

深度学习入门笔记

神经网络的结构、梯度下降法

1. activation 激活值：暂时理解为每一层节点上的数值
2. weight 权值：神经网络连线上的值（可以理解为神经元之间连接的强弱）
3. bias 偏置值：决定激活某个神经元的难易程度
4. sigmod 压缩/映射函数：将 activation * weight 的乘积和映射到 [0,1]的标准范围内

Sigmoid

How positive is this?

$\sigma(w_1 a_1 + w_2 a_2 + w_3 a_3 + \dots + w_n a_n - 10)$

“bias”

5. 神经网络就是一个巨大的函数：输入第一层的activation ——> 输出最后一层的activation
6. 训练的过程就是找到正确的权值 weight 和 偏置 bias
7. loss function 损失函数：和cost function代价函数 常常混用
8. cost function 代价函数：差的平方和，来衡量不精确性
 - weight/bias: cost function 的 input
 - 代价cost: output
 - 样本examples: cost function 的参数 parameters（注：可以理解为函数被固化下来的值，例如 $y = k*x$ 中的k）

Cost function

Input: 13,002 weights/biases

Output: 1 number (the cost)

Parameters: Many, many, many
training examples

()

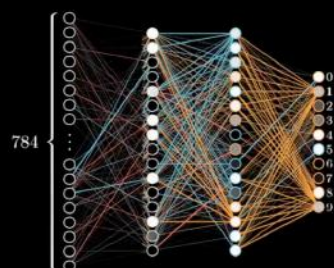
9. average cost 代价平均值 (经验风险) : 用几万个样本输出结果, 衡量训练后的神经网络的好坏

10. back propagation 反向传播算法(BP):

Backpropagation

Training in progress...

$\xi \rightarrow 5$



这个计算梯度的算法是神经网络的核心

The algorithm for computing this gradient efficiently, which is effectively the heart of how a neural network learns,



当我们提到让网络学习 实质上就是让代价函数的值最小

is that what we mean when we talk about a network learning is that it's just minimizing a cost function



11. gradient descent 梯度下降：

这种按照负梯度的倍数 不停调整函数输入值的过程

This process of repeatedly nudging an input of a function by some multiple of the negative gradient

梯度向量中的梯度值的相对大小，反映了改变哪个维度可以让函数值下降得更快

$$\vec{W} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{13,000} \\ w_{13,001} \\ w_{13,002} \end{bmatrix}$$

$$-\nabla C(\vec{W}) = \begin{bmatrix} 0.31 \\ 0.03 \\ -1.25 \\ \vdots \\ 0.78 \\ -0.37 \end{bmatrix}$$

w_0 should increase somewhat
 w_1 should increase a little
 w_2 should decrease a lot
 $w_{13,000}$ should increase a lot
 $w_{13,001}$ should decrease somewhat

每一项的相对大小更告诉了我们改变哪个值的影响更大
 the relative magnitudes of all these components kind of tells you which changes matter more

$$C(x, y) = \frac{3}{2}x^2 + \frac{1}{2}y^2$$

$$\nabla C(1, 1) = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

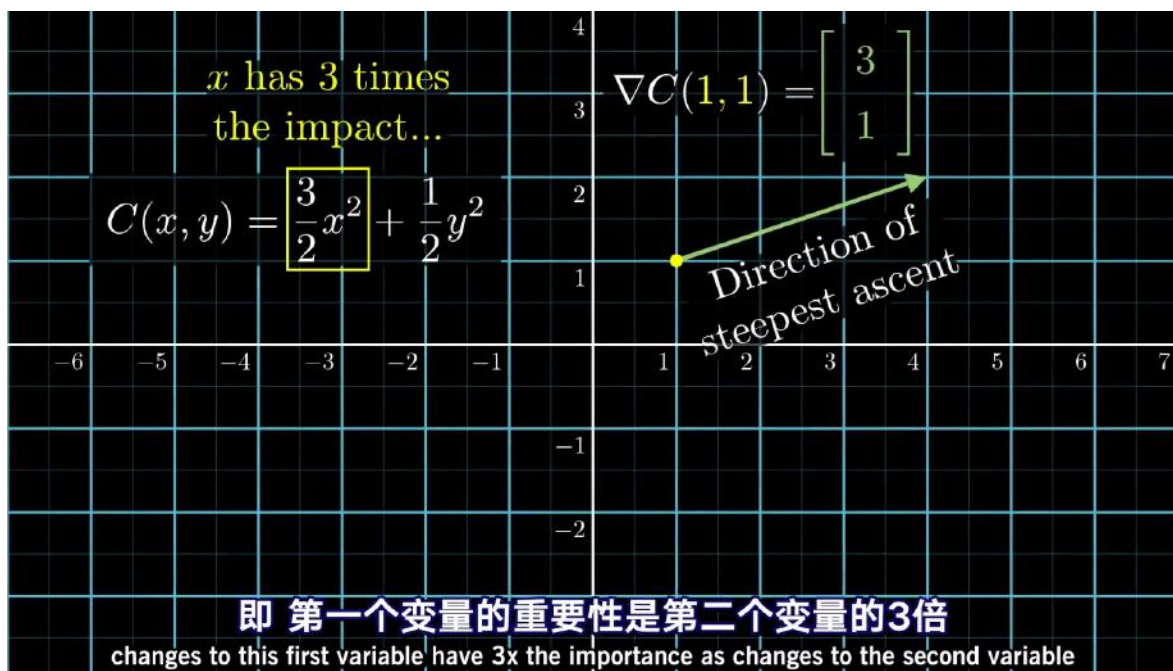
你算出来这个函数在某个点的梯度是 $[3, 1]$

and you compute that its gradient at some particular point comes out as $[3, 1]$

顺着这个梯度的方向移动 函数值增加得最快

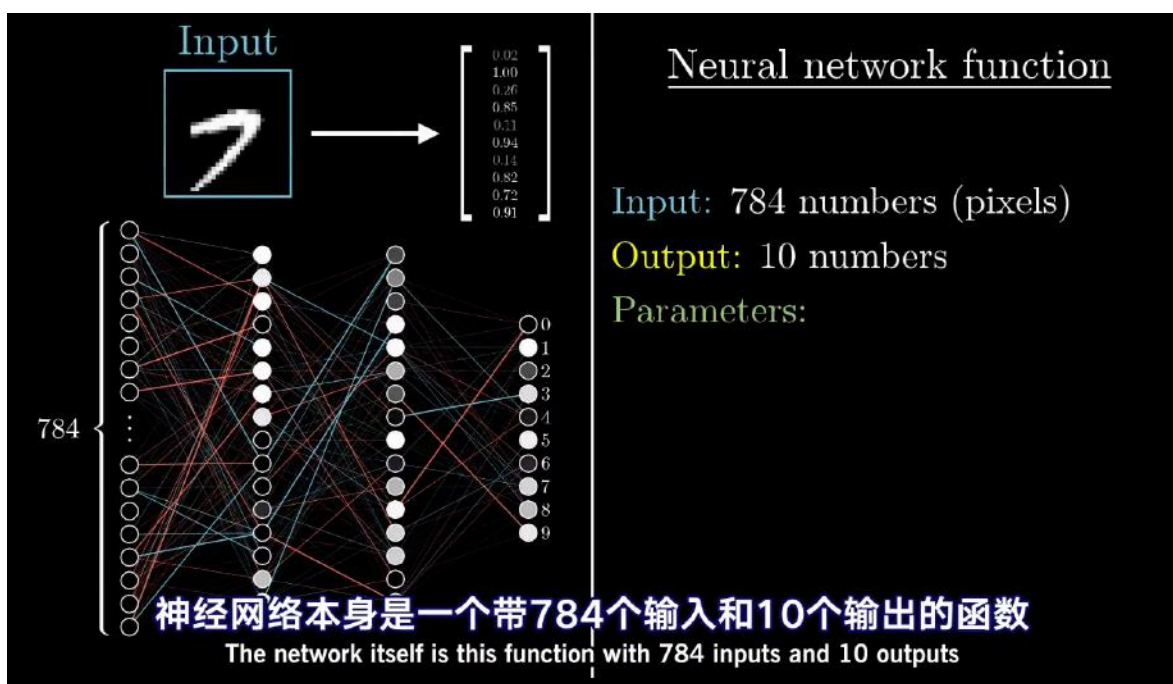
moving along this direction increases the function most quickly

另一种解读：

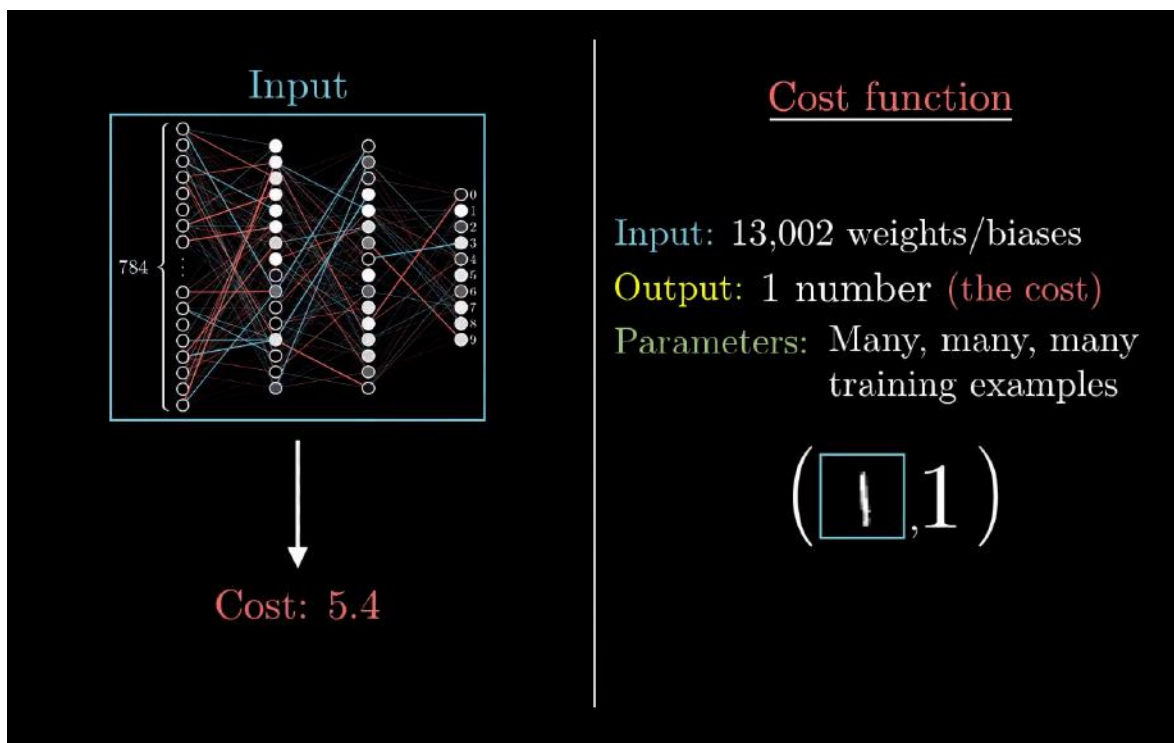


12. 总结各个抽象层次的含义：

— network本身：



— cost function:



– gradient of cost function:
 表示 weight/bias沿着哪个方向调整才能让cost function 的值下降得最快

