习题 4-1

我们有:

$$
\begin{aligned}
z =& w^T X + b \\
a =& \sigma(z)
\end{aligned}
$$

计算梯度:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w} =& \frac{\partial z}{\partial w}\frac{\partial a}{\partial z}\frac{\partial \mathcal{L}}{\partial a} \\
=& X^T \cdot \sigma'(z) \cdot \frac{\partial \mathcal{L}}{\partial a}
\end{aligned}
$$

考虑到:

$$
\begin{aligned}
x_i > 0, \forall x_i \in X \\
\sigma'(z) > 0
\end{aligned}
$$

故:

$$
sign(\frac{\partial \mathcal{L}}{\partial w}) = sign(\frac{\partial \mathcal{L}}{\partial a})
$$

即, 一次更新中, w 中 $w_1, \cdots, w_i$ 的正负号必然相同.
这在大部分情况下会导致参数更新的速率下降.
具体情况如图1所示, 引用自: cs231n lecture7 slides

习题 4-2

定义 XOR 问题如下:
输入:

$$
x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
$$

$$
x_i \in \{0, 1\}
$$

输出:

$$
\hat{y} = \begin{cases} 1 & \text{if } x_1 \neq x_2 \\ 0 & \text{otherwise} \end{cases}
$$

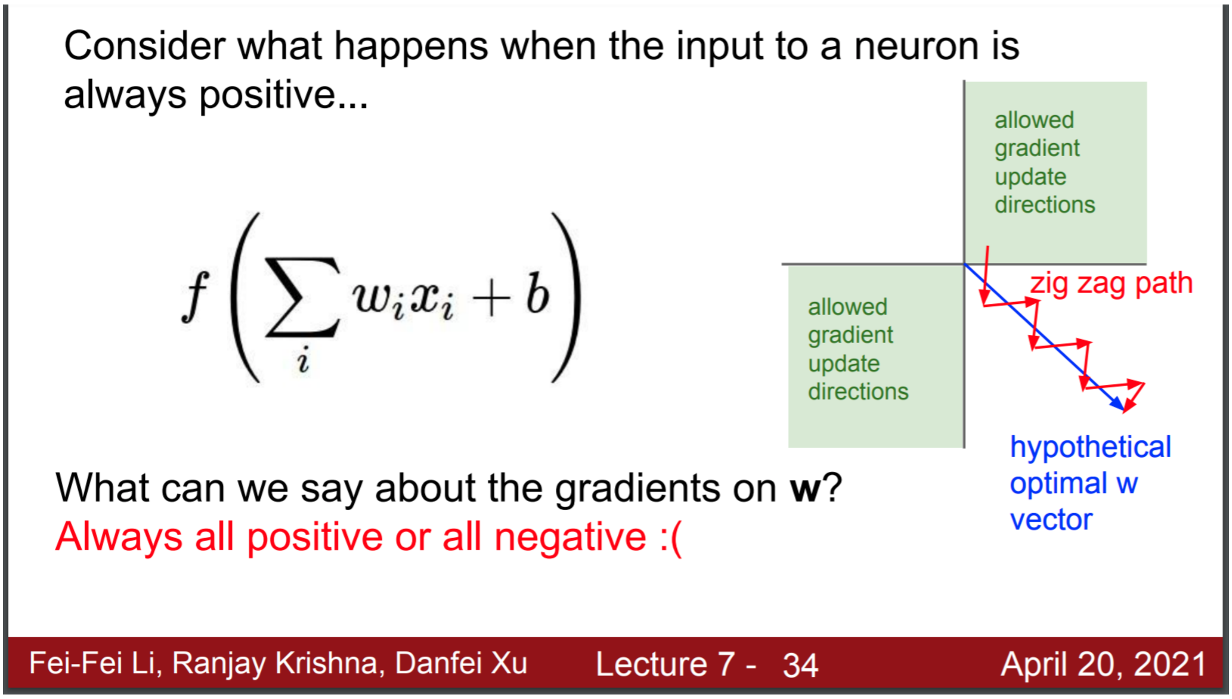

图 1: X 始终同号情况下的梯度更新情况

根据题意, 我们拥有以下模型参数:

$$
\begin{aligned}
w^{(1)} &= \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} \\
b^{(1)} &= \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \end{bmatrix} \\
w^{(2)} &= \begin{bmatrix} w_1^{(2)} & w_2^{(2)} \end{bmatrix} \\
b^{(2)} &= \begin{bmatrix} b^{(2)} \end{bmatrix}
\end{aligned}
$$

故:

$$
\begin{aligned}
z^{(1)} =& w^{(1)}x + b^{(1)} \\
a^{(1)} =& relu(z^{(1)}) \\
\hat{y} =& w^{(2)}a^{(1)} + b^{(2)}
\end{aligned}
$$

则其中一解为:

$$w^{(1)} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$
$$b^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$w^{(2)} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$
$$b^{(2)} = \begin{bmatrix} 0 \end{bmatrix}$$

参数为以上解时, 则:

x	$z^{(1)}$	$a^{(1)}$	$\hat{y}$
$[0,0]^T$	$[0,0]^T$	$[0,0]^T$	0
$[0,1]^T$	$[-1,1]^T$	$[0,1]^T$	1
$[1,0]^T$	$[1,-1]^T$	$[1,0]^T$	1
$[1,1]^T$	$[0,0]^T$	$[0,0]^T$	0

习题 4-3

对第一个隐藏层的第一个神经元, 我们有:

$$\begin{aligned} z_1^{(1)} =& w_1^{(1)} x + b_1^{(1)} \\ a_1^{(1)} =& relu(z_1^{(1)}) \\ =& \max(0, z_1^{(1)}) \end{aligned}$$

现假设, 对训练集中任意 x, 都有 $z_1^{(1)} \leq 0$.
考虑更新参数时计算的梯度:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_1^{(1)}} =& \frac{\partial z_1^{(1)}}{\partial w_1^{(1)}} \frac{\partial a_1^{(1)}}{\partial z_1^{(1)}} \frac{\partial \mathcal{L}}{\partial a_1^{(1)}} \\ =& x^T \cdot \mathbf{1}\{z_1^{(1)} > 0\} \cdot \frac{\partial \mathcal{L}}{\partial a_1^{(1)}} \\ =& x^T \cdot 0 \cdot \frac{\partial \mathcal{L}}{\partial a_1^{(1)}} \\ =& 0 \end{aligned}$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b_1^{(1)}} =& \frac{\partial z_1^{(1)}}{\partial b_1^{(1)}} \frac{\partial a_1^{(1)}}{\partial z_1^{(1)}} \frac{\partial \mathcal{L}}{\partial a_1^{(1)}} \\
=& 1 \cdot \mathbf{1}\{z_1^{(1)} > 0\} \cdot \frac{\partial \mathcal{L}}{\partial a_1^{(1)}} \\
=& 1 \cdot 0 \cdot \frac{\partial \mathcal{L}}{\partial a_1^{(1)}} \\
=& 0
\end{aligned}
$$

考虑更新参数时的情况:

$$
\begin{aligned}
w_1^{(1)} &:= w_1^{(1)} - \alpha \frac{\partial \mathcal{L}}{\partial w_1^{(1)}} \\
&= w_1^{(1)} \\
b_1^{(1)} &:= b_1^{(1)} - \alpha \frac{\partial \mathcal{L}}{\partial b_1^{(1)}} \\
&= b_1^{(1)}
\end{aligned}
$$

这意味着, 在该数据集中, $w_1^{(1)}$ 和 $b_1^{(1)}$ 将永远不会被更新, 且对整个模型毫无作用.
即, 它死亡了.
一种解决方案为: 替换 rule 函数或对其进行修改, 使 $z_1^{(1)} \leq 0$ 时, 通过激活函数依旧能输出一个微小的正数, 以保证进行反向传播时, 其梯度不为 0.

习题 4-5

令参数数量为 n:

$$
n = \frac{N}{L}(M_0 + 1) + (L-1)\frac{N}{L}(\frac{N}{L} + 1) + 1 \cdot (\frac{N}{L} + 1)
$$

其中:

- $\frac{N}{L}(M_0 + 1)$ 为输入层到第一层隐藏层之间的参数
- $(L-1)\frac{N}{L}(\frac{N}{L} + 1)$ 为第一层隐藏层到最后一层隐藏层之间的参数
- $1 \cdot (\frac{N}{L} + 1)$ 为最后一层隐藏层到输出层间的参数

习题 4-7

可以, 但这样做并不一定带来相应的好处.
当我们使用正则化时, 通常是为了避免过拟合. 考虑 $y = wx + b$, 我们可以发现过拟合主要是由 w 对特定数据过于敏感导致的, 而在这一问题中, b 并未扮演重要的角色 (w 是高维向量, 而 b 只是一个单一数值). 故对 b 进行正则化处理并不能造成多少效果.
引用自: Coursera - Improving Deep Neural Networks - Regularization