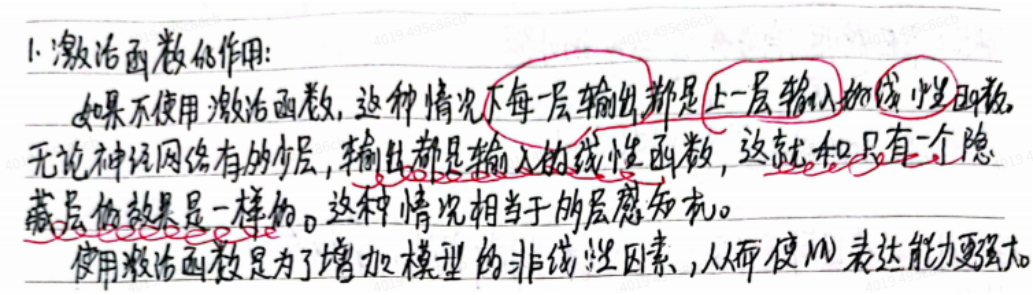
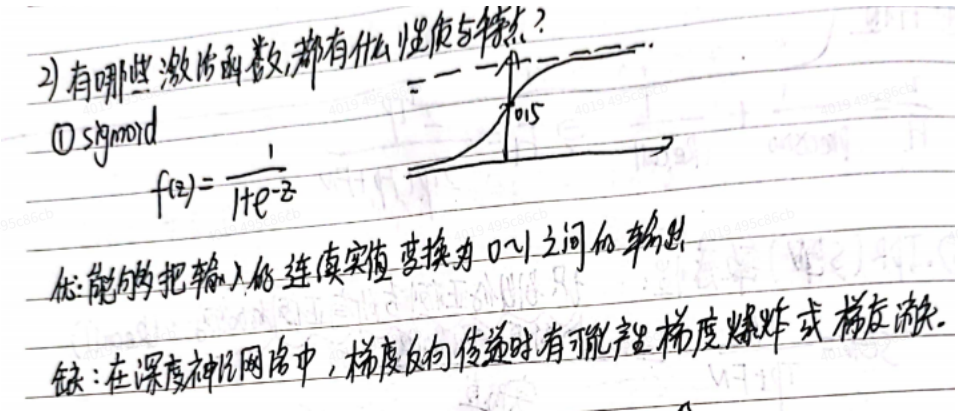
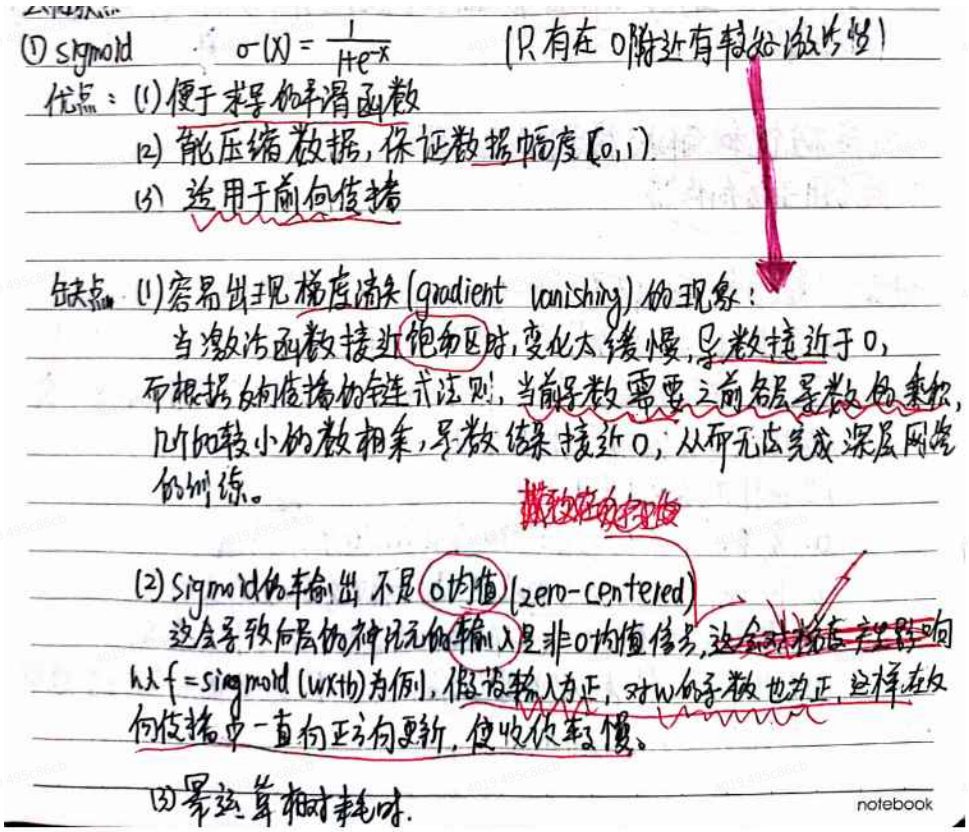
6.激活函数

1.激活函数的作用

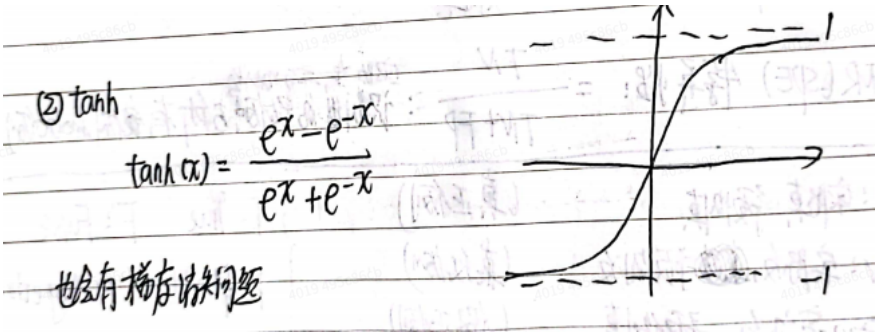


2.Sigmoid





3.Tanh



Date:

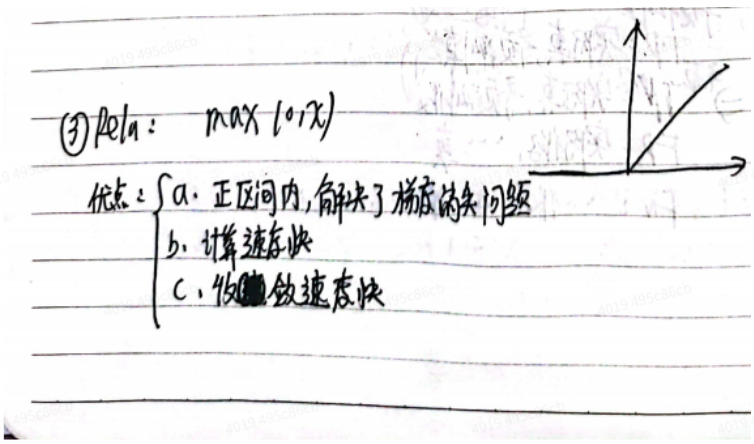
②tanh

·tanh函数将输入值压缩到-1,1的范围，因为它是0均值的。

缺点：存在梯度消失和幂运算的问题

改进：$\tanh(x) = 2\,\text{sigmoid}(x) - 1$

4.ReLU



Date:

④ Leaky ReLU

$f(x) = \max(2x, x)$

优点：

(1) 收敛速度比sigmoid和tanh快。原因是计算复杂度低，不需要进行指数运算。

(2) 梯度不会饱和，解决了梯度消失的问题

(3) 适合用于反向传播

缺点：(1) ReLU的本输出不是zero-centered

(2) 有Dead ReLU Problem（神经元坏死现象）：

某些神经元可能永远不能被激活，导致相应参数永远不能被更新（在负数部分，梯度为0）。

产生这种现象两个原因：

a. 参数初始化问题：可以采用Xavier初始化方法

b. learning rate太高导致在训练过程中参数更新太大；

避免将lr设置过大，或采用adagrad等自动调节lr的算法。

(3) ReLU不会对数据做幅度压缩，所以数据的幅度会随着模型层数的增加不断扩张

⑥ Leaky ReLU函数

$f(x) = \max(0.01x, x)$

用来解决ReLU带来的神经元坏死的问题。

但效果不一定比ReLU好。