正则化项 L1 和 L2 的直观理解

1. 正则化（Regularization）

机器学习中几乎都可以看到损失函数后面会添加一个额外项, 常用的额外项一般有两种，英文称作 ℓ1-norm 和 ℓ2-norm，中文称作 L1 正则化和 L2 正则化，或者 L1 范数和 L2 范数。

L1 正则化和 L2 正则化可以看作是损失函数的惩罚项。所谓『惩罚』是指对损失函数中的某些参数做一些限制。对于线性回归模型，使用 L1 正则化的模型叫做 Lasso 回归，使用 L2 正则化的模型叫做 Ridge 回归（岭回归）。下图是 Python 中 Lasso 回归的损失函数，式中加号后面一项即为 L1 正则化项。

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$$

下图是 Python 中 Ridge 回归的损失函数, 式中加号后面一项即为 L2 正则化项。

$$\min_{w} ||Xw - y||_2{}^2 + \alpha ||w||_2{}^2$$

一般回归分析中回归 w 表示特征的系数，从上式可以看到正则化项是对系数做了处理（限制）。L1 正则化和 L2 正则化的说明如下:

- L1 正则化是指权值向量 w 中各个元素的绝对值之和，通常表示为 $||w||_1$
- L2 正则化是指权值向量 w 中各个元素的平方和然后再求平方根(可以看到 Ridge 回归的 L2 正则化项有平方符号)，通常表示为$||w||_2$

那添加 L1 和 L2 正则化有什么用？下面是 L1 正则化和 L2 正则化的作用，这些表述可以在很多文章中找到。

- L1 正则化可以产生稀疏权值矩阵，即产生一个稀疏模型，可以用于特征选择
- L2 正则化可以防止模型过拟合；一定程度上，L1 也可以防止过拟合

2. 稀疏模型与特征选择

上面提到 L1 正则化有助于生成一个稀疏权值矩阵，进而可以用于特征选择。为什么要生成一个稀疏矩阵？

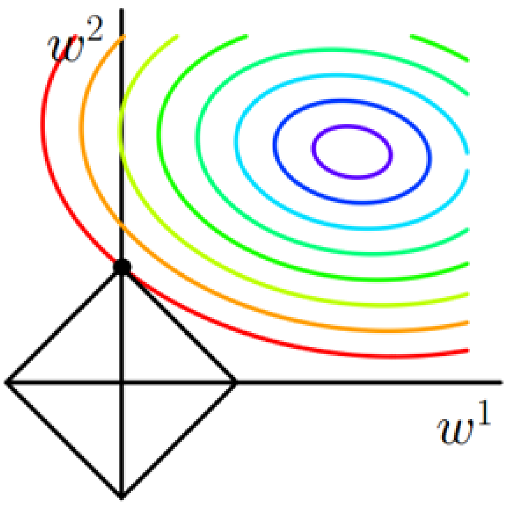
稀疏矩阵指的是很多元素为 0，只有少数元素是非零值的矩阵，即得到的线性回归模型的大部分系数都是 0。通常机器学习中特征数量很多，例如文本处理时，如果将一个词组作为一个特征，那么特征数量会达到上万个。在预测或分类时，那么多特征显然难以选择，但是如果代入这些特征得到的模型是一个稀疏模型，表示只有少数特征对这个模型有贡献，绝大部分

特征是没有贡献的，或者贡献微小（因为它们前面的系数是 0 或者是很小的值，即使去掉对模型也没有什么影响），此时我们就可以只关注系数是非零值的特征。这就是稀疏模型与特征选择的关系。

3. L1 正则化和特征选择

假设有如下带 L1 正则化的损失函数：$J=J_0+\alpha\sum|w|$

其中 J_0 是原始的损失函数，加号后面的一项是 L1 正则化项，α 是正则化系数。注意到 L1 正则化是权值的绝对值之和，J 是带有绝对值符号的函数，因此 J 是不完全可微的。机器学习的任务就是要通过一些方法（比如梯度下降）求出损失函数的最小值。当我们在原始损失函数 J_0 后添加 L1 正则化项时，相当于对 J_0 做了一个约束。令 $L=\alpha\sum|w|$，则 $J=J_0+L$，此时我们的任务变成在 L 约束下求出 J_0 取最小值的解。考虑二维的情况，即只有两个权值 w1 和 w2，此时 L=|w1|+|w2|，对于梯度下降法，求解 J_0 的过程可以画出等值线，同时 L1 正则化的函数 L 也可以在 w1w2 的二维平面上画出来。如下图：

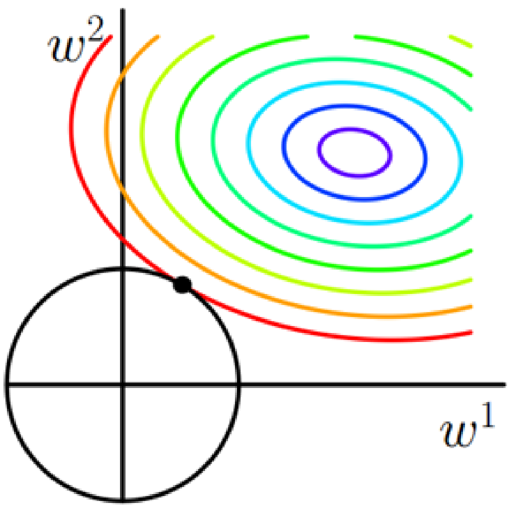


图中等值线是 J_0 的等值线，黑色方形是 L 函数的图形。在图中，当 J_0 等值线与 L 图形首次相交的地方就是最优解。上图中 J_0 与 L 在 L 的一个顶点处相交,这个顶点就是最优解。注意到这个顶点的值是(w1,w2)=(0,w)。可以直观想象，因为 L 函数有很多『突出的角』(二维情况下四个，多维情况下更多)，J_0 与这些角接触的机率会远大于与 L 其它部位接触的机率，而在这些角上，会有很多权值等于 0，这就是为什么 L1 正则化可以产生稀疏模型，进而可以用于特征选择。

而正则化前面的系数α可以控制 L 图形的大小。α 越小，L 的图形越大（上图中的黑色方框）；α 越大，L 的图形越小，可以小到黑色方框只超出原点范围一点点，这是最优点的值(w1,w2)=(0,w)中的 w 可以取到很小的值。

4. L2 正则化

假设有如下带 L2 正则化的损失函数：$J=J_0+\alpha\sum w^2$，同样可以画出他们在二维平面上的图形，如下：



二维平面下 L2 正则化的函数图形是个圆，与方形相比，被磨去了棱角。因此 J_0 与 L 相交时使得 w1 或 w2 等于零的机率小了许多，这就是为什么 L2 正则化不具有稀疏性的原因。

5. L2 正则化和过拟合

拟合过程中通常都倾向于让权值尽可能小，最后构造一个所有参数都比较小的模型。因为一般认为参数值小的模型比较简单，能适应不同的数据集，也在一定程度上避免了过拟合现象。可以设想一下对于一个线性回归方程，若参数很大，那么只要数据偏移一点点，就会对结果造成很大的影响；但如果参数足够小，数据偏移得多一点也不会对结果造成什么影响，专业一点的说法是『抗扰动能力强』。

那为什么 L2 正则化可以获得值很小的参数?

以线性回归中的梯度下降法为例。假设要求的参数为θ，hθ(x)是我们的假设函数，那么线性回归的代价函数如下：

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

那么在梯度下降法中，最终用于迭代计算参数θ的迭代式为：

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

其中α是 learning rate. 上式是没有添加 L2 正则化项的迭代公式，如果在原始代价函数之后添加 L2 正则化，则迭代公式会变成下面的样子：

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

其中λ就是正则化参数。从上式可以看到，与未添加 L2 正则化的迭代公式相比，每一次迭代，θ_j 都要先乘以一个小于 1 的因子，从而使得 θ_j 不断减小，因此总得来看，θ 是不断减小的。

最开始也提到 L1 正则化一定程度上也可以防止过拟合。之前做了解释，当 L1 的正则化系数很小时，得到的最优解会很小，可以达到和 L2 正则化类似的效果。

6. **正则化参数的选择**

- **L1 正则化参数**

  通常越大的λ可以让代价函数在参数为 0 时取到最小值。

- **L2 正则化参数**

  参考图 2，λ 越大，L2 圆的半径越小，最后求得代价函数最值时各参数也会变得很小。