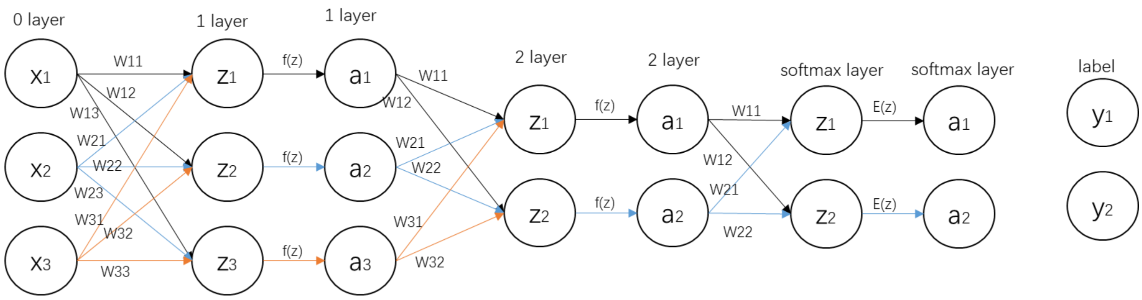
神经网络的原理分析

本文给出的神经网络结构为全连接网络+反向传播算法(BP)构建的，最后基于 Softmax 分类器进行分类，其中的原理推导也是基于上述结构。



多层感知机结构图

如上图所示，对于一个样本 x,其中有三个元素 X0=[x1,x2,x3],我们认为其有三个神经元。通过矩阵 W1 的映射得到 Z1=[z1,z2,z3]，再将 Z1 通过激活函数 f(x)可以得到 A1=f(Z1)=[a1,a2,a3]. 此时 A1 作为第一层网络的输入数据，第二层网络的输入数据，通过第二层网络的特征矩阵参数 W2,被映射为 Z2=[z1,z2],再基于激活函数 f(x) 得到 A2=f(Z2)=[a1,a2].

此时的 A2=[a1,a2]就是多层感知机的输入数据。我们将 A2 输入到 softmax 分类器当中进行分类，分析 softmax 原理，其实相当于在多层感知机当中再增加一层，只是激活函数与前层网络的激活函数不同。多层感知机的激活函数，一般使用 sigmoid,但是如果网络的深度太深，sigmoid 函数会出现梯度消失的现象，后来的深度神经网络一般会使用 ReLu 函数替代 sigmoid，解决梯度消失的现象。但是 softmax 层激活函数与 sigmoid 不同，是一个概率值函数的$\mathrm{E}(z_i)=\exp(z_i)/(\sum_{k=1}^{N(z)}\exp(z_k))$,所以在求导的时候会有一点区别。

此外，我们使用交叉熵替代原先的距离差作为损失函数，因为 softmax 输入的概率，而交叉熵表示得是输出概率和期望概率之间的距离,二者相差巨大，交叉熵越大，其计算方法如下所示：

$$\mathrm{E}=-\sum_x y(x)*\ln(p(x))$$

在本文网络结构中，损失函数的交叉熵可以表示为：

$$\mathrm{E}=-\sum_{k=1}^{K} y_k*\ln(a_k)$$

现在上图就构成了一个完整的神经网络模型，全链接感知机模型。现在需要考虑的是，如何在训练过程更新特征参数 W1，W2，W3. 这里我们需要介绍一个算法——反向传播算法(BP).

上述通过给定的样本计算出损失函数的过程称为正向传播,而通过损失函数的更新特征参数的过程就是反向传播了。

在介绍反向传播之前，我们需要知道一个求导法则——链式法则。

举个栗子：$\frac{\partial G(f(x))}{\partial x}=\frac{\partial G}{\partial f}*\frac{\partial f}{\partial x}$

回来本网络结构当中，如果已知输出数据 A1=[a1,a2,a3],如何对 w11 求导？

$$\mathrm{a1}=\mathrm{f(z1)}=\mathrm{f(x1*w11+x2*w21+x3*w31)}$$

$$\frac{\partial a1}{\partial w11}=\frac{\partial a1}{\partial z1}*\frac{\partial z1}{\partial w11}=f'(z1)*x1$$

其中$f'(z1)$就是激活函数的导数。

计算 w11 导数后，基于梯度下降法，可以对 w11 进行更新：$\mathrm{w11}=\mathrm{w11}-\partial\mathrm{w11}$.

深吸一口气，现在我们需要基于链式法则和反向传播算法计算整个网络特征参数。为了简化计算，我们会保存正向传播和反向传播计算过程的数据。

首先计算 softmax 层的特征参数 w11:

$$\frac{\partial E}{\partial w11}=\frac{\partial(-y1*\ln(a))}{\partial w11}=\frac{\partial E}{\partial z1}*\frac{\partial z1}{\partial w11}$$

我们令$\frac{\partial E}{\partial z1}=\delta_o$, 对于$\frac{\partial z1}{\partial w11}$的计算非常简单，即$\frac{\partial z1}{\partial w11}=a1(2\ \ layer)$

但是因为:

$$\mathrm{ai}=\frac{e^{zi}}{\sum_{k=1}^{K}e^{zk}}$$

所以 w11 不仅仅会影响到 z1, 进而影响到 a1, 但是 a2 也同样会受到 z1 的影响，进而被 w11 所影响。

对$\frac{\partial a1}{\partial z}$的求导不仅涉及到 a1，还涉及到 a2, 所以不能像普通的激活函数一般, 计算过程如下：

$$\delta_o=\frac{\partial E}{\partial z1}=\sum_{k=1}^{K}\frac{\partial E}{\partial ak}*\frac{\partial ak}{\partial z1}=\frac{\partial E}{\partial a1}*\frac{\partial a1}{\partial z1}+\sum_{k\neq1}^{K}\frac{\partial E}{\partial ak}*\frac{\partial ak}{\partial z1}$$

对于前一项的计算，我们令 A=$\sum_{k=1}^{K}e^{zk}$, 对，就是那个分母:

$$\frac{\partial E}{\partial a1}=-\frac{y1}{a1}$$

$$\frac{\partial a1}{\partial z1}=\frac{\partial(\frac{e^{z1}}{\sum_{k=1}^{K}e^{zk}})}{\partial z1}=\frac{e^{z1}(A-e^{z1})}{A^2}$$

此时前一项就等于:

$$\left(-\frac{y1}{a1}\right)*\frac{e^{z1}(A-e^{z1})}{A^2}=-\frac{y1*(A-e^{z1})}{A}=-\mathrm{y1}+\mathrm{y1}*\frac{e^{z1}}{A}$$

对于后一项当中的求导为:

$$\sum_{k\neq1}^{K}\frac{\partial E1}{\partial ak}*\frac{\partial ak}{\partial z1}=\sum_{k\neq1}^{K}-\frac{yk}{ak}*\frac{-e^{zk}}{A^2}*e^{z1}=\sum_{k\neq1}^{K}\frac{yk}{A}*e^{z1}$$

此时，我们把前后两项加在一起，则可以得到:

$$\delta_o=\frac{\partial E}{\partial z1}=-\mathrm{y1}+\mathrm{y1}*\frac{e^{z1}}{A}+\sum_{k\neq1}^{K}\frac{yk}{A}*e^{z1}=-\mathrm{y1}+e^{z1}*\sum_{k=1}^{K}\frac{yk}{A}$$

因为标签 y=[0, 0, 0…1, 0]肯定只有 1label, 所以$\sum_{k=1}^{K}yk=1$. 因此:

$$\delta_o=\frac{\partial E}{\partial z1}=\frac{e^{z1}}{A}-y1=\mathrm{a1}-\mathrm{y1}$$

同理，

$$\delta_1=\frac{\partial E}{\partial z2}=\frac{e^{z2}}{A}-y2=\mathrm{a2}-\mathrm{y2}$$

至此，我们扩展到一般情况，对于以 softmax 作为分类器的网络模型，根据 BP 和链式

法则计算其特征参数的导数∇w_{ij}^L可以表示为:

$$\nabla w_{ij}^L=\frac{\partial E}{\partial w_{ij}^L}=\frac{\partial E}{\partial z_j^L}*\frac{\partial z_j^L}{\partial w_{ij}^L}=\delta_j*a_i^{L-1}=\left(y_j-a_j^L\right)*a_i^{L-1}$$

其中a_i^{L-1}表示 softmax 层第 i 个输入数据，也是上一层网络的输出数据。

现在我计算完 softmax 层的特征参数的导数，那么对于整个网络当中其他参数的求导怎么计算呢?

因为无论是全连接网络还是卷积神经网络，其激活函数一般都是 sigmoid，或者是 ReLu，其求导计算方式都不像 E(x)那样复杂。

对于神经网络任意l层，其特征参数为w_{ij}^l可以按照如下方式计算:

$$\nabla w_{ij}^l=\frac{\partial E}{\partial w_{ij}^l}=\frac{\partial E}{\partial z_j^l}*\frac{\partial z_j^l}{w_{ij}^l}=\delta_j^l*\frac{\partial z_j^l}{w_{ij}^l}=\sum_{k=1}^{in_{N(l+1)}}\left(\frac{\partial E}{\partial z_k^{l+1}}*\frac{\partial z_k^{l+1}}{\partial a_j^l}\right)*\frac{\partial a_j^l}{\partial z_j^l}*\frac{\partial z_j^l}{w_{ij}^l}$$

$$=\sum_{k=1}^{in_{N(l+1)}}\left(\delta_k^{l+1}*w_{jk}^{l+1}\right)*f'\left(z_j^l\right)*a_i^{l-1}=\delta_j^l*a_i^{l-1}$$

$$\delta_j^l=\sum_{k=1}^{in\_N(l+1)}\left(\delta_k^{l+1}*w_{jk}^{l+1}\right)*f'\left(z_j^l\right)$$

对上述计算公式做一个简单的描述：根据反向传播，当我们需要计算δ_j^l的时候，我们需要计算本层的输出数据 a[⋯]和下一层网络之间的联系，因为是全连接网络，所以 a1 会影响到下一层网络的 z1，z2⋯zk，所以计算的时候需要都加上。每一个节点 z，都对应着一个δ。

由上式可以发现，当我们计算任一层网络的特征参数∇w_{ij}^l时，只需要计算δ_j^l和a_i^{l-1}，而δ_j^l的计算和上一层网络的δ_k^{l+1}有关，由此我们通过反向传播的思想，递归计算，极大得简化计算过程。

对于偏置参数的计算:

$$\nabla b_i^l=\frac{\partial E}{\partial z_i^l}*\frac{\partial z_i^l}{\partial b_i^l}=\delta_i$$

现在整个全连接神经网络的 BP 的所有公式推导完毕了，总结一下，总共有四个重要公式:

(1) softmax 层的δ计算公式:

$$\delta_i^L=\left(y_i-a_i^L\right)$$

(2) 中间隐藏层的δ计算公式:

$$\delta_j^l=\sum_{k=1}^{in\_N(l+1)}\left(\delta_k^{l+1}*w_{jk}^{l+1}\right)*f'\left(z_j^l\right)$$

(3) 参数 w 和 b 的导数:

$$\nabla w_{ij}^l=\delta_j^l*a_i^{l-1}$$
$$\nabla b_i^l=\delta_i^l$$

为了简便计算，我们将其写成向量形式，则有:

$$\boldsymbol{\delta}^L=\mathbf{y}-\boldsymbol{a}^L$$

$$\boldsymbol{\delta}^l=\left((\boldsymbol{W}^{l+1})^T*(\boldsymbol{\delta}^{l+1})\right)\odot f'(\boldsymbol{z}^l)$$

$$\nabla\boldsymbol{W}^l=\boldsymbol{\delta}^l*(\boldsymbol{a}^{l-1})^T$$
$$\nabla\boldsymbol{b}^l=\boldsymbol{\delta}^l$$