神经网络中，同样存在欠拟合和过拟合，理想状态是选择合适的模型，避开欠拟合和过拟合，但如果没有合适的模型，一般选择过拟合模型，然后通过某种方法来防止过拟合。

<https://blog.csdn.net/qq_37277944/article/details/82746635>

batch size：批量大小，每次训练从训练集中选取的样本数目

epoch：所有训练集的一次前向反馈和反向传播

iteration：使用batch size个样本进行一次前向反馈和反向传播。

optimizer：优化器，主要是优化寻找误差最小值的方法。在误差最小的过程中，一般的梯度下降有可能会碰见迭代步长太小导致训练时间长、找到了局部最优解而不是全局的最优解等问题，使用optimizer可以优化这一过程，常用的优化器有：

1. SGD，随机梯度下降
2. Adam，使用复杂的指数衰减
3. RMSProp，均方根误差衰减

训练集有1000个样本，batchsize=10，那么：

样本集1次epoch需要100次iteraton.

过拟合的处理：

对于一个模型，权重系数为：



误差函数为：



发生过拟合时，模型的训练误差一般比较小，可以通过正则化的操作，来惩罚高权重系数，来防止过拟合。

L1正则化：参考L1范数



L2正则化：参考L2范数



上述两个误差函数，如果权重系数大，则误差就越大，模式就会继续寻找最优解，从而可以惩罚高权重系数，进而来防止过拟合。

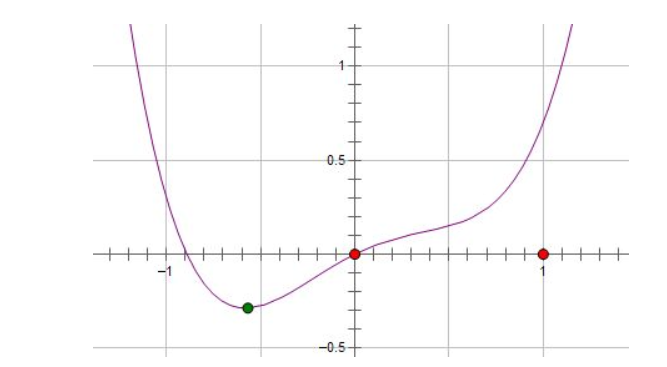
L1正则化的作用：选择稀疏模型，减少特征量，从而防止过拟合。

L2正则化的作用：会得到较小齐权（权重系数方差较小）的权重系数，从而防止过拟合。

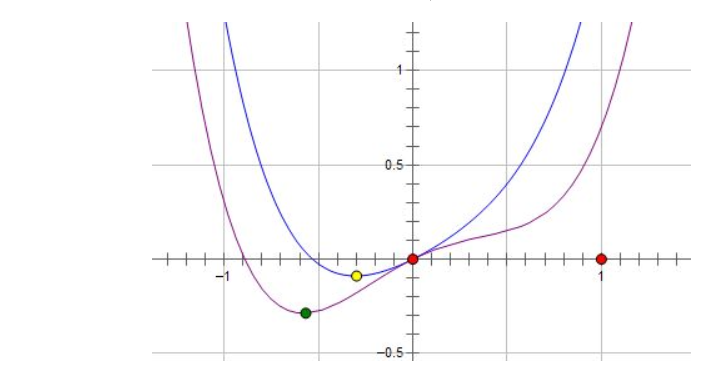
L1相比L2更容易得到稀疏模型的两种解释：

1. 对比两个向量(1, 0)和(0.5, 0.5)，L2更可能选择(0.5, 0.5)因为(1, 0)元素的平方和为1，而(0.5, 0.5)的平方和为0.5，所以L2会得到较小齐全的权重系数矩阵。
2. 参考：<https://www.zhihu.com/question/37096933/answer/70507353>

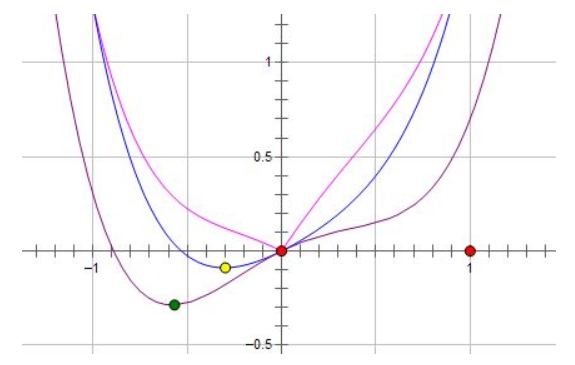
假设误差函数与某个系数x的关系如下图所示：，最优解在绿点处，x不为0



施加L2正则化：，蓝色曲线更接近二次曲线，最优解在黄点处，x仍然不为0.



施加L1正则化：，最优解在红点处，x=0，这样如果误差最小时，对应x的特征就被抛弃掉了，有用的特征集就会稀疏。



分析：

两种 正则化 能不能把最优的 x 变成 0，取决于原先的误差函数在 0 点处的导数。如果本来导数不为 0，那么施加 L2 正则化后导数依然不为 0，最优的 x 也不会变成 0。而施加 L1 正则化 时，只要 正则化项的系数 C 大于原先误差函数在 0 点处的导数的绝对值，x = 0 就会变成一个极小值点。上面只分析了一个参数 x。事实上 L1 正则化会使得许多参数的最优值变成 0，这样模型就稀疏了。

绝对值函数的导数为sign(x)函数

