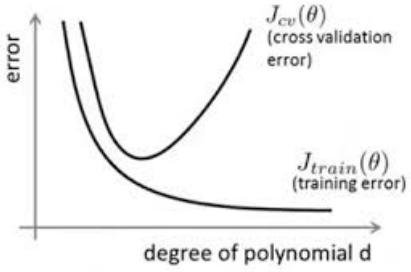
1. 什么是过拟合



随着训练过程的进行，模型复杂度增加，在训练集上的error逐渐减小，在验证集上的error逐渐增大——因为训练出来的网络过拟合了训练集，导致泛化性能差。

传统的函数拟合问题，一般是通过物理数学等推导出的一个含参数的模型（数学建模），模型复杂度是确定的，没有多余的能力拟合噪声。而机器学习算法的复杂度更高，一般都远高于具体问题的复杂度，数据量不足以支撑庞大的模型/参数。

> 有一个概念需要先说明，在机器学习算法中，我们常常将原始数据集分为三部分：training data、validation data，testing data。这个validation data是什么？它其实就是用来避免过拟合的，在训练过程中，我们通常用它来确定一些超参数（比如根据validation data上的accuracy来确定early stopping的epoch大小、根据validation data确定learning rate等等）。那为啥不直接在testing data上做这些呢？因为如果在testing data做这些，那么随着训练的进行，我们的网络实际上就是在一点一点地overfitting我们的testing data，导致最后得到的testing accuracy没有任何参考意义。因此，training data的作用是计算梯度更新权重，validation data如上所述，testing data则给出一个accuracy以判断网络的好坏。

2.降低过拟合的办法

2.1 正则化

L2正则化：目标函数中增加所有权重w参数的平方之和，逼迫所有w尽可能趋向零但不为零。因为过拟合的时候，拟合函数需要顾忌每一个点，最终形成的拟合函数波动很大，在某些很小的区间里，函数值的变化很剧烈，也就是某些w非常大。为此，L2正则化的加入就惩罚了权重变大的趋势。

L1正则化：目标函数中增加所有权重w参数的绝对值之和，逼迫更多w为零（变稀疏，L2因为其导数也趋0，奔向零的速度不如L1快）。稀疏正则化的一个优点是能实现特征的自动选择。一般来说，xi的大部分元素（也就是特征）都是和最终的输出yi没有关系或者不提供任何信息的，在最小化目标函数的时候考虑xi这些额外的特征，虽然可以获得更小的训练误差，但是在预测新的样本时，这些没用的特征权重反而会被考虑，从而干扰了对正确yi的预测。稀疏规则化算子的引入就是为了完成特征自动选择，它会学习地去掉这些无用特征，也就是把这些特征对应的权重置为0.

2.2 随机失活（Dropout)

L1、L2正则化是通过修改代价函数来实现的，而Dropout则是通过修改神经网络本身来实现的，它是在训练网络时用的一种技巧（trike）。

在训练过程中，让神经元以超参数p的概率被激活（也就是1-p的概率被置0），每个w因此随机参与，使得任意w都不是不可或缺的，效果类似数量巨大的模型集成。

运用了dropout的训练过程，相当于训练了很多个只有半数隐层单元的神经网络（后面简称为“半数网络”），每一个这样的半数网络，都可以给出一个分类结果，这些结果有的是正确的，有的是错误的。随着训练的进行，大部分半数网络都可以给出正确的分类结果，那么少数的错误分类结果就不会对最终结果造成大的影响。

2.3 逐层/批归一化（batch normalization)

给每层的输出都做一次归一化（网络上相当于加了一个线性变换层），使得下一层的输入接近高斯分布，避免了在学习过程中训练数据的分布各不相同，也不同于测试数据的分布，因此提高了泛化能力。

（BN非常重要，应该会单独讲，如果没有，记得去补充BN!!）

2.4 提前终止（early stopping)

理论上可能的局部极小值数量随参数的数量呈指数增长，到达某个精确的最小值是不良泛化的一个来源。典型的方法是根据交叉验证提前终止：若每次训练前，将训练数据划分为若干份，取一份为测试集，其他为训练集，每次训练完立即拿此次选中的测试集自测。因为每份都有一次机会当测试集，所以此方法称为交叉验证。交叉验证的错误率最小时可以认为泛化性能最好，这时候训练错误率虽然还在继续下降，但也得终止训练了。

2.5 数据集扩增（data augmentation)

更多的训练数据，意味着可以用更深的网络，训练出更好的模型。

如果不能获取更多数据，可以在原始数据上做改动，以图片数据集为例，可以做各种变换：

- 将原始图片旋转一个小角度
- 添加随机噪声
- 一些有弹性的畸变（elastic distortions），论文《Best practices for convolutional neural networks applied to visual document analysis》对MNIST做了各种变种扩增。
- 截取（crop）原始图片的一部分。比如DeepID中，从一副人脸图中，截取出了100个小patch作为训练数据，极大地增加了数据集。感兴趣的可以看《Deep learning face representation from predicting 10,000 classes》.