# 深度神经网络极简介

许志钦 xuzhiqin@sjtu.edu.cn

2023 年 8 月 16 日

神经网络是一个有许多参数的机器学习方法。假设我们有一组训练集 $S=\{(x_i,y_i)\}_{i=1}^n,(x_i,y_i)\in\mathbb{R}^2$。考虑一个只有一层隐藏层的神经网络，即两层神经网络，

$$f_\theta(x)=\sum_{j=1}^m a_j\sigma(w_jx+b_j).$$

为了方便起见，定义

$$h_i=f_\theta(x_i). \tag{1}$$

常见的激活函数 $\sigma(x)$ 为 $\mathrm{ReLU}(x)=\max\{0,x\}$、$\tanh(x)$ 等。我们这里假设 $x$, $w$, $b$, $a$ 都是一维的标量，对于高维的情形，后面会有介绍。我们可以通过（随机）梯度下降来调节参数。当梯度下降用在神经网络模型中时，它有一种特殊的名字：向后传播法。考虑训练的损失函数为均方差:

$$L_S=\frac{1}{2n}\sum_{i=1}^n(y_i-h_i)^2. \tag{2}$$

$a_j$ 是最外层的参数，它的求导比较简单：

$$\frac{\partial L_S}{\partial a_j}=\frac{1}{n}\sum_{i=1}^n(h_i-y_i)\sigma(w_jx+b_j).$$

对于 $w_j$，由于它不是最外层的参数，我们需要多次用到链式求导法则：

$$\begin{aligned}
\frac{\partial L_S}{\partial w_j}&=\frac{1}{n}\sum_{i=1}^n(h_i-y_i)\frac{\partial h_i}{\partial w_j}\\
&=\frac{1}{n}\sum_{i=1}^n(h_i-y_i)\frac{\partial(\sum_{l=1}^m a_l\sigma(w_lx_i+b_l))}{\partial w_j}\\
&=\frac{1}{n}\sum_{i=1}^n(h_i-y_i)a_j\frac{\partial(\sigma(w_jx_i+b_j))}{\partial w_j}\\
&=\frac{1}{n}\sum_{i=1}^n(h_i-y_i)a_j\frac{\partial\sigma(x)}{\partial x}|_{x=w_jx_i+b_j}\frac{\partial(w_jx_i)}{\partial w_j}\\
&=\frac{1}{n}\sum_{i=1}^n(h_i-y_i)a_j\frac{\partial\sigma(x)}{\partial x}|_{x=w_jx_i+b_j}x_i.
\end{aligned}$$

因为链式法则，梯度下降的计算顺序是 $h_i\to a_j\to\sigma\to x_i$，而信息流 (information flow) 的顺序是 $x_i\to\sigma\to a_j\to h_i$，这两者的顺序相反，因此称其为向后传播。优化后，通过取一些新的采样点计算测试误差来判断拟合的好坏。

下面用矩阵的形式写 $f_\theta(\boldsymbol{x})$。

$$f_\theta(\boldsymbol{x}) = \boldsymbol{W}^{[2]}\sigma\circ(\boldsymbol{W}^{[1]}\boldsymbol{x}+\boldsymbol{b}^{[1]}),$$

其中 $\boldsymbol{x}\in\mathbb{R}^{d\times 1}, \boldsymbol{W}^{[1]}\in\mathbb{R}^{m\times d}, \boldsymbol{b}^{[1]}\in\mathbb{R}^{m\times 1}, \boldsymbol{W}^{[2]}\in\mathbb{R}^{d_o\times m}$，“$\circ$” 的意思是对应元素的运算 (entry-wise operation)（例如对应元素相乘）。这里 $d$ 是输入数据的维度，$m$ 是隐藏层神经元的个数，$d_o$ 是输出维度。有时我们会同时计算 $n$ 个样本的值，比如在实际编程计算中，把 $\boldsymbol{x}$ 写成矩阵形式：

$$\boldsymbol{Y} = h(\boldsymbol{X}) = \boldsymbol{W}^{[2]}\sigma\circ(\boldsymbol{W}^{[1]}\boldsymbol{X}+\boldsymbol{B}^{[1]}),$$

其中 $\boldsymbol{X}\in\mathbb{R}^{d\times n}, \boldsymbol{Y}\in\mathbb{R}^{[d_o\times n]}, \boldsymbol{B}^{[1]}\in\mathbb{R}^{m\times n}$。$\boldsymbol{B}^{[1]}$ 的每一列都为 $b^{[1]}$，即 $\boldsymbol{B}^{[1]}=[b^{[1]},b^{[1]},\cdots,b^{[1]}]$。用这些符号可以类似得定义一个深度神经网络。

一个 $L$ 层神经网络记为

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}^{[L-1]}\sigma\circ(\boldsymbol{W}^{L-2}\sigma\circ(\cdots(\boldsymbol{W}^{[1]}\sigma\circ(\boldsymbol{W}^{[0]}\boldsymbol{x}+\boldsymbol{b}^{[0]})+\boldsymbol{b}^{[1]})\cdots)+\boldsymbol{b}^{[L-2]})+\boldsymbol{b}^{[L-1]}, \tag{3}$$

其中 $\boldsymbol{W}^{[l]}\in\mathbb{R}^{m_{l+1}\times m_l}, \boldsymbol{b}^{[l]}\in\mathbb{R}^{m_{l+1}}, m_0=d_{in}=d, m_L=d_o$，$\sigma$ 是一个向量函数。注意一下在计算网络层数时，输入层不计入（即层数为隐藏层 + 输出层的层数）我们把参数集记为

$$\boldsymbol{\theta} = (\boldsymbol{W}^{[0]},\boldsymbol{W}^{[1]},\cdots,\boldsymbol{W}^{[L-1]},\boldsymbol{b}^{[0]},\boldsymbol{b}^{[1]},\cdots,\boldsymbol{b}^{[L-1]}),$$

把 $\boldsymbol{W}^{[l]}$ 中的元素记为 $\boldsymbol{W}_{ij}^{[l]}$。也可以用递归的方法定义这个网络：

$$f_{\boldsymbol{\theta}}^{[0]}(\boldsymbol{x}) = \boldsymbol{x} \tag{4}$$

$$f_{\boldsymbol{\theta}}^{[l]}(\boldsymbol{x}) = \sigma\circ(\boldsymbol{W}^{[l-1]}f_{\boldsymbol{\theta}}^{[l-1]}(\boldsymbol{x})+\boldsymbol{b}^{[l-1]}), \quad 1\le l\le L-1 \tag{5}$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{x}) = \boldsymbol{W}^{[L-1]}f_{\boldsymbol{\theta}}^{[L-1]}(\boldsymbol{x})+\boldsymbol{b}^{[L-1]} \tag{6}$$