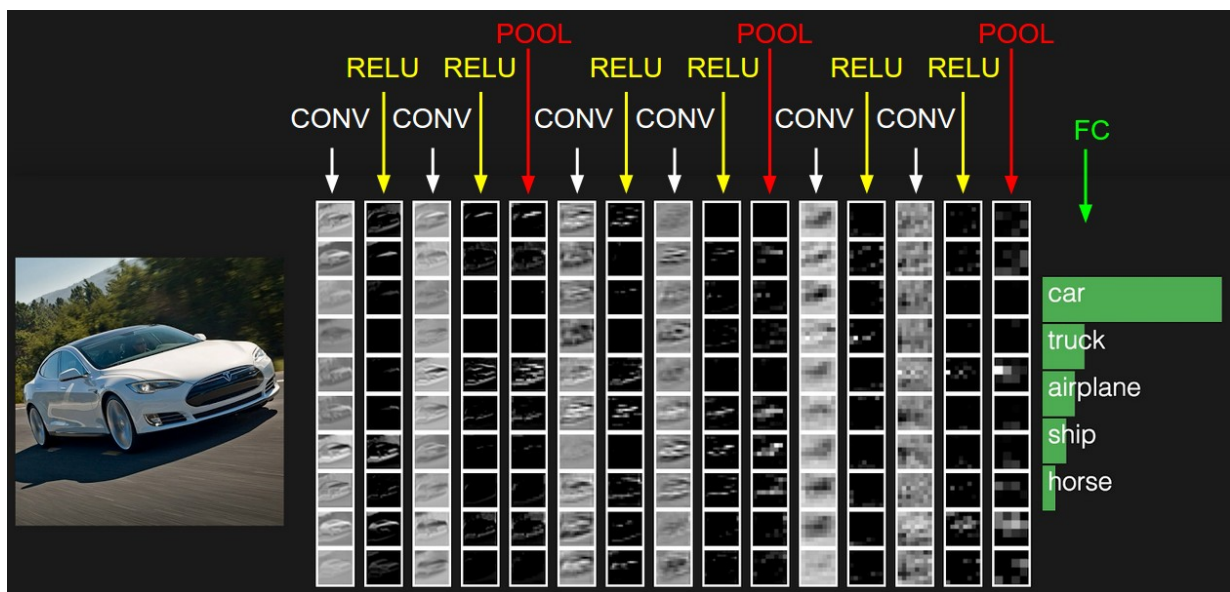
在卷积神经网络中，有3种最主要的层：

卷积运算层

pooling层（池化层）

全连接层

一个完整的神经网络就是由这三种层叠加组成的。



卷积神经网络的层级结构：

1. 数据输入层 / Input Layer
2. 卷积计算层 / CONV layer
3. ReLU 激励层 / Relu Layer
4. 池化层 / Pooling Layer
5. 全连接层 / FC layer

## 数据输入层

(有三种常见的数据处理方式)

### 1. 去均值

把输入数据各个维度都中心化到0

(也就是首先对输入的所有数据求一个平均值，然后将每一个输入数据减去均值后再作为输入数据输入带神经网络中)

### 1. 归一化

幅度归一化到同样的范围

(输入的数据可能会有多维，可能每一维度的范围幅度是不一致的，需要将其划归为一定的范围内。)

### 1. PCA/白话

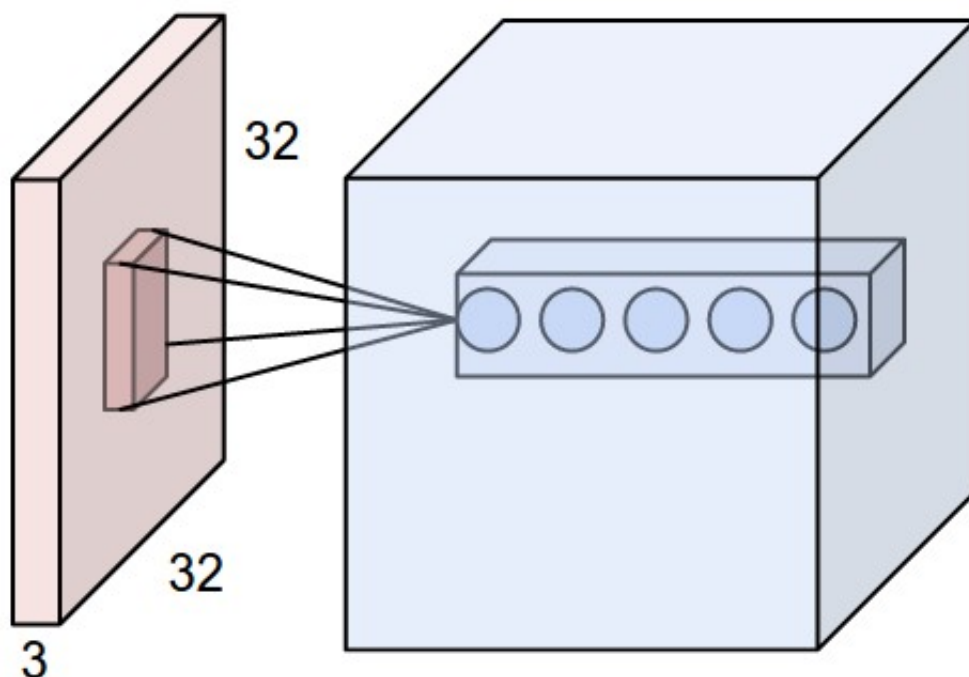
用PCA降为  
白化是对数据每一个特征轴上的幅度归一化

## 卷积计算层（最重要的一层）

在神经网络中，每一个神经元 与前一层的所有节点都有关联。

卷积神经网络中每一层的神经元只会和上一层的一些局部区域相连，这就是所谓的局部连接性，（局部关联。每个神经元看作一个Filter）

窗口滑动（receptive Field）滑动，filter对局部数据计算



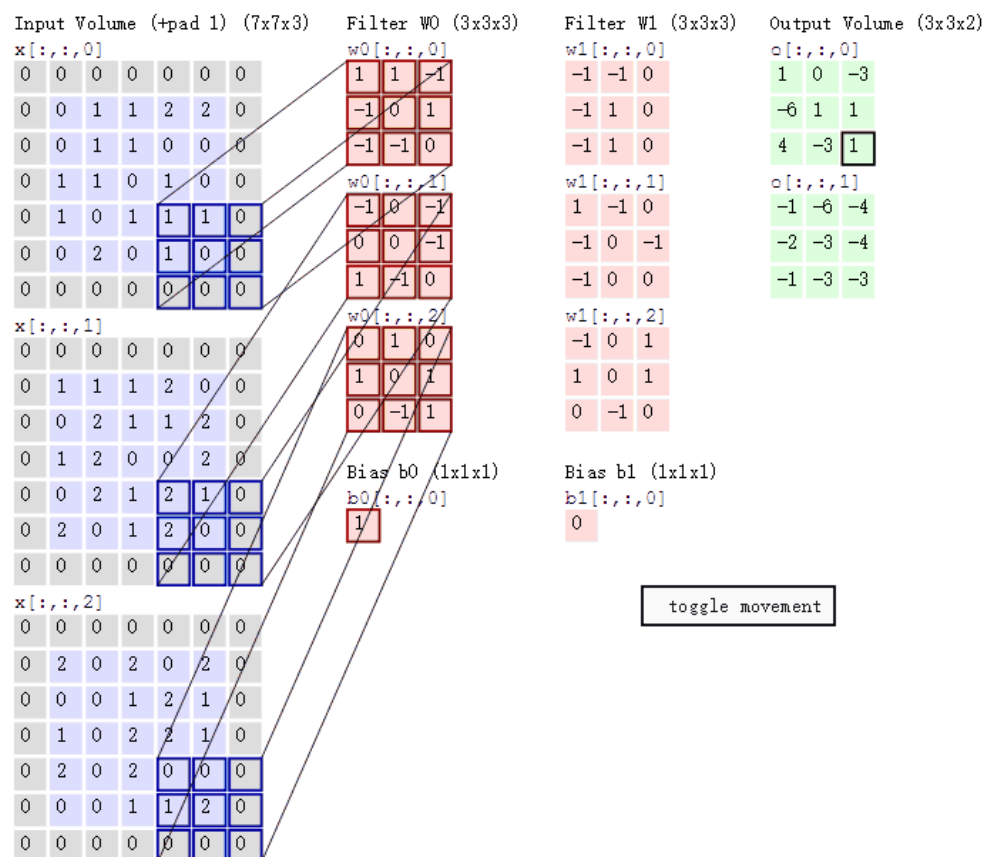
从输入的数据到输出数据，有三个超参数会决定输出数据的维度，分别是 **深度/depth**，**步长/stride** 和 **填充值/zero-padding**：

三个基本概念：

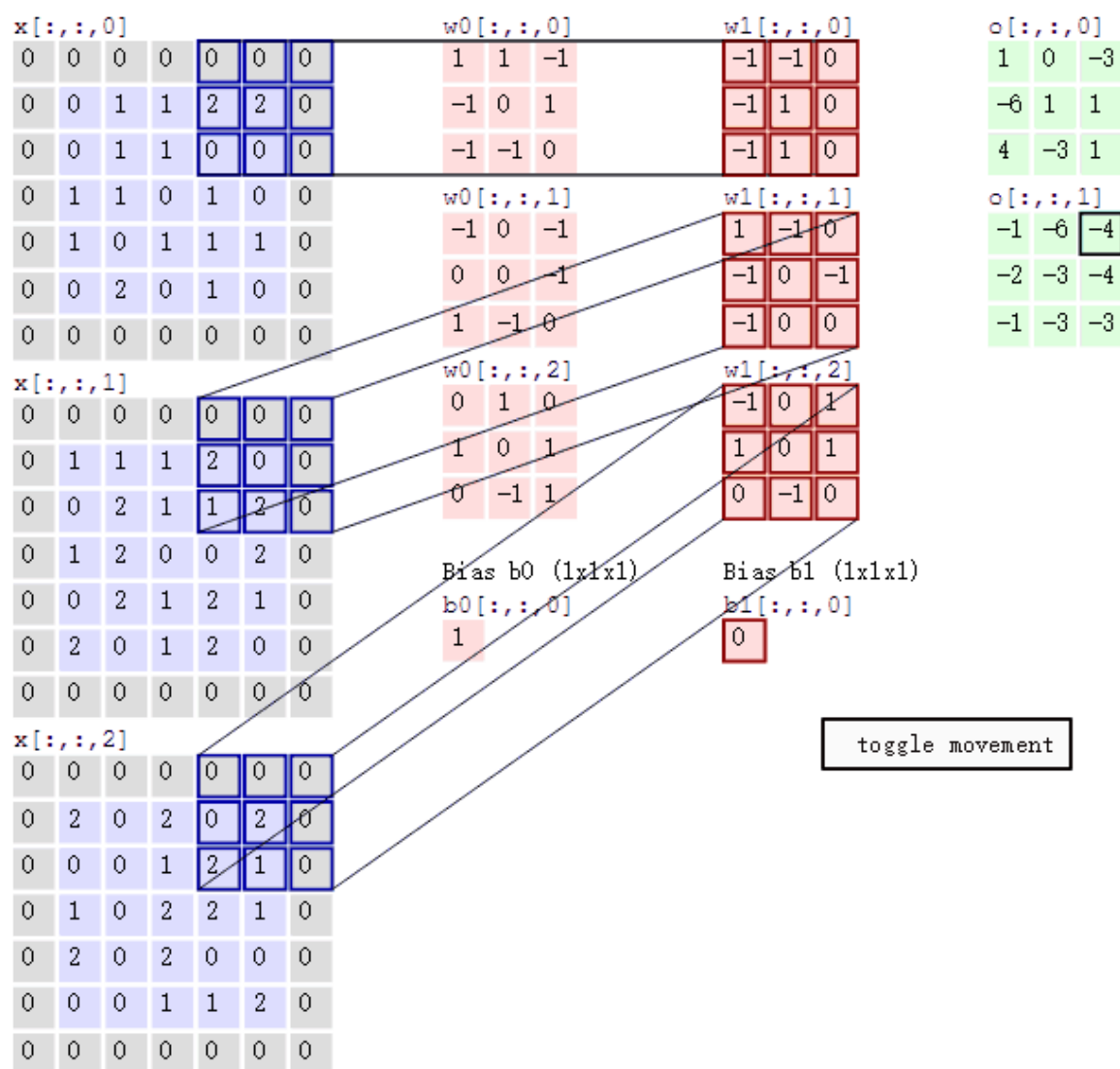
所谓**深度/depth**，简单说来指的就是卷积层中和上一层同一个输入区域连接的神经元个数。这部分神经元会在遇到输入中的不同feature时呈现activate状态，举个例子，如果这是第一个卷积层，那输入到它的数据实际上是像素值，不同的神经元可能对图像的边缘。轮廓或者颜色会敏感。

所谓**步长/stride**，是指的窗口从当前位置到下一个位置，『跳过』的中间数据个数。比如从图像数据层输入到卷积层的情况下，也许窗口初始位置在第1个像素，第二个位置在第5个像素，那么 $\text{stride}=5-1=4$ 。

所谓**zero-padding**是在原始数据的周边补上0值的圈数。（下面第2张图中的样子）



上图中可以看出，depth=2，表示有两个filter



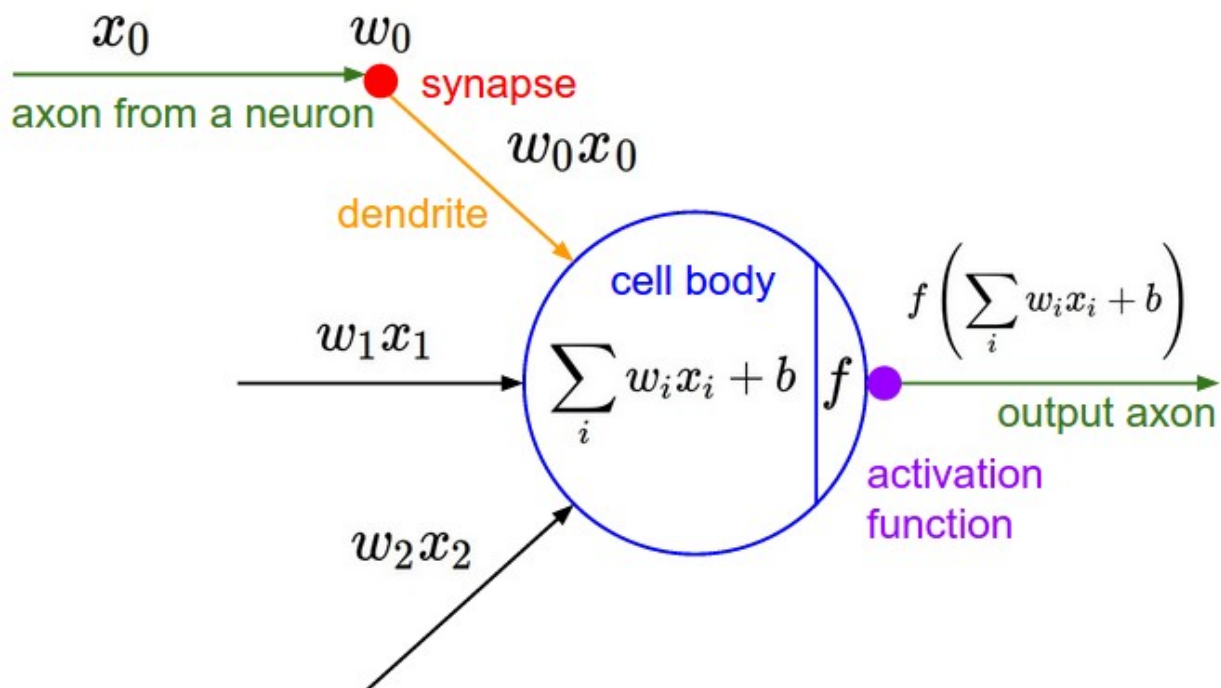
## 参数共享机制

假设每一个神经元连接数据窗的权重是固定的，可以将每一个神经元看作模板，（也就是说每一个神经元值关注输入数据的某一个特征）

1. 需要估算的权重个数减少  
一组固定权重和不同窗口内数据做内积：**卷积**

## 激励层

（把卷积层输出结果做非线性映射）

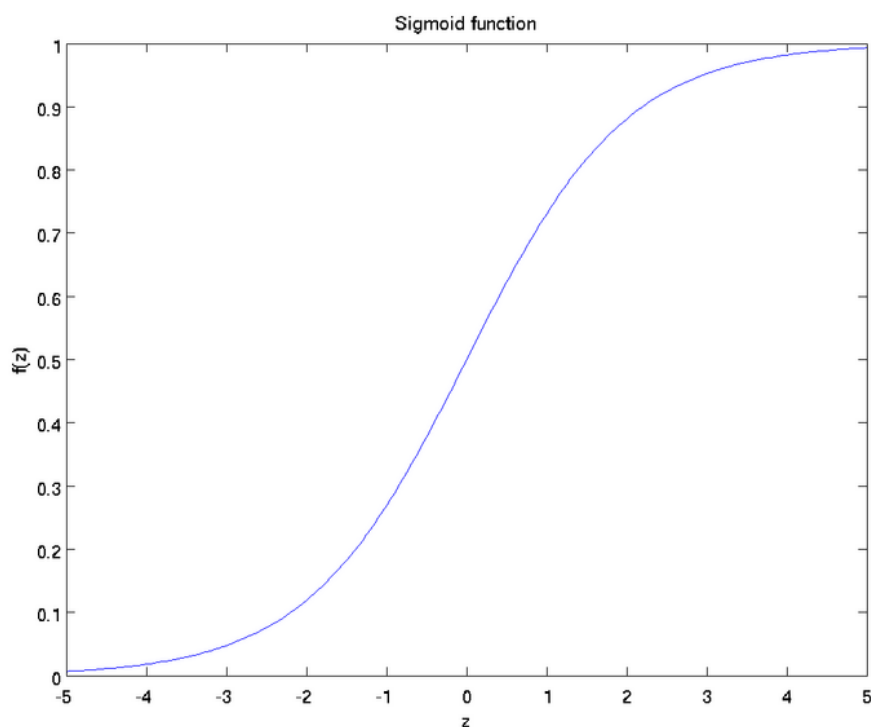


把卷积层输出结果做非线性映射

### Sigmoid

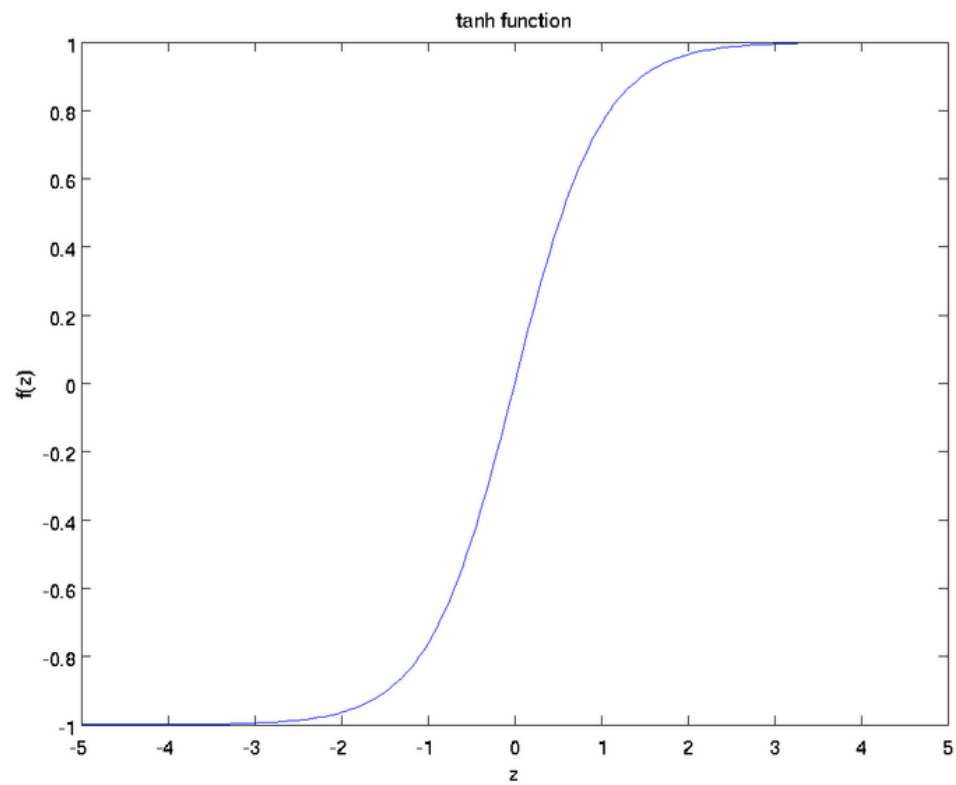
存在的问题：

1. 因为在实际计算的过程中是应用导数来求解的，但是这个函数中当 $x$ 值比较大的时候会比较趋于零
2. 训练过程中没有残差传回来，导致有问题



## Tanh

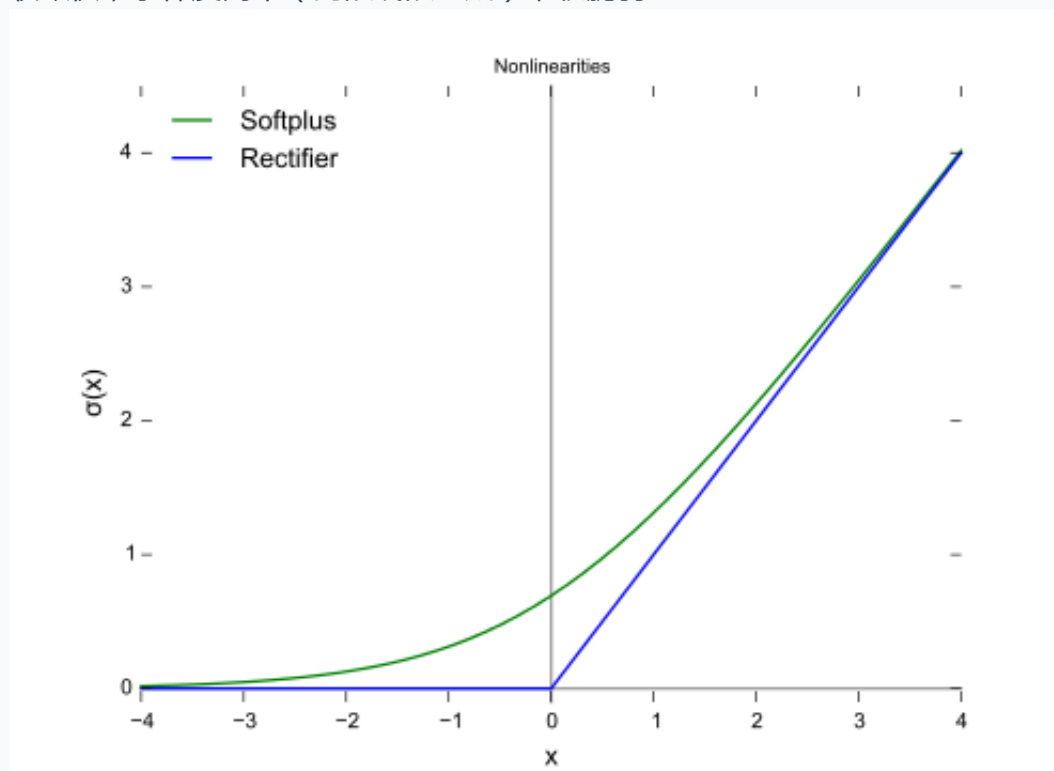
也同样存在sigmoid函数的缺陷



## ReLU

ReLU ( The Rectified Linear Unit/ 修正线性单元 )

收敛快，求梯度简单（两段线段组成），较脆弱

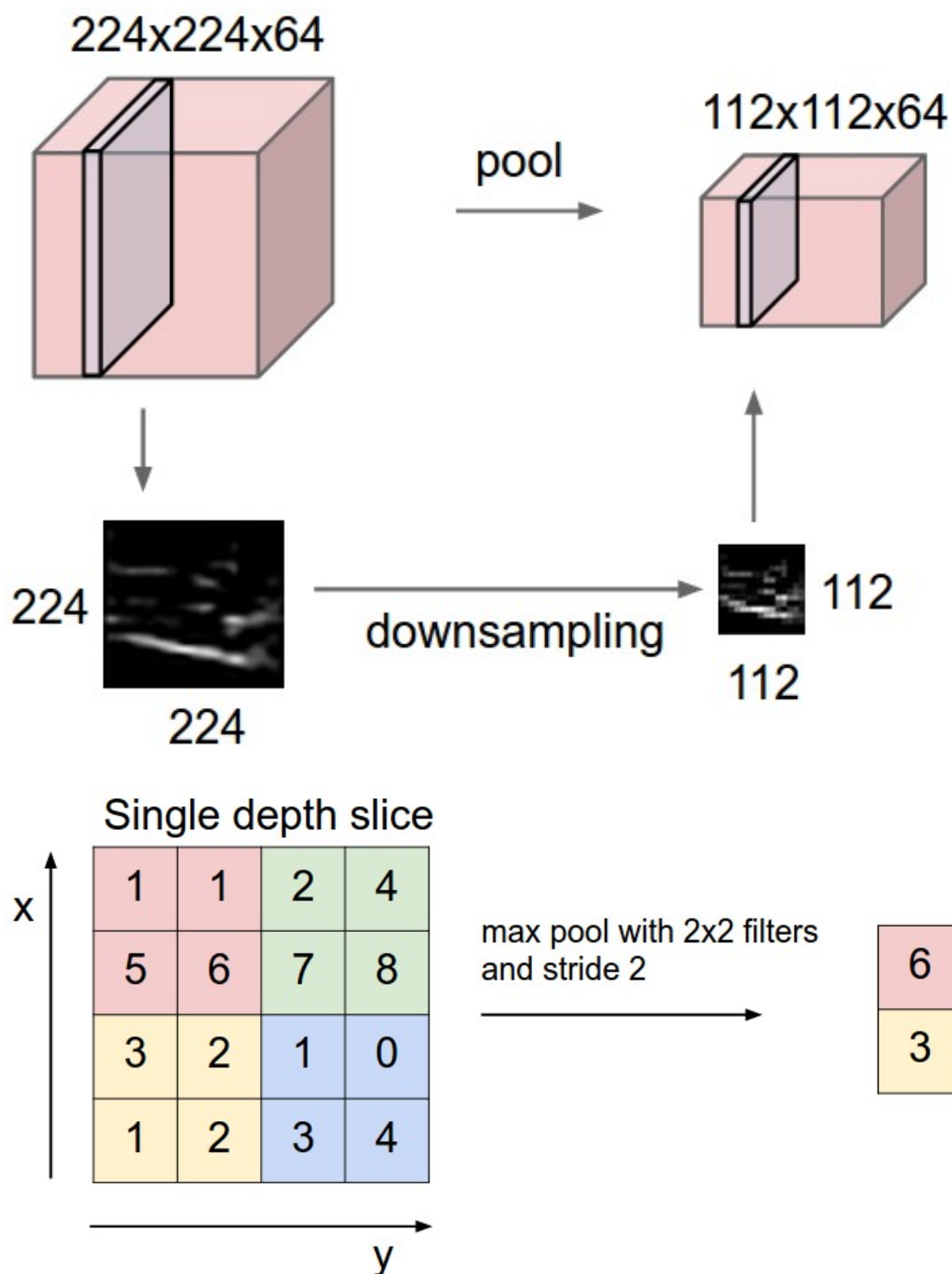


这个函数如果大于0则输出他本身，如果是小于0则全部归为0

## POOL（池化）层

Pooling层是夹在连续的卷积层中间的层。它的作用也非常简单，就是**逐步地压缩/减少数据和参数的量，也在一定程度上减小过拟合的现象**。Pooling层做的操作也非常简单，就是将原数据上的区域压缩成一个值(区域最大值/MAX或者平均值/AVERAGE)，最常见的Pooling设定是，将原数据切成2\*2的小块，每块里面取最大值作为输出，这样我们就自然而然减少了75%的数据量。需要提到的是，除掉MAX和AVERAGE的Pooling方式，其实我们也可以设定别的pooling方式，比如L2范数pooling。说起来，历史上average pooling用的非常多，但是近些年热度降了不少，工程师们在实践中发现max pooling的效果相对好一些。





## 全连接层/FC layer

两层之间所有神经元都有权重

通常全连接层在卷积神经网络尾部

## 一般CNN结构依次为

```

INPUT
[[CONV -> RELU]*N -> POOL?]*M
[FC -> RELU]*K
FC
  
```

## 卷积神经网络训练算法

同一般机器学习算法。先定义Loss function，衡量和实际结果之间差距  
找到最小化损失函数的W和b，CNN 中用书的算法是SGD。

SGD需要计算W和b的偏导。

BP 算法就是计算偏导用的。

BP算法的核心是求导链式法则。

**BP算法利用链式求导法则，逐级相乘直到求解出dW和db。**

**利用SGD/随机梯度下降，迭代和更新W和b**

## 卷积神经网络

### 优点

共享卷积核，对高维数据处理无压力

无需手动选取特征，训练好权重，即得特征

分类效果好

### 缺点

需要调参，需要大量样本，训练最好要GPU

物理含义不明确

## 典型神经网络

1. LeNet，这是最早用起来的卷积神经网络，Yann LeCun在论文LeNet提到。
2. AlexNet，2012 ILSVRC比赛远超第2名的卷积神经网络，和LeNet的结构比较像，只是更深，同时用多层小卷积层叠加提到大卷积层。
3. ZF Net，2013 ILSVRC比赛冠军，可以参考论文ZF Net
4. GoogLeNet，2014 ILSVRC比赛冠军，Google发表的论文Going Deeper with Convolutions有具体介绍。
5. VGGNet，也是2014 ILSVRC比赛中的模型，有意思的是，即使这个模型当时在分类问题上的效果，略差于google的GoogLeNet，但是在很多图像转化学习问题(比如object detection)上效果奇好，它也证明卷积神经网络的『深度』对于最后的效果有至关重要的作用。预训练好的模型在pretrained model site可以下载。