



AutoAugment: Learning Augmentation Strategies from Data

Ekin D. Cubuk *, Barret Zoph*, Dandelion Mané, Vijay Vasudevan, Quoc V. Le
Google Brain

汇报人：刘可心

汇报日期：2024.11.26

目录

- 现存问题
- 研究目标
- 实现方法
- 实验验证

现存问题

- **传统数据增强具有局限性**，现有的数据增强方法（例如水平翻转、平移等）尽管有效，但大多基于手工设计，缺乏系统性，无法全面捕获数据中的潜在不变性。
- **数据增强方法难以迁移**，许多数据增强方法对特定数据集有效（如 CIFAR-10 的翻转），但在其他数据集（如 MNIST）上的效果有限，难以泛化。
- 目前大部分研究主要关注改进网络架构，对更好、更自动化的数据增强方法研究较少，而**自动学习适合目标数据集的数据增强策略是一个尚未解决的重要问题**。

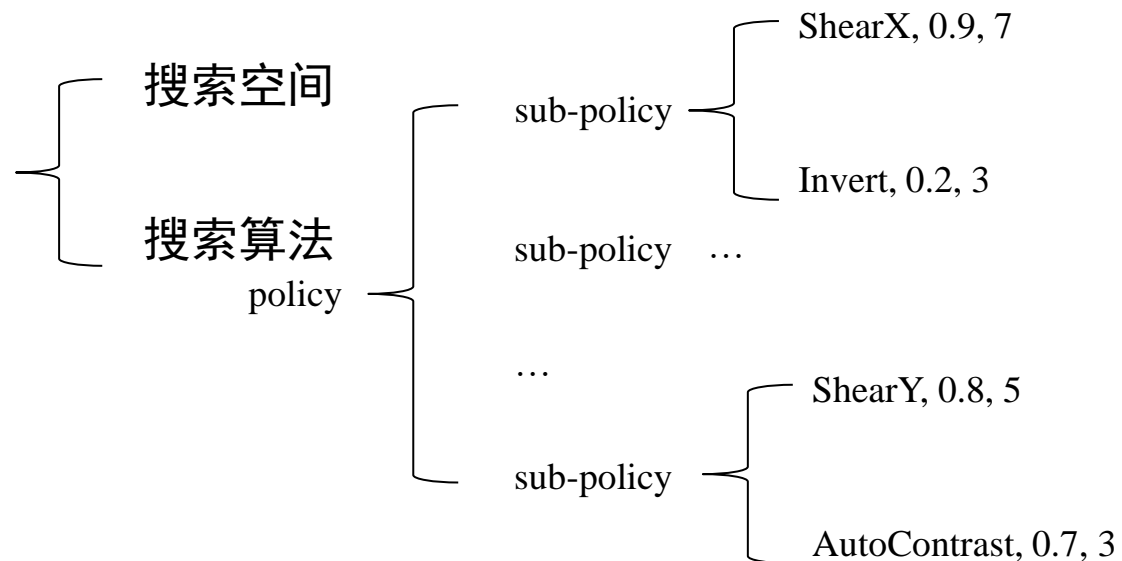
研究内容

- **提出AutoAugment**，自动找到目标数据集上有效的数据增强策略，优化增强操作的选择、顺序、概率以及强度，使神经网络训练后的验证精度达到最佳。
- **两种应用场景：**
 - 直接应用：在目标数据集上直接学习最优增强策略，以提高模型的性能。
 - 策略迁移：将一个任务上学习到的增强策略迁移到不同的数据集或模型中，减少直接应用的计算开销。
- 通过实验进行性能验证。

实现方法

- 问题定义

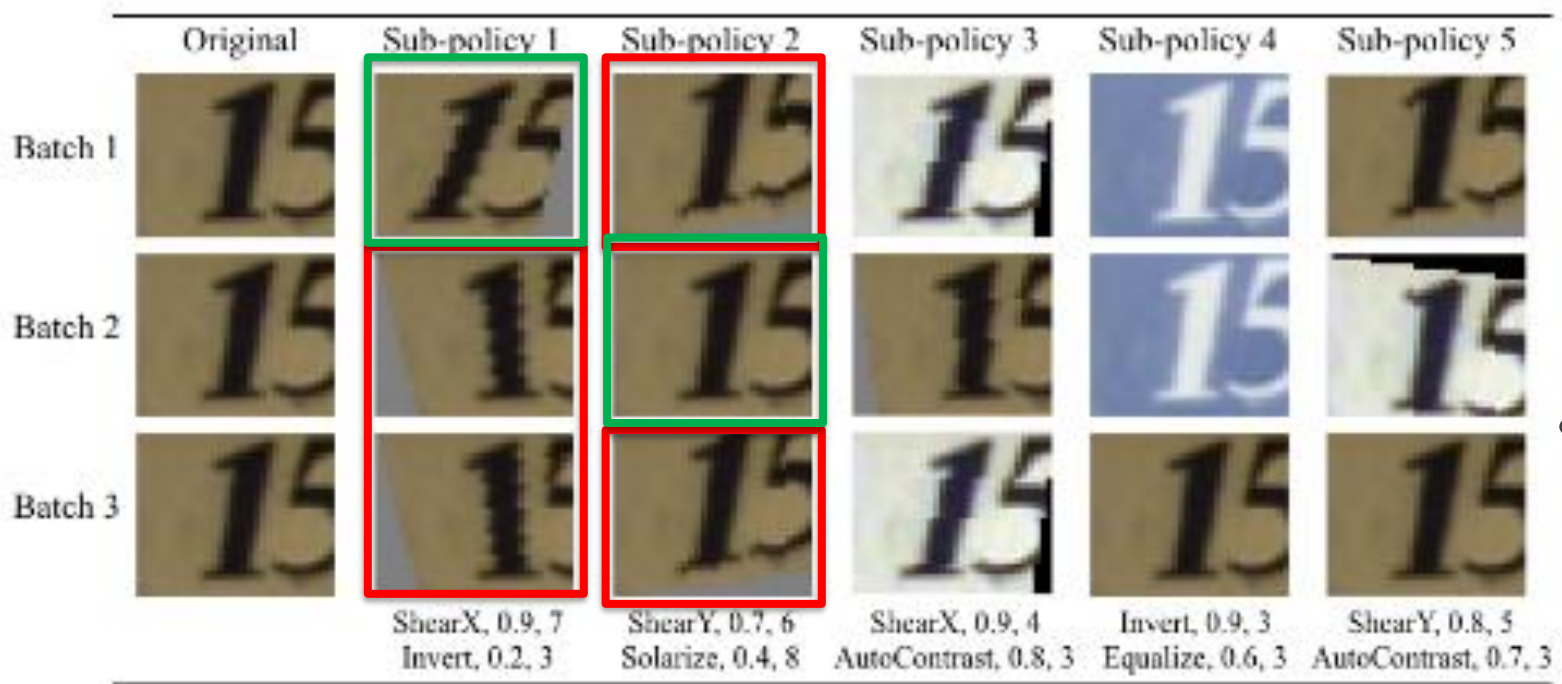
- 自动搜索最优的数据增强策略问题 → 离散搜索问题



- 搜索空间

- 搜索空间包含 16 种图像处理操作。
 - 每种操作的强度和概率被离散化为固定值域（例如，概率为 11 个离散值，强度为 10 个离散值）。
 - 一个策略由五个子策略组成，子策略由两个顺序应用的图像操作组成（关联两个超参数：操作的应用概率及强度）
 - 巨大的搜索空间：一个子策略 $(16 \times 10 \times 11)^2$ 一个策略 $(16 \times 10 \times 11)^{2 \times 5} = (16 \times 10 \times 11)^{10}$

随机性的保证



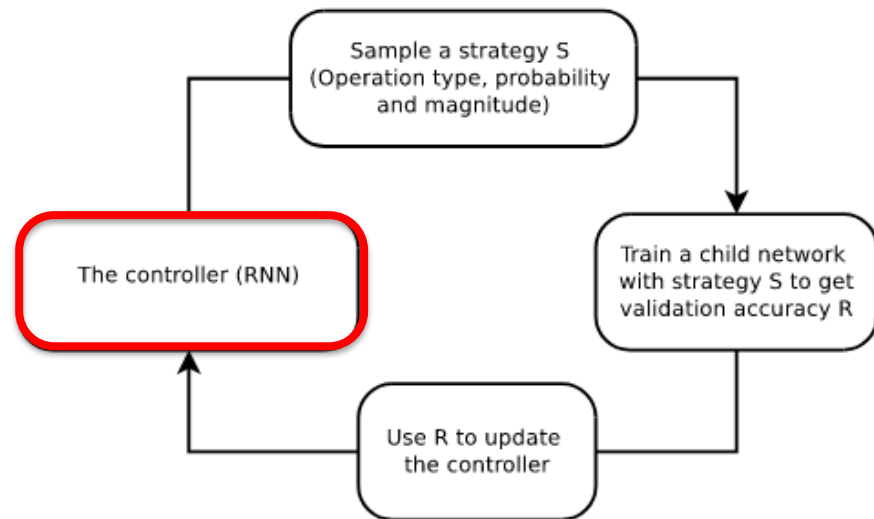
即使是对同一张原始图像，应用相同的策略，由于操作是否调用的概率性，以及 mini-batch 中子策略选择的随机性，生成的增强图像也可能完全不同。

- 子策略选择的随机性
 - 从5个子策略中随机选择一个生成变换后图像，用于训练。
 - 同一张图像在不同mini-batch中可以被应用不同的增强。
- 操作的随机性
 - 应用概率：操作可能在某个mini-batch中被应用，可能未被应用。
 - 固定强度：一旦操作被调用，则应用强度相同。

搜索算法

- 强化学习
- 两个组成部分 { 控制器 (RNN)
- 训练算法: 近端策略优化算法 (PPO)
- 控制器:
 - 使用单层的 LSTM
 - LSTM 每层包含 100 个隐藏单元
 - 输出: $2 \times 5B$ 次 Softmax 预测, B 表示 Softmax 的参数空间大小 (即增强操作、强度或概率的选项数)
 - 策略的联合概率: 所有 $10B$ 个 Softmax 输出概率的乘积

$P(\text{策略}) = \prod_{i=1}^{10B} p_i$ 衡量控制器生成该策略的信心



搜索算法

出于方便，我们选择使用 PPO 来训练控制器，尽管之前的工作已经表明其他方法（例如增强随机搜索和进化策略）可以表现得同样好甚至稍微好一些

• 训练算法

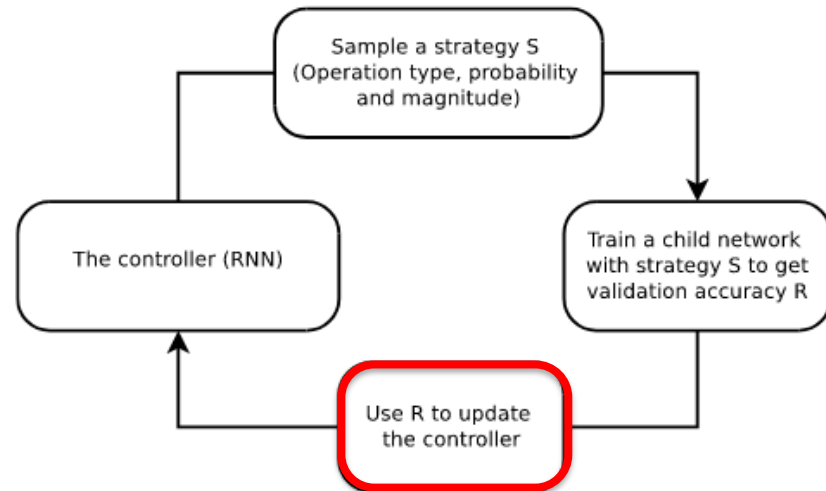
- Reward: 子模型在验证集上的性能（验证集上的准确率R）

- 强化学习优化：使用PPO（Proximal Policy Optimization）

- 损失函数：最大化策略的联合概率与奖励的乘积

$$J=E[R \cdot P(\text{策略})]$$

- 梯度更新：R用于缩放控制器的梯度，提升高质量策略的概率



PPO

- 在经典的策略梯度方法（Policy Gradient, PG）的基础上做了改进

- 目标函数裁剪： $\text{clip}(R(\theta), 1-\epsilon, 1+\epsilon)$

- 限制策略更新的幅度，防止过大的变化

- 提高样本效率

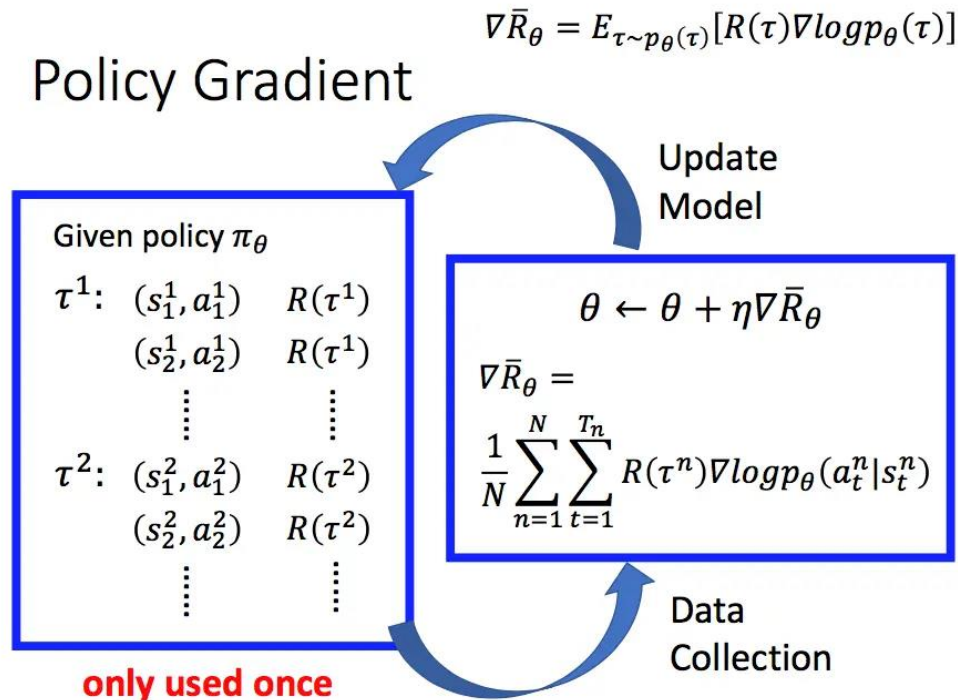
- 在经典 PG 方法中，采样的数据往往在一次更新后被废弃，PPO 支持在同一批数据上进行多次更新（多次优化目标函数），从而提升样本利用率，减少采样需求。

- 引入熵正则化

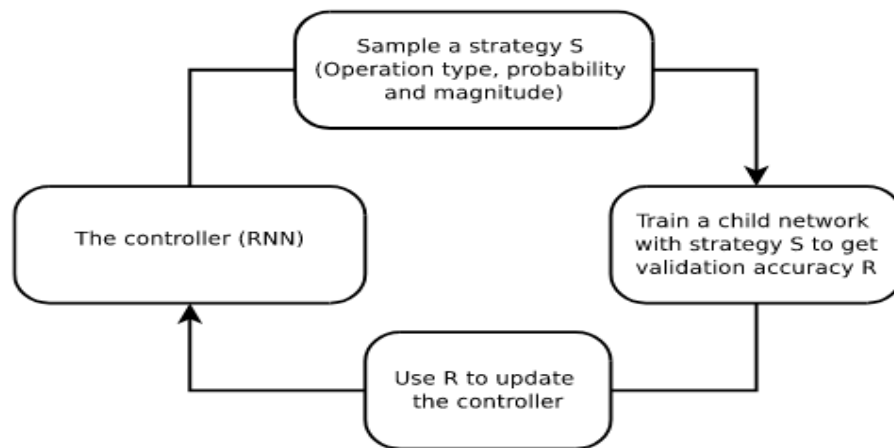
- 增加策略的熵，鼓励探索多样化的动作。

- 防止策略过早收敛到局部最优。

增加reward大的动作出现的概率，减小reward小的策略出现的概率



搜索过程



- 采样增强策略：
 - 控制器每次生成一个增强策略，该策略包括 5 个子策略（共 10 个操作）。每个策略由控制器 RNN 采样预测。
- 训练子模型：
 - 使用生成的增强策略对子模型进行训练。在验证集上评估子模型的性能，作为奖励信号返回给控制器。
- 更新控制器：
 - 使用奖励信号调整控制器的权重，提升生成优质策略的概率。
- 策略选优：
 - 在整个搜索过程中，控制器生成约 15,000 个策略。搜索结束后，选择验证集性能最好的 5 个策略，将其子策略整合为一个最终增强策略（包含 25 个子策略）。

策略迁移

- 迁移策略
 - 在源数据集（如 ImageNet）上，搜索出一组最优增强策略。将这些策略应用到目标数据集上，直接作为目标任务的增强方法，无需重新搜索。
- 迁移的适用场景
 - 目标数据集较小，重新搜索策略计算开销高。
 - 目标数据集与源数据集具有相似的特性（如分辨率、内容类型等）。

实验验证

Dataset	Model	Baseline	Cutout [12]	AutoAugment
CIFAR-10	Wide-ResNet-28-10 [67]	3.9	3.1	2.6±0.1
	Shake-Shake (26 2x32d) [17]	3.6	3.0	2.5±0.1
	Shake-Shake (26 2x96d) [17]	2.9	2.6	2.0±0.1
	Shake-Shake (26 2x112d) [17]	2.8	2.6	1.9±0.1
	AmoebaNet-B (6,128) [48]	3.0	2.1	1.8±0.1
	PyramidNet+ShakeDrop [65]	2.7	2.3	1.5 ± 0.1
Reduced CIFAR-10	Wide-ResNet-28-10 [67]	18.8	16.5	14.1±0.3
	Shake-Shake (26 2x96d) [17]	17.1	13.4	10.0 ± 0.2
CIFAR-100	Wide-ResNet-28-10 [67]	18.8	18.4	17.1±0.3
	Shake-Shake (26 2x96d) [17]	17.1	16.0	14.3±0.2
	PyramidNet+ShakeDrop [65]	14.0	12.2	10.7 ± 0.2
SVHN	Wide-ResNet-28-10 [67]	1.5	1.3	1.1
	Shake-Shake (26 2x96d) [17]	1.4	1.2	1.0
Reduced SVHN	Wide-ResNet-28-10 [67]	13.2	32.5	8.2
	Shake-Shake (26 2x96d) [17]	12.3	24.2	5.9

Test set error rates (%) on CIFAR-10, CIFAR-100, and SVHN datasets. Lower is better.

存在的问题

$$(16 * 10 * 11)^{10} \approx 10^{32}$$

- AutoAugment存在的一个问题是搜索空间巨大，这使得搜索只能在代理任务中进行：使用小的模型在ImageNet的一个小的子集（120类和6000图片）搜索。
- 谷歌在2019年又提出了一个更简单的数据增强策略：RandAugment。
- RandAugment相比AutoAugment的策略空间很小（ 10^2 vs 10^{32} ），所以它不需要采用代理任务，甚至直接采用简单的网格搜索。