正向传播和反向传播

一、 正向传播和反向传播

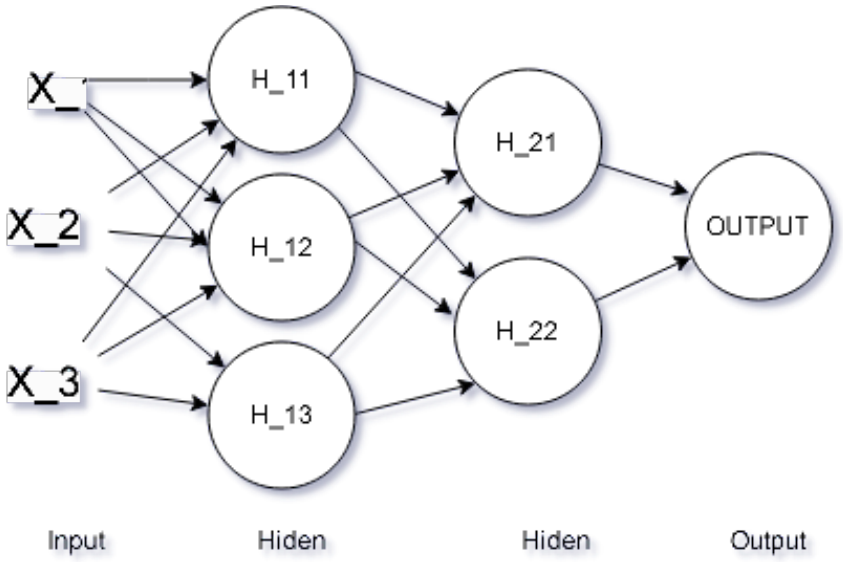

Figure 1 - 1: Net

1.1 正向传播

在上图中有 x_1、x_2、x_3 三个输入，中间的隐藏层有H_1, H_2两个，最后输出。

在神经元计算时，就是

$$
\begin{aligned}
&(x_1 \ \ x_2 \ \ x_3) \cdot \begin{pmatrix} w_{111} & w_{112} & w_{113} \\ w_{121} & w_{122} & w_{123} \\ w_{131} & w_{132} & w_{133} \end{pmatrix}^T \\
&= (x_1 \ \ x_2 \ \ x_3) \cdot \begin{pmatrix} w_{111} & w_{121} & w_{131} \\ w_{112} & w_{122} & w_{132} \\ w_{113} & w_{123} & w_{133} \end{pmatrix} \\
&= \begin{pmatrix} x_1 \cdot w_{111} + x_2 \cdot w_{112} + x_3 \cdot w_{113} \\ x_1 \cdot w_{121} + x_2 \cdot w_{122} + x_3 \cdot w_{123} \\ x_1 \cdot w_{131} + x_2 \cdot w_{132} + x_3 \cdot w_{133} \end{pmatrix} \\
&= \begin{pmatrix} H_{11} \\ H_{12} \\ H_{13} \end{pmatrix}
\end{aligned}
\tag{1 - 1}
$$

$$
\begin{aligned}
&\begin{pmatrix} H_{11} \\ H_{12} \\ H_{13} \end{pmatrix}^T \cdot \begin{pmatrix} w_{211} & w_{212} & w_{213} \\ w_{221} & w_{222} & w_{223} \end{pmatrix}^T \\
&= (H_{11} \ \ H_{12} \ \ H_{13}) \cdot \begin{pmatrix} w_{211} & w_{221} \\ w_{212} & w_{222} \\ w_{213} & w_{223} \end{pmatrix} \\
&= \begin{pmatrix} H_{11} \cdot w_{211} + H_{12} \cdot w_{212} + H_{13} \cdot w_{213} \\ H_{11} \cdot w_{221} + H_{12} \cdot w_{222} + H_{13} \cdot w_{223} \end{pmatrix} \\
&= \begin{pmatrix} H_{21} \\ H_{22} \end{pmatrix}
\end{aligned} \tag{1 - 2}
$$

$$
\begin{aligned}
&(H_{21} \ \ H_{22}) \cdot (w_{321} \ \ w_{322})^T \\
&= (H_{21} \cdot w_{21} + H_{22} \cdot w_{22}) \\
&= (\mathrm{OUTPUT})
\end{aligned} \tag{1 - 3}
$$

从左到右的计算。在全连接神经网络中，每一层的每个神经元都会与前一层的所有神经元或者输入数据相连，每一个神经元的输出=使用激活函数激活前一层函数的累加和。

1.2 反向传播

反向传播算法(Backpropagation，简称 BP 算法)。P 算法的学习过程由正向传播过程和反向传播过程组成。

在正向传播过程中，输入信息通过输入层经隐含层，逐层处理并传向输出层。如果预测值和教师值不一样，则取输出与期望的误差的平方和作为损失函数（损失函数有很多，这是其中一种）。将正向传播中的损失函数传入反向传播过程，逐层求出损失函数对各神经元权重的偏导数，作为目标函数对权重的梯度。根据这个计算出来的梯度来修改权重，网络的学习在权重修改过程中完成。误差达到期望值时，网络学习结束。

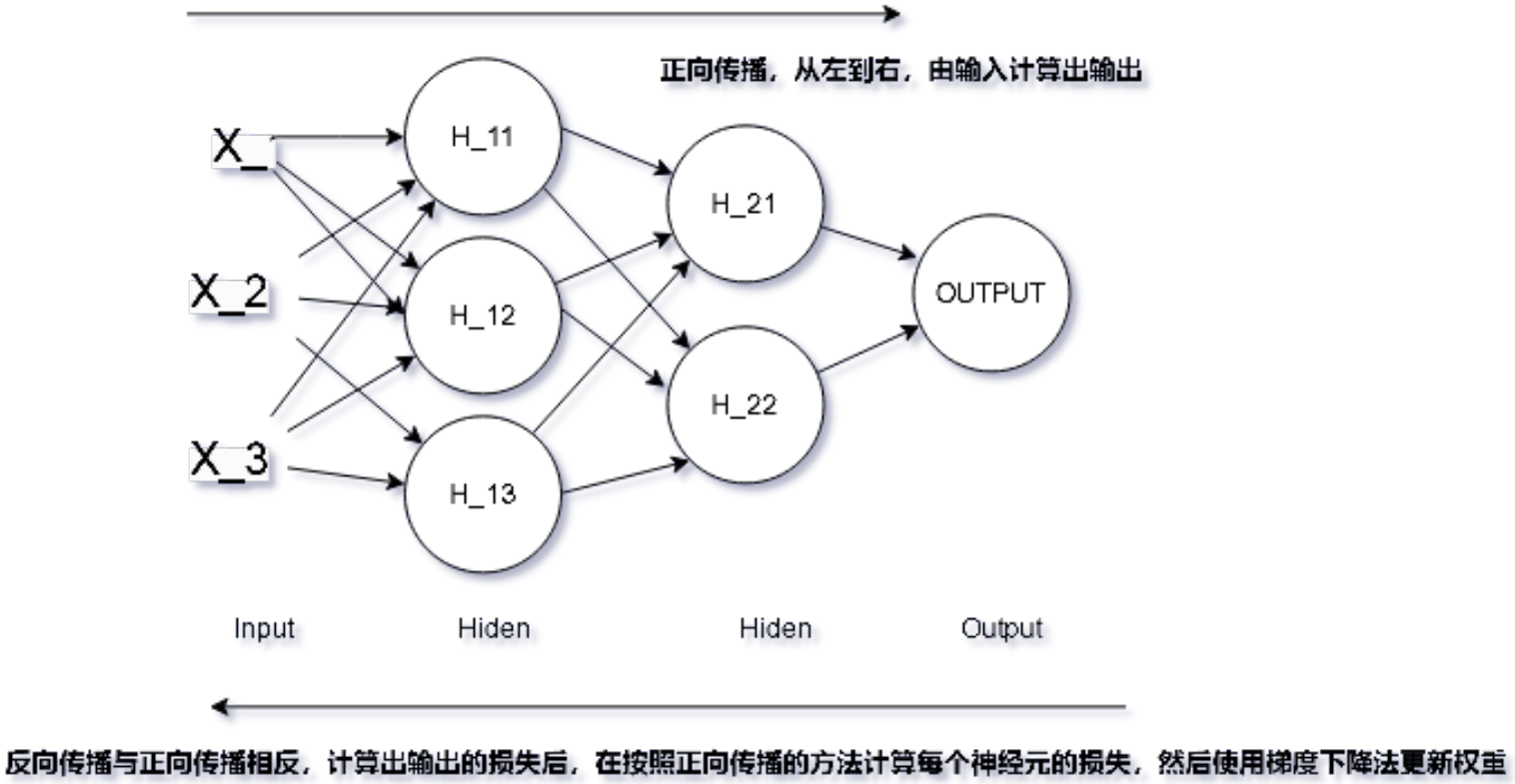

Figure 1 - 2: Net

1.2.1 误差计算

输出的误差可以写为 $\delta = \text{cost}(\text{OUTPUT})$，通过反向传播

$$\begin{aligned} \delta_{H_{21}} &= \delta \cdot w_{321} \\ \delta(H_{22}) &= \delta \cdot w_{322} \end{aligned} \tag{1 - 4}$$

$$\begin{aligned} \delta_{H_{11}} &= \delta_{H_{21}} \cdot w_{211} + \delta_{H_{22}} \cdot w_{221} \\ \delta_{H_{12}} &= \delta_{H_{21}} \cdot w_{212} + \delta_{H_{22}} \cdot w_{222} \\ \delta_{H_{13}} &= \delta_{H_{21}} \cdot w_{213} + \delta_{H_{22}} \cdot w_{223} \end{aligned} \tag{1 - 5}$$

1.2.2 梯度下降

最后使用梯度下降法来更新权重

梯度下降可以表示为：

$$\theta = \theta - \alpha \frac{\partial(J(\theta))}{\partial(\theta)} \tag{1 - 6}$$

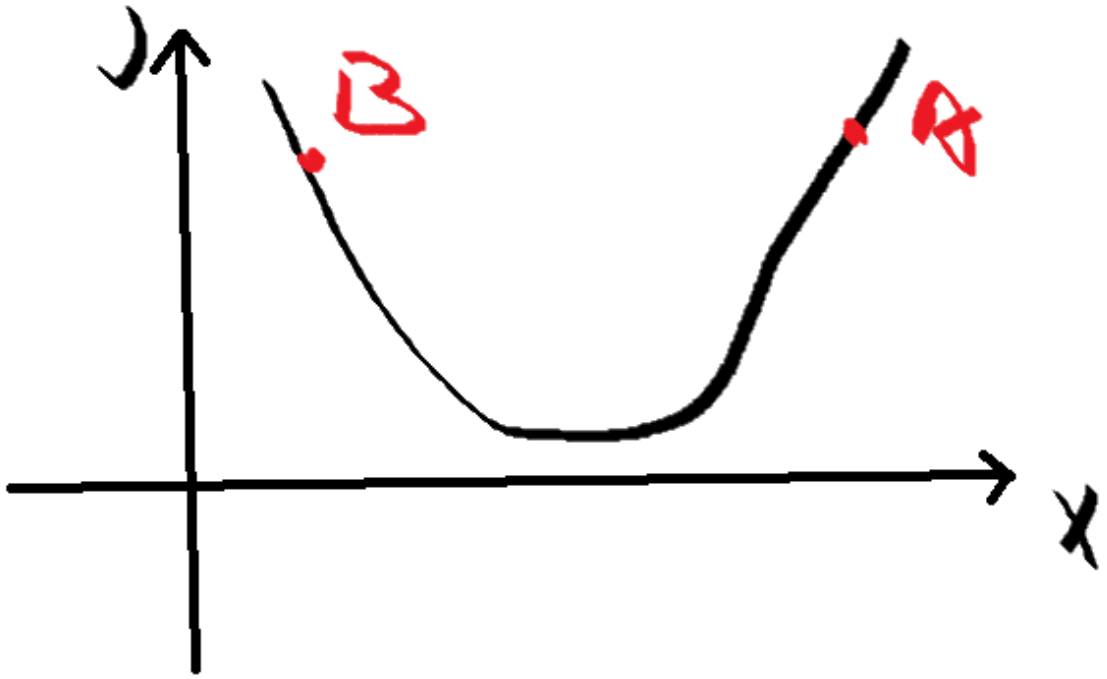

Figure 1 - 3: 梯度下降

可以看出在 A 点时，函数单调递增，导数为正，减去学习率乘导数，x 逐渐向最小值靠近；在 B 点时，函数单调递减，导数为负，这样就相当于加上学习率乘导数，x 向右移动，逐渐向最小值逼近。

$$
\begin{aligned}
w_{111} &= w_{111} - \alpha_{11}\theta_{H_{11}}\frac{\partial(H_{11}(a))}{\partial(a)}x_1 \\
w_{211} &= w_{211} - \alpha_{12}\theta_{H_{21}}\frac{\partial(H_{21}(b))}{\partial(b)}a \\
w_{311} &= w_{311} - \alpha_{13}\theta_{\mathrm{OUTPUT}}\frac{\partial(\mathrm{OUTPUT}(c))}{\partial(c)}b
\end{aligned}
\tag{1 - 7}
$$

二、 引用

1. 《机器学习笔记丨神经网络的反向传播原理及过程（图文并茂+浅显易懂） 神经网络反向传播原理-CSDN 博客》. 见于 2024 年 7 月 11 日. https://blog.csdn.net/fsfjdtpzus/article/details/106256925.
2. 《小白零基础学习：详解梯度下降算法：完整原理+公式推导+视频讲解 标准梯度下降算法 损函数-CSDN 博客》. 见于 2024 年 7 月 11 日. https://blog.csdn.net/zhouaho2010/article/details/102756411.