矩阵2范数的循环计算转换为矩阵计算

一个矩阵化计算技巧

龙鸿峰

2022 年 4 月 23 日

1 问题建模

1.1 问题简介

这个文档主要是介绍一种将矩阵2范数的计算从直接使用循环，转换为使用矩阵进行计算的方法。在使用核技巧计算核函数的时候，很容易碰到如下需要计算的等式：(高斯核函数)

$$K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{1}$$

其中，x_i, x_j分别表示2个特征向量，并且如果训练集中有N个样本，即$x_i, i = 1, 2, \ldots, N$。则，x_i, x_j均需要取N次，且核函数矩阵尺寸为$N \times N$。

传统的核函数计算方式是直接循环遍历矩阵的每一个元素，逐个计算，matlab的代码如下：(x_i, x_j均为列向量)

```
# 获得样本个数
xs_length=length(x_train);
# 使用循环计算核函数中每一个元素for
A=zeros(xs_length,xs_length);
for i=1:xs_length
    for j=1:xs_length
    # 高斯核函数
```

```
    A(i,j)=exp(-g*(x_train(:,i)-x_train(:,j))^2);
    end
end
```

这种做法就是一个一个的计算核函数的各个元素，当核函数比较大的时候效率就比较低，并且不方便并行计算和利用矩阵库。所以需要对其进行一些优化。

1.2 等价矩阵变换

考虑到想将上述的计算转换为矩阵操作(同样是高斯核函数)，即：使用矩阵运算来代替for循环，以此来提高计算效能并且简化代码。具体的变换结果如下：(x_i, x_j均为列向量)

```
A = transpose(x_train) * x_train;
# 下面一句等价于 M = repmat(diag(D),[1,N])
M = diag(diag(A))*ones(xs_length,xs_length);
A = M + transpose(M)-2*A;
A = exp(-g*A);
```

其中，代码部分的transpose()函数可以替换为转置符号，效果相同。推荐将transpose()函数改为转置符号使用。具体的操作给出，后面给出这样操作在数学上的等价性，并且给与证明

2 数学证明

2.1 数学建模

为了便于后续的证明，这里给出对问题的基本定义和一些相关的数学符号说明：假设一个数据集$\{x_i, y_i\}_{i=1}^{N}$，并且x_i均为列向量，所以训练用的数据可以构成如下的矩阵$x_{train} = [x_1, x_2, \ldots, x_k]$。

下面将给出具体的数学推导，先从(1)开始，令

$$\begin{aligned}&g = \tfrac{1}{2\sigma^2}\\&G(x_i, x_j) = \|x_i - x_j\|^2\end{aligned} \tag{2}$$

所以，方程1可以简化为：

$$K(x_i, x_j) = exp(-g \times G(x_i, x_j)) \tag{3}$$

接下来就是研究(2)中的函数$G(x_i, x_j)$如何计算了，为了直观展示，我们将核函数矩阵 K 的完整形式写成如下的形式：

$$\begin{aligned}G &= \begin{bmatrix} G(x_1, x_1) & G(x_1, x_2) & \cdots & G(x_1, x_N) \\ G(x_2, x_1) & G(x_2, x_2) & \cdots & G(x_2, x_N) \\ \vdots & \vdots & & \vdots \\ G(x_N, x_1) & G(x_N, x_2) & \cdots & G(x_N, x_N) \end{bmatrix}\\ K &= exp(-g \times G)\end{aligned} \tag{4}$$

至此，证明前的准备就完成了，可以开始操作了。

2.2 证明过程

首先我们可以将(2)中的$G(x_i, y_i)$，按照完全平方公式进行展开，得到如下的公式：

$$\begin{aligned} G(x_i, x_j) &= (x_i - x_j)^T (x_i - x_j) \\ &= x_i^T x_i - x_i^T x_j - x_j^T x_i + x_j^T x_j \end{aligned} \tag{5}$$

将(5)的结果带入到(4)中，可以发现：

$$G(x_i, x_j) = \begin{cases} 0, & \text{if } i = j \\ x_i^T x_i - x_i^T x_j - x_j^T x_i + x_j^T x_j, & \text{if } i \neq j \end{cases} \tag{6}$$

然后我们来研究一下这个矩阵$A = x_{train}^T x_{train}$，其展开式如下：

$$A = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_N \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_N \\ \vdots & \vdots & & \vdots \\ x_N^T x_1 & x_N^T x_2 & \cdots & x_N^T x_N \end{bmatrix} \tag{7}$$

可以很明显看出，矩阵$x_{train}^T x_{train}$中的元素很多都是(6)中需要的，所以关键在于如何取出这些元素，再辅助上简单运算即可实现矩阵G的计算。

首先取出矩阵A中的对角元素Q，这个操作的对应命令就是：Q = diag(diag(A))，即:

$$Q = \begin{bmatrix} x_1^T x_1 & & & \\ & x_2^T x_2 & & \\ & & \ddots & \\ & & & x_N^T x_N \end{bmatrix} \tag{8}$$

接着让Q乘上一个全1矩阵，得到矩阵M，具体为：

$$M = Q \times \mathbf{1} = \begin{bmatrix} x_1^T x_1 & x_1^T x_1 & x_1^T x_1 & x_1^T x_1 \\ x_2^T x_2 & x_2^T x_2 & x_2^T x_2 & x_2^T x_2 \\ \vdots & & & \vdots \\ x_N^T x_N & x_N^T x_N & x_N^T x_N & x_N^T x_N \end{bmatrix} \tag{9}$$

其中,**1**表示一个全1矩阵，且尺寸为$N \times N$。所以，接下来按照$G = M + M^T - 2 * A$进行计算，就会得到矩阵G。为了便于撰写，我们抽出G(1,2)进行展示

$$\begin{aligned} G(1,2) &= M(1,2) + M^T(1,2) - 2 * A(1,2) \\ &= x_1^T x_1 + x_2^T x_2 - 2 * x_1^T x_2 \end{aligned} \tag{10}$$

而如果将特征向量x_i, x_j展开，就可以发现：

$$x_i^T x_j = x_j^T x_i \tag{11}$$

因为均为2个特征向量x_i, x_j点乘。因此(10)最终可以化为：

$$\begin{aligned} G(1,2) &= x_1^T x_1 + x_2^T x_2 - x_1^T x_2 - x_2^T x_1 \\ G &= M + M^T - 2 * A \end{aligned} \tag{12}$$

至此，证明完毕，可以发现这个过程就是将原来需要循环计算的矩阵，转化为了矩阵进行计算。