

DARTS : 架构搜索的可微解决

DARTS: Differentiable Architecture Search

张娇昱

jiaoyu_zhang@zju.edu.cn

2019.03.23

01

什么是神经网络架构搜索

02

DARTS 架构与实现

03

Darts 总结与改进

04

答疑

1

神经网络架构搜索

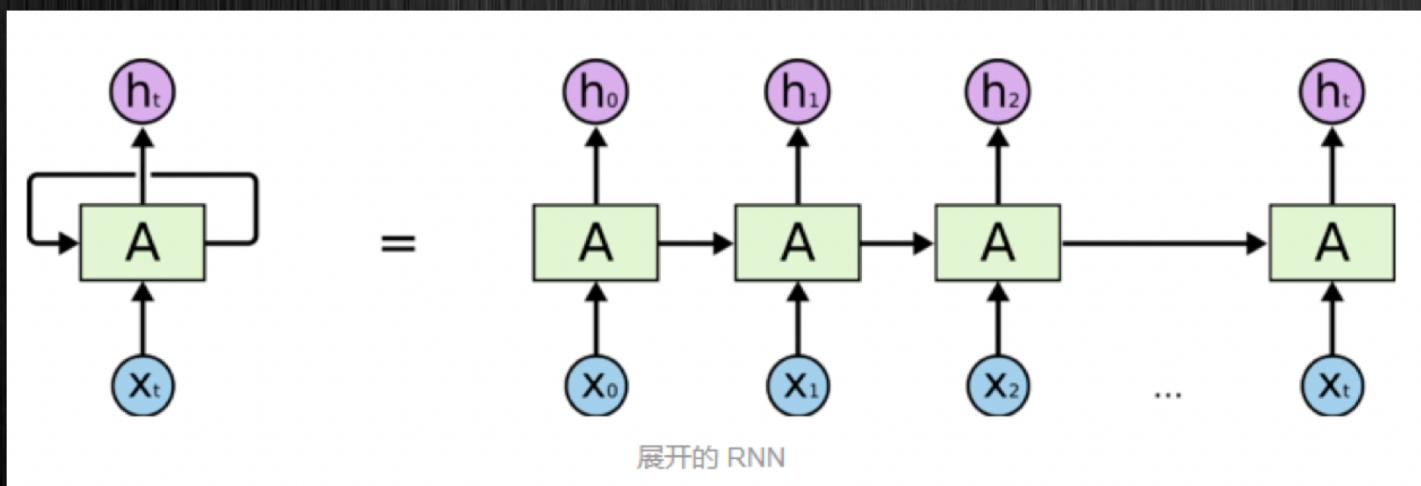
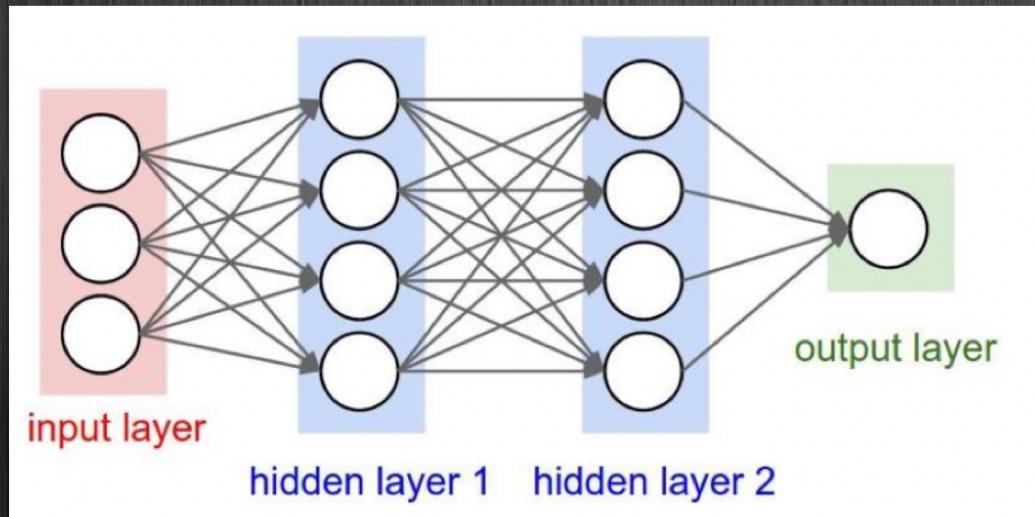
Neural Architecture Search

简介

主流方法

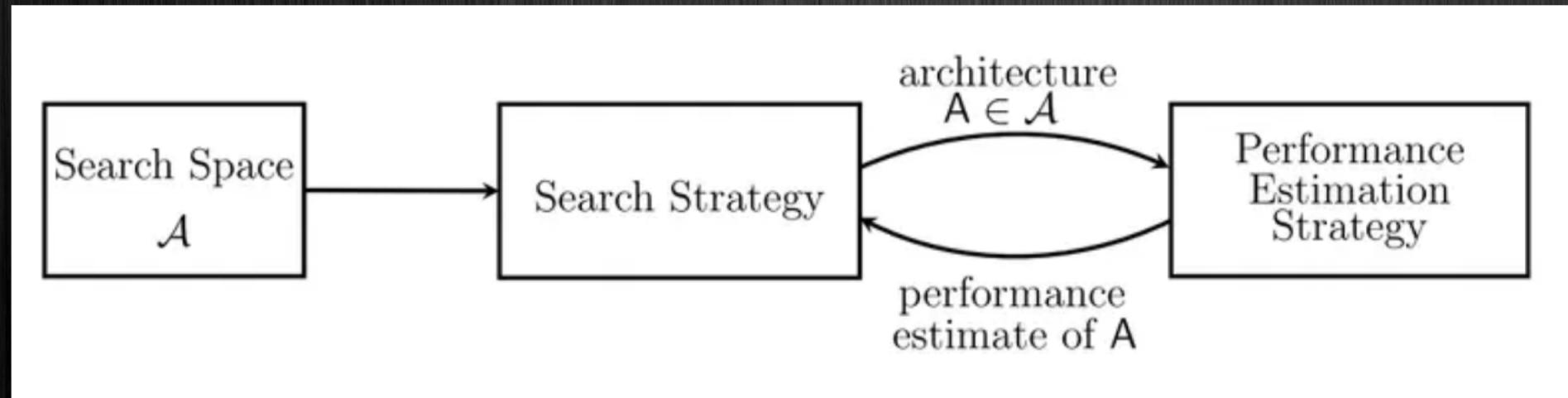


什么是神经网络架构搜索



什么是神经网络架构搜索

神经网络架构搜索：是属于AutoML中的一个分支，通常使用强化学习或进化算法来设计新的神经网络网络结构。比较经典的强化算法NAS和ENAS，进化算法AmoebaNet





研究意义

Search Space

搜索空间定义了优化问题的变量。深度学习模型的性能是由网络架构参数和对应的超参数来决定的。

Search Strategy

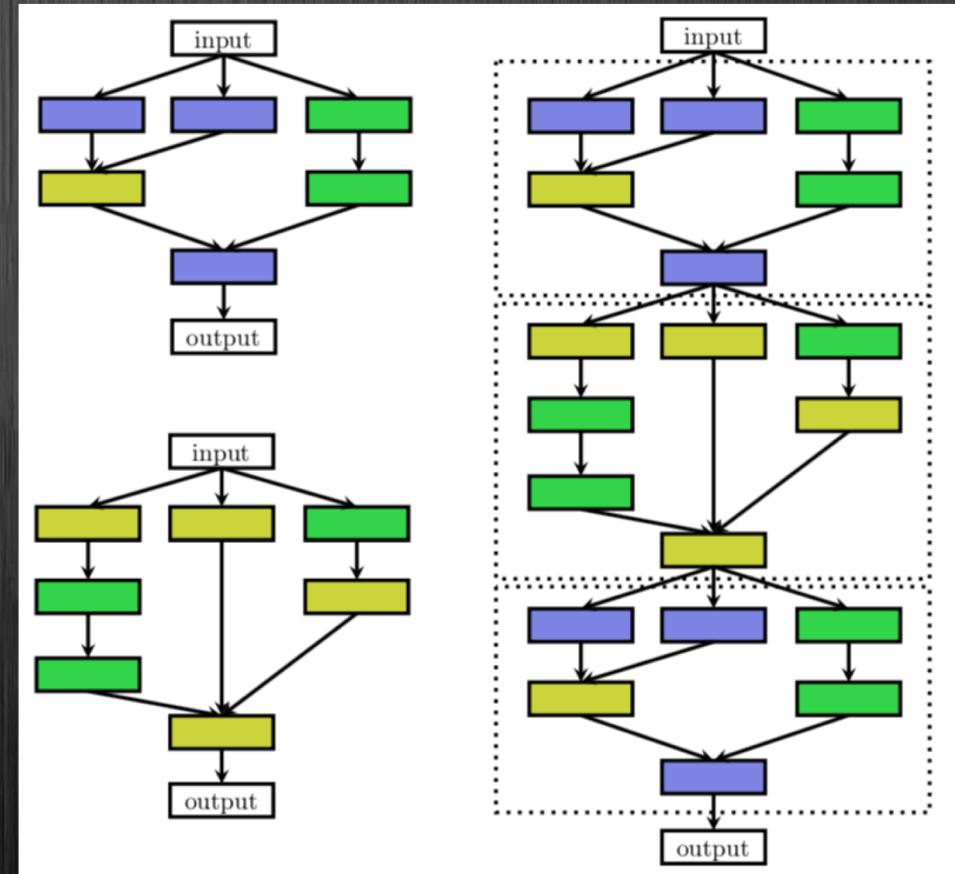
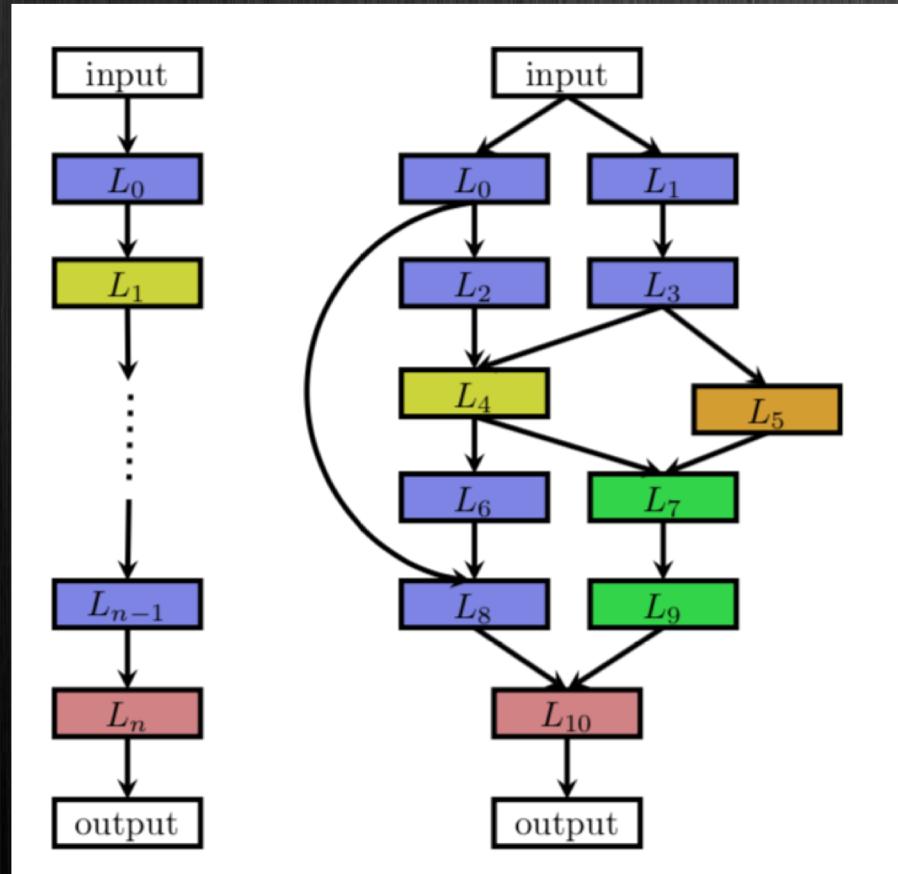
搜索策略：定义了使用怎样的算法可以快速、准确找到最优的网络结构参数配置。

Eval Strategy

用一些低保真的训练集来训练模型，借鉴于工程优化中的代理模型，参数级别的迁移。



神经网络架构搜索的三个要素



2

Darts

结构与实现

整体架构

优化策略

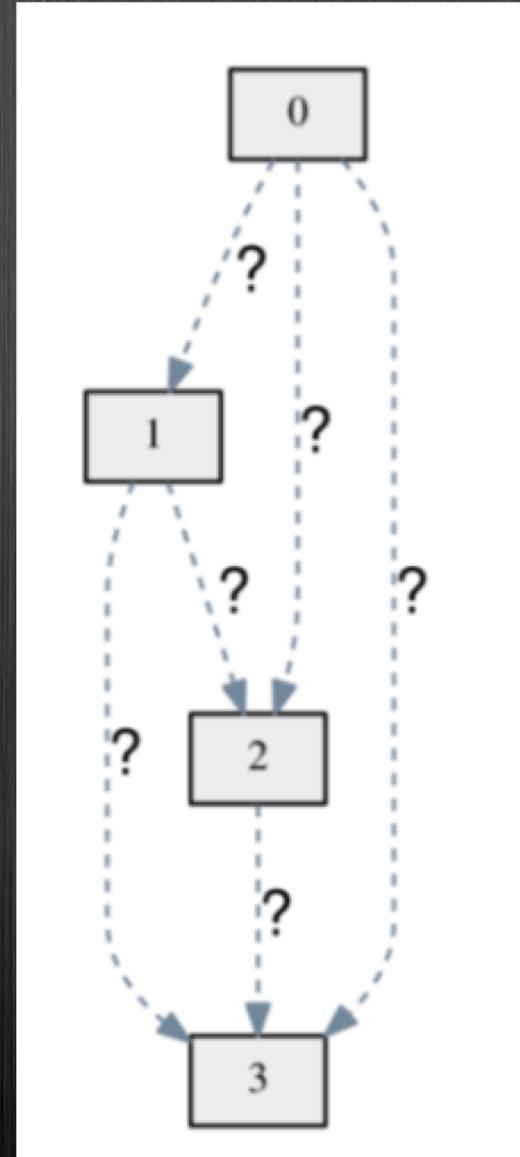
近似方法



Darts 搜索空间

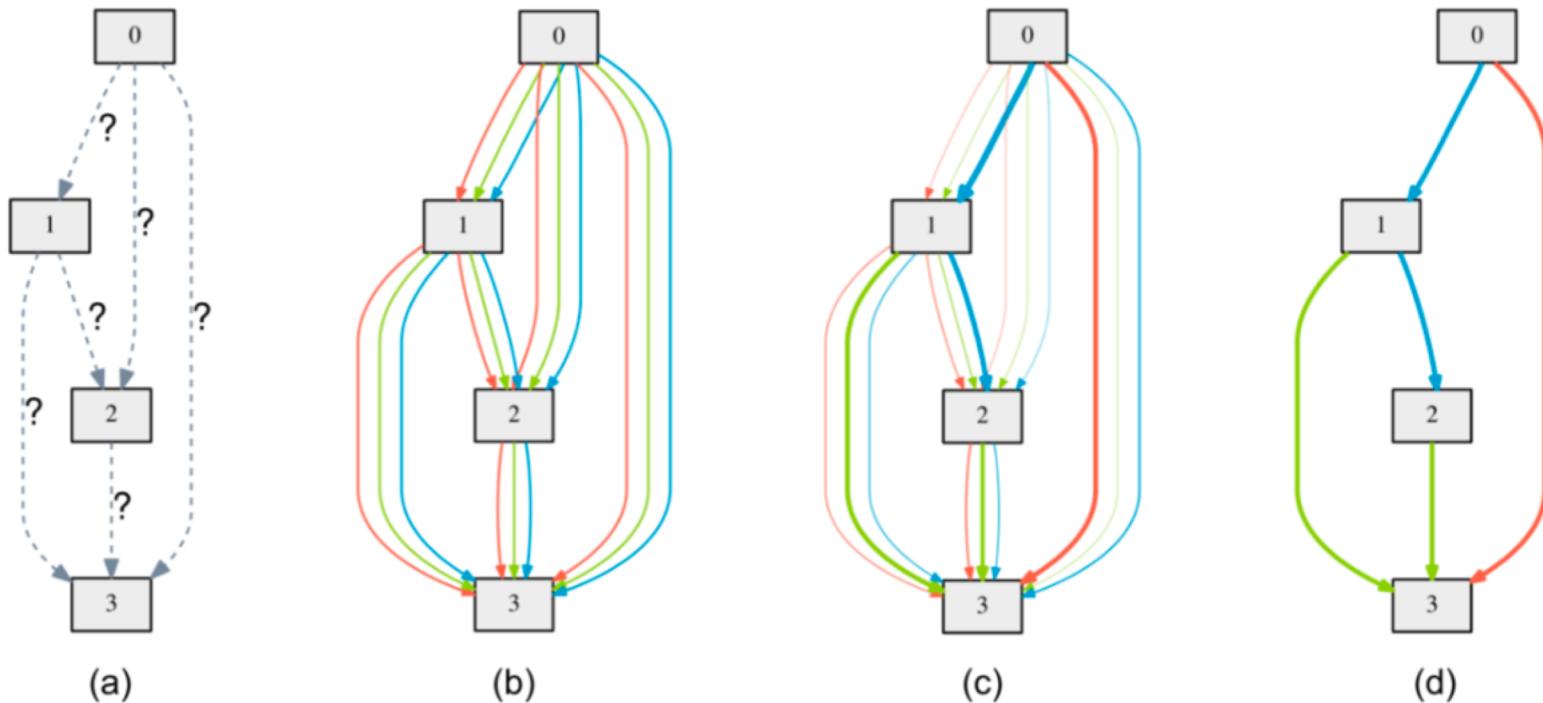
- 以一个cell（或者叫building block）作为一个搜索空间，它是由nodes和edges构成的有向无环图。

$$x^{(i)} = \sum_{j < i} o^{(i,j)}(x^{(j)})$$





Darts 搜索空间整体流程



1. 初始时，每条边即操作是未知的
2. 采用松弛策略，使每条边上都进行多种候选操作
3. 采用**bilevel optimization**来联合优化操作的混合概率和网络权重
4. 采用混合操作中概率最高的操作来替换混合操作，得到最终结构



Darts 优化策略

1. 将搜索空间可微化

darts通过引入softmax来做了一个简单的松弛方案，使得整个搜索空间变为可微的。

$$\text{Softmax : } S_i = \frac{e^i}{\sum_j e^j}$$

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

$$o^{(i,j)} = \operatorname{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$$



Darts 优化策略

2. 建模为双层优化问题

先定义两个loss： L_{train} 和 L_{val} 。很显然，我们的最终目标是找到一个 α 使得 L_{val} 最小，但是需要注意的是 L_{val} 不仅有关于 α ，还有关于 w 。

目标函数：

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ \text{s.t.} \quad & w^*(\alpha) = \operatorname{argmin}_w \mathcal{L}_{train}(w, \alpha) \end{aligned}$$

这就是典型的Bilevel Optimization Problem



Darts 优化策略

3. 近似迭代优化：梯度下降

darts中的近似迭代：通过 w 和 α 分别在权重和构架空间中的梯度下降步骤之间交替来完成优化。

Algorithm 1: DARTS – Differentiable Architecture Search

Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge (i, j)

while *not converged* **do**

1. Update weights w by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$
2. Update architecture α by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$

Replace $\bar{o}^{(i,j)}$ with $o^{(i,j)} = \text{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$ for each edge (i, j)



Darts 优化策略

$$\nabla_{\alpha} \mathcal{L}_{val}(w', \alpha) - \xi \nabla_{\alpha, w}^2 \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$$

在第k步的时候，首先采用架构 α_{k-1} ，通过 $\mathcal{L}_{train}(w_{k-1}, \alpha_{k-1})$ 获得现在的权重 w_k ，利用梯度更新 w_k ，再固定 w_k ，进行主问题的优化：

$$L_{val}(w_k - \xi \nabla \mathcal{L}_{train}(w_k, \alpha_{k-1}), \alpha_{k-1})$$

这里的 ξ 是网络权重 w_k 的学习率。

由上面的推断，我们可以得到构架 a 的梯度表示：

$$\nabla_{\alpha} L_{val}(w', \alpha) - \xi \nabla_{\alpha, w}^2 L_{train}(w, \alpha) \nabla_{w'} L_{val}(w', \alpha)$$



Darts 优化策略

$$\nabla_{\alpha} \mathcal{L}_{val}(w', \alpha) - \xi \nabla_{\alpha, w}^2 \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$$

可以注意到第二项中有一个二次梯度，这一项的计算是极为复杂的，所幸微分是可以有近似操作的。

$$\nabla_{\alpha, w}^2 \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \approx \frac{\nabla_{\alpha} \mathcal{L}_{train}(w^+, \alpha) - \nabla_{\alpha} \mathcal{L}_{train}(w^-, \alpha)}{2\epsilon}$$
$$w^+ = w + \epsilon \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \quad w^- = w - \epsilon \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$$

O(|a| |w|) → O(|a| + |w|)



Darts 优化策略

可以考虑一个特殊情况，学习率为0的时候，上式就变成了一个一阶近似。而 a 的梯度则完全取决于前一项。这样的话， a 和 w 就相对独立了，但是这种情况下虽然速度有了提高，效果却有明显的下降。

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10. Results marked with † were obtained by training the corresponding architectures using our setup.

Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	manual
Random + cutout	3.49	3.1	–	–
DARTS (first order) + cutout	2.94	2.9	1.5	gradient-based
DARTS (second order) + cutout	2.83 ± 0.06	3.4	4	gradient-based



Darts 优化策略

4. 还原为离散网络结构

得到最终参数 α ，要将它还原为离散的网络构架，我们进行两步操作：

1. 为每个节点选取最强的k个预处理，这个k的选取依据以下公式：

$$\max_{o \in \mathcal{O}, o \neq zero} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})}$$

2. 以argmax取代所有混合操作，成为最有可能的操作。



Darts 优化策略

为了和其他网络结构保持可对比性，darts的卷积单元 $k=2$ ，循环单元 $k=1$ 。

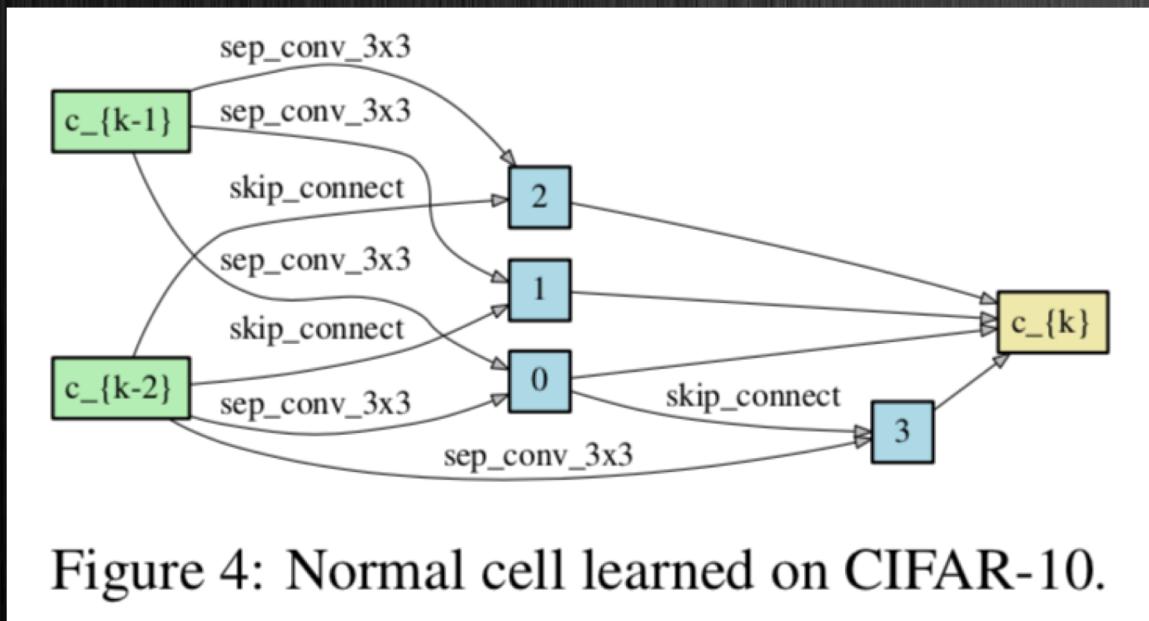


Figure 4: Normal cell learned on CIFAR-10.

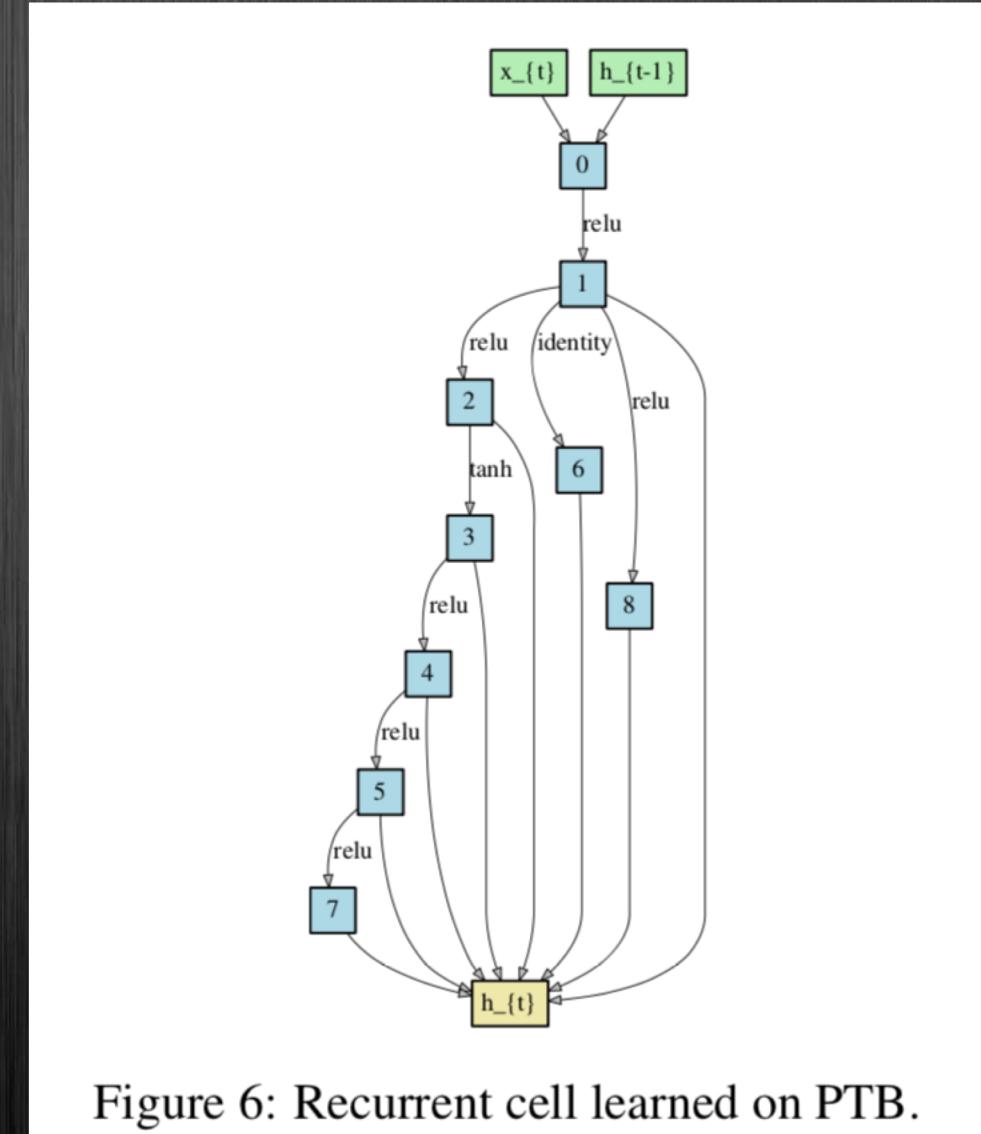


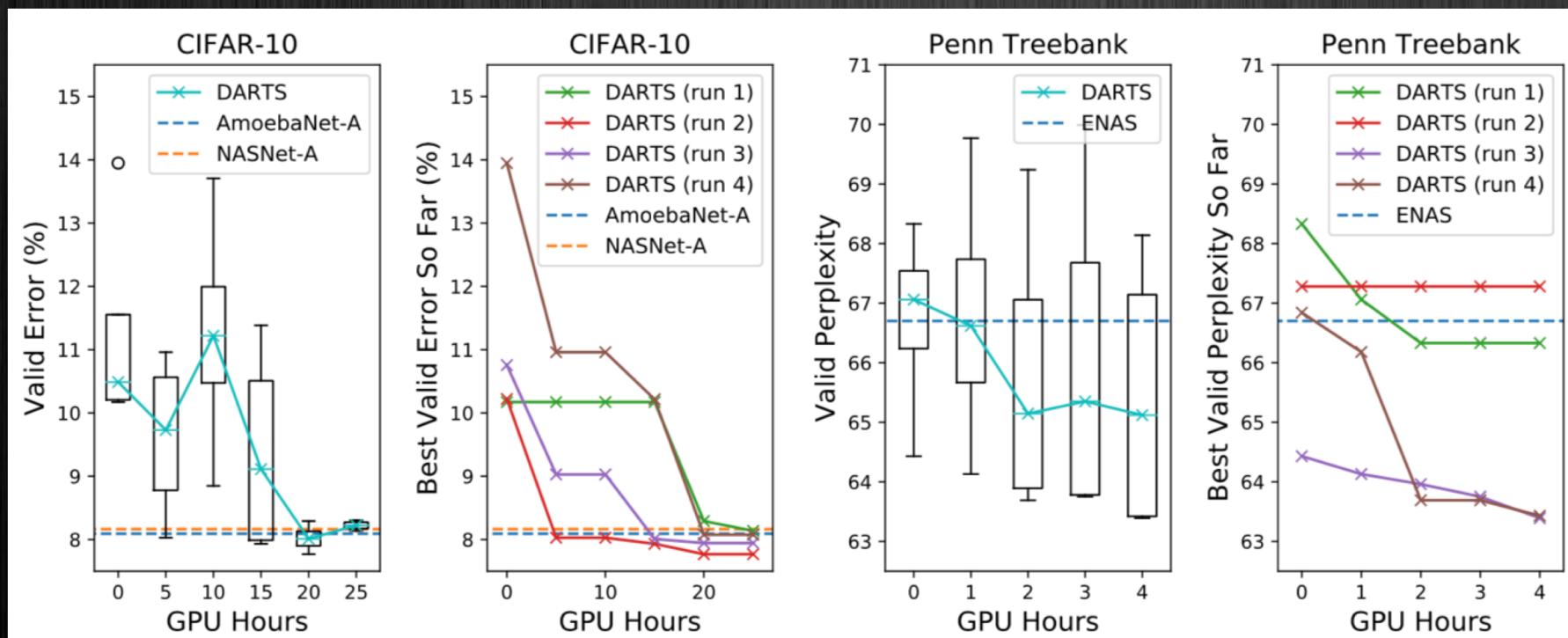
Figure 6: Recurrent cell learned on PTB.



Darts 实验结果

1. 架构搜索

Darts在CIFAR-10数据集上进行CNN的cell训练。在Penn Treebank上训练RNN cell。整个构架搜索是初始化敏感的，CNN cell经过长时间训练后能得到比较好的收敛，但是RNN cell的结果与初始化关系比较密切。





Darts 实验结果

2. 架构评估

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10. Results marked with † were obtained by training the corresponding architectures using our setup.

Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	manual
NASNet-A + cutout (Zoph et al., 2017)	2.65	3.3	1800	RL
NASNet-A + cutout (Zoph et al., 2017)†	2.83	3.1	3150	RL
AmoebaNet-A + cutout (Real et al., 2018)	3.34 ± 0.06	3.2	3150	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	evolution
Hierarchical Evo (Liu et al., 2017b)	3.75 ± 0.12	15.7	300	evolution
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	RL
Random + cutout	3.49	3.1	–	–
DARTS (first order) + cutout	2.94	2.9	1.5	gradient-based
DARTS (second order) + cutout	2.83 ± 0.06	3.4	4	gradient-based

 Darts 实验结果

3. 迁移学习

Table 3: Comparison with state-of-the-art image classifiers on ImageNet in the mobile setting.

Architecture	Test Error (%)		Params (M)	+ × (M)	Search Cost (GPU days)	Search Method
	top-1	top-5				
Inception-v1 (Szegedy et al., 2015)	30.2	10.1	6.6	1448	–	manual
MobileNet (Howard et al., 2017)	29.4	10.5	4.2	569	–	manual
ShuffleNet 2× (v1) (Zhang et al., 2017)	29.1	10.2	~5	524	–	manual
ShuffleNet 2× (v2) (Zhang et al., 2017)	26.3	–	~5	524	–	manual
NASNet-A (Zoph et al., 2017)	26.0	8.4	5.3	564	1800	RL
NASNet-B (Zoph et al., 2017)	27.2	8.7	5.3	488	1800	RL
NASNet-C (Zoph et al., 2017)	27.5	9.0	4.9	558	1800	RL
AmoebaNet-A (Real et al., 2018)	25.5	8.0	5.1	555	3150	evolution
AmoebaNet-B (Real et al., 2018)	26.0	8.5	5.3	555	3150	evolution
AmoebaNet-C (Real et al., 2018)	24.3	7.6	6.4	570	3150	evolution
PNAS (Liu et al., 2017a)	25.8	8.1	5.1	588	~225	SMBO
DARTS (searched on CIFAR-10)	26.9	9.0	4.9	595	4	gradient-based

3

总结与改进

Conclusion the validation

总结

改进

➤ 总结与不足

总结

引入了一种适用于卷积和循环结构的可微分网络体系结构搜索的新算法

实现了卓越的结构搜索效率，这归因于使用基于梯度的优化而非非微分搜索技术

证明DARTS在CIFAR-10和PTB上学习的体系结构可以迁移到ImageNet和WikiText-2上

不足

近似优化是简单的 w 与 α 的迭代更新，只是实践有效，应该有改进的空间

它对于RNN cell 的初始化敏感是需要优化的

现在的迁移学习就是简单的把cell运用到新的结构上，是否考虑添加distill，可能对于RNN这种表现一般的迁移有更好的效果。

4

答 疑

謝 謝 聆 听

——張 嬌 翩 2019-03-23