机器学习：让机器且备寻找一个函式的能力

深度学习的发展历程：

1958年，提出Perceptron

1969年，Perceptron 被发现学到了

1980年，提出了多层的

1986年，提出了反向传播

1989年，有人提出单层可以近似任意连续函数，多层的函数也没人用了

2006年，RBM initialization

2009年，开始用GPU加速

2011年，在语音识别上很有用

2012年，在图像上很有用

三个步骤：

a.定义函数集合

b.定义集合中每个函数的好坏程度

c.做优化

Model：就是机器学习中带有未知的Parameters的Function

y = b + wx

x为feature w为weight b为bias

Loss：损失函数，输入是Model的参数b、w；

超参数：机器学习中需要自己设定的参数

Sigmoid：足够多的Function加起来能够变成任何曲线，w改变sigmoid的斜率，b改变sigmoid的左右位置，c改变sigmoid的高度；

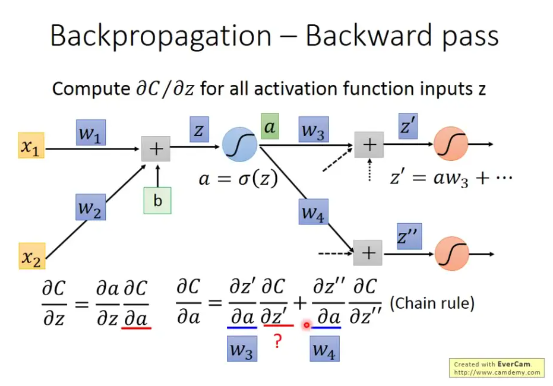
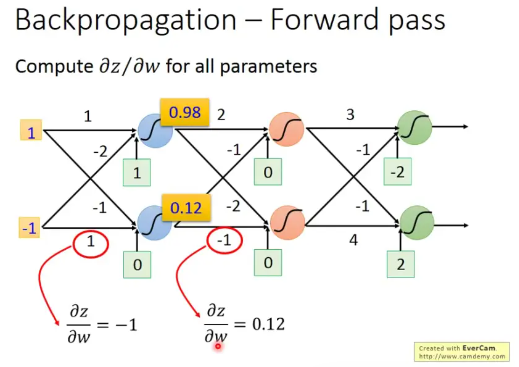
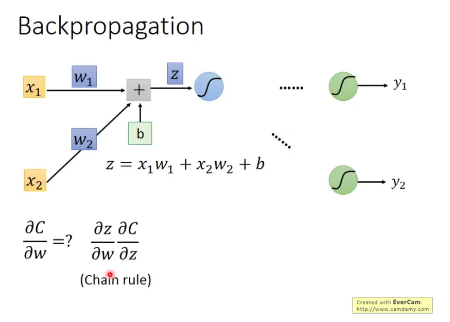
Epoch: 把所有的Batch都看过一遍叫做一个Epoch；

Update: 每一次更新参数叫做一次Update

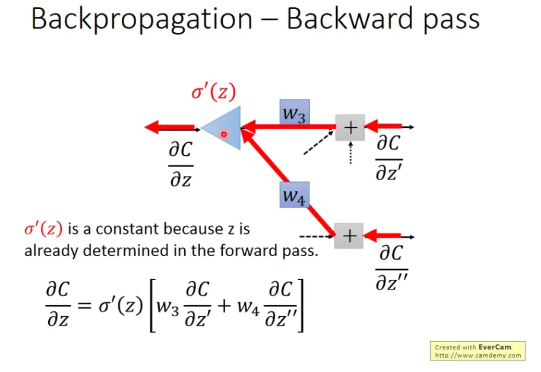
chain rule：

同支相乘,异支相加

前向传播：

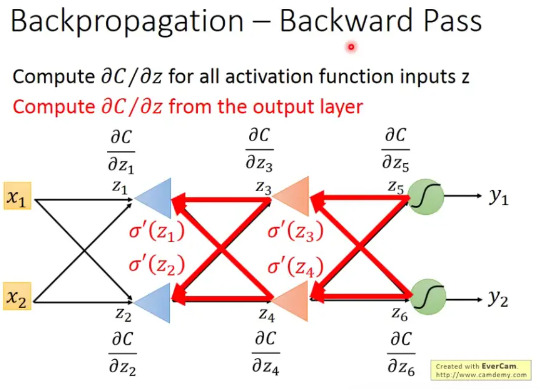


反向传播：



现在问题从Loss对z的偏导.转化威loss对z’和对z’’的偏导依次递归,然后从output layer反向传播回来

激活函数对参数的偏导.是在正向传播的时候就已经固定了，激活函数的导数是已经计算好的,它的值只取决于输入的值



实际计算的内容是建立一个反向的网络,可以计算出,loss对各个z的偏导.然后就可以计算出Loss对w的偏导数

