前言

看吴恩达老师*deeplearning.ai*课程中*week3*视频讲到多样本神经网络反向传播推导时, 又是一脸懵逼, 卡了好几天才搞明白, 把推导过程再和大家分享一下.

本人也是渣渣初学者, 如果对文章有任何疑问或希望转载, 请联系ch_yan@pku.edu.cn

如果读完后觉得有所收获, 请在我的github里点个star吧 ~

前情提要

考虑L层的神经网络和由m个样本,n个维度组成的数据集, 约定:

1. 第l层的神经元数为$n^{(l)}$. 显然$0 \leqslant l \leqslant L$且$n^{(0)} \equiv n$.
2. m个样本在第l层的输出分别为$a^{(l)(1)}, a^{(l)(2)}, \dots, a^{(l)(m)}$. 写成矩阵形式: $A^{(l-1)} = [\,a^{(l-1)(1)} \quad a^{(l-1)(2)} \quad \cdots \quad a^{(l-1)(m)}\,]$.
3. 对最后一层的第i个神经元$a_i^{(L)}$, 其在第m个样本上的损失函数$cost(a_i^{(L)})$为 $cost(a_i^{(L)}) = \begin{cases} (1-y_i)\log(1-a_i^{(L)}) & y_i = 0 \\ y_i \log(a_i^{(L)}) & y_i = 1 \end{cases} = y_i \log(a_i^{(L)}) + (1-y_i)\log(1-a_i^{(L)})$

则$A^{(l-1)}$与$A^{(l)}$的关系为(右下角为矩阵维数, 帮助理解):

$$Z^{(l)}_{n^{(l)}\times m} = W^{(l-1)}_{n^{(l)}\times n^{(l-1)}} A^{(l-1)}_{n^{(l-1)}\times m} + \underbrace{[b^{(l)} b^{(l)} \cdots b^{(l)}]_{n^{(l)}\times m}}_{m个} \tag{1}$$

$$A^{(l)}_{n^{(l)}\times m} = g(Z^{(l)}) \tag{2}$$

其中$W^{(l-1)}$是$n^{(l)} \times n^{(l-1)}$维的系数矩阵, $W_{ij}^{(l-1)}$代表第$l-1$层的第j个神经元占第l层的第i个神经元的权重.

同时可知, 神经网络的总损失函数J为(对最后一层的每个神经元和每个样本求和):

$$J = -\frac{1}{m}\sum_{i=1}^{n^{(l)}}\sum_{j=1}^{m}(y_i \log(a_i^{(L)(j)}) + (1-y_i)\log(1-a_i^{(L)(j)})) \tag{3}$$

此外, 我们还需要一些矩阵求导的知识(非常基础):

$$\begin{aligned} \mathrm{d}f = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\partial f}{\partial x_{ij}}\mathrm{d}x_{ij} &= \mathrm{tr}(\begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{12}} & \cdots & \frac{\partial f}{\partial x_{1n}} \\ \frac{\partial f}{\partial x_{21}} & \frac{\partial f}{\partial x_{22}} & \cdots & \frac{\partial f}{\partial x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_{m1}} & \frac{\partial f}{\partial x_{m2}} & \cdots & \frac{\partial f}{\partial x_{mn}} \end{bmatrix}^T \begin{bmatrix} \mathrm{d}x_{11} & \mathrm{d}x_{12} & \cdots & \mathrm{d}x_{1n} \\ \mathrm{d}x_{21} & \mathrm{d}x_{22} & \cdots & \mathrm{d}x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{d}x_{m1} & \mathrm{d}x_{m2} & \cdots & \mathrm{d}x_{mn} \end{bmatrix}) \\ &= \mathrm{tr}((\frac{\partial f}{\partial x})^T \mathrm{d}x) \end{aligned} \tag{4}$$

当f是对自变量矩阵x(维度: $m \times n$)中逐元素的函数时, 有

$$\mathrm{d}f(x) = f'(x) \odot \mathrm{d}x = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{12}} & \cdots & \frac{\partial f}{\partial x_{1n}} \\ \frac{\partial f}{\partial x_{21}} & \frac{\partial f}{\partial x_{22}} & \cdots & \frac{\partial f}{\partial x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_{m1}} & \frac{\partial f}{\partial x_{m2}} & \cdots & \frac{\partial f}{\partial x_{mn}} \end{bmatrix} \odot \mathrm{d}x \tag{5}$$

和一些关于矩阵的迹的知识(也非常基础):

当ABC三个矩阵维度相同, 均为$m \times n$时, 有

$$tr(A^T(B \odot C)) = \mathrm{tr}((A \odot B)^T C) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}B_{ij}C_{ij} \tag{6}$$

其中$\odot$代表两个维度相同的矩阵位置相同的元素相乘得到的维度不变的新矩阵, 例如 $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \odot \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 9 \end{bmatrix}$

当AB, BA均合法时, 有

$$tr(AB) = tr(BA) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}B_{ij} \tag{7}$$

正式开始

首先明确我们的目标为: 通过矩阵相乘等形式, 直接计算$\frac{\partial J}{\partial W^{(l)}}$和$\frac{\partial J}{\partial b^{(l)}}$.

我们令J直接对$Z^{(l)}$求微分:

$$\mathrm{d}J = \mathrm{tr}((\frac{\partial f}{\partial Z^{(l)}})^T \mathrm{d}Z^{(l)}) \tag{8}$$

第一步: 假设$\frac{\partial J}{\partial Z^{(l)}}$已知，求$\frac{\partial J}{\partial W^{(l)}}$和$\frac{\partial J}{\partial b^{(l)}}$

假设$\frac{\partial J}{\partial Z^{(l)}}$已知, 让我们继续计算$\mathrm{d}Z^{(l)}$:

$$\mathrm{d}Z^{(l)} = \mathrm{d}(W^{(l-1)}A^{(l-1)} + B^{(l)}) \tag{9}$$

其中$B^{(l)} = \begin{bmatrix} b^{(l)} & b^{(l)} & \cdots & b^{(l)} \end{bmatrix}$.

从(9)出发, 我们可以分别对$W^{(l-1)}$和$B^{(l)}$求微分. 让我们先对$W^{(l-1)}$求微分:

$$\mathrm{d}Z^{(l)} = \mathrm{d}W^{(l-1)}A^{(l-1)} \tag{10}$$

代入(8):

$$\mathrm{d}J = \mathrm{tr}((\frac{\partial f}{\partial Z^{(l)}})^T \mathrm{d}W^{(l-1)}A^{(l-1)}) \overset{(7)}{=} tr(A^{(l-1)}(\frac{\partial f}{\partial Z^{(l)}})^T \mathrm{d}W^{(l-1)}) = \mathrm{tr}((\frac{\partial f}{\partial Z^{(l)}}A^{(l-1)^T})^T \mathrm{d}W^{(l-1)}) \tag{11}$$

参照(4)可知:

$$\frac{\partial J}{\partial W^{(l-1)}} = \frac{\partial f}{\partial Z^{(l)}}A^{(l-1)^T} \tag{12}$$

回到(9), 对$B^{(l)}$求微分:

$$\mathrm{d}Z^{(l)} = \mathrm{d}B^{(l)} \tag{13}$$

代入(8):

$$\mathrm{d}J = \mathrm{tr}((\frac{\partial f}{\partial Z^{(l)}})^T \mathrm{d}W^{(l)}) = \mathrm{tr}((\frac{\partial f}{\partial Z^{(l)}})^T \mathrm{d}B^{(l)}) \tag{14}$$

设$\frac{\partial f}{\partial Z^{(l)}} = U_{n^{(l)} \times m}$, 则(14)可以改写为:

$$\mathrm{d}J = \mathrm{tr}([\,u_1 \quad u_2 \quad \cdots \quad u_m\,]^T \mathrm{d}\underbrace{[\,b^{(l)} \quad b^{(l)} \quad \cdots \quad b^{(l)}\,]}_{m\text{个}}) = \mathrm{tr}(\begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_m^T \end{bmatrix} \mathrm{d}\,[\,b^{(l)} \quad b^{(l)} \quad \cdots \quad b^{(l)}\,]) \quad (15)$$

$$= \mathrm{tr}(u_1^T \mathrm{d}b^{(l)} + u_2^T \mathrm{d}b^{(l)} + \cdots + u_m^T \mathrm{d}b^{(l)}) = \mathrm{tr}((u_1^T + u_2^T + \cdots + u_m^T)\mathrm{d}b^{(l)})$$

$$= \mathrm{tr}((u_1 + u_2 + \cdots + u_m)^T \mathrm{d}b^{(l)})$$

其中$u_{(i)}$是U的第i列. 于是我们得到$\frac{\partial J}{\partial b^{(l)}}$:

$$\frac{\partial J}{\partial b^{(l)}} = u_1 + u_2 + \cdots + u_m = \begin{bmatrix} \sum_{j=0}^{m} (\frac{\partial J}{\partial Z^{(l)}})_{1j} & \sum_{j=0}^{m} (\frac{\partial J}{\partial Z^{(l)}})_{2j} & \cdots & \sum_{j=0}^{m} (\frac{\partial J}{\partial Z^{(l)}})_{n^{(l)}j} \end{bmatrix} \quad (16)$$

如此, 我们在已知$\frac{\partial J}{\partial Z^{(l)}}$的前提下, 得到了$\frac{\partial J}{\partial W^{(l-1)}}$(12)和$\frac{\partial J}{\partial b^{(l)}}$(16).

接下来我们的问题就是: 如何得到$\frac{\partial J}{\partial Z^{(l)}}$.

我们的思路是: 如果$\frac{\partial J}{\partial Z^{(l)}}$能被$\frac{\partial J}{\partial Z^{(l+1)}}$迭代求得, 那我们只需知道$\frac{\partial J}{\partial Z^{(L)}}$, 就可以反向地求得每一层的$\frac{\partial J}{\partial Z^{(l)}}$、$\frac{\partial J}{\partial W^{(l-1)}}$和$\frac{\partial J}{\partial b^{(l)}}$.

第二步: 证明$\frac{\partial J}{\partial Z^{(l)}}$能被$\frac{\partial J}{\partial Z^{(l+1)}}$迭代求得

让我们回到(9)式, 这次我们对$A^{(l-1)}$求微分, 并将(2)式代入:

$$\mathrm{d}Z^{(l)} = \mathrm{d}(W^{(l-1)}A^{(l-1)} + B^{(l)}) = W^{(l-1)}\mathrm{d}A^{(l-1)} \overset{(5)}{=} W^{(l-1)}(g'(Z^{(L-1)}) \odot \mathrm{d}Z^{(L-1)}) \quad (17)$$

将(17)代入(8):

$$\mathrm{d}J = \mathrm{tr}((\frac{\partial f}{\partial Z^{(l)}})^T W^{(l-1)}(g'(Z^{(L-1)}) \odot \mathrm{d}Z^{(L-1)})) = \mathrm{tr}(\underbrace{(W^{(l-1)^T}\frac{\partial f}{\partial Z^{(l)}})^T}_{A^T}\underbrace{(g'(Z^{(L-1)})}_{B} \odot \underbrace{\mathrm{d}Z^{(L-1)}}_{C})) \quad (18)$$

$$\overset{(6)}{=} tr\{[(W^{(l-1)^T}\frac{\partial f}{\partial Z^{(l)}}) \odot g'(Z^{(L-1)})]^T \mathrm{d}Z^{(L-1)}\}$$

对照(4)式, 我们得到了$\frac{\partial J}{\partial Z^{(l-1)}}$和$\frac{\partial J}{\partial Z^{(l)}}$之间的递推式:

$$\frac{\partial J}{\partial Z^{(l-1)}} = (W^{(l-1)^T}\frac{\partial f}{\partial Z^{(l)}}) \odot g'(Z^{(L-1)}) \quad (19)$$

其中$g'(Z^{(L-1)})$取决于激活函数的形式.

到此, 我们发现只要求得$\frac{\partial J}{\partial Z^{(L)}}$, 就可以反向求出$\frac{\partial J}{\partial Z^{(L-1)}}$, $\frac{\partial J}{\partial Z^{(L-2)}}$, ..., $\frac{\partial J}{\partial Z^{(1)}}$, 从而求出我们需要的所有$\frac{\partial J}{\partial W}$和$\frac{\partial J}{\partial b}$.

第三步: 求解$\frac{\partial J}{\partial Z^{(L)}}$

首先让我们将(3)抽象成向量的形式:

$$J = -\frac{1}{m}\sum_{i=1}^{m}[y^T \log(a^{(L)(i)}) + (e-y)^T \log(e - a^{(L)(i)})] \quad (21)$$

其中e是维度为$n^{(L)} \times 1$, 各元素都为1的矩阵.

对(21)两边求微分:

$$\begin{aligned}\mathrm{d}J &= -\frac{1}{m}\sum_{i=1}^{m}[y^T\mathrm{d}\log(a^{(L)(i)}) + (e-y)^T\mathrm{d}\log(e-a^{(L)(i)})] \\ &= -\frac{1}{m}\sum_{i=1}^{m}[y^T(v_1^{(L)(i)}\odot g'(z^{(L)(i)})\odot \mathrm{d}z^{(L)(i)}) - (e-y)^T(v_2^{(L)(i)}\odot g'(z^{(L)(i)})\odot \mathrm{d}z^{(L)(i)})]\end{aligned} \tag{22}$$

其中矩阵$v_1^{(L)(i)} = \begin{bmatrix}\frac{1}{a_1^{(L)(i)}} & \frac{1}{a_2^{(L)(i)}} & \cdots & \frac{1}{a_{n^{(L)}}^{(L)(i)}}\end{bmatrix}^T$, $v_2^{(L)(i)} = \begin{bmatrix}\frac{1}{1-a_1^{(L)(i)}} & \frac{1}{1-a_2^{(L)(i)}} & \cdots & \frac{1}{1-a_{n^{(L)}}^{(L)(i)}}\end{bmatrix}$.

接下来我们以激活函数为$sigmoid$函数为例继续计算. 此时有:

$$g'(x) = (\frac{1}{1+e^{-x}})' = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}}(1-\frac{1}{1+e^{-x}}) = g(x)(1-g(x)) \tag{23}$$

将(23)代入(22)可知:

$$\mathrm{d}J = -\frac{1}{m}\sum_{i=1}^{m}[y^T(e-a^{(L)(i)})\odot \mathrm{d}z^{(L)(i)} - (e-y)^T a^{(L)(i)}\odot \mathrm{d}z^{(L)(i)}] \tag{24}$$

考虑到y, $a^{(L)(i)}$与$\mathrm{d}z^{(L)(i)}$维度相同, 应用(6):

$$\begin{aligned}\mathrm{d}J &= -\frac{1}{m}\sum_{i=1}^{m}[y\odot(e-a^{(L)(i)}) - (e-y)\odot a^{(L)(i)}]^T\mathrm{d}z^{(L)(i)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}(y\odot e - e\odot a^{(L)(i)})^T\mathrm{d}z^{(L)(i)} = -\frac{1}{m}\sum_{i=1}^{m}(y-a^{(L)(i)})^T\mathrm{d}z^{(L)(i)} \\ &= -\frac{1}{m}\mathrm{tr}(\begin{bmatrix}y-a^{(L)(1)} & y-a^{(L)(2)} & \cdots & y-a^{(L)(m)}\end{bmatrix}^T\begin{bmatrix}\mathrm{d}z^{(L)(1)} & \mathrm{d}z^{(L)(2)} & \cdots & \mathrm{d}z^{(L)(m)}\end{bmatrix}) \\ &= -\frac{1}{m}\mathrm{tr}(\begin{bmatrix}y-a^{(L)(1)} & y-a^{(L)(2)} & \cdots & y-a^{(L)(m)}\end{bmatrix}^T\mathrm{d}Z^{(L)})\end{aligned} \tag{25}$$

对照(4)可知:

$$\frac{\partial J}{\partial Z^{(L)}} = -\frac{1}{m}\begin{bmatrix}y-a^{(L)(1)} & y-a^{(L)(2)} & \cdots & y-a^{(L)(m)}\end{bmatrix} \tag{26}$$

结论

综合(12), (16), (26)可知, 只要知道$\frac{\partial J}{\partial Z^{(l)}}$, 就可以求出$\frac{\partial J}{\partial W^{(l-1)}}$, $\frac{\partial J}{\partial b^{(l)}}$, 以及$\frac{\partial J}{\partial Z^{(l-1)}}$:

$$\frac{\partial J}{\partial Z^{(l)}} \Rightarrow \begin{cases}\frac{\partial J}{\partial W^{(l-1)}} = \frac{\partial J}{\partial Z^{(l)}}{A^{(l-1)}}^T \\ \frac{\partial J}{\partial b^{(l)}} = \begin{bmatrix}\sum_{j=0}^{m}(\frac{\partial J}{\partial Z^{(l)}})_{1j} & \sum_{j=0}^{m}(\frac{\partial J}{\partial Z^{(l)}})_{2j} & \cdots & \sum_{j=0}^{m}(\frac{\partial J}{\partial Z^{(l)}})_{n^{(l)}j}\end{bmatrix} \\ \frac{\partial J}{\partial Z^{(l-1)}} = ({W^{(l-1)}}^T\frac{\partial J}{\partial Z^{(l)}})\odot g'(A^{(l-1)})\end{cases} \tag{27}$$

当激活函数取$sigmoid$函数时

$$\frac{\partial J}{\partial Z^{(L)}} = -\frac{1}{m}\begin{bmatrix}y-a^{(L)(1)} & y-a^{(L)(2)} & \cdots & y-a^{(L)(m)}\end{bmatrix} \tag{28}$$

另: 在python中对$\frac{\partial J}{\partial b^{(l)}}$和$\frac{\partial J}{\partial Z^{(L)}}$有更为简化的写法(下文代码中分别用db和dZ代替):

```
db = np.sum(dZ, axis=1, keepdims=True)
dZ = -1.0 / m * (y - A) ## 利用numpy的广播
```

完结, 撒花~