

前言概述部分

1. 可视媒体包含哪些类型，涉及哪些研究领域；

包含类型：可视媒体是承载视觉信息的媒体类型，主要包括视频、音频、图形、图像等

研究领域：计算机图形学、计算机视觉、人工智能

视频处理、图像处理、计算光谱成像、智能影像处理、目标检测识别等

2. 计算机视觉是一门什么学科；

计算机视觉是一门研究如何使机器“看”的科学，更进一步的说，就是指用摄影机和电脑代替人眼对目标进行识别、跟踪和测量等机器视觉，并进一步做图形处理，使电脑处理成为更适合人眼观察或传送给仪器检测的图像。

3. 说明计算机视觉和计算机图形学的差异；

Input \ Output	Image	Knowledge
	Image	Knowledge
Image	Digital Image Processing	Computer Vision
Knowledge	Computer Graphics	Artificial Intelligence

色彩和图像基础

4. 常用的色彩空间，每个色彩空间包含的分量及其含义；

RGB(红色、绿色、蓝色)

HSV(hue 色相、saturation 饱和度、value 明度)

YUV(Y 灰度、uv 色差)

5. 为什么视频中更多使用类 YUV 色彩空间；

兼容彩色黑白显示设备，考虑数据压缩

相较于 RGB 视频传输，YUV 只需要占用极少的频宽

人的视觉系统对于灰度的改变比色彩的改变更敏感，可以适当的抛弃 UV 的分量达到压缩的目的

灰度信号与色度信号分离，用灰度信号 Y 解决了黑白电视机与彩色电视机的兼容问题

6. 图像在计算机中存储表示的方式-矩阵；

7. 图像的色彩调整的处理过程：对比度、色彩饱和度、亮度；（描述思想）

亮度：

一副图像的亮度对比度调节属于图像的灰度线性变换，其公式如下：

$$y = [x - 127.5 * (1 - B)] * k + 127.5 * (1 + B);$$

x 为调节前的像素值，y 为调节后的像素值。

其中 B 取值[-1,1]，调节亮度；

k 调节对比度， $\arctan(k)$ 取值[1,89]，所以

$$k = \tan((45 + 44 * c) / 180 * \pi);$$

其中 c 取值[-1,1]。通常我们用该值来设置对比度

特别的，

当 $B=0$ 时: $y = (x - 127.5) * k + 127.5$; 这时只调节对比度。

当 $c=0$ 时, $k = 1$: $y = x + 255 * B$; 这时只调节亮度。

饱和度:

1. 算法原理

这些公式仅适用于 RGB 图像, 对于 Lab 颜色图像而言, 这些公式将不再适用。

在公式中 RGB 调整前颜色的 R、G、B 分量, RGB' 调整后颜色的 R、G、B 分量。三个通道分别做相同的调整运算。nParameter 表示调整参数, 取值范围为 $[-100, 100]$ 。归一化后为 $\text{percent} \in [-1, 1]$ 。

1.1. 饱和度调整

对每个像素点,

$$\text{rgbMax} = \max(\max(R, G), B) \quad (1)$$

$$\text{rgbMin} = \min(\min(R, G), B) \quad (2)$$

$$\text{delta} = (\text{rgbMax} - \text{rgbMin}) / 255 \quad (3)$$

如果 $\text{delta} = 0$, 则不调整, 继续判断操作下一个像素。若 $\text{delta} > 0$

$$\text{value} = (\text{rgbMax} + \text{rgbMin}) / 255 \quad (4)$$

HSL 中的 L 为:

$$L = \text{value} / 2 \quad (5)$$

饱和度 S 为:

$$\begin{cases} S = \text{delta} / \text{value} & L < 0.5 \\ S = \text{delta} / (2 - \text{value}) & L \geq 0.5 \end{cases} \quad (6)$$

1) 如果 $\text{percent} \geq 0$, 首先求出 α ,

$$\begin{cases} \alpha = S & \text{percent} + S \geq 1 \\ \alpha = 1 - \text{percent} & \text{else} \end{cases} \quad (7)$$

$$\alpha = 1 / (\alpha + 1) \quad (8)$$

$$\text{RGB}' = \text{RGB} + (\text{RGB} - L \times 255) \times \alpha \quad (9)$$

2) 如果 $\text{percent} < 0$, $\alpha = \text{percent}$

$$\text{RGB}' = L \times 255 + (\text{RGB} - L \times 255) \times (1 + \alpha) \quad (10)$$

灰度:

8. OpenCV 中图像的基本运算: 打开图像文件、显示图像文件、访问像素颜色值; (函数, 怎么做)

```
#include "stdafx.h"
```

- #include <opencv2/core/core.hpp>
- #include <opencv2/highgui/highgui.hpp>
- int _tmain(int argc, _TCHAR* argv[])
- {
- cv::Mat image = cv::imread("Winter.jpg");
- cv::namedWindow("WinTitle");
- cv::imshow("WinTitle", image);
- cv::waitKey(5000);
- return 0;

- }
- cv::Mat 存储图像
- image.cols, image.rows, image.channels
- 像素颜色值 image.at<uchar>(j,i)
- 像素颜色值 image.at<cv::Vec3b>(j,i)[channel]
- uchar, vec3b [0..255]
- uchar tmp = 0;
- tmp = image.at<cv::Vec3b>(j,i)[0];
- image.at<cv::Vec3b>(j,i)[0] = image.at<cv::Vec3b>(j,i)[2];
- image.at<cv::Vec3b>(j,i)[2] = tmp;

Mat src,dst;

src=imread("1.jpg",1);载入三通道的彩色图像

namedWindow("Display", 1);创建窗体

imshow("Display", src);图片显示到指定窗体

访问像素颜色值:

for (int row = 0; row < src.rows; row++)

{

for (int col = 0; col < src.cols; col++)

{

for (int c = 0; c < 3; c++) //3 个通道

{

dst.at<Vec3b>(row, col)[c] = 对原图像素的操作;

}

}

}

图像滤波

9. 线性滤波、滤波核的基本概念;

线性滤波: 用相邻像素的线性组合(加权和)替换每个像素

滤波核: 线性组合的处理方法即为核(内核)

10. 均值、中值、最大最小值、高斯、双边滤波的概念;

均值滤波: 用中心像素及其周围像素的平均值来代替中心像素的值

高斯滤波: 对每个坐标值对应的像素点做下式的变换

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

中值滤波: 用中心像素及其周围像素的中值来代替中心像素的值

最大最小值滤波: 如果中心像素值比周围最小值小, 则把中心像素替换为周围的最小值, 反之亦然

双边滤波: 对每个坐标的像素值做下述变换。

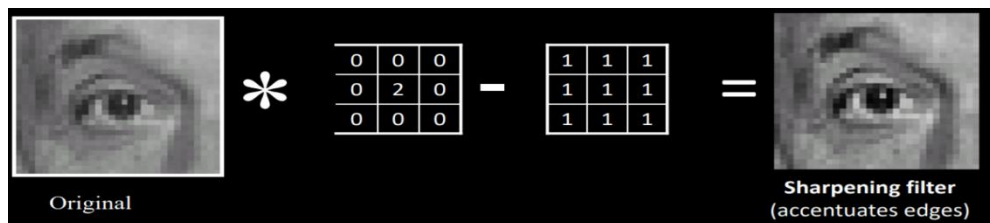
$$BF[I]_p = \overset{\text{new}}{\frac{1}{W_p}} \sum_{q \in S} \overset{\text{not new}}{G_{\sigma_s}(\|p - q\|)} \overset{\text{new}}{G_{\sigma_r}(\|I_p - I_q\|)} I_q$$

normalization factor
space weight
range weight

对每个坐标的像素值做上述变换。

11. 通过滤波实现图像锐化、边缘提取的计算思路；

图像锐化：



边缘提取：

该算子包含两组3x3的矩阵，分别为横向及纵向，将之与图像作平面卷积，即可分别得出横向及纵向的高度差分近似值。如果以A代表原始图像，Gx及Gy分别代表经横向及纵向边缘检测的图像灰度值，其公式如下：

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

<http://blog.csdn.net/>

其中f(a,b), 表示图像(a,b)点的灰度值；

图像的每一个像素的横向及纵向灰度值通过以下公式结合，来计算该点灰度的大小：

$$G = \sqrt{G_x^2 + G_y^2}$$

通常，为了提高效率 使用不开平方的近似值：

$$|G| = |G_x| + |G_y|$$

如果梯度G大于某一阈值 则认为该点(x,y)为边缘点。

照片计算

12. Seamcarving 中 seam 的计算过程

给出能量矩阵，计算 seam；

5 8 12 3

5 8 12 3

4 2 3 9 9 7 6 12
 7 3 4 2 14 9 10 8
 4 5 7 8 13 14 15 16

解法：第 1 行保持不变

第 2 行：4 可+5 或者+8，选小的，变为 9；2 可+5，+8，+12，选小的，变为 7；依次计算出第 2 行的值，

原来的第 3 行与计算出的第 2 行进行同样的计算；

依次计算出 4 行，从第 4 行最小的数开始，向上找上一行与之最小值相邻的最小值。

13. 暗通道去雾中暗通道的计算过程

给出图像数据，计算暗通道；

去雾模型

$I = J * t + A(1 - t)$ I :有雾图像 J :要恢复的无雾图像 A :大气光成分 t :透射率

两边同时除以每个通道的大气光值， c 表示 rgb 的某一通道

$I^c / A^c = (J^c / A^c) * t + 1 - t$

假设在图像中一定大小的矩形窗口 $\Omega(x)$ 内

两边用最小化算子作最小化运算

$$\min_{\Omega} \left(\min_c \frac{I^c}{A^c} \right) = \left\{ \min_{\Omega} \left(\min_c \frac{J^c}{A^c} \right) \right\} t + 1 - t$$

右式第一项值趋于 0

$$t = 1 - \min_{\Omega} \left(\min_c \frac{I^c}{A^c} \right)$$

Dark Channel Prior（暗通道先验）

$\min(\min(r, g, b), \text{local patch})$

local patch = size * size

图像特征

14. SIFT、HOG 特征的概念；

SIFT 特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性

HOG 特征是一种在计算机视觉和图像处理中用来进行物体检测的特征描述子。HOG 特征通过计算和统计图像局部区域的梯度方向直方图来构成特征

15. SIFT、HOG 特征的计算过程；

SIFT 特征计算过程:

1. 尺度空间的极值检测：搜索所有尺度空间上的图像，通过高斯微分函数来识别潜在的对尺度和选择不变的兴趣点。
2. 特征点定位：在每个候选的位置上，通过一个拟合精细模型来确定位置尺度，关键点的选取依据他们的稳定程度；
3. 特征方向赋值：基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向，后续的所有操作都是对于关键点的方向，尺度和位置进行变换，从而提供这些特征的不变性。
4. 特征点描述：在每个特征点周围的邻域内，在选定的尺度上测量图像的局部梯度，这些梯

度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变换。

HOG 特征计算过程：

- 1) 灰度化（将图像看做一个 x,y,z （灰度）的三维图像）；
- 2) 采用 Gamma 校正法对输入图像进行颜色空间的标准化（归一化）；目的是调节图像的对比度，降低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰；
- 3) 计算图像每个像素的梯度（包括大小和方向）；主要是为了捕获轮廓信息，同时进一步弱化光照的干扰。
- 4) 将图像划分成小 cells（例如 $6*6$ 像素/cell）；
- 5) 统计每个 cell 的梯度直方图（不同梯度的个数），即可形成每个 cell 的 descriptor；
- 6) 将每几个 cell 组成一个 block（例如 $3*3$ 个 cell/block），一个 block 内所有 cell 的特征 descriptor 串联起来便得到该 block 的 HOG 特征 descriptor。
- 7) 将图像 image 内的所有 block 的 HOG 特征 descriptor 串联起来就可以得到该 image（你要检测的目标）的 HOG 特征 descriptor 了。这个就是最终的可供分类使用的特征向量了。

16. 特征维度的概念，特征相似度一般怎么度量；

特征维度就是图像中特征向量的个数，特征向量可以理解为坐标轴

特征相似度度量方法：空间 欧氏距离 路径：曼哈顿距离

17. 两张图像进行全景图拼接的计算思路（利用 SIFT 特征）；

- 1.对两张图进行特征点提取
- 2.对特征点进行匹配
- 3.进行图像配准
- 4.把图像拷贝到另一幅图像的特定位置
- 5.对重叠边界进行特殊处理

18. 图像分类的计算思路（利用 HOG 特征）；

- 1.HOG 特征的数据集与标签数据集制作处理
- 2.训练 svm 分类器
- 3.加载分类器进行预测

视频

19. 帧率、码率；隔行、逐行扫描的概念；

帧率：以帧称为单位的位图图像连续出现在显示器上的频率

码率：视频数据（视频色彩量、亮度量、像素量）每秒输出的位数，一般单位为 kbps

逐行扫描：每一帧图像由电子束顺序地一行接着一行连续扫描而成

隔行扫描：把每一帧图像通过两场扫描完成

20. 视频编码压缩的必要性；

- 1.要使数字电视信号适合于实际存储和传输，必须压缩数据量，降低传输数据码率
- 2.数据用来记录和传送信息，是信息的载体；数据的处理结果是信息。数据压缩的目的是在传送和处理信息时，尽量减少数据量与信息量的关系：数据量=信息量+冗余数据量
- 3.冗余种类
时间上的冗余：相邻帧的强相关性
空间上的冗余：相邻像素的强相关性

感知上的冗余：人眼对信息的敏感性

统计上的冗余：统计的概率

21. 视频压缩和通用数据压缩的差异；

视频压缩除了对单幅图像进行编码外还利用图像序列中的相关原则，将冗余去掉。

22. 文件格式和编码方式的差异；

文件格式即容器，就是把编码器生成的多媒体内容混合封装在一起的标准。

编码方式就是指通过特定的压缩技术，将某个视频格式的文件转换成另一种视频格式文件的方式

同一种容器可以使用不同的编码，所以不能简单的通过文件格式或容器来区分视频

深度学习

23. 机器学习、深度学习的概念和关系；

AI（Artificial Intelligence）人工智能：对人的意识、思维过程的模拟

ML（Machine Learning）机器学习：

实现 AI 的方法

计算机利用已有数据，得到某个模型，并利用此模型预测未来的一种方法

利用到模式识别、数据挖掘、计算机视觉、语音识别、自然语言处理等

DL（Deep Learning）深度学习：机器学习中一种对数据进行表征学习的方法，实质上是层数更多的神经网络学习方法。

24. 卷积神经网络中卷积核计算

给出输入、卷积核、Stride, padding 信息，计算输出；

25. 卷积神经网络的输入输出尺寸计算

给出输出尺寸、卷积核尺寸、stride, padding 信息，给出输出尺寸；

$$o = \left\lfloor \frac{n + 2p - f}{s} \right\rfloor + 1$$

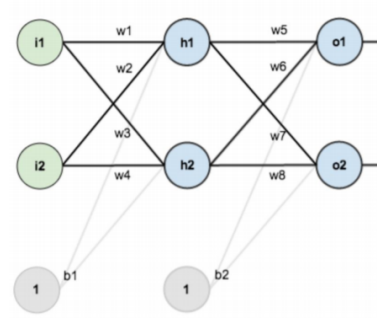
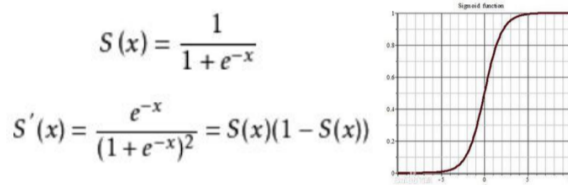
26. 全连接神经网络前向和后向计算过程

给出网络结构，初始化参数，前向计算得到结果，并反向更新参数；

• 三层全连接网络

- 输入层：两个神经元 i1、i2
- 隐层：h1、h2
- 输出层：o1、o2

- w_i 表示连接上的weights, b_i 表示biases
- 神经元上的激活函数是sigmoid

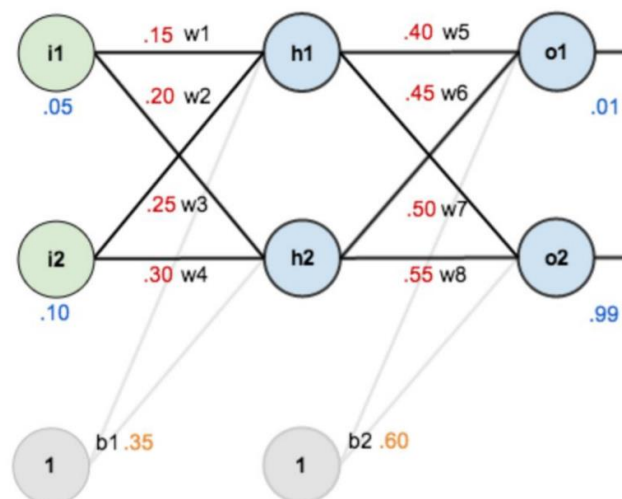


训练过程的初始状态

• 当前训练样本

- 输入 = [0.05, 0.10]
- 预期输出 = [0.01, 0.99]
- 当前 w 、 b 如图(注意：为简化计算此处同层共享 b ，且 b 固定)

- 训练目标：调节 w 使得真实输出尽量接近预期输出



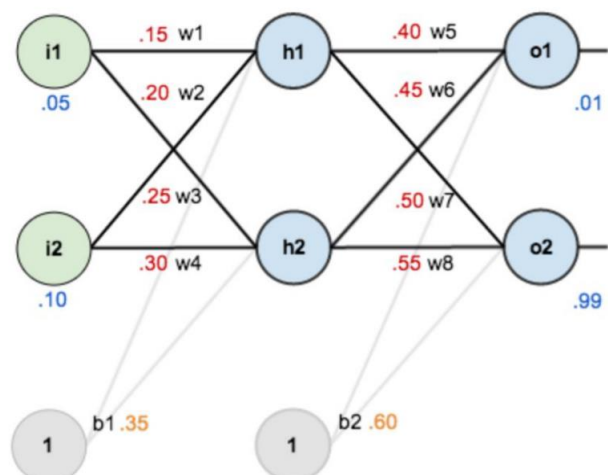
前向计算：输入层->隐层

• h1为例：

- $net_{h1} = w1*i1 + w2*i2 + b1 = 0.15*0.05 + 0.2*0.1 + 0.35 = 0.3775$
- $out_{h1} = 1/(1 + \exp(-0.3775)) = 0.593269992$

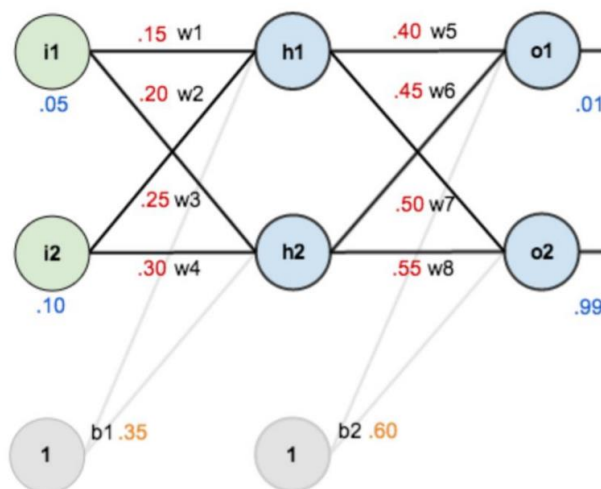
• 同理可得

- $out_{h2} = 0.596884378$



前向计算：隐层->输出层

- 以o1为例：
 - $net_{o1} = w5 \cdot out_{h1} + w6 \cdot out_{h2} + b2 = 0.4 \cdot 0.593269992 + 0.45 \cdot 0.596884378 + 0.6 = 1.105905967$
 - $out_{o1} = 1 / (1 + \exp(-1.105905967)) = 0.75136507$
- 同理可得
 - $out_{o2} = 0.772928465$
- 实际输出与预期输出存在差异，需要更新网络参数



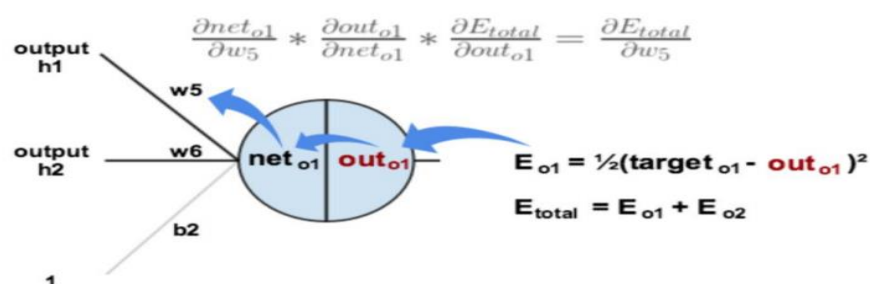
反向传播 Loss函数

- 定义Loss函数：
 - $E_{total} = \sum (target - output)^2 / 2$
 - $E_{o1} = (0.01 - 0.75136507)^2 / 2 = 0.274811083$
 - $E_{o2} = 0.023560026$
 - $E_{total} = E_{o1} + E_{o2} = 0.298371109$

反向传播：输出层参数更新

- 以w5为例：

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$



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

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

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

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

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

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$$

更新参数

- 计算梯度，根据梯度和学习率(learning rate)更新参数

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 * 0.082167041 = 0.35891648$$

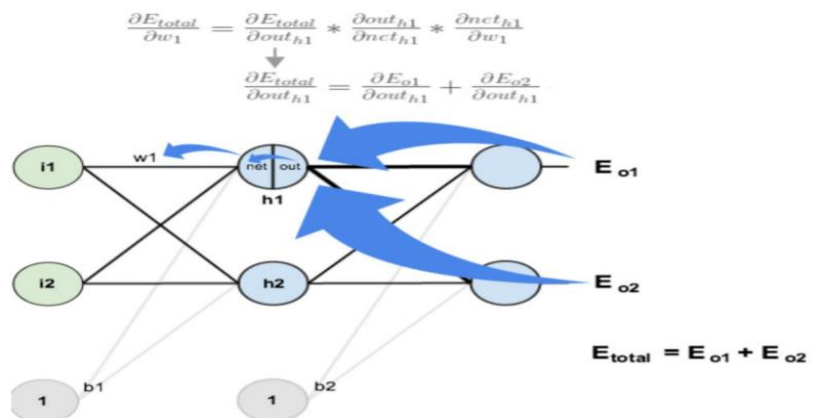
$$w_6^+ = 0.408666186$$

$$w_7^+ = 0.511301270$$

$$w_8^+ = 0.561370121$$

反向传播：隐层参数更新

- 以w1为例：



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

$$\frac{\partial E_{o1}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} = 0.74136507 * 0.186815602 = 0.138498562$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial out_{h1}} = w_5 = 0.40$$

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

• 同理可得

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

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} = 0.055399425 + -0.019049119 = 0.036350306$$

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

$$\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1 - out_{h1}) = 0.59326999(1 - 0.59326999) = 0.241300709$$

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$\frac{\partial net_{h1}}{\partial w_1} = i_1 = 0.05$$

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

$$\frac{\partial E_{total}}{\partial w_1} = 0.036350306 * 0.241300709 * 0.05 = 0.000438568$$

$$w_1^+ = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.000438568 = 0.149780716$$

$$w_2^+ = 0.19956143$$

$$w_3^+ = 0.24975114$$

$$w_4^+ = 0.29950229$$

