

1. 什么是激活函数？

在多层神经网络中，上层节点的输出和下层节点的输入之间具有一个函数关系，这个函数称为激活函数。

2. 激活函数作用？

若没有激活函数，无论多少层神经网络，输出与输入都是线性关系。加入非线性的激活函数后，神经网络的表达能力更加强大，理论上能逼近任意函数。

3. 饱和激活函数 / 非饱和激活函数

饱和激活函数

x趋紧于无穷时，激活函数导数趋紧于0。

常见：sigmoid / tanh

非饱和激活函数

常见：ReLU

优势：能解决梯度消失问题 / 能加快收敛速度

4. 什么是梯度爆炸/消失？

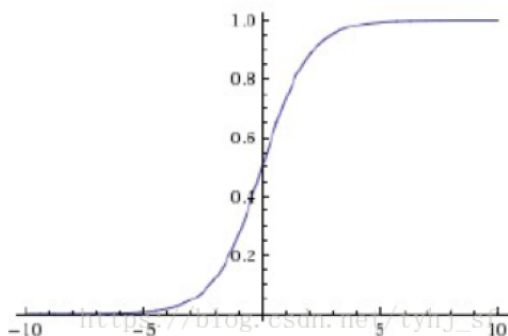
在梯度下降的过程中，如果某一层对激活函数求导 >1 ，那么随着层数的增多，最终的求出的梯度更新将以指数形式增加，即发生**梯度爆炸**，如果此部分小于1，那么随着层数增多，求出的梯度更新信息将会以指数形式衰减，即发生了**梯度消失**。

5. 常用激活函数总结

sigmoid

公式： $f(x) = \frac{1}{1+e^{-x}}$

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



缺点：

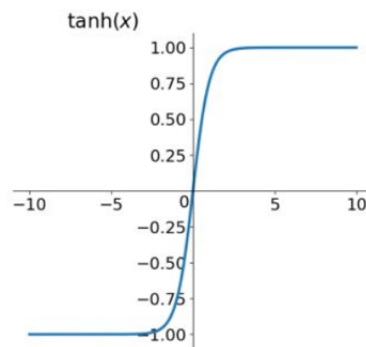
1. 存在梯度爆炸和梯度消失问题，梯度爆炸可能性较小，梯度消失可能性很大。

2. 指数函数运算消耗资源
3. 中心点非0（容易导致zigzag现象）

tanh

公式: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

导数: $f(x)' = \frac{4}{(e^x + e^{-x})^2}$

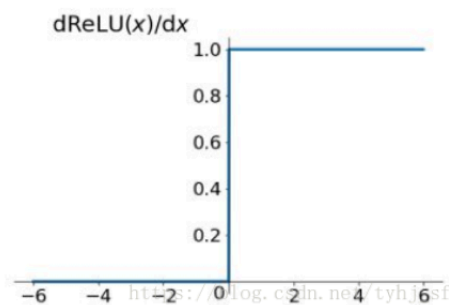
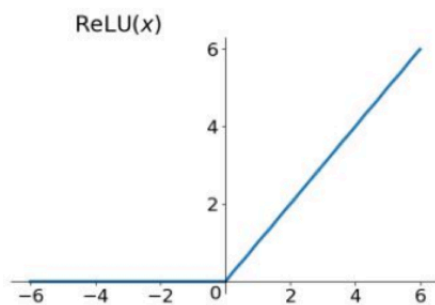


缺点:

1. 存在梯度爆炸和梯度消失问题，梯度爆炸可能性较小，梯度消失可能性很大。
2. 包含幂运算，运算代价较大

ReLU

公式: $f(x) = \max(0, x)$



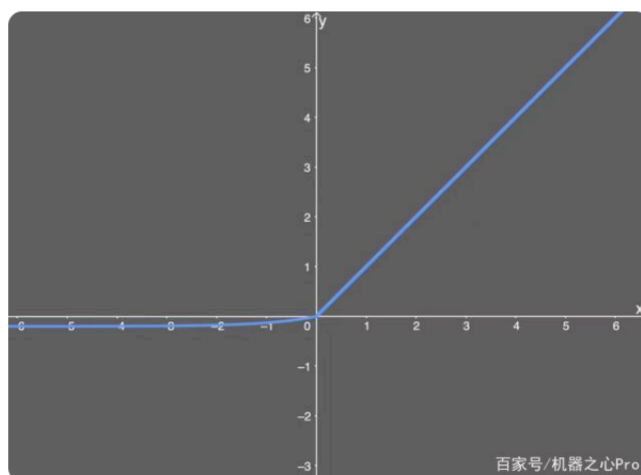
优点:

1. 解决梯度消失问题（非饱和函数）
2. 计算效率高，只需做max运算
3. 收敛速度快（梯度=1）

缺点:

1. 小于0的神经元会被永久置为0，导致部分神经元坏死

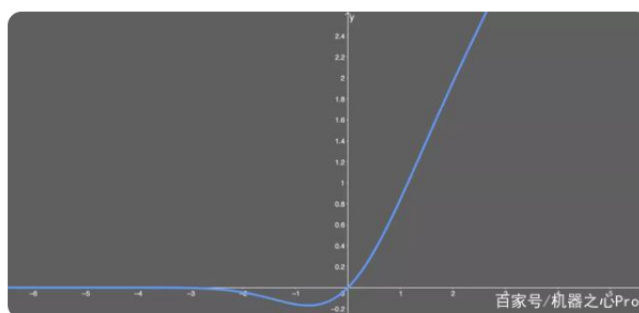
ELU



解决了Relu节点置0死亡问题

Gelu (Gaussian error linear units, 高斯误差线性单元)

公式: $Gelu(x) = x\phi(x)$, 其中, ϕ 是正态分布



加入随机正则的思想, 有点类似于**dropout**, $\phi(x)$ 决定x中有多少信息被保留, $\phi(x)$ 越大, x越有可能保留, 反之, 越有可能置0

softmax

把输入映射为0-1之间的实数, 并且归一化保证和为1

公式: $S_i = \frac{e^i}{\sum_j e^j}$

带有温控系数的softmax

公式: $S_i = \frac{e^{x_i/T}}{\sum_j e^{x_j/T}}$

若温控系数 T 越大, 元素之间的差距会变小。

若温控系数 T 越小, 元素之间差距会越大, 结果越趋向于one-hot向量。