

结构化机器学习项目--机器学习策略（1）

1.1 为什么是ML策略

什么是机器学习策略？

机器学习策略是能改善系统性能的方法。

ideas：

- 收集更多数据
- 收集更diverse训练集
- 使用梯度下降法训练算法，训练久一点
- 尝试Adam算法
- 使用规模更大的神经网络
- 使用规模小的神经网络
- 尝试dropout
- 增加L2 正则化
- 修改网络的架构
 - 修改激活函数
 - 修改隐藏层单元

当你想优化一个深度学习网络时，你通常有很多想法可以去尝试。需要判断哪些是值得尝试的，哪些是值得放弃的。可以指引你朝着最有希望的方向前进。

事实上，深度学习策略，在深度学习时代也在变化，因为现在深度学习能做到的，和上一代机器学习不太一样。希望这些策略可以帮助你使你的系统尽快投入使用。

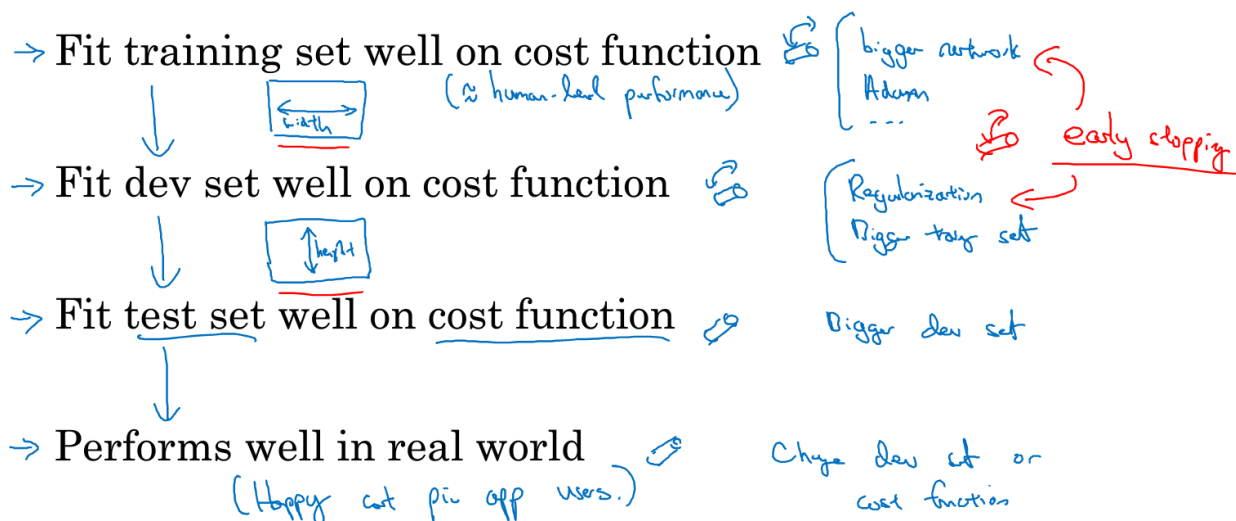
1.2 正交化

正交化的概念的理解：正如你在电视机图像太宽，你要调整电路设计师设计好的宽度按钮，如果你的图像太高，你要调整控制高度的按钮。

正交化有个概念你要非常清楚，到底是四个问题中的哪一个。

在机器学习中，我们必须清楚到底是什么地方出问题，然后找到对应的控制按钮，就可以对问题作出相应的调整，优化机器学习系统。

Chain of assumptions in ML



- 训练集在代价函数上表现得好
 - 否则，使用更大的网络，更好的算法如Adam等。
- 开发集在代价函数上表现得好
 - 否则，使用正则化，更大的训练
- 测试集在代价函数上表现得好
 - 否则，使用更大的开发集
- 在真实的系统环境中表现好
 - 否则，修改开发测试集，修改代价函数

1.3 单一数字评估指标

使用查准率和查全率的调和平均数作为新的单一评估指标。

Example : Cat vs Non- cat
y = 1, cat image detected

	Actual class y	
	1	0
Predict class \hat{y}		
1	True positive	False positive
0	False negative	True negative

查全率

$$\text{Precision (\%)} = \frac{\text{True positive}}{\text{Number of predicted positive}} \times 100 = \frac{\text{True positive}}{(\text{True positive} + \text{False positive})} \times 100$$

查准率

Of all the images that actually have cats, what fraction of it did we correctly identifying have cats?

$$\text{Recall (\%)} = \frac{\text{True positive}}{\text{Number of predicted actually positive}} \times 100 = \frac{\text{True positive}}{(\text{True positive} + \text{True negative})} \times 100$$

Classifier	Precision (p)	Recall (r)
A	95%	90%
B	98%	85%

通过调和平均数得出的评估指标

$$\text{F1-Score} = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

Classifier	Precision (p)	Recall (r)	F1-Score
A	95%	90%	92.4 %
B	98%	85%	91.0%

Classifier A is a better choice. F1-Score is not the only evaluation metric that can be use, the average, for example, could also be an indicator of which classifier to use.

通过单一指标F1-score , 我们就可以很容易的评判分类器A的效果更好。

另一个例子：

Algorithm	US	China	India	Other
A	3%	7%	5%	9%
B	5%	6%	5%	10%
C	2%	3%	4%	5%
D	5%	8%	7%	2%
E	4%	5%	2%	4%
F	7%	11%	8%	12%

Algorithm	US	China	India	Other	Average
A	3%	7%	5%	9%	6%
B	5%	6%	5%	10%	6.5%
C	2%	3%	4%	5%	3.5%
D	5%	8%	7%	2%	5.25%
E	4%	5%	2%	4%	3.75%
F	7%	11%	8%	12%	9.5%

通过计算平均，根据平均值这个单一指标，算法C的平均误差最低。

所以，你的机器学习过程，往往是你有一个想法。你尝试实现它，看它效果好不好。

单一评价指标，可以帮助你的团队，做出评价算法的决策的效率。

1.4 满足和优化指标

Another cat classification example

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms

$$\text{Cost} = \text{accuracy} - 0.5 \times \text{Running Time}$$

maximize accuracy
subject to Running Time $\leq 100 \text{ ms.}$

N metrics: 1 optimizing
N-1 satisfying

Wakewords / Trigger words

Alara, OK Google,

Hey Siri, ni hao baidu
你 好 百 度

accuracy.
#false positive

maximize accuracy.
s.t. ≤ 1 false positive
every 24 hours.

通过猫分类这个例子，算法的满足指标是运行时间，准确率是优化指标。

1.5 训练/开发/测试集的划分

如何设立开发集和测试集？

例子：

- 开发一个猫分类器。

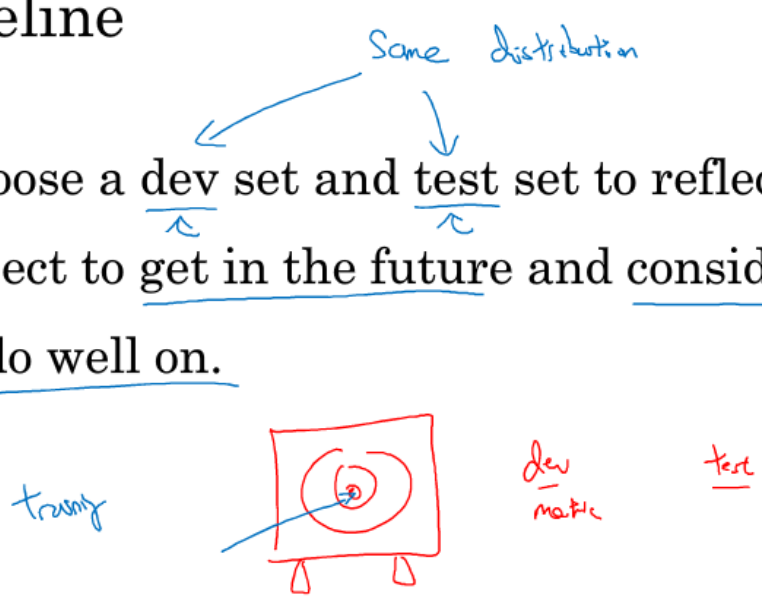
指导方针

Guideline

Choose a development set and test set to reflect data you expect to get in the future and consider important to do well.

Guideline

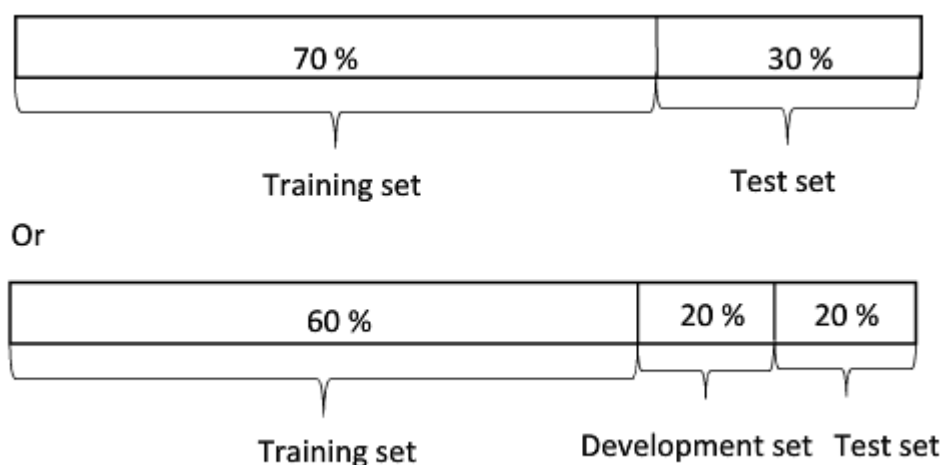
Choose a dev set and test set to reflect data you expect to get in the future and consider important to do well on.



训练、开发、测试集选择设置的一些规则和意见：

- 训练、开发、测试集的设置会对产品带来非常大的影响；
- 在选择**开发集**和**测试集**时要使二者来自同一分布，且从所有数据中随机选取；
- 所选择的开发集和测试集中的数据，要与未来想要或者能够得到的数据类似，即模型数据和未来数据要具有相似性；
- 设置的测试集只要足够大，使其能够在过拟合的系统中给出高方差的结果就可以，也许10000左右的数目足够；
- 设置开发集只要足够使其能够检测不同算法、不同模型之间的优劣差异就可以，百万大数据中1%1%的大小就足够；

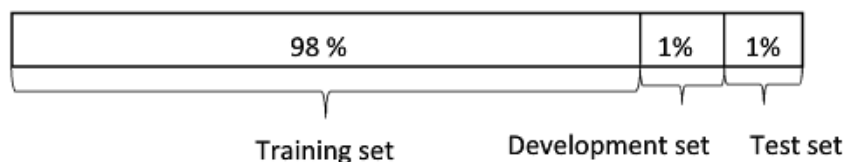
传统的划分方法有三七分，622分。



大数据时代的划分方法，只要拿出2%的数据平分给开发集和测试集就可以了。

Modern era – Big data

Now, because a large amount of data is available, we don't have to compromised as much and can use a greater portion to train the model.



1.6 什么时候该改变开发和测试集的指标

例子：判断是不是不猫

一个猫分类器给爱猫人士识别出猫的图片，算法的评估指标是分类错误率。

Algorithm	Classification error [%]
A	3%
B	5%

看起来似乎算法A比B更好，但是使用算法A会给把色情图识别为猫。

算法B，有5%的误差，但是识别出的图片不会有色情图片。从一个公司，一个用户的角度考虑，算法B更佳。

所以评估指标，开发集，测试集应该需要更改。

错误分类的误差指标应该如下

$$Error : \frac{1}{m_{dev}} \sum_{i=1}^{m_{dev}} \mathcal{L}\{\mathcal{Y}^{(i)} \neq y^{(i)}\}$$

问题是评估指标把色情图片和非色情图片识别为通道灯重要，所以我们应该给图片设置不同的权重。给色情图片设置大的权重。

$$w^{(i)} = \begin{cases} 1 & \text{if } x^{(i)} \text{ is non-pornographic} \\ 10 & \text{if } x^{(i)} \text{ is pornographic} \end{cases}$$

The function becomes:

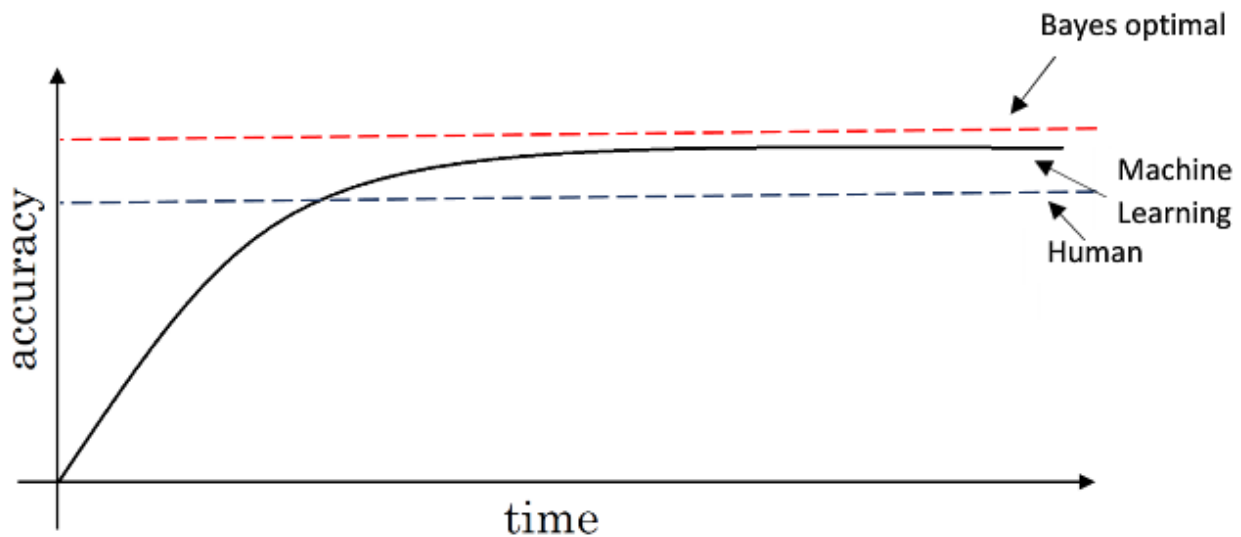
$$Error : \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m_{dev}} w^{(i)} \mathcal{L}\{\mathcal{Y}^{(i)} \neq y^{(i)}\}$$

指导方针

1. 定义正确的评估指标来更好的给分类器的好坏进行排序；
2. 优化评估指标。

1.8 把人类的表现作为参考

因为在很多功能，人类的表现堪称完美。人类能达到的水平和贝叶斯误差相差不远。



为什么和人类的表现比较？

Why compare to human-level performance

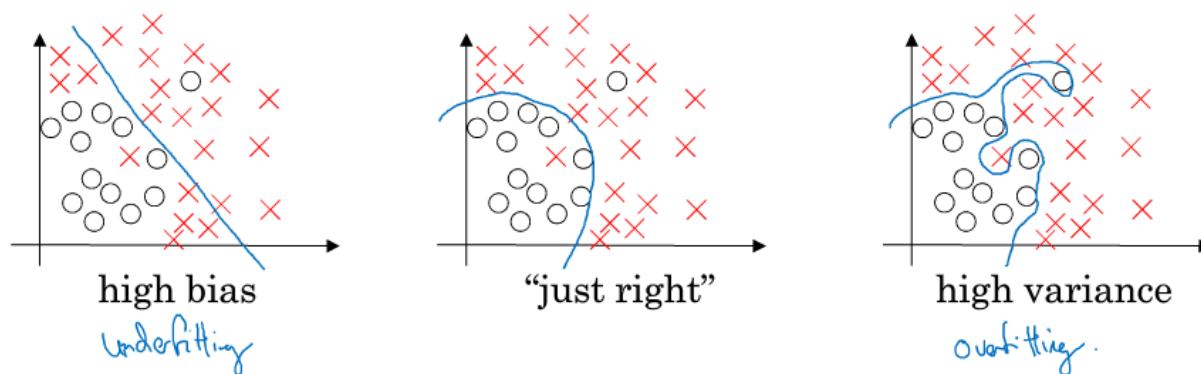
Humans are quite good at a lot of tasks. So long as ML is worse than humans, you can:

- - Get labeled data from humans. (x, y)
- - Gain insight from manual error analysis:
Why did a person get this right?
- - Better analysis of bias/variance.

- 可以获得数据的标签
- 可以更好地分析偏差和方差

1.9 可避免误差

Bias and Variance



Bias and Variance

Cat classification

Human-level $\approx 0\%$

Training set error:

Dev set error:

high variance

high bias

high bias
high variance

low bias
low variance



通过人类的表现，可以知道训练集的表现好还是不好？

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

算法A，和人类的表现差距7%，说明算法欠拟合，为了解决这个问题，我们使用减少偏差技术，比如加大训练的神经网络，训练神经网络更长时间。

算法B，训练集表现的不错，和人类的表现差距只有0.5%，开发集和训练集相差2%，所以为了解决这个问题，我们用减少方差技术，比如，正则化，或者增加训练集的大小。

1.10 理解人的表现

人类水平的误差是一个贝叶斯误差。

例子：医学影像识别。

在这个例子，贝叶斯误差被定义为小于等于0.5%。

	Classification error (%)
Typical human	3.0
Typical doctor	1.0
Experienced doctor	0.7
Team of experienced doctors	0.5

误差分析

	Classification error (%)		
	Scenario A	Scenario B	Scenario C
Human (proxy for Bayes error)	1	1	0.5
	0.7	0.7	
	0.5	0.5	
Training error	5	1	0.7
Development error	6	5	0.8

- 如果人类误差和训练误差的差距大于训练误差和开发误差的差距，则应该专注于减少偏差。
- 如果人类误差和训练误差的差距小于训练误差和开发误差的差距，则应该专注于减少方差。
- 人类水平误差是贝叶斯误差。

总结：

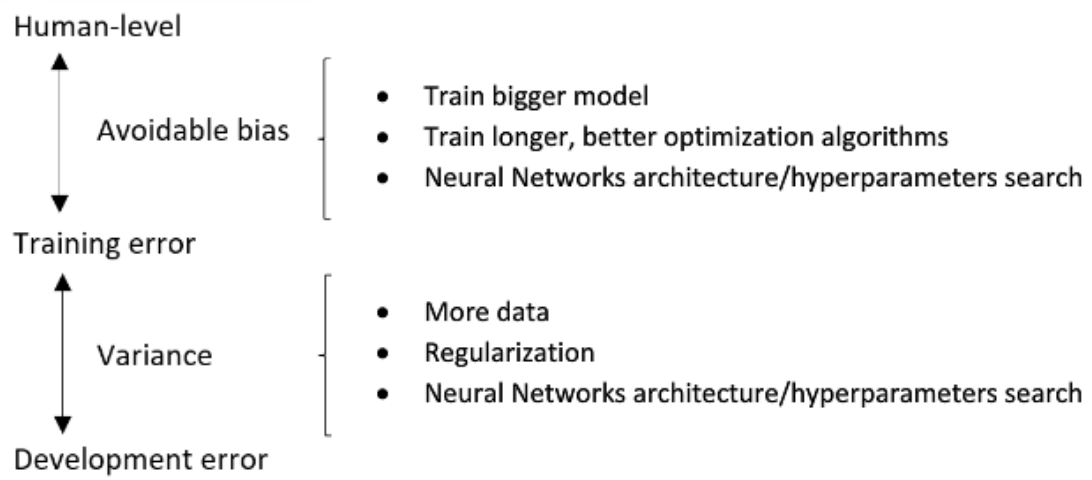
对人类水平误差有一个大概的估计，可以让我们去估计贝叶斯误差，这样可以让我们更快的做出决定：**减少偏差**还是**减少方差**。

而这个决策技巧通常都很有效果，直到系统的性能开始超越人类，那么我们对贝叶斯误差的估计就不再准确了，再从减少偏差和减少方差方面提升系统性能就会比较困难了。

1.11 超过人类的表现

1.12 改善模型表现

Summary



减少可避免误差

- 训练更大的模型
- 训练时间更长，训练更好的优化算法（Momentum，RMSPROP，Adam）
- 寻找更好的网络结构（RNN，CNN），寻找更好的超参数。

减少方差

- 收集更多的数据
- 正则化，（L2,dropout,数据增强）
- 寻找更好的网络结构（RNN，CNN），寻找更好的超参数。