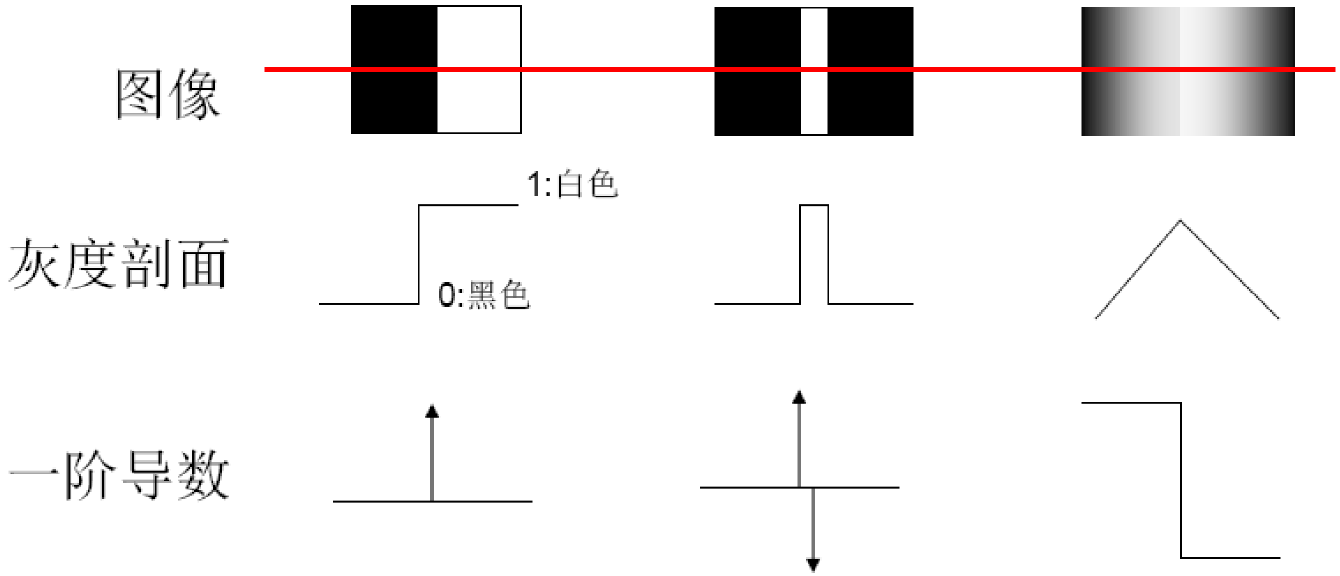
多媒体技术

回顾

- 视觉
  - BMP文件格式
  - GIF文件格式
- 图像分割
  - 基于阈值的分割
    - 多阈值法
    - 灰度直方图阈值法
    - OTSU
  - 基于边缘的分割
    - 导数、梯度、差分
    - 相关和卷积

2.3.3 视觉媒体数字化

- 图像处理技术
  - （2）基于边缘的分割



结论：一阶导数在图像由暗变明的位置处有1个向上的阶跃，而其他位置都为0，这表明可用一阶导数的幅度值来检测边缘的存在，幅度峰值一般对应边缘位置。

2.3.3 视觉媒体数字化

- 一个二元连续函数表示为*f*(x,y)，它在 (x,y)的梯度可表示为：

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

梯度是一个矢量，模和方向角为：

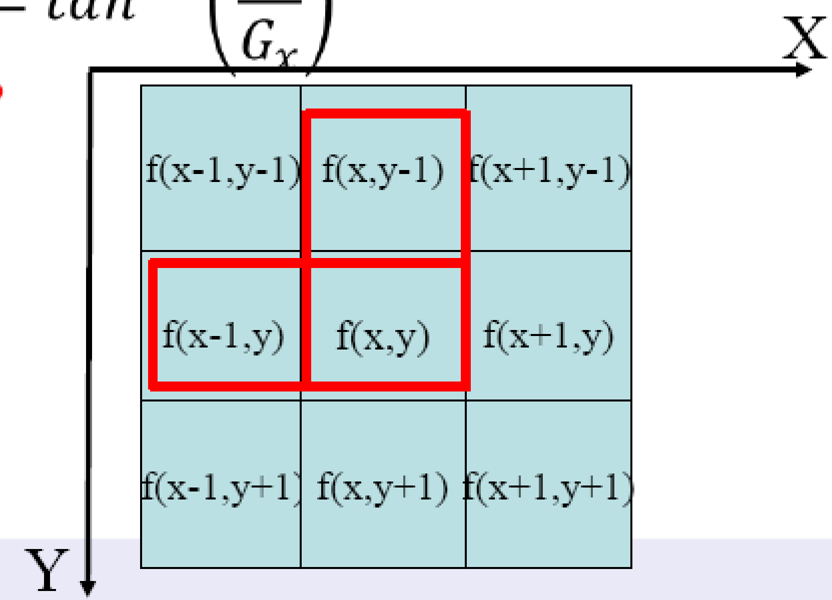
$$\nabla f = mag(\nabla f) = \sqrt{G_x^2 + G_y^2}$$

$$\alpha(x,y) = tan^{-1}\left(\frac{G_y}{G_x}\right)$$

问题：离散数字图像如何求梯度？

可用一阶差分代替求偏导：

$$\nabla_x f(x,y) = f(x,y) - f(x-1,y)$$
$$\nabla_y f(x,y) = f(x,y) - f(x,y-1)$$



2.3.3 视觉媒体数字化

一个二元连续函数表示为$f(x,y)$，它在 (x,y)的梯度可表示为：

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

梯度是一个矢量，模和方向角为：

$$\nabla f = mag(\nabla f) = \sqrt{G_x^2 + G_y^2}$$

$$\alpha(x,y) = tan^{-1}\left(\frac{G_y}{G_x}\right)$$

梯度：一阶差分求偏导；梯度的模： $\nabla f = mag(\nabla f) = \sqrt{G_x^2 + G_y^2}$

原始图像

Gx

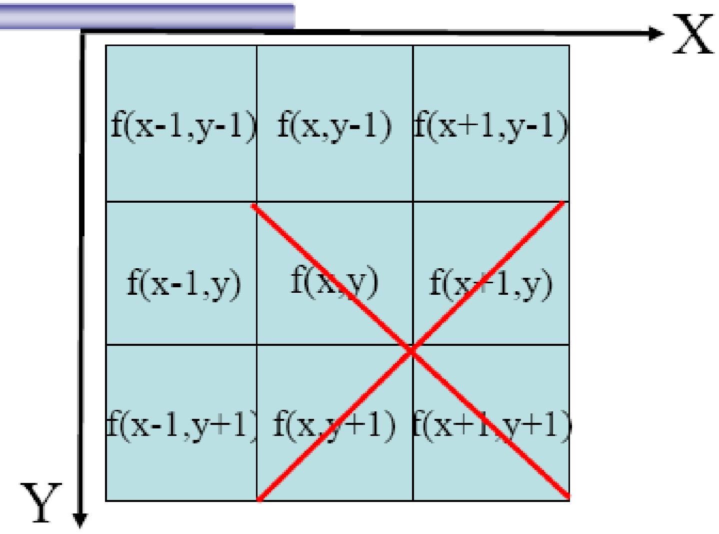
Gy

$\nabla \mathbf{f} \approx |Gx| + |Gy|$

2.3.3 视觉媒体数字化

Roberts方法：采用对角方向相邻两像素之差表示微分，即

$$\nabla_x f(x,y) = f(x,y) - f(x+1,y+1)$$
$$\nabla_y f(x,y) = f(x+1,y) - f(x,y+1)$$



相关与卷积：信号与系统分析中基本运算相关与卷积，在实际图像处理中都表现为邻域运算。

– 邻域运算：输出图像中每个像素是由对应的输入像素及其一个邻域内的像素共同决定；

– 通常邻域是远比图像尺寸小的一规则形状 。如一个点的邻域定义为以该点为中心的一个圆内部或边界上点的集合 。

2.3.3 视觉媒体数字化

- 相关运算

两个连续函数f(x)… $x)\circ f(x)=\int$…

– 给定图像f(x,y)… 小m*m…

– 相关运算：首… ,(m-1)/2… 应

$$g(x,y)=T\circ f(x \ldots \frac{m-1}{2}, y+j-\frac{\quad}{2})$$

f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

T00	T01	T02
T10	T11	T12
T20	T21	T22

当$m=3$时

$$g(x,y)=T(0,0)f(x-1,y-1)+T(0,1)f(x-1,y)+T(0,2)f(x-1,y+1)+\\ T(1,0)f(x,y-1)+T(1,1)f(x,y)+T(1,2)f(x,y+1)+\\ T(2,0)f(x+1,y-1)+T(2,1)f(x+1,y)+T(2,2)f(x+1,y+1)$$

将图像中的每个像素与模板中的每个元素对应相乘，所有乘积之和作为中心像素的新值g(x, y)。

2.3.3 视觉媒体数字化

- 卷积运算

两个**连续函数**f(... (x)*f(

– 给定图像f(x, ... 大小m

– 卷积运算：德 ... -1)/2)

$$g(x,y)=T\circ f(x \ldots + \frac{m-\ldots}{2}$$

f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

T00	T01	T02
T10	T11	T12
T20	T21	T22

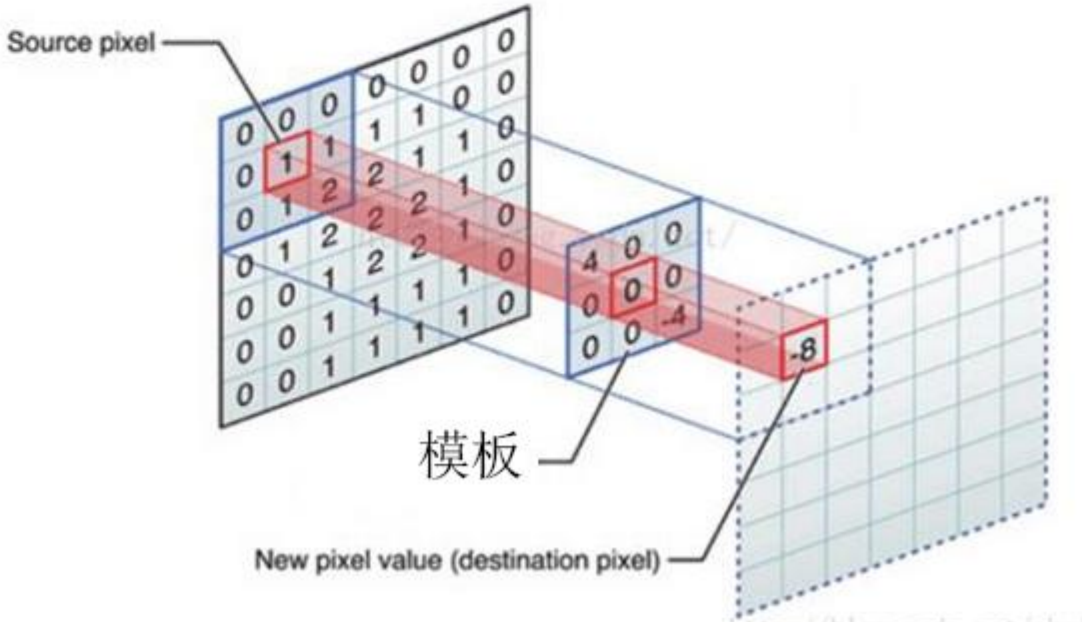
当$m=3$时

$$\begin{aligned} g = {} & T(0,0)f(x+1,y+1)+T(0,1)f(x+1,y)+T(0,2)f(x+1,y-1)+ \\ & T(1,0)f(x,y+1)+T(1,1)f(x,y)+T(1,2)f(x,y-1)+ \\ & T(2,0)f(x-1,y+1)+T(2,1)f(x-1,y)+T(2,2)f(x-1,y-1) \end{aligned}$$

模版（卷积核）旋转180度；再加权平均（类似相关运算）。

2.3.3 视觉媒体数字化

例题：相关运算



模板：$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -4 \end{bmatrix}$

像素位置（2，2）的相关运算

选择邻域3*3：$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$

$$\begin{array}{r} 4\times 0 \\ 0\times 0 \\ 0\times 0 \\ 0\times 0 \\ 0\times 1 \\ 0\times 1 \\ 0\times 0 \\ 0\times 1 \\ +\ \ 2\times(-4) \\ \hline -8 \end{array}$$

原始图像7行*7列

问题：如何对整幅图像进行相关操作？

多媒体技术

2.3.3 视觉媒体数字化

对数字图像做相关运算其实就是利用模板在图像上滑动：

步骤1：根据模板的尺寸，选取当前像素的邻域；

步骤2：将该邻域上的像素灰度值与模板上的对应的数值相乘；

步骤3：相乘后的值相加作为当前像素的灰度值；

步骤4：重复上述步骤；

- 原始图片为7*7，模板的大小为3*3；由于像素邻域与模板大小一致，因此当相关运算后只能获得5*5的图片，计算后的图片和原始图片尺寸不一致；
- 方法：原始图像的边缘拷贝到处理后的图像或者**边界填充为0**；

0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

2.3.3 视觉媒体数字化

f(x,y)	f(x+1,y)
f(x,y+1)	f(x+1,y+1)

- Roberts算子

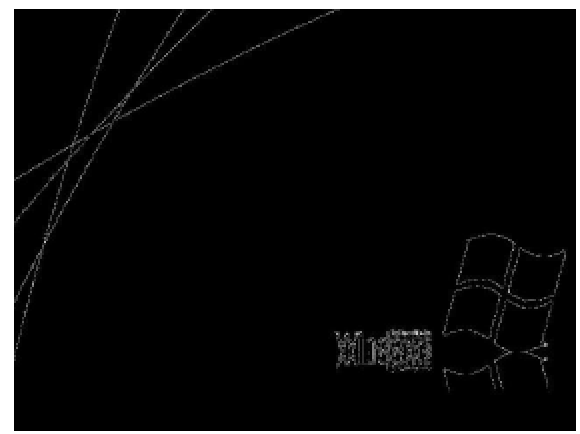
$$\nabla_x f(x,y) = f(x,y) - f(x+1,y+1)$$
$$\nabla_y f(x,y) = f(x+1,y) - f(x,y+1)$$

$$\Rightarrow Gx \begin{array}{|c|c|}\hline 1 & \\ \hline & -1 \\ \hline\end{array} \qquad Gy \begin{array}{|c|c|}\hline & 1 \\ \hline -1 & \\ \hline\end{array}$$

程序实现

Matlab ： edge(imgGray, ‘Roberts’)

Roberts算子



2.3.3 视觉媒体数字化

- 设定阈值提高边缘检测效果

梯度图像　　　　阈值分割结果（二值化）

$$g(x,y)=\begin{cases}1, & f(x,y)\geq T\\ 0, & f(x,y)<T\end{cases}$$

其中：“1”表示物体（对象、目标）

“0”表示背景。

2.3.3 视觉媒体数字化

Roberts算子

	2	1593	1594	1595	1596	1597	1598	1599	1600
1180	165	165	165	165	165	165	165	165	165
1181	165	165	165	165	165	165	165	165	165
1182	165	165	165	165	165	165	165	165	165
1183	165	165	165	165	165	165	165	165	165
1184	165	165	165	165	165	165	165	165	165
1185	165	165	165	165	165	165	165	165	165
1186	165	165	165	165	165	165	165	165	165
1187	165	165	165	165	165	165	165	165	165
1188	165	165	165	165	165	165	165	165	165
1189	165	165	165	165	165	165	165	165	165
1190	165	165	165	165	165	165	165	165	165
1191	165	165	165	165	165	165	165	165	165
1192	165	165	165	165	165	165	165	165	165
1193	163	163	163	163	163	163	163	163	163
1194	170	170	170	170	170	170	170	170	170
1195	160	160	160	160	160	160	160	160	160
1196	167	167	167	167	167	167	167	167	167
1197	168	168	168	168	168	168	168	168	168
1198	164	164	164	164	164	164	164	164	164
1199	175	175	175	175	175	175	175	175	175
1200	170	170	170	170	170	170	170	170	170

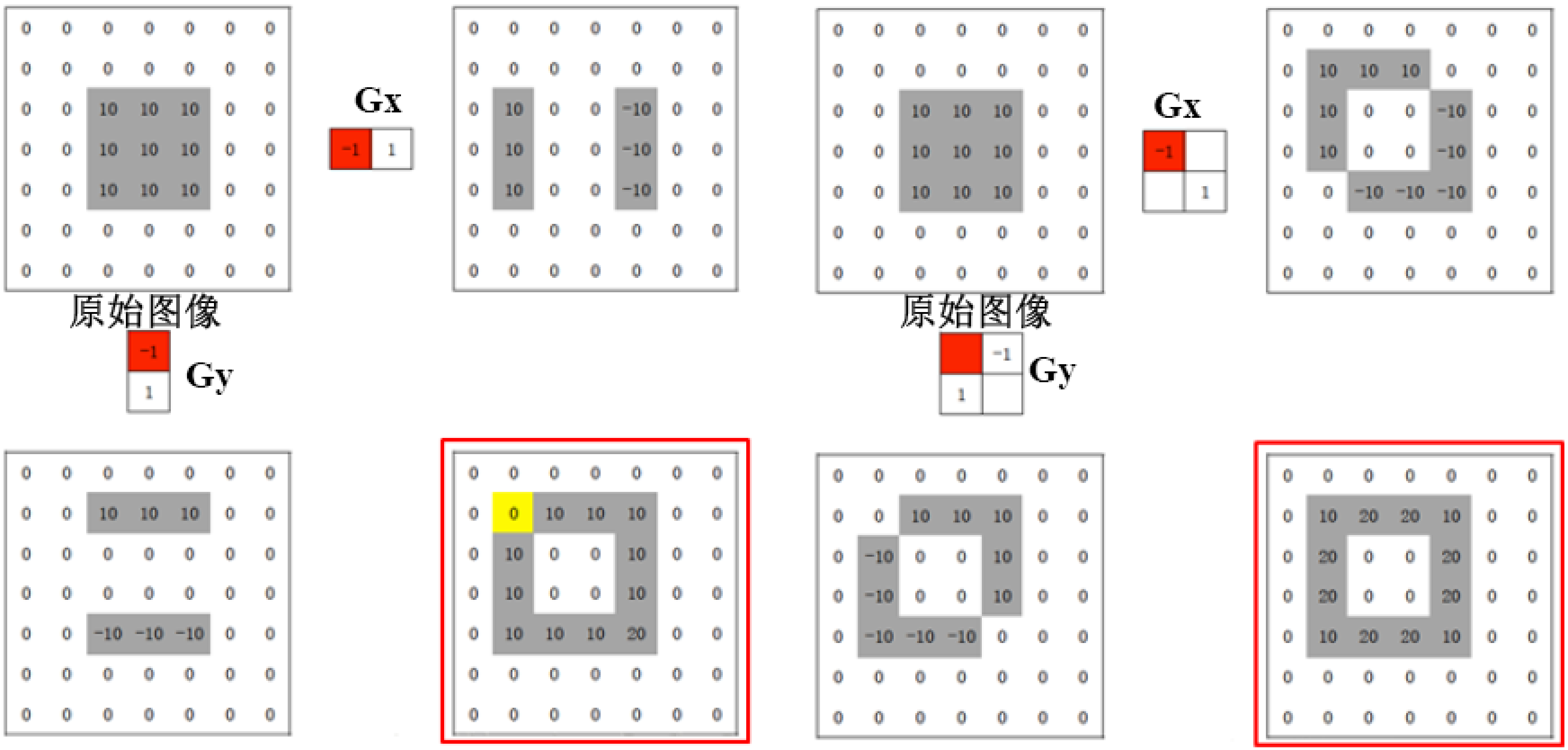
原始图像

	2	1593	1594	1595	1596	1597	1598	1599	1600
1180		0	0	0	0	0	0	0	0
1181		0	0	0	0	0	0	0	0
1182		0	0	0	0	0	0	0	0
1183		0	0	0	0	0	0	0	0
1184		0	0	0	0	0	0	0	0
1185		0	0	0	0	0	0	0	0
1186		0	0	0	0	0	0	0	0
1187		0	0	0	0	0	0	0	0
1188		0	0	0	0	0	0	0	0
1189		0	0	0	0	0	0	0	0
1190		0	0	0	0	0	0	0	0
1191		0	0	0	0	0	0	0	0
1192		0	0	0	0	0	0	0	0
1193		0	0	0	0	0	0	0	0
1194		0	0	0	0	0	0	0	0
1195		0	0	0	0	0	0	0	0
1196		0	0	0	0	0	0	0	0
1197		0	0	0	0	0	0	0	0
1198		0	0	0	0	0	0	0	0
1199		0	0	0	0	0	0	0	0
1200		0	0	0	0	0	0	0	0

梯度图像(0或1）

2.3.3 视觉媒体数字化

思考：为什么选择对角线方向差分？



对角线差分没有遗漏边缘候选点，向前差分遗漏黄色像素；

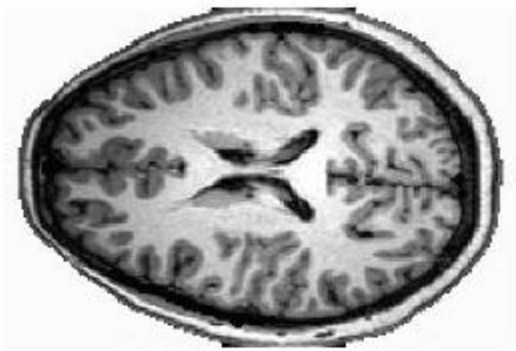
2.3.3 视觉媒体数字化

- 其他算子

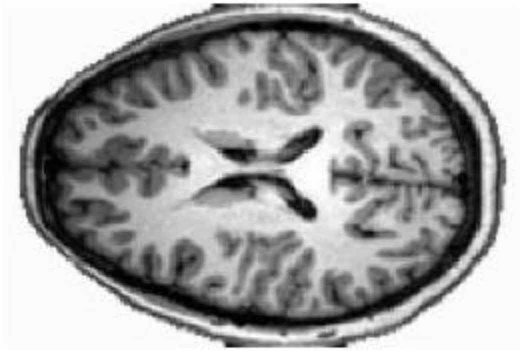
均值滤波：用邻域(3*3)全体像素的平均值来代替原来像素值，降低了图像的“尖锐”变化。

均值滤波算子：

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$



原始图像



滤波结果

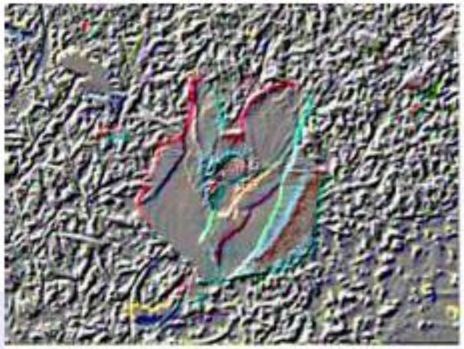
2.3.3 视觉媒体数字化

原始图像

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -7 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

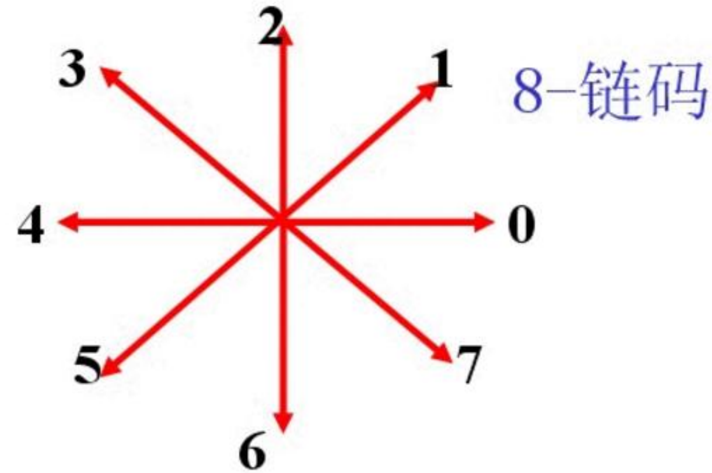


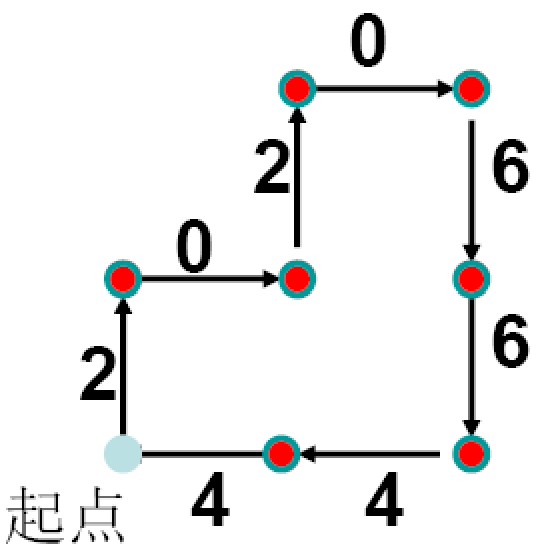
2.3.3 视觉媒体数字化

- 相关和卷积的优势
  - **并行性**：每个像素计算独立，可以同时、并行计算，容易移植到GPU计算;
  - **易修改**：模版或卷积核

2.3.3 视觉媒体数字化

- 图像边界表示
  - 用链码表示：**20206644**





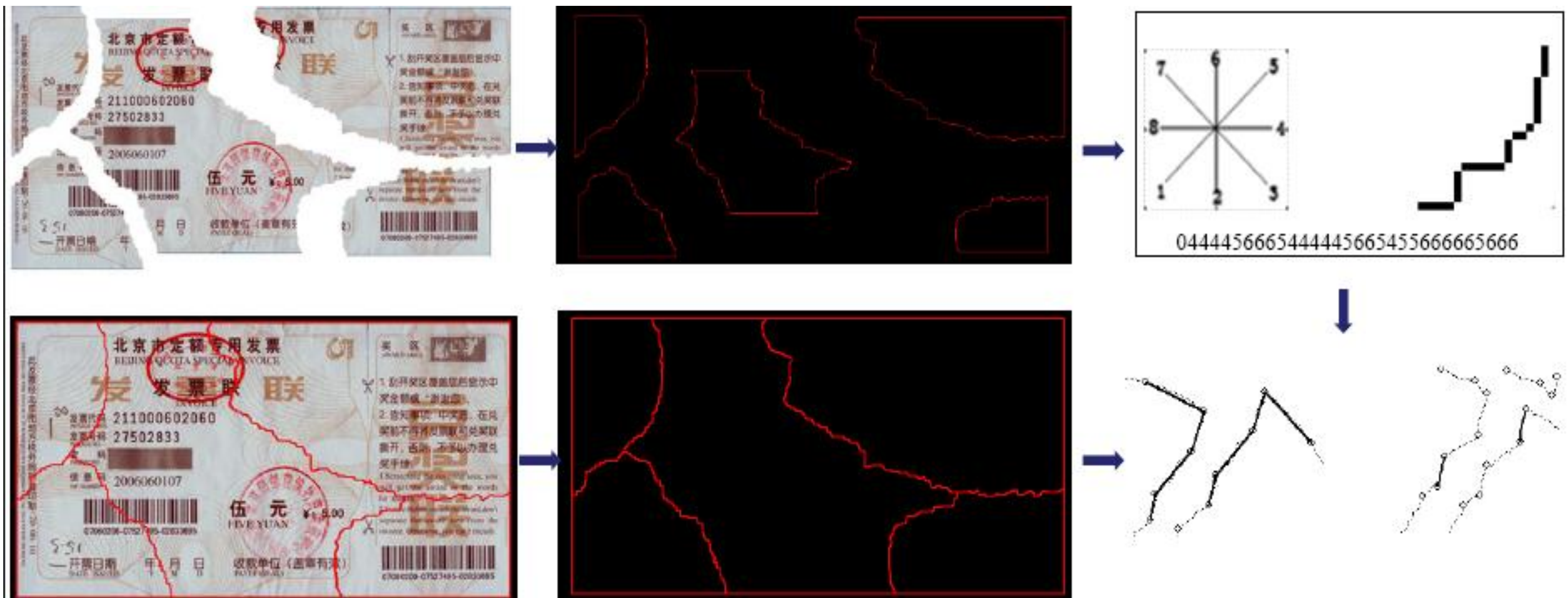
–这个结果与起点有关，为避免这个问题，用循环码表示：**02066442**

–首尾相连循环形成的自然数最小

2.3.3 视觉媒体数字化

- 图像边缘应用

新闻：小朋友误剪碎4000元现金 我行员工拼接帮复原



2.3.3 视觉媒体数字化

- 图像处理技术
  - （3）基于区域的分割

原理：将具有相似性质的像素合起来构成区域

种子填充算法

➢ 步骤1：先对每个需要分割的区域找一个种子像素作为生长起点；

➢ 步骤2：将种子像素周围邻域中与种子像素有相同或相似性质的像素（根据某种事先确定的生长或相似准则来判定）合并到种子像素所在的区域中；

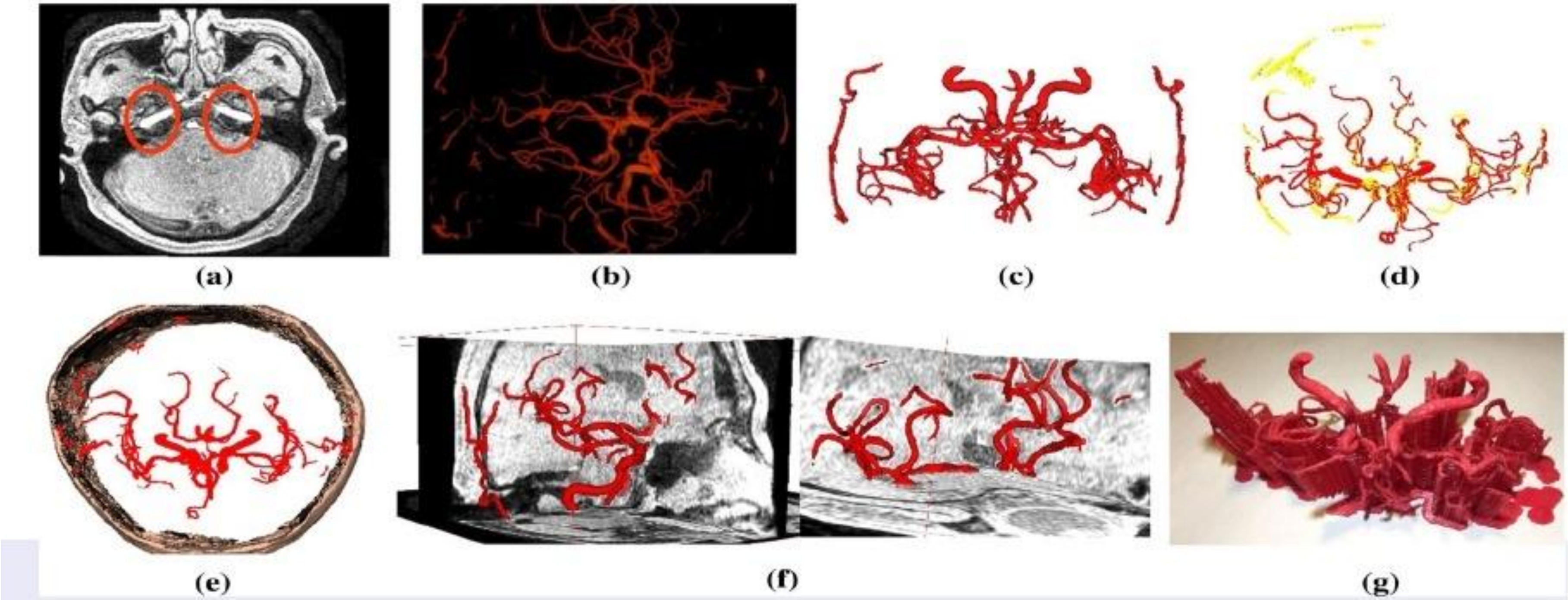
➢ 步骤3：将新像素当作新的种子像素继续进行上面的过程，直到再没有满足条件的像素结束。

2.3.3 视觉媒体数字化

- 关键

  1. 选择或确定一组能正确代表所需区域的种子像素

  先验知识、聚类中心等，如脑血管模型



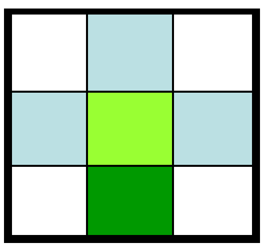
2.3.3 视觉媒体数字化

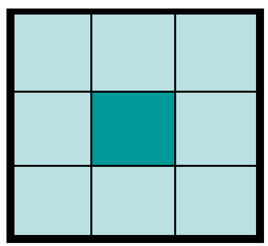
- 关键

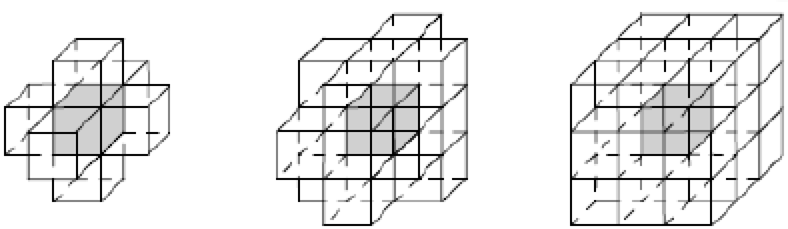
  2. 确定在生长过程中能将相邻像素包括进来的准则

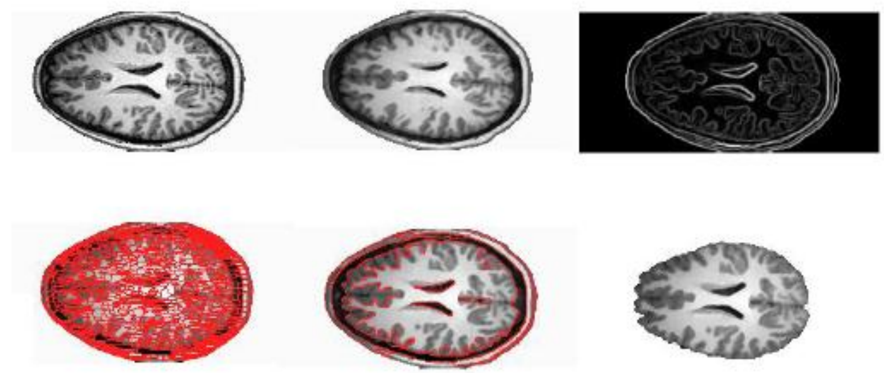
  连通性和邻近性等：4邻域、8邻域像素或邻接区域

  灰度相似性、梯度相似性、统计相似性：阈值


4连通


8连通





分水岭算法：设置每个小区域边缘的梯度信息和灰度信息阈值

2.3.3 视觉媒体数字化

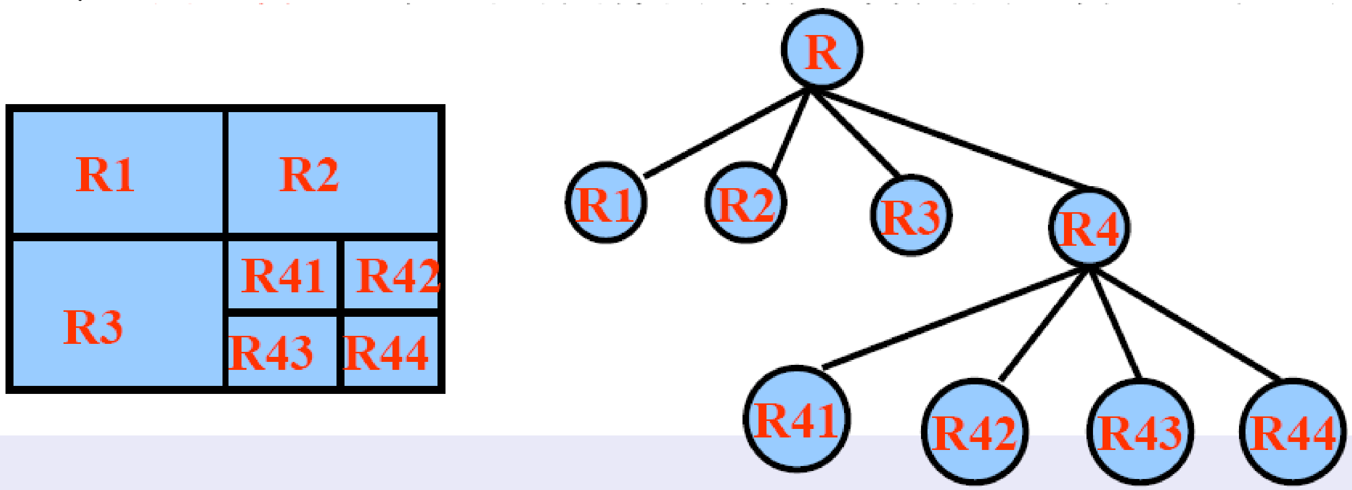
- 关键

3. 制定让生长过程停止的条件或准则

（a）遍历完每个像素

局限：每个像素运算复杂度过高，则遍历整个图像时间长

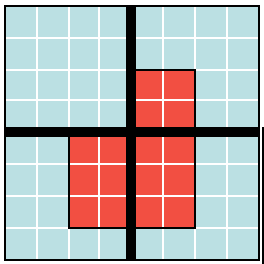
（b）四叉树：递归的将图像四等分，并用四叉树结构表示



2.3.3 视觉媒体数字化

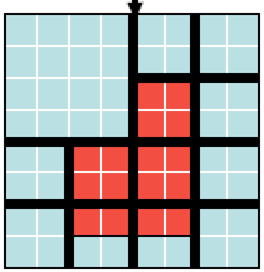
示例

图中红色区域为目标，其它区域为背景，如何将目标区域表示为四叉树？



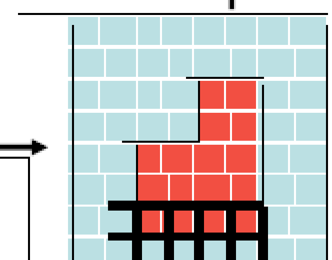
(a)

整个图像先将其分裂成如图：(a)所示的四个正方形区域，由于左上角区域满足条件不必继续分裂，其它三个区域继续分裂而得到(b)

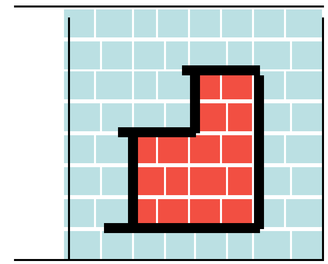


(b)

此时除包括目标下部的两个子区域外，其它区域都可分别按目标和背景合并。对下面的两个子区域继续分裂可得到(c)



(c)



(d)

此时所有区域都已满足条件，目标图像用四叉树表示。

2.3.3 视觉媒体数字化

- 图像处理技术
  - 灰度图像分割方法

    （1）基于阈值的分割：通过阈值对不同物体进行分割

    （2）基于边缘的分割：通过直接确定区域间的边界来实现分割

    （3）基于区域的分割：把各像素划归到各个物体或区域中

思考：彩色图像分割？

2.3.3 视觉媒体数字化

- 例如
  - 彩色图像转为灰度图像

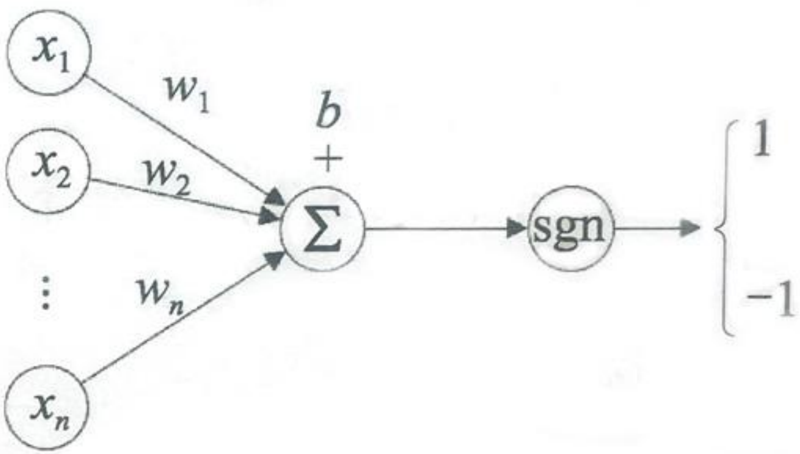
```
A=imread('123.jpg');
C=zeros(size(A,1),size(A,2));
for i = 1:rows
for j= 1:cols
    r =  A(i, j, 1);
    g = A(i, j, 2);
    b = A(i, j, 3);
    C(i,j)=0.2989 * R + 0.5870 * G + 0.1140 * B ;
end
end
```

24位：768KB

8位：256KB

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
  - 神经网络
    - 一种模仿生物神经网络结构和功能的数学模型或计算模型。由大量的节点（或称为神经元）相互之间连接构成。



基础结构：感知机示意图

$$f(x_1, x_2, \cdots, x_n) = \begin{cases} 1, & w_1x_1 + w_2x_2 + \cdots x_nw_n + b > 0 \\ -1, & w_1x_1 + w_2x_2 + \cdots x_nw_n + b \leqslant 0 \end{cases}$$

2.3.3 视觉媒体数字化

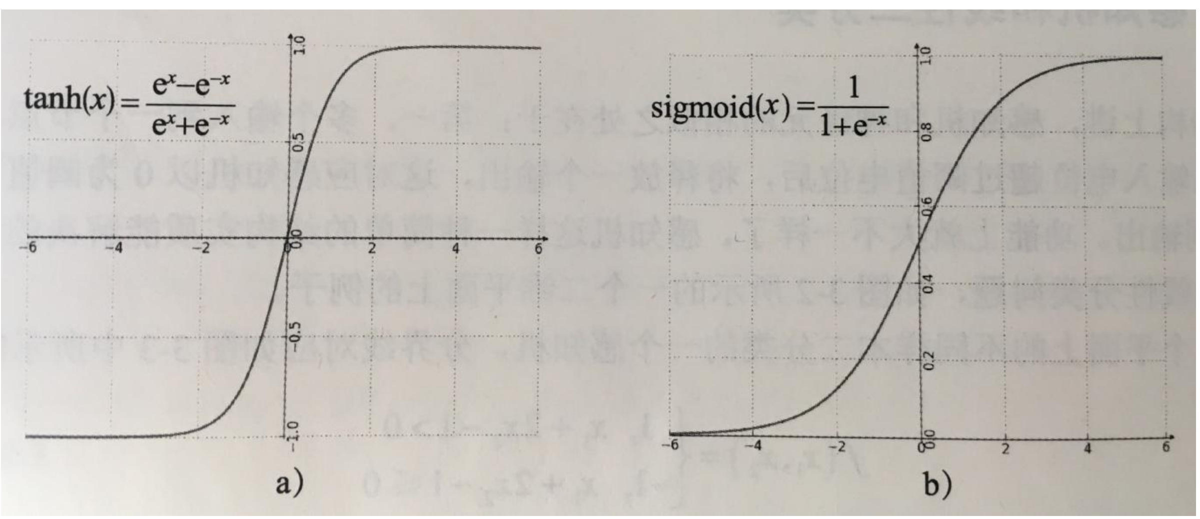
- 新技术：卷积神经网络
  - 神经网络
    - 一般的：输入向量做线性变换，加上一个位移，然后再做非线性变换
    - x表示向量$(x_1,x_2,\ldots,x_n)$，w表示对应系数的向量，表示为公式：

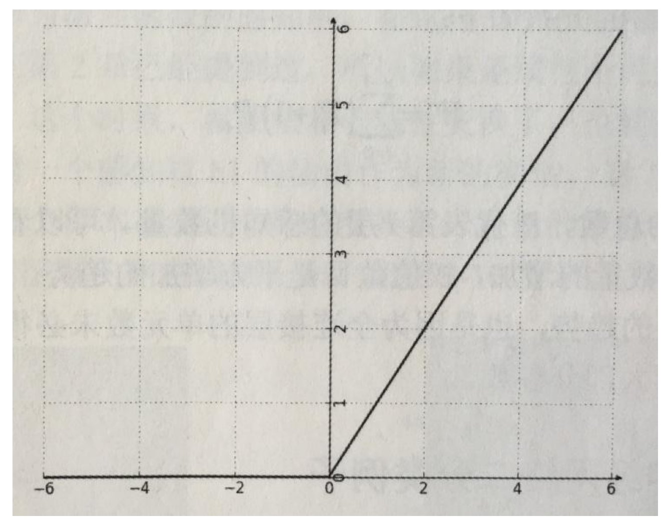
$$f(x) = g(w \bullet x + b)$$

    - g表示一个非线性变换（激活函数），w和x二者点积

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
  - 神经网络
    - 激活函数
    - 比如：sgn函数、tanh函数、sigmoid函数、ReLU函数



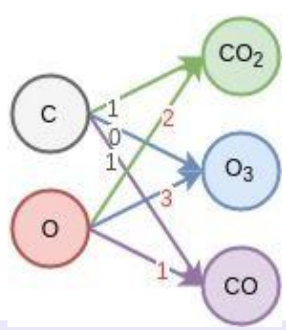


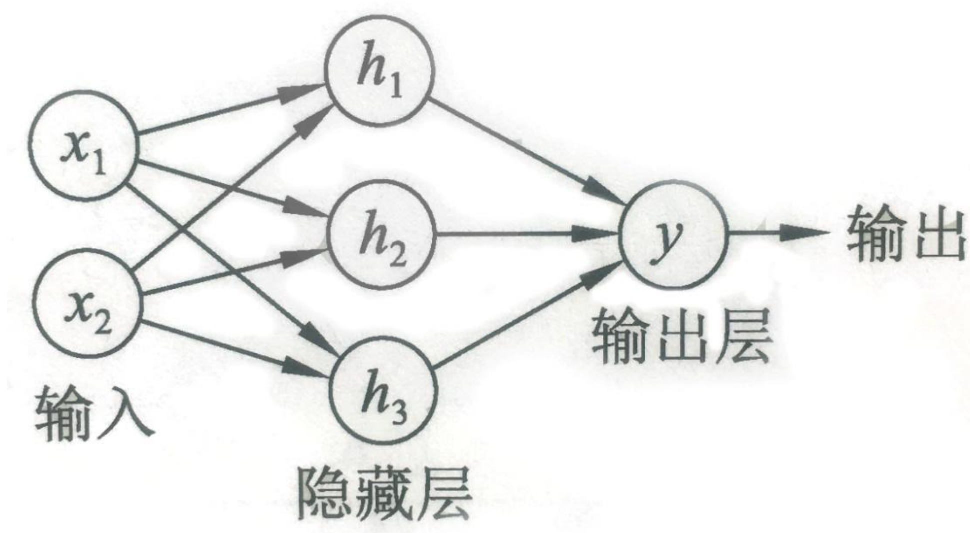
2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
  - 神经网络
    - **分层结构**
    - 如：输入层和3个感知机相连，3个感知机再和一个感知机相连

$$\vec{y} = a(W \cdot \vec{x} + b)$$

- $\vec{x}$ 是输入向量
- $\vec{y}$ 是输出向量
- $\vec{b}$ 是偏移向量
- W是权重矩阵
- a()是激活函数

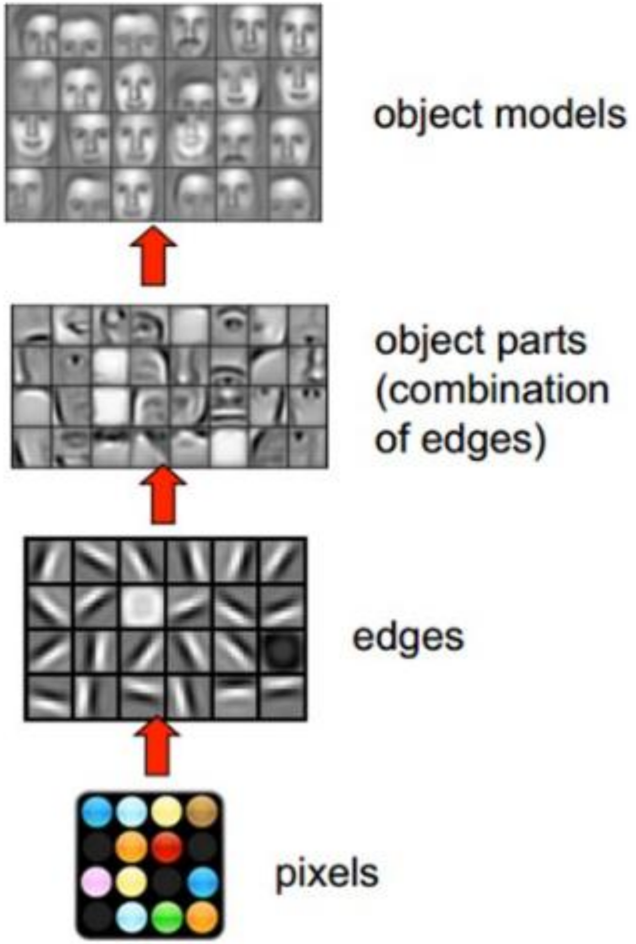


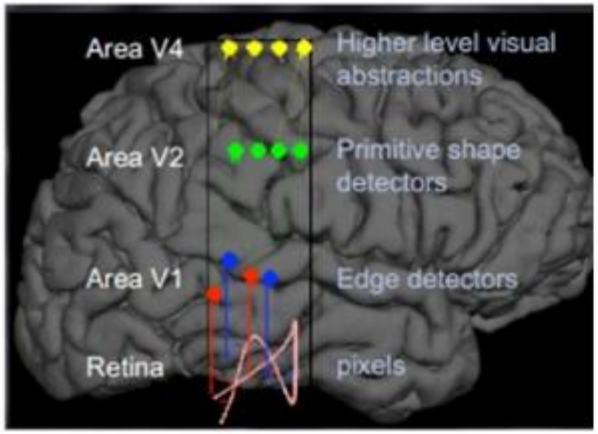


一个简单的神经网络的例子

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

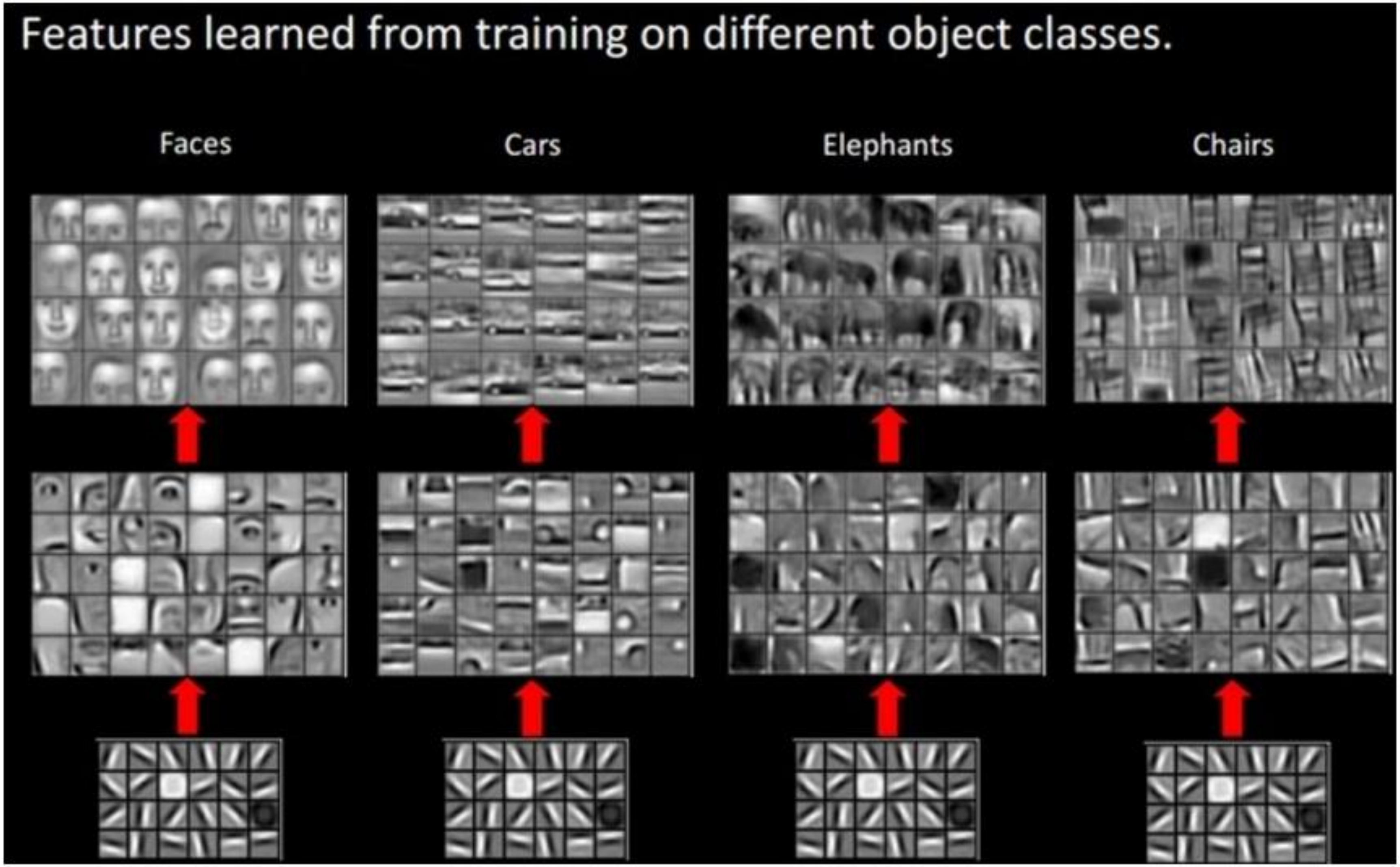




可视皮层是分级的：
✓摄入原始信号（瞳孔摄入像素 Pixels）
✓做初步处理（大脑皮层某些细胞发现边缘和方向）
✓抽象（大脑判定，眼前的物体的形状，是圆形的）
✓进一步抽象（大脑进一步判定该物体是只气球）

2.3.3 视觉媒体数字化

- 新技术：卷积神经网络

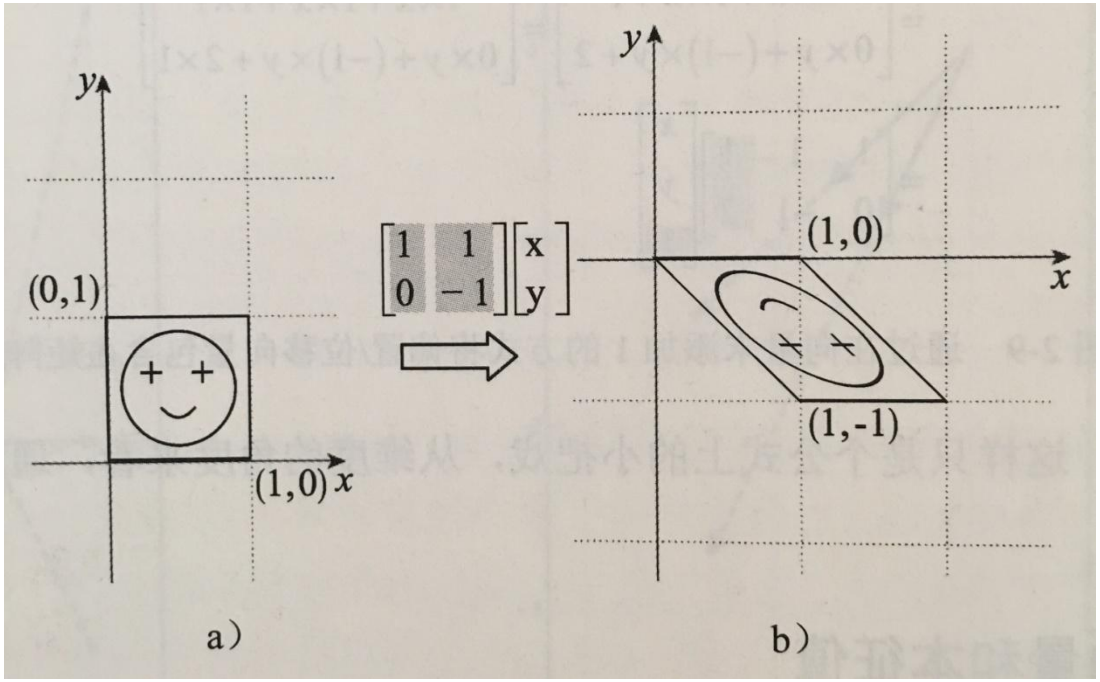


2.3.3 视觉媒体数字化

- 新技术：卷积神经网络
  - 神经网络

$$\vec{y} = a(W \cdot \vec{x} + b)$$

    - W：缩放、旋转、维度
    - b：平移
    - a：扭曲



2.3.3 视觉媒体数字化

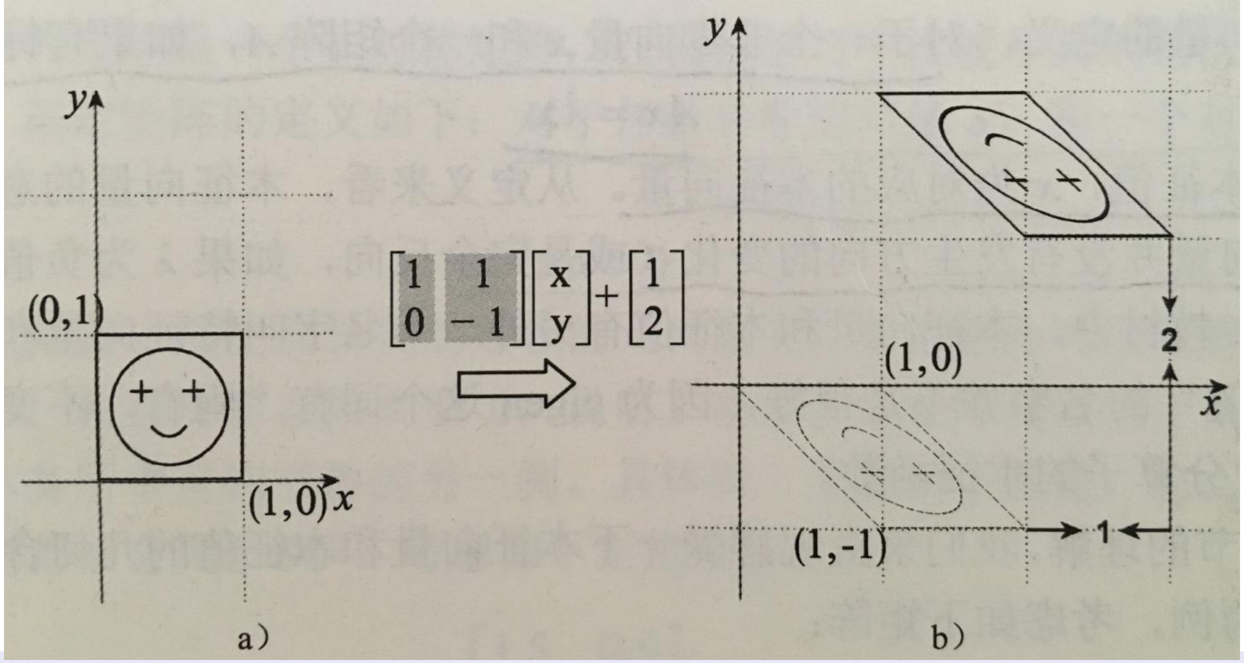
- 新技术：卷积神经网络
  - 神经网络

$$\vec{y} = a(W \cdot \vec{x} + b)$$

➢ **W**：缩放、旋转、维度

➢ **b**：平移

➢ **a**：扭曲



2.3.3 视觉媒体数字化

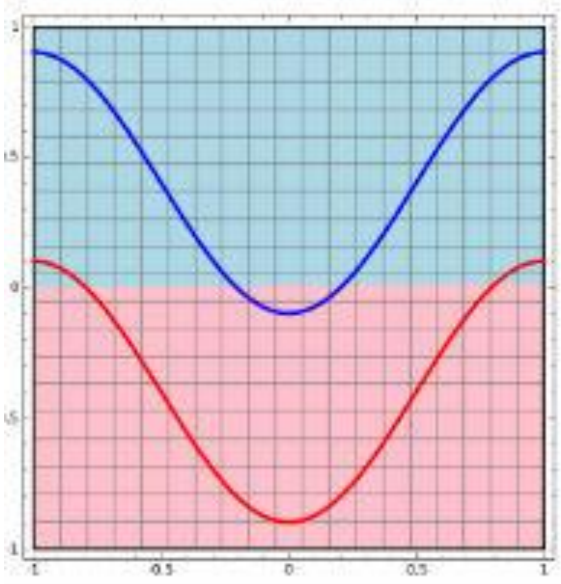
- 新技术：卷积神经网络
  - 神经网络

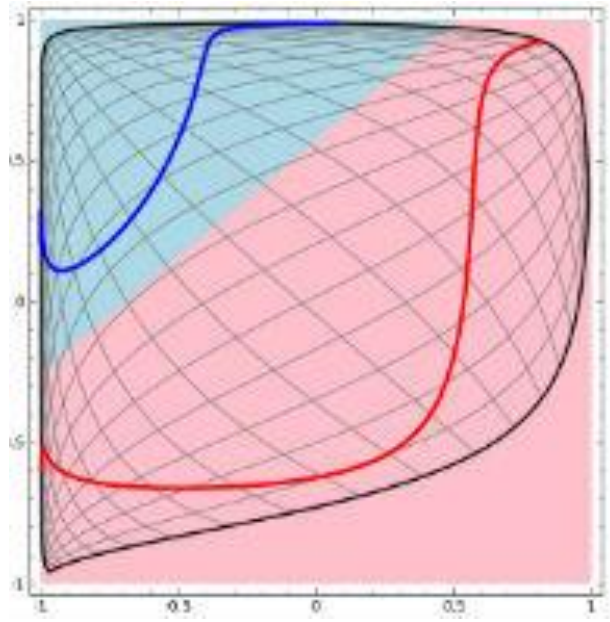
$$\vec{y} = a(W \cdot \vec{x} + b)$$

➢ **W**：缩放、旋转、维度

➢ **b**：平移

➢ **a**：扭曲





2.3.3 视觉媒体数字化

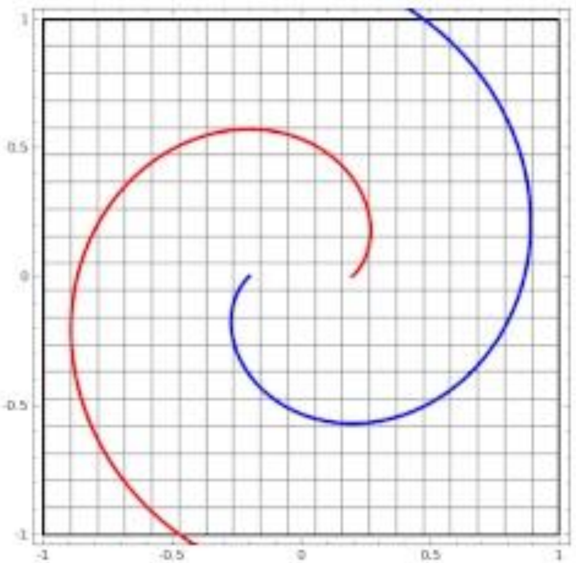
- 新技术： 卷积神经网络
  - 神经网络

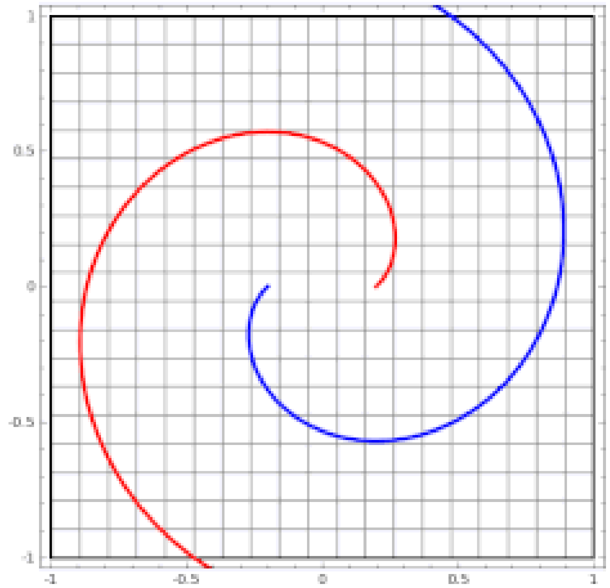
$$\vec{y} = a(W \cdot \vec{x} + b)$$

➢ **W**： 缩放、旋转、维度

➢ **b**： 平移

➢ **a**： 扭曲





2.3.3 视觉媒体数字化

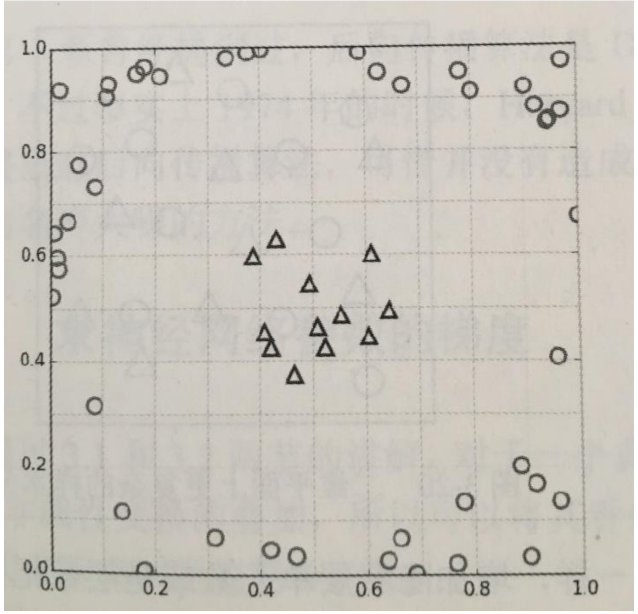
- 新技术：卷积神经网络
  - 神经网络

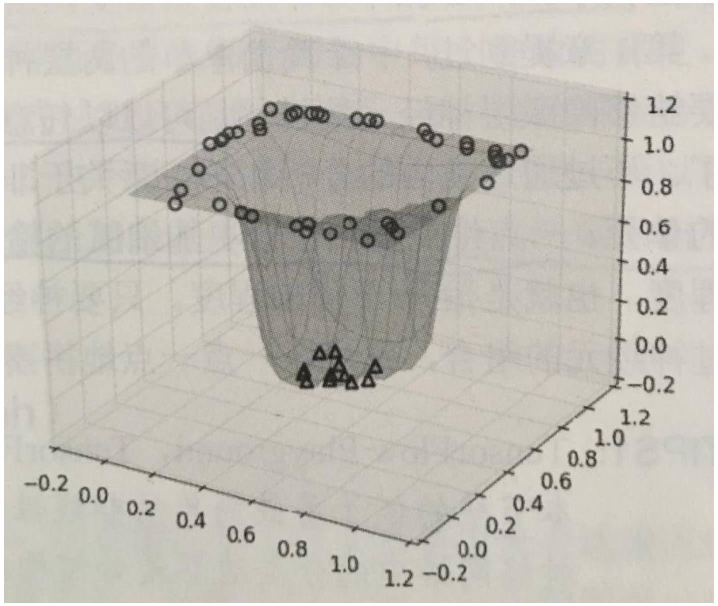
$$\vec{y} = a(W \cdot \vec{x} + b)$$

➢ ***W***：缩放、旋转、维度

➢ ***b***：平移

➢ ***a***：扭曲





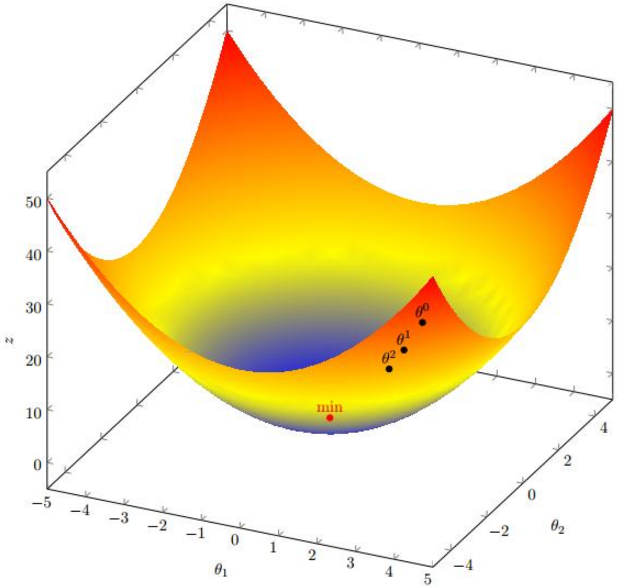
2.3.3 视觉媒体数字化

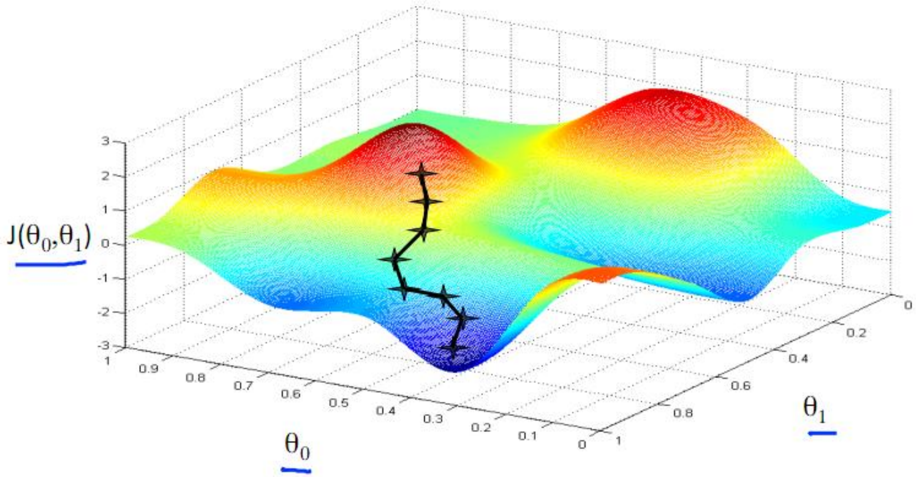
- 新技术：卷积神经网络
  - 训练神经网络
    - 如何学习每一层的权重矩阵*W*
    - 损失函数或目标函数
      - 用于衡量预测值和真实值的差异的方程
      - 比如：均方差MSE

$$MSE = (\sum_{i=1}^{n}(Y - f(X))^2)/N$$

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

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

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

$$=\mathbf{grad}\, f(x_0,y_0)\cdot\boldsymbol{e}_l=|\mathbf{grad}\, f(x_0,y_0)|\cos\theta,$$

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

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

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