应用机器学习的建议

> 当我们运用训练好了的模型来预测未知数据的时候发现有较大的误差，我们下一步可以做什么？

1. 获得更多的训练样本——通常是有效的，但代价较大，下面的方法也可能有效，可考虑先采用下面的几种方法。
2. 尝试减少特征的数量
3. 尝试获得更多的特征
4. 尝试增加多项式特征
5. 尝试减少正则化程度λ
6. 尝试增加正则化程度λ

一、评估机器学习算法的性能

1.1 定义问题

> 假设需要在10个不同次数的二项式模型之间进行选择：

1. $h_\theta(x) = \theta_0 + \theta_1 x$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$

10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

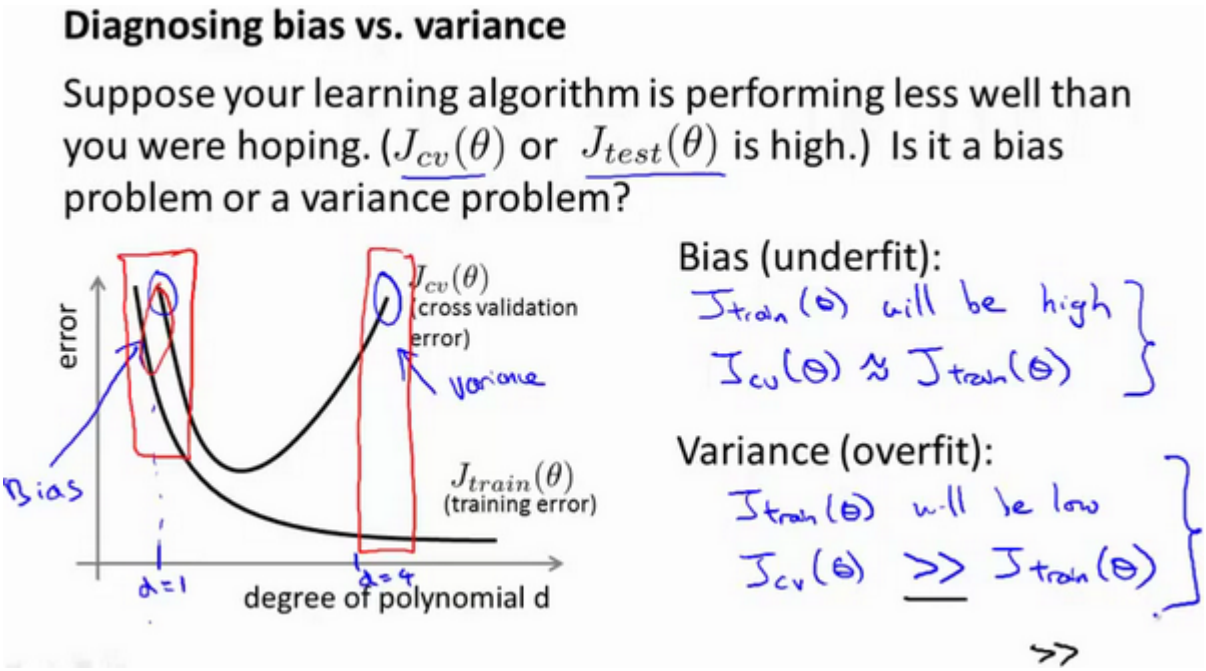
- 分割数据集
  - 60% 训练集
  - 20% 验证集
  - 20% 测试集
- 模型选择的方法为：

1. 使用训练集训练出10个模型
2. 用10个模型分别对交叉验证集计算得出交叉验证误差（代价函数的值）
3. 选取代价函数值最小的模型
4. 用步骤3中选出的模型对测试集计算得出推广误差（代价函数的值）

- 就是说
  - 训练集来训练模型
  - 验证集来筛选模型
  - 测试集得出误差/损失值

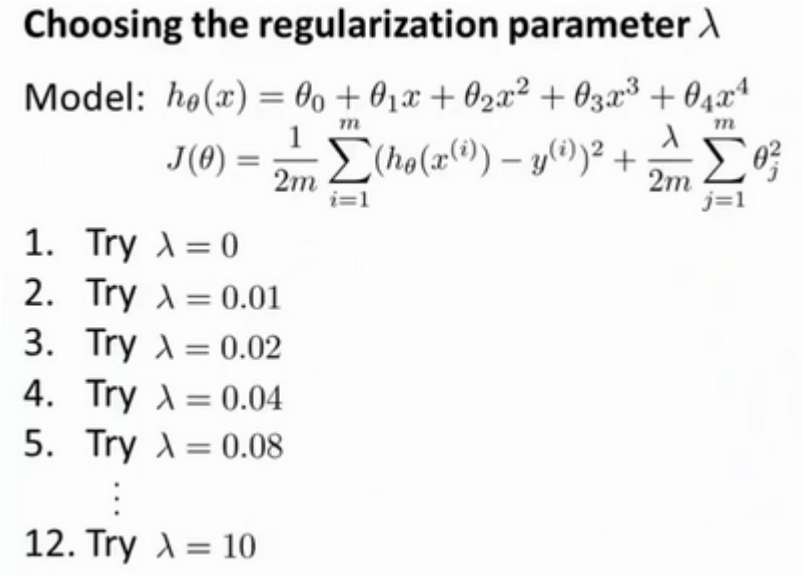
1.2 偏差和方差

> 高偏差/高方差 ——》 欠拟合或者过拟合



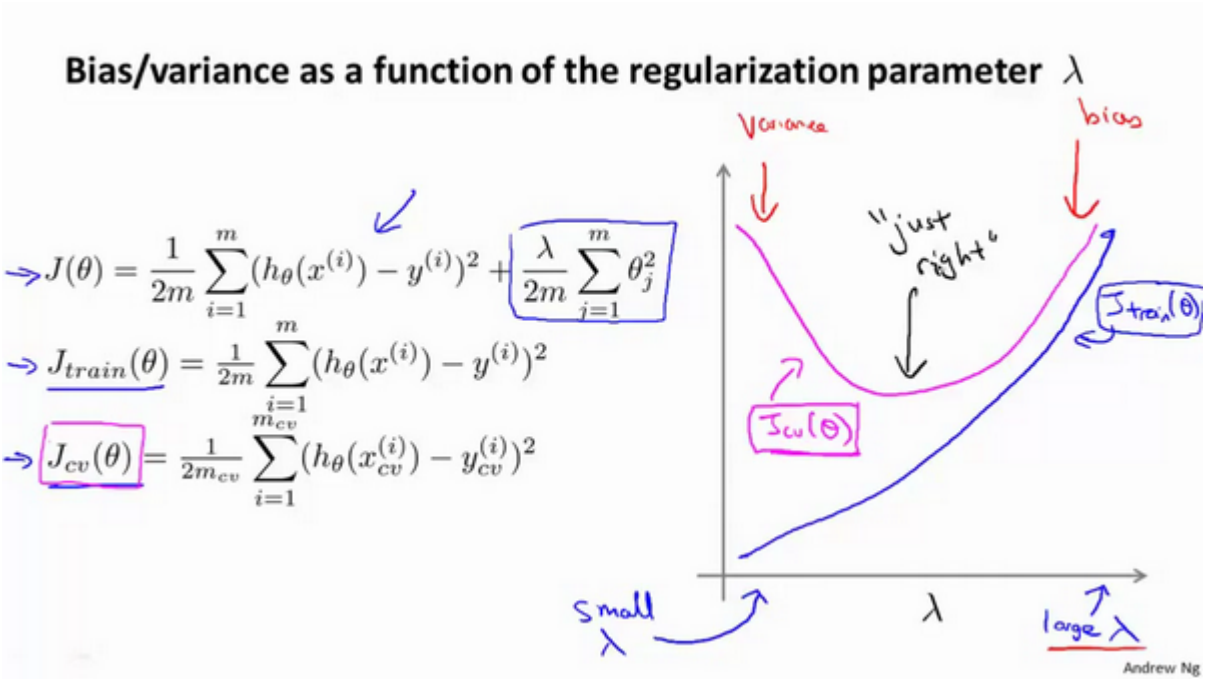
- 不同degree/多项式次数 的实验结果显示 随着算法拟合能力的提高，bias 和 variance 会变化，结果由欠拟合到过拟合

> 正则化



选择的方法为：

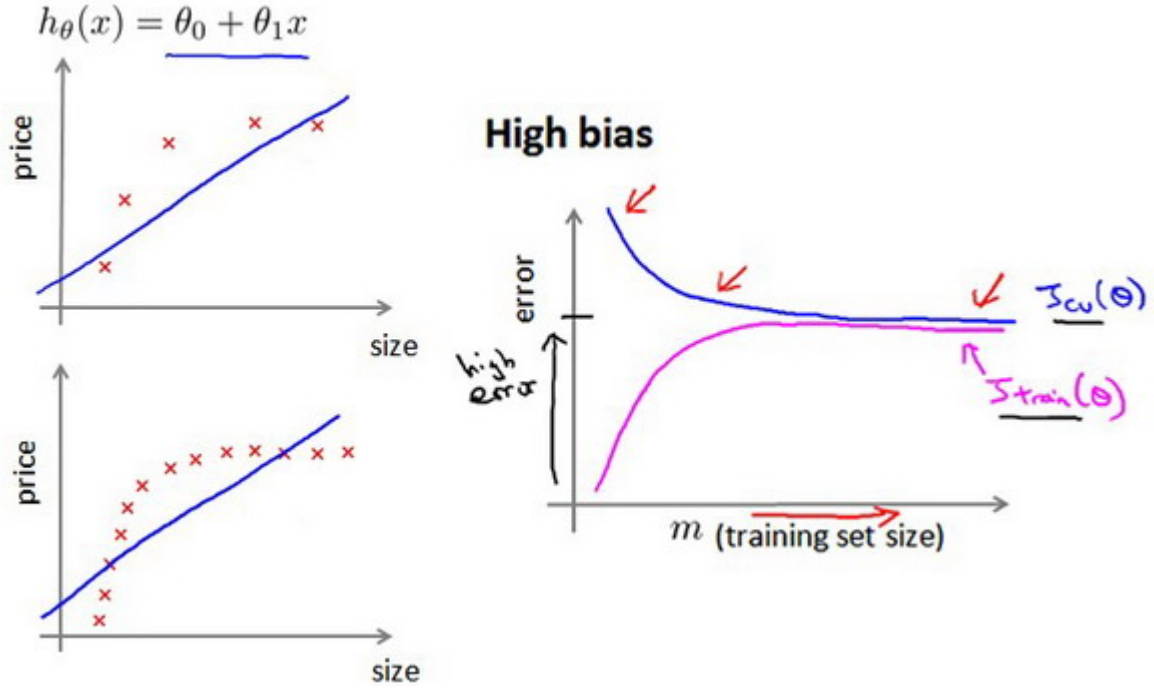
1. 使用训练集训练出12个不同程度正则化的模型
2. 用12个模型分别对交叉验证集计算的出交叉验证误差
3. 选择得出交叉验证误差**最小**的模型
4. 运用步骤3中选出模型对测试集计算得出推广误差，我们也可以同时将训练集和交叉验证集模型的代价函数误差与λ的值绘制在一张图表上：



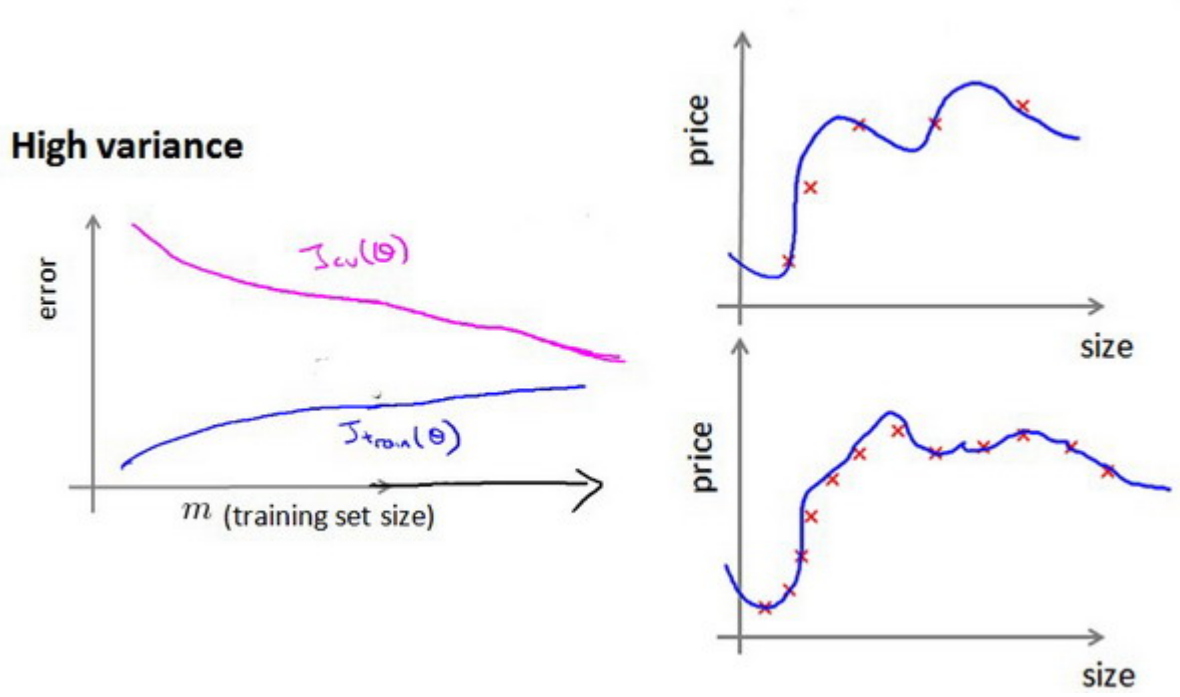
当λ较小时，训练集误差较小（过拟合）而交叉验证集误差较大，随着λ的增加，训练集误差不断增加（欠拟合），而交叉验证集误差则是先减小后增加

1.3 学习曲线

如何利用学习曲线识别高偏差/欠拟合：作为例子，我们尝试用一条直线来适应下面的数据，可以看出，无论训练集有多么大误差都不会有太大改观：



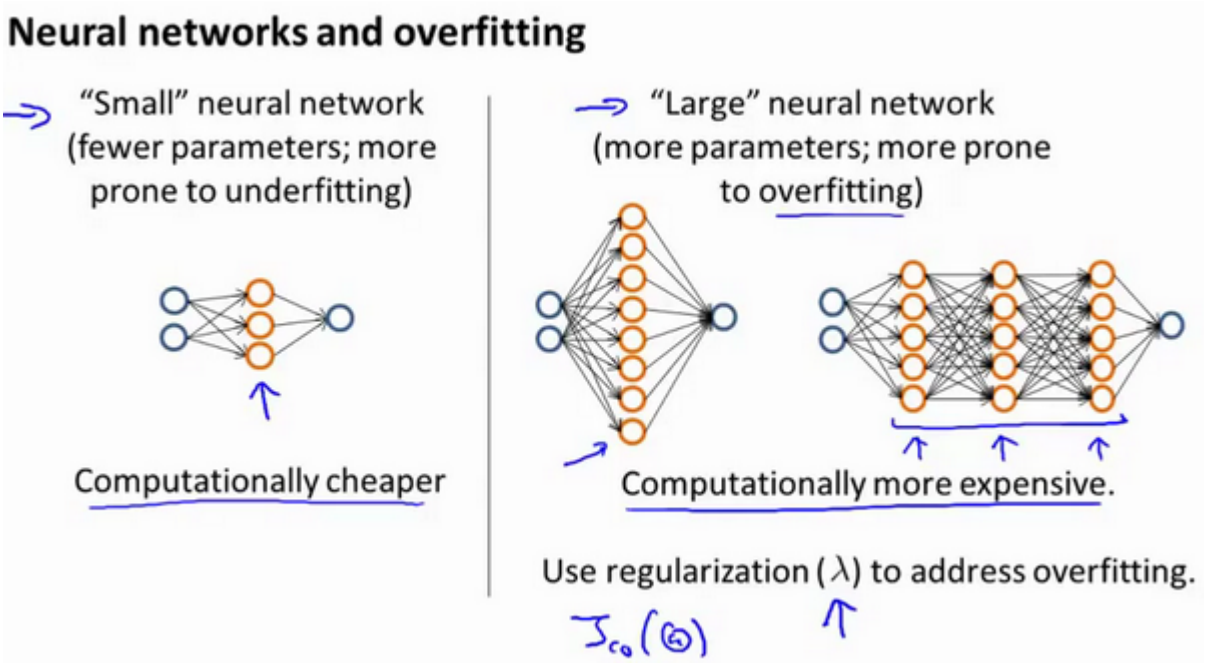
也就是说在高偏差/欠拟合的情况下，增加数据到训练集不一定能有帮助。 如何利用学习曲线识别高方差/过拟合：假设我们使用一个非常高次的多项式模型，并且正则化非常小，可以看出，当交叉验证集误差远大于训练集误差时，往训练集增加更多数据可以提高模型的效果。



1.4 如何解决

1. 获得更多的训练样本——解决高方差

2. 尝试减少特征的数量——解决高方差
3. 尝试获得更多的特征——解决高偏差
4. 尝试增加多项式特征——解决高偏差
5. 尝试减少正则化程度λ——解决高偏差
6. 尝试增加正则化程度λ——解决高方差



使用较小的神经网络，类似于参数较少的情况，容易导致高偏差和欠拟合，但计算代价较小
使用较大的神经网络，类似于参数较多的情况，容易导致高方差和过拟合，虽然计算代价比较大，但是可以通过正则化手段来调整而更加适应数据。 通常选择较大的神经网络并采用正则化处理会比采用较小的神经网络效果要好。 对于神经网络中的隐藏层的层数的选择，通常从一层开始逐渐增加层数，为了更好地作选择，可以把数据分为训练集、交叉验证集和测试集，针对不同隐藏层层数的神经网络训练神经网络， 然后选择交叉验证集代价最小的神经网络。

二、选择合适的模型

> 垃圾邮件分类问题

1. 收集更多的数据，让我们有更多的垃圾邮件和非垃圾邮件的样本
2. 基于邮件的路由信息开发一系列复杂的特征
3. 基于邮件的正文信息开发一系列复杂的特征，包括考虑截词的处理
4. 为探测刻意的拼写错误（把**watch** 写成**w4tch**）开发复杂的算法

> 推荐构建一个学习算法的方法：

1. 简单的能快速实现的算法作为开始，实现并使用交叉验证集测试这个算法
2. 绘制学习曲线，确定下一步是增加数据还是添加更多的特征，或其他选择
3. 人工检查交叉验证集中出错的样本，这些样本是否有某种系统化的趋势

例：

1. 误差分析要做的既是检验交叉验证集中我们的算法产生错误预测的所有邮件，看：是否能将这些邮件按照类分组。例如医药品垃圾邮件，仿冒品垃圾邮件或者密码窃取邮件等。然后看分类器对哪一组邮件的预测误差最大，并着手优化。
2. 思考怎样能改进分类器。例如，发现是否缺少某些特征，记下这些特征出现的次数。例如记录下错误拼写出现了多少次，异常的邮件路由情况出现了多少次等等，然后从出现次数最多的情况开始着手优化。
3. 误差分析并不总能帮助我们判断应该采取怎样的行动。有时我们需要尝试不同的模型，然后进行比较，在模型比较时，用数值来判断哪一个模型更好更有效，通常我们是看交叉验证集的误差。

2.2 类偏斜的误差度量

2.2.1 偏斜类（skewed classes）的问题

> 为我们的训练集中有非常多的同一种类的样本，只有很少或没有其他类的样本。

例如我们希望用算法来预测癌症是否是恶性的，在我们的训练集中，只有0.5%的实例是恶性肿瘤。假设我们编写一个非学习而来的算法，在所有情况下都预测肿瘤是良性的，那么误差只有0.5%。然而我们通过训练而得到的神经网络算法却有1%的误差。这时，误差的大小是不能视为评判算法效果的依据的。

2.2.2 查准率和查全率

		预测值	
		Positive	Negtive
实际值	Positive	TP	FN 查全率：TP/(TP+FN)
	Negtive	FP	TN
		查准率：TP/(TP+FP)	

在所有实际上有恶性肿瘤的病人中，成功预测有恶性肿瘤的病人的百分比，越高越好。 如果我们希望只在非常确信的情况下预测为真（肿瘤为恶性），即我们希望更高的查准率，我

们可以使用比0.5更大的阀值，如0.7，0.9。这样做我们会减少错误预测病人为恶性肿瘤的情况，同时却会增加未能成功预测肿瘤为恶性的情况。 如果我们希望提高查全率，尽可能地让所有有可能是恶性肿瘤的病人都得到进一步地检查、诊断，我们可以使用比0.5更小的阀值，如0.3。

> F1值 = 2 PR / (P + R)

4.2.1 分类模型的评价

(3) F-score

➢ **F-score是准确率与召回率的一个调和指标[0, 1]。**

$$F-Score=(1+\beta^2)*\frac{Precision*Recall}{\beta^2*Precision+Recall}$$

—— β是用来平衡Precision，Recall在F-score计算中的权重

- 如果取1,表示Precision与Recall一样重要
- 如果取小于1,表示Precision比Recall重要（F1分数）
- 如果取大于1,表示Recall比Precision重要

59

2.3 机器学习的数据