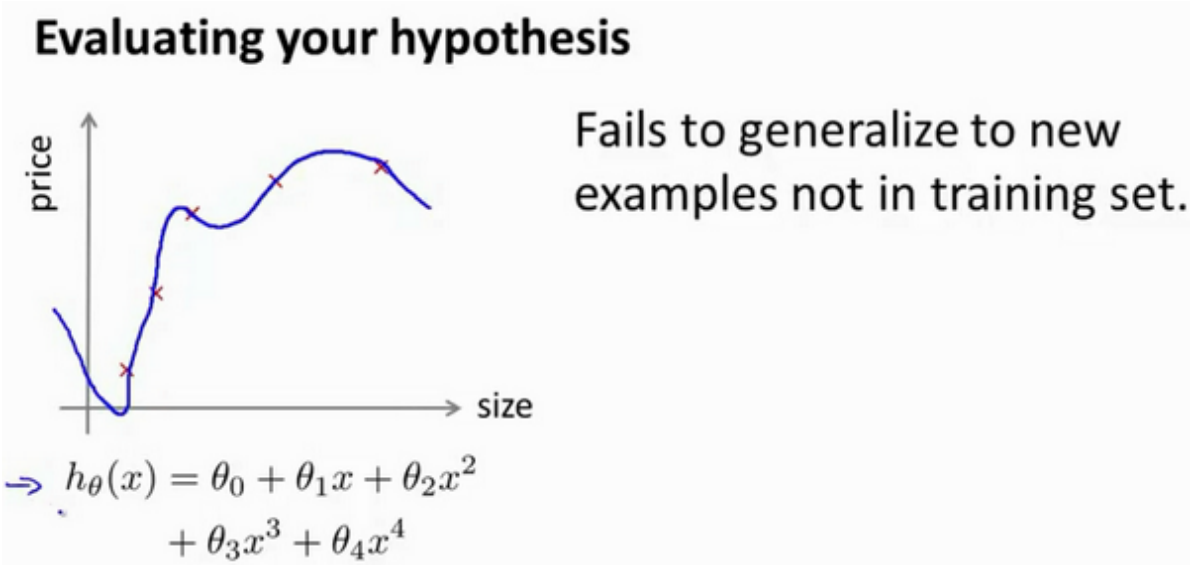
十、应用机器学习的建议(Advice for Applying Machine Learning)

10.1 决定下一步做什么

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

10.2 评估一个假设

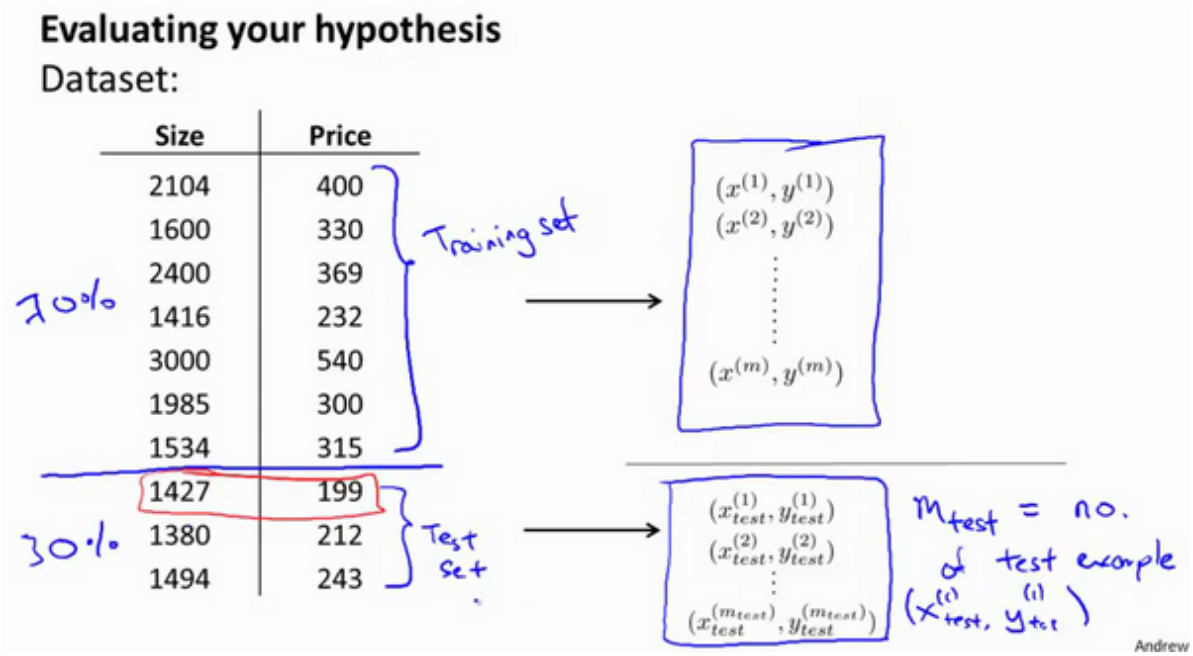


当我们确定学习算法的参数的时候，我们考虑的是选择参量来使训练误差最小化，有人认为得到一个非常小的训练误差一定是一件好事，但我们已经知道，仅仅是因为这个假设具有很小的训练误差，并不能说明它就一定是一个好的假设函数。而且我们也学习了过拟合假设函数的例子，所以这推广到新的训练集上是不适用的。

那么，你该如何判断一个假设函数是过拟合的呢？对于这个简单的例子，我们可以对假设函数$h(x)$进行画图，然后观察图形趋势，但对于特征变量不止一个的这种一般情况，还有像有很多特征变量的问题，想要通过画出假设函数来进行观察，就会变得很难甚至是不可能实现。

因此，我们需要另一种方法来评估我们的假设函数过拟合检验。

为了检验算法是否过拟合，我们将数据分成训练集和测试集，通常用70%的数据作为训练集，用剩下30%的数据作为测试集。很重要的一点是训练集和测试集均要含有各种类型的数据，通常我们要对数据进行“洗牌”，然后再分成训练集和测试集。



测试集评估在通过训练集让我们的模型学习得出其参数后，对测试集运用该模型，我们有两种方式计算误差：

1. 对于线性回归模型，我们利用测试集数据计算代价函数J
2. 对于逻辑回归模型，我们除了可以利用测试数据集来计算代价函数外：

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

误分类的比率，对于每一个测试集样本，计算：

$$err(h_\theta(x), y) = \begin{cases} 1 \; if \; h(x) \geq 0.5 \; and \; y = 0, or \; if \; h(x) < 0.5 \; and \; y = 1 \\ 0 \; Otherwise \end{cases}$$

然后对计算结果求平均。

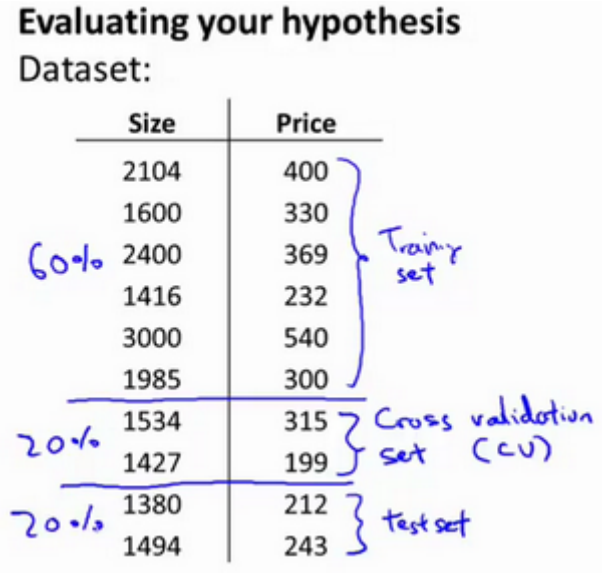
10.3 模型选择和交叉验证集

假设我们要在10个不同次数的二项式模型之间进行选择：

$$\begin{aligned} &1. \quad h_\theta(x) = \theta_0 + \theta_1 x \\ &2. \quad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \\ &3. \quad h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3 \\ &\quad \vdots \\ &10. \quad h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10} \end{aligned}$$

显然越高次数的多项式模型越能够适应我们的训练数据集，但是适应训练数据集并不代表着能推广至一般情况，我们应该选择一个更能适应一般情况的模型。我们需要使用交叉验证集来帮助选择模型。

即：使用60%的数据作为训练集，使用 20%的数据作为交叉验证集，使用20%的数据作为测试集



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）

Train/validation/test error

Training error:

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

Cross Validation error:

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

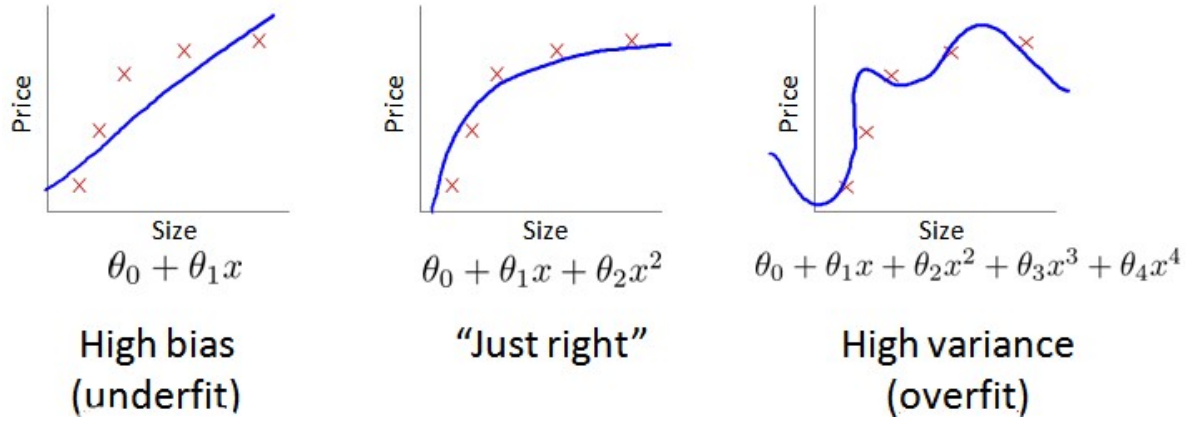
Test error:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

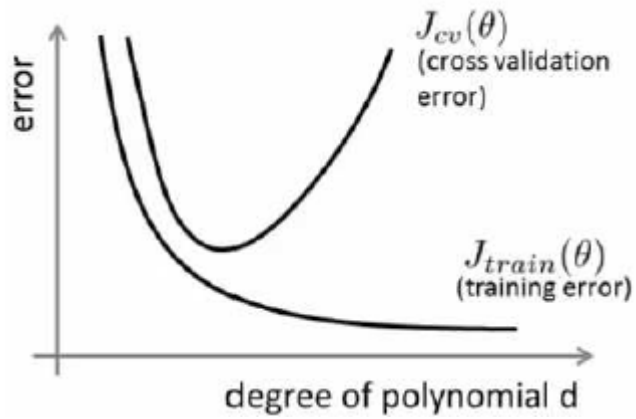
10.4 诊断偏差和方差

高偏差和高方差的问题基本上来说是欠拟合和过拟合的问题。
模型的期望预测误差可以分解为：$E[(y - \hat{f}(x))^2] = \text{Bias}^2 + \text{Variance} + \text{Noise}$
即：

- **Bias²：** 模型假设带来的系统性误差（模型太简单 → 欠拟合）
- **Variance：** 模型对数据敏感造成的不稳定（模型太复杂 → 过拟合）
- **Noise：** 数据本身的随机噪声（无法消除）



我们通常会通过将训练集和交叉验证集的代价函数误差与多项式的次数绘制在同一张图表上来帮助分析：



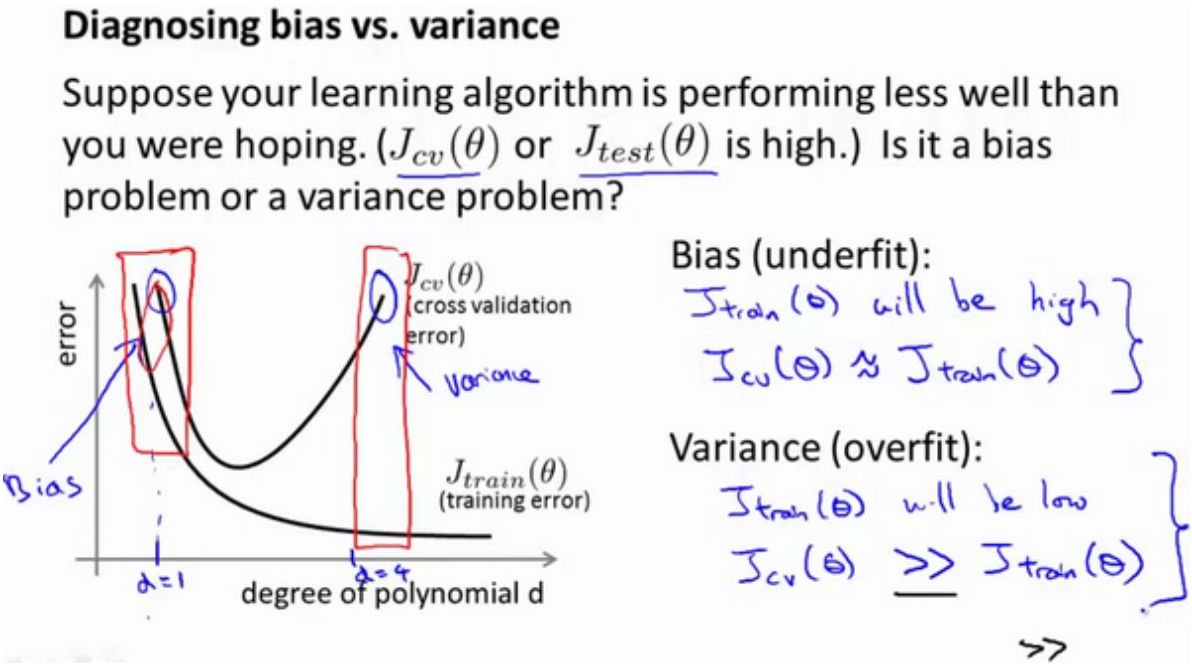
对于训练集，当 d 较小时，模型拟合程度更低，误差较大；随着 d 的增长，拟合程度提高，误差减小。

对于交叉验证集，当 d 较小时，模型拟合程度低，误差较大；但是随着 d 的增长，误差呈现先减小后增大的趋势，转折点是我们的模型开始过拟合训练数据集的时候。

Bias/variance

Training error: $$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

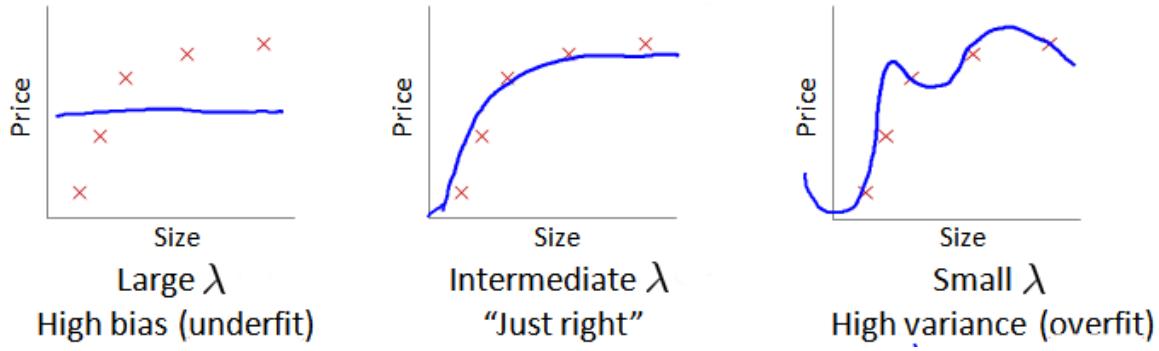
Cross Validation error: $$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$



训练集误差和交叉验证集误差近似时：偏差/欠拟合
交叉验证集误差远大于训练集误差时：方差/过拟合

10.5 正则化和偏差/方差

在我们在训练模型的过程中，一般会使用一些正则化方法来防止过拟合。但是我们可能会正则化的程度太高或太小了，即我们在选择λ的值时也需要思考与刚才选择多项式模型次数类似的问题。



我们选择一系列的想要测试的 λ 值，通常是 0-10之间的呈现2倍关系的值（如：$0, 0.01, 0.02, 0.04, 0.08, 0.15, 0.32, 0.64, 1.28, 2.56, 5.12, 10$共12个）。 我们同样把数据分为训练集、交叉验证集和测试集。

Choosing the regularization parameter λ

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

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

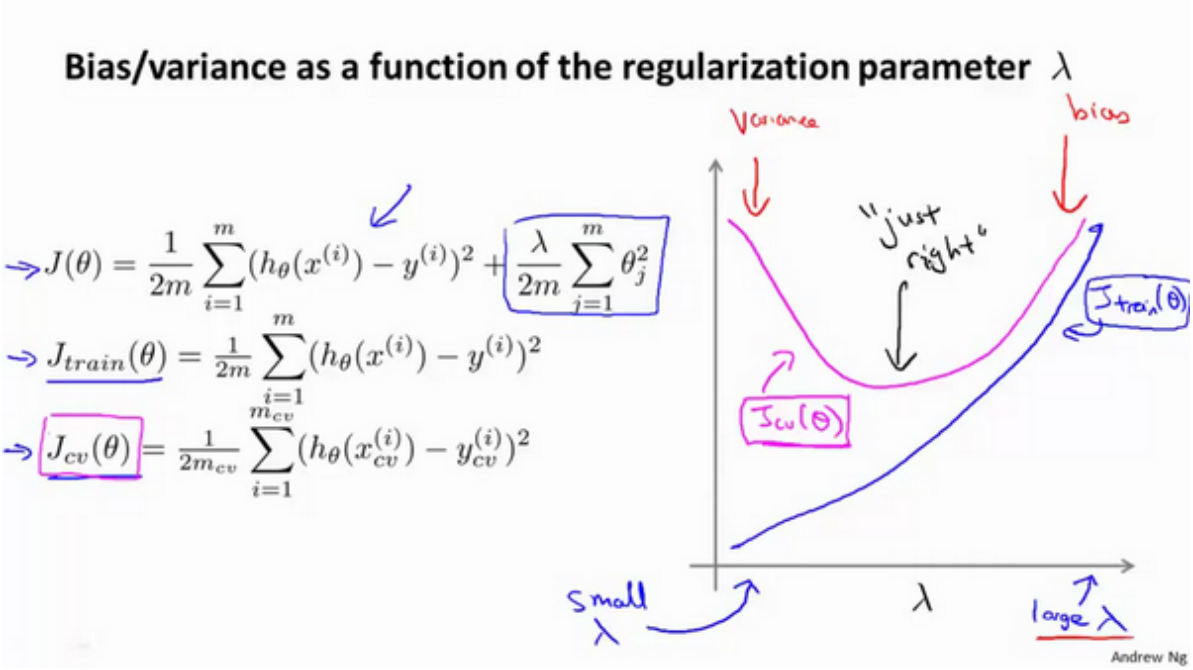
1. Try $\lambda = 0$
2. Try $\lambda = 0.01$
3. Try $\lambda = 0.02$
4. Try $\lambda = 0.04$
5. Try $\lambda = 0.08$

   ⋮

12. Try $\lambda = 10$

选择λ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：

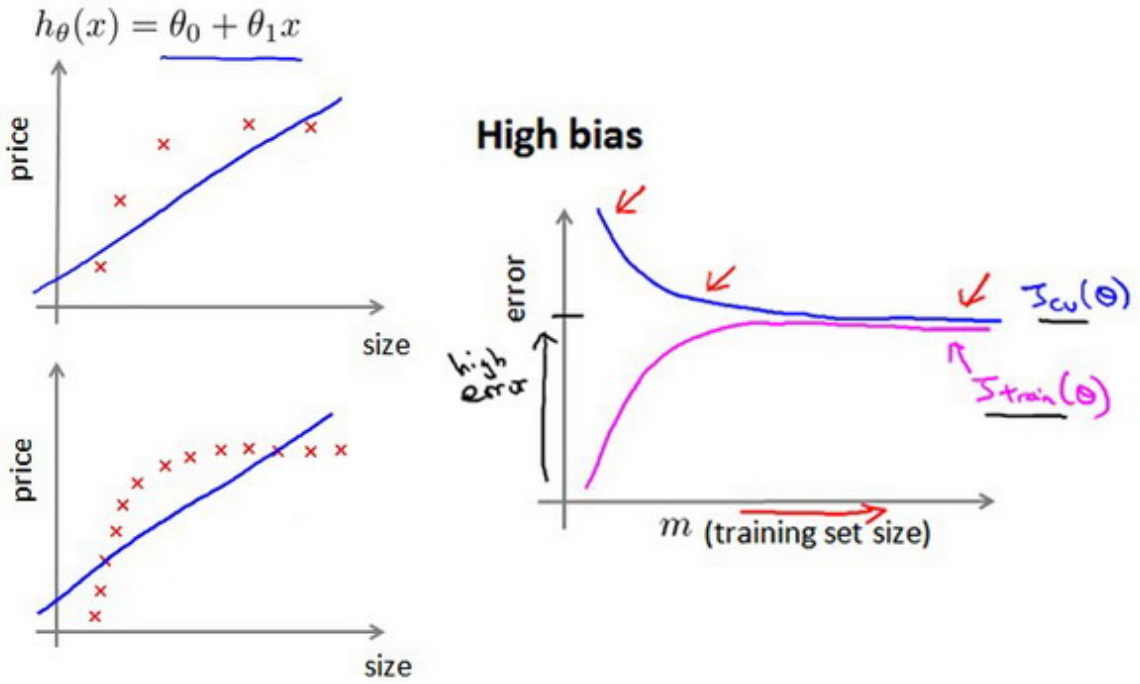


• 当 λ 较小时，训练集误差较小（过拟合）而交叉验证集误差较大
• 随着 λ 的增加，训练集误差不断增加（欠拟合），而交叉验证集误差则是先减小后增加

10.6 学习曲线

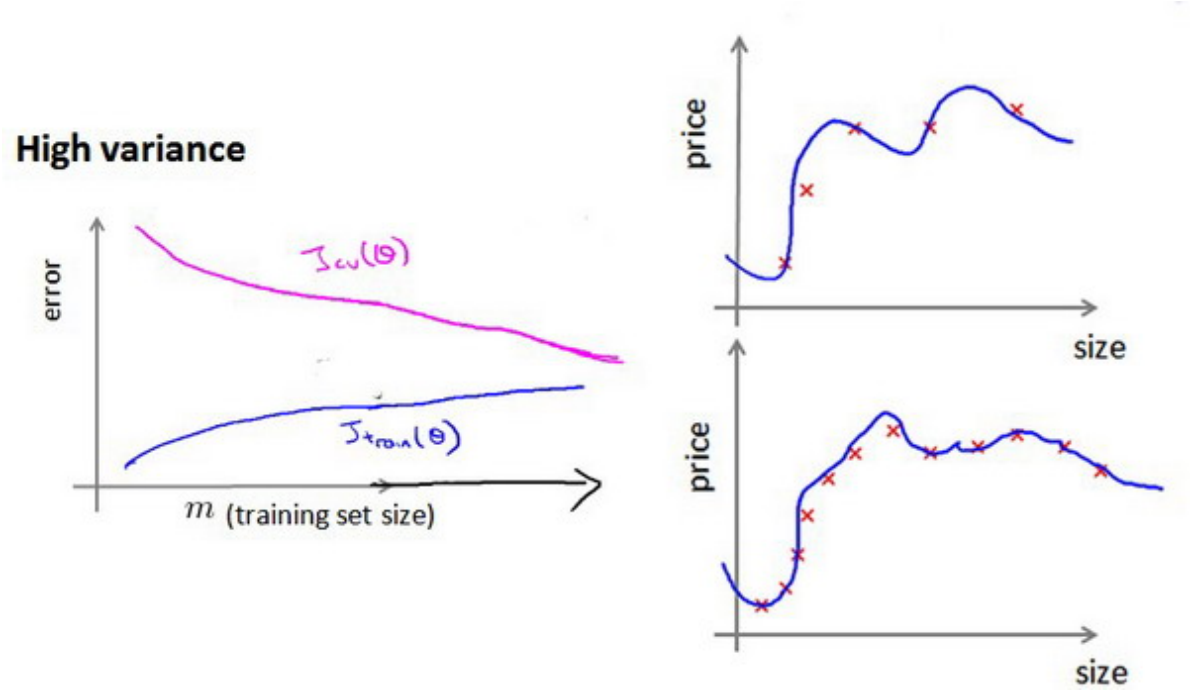
学习曲线就是一种很好的工具，我经常使用学习曲线来判断某一个学习算法是否处于偏差、方差问题。学习曲线是学习算法的一个很好的**合理检验（sanity check）**。学习曲线是将训练集误差和交叉验证集误差作为训练集样本数量（m）的函数绘制的图表。

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。



也就是说在高方差/过拟合的情况下，增加更多数据到训练集可能可以提高算法效果。

10.7 决定下一步做什么

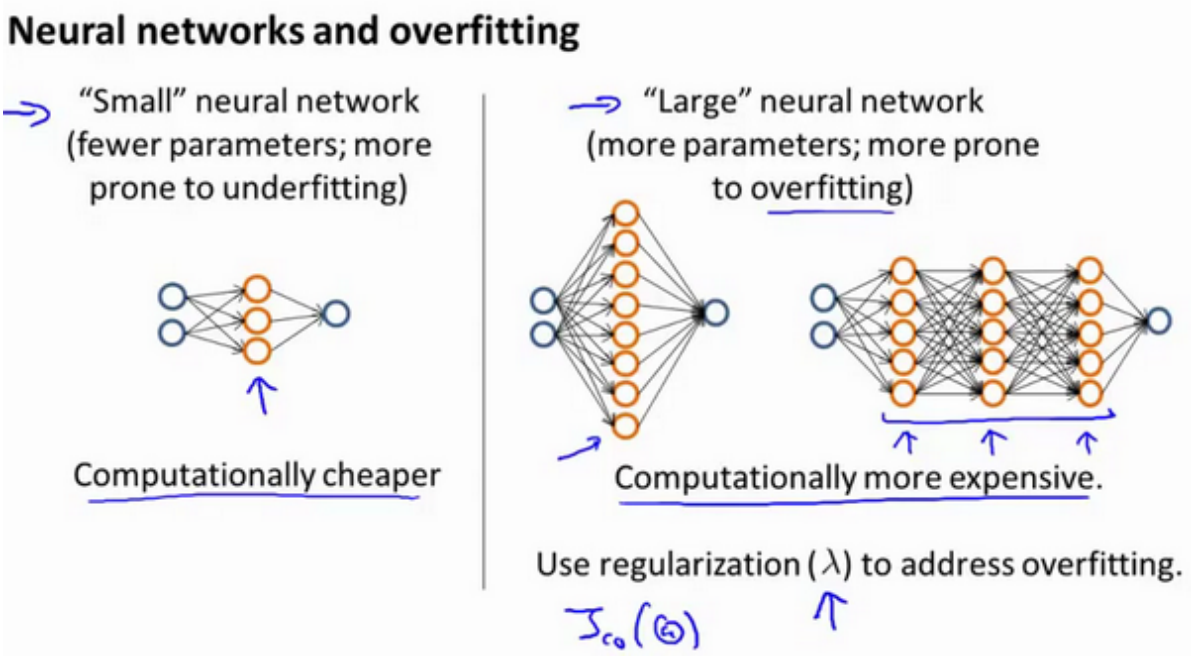
1. **获得更多的训练样本——解决高方差**
2. 尝试减少特征的数量——解决高方差
3. 尝试获得更多的特征——解决高偏差
4. 尝试增加多项式特征——解决高偏差
5. 尝试减少正则化程度λ——解决高偏差
6. 尝试增加正则化程度λ——解决高方差

解释：

增加 λ → 降低模型复杂度 → 降低方差 → 可能增加偏差

减小 λ → 提高模型复杂度 → 降低偏差 → 可能增加方差

神经网络的方差和偏差：



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。