法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：大数据分析挖掘
  - 新浪微博：ChinaHadoop

第七讲

图像数据处理及分析

--梁斌

目录

- 计算机视觉
- 图像处理工具 -- scikit-image
- scikit-image基本操作
- scikit-image图像数据处理
- 常用的图像特征方法
- 分类预测模型：人工神经网络
- 实战案例：电影口碑与海报图像的相关性分析

目录

计算机视觉

什么是计算机视觉

- 从图像和视频中提取数值或符号信息的计算系统
- 让计算机能够和人类一样“看到并理解”图像
- 主要应用

图像理解—高级视觉
识别
鉴别
监测

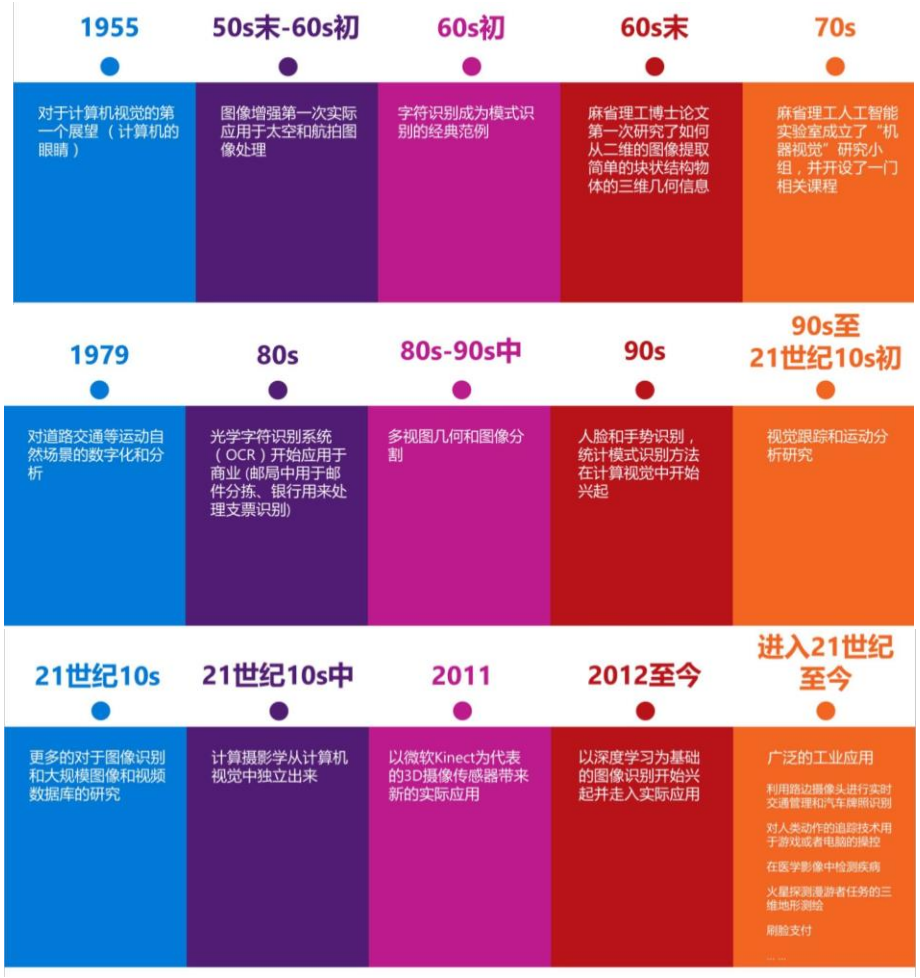
图像分析—中级视觉
运动
分割
跟踪
多视图几何

图像处理—低级视觉
线性滤波
边缘检测
纹理

图像获取—成像
相机模型
相机标定
辐射测定
颜色

计算机视觉

CV发展历史



目录

scikit-image

scikit-image

- 是Python中用来进行图像处理的常用包之一
- 图像数据通过NumPy中的ndarray表示
- 通常和NumPy、SciPy共同使用进行图像数据的处理
- 安装
  - pip install -U scikit-image

目录

scikit-image基本操作

skimage的图像数据

- skimage中的图像数据是由NumPy的多维数组（ndarray）表示的
- 由skimage加载的图像数据可以调用其他常用的包进行处理和计算，如matplotlib，scipy等

数据类型和像素值

- CV中图像的像素值通常有以下两种处理范围
  1. 0 - 255, 0：黑色，255：白色
  2. 0 - 1,　0：黑色，1　：白色
- skimage支持以上两种像素范围，至于如何选择是根据数组的dtype决定的
  1. float-> 0-1
  2. unsigned bytes -> 0-255
  3. unsigned 16-bit integers -> 0-65535

示例代码： 01_scikit_image_basic.ipynb

scikit-image基本操作

数据类型和像素值

- 像素值数据类型转换
  - img_as_float, img_as_ubyte
- 推荐使用float，skimage包内部大多数用的是float类型，即像素值是0-1

显示图像

- 通过matplotlib, plt.imshow()，可以指定不同的color map

图像I/O

- 加载图像，skiamge.io.imread()
- 同时加载多个图像，skimage.io.imread_collection()
- 保存图像，skimage.io.imsave()

示例代码： 01_scikit_image_basic.ipynb

目录

scikit-image图像数据处理

图像数据

- 图像数据是多维数组，前两维表示了图像的高、宽第三维表示图像的通道个数，比如RGB，第三个维度为3，因为有三个通道；灰度图像没有第三个维度
- 分割和索引
  - 像操作ndarray一样操作即可

色彩空间, RGB, HSV, Gray...

- RGB转Gray, skimage.color.rgb2gray()

颜色直方图

- 直方图是一种能快速描述图像整体像素值分布的统计信息，skimage.exposure.histogram
  - 如：可以根据直方图选定阈值用于调节图像对比度

示例代码： 02_image_proc.ipynb

scikit-image图像数据处理

对比度

- 增强图像数据的对比度有利于特征的提取，不论是从肉眼还是算法来看都有帮助
- 更改对比度范围

  skimage.exposure.rescale_intensity(image, in_range=(min, max))

  原图像数据中，小于min的像素值设为0，大于max的像素值设为255
- 直方图均衡化

  自动调整图像的对比度 skimage.exposure.equalize_hist(image)

  [注意]均衡化后的图像数据范围是[0, 1]

示例代码： 02_image_proc.ipynb

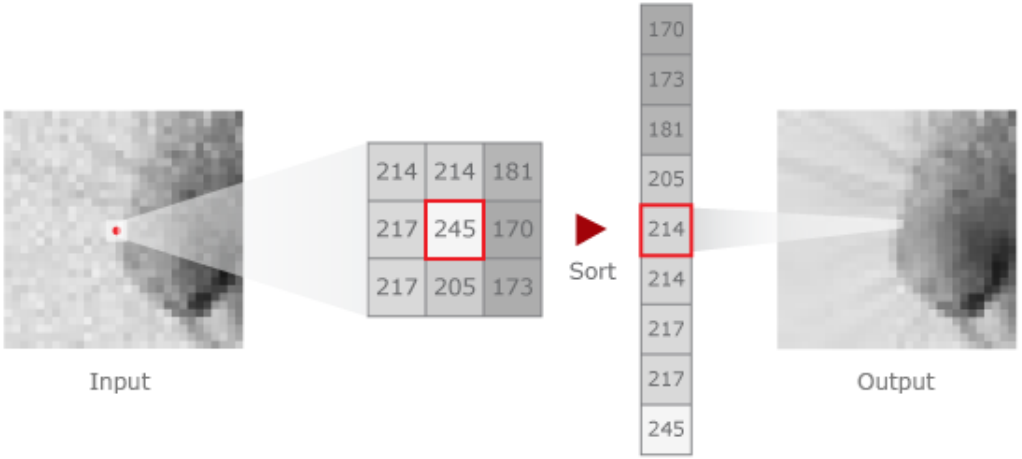
scikit-image图像数据处理

图像滤波

- 滤波是处理图像数据的常用基础操作
- 滤波操作可以去除图像中的噪声点，由此增强图像的特征

中值滤波

skimage.filters.rank.median



高斯滤波

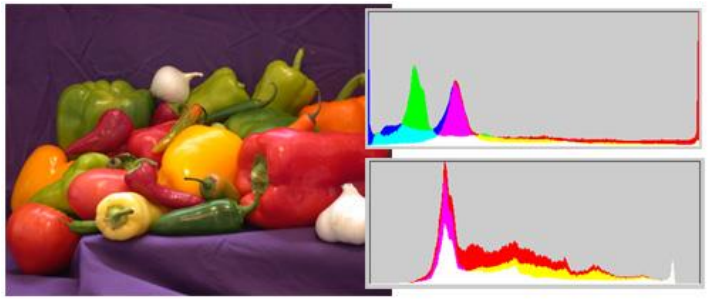
skimage.filters.gaussian

示例代码： 02_image_proc.ipynb

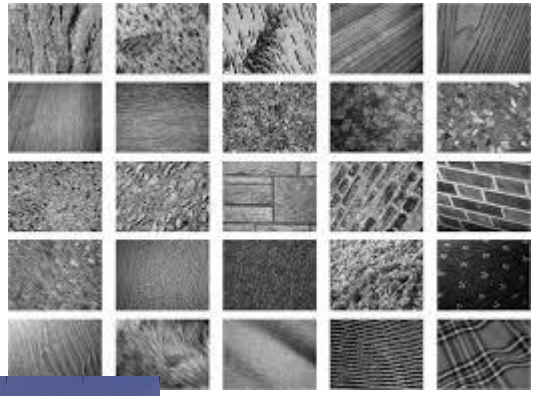
目录

常用的图像特征

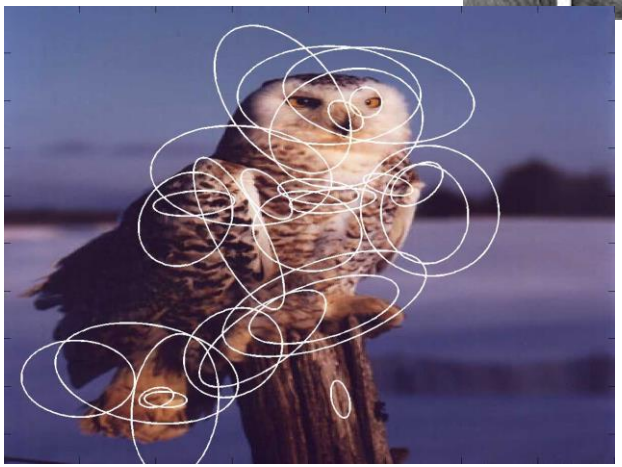
1. **颜色特征**



2. **纹理特征**



3. **形状特征**

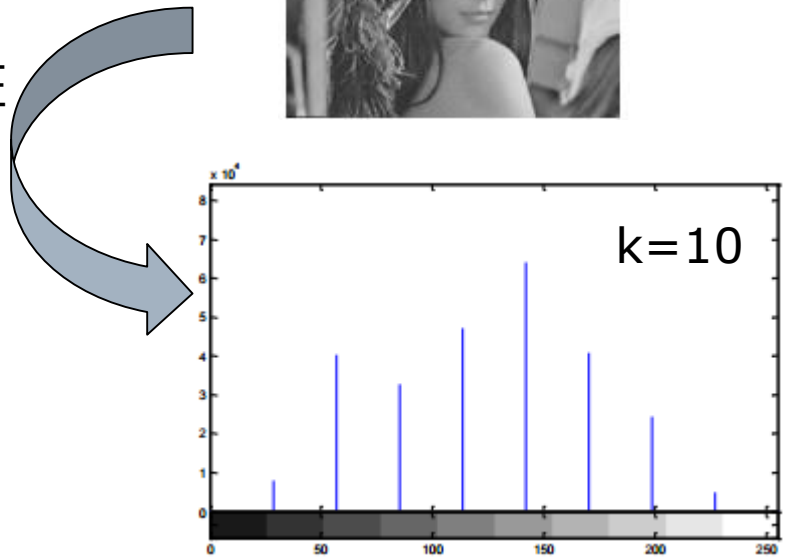


4. **skimage中的特征方法**
   - skimage.feature
   - http://scikit-image.org/docs/dev/api/skimage.feature.html

常用的图像特征

颜色特征

- 图像检索中应用最为广泛的视觉特征
  - 颜色直方图，如：



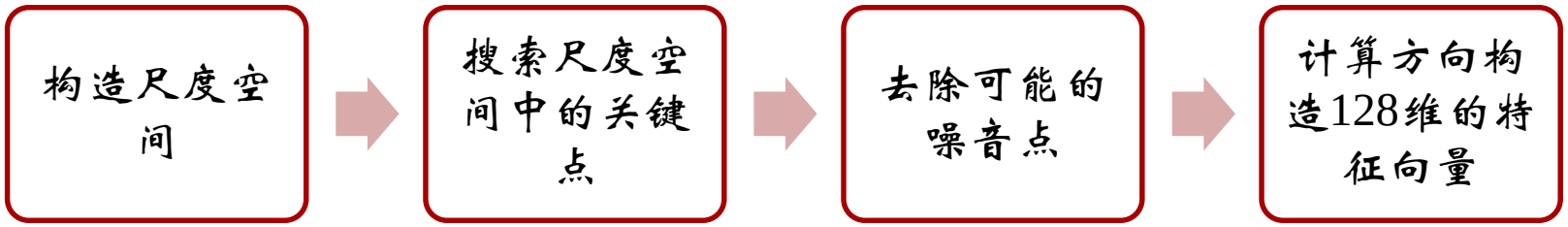
  - 从512x512的灰度图像中提取维度为k的颜色直方图，就是讲256种灰度颜色分为k个区间，然后计算每个区间中像素点总数为多少。
  - 如果k=20，则得到的20维直方图向量维 (0, 0, 750, 14613, 24233, 11126, 12943, 19345, 22012, 23122, 27978, 33309, 25312, 15992, 9563, 12967, 8045, 828, 6, 0)

示例代码： 03_image_feat.ipynb

常用的图像特征

图像形状特征

- 形状特征的表达必须以对图像中物体或区域的分割为基础
- SIFT (Scale-invariant feature transform)，在尺度空间中所提取的图像局部特征点。SIFT特征点提取较为方便，提取速度较快，对于图像的缩放等变换比较鲁棒，因此得到了广泛的应用。



示例代码： 03_image_feat.ipynb

常用的图像特征

图像形状特征

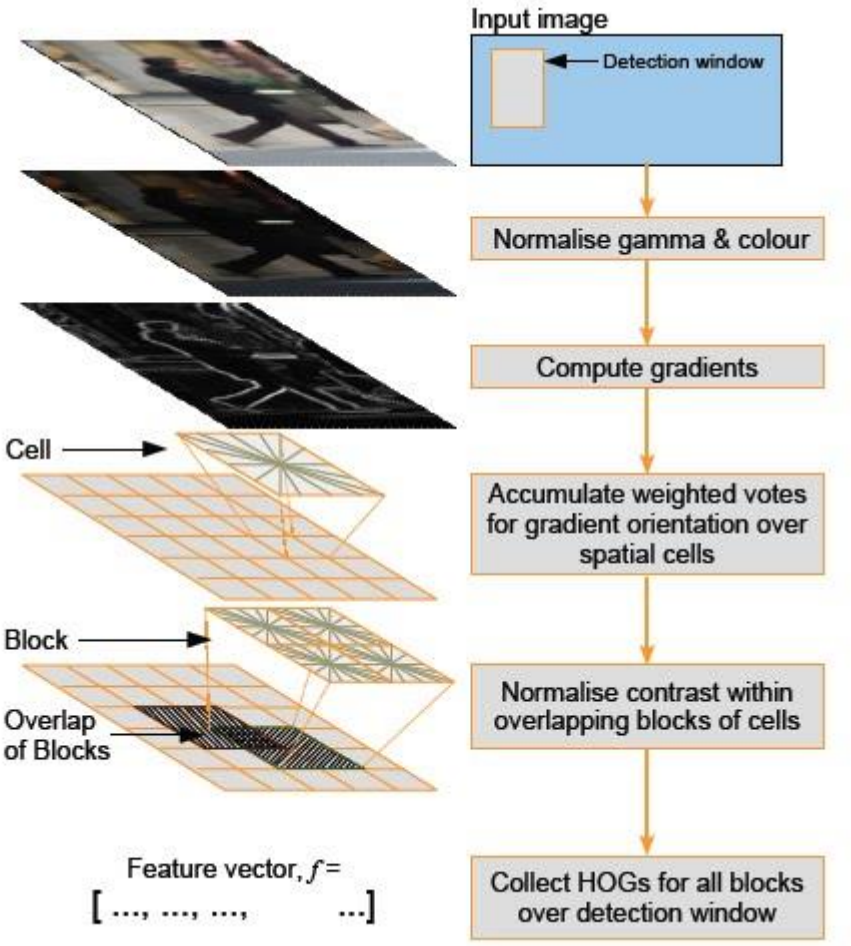
- HOG (Histogram of Oriented Gradient)，用于检测物体的特征描述，通过计算和统计图像局部区域的梯度方向直方图来构建特征
- 由于HOG是在图像的局部方格单元上操作，所以它对图像几何的和光学的形变都能保持很好的不变性
- 在粗的空域抽样、精细的方向抽样以及较强的局部光学归一化等条件下，只要行人大体上能够保持直立的姿势，可以容许行人有一些细微的肢体动作，这些细微的动作可以被忽略而不影响检测效果
- HOG特征特别适合于做图像中的人体检测

示例代码： 03_image_feat.ipynb

常用的图像特征

图像形状特征

- HOG

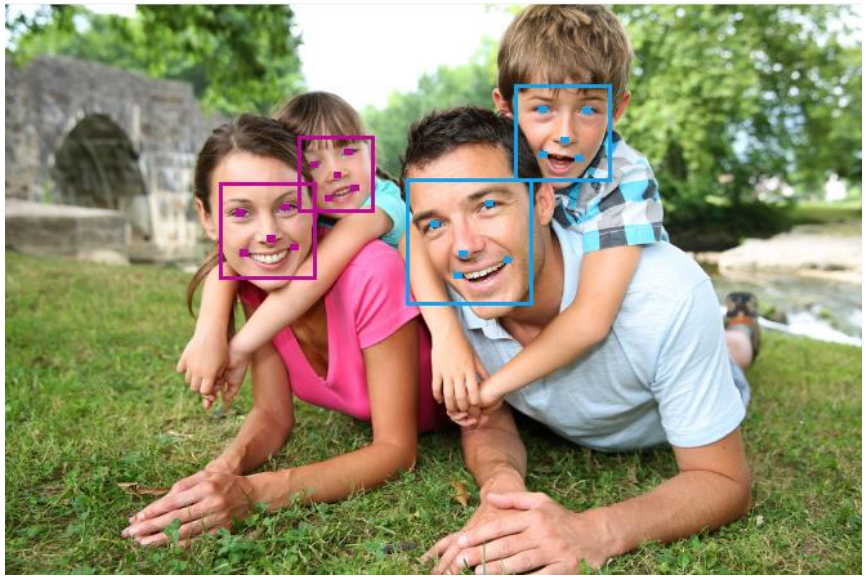


示例代码： 03_image_feat.ipynb

常用的图像特征

人脸检测

- 借助微软的 Cognitive-Face-Python
- 访问https://www.microsoft.com/cognitive-services/en-us/face-api获取服务key
- 安装模块 pip install cognitive_face
- 通过key值调用“人脸检测”云服务



示例代码： 03_image_feat.ipynb

目录

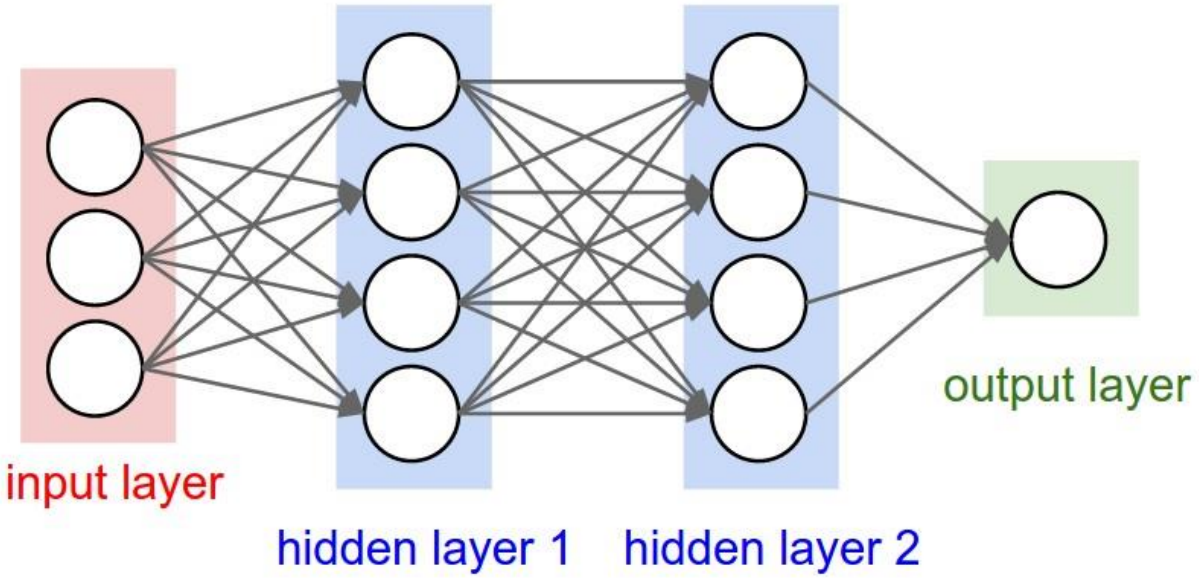
人工神经网络

背景

- 以人脑中的神经网络为启发，有多个版本
- 最著名的是1980年的backpropagation

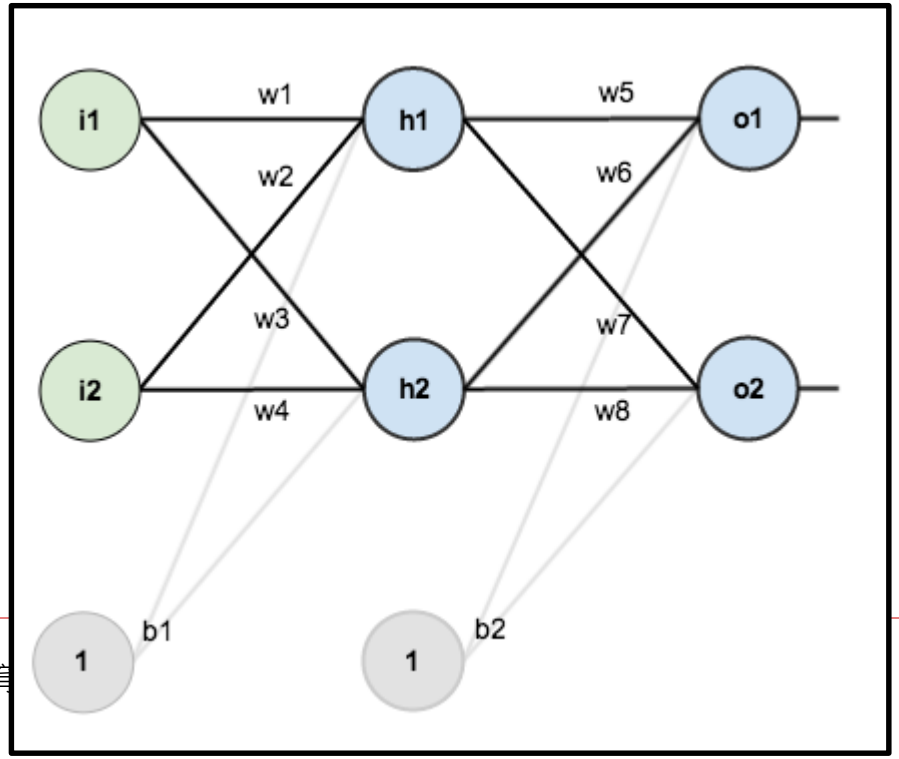
神经网络基本组成

- 输入层（input layer），隐藏层（hidden layers），输出层（output layer）



人工神经网络

- 每层由神经元（neuron）或单元（unit）组成
- 输入层（input layer）是由训练集的样本特征向量传入
- 经过连接节点的权重（weight）传入下一层，上一层的输出是下一层的输入
  - 上一层中的加权求和，然后根据非线性方程转化为下一层的输入
- 下图是3层神经网络（通常不算输入层）



人工神经网络

- 对于多层神经网络，理论上，如果有足够多的隐藏层和足够多的训练样本，可以拟合出任意方程

设计神经网络结构

- 包括确定网络层数，每层的节点个数
- 特征向量在被传入输入层之前要进行标准化（normalize）到0-1间，为了加速训练过程
- 神经网络既可以用来解决分类问题，也可以解决回归问题
- 输出层的单元数是类别的个数
- 没有明确的规则来确定隐藏层的个数和节点的个数
    - 一般是通过交叉验证的方法确定

人工神经网络

Backpropagation算法

- 通过迭代处理训练集中的样本
- 对比经过神经网络后输出层的预测值和真实值
- 从输出层经过隐藏层反向传播到输入层，以最小化误差更新连接权重

算法过程

1. 初始化权重(weights)和偏置(bias): 随机初始化在-1到1的值，每个节点有一个偏向
2. 每个训练样本执行以下步骤：

   2.1 由输入层前向传播，根据初始化的参数及激活函数，计算出节点的输出，直到输出层

   2.2 反向传播计算误差，更新权重和偏置

人工神经网络

算法过程

- 每个训练样本执行以下步骤（续）：

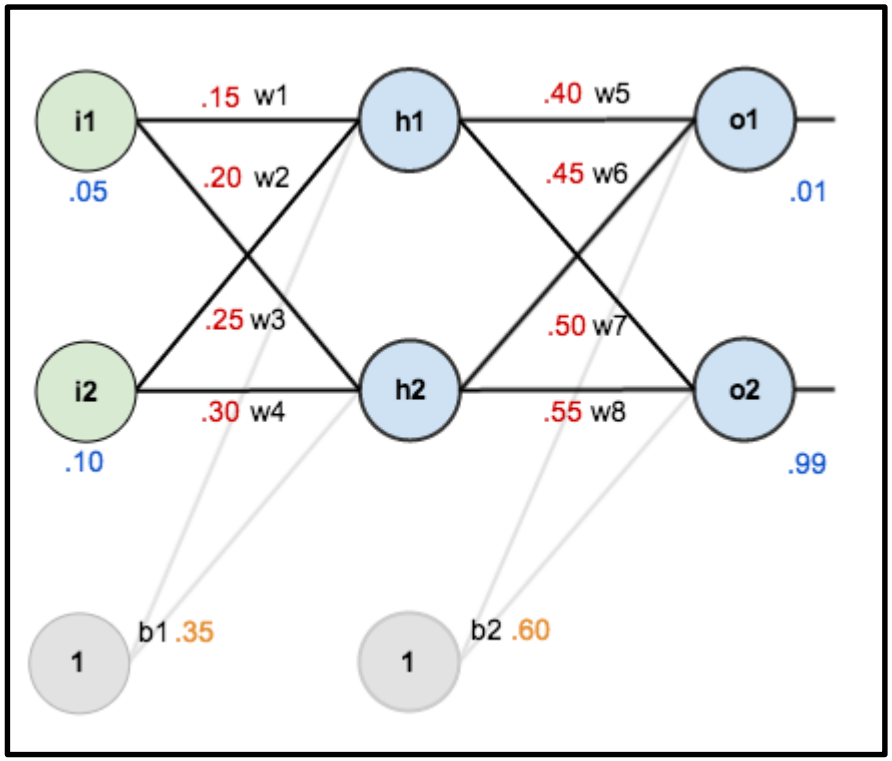
  3. 终止条件

  - 1) 权重的更新低于某个阈值,
  - 2) 预测的错误率低于某个阈值
  - 3) 达到预设的循环次数

人工神经网络

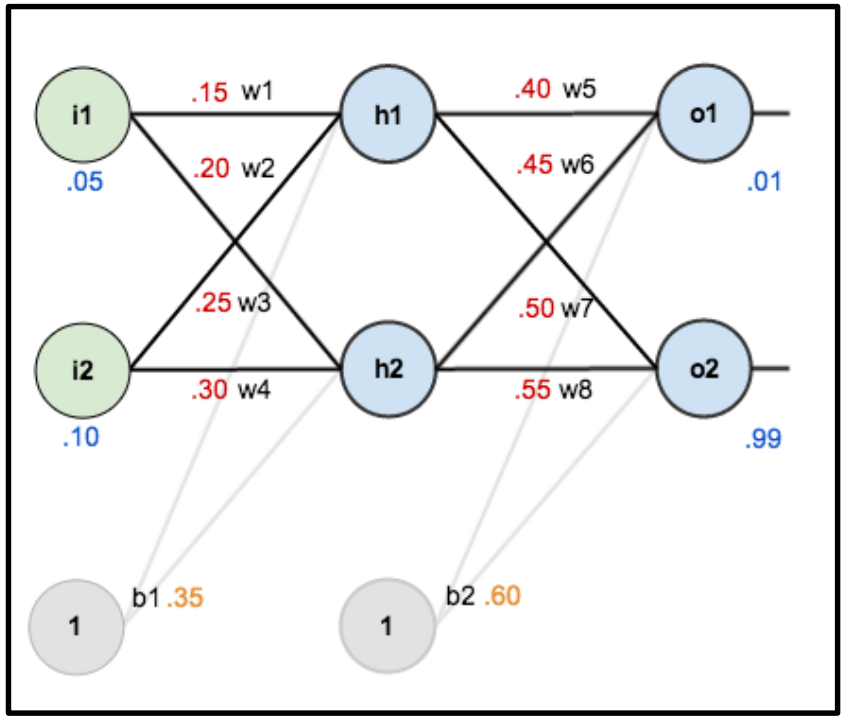
例子

1. 初始化权重(w)和偏置(b)，提供输入(i)输出(o)



人工神经网络

2.1. 前向计算



- 隐含层 h1：

$$net_{h1} = w1 \times i1 + w2 \times i2 + b1 \times 1$$

$$net_{h1} = 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35 \times 1$$
$$= 0.3775$$

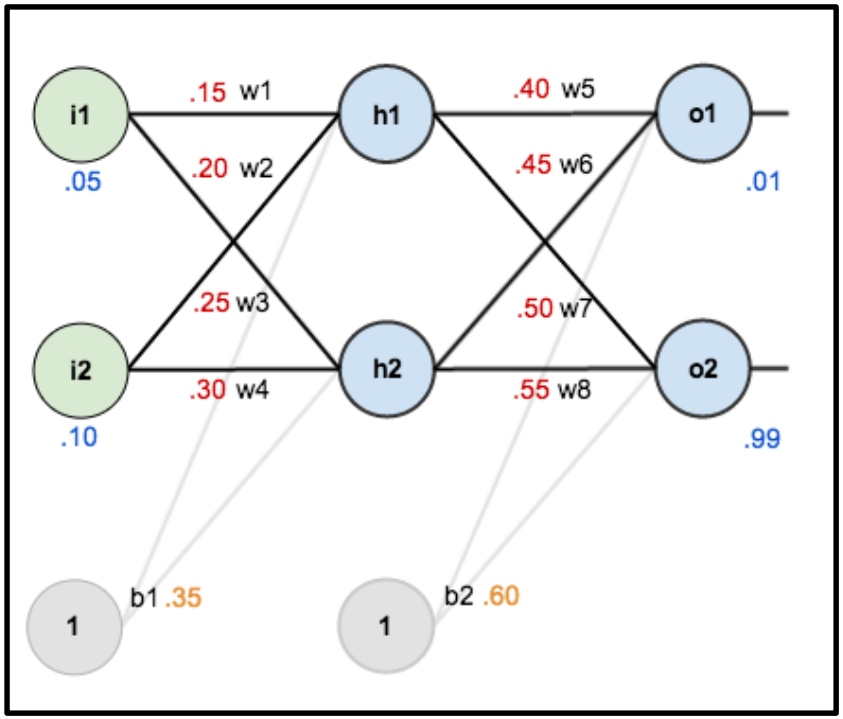
$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}}$$
$$= 0.593269992$$

- 隐含层 h2：

$$out_{h2} = 0.596884378$$

人工神经网络

2.1. 前向计算



- 输出层 o1：

$$net_{o1} = w5 \times out_{h1} + w6 \times out_{h2} + b2 \times 1$$

$$net_{o1} = 0.4 \times 0.593269992 + 0.45 \times 0.596884378 + 0.6 \times 1 = 1.105905967$$

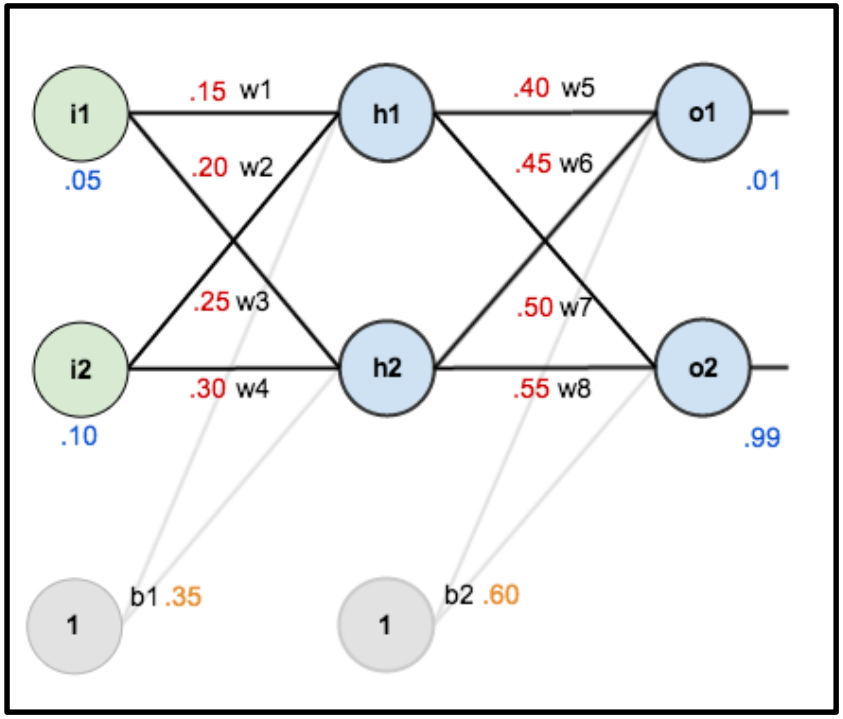
$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

- 输出层 o2：

$$out_{o2} = 0.772928465$$

人工神经网络

2.2. 计算整体误差



- 整体误差：$E_{total} = E_{o1} + E_{o2}$

$= 0.274811083 + 0.023560026$

$= 0.298371109$

- 计算每个输出神经元的误差，这些误差的和就是整体误差：

$$E_{total} = \sum \frac{1}{2}(target - out)^2$$

- O1的误差：

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2$$

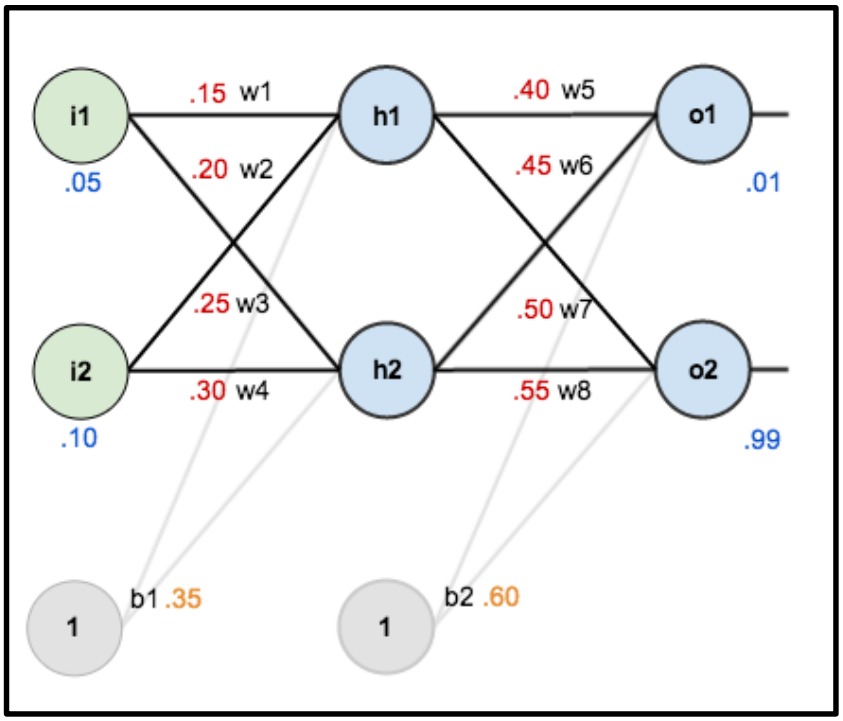
$$= \frac{1}{2}(0.01 - 0.75136507)^2$$

$$= 0.274811083$$

- O2的误差：$E_{o2} = 0.023560026$

人工神经网络

2.2. 反向传播

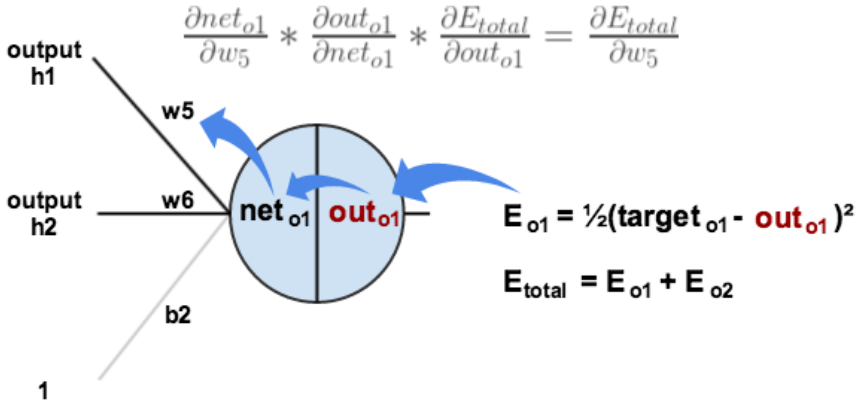


- 输出层

对于w5，需知道w5与总体误差的关系，即

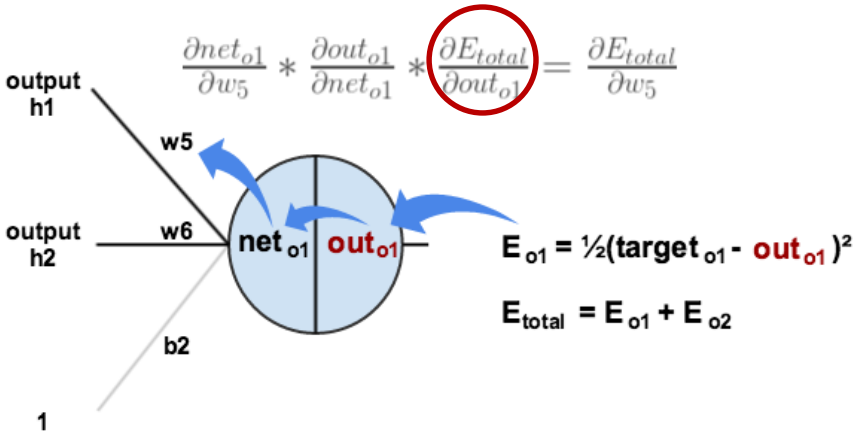
$\frac{\partial E_{total}}{\partial w5}$ 根据链式法则，有

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w5}$$



人工神经网络

2.2. 反向传播 (输出层)

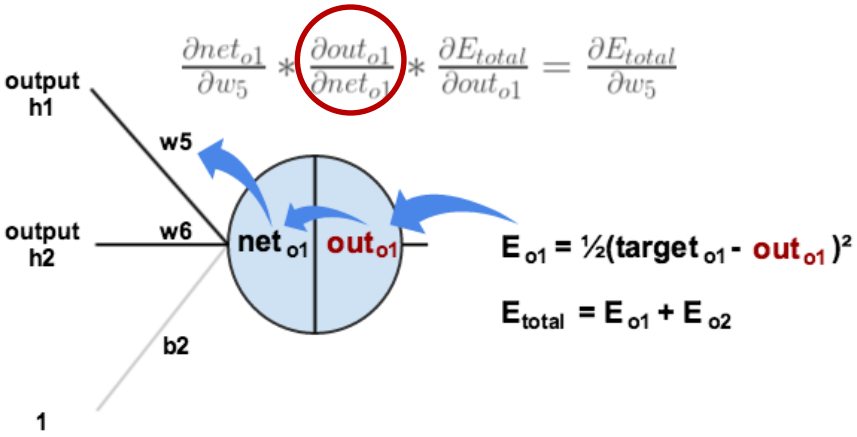


$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2 \Rightarrow \frac{\partial E_{total}}{\partial out_{o1}} = 2 \times \frac{1}{2}(target_{o1} - out_{o1})^{2-1} \times (0-1)$$

$$\begin{aligned} \frac{\partial E_{total}}{\partial out_{o1}} &= -(target_{o1} - out_{o1}) \\ &= -(0.01 - 0.75136507) \\ &= 0.74136507 \end{aligned}$$

人工神经网络

2.2. 反向传播(输出层)

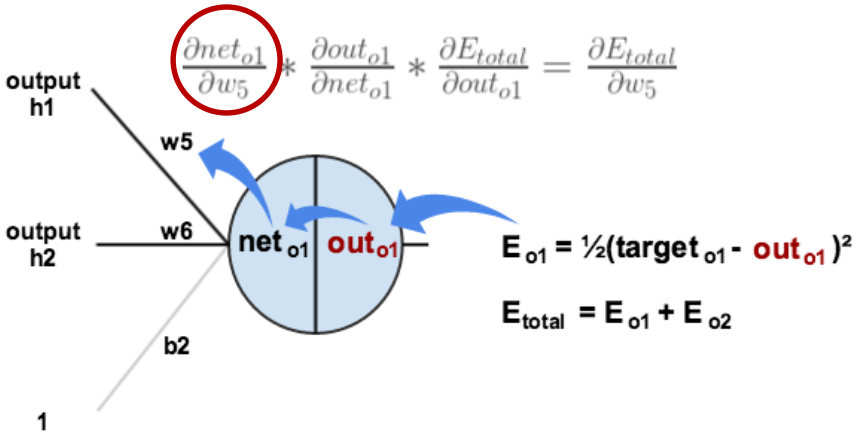


$$out_{o1} = \frac{1}{1+e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1-out_{o1})$$
$$= 0.75136507 \times (1-0.75136507)$$
$$= 0.186815602$$

人工神经网络

2.2. 反向传播(输出层)

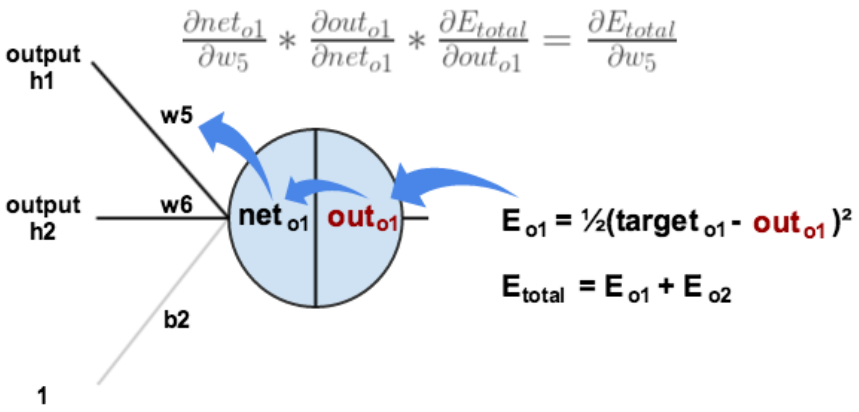


$$net_{o1} = w5 \times out_{h1} + w6 \times out_{h2} + b_2 \times 1 \quad \Rightarrow \quad \frac{\partial net_{o1}}{\partial w5} = out_{h1} = 0.593269992$$

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w5}$$
$$= 0.74136507 \times 0.186815602 \times 0.593269992$$
$$= 0.082167041$$

人工神经网络

2.2. 反向传播 (输出层)– 更新权重



$$w5^{+} = w5 - \eta \times \frac{\partial E_{total}}{\partial w5}$$
$$= 0.4 - 0.5 \times 0.082167041$$
$$= 0.35891648$$

同理

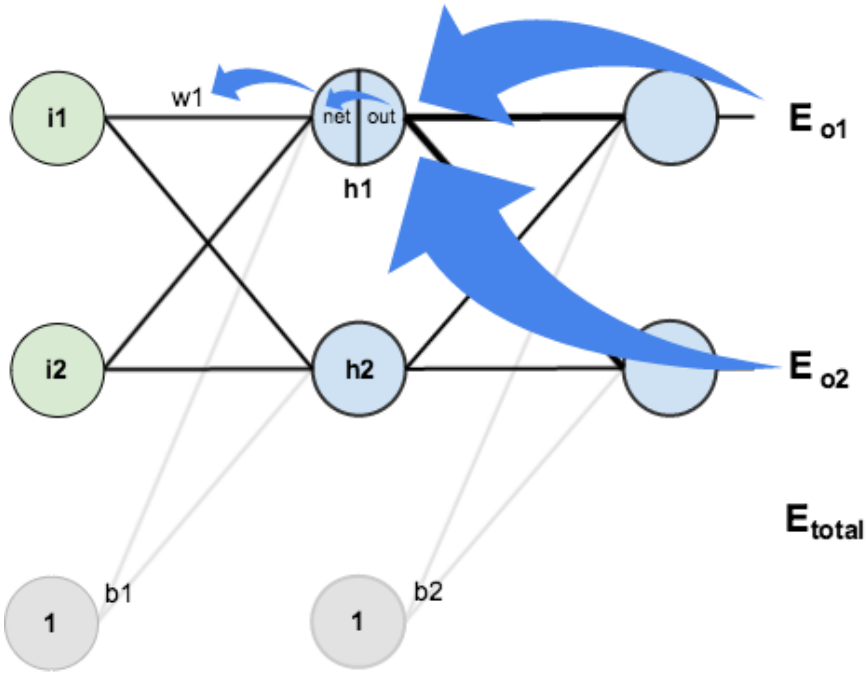
$$w6^{+} = 0.408666186$$
$$w7^{+} = 0.511301270$$
$$w8^{+} = 0.561370121$$

人工神经网络

2.2. 反向传播 (隐含层)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$E_{total} = E_{o1} + E_{o2}$

$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

out_{h1}对out_{o1}和out_{o2}都有影响，所以

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial out_{h1}}$$

$$\frac{\partial E_{o1}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}}$$

$$= 0.74136507 \times 0.186815602$$

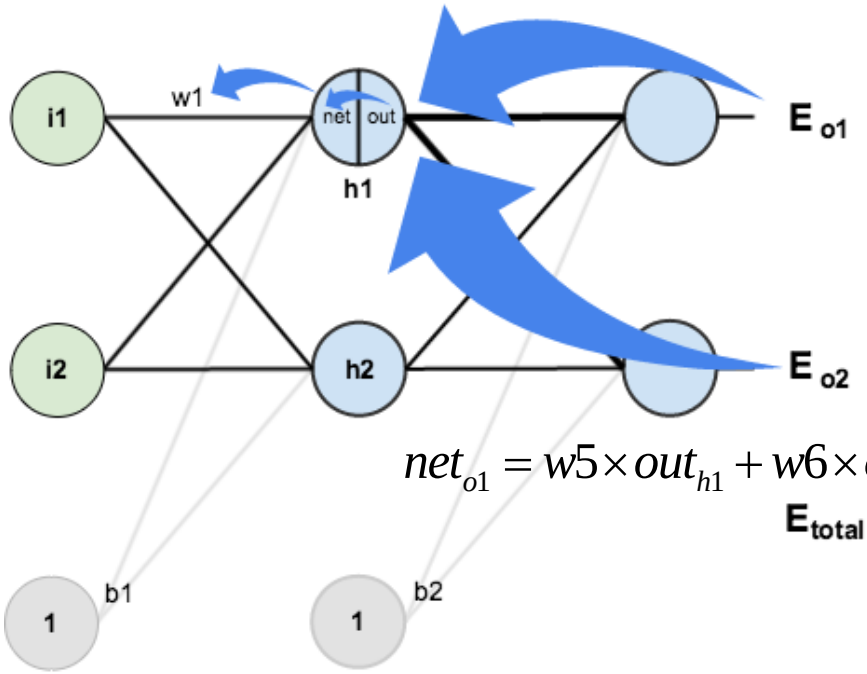
$$= 0.138498562$$

人工神经网络

2.2. 反向传播 (隐含层)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

out_{h1}对out_{o1}和out_{o2}都有影响，所以

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial out_{h1}}$$

$$net_{o1} = w5 \times out_{h1} + w6 \times out_{h1} + b2 \times 1$$

$$\frac{\partial net_{o1}}{\partial out_{h1}} = w5 = 0.40$$

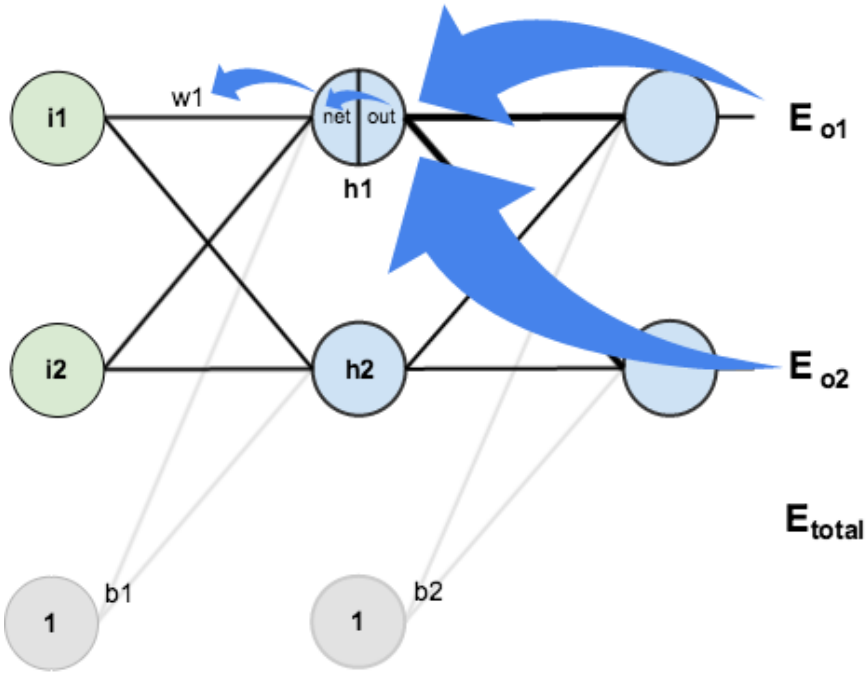
$E_{total} = E_{o1} + E_{o2}$

人工神经网络

2.2. 反向传播 (隐含层)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$E_{total} = E_{o1} + E_{o2}$

$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

out_{h1}对out_{o1}和out_{o2}都有影响，所以

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial out_{h1}}$$

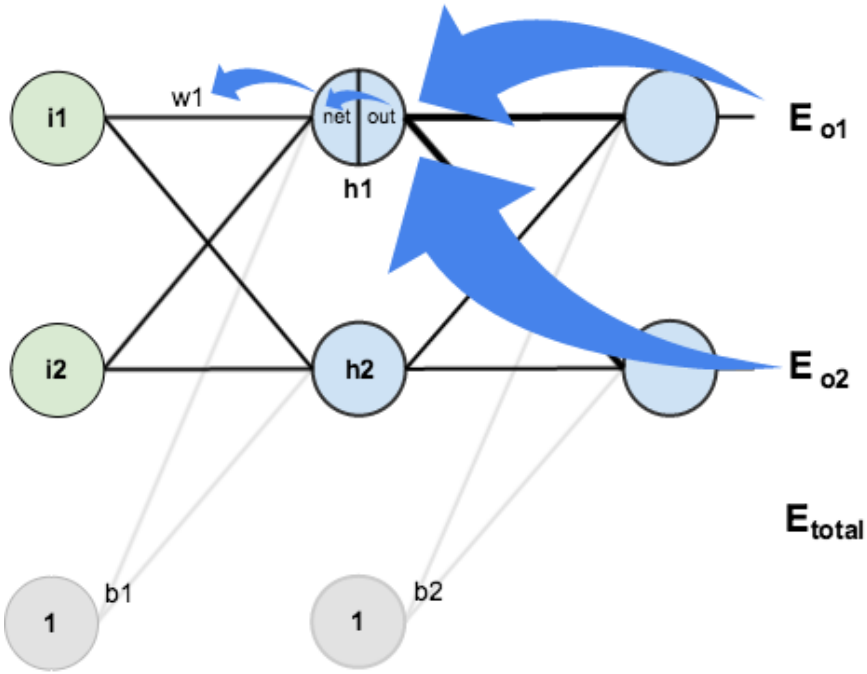
$$\frac{\partial E_{o1}}{\partial out_{h1}} = 0.138498562 \times 0.40 = 0.055399425$$

人工神经网络

2.2. 反向传播 (隐含层)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

out_{h1}对out_{o1}和out_{o2}都有影响，所以

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

同理 $\frac{\partial E_{o2}}{\partial out_{h1}} = -0.019049119$

$$\frac{\partial E_{total}}{\partial out_{h1}} = 0.055399425 + (-0.019049119)$$

$$= 0.036350306$$

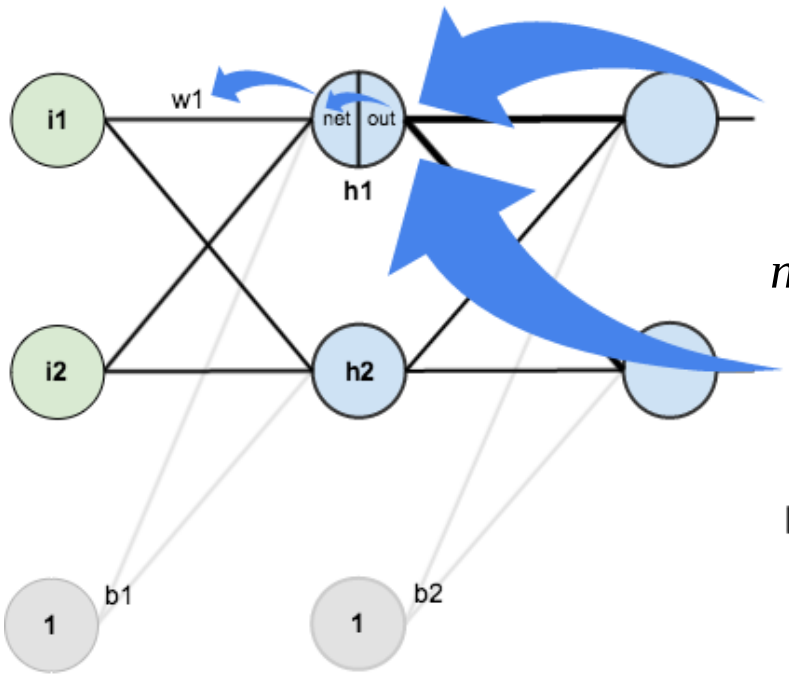
$$E_{total} = E_{o1} + E_{o2}$$

人工神经网络

2.2. 反向传播 (隐含层)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$E_{o1}$$

$$E_{o2}$$

$$E_{total} = E_{o1} + E_{o2}$$

$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}}$$

$$\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1-out_{h1})$$

$$= 0.59326999(1-0.59326999)$$

$$= 0.241300709$$

$$net_{h1} = w1 \times i1 + w2 \times i2 + b1 \times 1$$

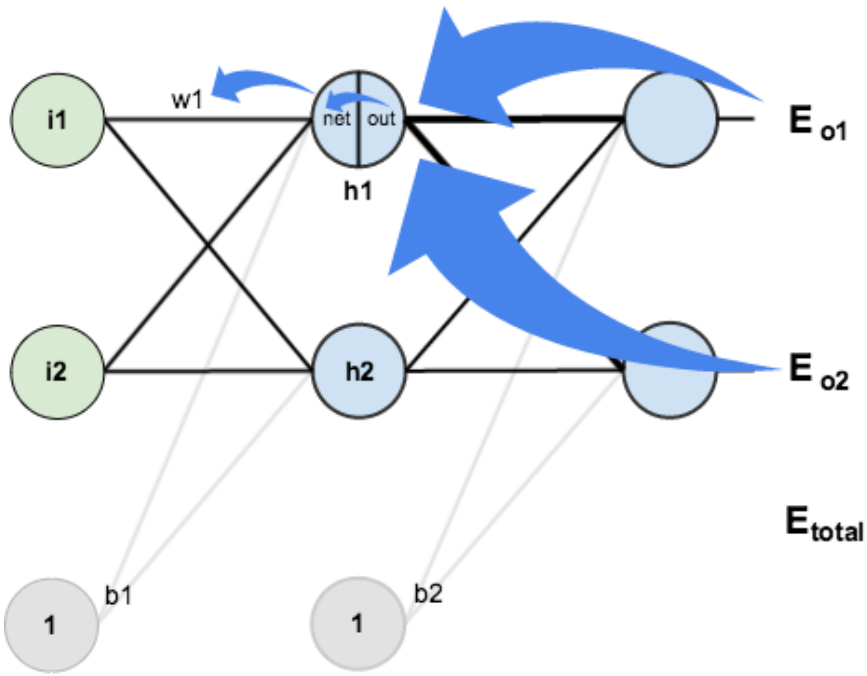
$$\frac{\partial net_{h1}}{\partial w1} = i1 = 0.05$$

人工神经网络

2.2. 反向传播 (隐含层)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

$$\frac{\partial E_{total}}{\partial w1} = 0.036350306 \times 0.241300709 \times 0.05$$

$$= 0.000438568$$

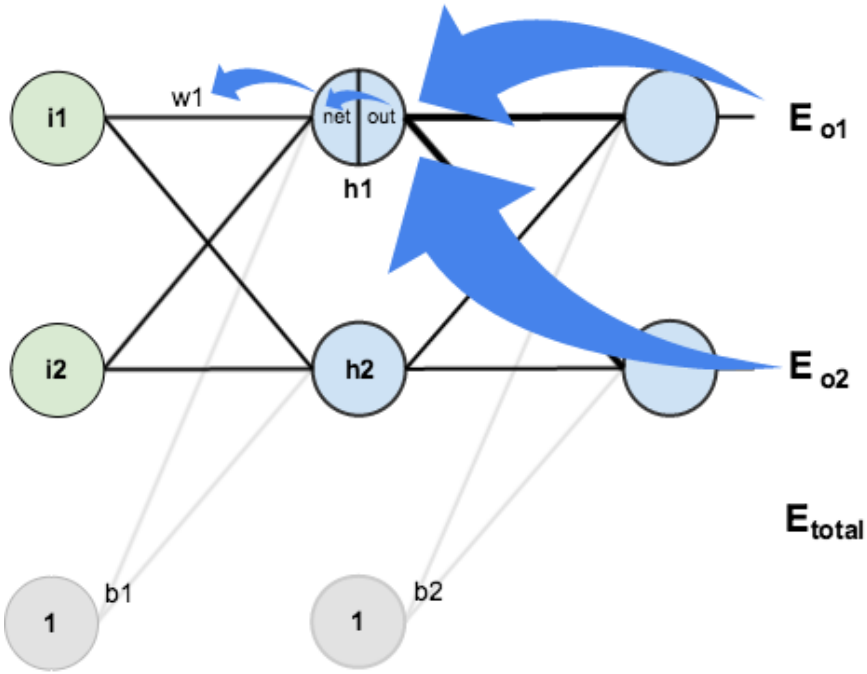
$$E_{total} = E_{o1} + E_{o2}$$

人工神经网络

2.2. 反向传播 (隐含层—更新权重)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$E_{total} = E_{o1} + E_{o2}$$

$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w1}$$

$$w1^{+} = w1 - \eta \times \frac{\partial E_{total}}{\partial w1}$$

$$= 0.15 - 0.5 \times 0.000438568$$

$$= 0.149780716$$

同理

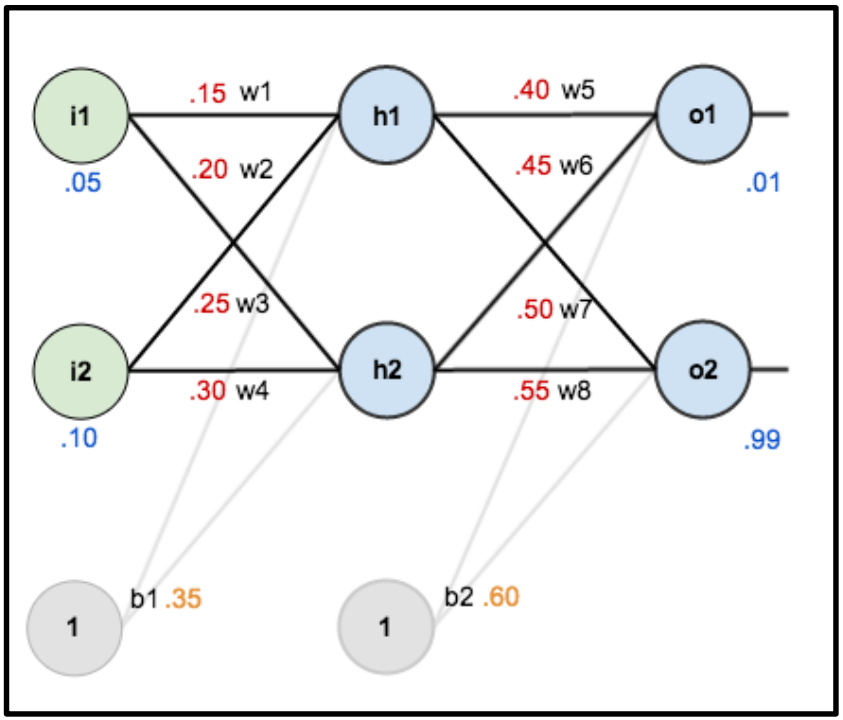
$$w2^{+} = 0.19956143$$

$$w3^{+} = 0.24975114$$

$$w4^{+} = 0.29950229$$

人工神经网络

3. 终止迭代



- 以上步骤完成了对于一个训练样本在神经网络中的一次迭代过程：前向传播，反向传播及权重更新
- 通常需要多次迭代，直到误差小于预设的阈值
- 而且，需要多个训练样本，每个样本执行以上过程
- 因此，这样训练出的网络才能对新的样本有预测能力

示例代码： lect07_nn

目录

实战案例：电影口碑与海报图像的相关性分析

- 通过人脸检测获取海报中人脸个数及颜色均值
- 分析海报人脸个数、颜色均值与评分的关系

示例代码： lect07_proj

参考

- 深度 | 微软亚洲研究院常务副院长郭百宁：计算机视觉的最新研究与应用
  http://chuansong.me/n/401296742796
- scikit-image官网
  http://scikit-image.org/
- 中值滤波
  http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm
- 高斯滤波
  http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm
- SIFT特征提取分析
  http://blog.csdn.net/abcjennifer/article/details/7639681
- SIFT demo
  http://people.cs.ubc.ca/~lowe/keypoints/

参考

- HOG 特征-理解篇
  http://blog.csdn.net/abcjennifer/article/details/7365651
- 微软Cognitive-Face-Python
  https://github.com/Microsoft/Cognitive-Face-Python
- 逐步教你神经网络中的反向传播
  https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

疑问

- 问题答疑： http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

小象问答 @Robin_TY

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象
- 新浪微博：ChinaHadoop