

Evaluating a Learning Algorithm.

如果模型效果不好，我们可能会尝试以下方式：

1. 增添新数据
2. 减少/增加特征
3. 调参
4. 更复杂的模型/特征组合

但如果模型在训练数据上效果很好，但仍有可能是不好的（如过拟合）

因此为了验证模型的泛化能力，将数据集分为训练集和测试集

- 在训练集上最小化 $J_{train}(\theta)$ 和学习参数 θ
- 在测试集上计算 $J_{test}(\theta)$

如：模型选择（可能完全不同模型：决策树 VS 神经网络；线性模型的不同 hypothesis；同一模型不同参数等）

$$d=1 \quad h_{\theta}(x) = \theta_0 + \theta_1 x \rightarrow \theta^{(1)} \rightarrow J_{test}(\theta^{(1)})$$

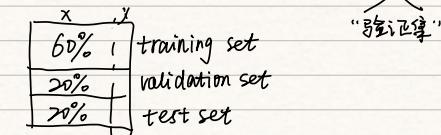
$$d=2 \quad h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \rightarrow \theta^{(2)} \rightarrow J_{test}(\theta^{(2)}) \times$$

$$d=3 \quad h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \rightarrow \theta^{(3)} \rightarrow J_{test}(\theta^{(3)})$$

但有个问题， d 作为超参数，也可能过于拟合测试集！

[想法：如果你是根据模型在某一部分数据上的表现来选择模型，那就有可能过拟合，不能衡量泛化误差！]

因此，在训练集中再划分一部分数据用于调整模型“超参数” → 一种无法在训练中优化但可以手动选择的参数。



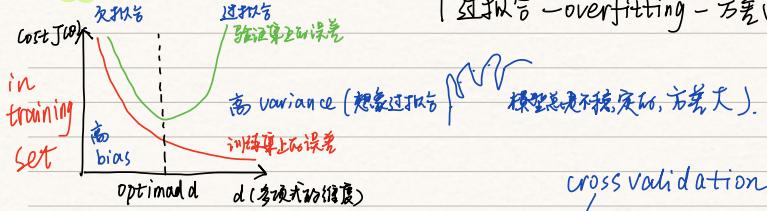
三个部分：训练集：在训练集中优化参数 θ ，找到一组最佳参数实现该模型的最佳性能。——平时练习

验证集：调整不同的“超参数”组合（类似模型选择）——思考

测试集：衡量模型泛化能力 —— 预测

单数据量时

Bias vs. Variance



{ 欠拟合 - underfitting - 偏差 bias：模型无法继续减小训练误差（如模型太弱）

过高拟合 - overfitting - 方差 variance：训练误差远小于验证误差（训练集太小）

/ 模型复杂度

or 训练集太多时

但数据集大小与模型复杂性无关

数据越多，我们越可能尝试拟合一一个更复杂的模型；反之。

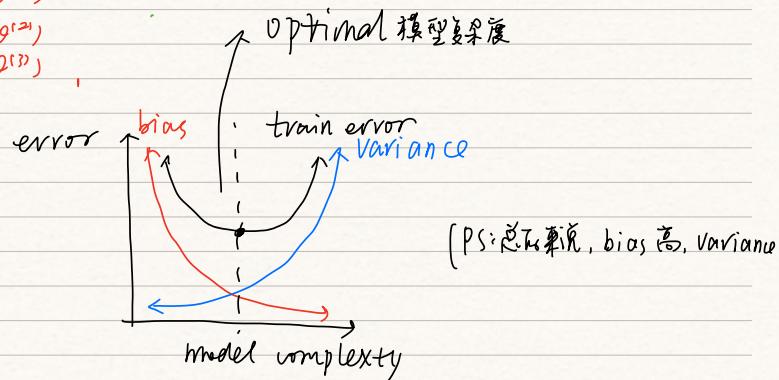
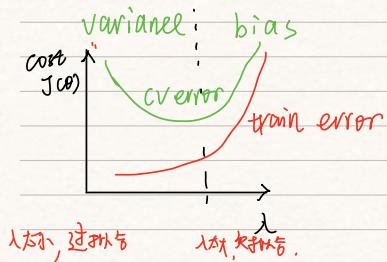
添加正则项后（只在 $cost J_{train}(\theta)$ 上加）， $J_{cv}(\theta)$ 及 $J_{test}(\theta)$ 上不加正则项！($\lambda=0$)

在上面找到了最佳的 d : $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

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

λ 作为超参数：

$$\begin{aligned}\lambda = 0 &\rightarrow \text{Min } J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)}) \\ \lambda = 0.01 &\rightarrow \text{Min } J(\theta) \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)}) \\ \lambda = 0.02 &\rightarrow \text{Min } J(\theta) \rightarrow \theta^{(3)} \rightarrow J_{cv}(\theta^{(3)})\end{aligned}$$



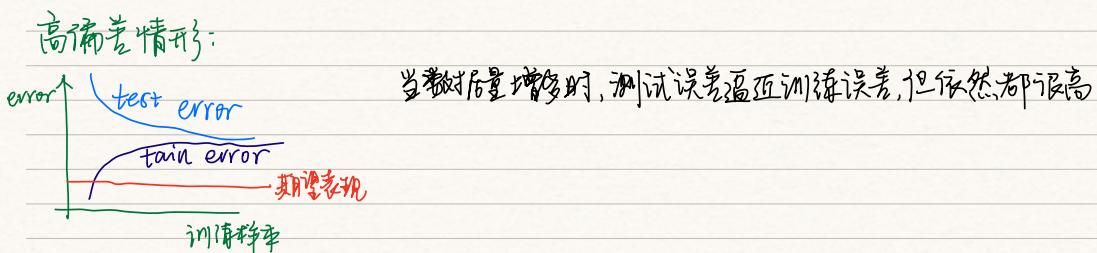
Learning Curve

学习曲线也是由训练误差和验证集上的误差（测试误差）组成的。

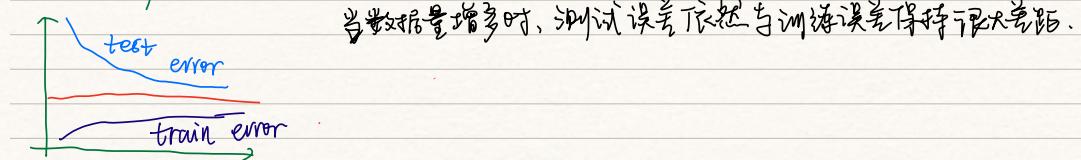
学习曲线目的：检查算法是否工作正常，改进算法。横轴是样本数。

两个思想误区：
 ① 数据量少时，模型效果不好。训练数据量越大，模型效果越好。
评判一个模型好不好，主要看的是泛化误差。但对训练误差而言，数据量少时，训练误差总很小的！（可以完美拟合一两个样本点）
 ② 数据量尽管增大，但对模型的提升依然有限。泛化误差一开始倾向于减小，但最终只能接近训练误差且无法小于其。还可能依然与训练误差保持较大差距（过拟合）。

更正：模型一定时，其训练误差反而越大



高方差情形



① 想考：当数据量少时，不可能欠拟合。只有数据量多时，才可能有欠拟合问题。因此增加训练数据无法解决欠拟合，只能从调整模型出发。

② 当数据量少时，总是过拟合。当数据量多时，可能缓解过拟合，也可能不缓解。

③ 模型一定时，增加训练样本对模型的提升是有限的。因为泛化误差只能逼近训练误差；但训练误差会随着数据增多，误差也会增大。（假设数据中存在噪声）

④ 为什么学习曲线的图像和前面分析正则项和模型复杂度时的图像不一样？

答：因为自变量不同。 起步

在学习曲线的一个图上仅可体现过去以及此次对以后之二

但若转动是入/ d , 则可在同一个圆上同时体现过桥与欠桥。

木質型診斷總結：

解决高方差(过拟合): 增大样本量, 减少特征数量, 加大正则项参数
解决高偏差(欠拟合): 增加特征数量, 添加多项式特征, 减小正则项参数

对神经网络而言，也可以通过增加层数和总数来解决欠拟合，增加正则项解决过拟合。

Spam classifier

检测垃圾邮件：从邮件中收集 1000~5000 个常用词，做为词向量。

垃圾邮件中可能更常含有 buy, deal, discount 等词语

其他提高准确率的方法：①收集大量数据

②开发复杂的特征(如利用标题数据,发件人等)

③ 识别并写错误.

Error Analysis

当算法效果不好时，可以采用以下方式：

① 从一个简单的算法开始，快速实现，并看其交叉验证的表现

② 画学习曲线图

③ 手动分析错误样本.

另外：词是用各自形式，还是统一成词根(stemming) eg. *discomfort*, *discomforts*, -ed, -ing > 以方便快速验证！看数据对比
大写还是小写？

有想法就趁早越好，初步实践看看效果！

精度 Accuracy & 召回率 Recall

背景：仅根据准确率 accuracy，在数据分布是偏斜（skew）的情况下

例如，对于二分类而言，若99%的人都没患病，那么一个总是预测为未患病的模型，也有99%的准确率。

因此，提出精度(查准率)(Precision)和召回率(查全率)(Recall)两个指标来验证模型性能。

计算方式:		实际	预测		1	0
		TP	FP			
1	TP					
	FN					
0	FP					
	TN					

1/0：1是我们关注的类别，0是不关注的其他类别

Positive / Negative : 由预测值 = 1 / 0 决定

True / False : 由真实值 = 1/0 决定

注的样本。】

Precision = $\frac{TP}{TP+FP}$, 理解: 直观上看, 精度和准确度的区别就在于准确度关心所有类别上任何一个正确率, 即使是标签为0的那些我们不关心。精度衡量的是模型对于那些标签为1的样本预测正确的能力。

Recall = $\frac{TP}{TP+FN}$, 理解: 召回率是指,能不能把所有标签为1的样本都找出来的能力。

两个法官 < 精度高的：小心谨慎，难以做决定但往往决策正确，没有冤假错案，能一眼识别好人坏人。
召回率高：马虎大意，但勇于尝试，无论错对时的都去改正，相比之下经验更丰富，救了更多人。

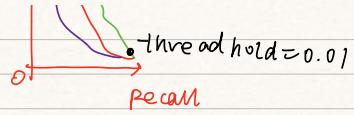
两个法官 < 精度高的：小心谨慎，难以做出决定但往往决策正确，没有冤假错案，能一眼识别好人坏人。
召回率高的：马马虎虎，但勇于尝试，无论错对的都去试，相比之下经验丰富，救了更多人。

① Logistic Regression

precision threshold = 0.99

threshold: 0.5

if $h(x) \geq 0.5$ predict 1
 < 0.5 predict 0



阈值降低 \swarrow 没识别为正类的样本降低，召回率提高。

更容易识别为正类，就有可能把负类识别为正类，精度降低。

阈值提高 \swarrow 不会轻易把负类识别为正类，准确率提高。

召回率降低。

② 设计一个指标，综合 P 和 R。

e.g. $\frac{P+R}{2}$: 若 $P=0.99, R=0.1$, 平均值也很大

F值/F1值: $\frac{2PR}{P+R}$: $P=R=0$ 时, $F_1=0$. ✓