Datawhale零基础入门CV赛事
街景字符编码识别
训练调参与模型集成

分享人：安晟

目录

contents

赛事资源

天池新人赛由天池与Datawhale联合发起，并提供学习内容和组织学习：

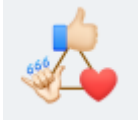
- Datawhale是一个专注于数据科学与AI领域的开源组织；
- CV直播PPT 可关注Datawhale公众号，回复关键词 **CV直播** 下载；
- 同时可以加入Datawhale数据竞赛交流群，一起组队参赛，交流学习；

个人介绍

安晟

- ✓ Datawhale成员，开源贡献者
- ✓ 专注CV方向的算法工程师
- ✓ 邮箱：anshengmath@163.com

Datawhale CV小组开源项目：动手学CV-Pytorch版
https://github.com/datawhalechina/dive-into-cv-pytorch



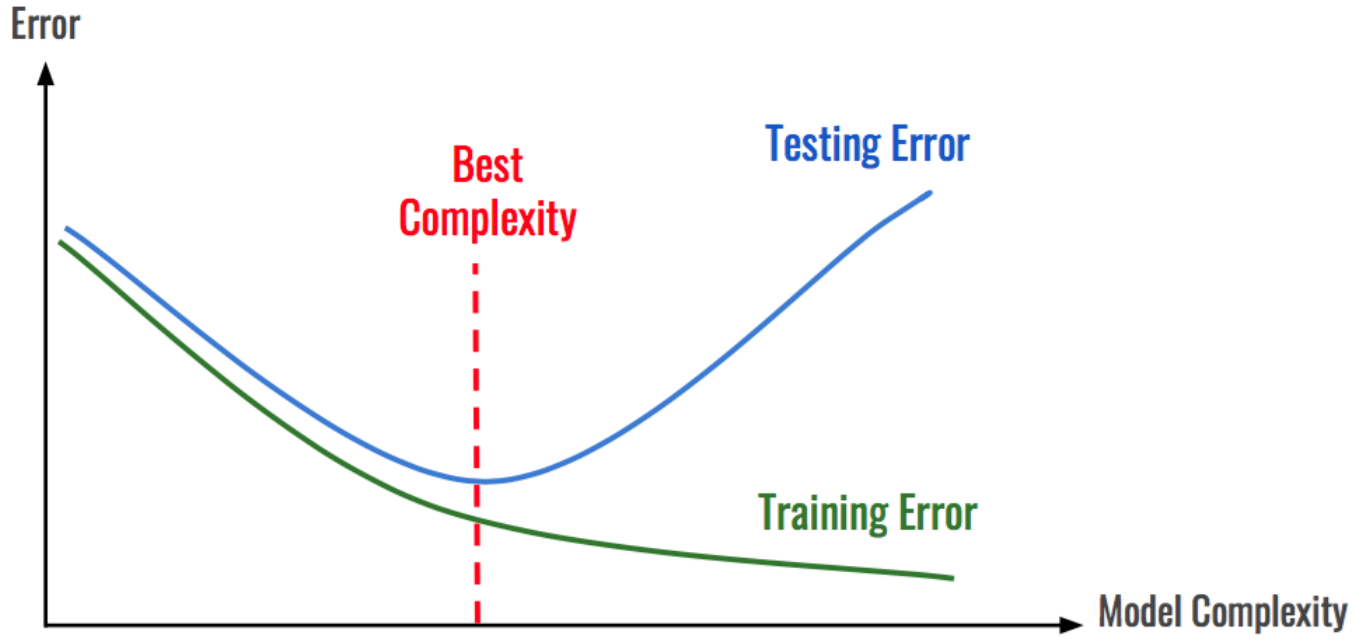
Part 1 模型训练与验证

为什么要设置验证集

Datawhale

用于调整模型超参数，有效判断模型状态，防止过拟合

CNN模型的拟合能力很强，甚至会强行记住训练样本的一些无关紧要的细节来达到loss的不断下降，因此一味追求训练集loss的下降，可能导致模型在测试集的泛化效果较差。

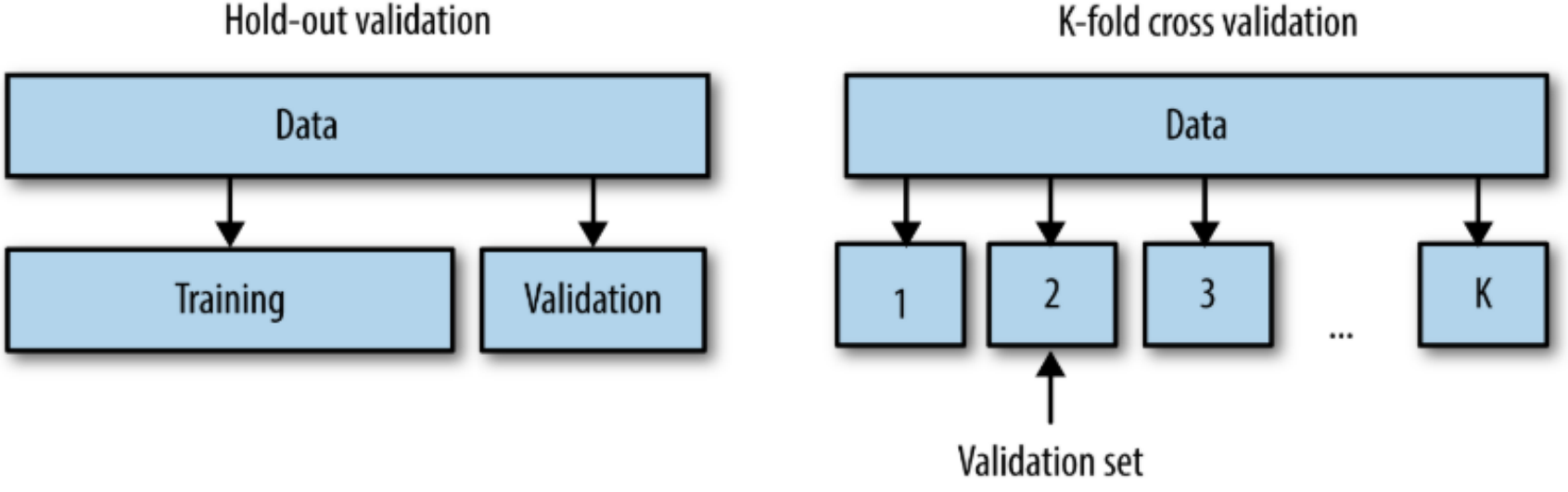


验证集划分方法

Datawhale

两种常见的验证集划分方法：

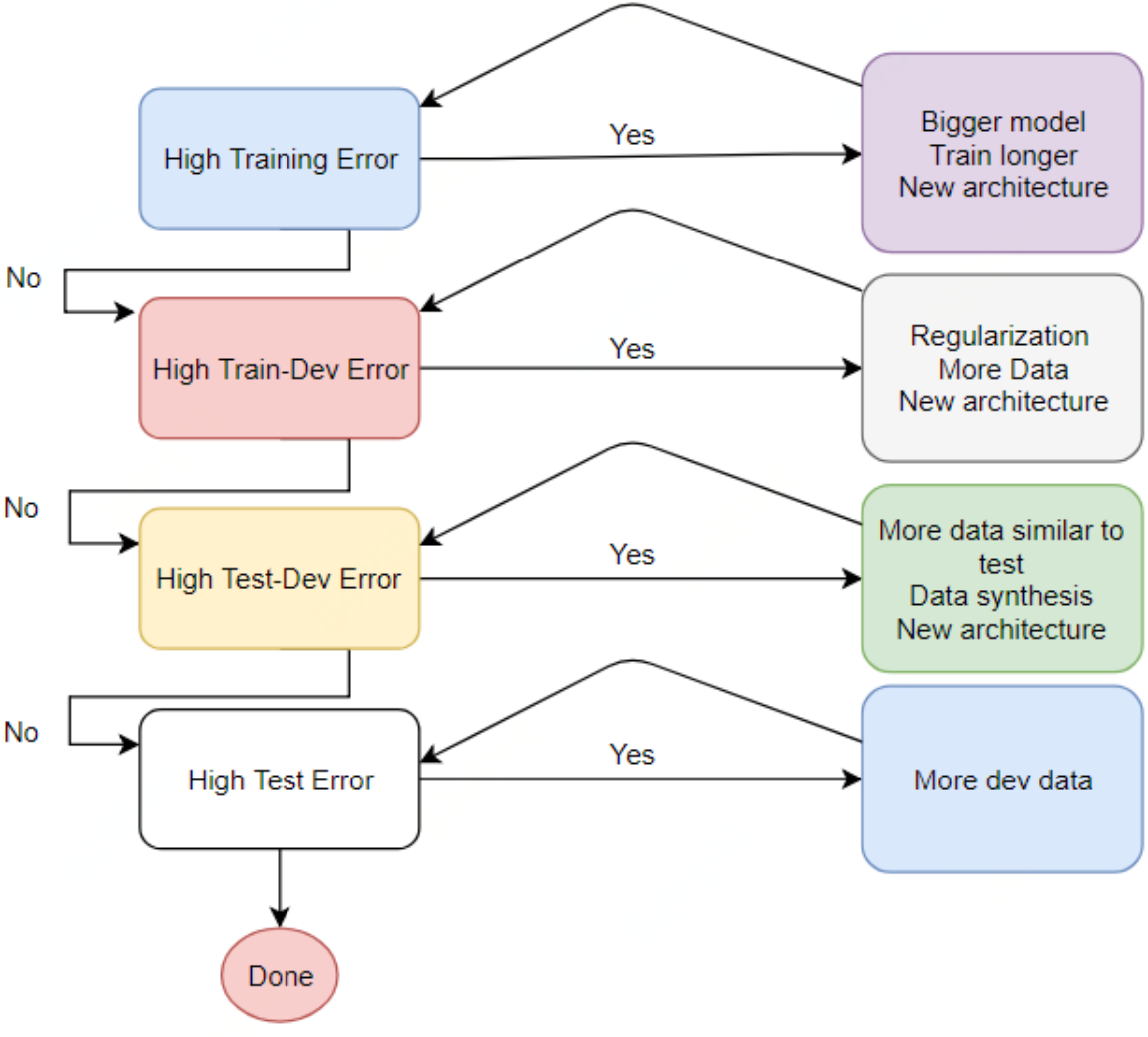
- 留出法（Hold-Out）
- K折交叉验证（K-fold Cross Validation）



Part 2 调参流程

调参指导框架

Datawhale



观察数据

在你开始训练，甚至是编写任何代码之前，你需要大量的观察数据

观察什么？

- 了解问题的背景，数据的样式，标注信息的格式等等。
- 观察数据的分布，类别是否存在偏差？
- 是否有脏数据？是否有大量重复数据？

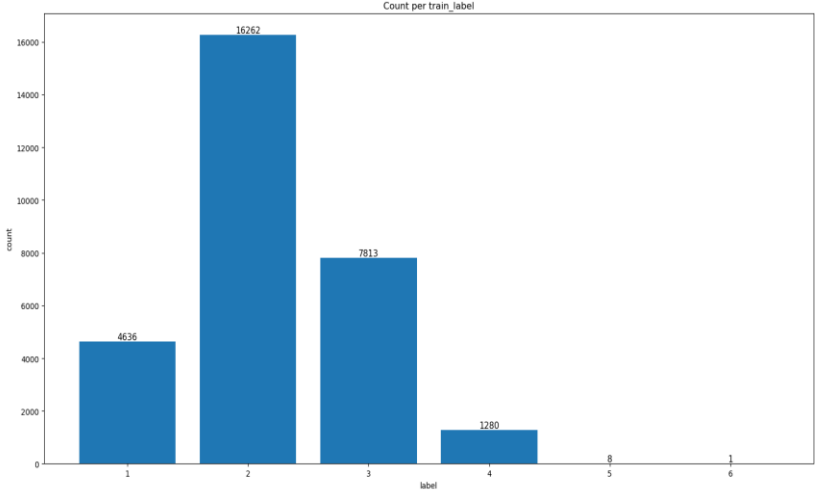
这个过程实际上会让我们对问题有很好的理解，并且设计出合理的训练框架来求解（虽然baseline已经帮我们跳过了这一步骤）。

这样说可能有些空洞，我们来结合赛题看下：

观察数据

Datawhale

- 不同字符数量图片的占比，是否存在偏差？
- 是否有脏数据？是否有大量重复数据？
- 局部信息足够吗？是否需要上下文信息？
- 位置信息重要吗？
- 数据适合什么样的数据增强？什么样的数据增强不适用？



搭建初始框架

基于对数据的清晰认识，就可以搭建baseline训练框架了。
建议你遵循以下原则：

- 固定随机种子
  固定所有可能的随机种子。包括python，numpy，torch，tensorflow的随机种子，这对你复现遇到的bug并查找原因非常重要。
- Don‘t be a hero
  baseline尽可能简单。无论你是否是新手，你的脑海中都可能已经浮现了很多关于如何优化模型的想法了。
  作为baseline，不要做任何花里胡哨的操作×
  使用尽可能简单，最有把握的官方实现的网络作为baseline，只使用最基本的数据增强。
- 将输入进网络的数据进行可视化.
  这可以很直观的看出预处理环境是否有bug
- 设置合理的评价指标，并打印或绘制足够的信息来监控训练状态

让baseline走上正轨

有了最基本的baseline训练&验证框架，就可以开始实验了。
这里有几个基本参数的设置经验，可以让baseline快速走上正轨，

- 初始学习率选择

  一个合适的初始学习率，应该能够让训练初期每个batch的loss非常快速的下降。通常，你应该首先尝试，1e-2，1e-3这样的数值。
- 优化方法选择

  Adam is all you need。

  Adam具有很强的适应性，能够自适应的调整学习率的大小来完成快速收敛，这是其内在原理决定的。使用Adam优化器通常会让模型收敛更快，并且让初始学习率的可选范围变广。

  尽管对于CNN的训练来说，一个精细调整的基于SGD的训练通常在效果上可以略微超过Adam，但绝大多数情况我都会使用Adam。
- 学习率阶段性下降策略
- 使用预训练模型

调参原则

我知道大家最不缺的就是奇思妙想，各种超参数，各种idea都想要去尝试。即便是刚刚入门的同学，也能把欠拟合-过拟合的概念以及他们可能的解决方案说的头头是道，所以这里我不再赘述。

唯一强调一点，调参的过程中，你要时刻保持思路的清晰，知道自己在做什么，为什么要这么做。你可以参考前面我们介绍的框架来指导自己的实验。

通常我在工作或者比赛中，会问自己这样几个问题：

- 是否理清了当前问题的主要矛盾？
  只针对当前最核心的问题进行解决。
  是否每个实验都能带来正向提升。
- 当前进行的实验是否能带来确定性的结论？
  单一变量原则

■ 不出意外的翻车

我第一次运行baseline，不负众望的跑出了0.33的高分，可以说翻车翻的很彻底。

相信很多刚入门的小伙伴都经历过0.3-0.4分的绝望，下面我就结合实际赛题带大家了解下，如何按照前面介绍的方法论，寻找正确的调参知识。

日期: 2020-05-22 01:15:06 **排名:** 无

score: 0.6875

日期: 2020-05-22 00:26:34 **排名:** 无

score: 0.3344

实战调参介绍:
https://github.com/datawhalechina/dive-into-cv-pytorch/tree/master/beginner/chapter02_image_classification_introduction/2.5_SVHN_in_action

Part 3 模型集成

模型集成：榨出最后一点成绩

当我们已经对单模型进行了充分的调参达到了不错的效果之后，我们还可以使用集成学习来尽可能的“压榨”出最后一点成绩的提升。

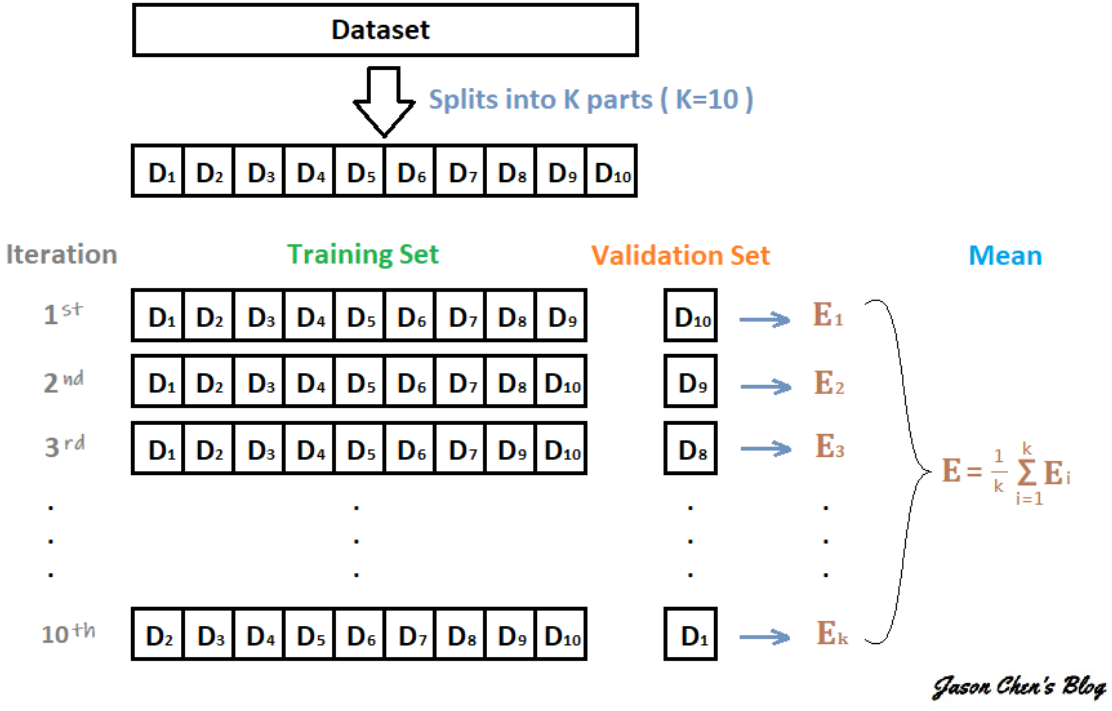
实际上，是非常不推荐大家过多通过模型集成来上分的，因为这在实际应用中毫无意义。大多数正式比赛也都会对模型集成，以及使用的模型大小，推理速度进行限制。由于是入门练习赛，这次比赛没有对模型集成进行要求。

机器学习中的集成学习可以在一定程度上提高预测精度，常见的集成学习方法有Stacking、Bagging和Boosting，同时这些集成学习方法在CV场景下同样试用，只是受算力的制约会更加明显。

通过交叉验证集成多模型

Datawhale

下面假设构建了10折交叉验证，训练得到10个CNN模型。



那么得到的10个CNN模型可以使用如下方式进行集成：

- 对预测的结果的概率值进行平均，然后解码为具体字符；
- 对预测的字符进行投票，得到最终字符。

TTA(Test Time Augmentation)

Datawhale

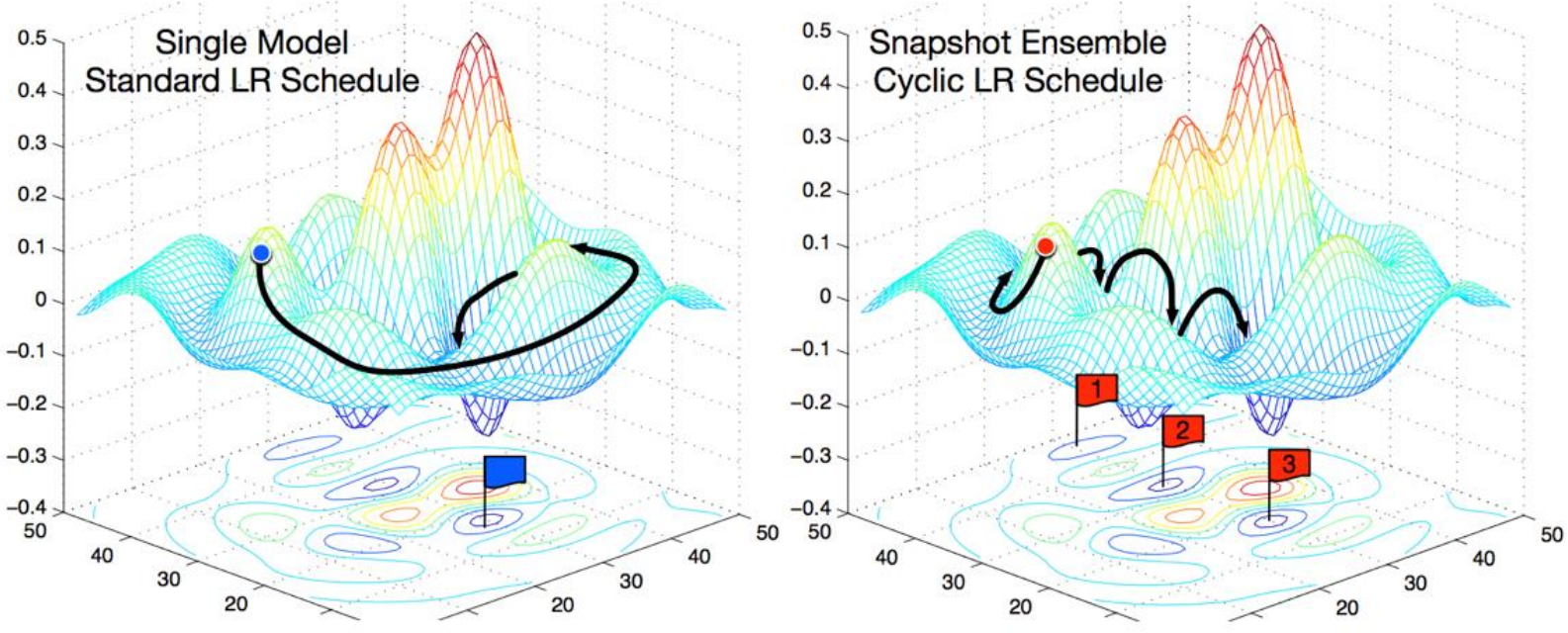
我们不光可以将同一份数据喂入多个模型，也可以将多份相似数据喂入同一个模型，这也是一种常用的集成学习技巧，称为测试集数据扩增（Test Time Augmentation，简称TTA）。

也就是说，数据扩增不仅可以在训练时候用，而且可以同样在预测时候进行数据扩增，对同一个样本预测多次，然后对多次结果进行平均。

Snapshot

Datawhale

在论文Snapshot Ensembles中，作者提出使用cyclical learning rate进行训练模型，并保存精度比较好的一些checkopint，最后将多个checkpoint进行模型集成



Part 4 Q&A

Datawhale

一个专注于AI领域的开源组织