



多媒体技术

回顾

- 图像分割
 - 基于边缘的分割
 - 相关和卷积
 - 算子
 - 基于区域的分割
 - 种子填充算法
- 卷积神经网络
 - 神经网络
 - 线性变换、位移、非线性变换
 - 训练：学习得到 W 等参数
 - 目标函数缩小：梯度下降法

2.3.3 视觉媒体数字化

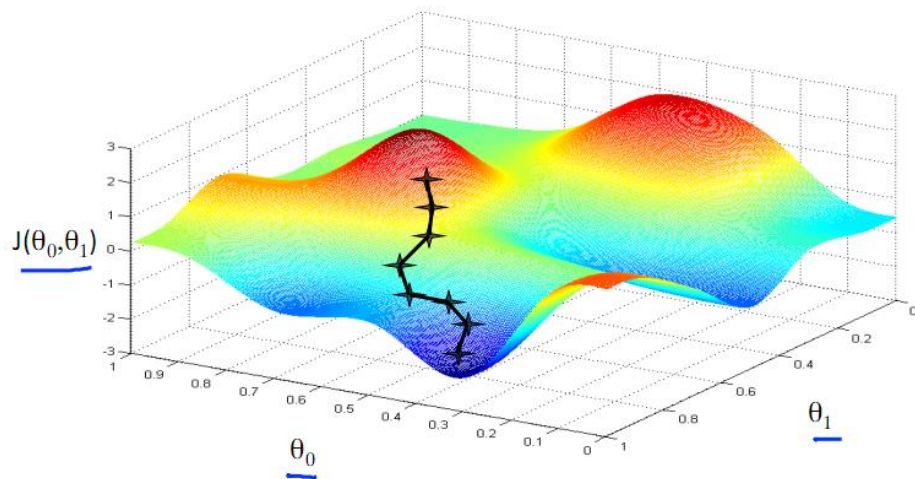
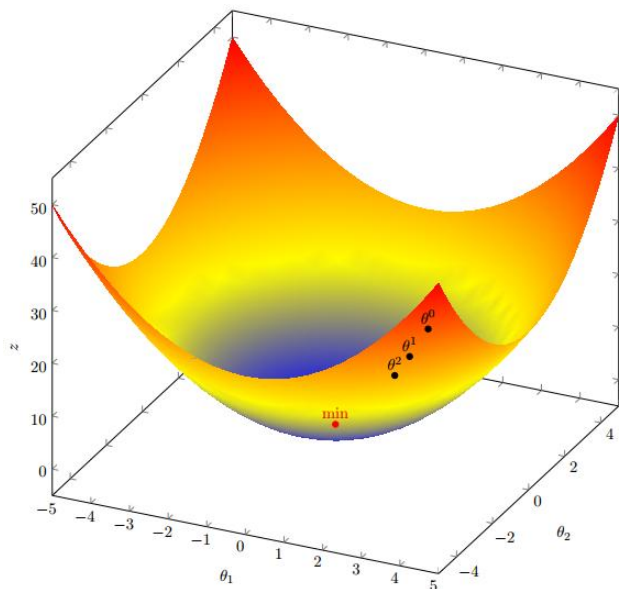
- 新技术：卷积神经网络

- 训练神经网络

- 如何学习每一层的权重矩阵 W

- 损失函数或目标函数

- 尽可能使损失函数(loss function)缩小：梯度下降法



2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 梯度下降法

- 方向导数

定理 如果函数 $f(x, y)$ 在点 $P_0(x_0, y_0)$ 可微分, 那么函数在该点沿任一方向 l 的方向导数存在, 且有

$$\left. \frac{\partial f}{\partial l} \right|_{(x_0, y_0)} = f_x(x_0, y_0) \cos \alpha + f_y(x_0, y_0) \cos \beta,$$

其中 $\cos \alpha$ 和 $\cos \beta$ 是方向 l 的方向余弦.

函数在一点处沿**某一方向的方向导数**反映了函数沿该**方向的变化率**

在同一点的所有方向导数中, 是否有最大值? 怎样的方向?

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 梯度下降法

- 梯度：一个向量($f_x(x_0, y_0), f_y(x_0, y_0)$)

这向量称为函数 $f(x, y)$ 在点 $P_0(x_0, y_0)$ 的梯度，记作 $\mathbf{grad} f(x_0, y_0)$ 或 $\nabla f(x_0, y_0)$ ，即

$$\mathbf{grad} f(x_0, y_0) = \nabla f(x_0, y_0) = f_x(x_0, y_0)\mathbf{i} + f_y(x_0, y_0)\mathbf{j}.$$

$$\begin{aligned}\left. \frac{\partial f}{\partial l} \right|_{(x_0, y_0)} &= f_x(x_0, y_0) \cos \alpha + f_y(x_0, y_0) \cos \beta \\ &= \mathbf{grad} f(x_0, y_0) \cdot \mathbf{e}_l = |\mathbf{grad} f(x_0, y_0)| \cos \theta,\end{aligned}$$

其中 $\theta = (\mathbf{grad} f(x_0, y_0), \mathbf{e}_l)$. $\mathbf{e}_l = (\cos \alpha, \cos \beta)$

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 梯度下降法

- 梯度：一个向量($f_x(x_0, y_0), f_y(x_0, y_0)$)

(1) 当 $\theta = 0$, 即方向 e_l 与梯度 $\mathbf{grad} f(x_0, y_0)$ 的方向相同时, 函数 $f(x, y)$ 增加最快. 此时, 函数在这个方向的方向导数达到最大值, 这个最大值就是梯度 $\mathbf{grad} f(x_0, y_0)$ 的模, 即

$$\left. \frac{\partial f}{\partial l} \right|_{(x_0, y_0)} = |\mathbf{grad} f(x_0, y_0)|.$$

(2) 当 $\theta = \pi$, 即方向 e_l 与梯度 $\mathbf{grad} f(x_0, y_0)$ 的方向相反时, 函数 $f(x, y)$ 减少最快, 函数在这个方向的方向导数达到最小值, 即

$$\left. \frac{\partial f}{\partial l} \right|_{(x_0, y_0)} = -|\mathbf{grad} f(x_0, y_0)|.$$

负梯度的方向是函数值减少最快的方向

2.3.3 视觉媒体数字化

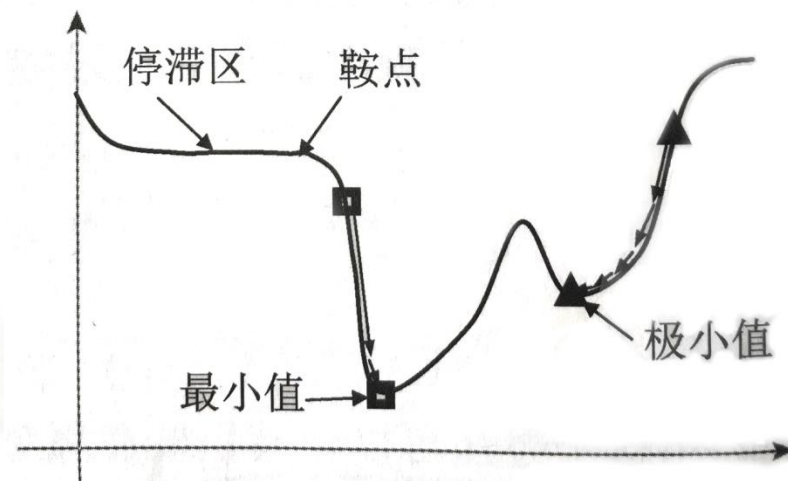
- 新技术：卷积神经网络
 - 梯度下降法

给定目标函数 $f(x)$ 和初始点 x_0
重复：

$$\Delta x_t = -\nabla f(x_t)$$

$$x_{t+1} = x_t + \eta \Delta x_t$$

停止，如果 $|\Delta x_t| < \varepsilon$



x 更新：

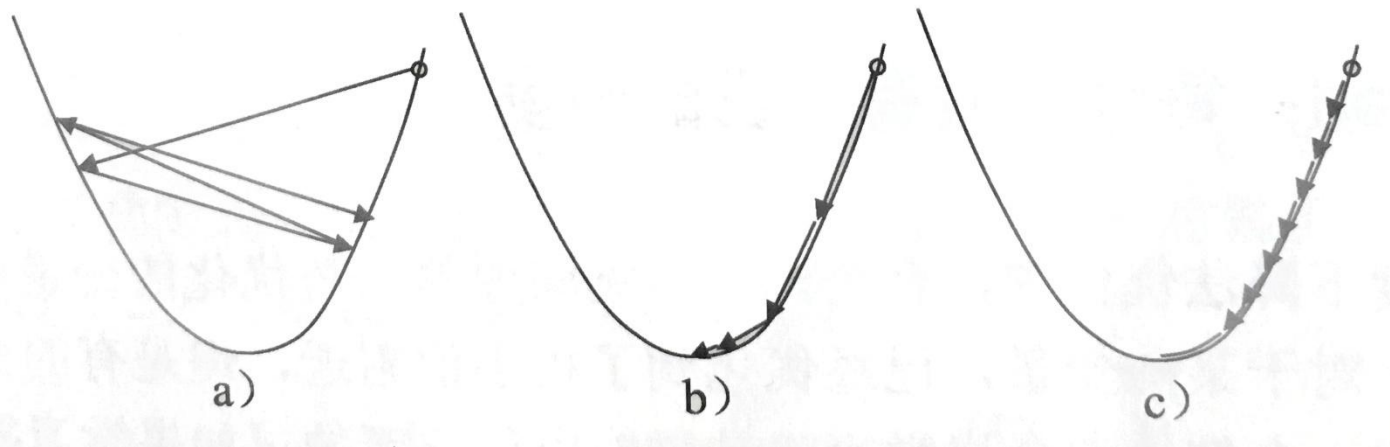
将梯度乘以一个系数 η ，控制更新时步长的大小，叫做
学习率

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 梯度下降法

- 学习率



a: 学习率过大，导致不收敛

c: 学习率过小，迭代步数多

b: 合适的学习率，既保证收敛，又保证效率

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 梯度下降法

- AdaGrad: 自适应学习率

$$(x_{t+1})_i = (x_t)_i - \frac{\eta}{\sqrt{\sum_{\tau=1}^t (\nabla f(x_\tau))_i^2}} \cdot (\nabla f(x_t))_i$$

i 表示第 i 个变量， t 表示第 t 次迭代：

- 1) 每个变量用**不同的**学习率，这个学习率一开始比较大，用于快速梯度下降。
- 2) 随着优化的进行，已经下降很多的变量**减缓**学习率，没怎么下降的变量，保持一个较大的学习率。
- 3) 根据**历史学习率累积总量**来决定当前学习率减小的程度。

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 后向传播算法

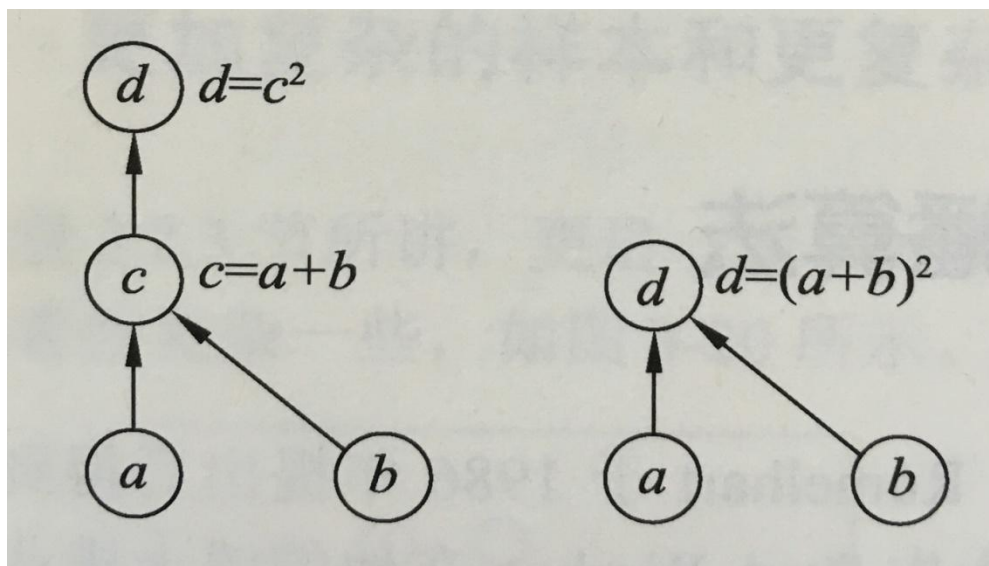
- 多层网络，如2层： $\vec{y} = a_2(W_2 \cdot (a_1(W_1 \cdot \vec{x} + b_1)) + b_2)$
 - 计算神经网络中每个参数 W 、 b 的梯度
 - 基本思想：通过链式法则求出所有参数对损失函数的梯度
 - 可利用计算图进行

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 后向传播算法

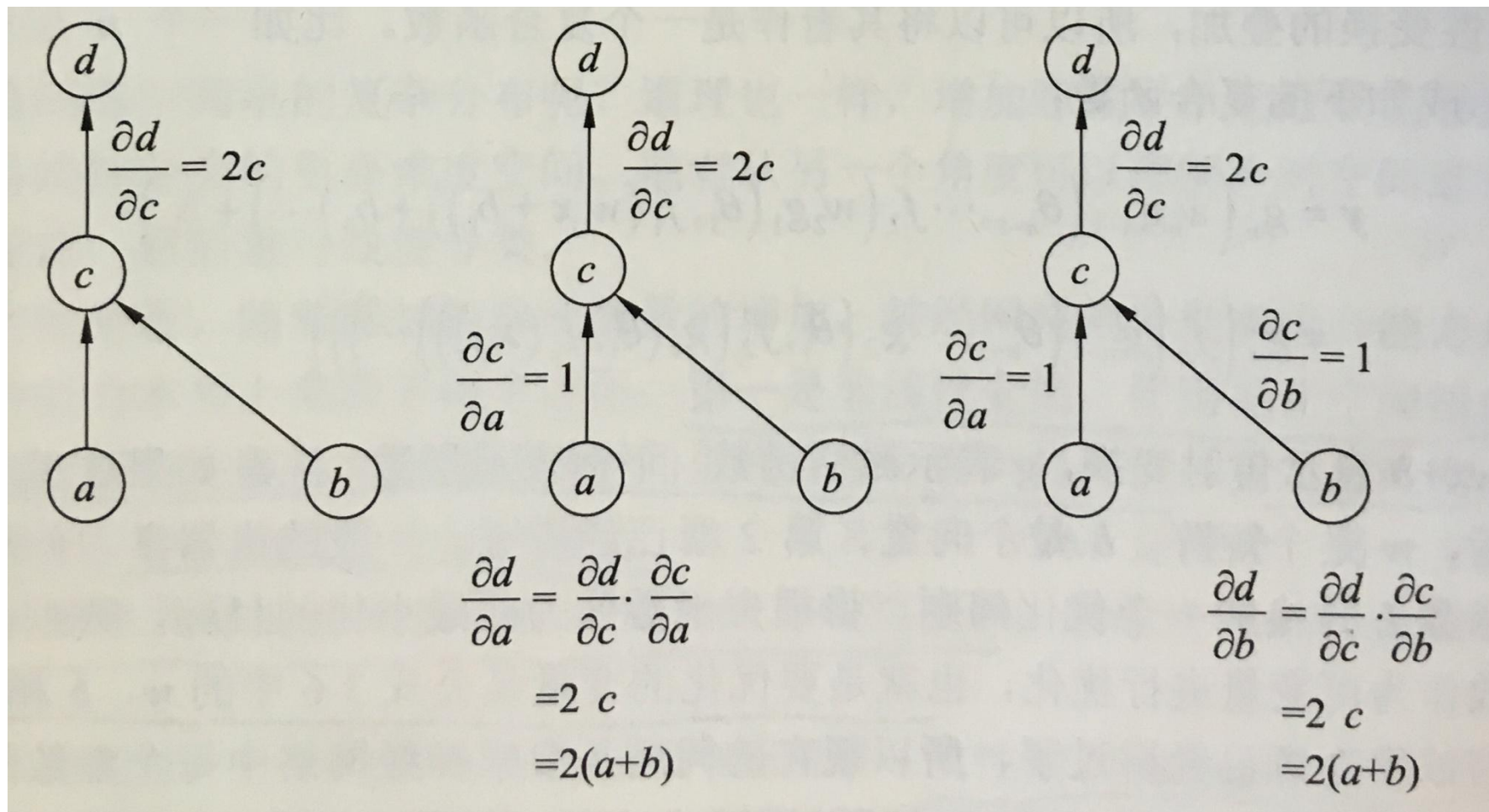
- 节点：变量
 - 边：变量间的关系



计算图的例子

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络



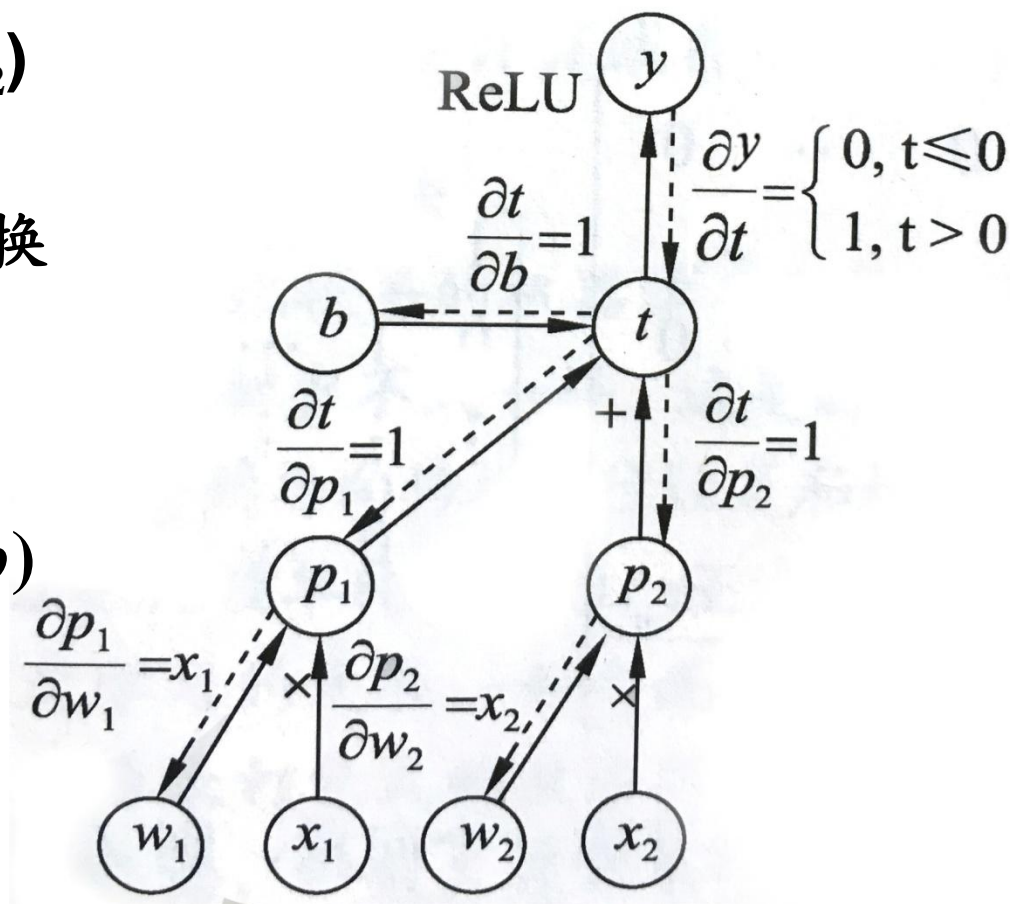
2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 输入：二维向量 (x_1, x_2)
- 优化参数： w_1, w_2 和 b
- $t \rightarrow y$: ReLU非线性变换

$$\text{ReLU}(w \bullet x + b) = \text{ReLU}(w_1 * x_1 + w_2 * x_2 + b)$$

沿着 y 出发把虚线连接的路径上的梯度乘起来，直到叶子节点



一个感知机的例子

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 梯度下降

数据量大

- 全量数据梯度下降：每次计算梯度的时候考虑所有训练数据

- 假设有 N 个样本，计算损失函数的时候，既对所有的 N 个样本都求一遍损失函数的值，求平均。

- 随机梯度下降：每次从训练样本中随机抽取一个样本用来计算损失函数

- 将相应计算出的梯度作为当前一步梯度下降的依据

- 小批量数据：随机选取一定数量的部分样本

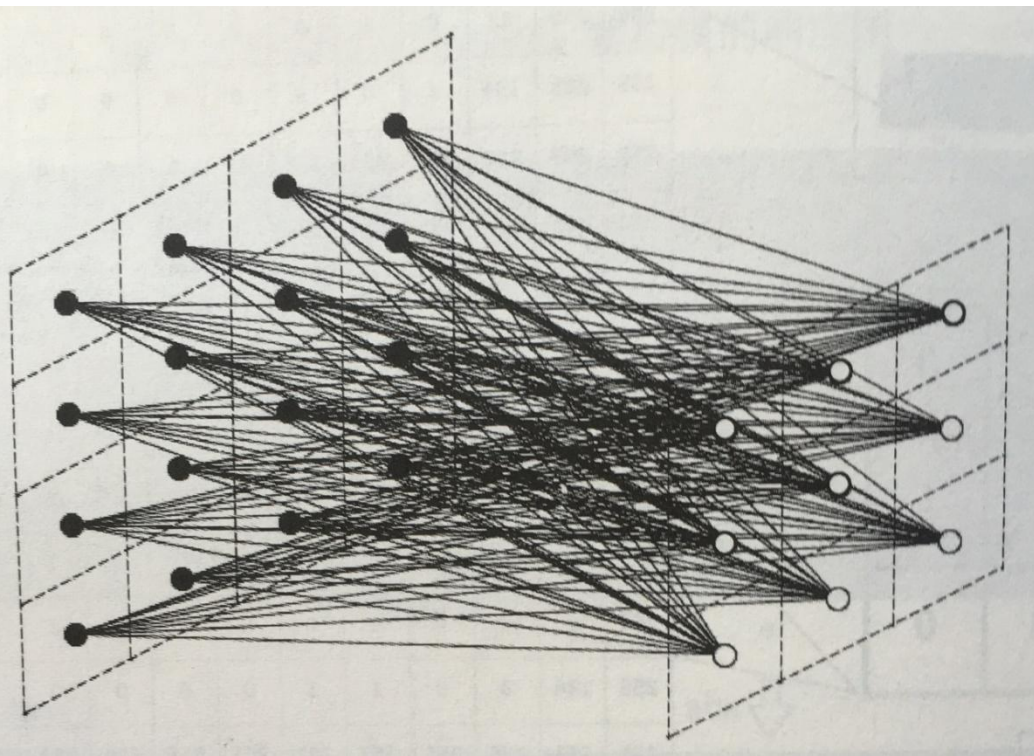
- 随机初始化

- 权重随机初始化：如高斯分布随机数，再乘较小的数

快
不稳定

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
— 神经网络



全连接：

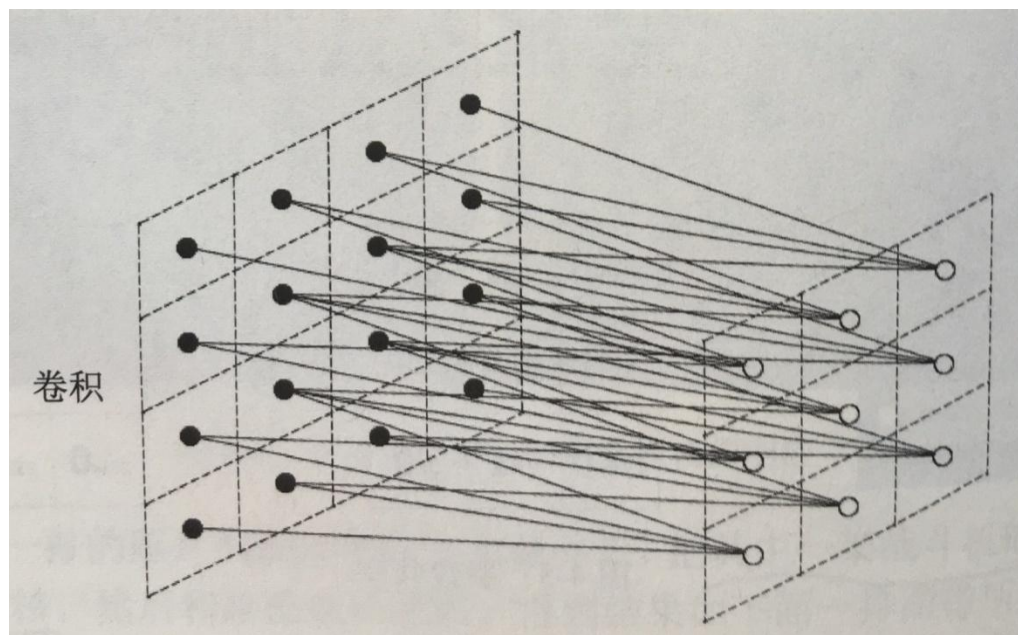
每一层网络都和相邻层全部连接

局限：

- 训练的参数数量大
- 没有考虑图像中像素的空间分布，像素距离远近一视同仁

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - CNN: Convolutional Neural NetWork
 - 相邻层之间的神经元不是全连接，而是部分连接
 - 某个神经元的感知区域来自于上层的部分神经元，而不是像标准的神经网络那样与所有的神经元相连接

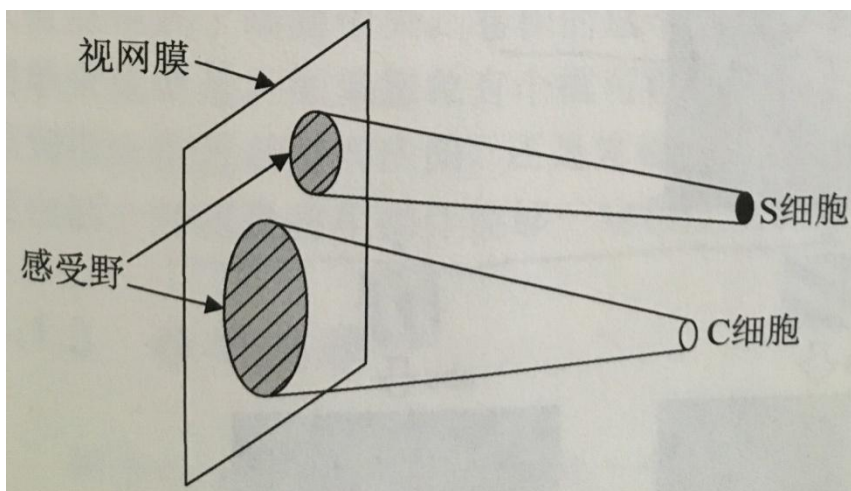


2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 感受野

- 猫视觉皮层：简单细胞、复杂细胞
 - 每个细胞只对特定方向的条形图样刺激有反应
 - 简单细胞对应的视网膜上的光感受细胞所在的区域小
 - 复杂细胞则对应更大的区域，这个区域叫感受野

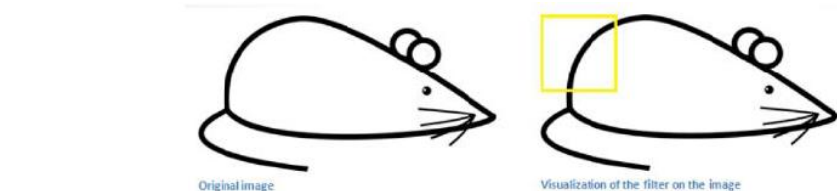


2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 参数共享

- 卷积核(滤波器)在任何位置都是不变的
 - 把和卷积核做卷积之后得到的结果叫做特征响应图
 - 认为卷积核代表某种特征(或一种图像模式)，卷积结果是输入对这种特征的响应，相似度越高响应值越大
 - 具体应用中往往有多个卷积核，如下24个卷积核示例



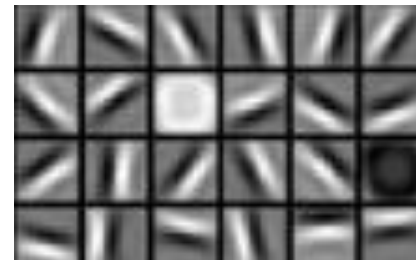
0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

图像

⊗

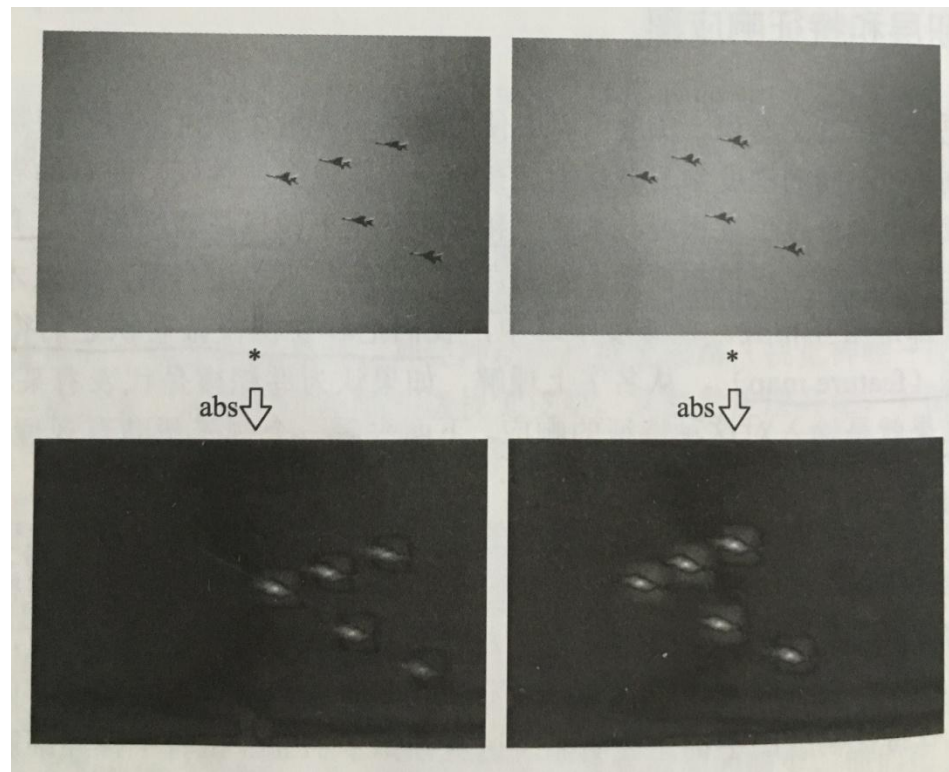
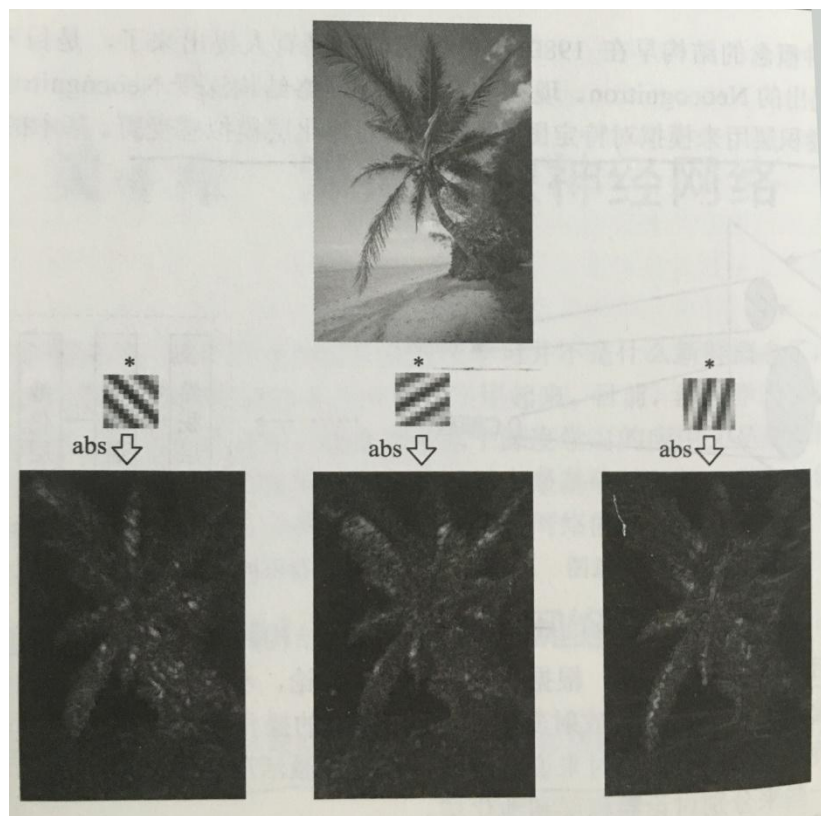
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

卷积核 filter



2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 参数共享

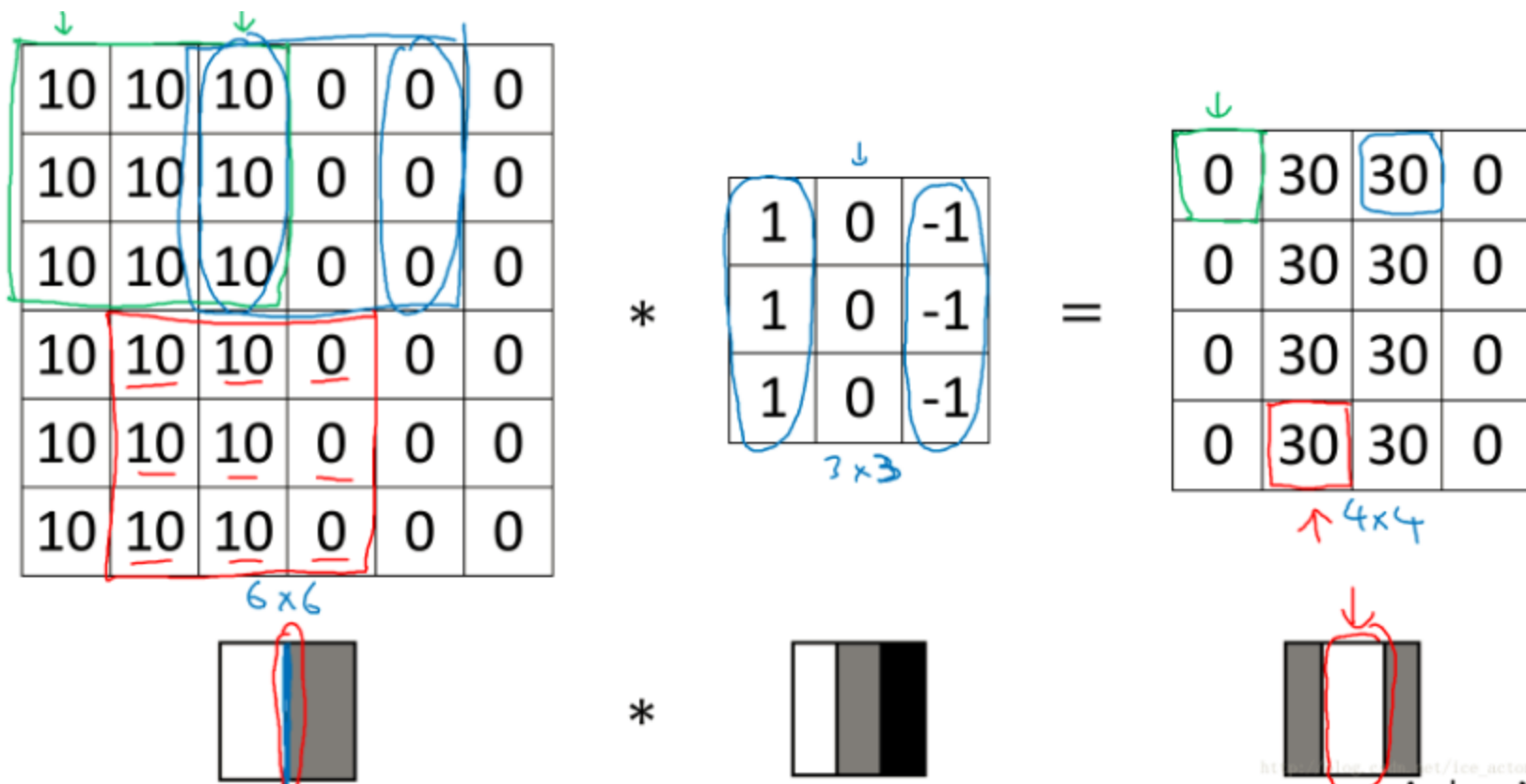


2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 参数共享

- 卷积核：垂直边缘



2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 卷积层：计算
 - 池化层：压缩数据减少参数
 - 全连接层：输出

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 卷积层：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

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 卷积层：计算

INPUT IMAGE					
18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT		
1	0	1
0	1	0
1	0	1

429

INPUT IMAGE					
18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT		
1	0	1
0	1	0
1	0	1

429

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 卷积层：计算

- same padding: 给图像填加一层 0 边界
 - valid padding: 不填加

0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

WEIGHT

1	0	1
0	1	0
1	0	1

139

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 卷积层：计算
 - 步长：移动像素数
 - 如：步长为2

INPUT IMAGE

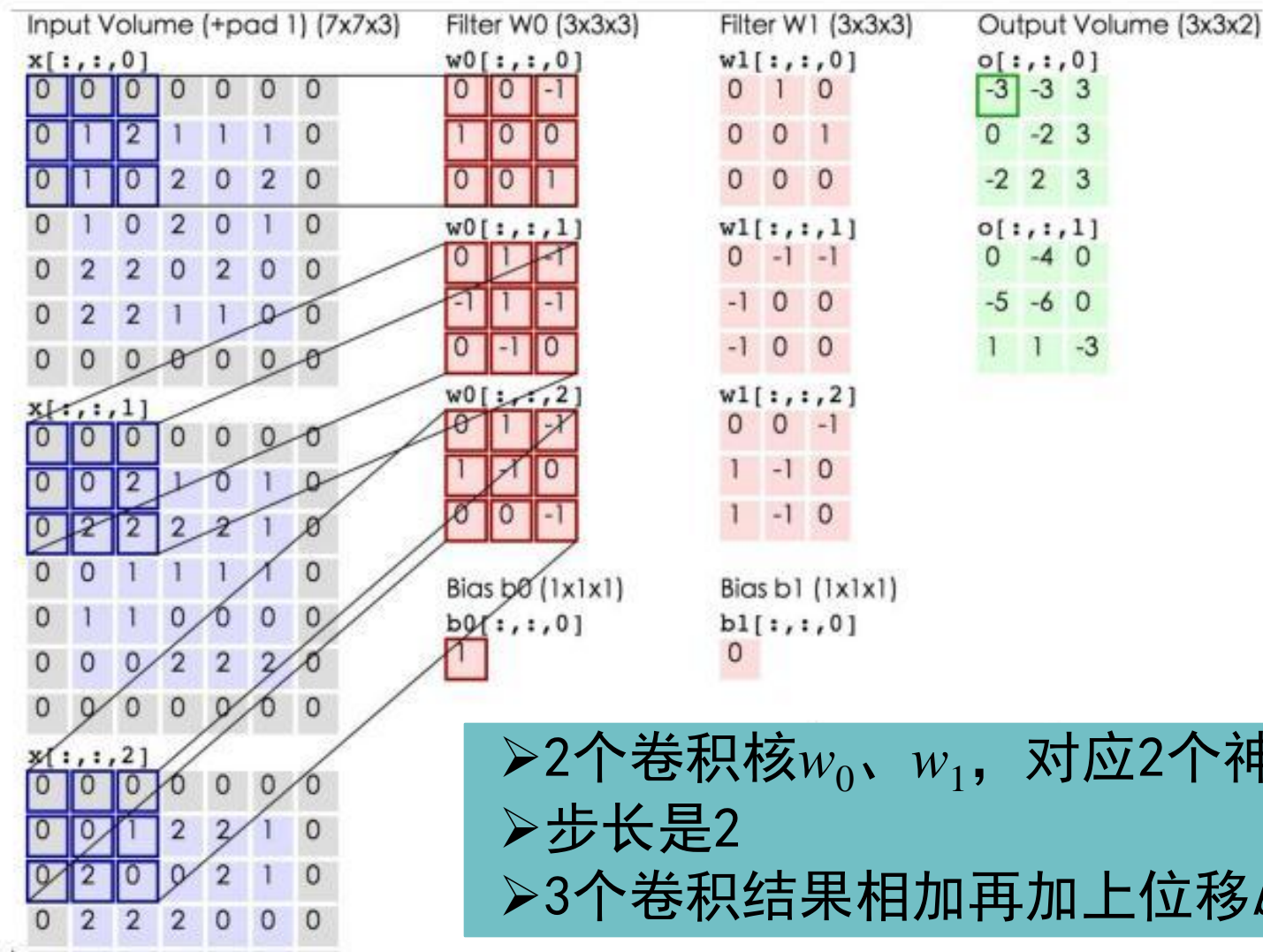
18	54	51	239	244
55	121	75	78	95
35	24	204	113	109
3	154	104	235	25
15	253	225	159	78

WEIGHT

1	0	1
0	1	0
1	0	1

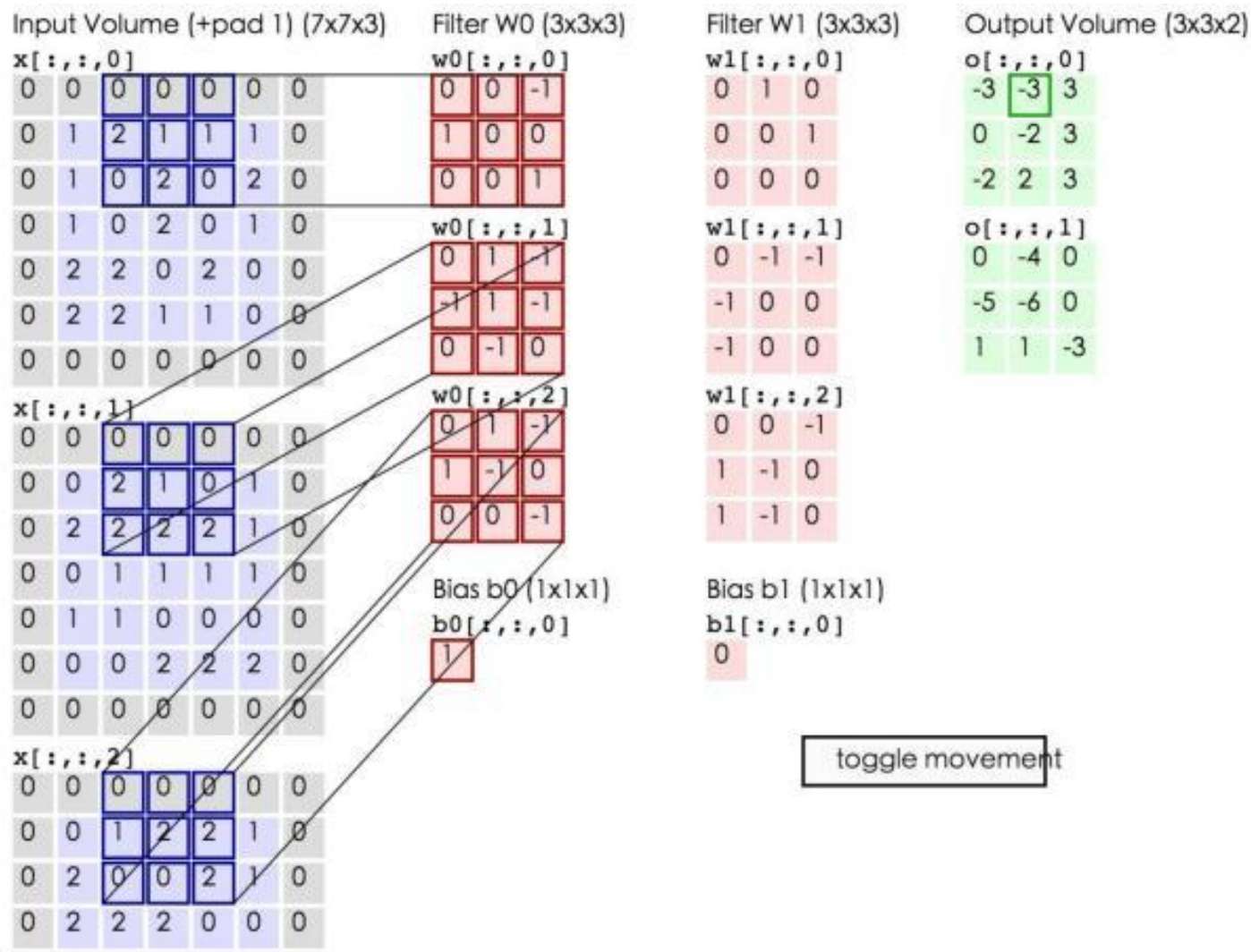
429

2.3.3 视觉媒体数字化



- 2个卷积核 w_0 、 w_1 ，对应2个神经元
- 步长是2
- 3个卷积结果相加再加上位移 b

2.3.3 视觉媒体数字化



2.3.3 视觉媒体数字化

Input Volume (+pad 1) (7x7x3)

$x[:, :, 0]$

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

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

$x[:, :, 1]$

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

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

$x[:, :, 2]$

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

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

Filter W0 (3x3x3)

$w0[:, :, 0]$

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

$w0[:, :, 1]$

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

$w0[:, :, 2]$

1	1	1
-1	1	0
-1	1	0

Bias b0 (1x1x1)

$b0[:, :, 0]$

1

Filter W1 (3x3x3)

$w1[:, :, 0]$

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

$w1[:, :, 1]$

0	0	-1
1	0	0
1	1	0

$w1[:, :, 2]$

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

Bias b1 (1x1x1)

$b1[:, :, 0]$

0

Output Volume (3x3x2)

$o[:, :, 0]$

5	-1	4
7	3	2
8	5	11

$o[:, :, 1]$

-2	0	5
-4	8	7
1	0	6

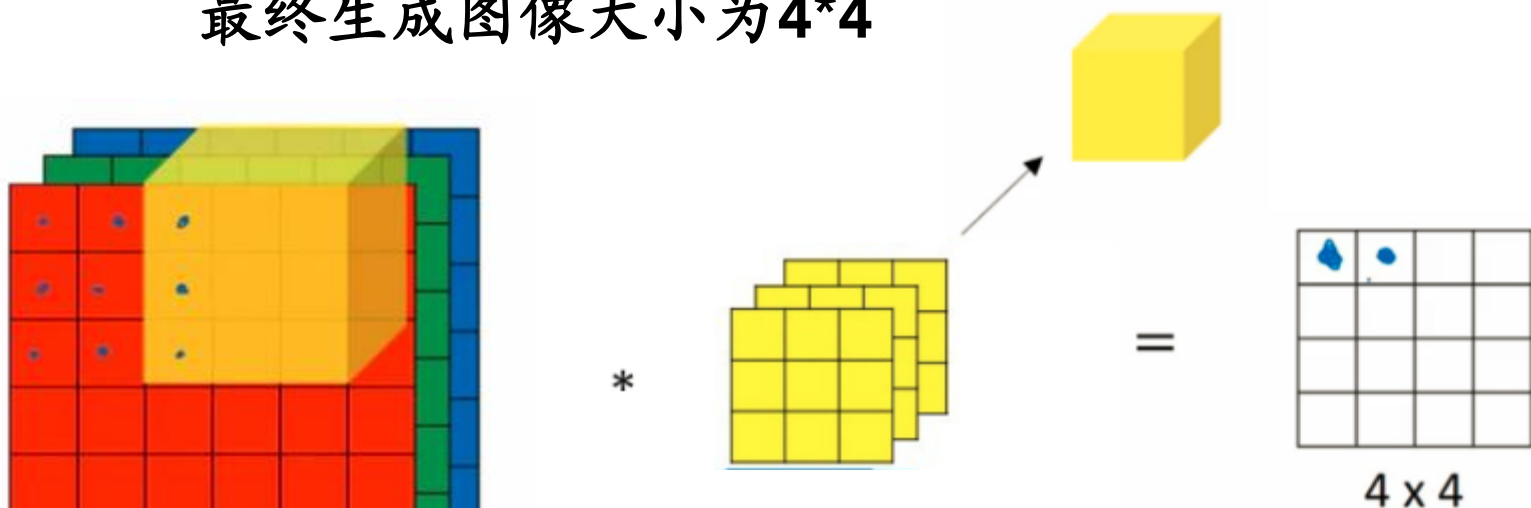
toggle movement

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 卷积层：计算

- 如：在RGB图像上进行卷积，过滤器的大小不是 3×3 而是有 $3 \times 3 \times 3$ ，最后的3对应为通道数（channels）
 - 卷积结果中每个像素值为 $3 \times 3 \times 3$ 过滤器对应位置和图像对应位置相乘累加，过滤器依次在RGB图像上滑动，最终生成图像大小为 4×4



2.3.3 视觉媒体数字化

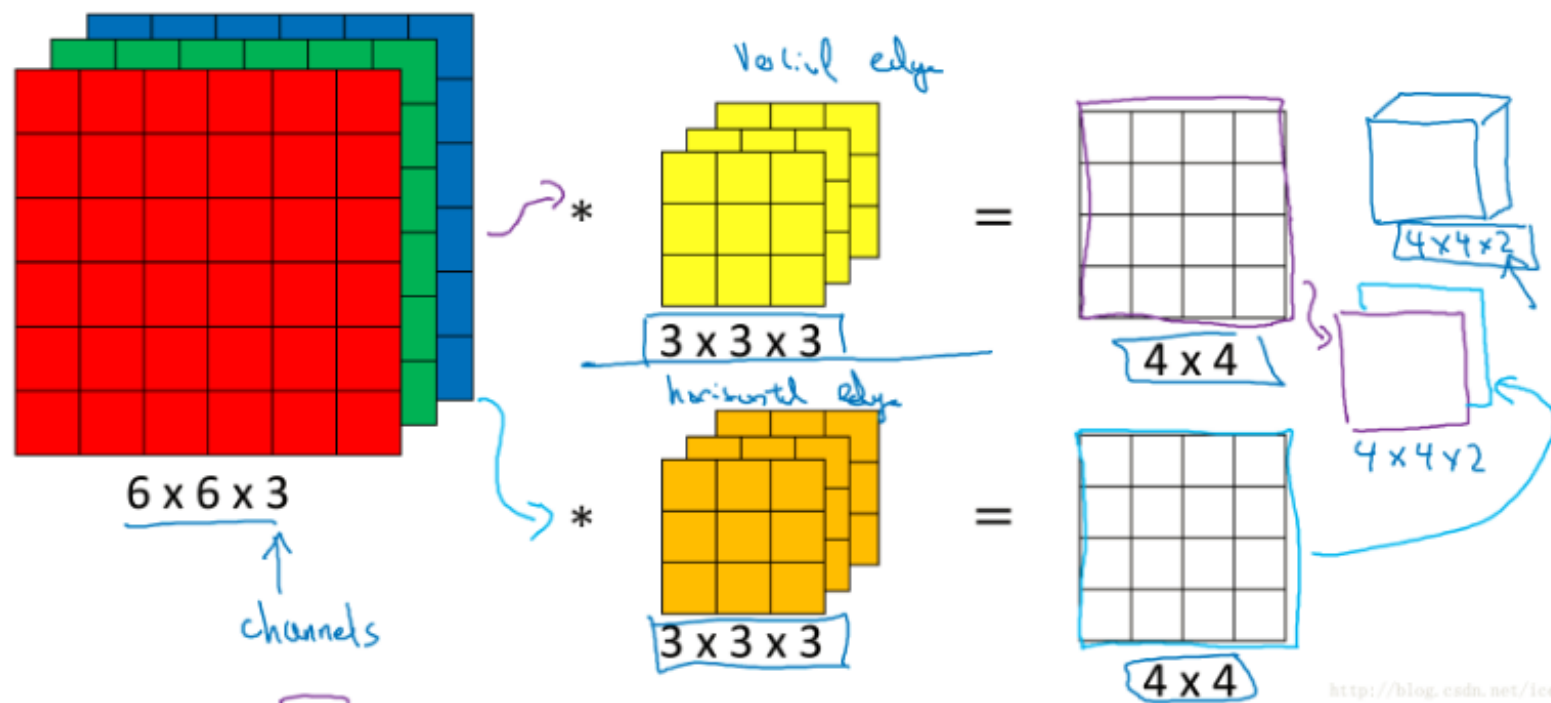
- 新技术：卷积神经网络

- 卷积层：计算

- 如果不仅在图像中检测一种类型的特征，而是要同时检测垂直边缘、水平边缘、45度边缘等等，也就是多个过滤器的问题。
 - 如果有两个过滤器，最终生成图像为 $4*4*2$ 的立方体，这里的2来源于采用了两个过滤器。
 - 如果有10个过滤器那么输出图像就是 $4*4*10$ 的立方体

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 单层卷积网络：例子



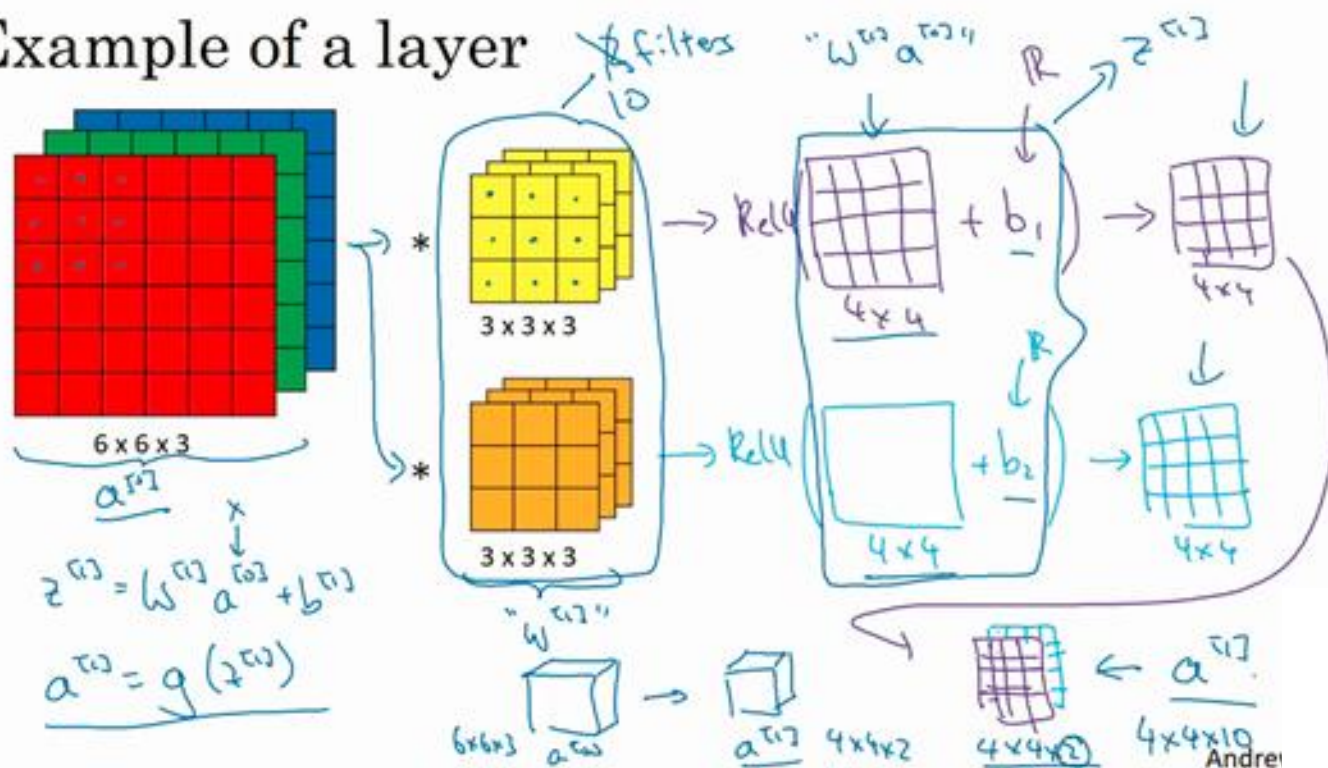
http://blog.csdn.net/ice_actor

对两个4*4的矩阵，分别加入偏差 b_1 和 b_2 ，然后对加入偏差的矩阵做非线性的Relu变换，得到2个新的4*4矩阵，这就是卷积神经网络的一层，如下图

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 单层卷积网络：例子

Example of a layer



$$z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$$

$$a^{[1]} = g(z^{[1]})$$

若10个过滤器，则有 $(27+1) * 10 = 280$ 个参数

2.3.3 视觉媒体数字化

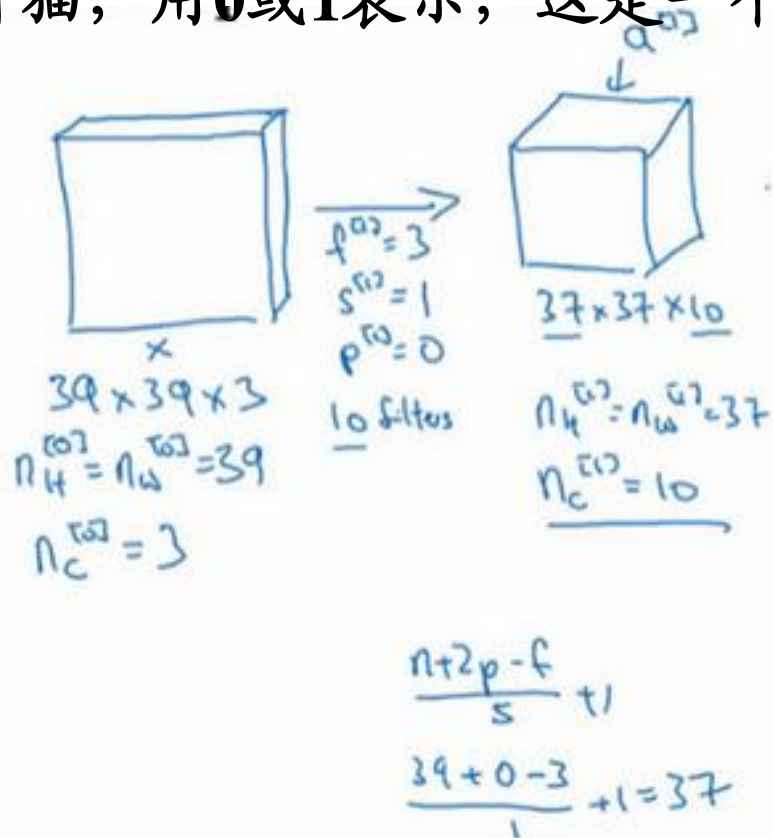
- 新技术：卷积神经网络

- 简单卷积网络：例子

- 假设有一张图片，进行分类或识别，把这张图片输入定义为 x ，辨别图片中有没有猫，用0或1表示，这是一个分类问题

输入：39*39*3的图片；

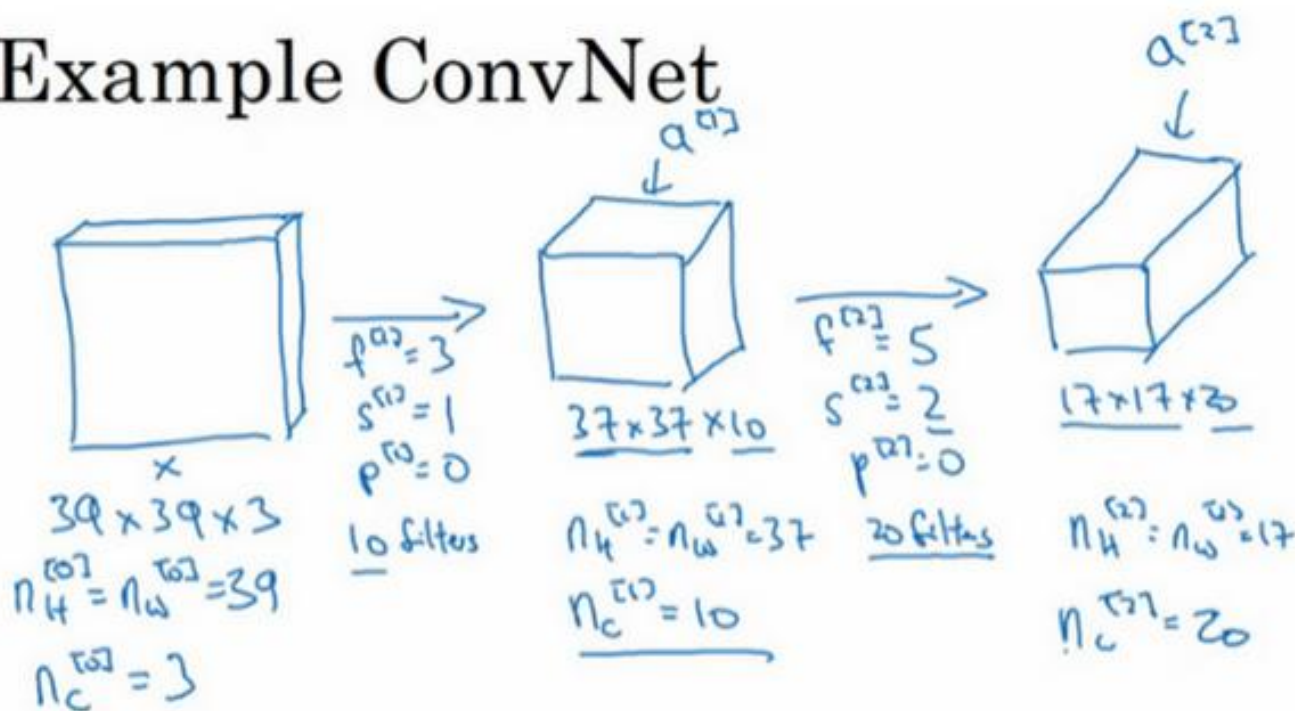
第一层：10个过滤器，3*3大小，步长1，valid padding，输出结果维度37*37*10



2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 简单卷积网络：例子

Example ConvNet

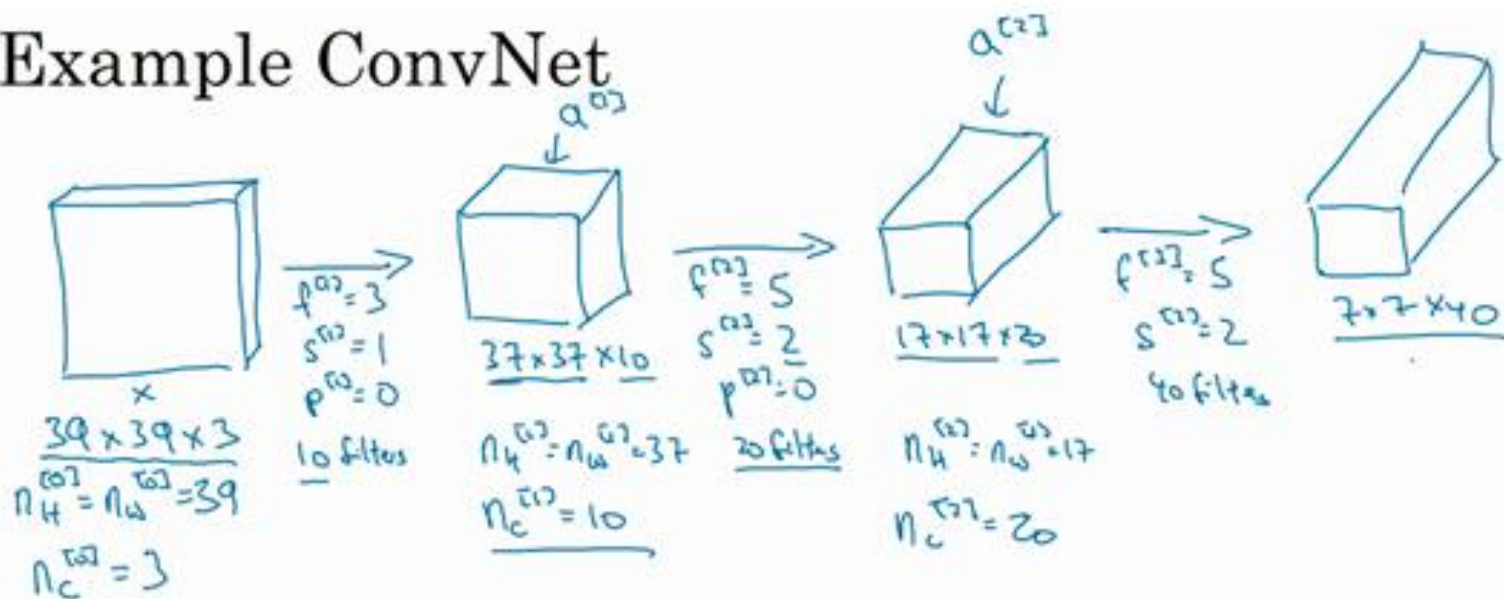


第二层：20个过滤器，5*5大小，步长2，valid padding，输出结果维度17*17*20

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 简单卷积网络：例子

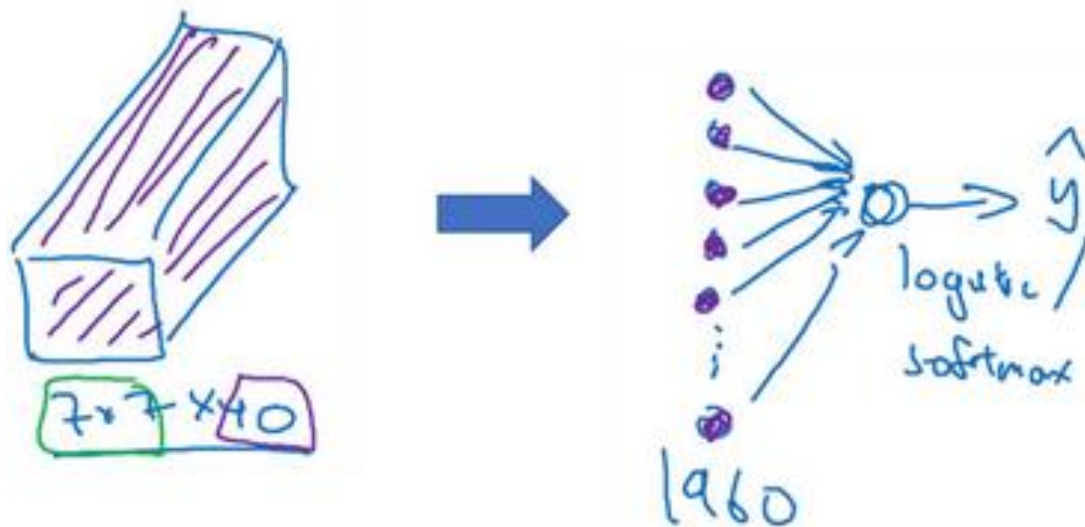
Example ConvNet



第三层：40个过滤器，5*5大小，步长2，valid padding，输出结果维度7*7*40=1960个特征

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 简单卷积网络：例子



Softmax: 对于一个n分类问题, 给定输入 x 属于第 i 类(y_i)的一种原始度量 $h(x, y_i)$, 属于某一类的概率:

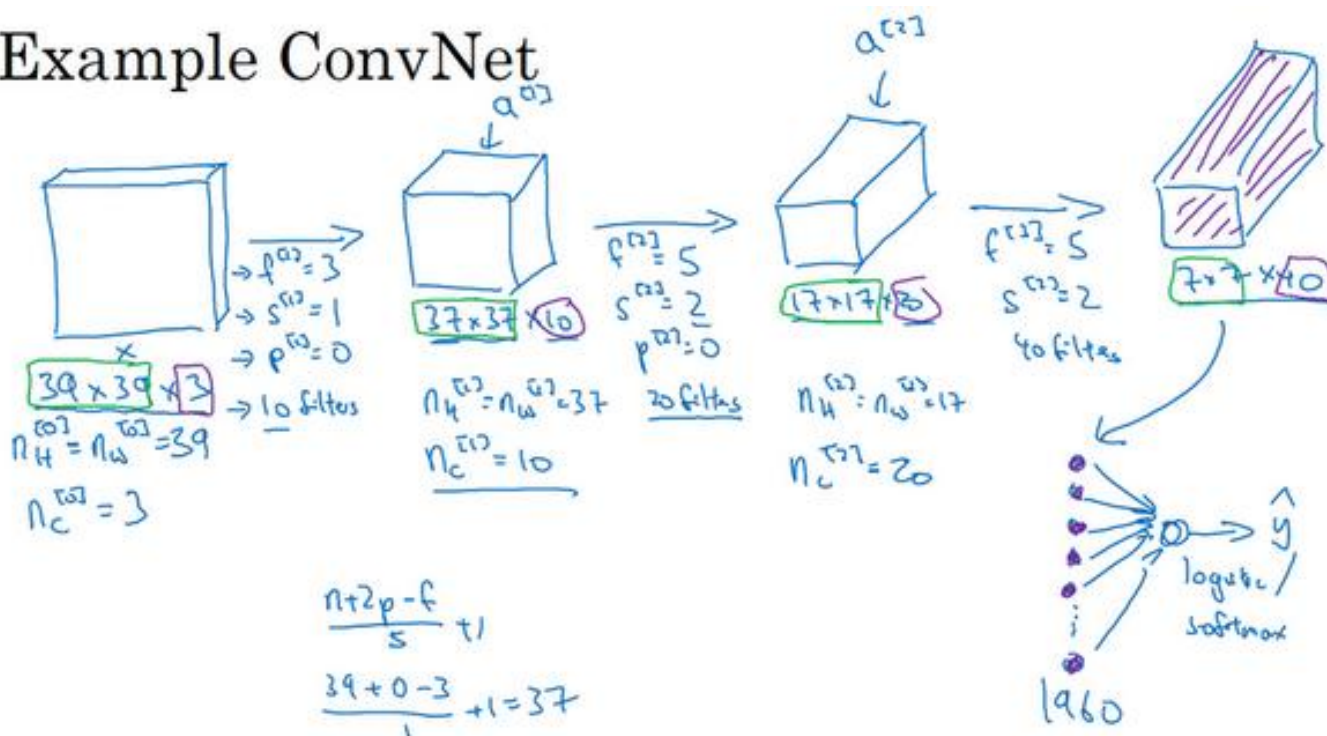
$$P(y|x) = \frac{e^{h(x, y_i)}}{\sum_{j=1}^n e^{h(x, y_j)}}$$

平滑或展开成1960个单元, 成为一个向量, 用logistic回归 (二分类) 或softmax回归 (k类, 属于某一类的概率)

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 简单卷积网络：例子

Example ConvNet



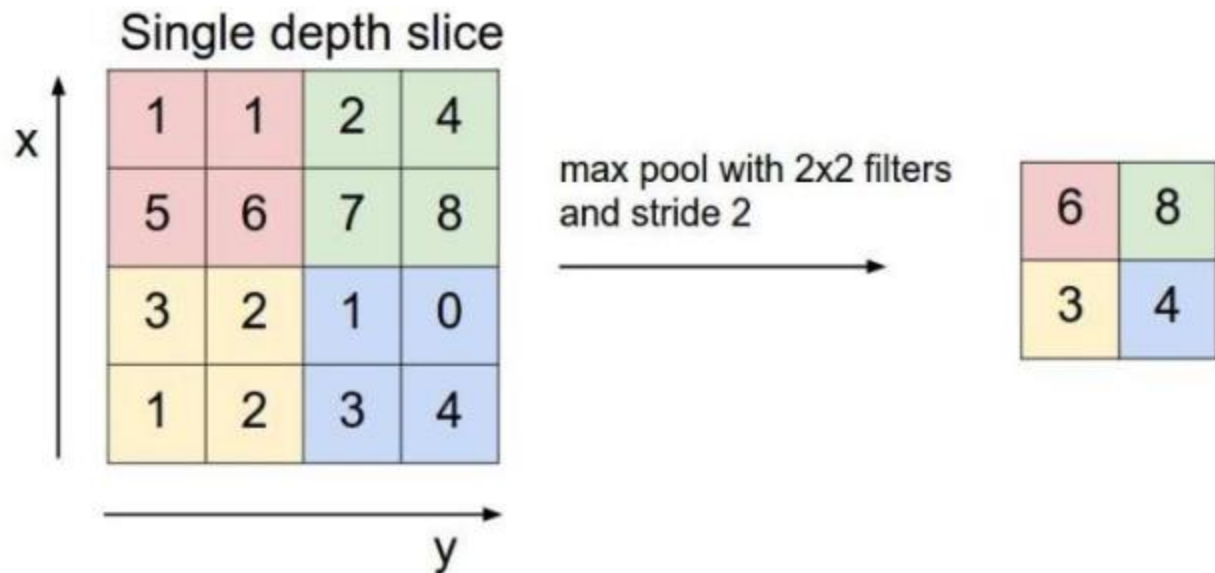
图像越来越小：37、17、7；通道越来越多：3、10、20、40

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 池化层：压缩数据减少参数

- 对特征响应图上的一个给定区域求出一个能代表这个区域特点的值
 - 最大值池化：取出小区域里的最大值作为结果
 - 平均值池化：取小区域里的平均值



对4*4的图做2*2的池化，步长为2
若多通道需要做，
那分通道计算池化操作

2.3.3 视觉媒体数字化

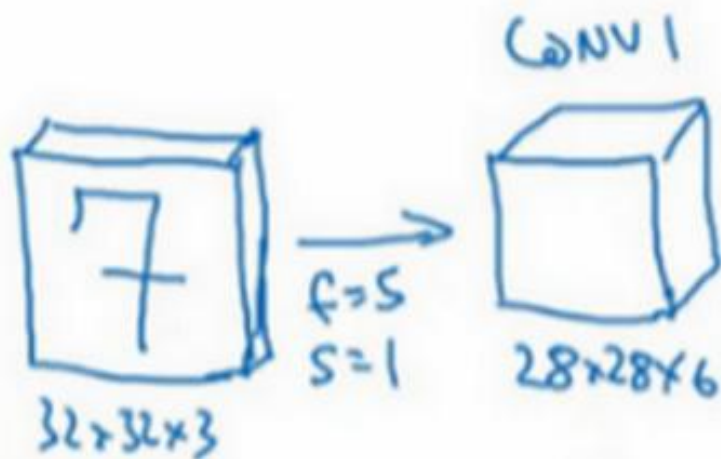
- 新技术：卷积神经网络
 - 全连接层：输出
 - 类似神经网络的全连接层
 - 卷积和池化提取特征、减少数据参数，需要全连接层生成需要的输出
 - 损失函数，计算预测误差

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

- 示例：手写体数字识别

- 识别一张 $32*32*3$ RGB图片中含有的某个数字，比如7
 - 和经典网络LeNet-5非常相似

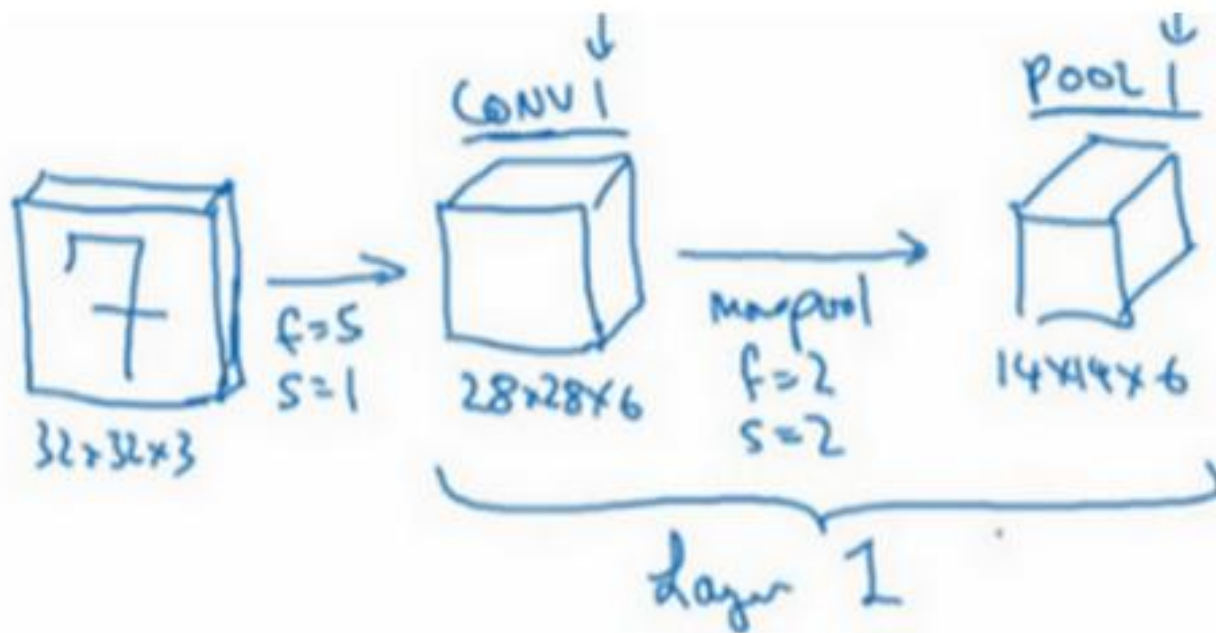


输入： $32 \times 32 \times 3$ 的图片；

第一层：6个过滤器， 5×5 大小，步长1，valid padding，输出结果维度 $28 \times 28 \times 6$

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 示例：手写体数字识别

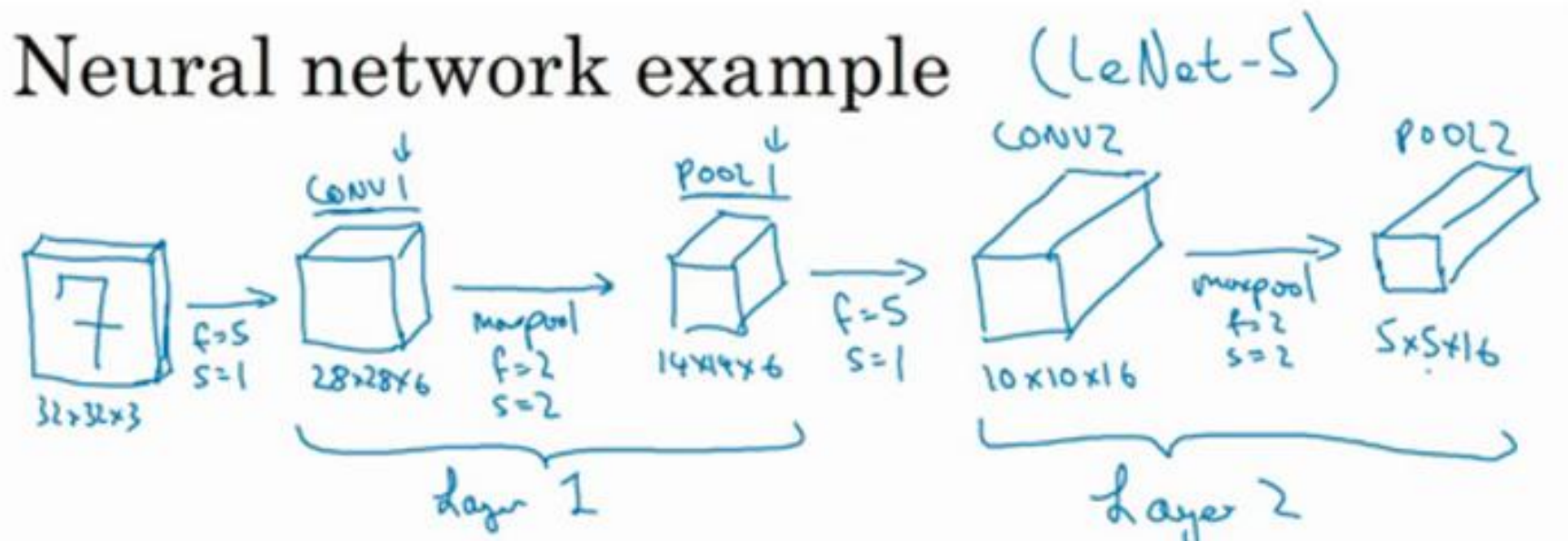


池化层：最大池化，2*2大小，步长2，不填充，输出结果维度14*14*6

层：一种卷积+池化为一层，一种每个单独为一层

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
 - 示例：手写体数字识别



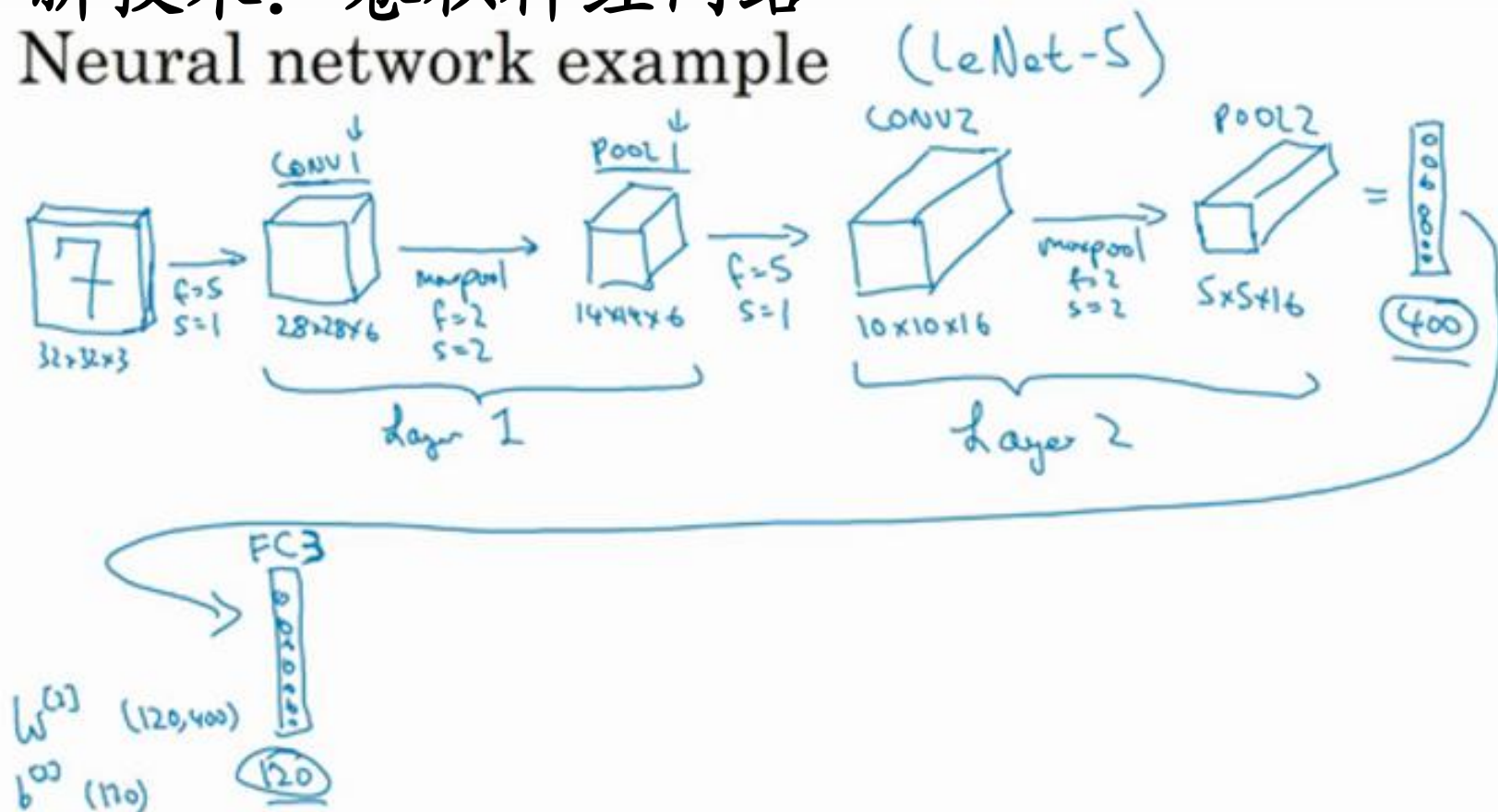
卷积层：16个过滤器， 5×5 大小，步长1，不填充，输出结果维度 $10 \times 10 \times 16$

池化层：最大池化， 2×2 大小，步长2，不填充，输出结果维度 $5 \times 5 \times 16 = 400$

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

Neural network example

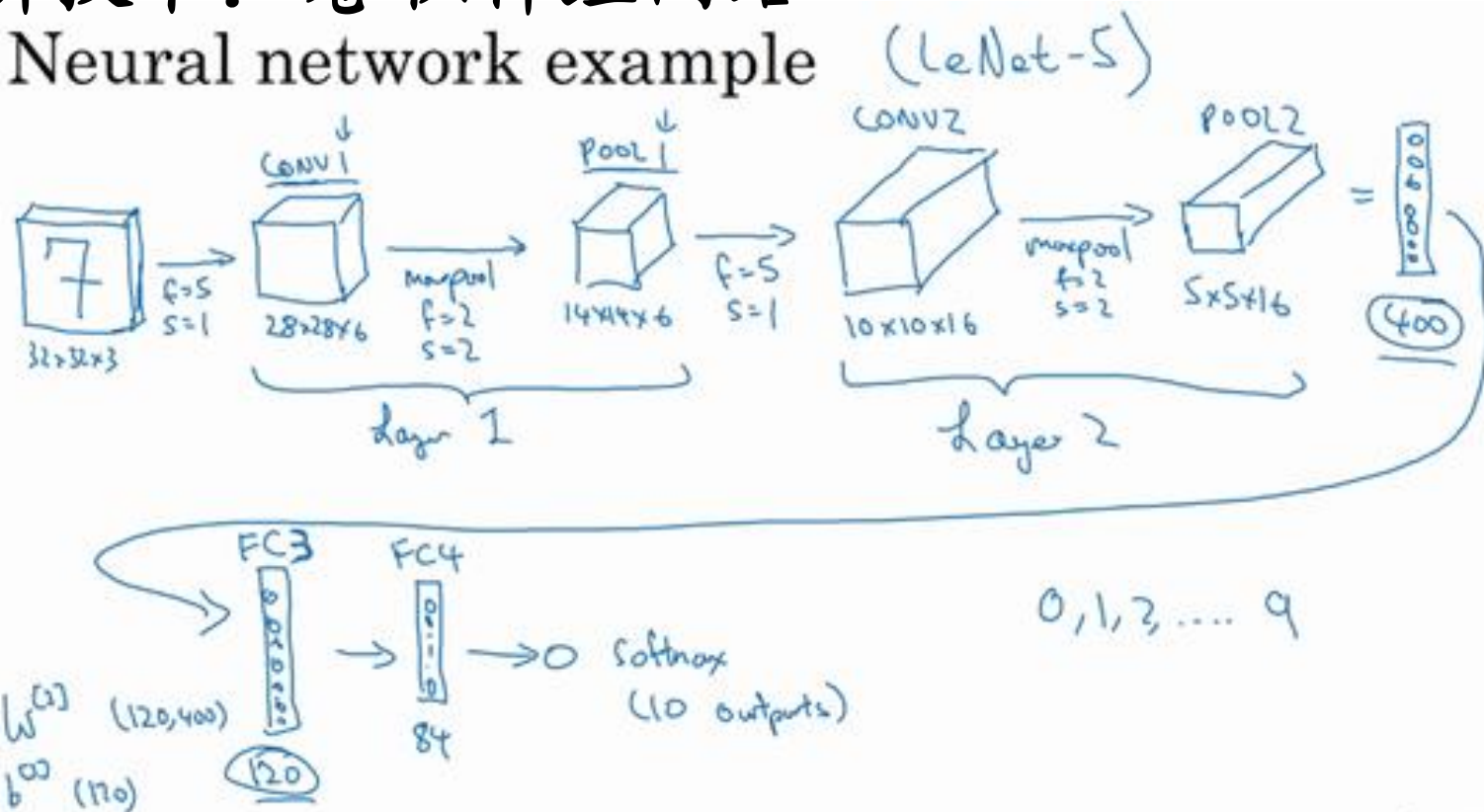


将POOL2平整化为大小为400的一维向量，构建下一层(含有120个单元)，这是第一个全连接层(Fully connected)，标记为FC3。权重矩阵维度 120×400 ，还有一个偏差参数。输出120个维。

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

Neural network example



对这个120个单元再添加一个全连接层，这层更小，假设它含有84个单元，标记为FC4。

用84个单元填充一个softmax单元，这个softmax有10个输出，因为识别0-9这10个手写数字。

思考题

配置流行的**CNN**开源库（如**TensorFlow**），利用公开的数据集（如**ImageNet**、**MNIST**、**CIFAR-10**等），实现手写识别或图像识别。

推荐书目：郑泽宇等，**TensorFlow**实战**Google**深度学习架构，电子工业出版社，**2017**年**3**月

推荐语言：**Python**

作业4

- 1、图像分割的种子填充算法的基本步骤是什么？如何提高该算法分割结果的正确性？
- 2、卷积神经网络包含哪些基本步骤？参数共享在哪个步骤实现，降采样通过哪个步骤实现？
- 3、卷积神经网络与神经网络的区别是什么？
- 4、假设输入图像大小 $n*n$ ，过滤器大小 $f*f$ ，在步长为1且不填充边界情况下卷积后的输出图像大小是多少？
- 5、假设输入图像大小 $n*n$ ，过滤器大小 $f*f$ ，步长为 s ， p 为每边界填充像素数，卷积后的输出图像大小是多少？