

梯度爆炸与梯度消失

考虑一个具有L层, 输入x, 输出0到1的MLP

每一层L由变换 $f_i$ 定义, 记变换的参考为权重 $w^{(l)}$

其递推函数是 $h^{(l)}$  ( $h^{(0)} = x$ )

由于  $h^{(1)} = f_1(h^{(0)}) \Rightarrow 0 = f_1 \circ \dots \circ f_L(x) \Rightarrow loss = g(w)$   
 $= g \circ f_L \circ \dots \circ f_1(x)$

在 $h$ 与 $x$ 都是向量 以向量形式写输入

则更新 $w$ 时  $w_t$ 是 $f_t$ 的参数  $h^{(t)} = f_t(h^{(t-1)})$

$$\frac{\partial g}{\partial w_t} = \frac{\partial g}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \dots \frac{\partial h^{(t+1)}}{\partial h^{(t)}} \frac{\partial h^{(t+1)}}{\partial w_t}$$

考虑MLP最后一层

$h^t = f_t(h^{t-1}) = \sigma(w^t h^{t-1} + b^t)$   $\sigma$ 是 $\sigma$ 函数  
 $= \sigma(w^t h^{t-1})$  忽略 $b^t$  偏置

$\Rightarrow \frac{\partial h^t}{\partial h^{t-1}} = \text{diag}(\sigma'(w^t h^{t-1})) w^t$  ?

ReLU  $\sigma' = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$

$\sigma'(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  与 $x$ 同shape向量

$f = \sigma(w x) \Rightarrow$  求导后  $\text{diag}(\sigma') w \Rightarrow m \times n$   
 $m \times 1 \quad m \times n \quad n \quad m \times m \quad m \times n$

$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} \\ \vdots \\ \frac{\partial f_m}{\partial x_n} \end{pmatrix}_{m \times n}$  ✓

或12神经网络  
 $\prod_{i=t}^{L-1} \frac{\partial h^{i+1}}{\partial h^i} = \prod_{i=t}^{L-1} \text{diag}(\sigma'(w^i h^{i-1})) \cdot w^i$   
 $\approx \prod_{i=t}^{L-1} w^i$

若 $L-t$ 很大, 则 $\frac{\partial \sigma}{\partial w_t}$ 会很大  $d^t$  传播方向  $d^t$  是步长  
 $w_t = w^t - d^t \left( \frac{\partial g}{\partial w} \right) \frac{\partial \sigma}{\partial w_t} \Rightarrow$  导致梯度爆炸 inf  
值超出值域

学习率 $\alpha$ 太大  $\rightarrow$  大参数值  $\rightarrow$  更大的梯度

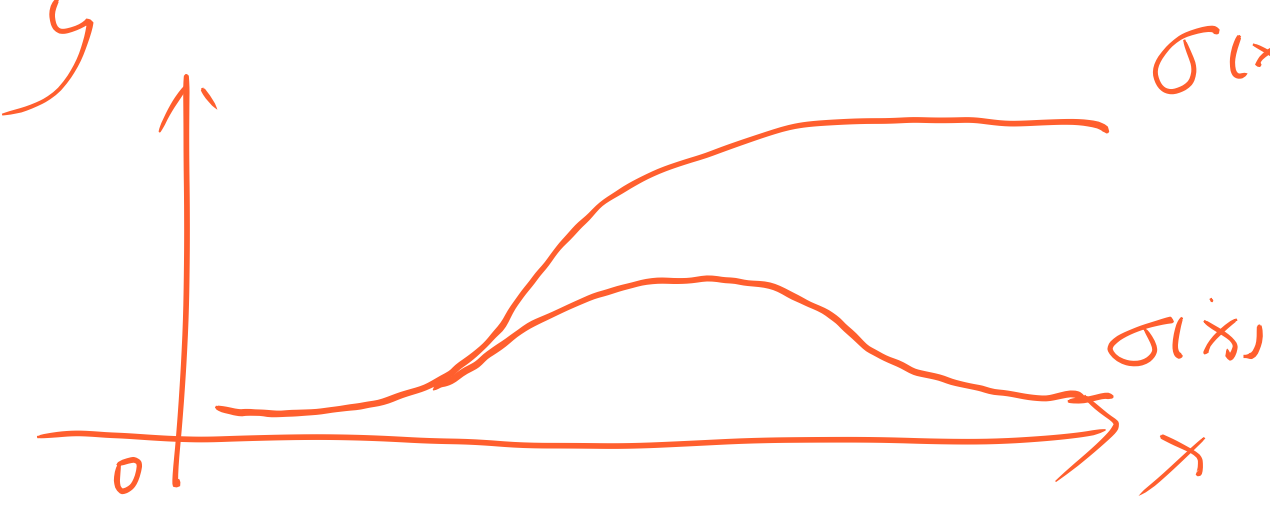
$\alpha > 1 \rightarrow$  训练无法进展

导致 需要在训练过程中动态调整 $\alpha$   $\leftarrow$  自适应学习率

求梯度爆炸

$\sigma = \text{sigmoid}$  时

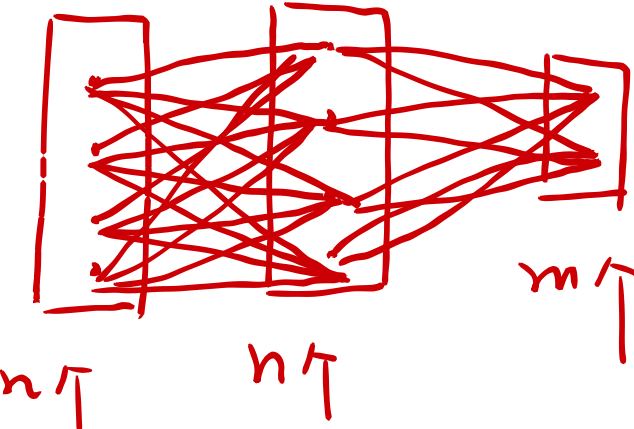
$\sigma(x) = \frac{1}{1+e^{-x}} \quad \sigma'(x) = \sigma(x)(1-\sigma(x))$



$\text{diag}(\sigma'(w^t h^{t-1}))$  很小  $\Rightarrow \frac{\partial \sigma}{\partial h^t}$  很小

2017.11.17 回到感知机 relu 探索

$$h_1 = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$
  
 $n \times 1 \quad n \times n \quad n \times n \quad 1 \times n$

$\Rightarrow$  设计一个网络(分类器 m个类别)  全连接网络

$h_1 = w_1 x + b_1 \quad w_1: n \times n$

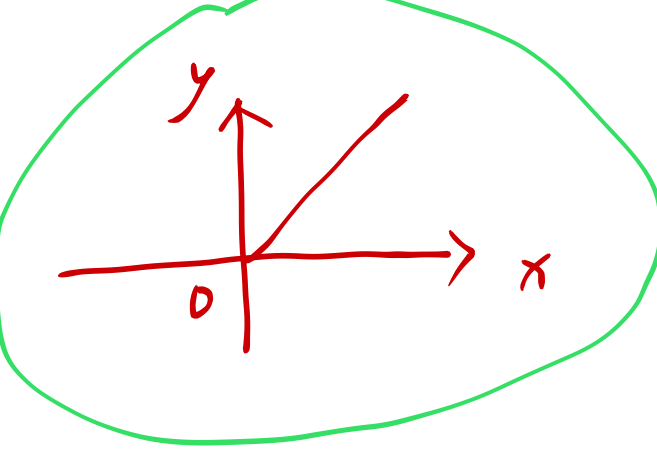
$h_1' = \text{relu}(h_1)$

$h_2 = w_2 h_1' + b_2 \quad w_2: m \times n$

$g = \text{loss}(h_2, y)$

$\frac{\partial g}{\partial x} = \frac{\partial g}{\partial h_2} \frac{\partial h_2}{\partial x}$

分析 $w_1, w_2$   
 $\frac{\partial g}{\partial w_2} = \frac{\partial g}{\partial h_2} \left[ \frac{\partial h_2}{\partial w_2} \right]$ ,  $\frac{\partial g}{\partial h_2}$  的梯度值会比较明显

$\frac{\partial g}{\partial w_1} = \frac{\partial g}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1} \cdot \frac{\partial h_1}{\partial w_1}$   relu图像

若 $h_1(i) < 0$  则 $\frac{\partial h_1}{\partial w_1} = 0$  则 $h_1(i)$ 这个神经元没有参与

从正向传播看

$h_1(i) = b_1(i) + \sum_{j=1}^n w_{1,i,j} x(j) < 0 \Rightarrow h_1'(i) = 0$  相当于 $w_{1,i,j}$  对之后没有作用  
 $j=1, \dots, n$

从反向传播看

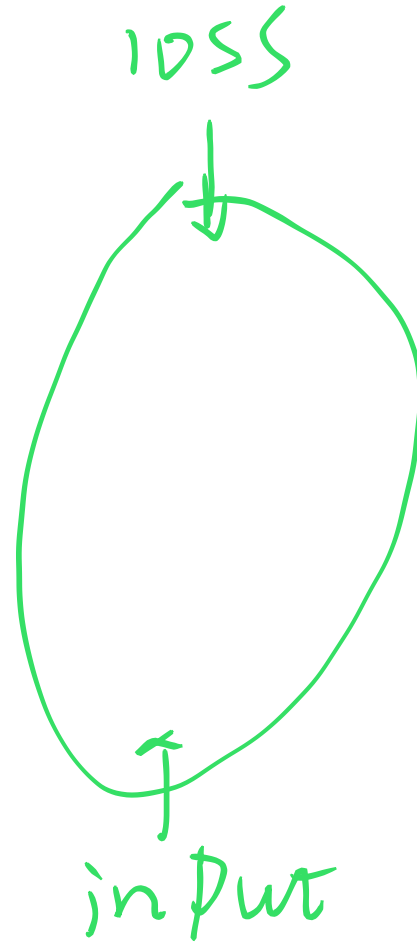
$\frac{\partial g}{\partial w_{1,i,j}} \cdot j=1, \dots, n = \frac{\partial g}{\partial h_2} \frac{\partial h_2}{\partial h_1} \left[ \frac{\partial h_1}{\partial w_{1,i,j}} \right] \frac{\partial h_1}{\partial w_{1,i,j}} = 0$   
为0

$w_{1,i,j} \cdot j=1, \dots, n$  参数不更新

解释梯度消失现象:

第二, relu函数在负半区的导数为0, 所以一旦神经元激活值进入负半区, 那么梯度就会为0, 而正值不变, 这种操作被成为单侧抑制。(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的。)正因为有了这单侧抑制, 才使得神经网络中的神经元也具有了稀疏激活性。尤其体现在深度神经网络模型(CNN)中, 当模型增加N层之后, 理论上ReLU神经元的激活率将降低2的N次方倍。

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, 对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。



$d = -g$  梯度

$w_{t+1} = w_t + d w_t = w_t - d g$

$\frac{\partial g}{\partial w_t} = \frac{\partial g}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \dots \frac{\partial h^{(t+1)}}{\partial h^{(t)}} \frac{\partial h^{(t+1)}}{\partial w_t}$   
 $\Rightarrow$   $t$  增加时  $\frac{\partial g}{\partial w_t}$  会趋于0

当层变化的时候

$(AX)' ? \quad AX = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$   
 $\frac{d(AX)}{dx} = \begin{bmatrix} a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n \\ a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n \\ \vdots \\ a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}$

求导

$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = \begin{bmatrix} \nabla f_1(x) \\ \nabla f_2(x) \\ \vdots \\ \nabla f_n(x) \end{bmatrix}$   
 $A$