

不同激活函数论述

笔记本: pytorch

创建时间: 2023/4/2 10:59

更新时间: 2023/4/2 19:59

作者: Kaiyuecui

URL: <https://zhuanlan.zhihu.com/p/428448728>

1、

神经网络其实也就是为了寻找一种非线性的空间变换

如果仅仅是线性函数的话, 不论几层最终依旧是一个线性函数。

激活函数大全

<https://dashee87.github.io/deep%20learning/visualising-activation-functions-in-neural-networks/>

2、激活函数中饱和的意思

假设 $h(x)$ 是一个激活函数

如果其导数, 在 x 趋向于正无穷时, 值为0, 我们称为左饱和。类似的也有右饱和, 左右都饱和则为饱和

2、sigmoid

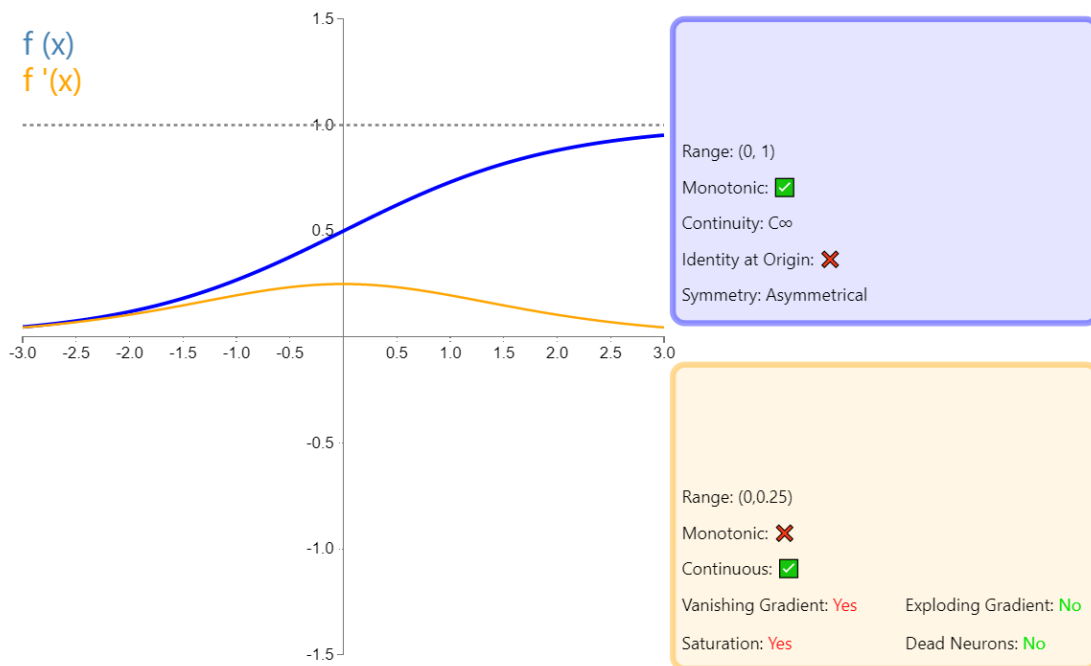
sigmoid是最常用的激活函数。

Sigmoid函数由下列公式定义

$$S(x) = \frac{1}{1 + e^{-x}}$$

其对 x 的导数可以用自身表示:

$$S'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = S(x)(1 - S(x))$$



注;sigmoid有梯度消失的问题，因为当值变大或者变小时，其梯度很小接近于0。
sigmoid后来在分类问题中，tanh用的更多。因为其是反对称的。

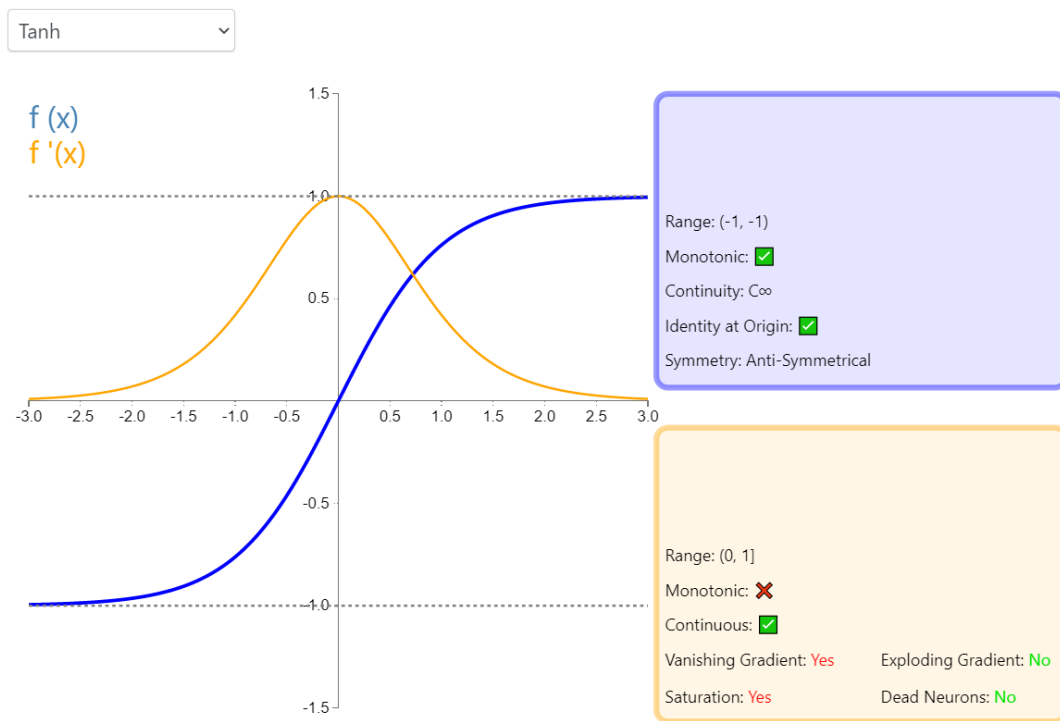
sigmoid的另一个缺点，都是正数，如果初始值不好，可能会导致出现zigzag现象，更新缓慢问题。

3、tanh

tanh与sigmoid其实是一个线性关系

$$y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$y = \tanh(x) = 2\text{sigmoid}(2x) - 1$$



相比于sigmoid，其输出不全是正数，可以缓解zigzag现象。
 但是它和sigmoid一样，都求导复杂，且容易出现梯度消失。
 4、relu

$$y = \max(0, x)$$

导数：

$$y = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$



relu的优点:

①没有饱和区，不会出现梯度消失的问题

②如果输入是个负数，就会使输出是0，（也就是说：在输入是负值的情况下，它会输出0，那么神经元就不会被激活。这意味着同一时间只有部分神经元会被激活，从而使得网络很稀疏，进而对计算来说是非常有效率的。）

那么问题来了：这种稀疏性有何作用？换句话说，我们为什么需要让神经元稀疏？不妨举栗子来说明。当看名侦探柯南的时候，我们可以根据故事情节进行思考和推理，这时用到的是我们的大脑左半球；而当看蒙面唱将时，我们可以跟着歌手一起哼唱，这时用到的则是我们的右半球。左半球侧重理性思维，而右半球侧重感性思维。也就是说，当我们在进行运算或者欣赏时，都会有一部分神经元处于激活或是抑制状态，可以说是各司其职。再比如，生病了去医院看病，检查报告里面上百项指标，但跟病情相关的通常只有那么几个。与之类似，当训练一个深度分类模型的时候，和目标相关的特征往往也就那么几个，因此通过ReLU实现稀疏后的模型能够更好地挖掘相关特征，拟合训练数据。

③计算简单，没有复杂的函数求导

④收敛比sigmoid和tanh快

缺点:

relu函数，如果学习率取得不恰当，过大的时候，会导致我们神经网络中几乎百分之40的神经单元都被抑制了。