神经网络

神经网络近几年飞速发展，让我们来看一看它是什么。

什么是神经网络

神经网络是由神经元构成的。而神经元又主要由2种计算构成：加权求和和非线性变换（又叫激活函数）。下面我们就一步一步从神经元到神经网络：

单个神经元

例如一个数据有3个输入$X=[x_1,x_2,x_3]^\top$，那么神经元的数学模型就是：

1. 加权求和：

$$z=w_1x_1+w_2x_2+w_3x_3$$

2. 激活函数

$$a=f(z)$$

   其中$f(\cdot)$是一个非线性的函数即可。常见的有ReLU, tanh, sigmoid等函数。其实，各种各样的激活函数，只要是非线性的都可以。当然这些函数的效果有好有坏。我们以sigmoid为例，那么$a=\frac{1}{1+e^{-z}}$。

那么a就是这个神经元的输出。

为了增加这个模型的表现能力，往往会增加一个偏移量b，那么整个神经元就可以描述成：

$$\begin{aligned} z &= w_1x_1+w_2x_2+w_3x_3+b \\ a &= \frac{1}{1+e^{-z}} \end{aligned}$$

后面为了表述方便，我们还是把$\frac{1}{1+e^{-z}}$写成$f(z)$的形式。

多个神经元

上面的例子是一个神经元的例子，我们可以使用多个(I个)神经元，那么对于第i个神经元，它的数学模型就是

$$\begin{aligned} z_i &= w_{i,1}x_1+w_{i,2}x_2+w_{i,3}x_3+b_i \\ a_i &= f(z_i) \end{aligned}$$

我们可以用线性代数的方法简化这个表述：

$$\begin{aligned} \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_I \end{bmatrix}}_{\boldsymbol{z}} &= \begin{bmatrix} w_{1,1}x_1+w_{1,2}x_2+w_{1,3}x_3+b_1 \\ \vdots \\ w_{I,1}x_1+w_{I,2}x_2+w_{I,3}x_3+b_I \end{bmatrix} = \underbrace{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & b_1 \\ \vdots & & & \\ w_{I,1} & w_{I,2} & w_{I,3} & b_I \end{bmatrix}}_{W} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}}_{\boldsymbol{x}} \\ \begin{bmatrix} a_1 \\ \vdots \\ a_I \end{bmatrix} &= f\left(\begin{bmatrix} z_1 \\ \vdots \\ z_I \end{bmatrix}\right) \end{aligned}$$

也就是

$$\begin{aligned} \boldsymbol{z} &= W\cdot\boldsymbol{x} \\ \boldsymbol{a} &= f(\boldsymbol{z}) \end{aligned}$$

这里我们可以看到，通过把x扩展出一个1，就可以把偏移b合并到权重W里面，进而简化公式的表达。这样，在推导公式和梯度的时候就可以少考虑一个变量。

多个神经元+多条数据

上面的例子中有I个神经元，但是只有一个数据，每个数据由3个特征组成。在实践中，数据的形式往往是$X\in\mathbb{R}^{E\times M}$，其中$E$是数据的个数，$M$才是特征，也就是说

$$X=\begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^E & x_2^E & x_3^E & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{1^\top} \\ \vdots \\ \boldsymbol{x}^{E^\top} \end{bmatrix}$$

这里的上标$1,\dots,E$指的是第几个数据。因此我们需要把1.1.2的公式转置一下，得到

$$\begin{aligned} Z &= X\cdot W^\top \\ A &= f(Z) \end{aligned}$$

其中$X\in\mathbb{R}^{E\times M}$，$W\in\mathbb{R}^{I\times M}$，$Z\in\mathbb{R}^{E\times I}$，$A\in\mathbb{R}^{E\times I}$。

多层神经元

上面的I个神经元是同时接受输入数据X的，所以我们把它们称为一层M个输入I个输出的神经网络。然而，神经网络的丰富表达能力往往来自于深层的网络。因此，我们再增加一层网络，这个网络接收第一层网络的输出。为了方便记录，我们用大写字母的下标表示第几层，那么第一层的数学模型就是：

$$\begin{aligned} Z_1 &= X\cdot W_1^\top \\ A_1 &= f(Z_1) \end{aligned}$$

第二层的原理和第一层一样，只不过它接受的信息不是$X\in\mathbb{R}^{E\times M}$而是$A_1\in\mathbb{R}^{E\times I_1}$，所以：

$$\begin{aligned} Z_2 &= A_1\cdot W_2^\top \\ A_2 &= f(Z_2) \end{aligned}$$

其中$A_1\in\mathbb{R}^{E\times I_1}$，$W_2\in\mathbb{R}^{I_2\times I_1}$，$Z_2\in\mathbb{R}^{E\times I_2}$，$A_2\in\mathbb{R}^{E\times I_2}$。

借由这个原理，人们就可以不断的增加神经网络的层数。唯一需要注意的就是上一层的输出数量需要等于下一层的输入数量。

不过，第一层的输入必须等于X的特征数量M，最后一层的输出需要等于真实值Y的维度N。由于这个特殊限制，我们往往叫第一层为输入层，最后一层为输出层。输入层和输出层之间的叫做隐藏层。要注意的是，隐藏层并非是必须的。

总结

可以看出，神经网络并没有什么神秘的地方，它只是大量神经元的堆积，而且神经元内部的计算也十分简单，也就是加权求和和一个非线性变换，仅此而已。

当然，这并不意味着神经网络十分肤浅。因为在这个最基本的结构上，人们可以做大量的改进和变换。

为什么神经网络变得热门

可以看出，神经网络不过是一大堆简单计算的堆积，为什么神经网络很热门呢？原因有以下几点：

- 智能设备的普及使得可供获取的数据急剧增加。这对于数据驱动的算法来说至关重要。
- 硬件计算能力的提升
- 简单的计算结构（加权求和+非线性变换）。这不仅使得入门简单，更极大的简化了优化过程。
- 强大的表现能力：尽管计算简单，但是利用多层神经网络，依然能表达出丰富的信息[1]

[1] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.

下面我们会讲到神经网络中的反向传播，这就是神经网络热门的核心原因之一。

然后用自己的代码（不用PyTorch）搭建一个最基本的神经网络。然后再讲解一下数据预处理和数据集。

In [ ]: