10 应用机器学习的建议

2022年11月8日 21:02

10.1决定下一步怎么做

Debugging a learning algorithm:
Suppose you have implemented regularized linear regression to predict housing prices.

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

However, when you test your hypothesis on a new set of houses, you find that it makes unacceptably large errors in its predictions. What should you try next?

依然以预测房价为例，假设我们已构建了带有正则化的代价函数并据此搭建了预测房价的模型，但是当对其测试后发现预测值与实际值的误差很大，那么接下来该怎么做？

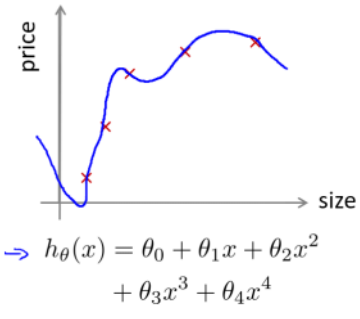
- Get more training examples
- Try smaller sets of features $x_1, x_2, x_3, \ldots, x_{100}$
- Try getting additional features
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, \text{etc.})$
- Try decreasing λ
- Try increasing λ

- 一种方法是，得到更多的训练样本(没啥用)
- 选用更少的特征防止过拟合
- 获取更多的特征量来收集更多的数据
- 增加特征多项式
- 减小或增大正则化参数的值

选择哪种方法很关键。

机器学习诊断法(Machine learning diagnostic):
通过执行一种测试来了解算法哪里出了问题，也能得知要想改进一种算法的效果怎样的尝试是有意义的。这些诊断法的实现是需要花费一些时间的。

10.2评估假设



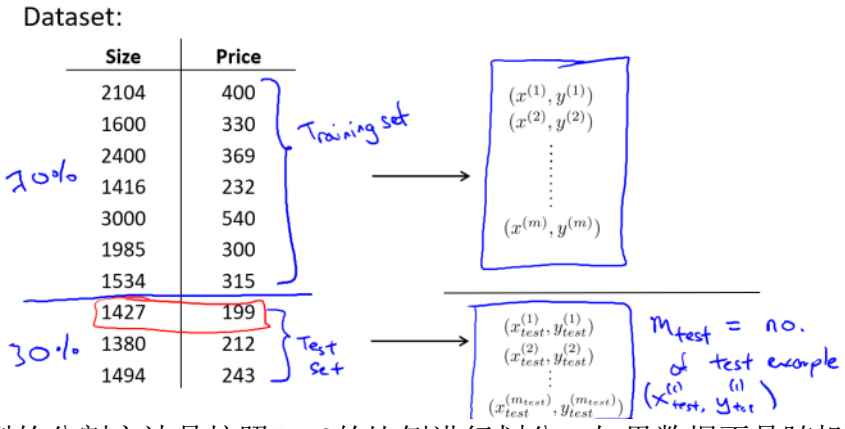
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Fails to generalize to new examples not in training set.

x_1 = size of house
x_2 = no. of bedrooms
x_3 = no. of floors
x_4 = age of house
x_5 = average income in neighborhood
x_6 = kitchen size
$\vdots$
x_{100}

对于过拟合的假设，其无法泛化到新的样本。

一种评估的方法是将数据分为训练集和测试集，



其中一种典型的分割方法是按照7：3的比例进行划分。如果数据不是随机排列的，那么最好应该打乱顺序。

- Learn parameter θ from training data (minimizing training error $J(\theta)$) 70%
- Compute test set error:

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta(x^{(i)}_{test}) - y^{(i)}_{test} \right)^2$$

具体应用到线性模型上，就是拿70%的数据进行最小化参数的求解训练模型，剩下30%的数据用来计算误差。

对于逻辑回归也是同理：

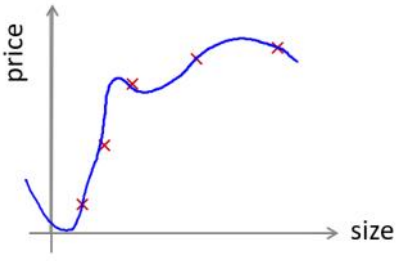
- Learn parameter θ from training data
- Compute test set error:

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

- Misclassification error (0/1 misclassification error):

用70%的数据用来训练参数，然后用剩下30%的数据进行误差计算和评估，另一种形式的测试度量是错误分类(Misclassification error)用来表示分类正确或错误的情况。

10.3模型选择和训练、验证、测试集



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

Once parameters $\theta_0, \theta_1, \ldots, \theta_4$ were fit to some set of data (training set), the error of the parameters as measured on that data (the training error $J(\theta)$) is likely to be lower than the actual generalization error.

我们已经知道，在过拟合中学习的算法虽然能很好地拟合训练集但它并不是一个好的假设，所以训练误差不能用来判断新样本的拟合好坏。

下面来考虑样本选择的问题，

Model selection

d = degree of polynomial

1. $h_\theta(x) = \theta_0 + \theta_1 x$ → $\theta^{(1)}$ → $J_{test}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ → $\theta^{(2)}$ → $J_{test}(\theta^{(2)})$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$ → $\theta^{(3)}$ → $J_{test}(\theta^{(3)})$

⋮

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

Choose $\theta_0 + \ldots \theta_5 x^5$

How well does the model generalize? Report test set error $J_{test}(\theta^{(5)})$.

Problem: $J_{test}(\theta^{(5)})$ is likely to be an optimistic estimate of generalization error. I.e. our extra parameter (d = degree of polynomial) is fit to test set.

假设d为我们需要选择的假设函数的最高次幂，那么此时除了模型参数 θ 外我们还需要考虑d这一参数。我们可以依次对不同d下的假设函数通过拟合数据集得到参数 θ ，从而利用测试集计算出误差，根据哪个模型对应的是最小误差进行选择。

比如最后我们从这10个模型中判断出第五个模型的误差最小，从而选择第五个假设函数，但是当它应用到新的样本上时，我们依然不能很公平地估计出其泛化能力。这是因为我们拟合了一个新的参数d，也就是多项式的次数，而它的取值是我们建立在已有的测试集的基础上确定的，所以假设函数可能对测试集的表现好过新的未经测试的样本。

从而可以采取下面的方法进行解决：

Evaluating your hypothesis

Dataset:

	Size	Price	
60%	2104	400	Training set
	1600	330	
	2400	369	
	1416	232	
	3000	540	
	1985	300	
20%	1534	315	Cross validation set (CV)
	1427	199	
20%	1380	212	test set
	1494	243	

$(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, ⋮, $(x^{(m)}, y^{(m)})$

$(x_{cv}^{(1)}, y_{cv}^{(1)})$, $(x_{cv}^{(2)}, y_{cv}^{(2)})$, ⋮, $(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$ — m_{cv} = no. of cv examples $(x_{cv}^{(i)}, y_{cv}^{(i)})$

$(x_{test}^{(1)}, y_{test}^{(1)})$, $(x_{test}^{(2)}, y_{test}^{(2)})$, ⋮, $(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$ — m_{test}

我们不再将数据分为两部分，而是三部分：训练集(training set)、交叉验证集(cross validation set)、测试集(test set)。一般来说它们分别占比60%、20%、20%。我们据此可以定义三者的误差：

Training error:

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

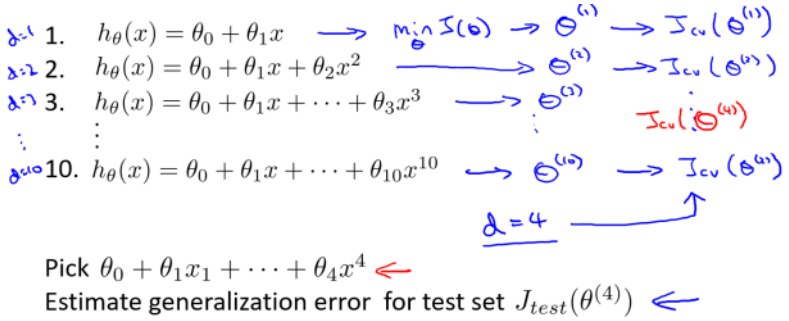
Cross Validation error:

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

Test error:

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

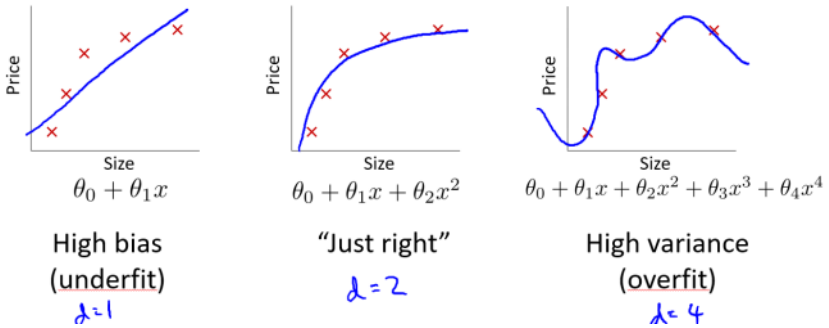
那么回到模型的选择问题上：



我们将使用验证集来选择模型，而不是原来的测试集。具体来讲，首先通过训练集找到每一个假设对应的最小化参数，接下来我们用交叉验证集来测试假设函数的误差，选择最小误差的假设作为我们的模型。比如最后发现第四个模型的误差最小，那么我们就选择四次多项式模型。最后，我们用测试集的误差来估计模型的泛化误差。

10.4诊断偏差与方差

当我们运行一个学习算法时，如果算法表现不理想，那么大概率是两种情况：偏差(bias)或方差(variance)较大，要么是欠拟合、要么是过拟合。如何判断是一个重要的问题。

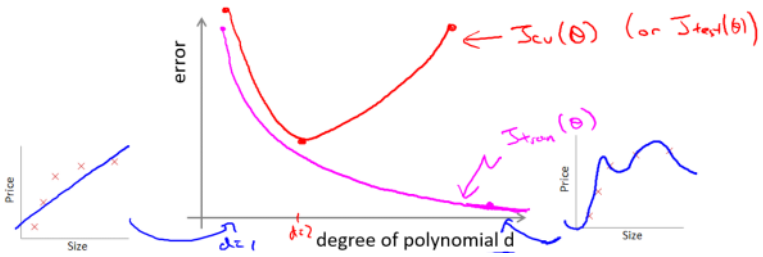


依然是这幅图，反映了欠拟合、恰好拟合和过拟合三种情况。

沿用上述所讲的训练误差和验证误差的定义，

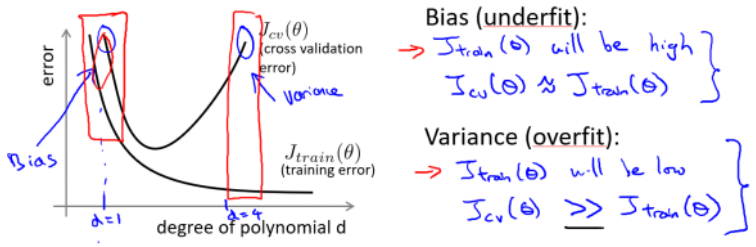
Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

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



横坐标为多项式的次数，纵坐标为误差。随着次数的增加，拟合的程度越来越高。画出训练误差：随着多项式次数的增加，模型对训练集拟合得越来越好，以至于到更高的次数时误差已经非常小了；画出交叉验证误差：当次数很小的时候，不能够很好地拟合训练集，从而交叉验证误差也会很大；而用中等大小次数的多项式来拟合的时候，会得到一个更小的交叉验证误差；但当次数更大时，发生了过拟合，就会得到一个较大的交叉验证误差。

假设我们得出了一个学习算法，但是它的效果没有期望中的好，那么我们该如何判断它出现了高偏差还是高方差的问题？



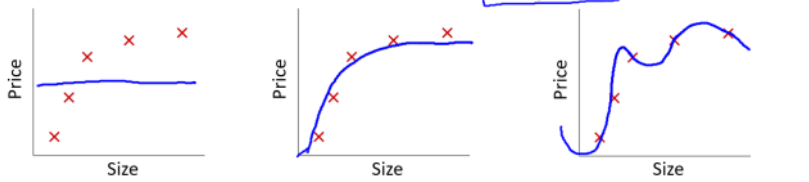
根据曲线可以很好地判断。总结来说：当训练误差和交叉验证误差都很大且很接近时，此时大概率发生的是欠拟合、有高偏差；反过来，当训练误差很小、交叉验证误差很大，且二者相差很大时，此时大概率发生的是过拟合、有高方差。

10.5正则化和偏差、方差

假设我们要对一个高阶的多项式进行拟合，

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

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



Large λ
High bias (underfit)
$\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$
$h_\theta(x) \approx \theta_0$

Intermediate λ
"Just right"

Small λ
High variance (overfit)
$\lambda = 0$

我们通过加入正则化项来防止过拟合。

当正则化参数设置得很大时，惩罚力度会很大，假设中的参数大部分都接近0了。最后的拟合结果接近一条水平直线，产生了欠拟合的问题。因此，这个假设处于高偏差。

另一种极端的情况是，当正则化参数很小，通常会出现过拟合的情况，此时假设处于高方差。

当取一个大小适中的正则化参数时，才能得到一组对数据拟合比较合理的参数值。

如何自动选择一个最佳的正则化参数？

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

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

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

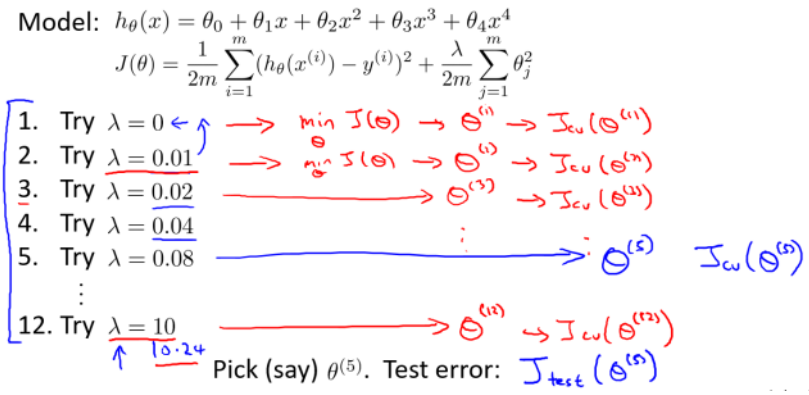
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

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

$J(\theta)$

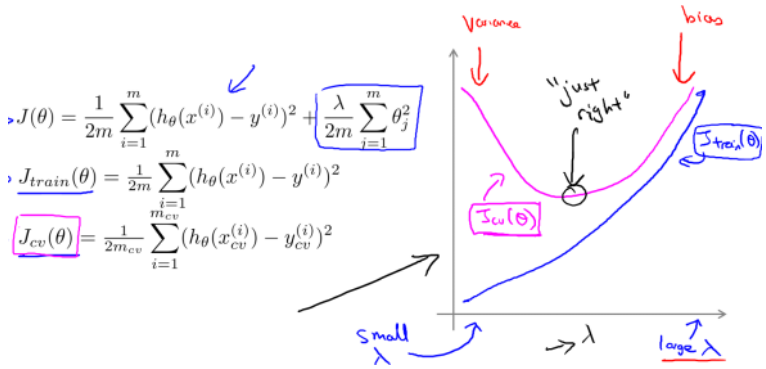
J_{train}
J_{cv}
J_{test}

定义训练误差，在原有的代价函数也即目标函数的基础上去掉了正则化项，与此类似地，定义交叉验证误差和测试集误差。



按照我们以往的思路，是分别对正则化参数进行赋值，最小化代价函数得到假设参数，然后用交叉验证误差对其进行评价，选取误差最小的模型作为最终选择。最后看看它在测试集上的表现如何，用其作为该模型在新样本上的泛化性能。

随着正则化参数的变化，训练误差和验证误差怎么变化？

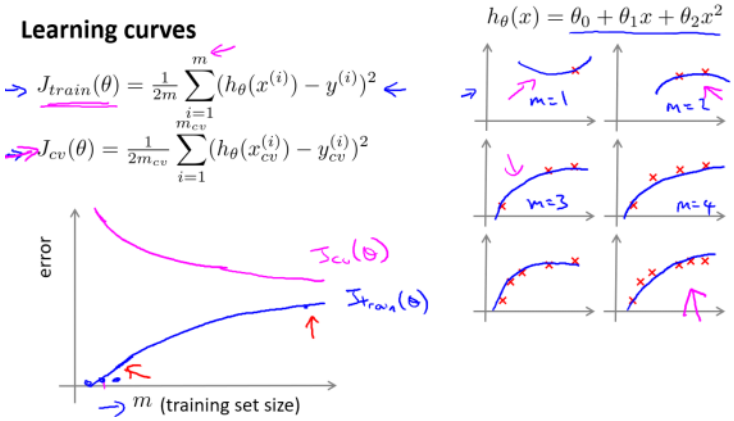


当正则化参数很小时，很有可能发生过拟合，对参数拟合的很好，训练误差很小，而交叉验证误差很大；而当很大时，很可能发生欠拟合，对参数拟合不足，训练误差很大，交叉验证集误差也会很大；而总会有一个中间值。使训练误差和验证误差都相对较小。

10.6学习曲线

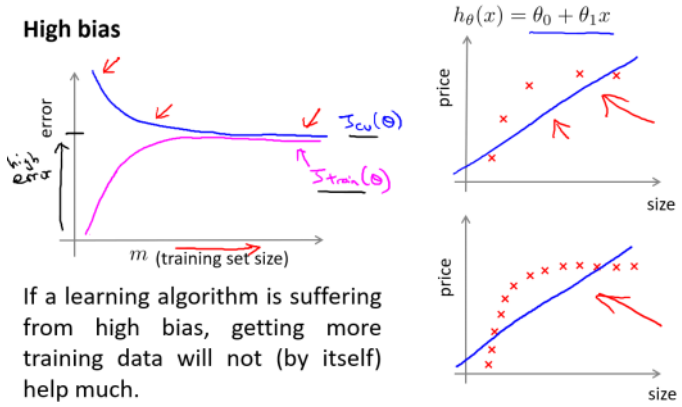
检查算法是否正常以及改进算法可以通过学习曲线这种工具 。

首先绘制不同样本数量下的训练误差和验证误差，



当样本数量很少时，能够拟合得非常好，此时的训练误差会很小，但泛化性能不好，此时的验证误差会很大；而随着训练样本容量的增加，要保证对所有样本拟合得很好就越来越困难，训练误差也会越来越大，但由于训练样本很多，泛化性能会相对可观，那么验证误差会下降。

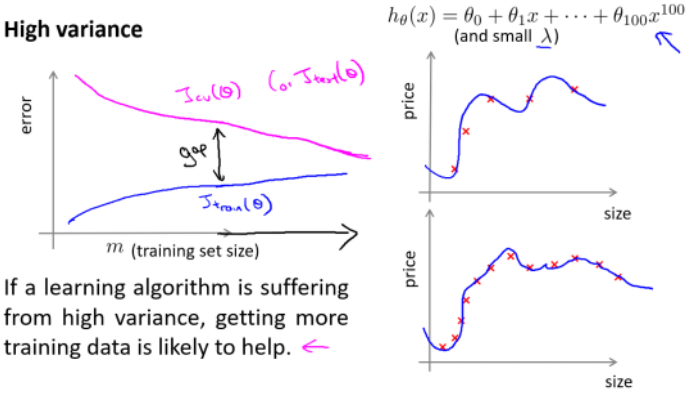
假设我们的模型发生了欠拟合，也即高偏差，



If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

可以想见，无论是样本数量很好还是样本数量很多，都不会对数据集拟合得很好，测试误差都会很大。而训练误差一开始会很小，但当样本容量增加时，验证误差也会升高，逐步接近训练误差。从这里我们也可以看出，当算法的验证误差很高的时候，即使增加训练样本的数量对算法的改进也无益。

假设模型发生了过拟合，也即高方差，



If a learning algorithm is suffering from high variance, getting more training data is likely to help.

当训练数量很少、正则化参数很小时，会发生相应的过拟合，此时训练误差很小，而由于过拟合，验证误差很大；随着数量的增加，更好地拟合全部样本变得越来越困难，训练误差也随之增大，虽然随着样本数量的增加，验证误差会有所缓解，但其依然很高。两个曲线有逼近的趋势。
这个曲线也反映出，当发生过拟合时，使用更多的训练数据对算法是有帮助的。

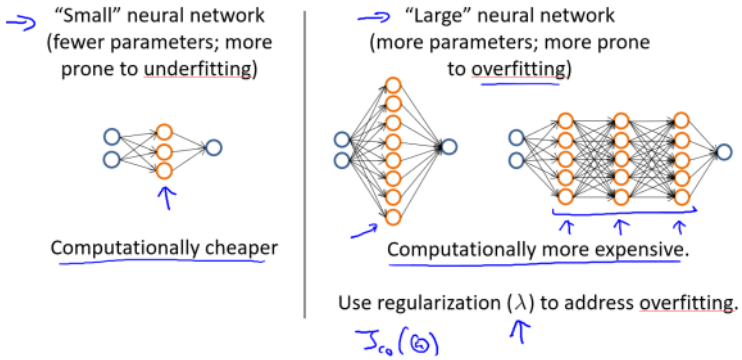
10.7决定接下来做什么

如前所述，当我们发现算法有问题时提出了以下的解决方案：

- Get more training examples → fixes high variance
- Try smaller sets of features → fixes high variance
- Try getting additional features → fixes high bias
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, \text{etc})$ → fixes high bias.
- Try decreasing λ → fixes high bias
- Try increasing λ → fixes high variance

但怎么判断哪种方案是有效的呢？

- 获得更多的训练样本对于解决高方差问题有效
- 减少特征量对高方差问题有效
- 增加特征数目有助于解决高偏差问题
- 增加特征多项式也相当于增加特征量，是一种用于修正高偏差问题的方式
- 减小正则化因子可以修正高偏差
- 增大正则化因子可以修正高方差



选用比较简单的神经网络模型，参数不多，容易出现欠拟合，其优势在于计算量较小；拟合较大型的神经网络结构，隐含单元数很多，参数较多，容易出现过拟合，其劣势在于计算量很大，但通常不是最主要的，如果发生过拟合可以通过正则化的方法来修正。在选择多少层隐藏层的问题上，依然可以将数据分割为训练集、验证集、测试集进行评估，看看哪种数量层数的模型是比较理想的。