

第三篇第一周

1、为什么是ML策略

如何构建你的机器学习项目，也就是ML策略，如何快速高效的优化你的机器学习系统，那么什么是机器学习策略呢？

例子：

Motivating example



云课堂



假设你正在调试你的猫分类器

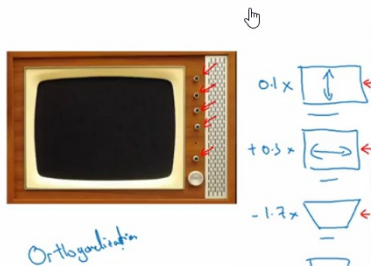
Let's say you are working on your cat classifier.

经过一段时间的调整，你的系统达到了90%的准确率，但是对你的应用程序来说不够好，你可能有很多想法去改善你的系统，比如采集更多的数据，收集更多姿势的猫咪，多样化的反例集，或者再用梯度下降算法训练久一点，或者尝试完全不同的优化算法，比如Adam算法，或者尝试规模更大或者更小的神经网络，或者尝试dropout或者L2正则化，或者修改网络的架构。。但是你可能花费很多时间做了错误的事情，就是说效果并不好，选择好的策略，就可以让深度学习系统更快的投入使用。

2、正交化

直观上是这样的：

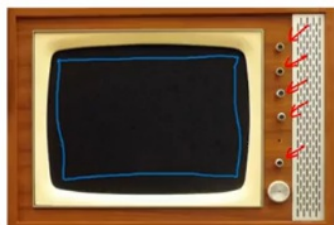
TV tuning example



云课堂

设计这样的旋钮 使得每个旋钮都只调整一个性质
had designed the knobs so that each knob kind of
does only one thing.

Andrew Ng



Car



→ Steering

→ Acceleration
Braking

Orthogonalization

分开独立的速度控制 要难得多
and a separate, distinct set of controls for controlling
the speed.

Andrew Ng

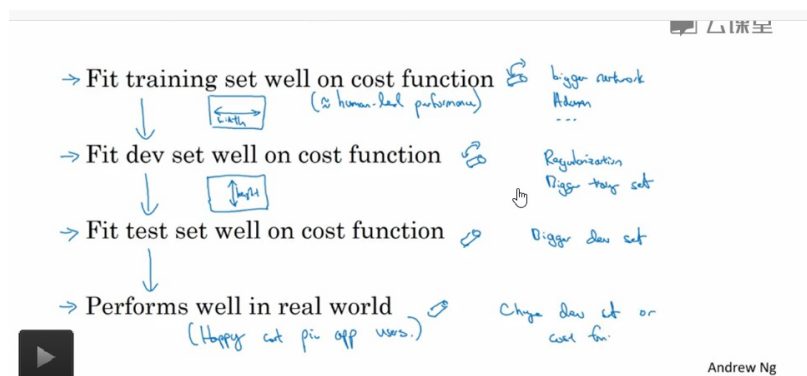
所以，正交化的概念是指：你可以想出一个维度，这个维度你想做的是控制转向角，还有另外一个维度来控制你的速

度，那么你就只需要一个按钮尽量只控制转向角，另外一个按钮，在这个开车的例子里，其实就是油门和刹车，控制你的速度，但是如果你有一个旋钮将两者结合起来，比如说这样一个控制装置同时影响你的旋转角和速度，同时改变了两个性质，那么就很难让你的车子以想要的速度和转角前进，然而正交化之后，正交意味着互成90度，设置出正交化的控制装置，最理想的情况就是和你实际想控制的性质一样，这样你调整参数就容易很多，可以单独控制转角，还有油门和刹车，令车子以你想要的方式运动。

要弄好一个监督学习系统，你通常需要调整你的系统的旋钮，确保四件事情：

- 1、至少系统在训练集上得到的结果不错，所以训练集上的表现必须通过某种评估，达到能接收的程度。对于某些应用，这可能意味着达到人类水平的表现，
- 2、在开发集上有好的表现。
- 3、在测试集上也有好的表现。
- 4、系统在测试集上，在实际使用中令人满意。

如果你的算法在成本函数上不能很好的拟合训练集，所以你用用来调试的旋钮，你可能可以训练更大的网络，或者切换更好的优化算法，比如Adam优化算法。相比之下，如果发现算法对开发集的拟合很差，那么应该有独立的一组按钮，你可以用正则化的按钮调节，尝试让系统满足第二个条件，也可以增大训练集。不满足第三个条件的话，可以增大开发集，因为出现了过拟合了。不满足第四个条件的话，你需要改变开发集或者成本函数，因为如果根据某个成本函数，系统在测试集上做的很好，但是他无法反映你的算法在现实世界中的表现，这意味着要么你的开发集设置不正确，要么你的成本函数测量的指标不对：

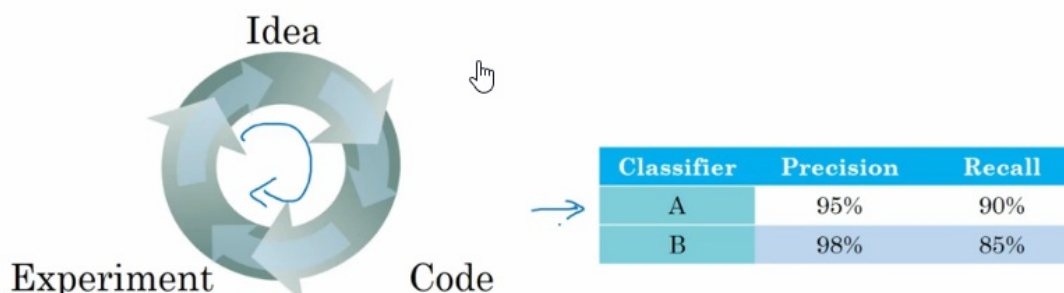


训练神经网络的时候，一般不要早期停止，早期停止有点难以分析，因为这个按钮同时影响你对训练集的拟合，早期停止，对训练集的拟合就不太好，但是他同时改善开发集的表现，所以这个按钮没那么正交化，因为它同时影响两件事情，就像一个按钮同时影响电视画面的长度和宽度，不要说这样就不要用，你想用还是可以的，你用其他正交化手段的话，会简单不少。

1.3、单一数字评估指标

例子：

Using a single number evaluation metric

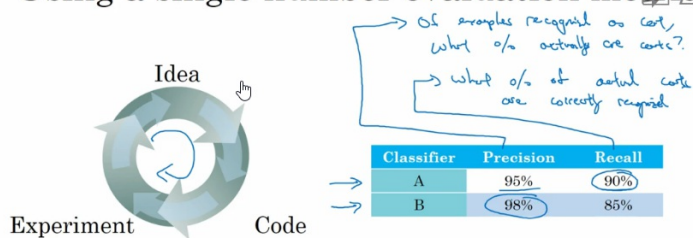


比如说对于你的猫分类器 之前你搭建了某个分类器A
So let's say for your cat classifier, you had previously built some classifier A.

通过改变超参数，还有改变训练集等手段，你现在训练处一个新的分类器B，所以评估你的分类器的一个合理方式是，观

察他的查准率和查全率，使用查准率或者查全率作为评估指标的时候，有个问题，如果分类器A在查全率上表现更好，分类器B在查准率上表现更好：

Using a single number evaluation metric



分类器B在查准率上表现更好

the classifier B does better on precision,

你就无法判断哪个分类器更好。如果你尝试了很多不同的想法，很多不同的超参数，你希望能够快速试验不仅仅是两个分类器，或者10几个分类器，选出最好的那个，如果有两个评估指标，就很难去快速二选一或者十选一，所以不建议同时使用这两个指标，找到一种新的评估指标，能够结合上述两个指标，结合两种的标准方法是所谓的F1分数，你可以认为它是查准率P和查全率R的平均值：

$$F_1 \text{ score} = \text{Herge of } P \text{ and } R. \\ \left(\frac{2}{\frac{1}{P} + \frac{1}{R}} \right) \text{ "Harmonic mean"}$$

在这个例子中：

Diagram illustrating the relationship between Idea, Experiment, and Code. A circular arrow connects them. A table compares Classifier A and B on Precision, Recall, and F1 Score. Handwritten notes explain F1 Score.

Classifier	Precision	Recall	F1 Score
A	95%	90%	92.4%
B	98%	85%	91.0%

Handwritten notes: "what % of actual cats are correctly recognized" (Recall), "F1 score = 'Average' of P and R. (mean)"

你可以马上看出 分类器A的F1分数更高

you can then see right away that classifier A has a

假设F1是结合查准率和查全率的合理方式，你可以快速的选出分类器A，淘汰分类器B，但实数评估指标，你的迭代速度肯定很快，它可以加速改进你的机器学习算法的迭代过程，

4、满足和优化指标

要把你顾及到的所有指标，组合成单实数评估指标，有时并不容易：

Another cat classification example

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

但除了准确度之外 我们还需要考虑运行时间

but let's say that in addition to accuracy you also care about the running time.

需要多少时间来分类一张图片，将准确率和运行时间组合成一个评估指标，所以成本，比如说，总成本是准确率减去0.5乘以运行时间，这种组合可能太刻意。

你还可以做其他事情，就是你可能选择一个分类器，能够最大限度的提高准确率，但必须满足运行时间的要求：

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

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

达到之后你不在乎这指标有多好
and beyond that you don't really care,

也就是说一个是优化目标，一个是满足目标，但满足目标实现的情况下，最大限度的优化目标，所以这里B的效果最好，因为在运行时间小于100ms的分类器中，B的准确率最好。所以更一般的说，如果你要考虑N个评估指标，有时候选择其中一种指标作为优化指标是合理的，所以你可以尽量优化那个指标，剩下的N-1个指标都是满足指标，意味着他们只要达到一定的阈值，就可以了。

5、训练、开发、测试集的划分

验证集合测试集的划分例子：

Cat classification dev/test sets

云课堂

Regions:

- US
- UK
- Other Europe
- South America
- India
- China
- Other Asia
- Australia



Dev

Test

事实证明这个想法非常糟糕 因为这个例子中

It turns out, this is a very bad idea because in this example,

你的开发集和测试集来自不同的分布，所以这样很糟糕，我们尽量要使训练集、验证集、测试集来自同一分布。如果开发集合测试集来自不同的分布，就像你设了一个目标，让你的团队花了很长的时间去逼近靶心，结果几个月工作之后你会发现，你说：“等等，测试的时候，你把靶心移开”。团队的成员会说：

Cat classification dev/test sets

云课堂

Regions:

- US
- UK
- Other Europe
- South America
- India
- China
- Other Asia
- Australia



dev set
+
Metric

Idea

为什么你让我们花那么多个月的时间去逼近那个靶心 然后突然间

why did you make us spend months optimizing for a different bull's eye when suddenly,

Experiment

Code

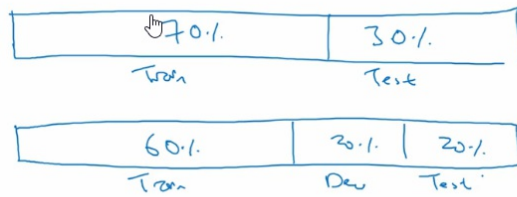
Andrew Ng

你可以把靶心移动到不同的位置，（也就是期末画的重点跟考试内容不一样，这就白复习了，很可怕，可能挂科！！），所以为了避免这种情况，将所有数据重新洗牌，放入开发集和测试集，使得开发集合测试集来自同一分布。

6、训练、开发、测试集的大小

在机器学习中，可以用7比3的比例划分训练集和测试集，如果你要划分训练集，开发集，测试集，可以采用6:2:2的比例。

Old way of splitting data



在机器学习的早期 这样分是相当合理的

In earlier eras of machine learning, this was pretty reasonable,

在深度学习中，我们可能使用很大的数据集。比如你有100万个训练例子，这样分可能很合理，98%作为训练集，百分之1作为开发集，百分之1作为测试集。

7、什么时候改变开发集和测试集的指标

Cat dataset examples



Metric: classification error

Algorithm A: 3% error

Algorithm B: 5% error

所以算法A似乎做得更好

so it seems like Algorithm A is doing better.

A算法似乎做的更好，但是A算法由于某种原因，把很多色情图片分类成猫了，所以部署A算法，用户可能看到更多的猫，因为它的准确率比较高，但是它也会给用户推荐一些色情图片，这是公司不能接受的。相比之下，B算法的误差是5%，这种分类器虽然得到较少的图像，但是它不会推送色情图片，所以从公司，用户的角度来看，算法B实际上是更好的算法，因为它不会让任何色情图片通过，在这个算法中，发生的事情就是，算法A在评估指标上做的更好，它的误差达到3%，但是实际上是更糟糕的算法。用户更倾向于使用算法B，所以，当算法无法正确衡量算法之间的优劣时，在这种情况下，原来的算法错误的预测算法A是更好的算法，这就发出了信号，你应该改变评估指标了，或者要改变开发集和测试集，解决办法之一可以在错误率上给色情图片加上更大的权重，这样如果有色情图片，那么错误了乘以这个权重就会变得很大：

Cat dataset examples

Metric + Dev : Prefer A
You/Users : Prefer B.



Metric: classification error

Algorithm A: 3% error → pornographic

✓ Algorithm B: 5% error

Error: $\frac{1}{n} \sum_{i=1}^n \omega_i |y_{pred}^{(i)} - y^{(i)}|$
你把色情图片分类成猫这一错误的惩罚权重加大10倍
10 times bigger weights to classify pornographic images correctly.

这个方法不好用，因为你必须在开发集和测试集里面把色情图片标记出来，这样才能这个加权函数，但是粗略的结论是，如果你的评估指标，无法正确评估好算法的排名，那么就需要花时间定义一个新的评估指标，改变评估指标的意义

在于，已知两个分类器，哪一个更加适合你的应用，也就是说如果你对旧的指标不满，那么你就不要去使用不满的错误的指标，而应该去尝试使用新的指标，能够更加符合你的偏好，定义出实际更加适合的算法。



Another example

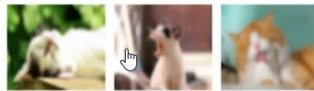
Algorithm A: 3% error

✓ Algorithm B: 5% error ←

Dev/test



User images



这是另一个指标和开发集测试集出问题的例子

So this would be another example of your metric and dev test sets falling down.

Andrew Ng

问题在于，你做评估用的是很漂亮的高分辨率的开发集和测试集，图片取景很专业，但是你的用户，真正关心的是，他们上传的图片能不能被正确识别，有些图片可能没那么专业，有点模糊，取景很业余，所以方针是，如果你在指标上表现很好，在当前开发集和测试集分布上表现的很好，但是你的实际应用程序，你真正关注的地方表现不好，那么就需要修改指标，或者你的开发测试集：



Another example

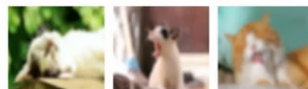
Algorithm A: 3% error

✓ Algorithm B: 5% error ←

→ Dev/test



→ User images



If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and dev test set.

那么就应该改变你的开发测试集

then that's a good time to change your dev test set



Another example

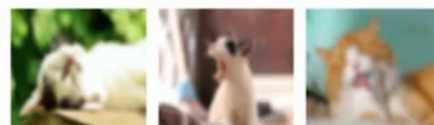
Algorithm A: 3% error

✓ Algorithm B: 5% error ←

→ Dev/test



→ User images



If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and dev test set.

让你的数据更能反映你实际需要处理好的数据

so that your data better reflects the type of data you actually need to do well on.

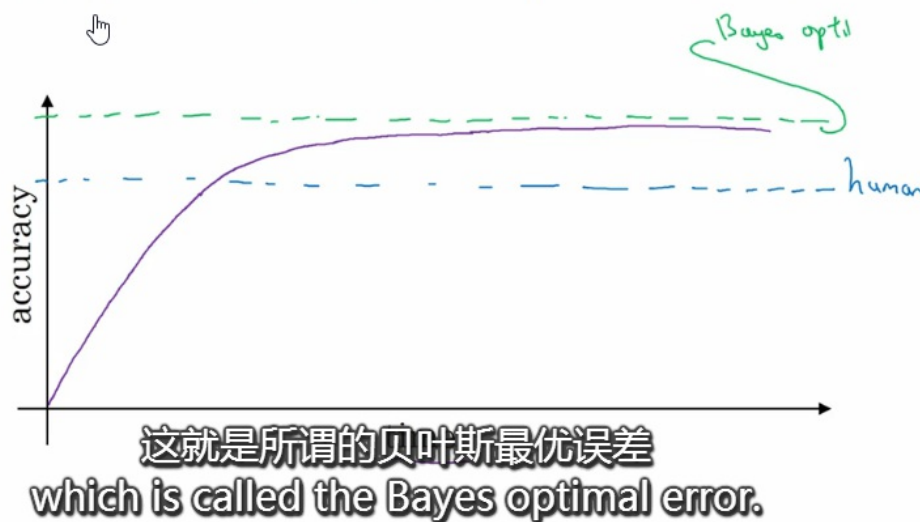
Andrew Ng



有一个评估指标和开发集让你可以更快做出决策，哪一个算法是更好的，这真的可以加快你和你团队的迭代速度，所以，即使你无法定义出一个很完美的评估指标和开发集，你直接快速设立出来，然后使用它们来驱动你的团队的迭代速度，如果在这之后，你发现选的不好，你有更好的想法，那么马上可以改，最好不要在没有指标和开发集时跑太久，因为这样会减慢团队的速度。

8、为什么是人的表现

Comparing to human-level performance 云课堂



数据越来越大，模型越来越复杂时，accuracy无法越过的阈值，就是所谓的贝叶斯最优误差，理论上是最可能达到的最优误差，

9、可避免偏差

Training error 8%
Dev error 10%

我们经常使用猫分类器来做例子
We have used Cat classification a lot and giving a picture,

Humans 1%
Training error 8%
Dev error 10%

比如人类具有近乎完美的准确度 所以人类水平的错误是百分之一

let's say humans have near-perfect accuracy so the human level error is one percent.

这种情况下，训练误差是8%，测试误差是10%，那么你可能想在训练集上得到更好的结果，但实际上，训练误差跟人类相比相差巨大，说明你的算法拟合程度不好，这种情况下，你应该把重点放在减小偏差的任务上，你可以训练更大的神经网络，或者训练更久一点的梯度下降，接下来，假设：

Humans	1%	7.5%
Training error	8%	8%
Dev error	10%	10%

Focus on bias

假设人类水平错误实际上是7.5%。
let's say that human level error is actually 7.5%.

在这个例子中，偏差非常接近人类的水平，这是你可以专注于另外一个分量，那就是减小学习算法的方差，你可以尝试正则化，让验证误差更接近训练误差。

人类水平在图像识别上非常接近贝叶斯误差，所以当训练误差离贝叶斯误差太远的话，也就是偏差太大，这是要专注于调节偏差，如果训练误差接近贝叶斯误差，但是验证误差（方差）太大的话，就要专注于调节方差。

10、理解人的表现

人类水平误差，用来估计贝叶斯误差，那就是理论最低的误差，任何函数，不管是现在还是未来，能够达到的最低值：

Medical image classification example:



我们先记住这点 然后看看医学图像分类例子
So bearing that in mind, let's look at a medical image classification example.

Medical image classification example:

Suppose:

- (a) Typical human 3 % error
- (b) Typical doctor 1 % error



普通的医生 也许是普通的放射科医生 能达到1%的误差
A typical doctor, maybe a typical radiologist doctor, achieves 1% error.

Medical image classification example:



Suppose:

(a) Typical human 3 % error

(b) Typical doctor 1 % error

还有一队经验丰富的医生, 就是说如果你有一个经验丰富的医生团队

And a team of experienced doctors, that is if you get a team of experienced doctors



那么你应该如何定义人类水平的误差, 人类水平误差的定义, 就是如果你想要替代或者估计贝叶斯误差, 那么一队经验丰富的医生讨论和辩证之后, 可以达到0.5%的误差, 我们知道贝叶斯误差小于等于0.5%, 因为有些系统, 这些医生团队可以达到0.5%的误差, 也许有经验更丰富的医生做的更好, 但是我们知道最优秀的误差不能高于0.5%:

Suppose:

(a) Typical human 3 % error

(b) Typical doctor 1 % error

(c) Experienced doctor 0.7 % error

那么在这个背景下 我就可以用0.5%估计贝叶斯误差
So what I would do in this setting is use 0.5% as our estimate for Bayes error. $\leq 0.5\%$



What is "human-level" error?

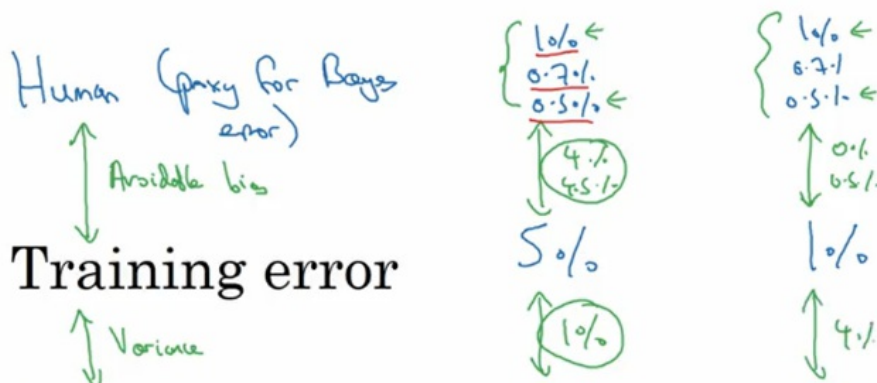


所以这里讲人类误差定义为0.5%

例子:

Error analysis example

云课堂



偏差和方差的问题, 如上图。

