Learning Latent Architectural Distribution in Differentiable Neural Architecture Search via Variational Information Maximization

- 作者：Yaoming Wang，Yuchen Liu，Wenrui Dai et.al.
- 机构：Shanghai Jiao Tong University
- 会议：ICCV 2021
- 地址：https://openaccess.thecvf.com/content/ICCV2021/html/Wang_Learning_Latent_Architectural_Distribution_in_Differentiable_Neural_Architecture_Search_via_ICCV_2021_paper.html
- 代码：暂无

论文主要内容

摘要

现有的differentiable NAS搜索过程做了个不合适的假设：edge之间相互独立。这篇文章将架构分布看成数据点的latent representation。并提出Variational Information Maximization NAS (VIM-NAS)去建模latent representation。最大化互信息学习一个（架构分布representation）CNN实现快速收敛。

贡献

1. 提出了一种新的视角：在NAS里，网络架构分布能被看作给定数据集后的latent representation
2. 使用neural network去建模有依赖关系的架构分布（simple yet effective）
3. 给出了一个搜索策略，即最大化互信息的variational lower bound
4. VIM-NAS is the **fastest** differentiable NAS approaches up to now.

研究内容

Motivation

- NAS可以看作是在给定数据集上搜索一种偏好（preference）
- 现有的微分的NAS做了一个不合适的假设

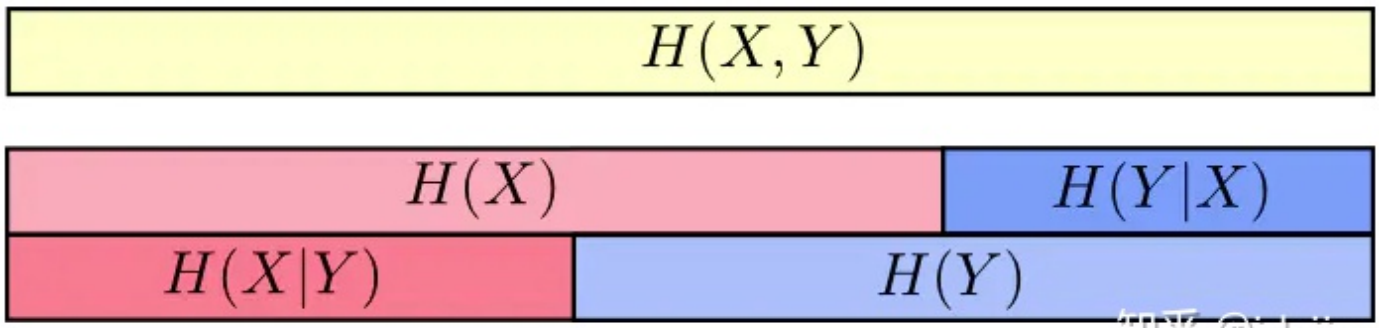
    - “The predefined architectural distribution on each edge is independent of each other”
    - 即假设不同edge上的op分布独立
    - 例如：架构中的一条边如果选了op A，那么另一条边更适合选op B（有依赖关系）
- 对于Differentiable NAS的search cost高
- 这篇文章就提出Variational Information Maximization（VIM）-NAS，通过最大化data points和latent architecture representation的互信息来高效搜索

方法

介绍

Differentiable NAS：核心是将离散的operation松弛成连续的

$$f_{i,j}\left(x_i\right)=\sum_{o\in\mathcal{O}}\frac{\exp\left(\alpha_{i,j}^{o}\right)}{\sum_{o'\in\mathcal{O}}\exp\left(\alpha_{i,j}^{o'}\right)}\cdot o\left(x_i\right)$$



熵的关系图

算法流程

VIM-NAS

数据集 $\mathcal{D}$ ： $\{(x_n,y_n)\}_{n=1}^{N}$ ，网络架构： A (都看作随机变量)

$p_\phi(\mathcal{D},A)=p(\mathcal{D})p_\phi(A\mid\mathcal{D})$ ， ϕ **是NAS的优化对象（分布参数）。**

在Differentiable NAS中，优化对象：learn the **parameter** ϕ that accurately predicts the specific

dataset $\mathcal{D}$ using the architecture A. 为此文章优化最大互信息：

$$\begin{aligned}\max_{\phi} I_{\phi}(\mathcal{D}, A) &= \mathbb{E}_{p_{\phi}(\mathcal{D},A)}\left[\log \frac{p_{\phi}(\mathcal{D}, A)}{p(\mathcal{D}) p_{\phi}(A)}\right] \\ &= H(\mathcal{D}) - H_{\phi}(\mathcal{D} \mid A)\end{aligned}$$

所以优化目标：

$$\max_{\phi} -H_{\phi}(\mathcal{D} \mid A) = \mathbb{E}_{p_{\phi}(\mathcal{D},A)}\left[\log p_{\phi}(\mathcal{D} \mid A)\right]$$

直接在高维上优化这个条件熵困难，转而优化它的lower bound：

$$\begin{aligned}I_{\phi}(\mathcal{D}, A) =& H(\mathcal{D}) + \mathbb{E}_{p_{\phi}(\mathcal{D},A)}\left[\log q_{\theta}(\mathcal{D} \mid A)\right] \\ &+ D_{KL}\left(p_{\phi}(\mathcal{D} \mid A) \| q_{\theta}(\mathcal{D} \mid A)\right) \\ \geq& H(\mathcal{D}) + \mathbb{E}_{p_{\phi}(\mathcal{D},A)}\left[\log q_{\theta}(\mathcal{D} \mid A)\right]\end{aligned}$$

真实的分布P由Q（**由带权重 θ 的神经网络给出，这个 θ 是weight**）来近似，当P=Q，就是在直接优化互信息。

最终的优化目标：

$$\max_{\theta,\phi} \mathbb{E}_{p_{\phi}(\mathcal{D},A)}\left[\log q_{\theta}(\mathcal{D} \mid A)\right]$$

$$\max_{\theta,\phi} \mathcal{L}(\phi, \theta; \mathcal{D}) = \sum_{d \in \mathcal{D}} \mathbb{E}_{p_{\phi}(A|\mathcal{D})}\left[\log q_{\theta}(\mathcal{D} \mid A)\right] \ ?$$

$$\max_{\theta,\phi} \mathcal{L}(\phi, \theta; \mathcal{D}) = \sum_{d \in \mathcal{D}} \mathbb{E}_{p_{\phi}(A|d)}\left[\log q_{\theta}(d \mid A)\right]$$ 式中的期望可以用MC算

- 已有的differentiable NAS方法会假设：$p_{\phi}(A \mid \mathcal{D}) = \prod_{i} p_{\phi_i}\left(A_i \mid \mathcal{D}\right)$，所有架构参数独立，简化搜索过程
- 文章给出的想法是利用CNN能拟合任意函数的性质，来表示架构分布 $p_{\phi}(A \mid \mathcal{D})$

Algorithm 1 VIM-NAS

Input: Data $\mathcal{D} = \{x_n, y_n\}_{1:N}$, initialized network weights θ, initialized architectural neural network parameters $\boldsymbol{\phi}$, and input Gaussian noise ϵ.

Output: The searched final architecture.

while not converged **do**
  Sample Gaussian noise $\xi \sim \mathcal{N}(0, 1)$.
  Sample architecture $A = \boldsymbol{\phi}(\epsilon) + \xi$, $A \sim p_{\boldsymbol{\phi}}(A|\mathcal{D})$.
  Update weights θ by descending $\nabla_\theta \mathcal{L}(\phi, \theta; \mathcal{D})$.
  Update network $\boldsymbol{\phi}$ by descending $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\phi, \theta; \mathcal{D})$.
end while
Derive the final architecture based on the learned $\boldsymbol{\phi}$.

表示分布的网络

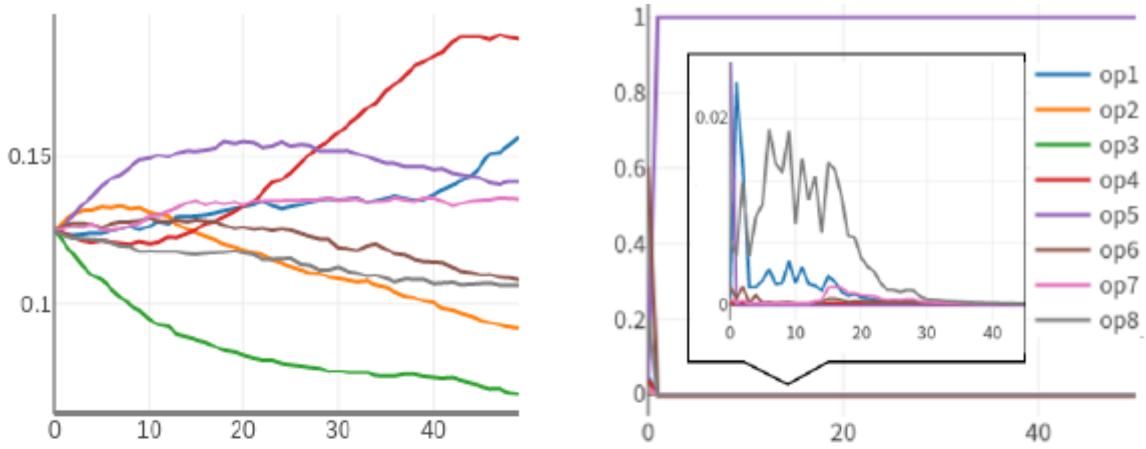
ConvReLUBN(3,14,3)-ConvReLUBN(14,1,3)

(a) Initialized feature map

(b) Convergent feature map after one epoch

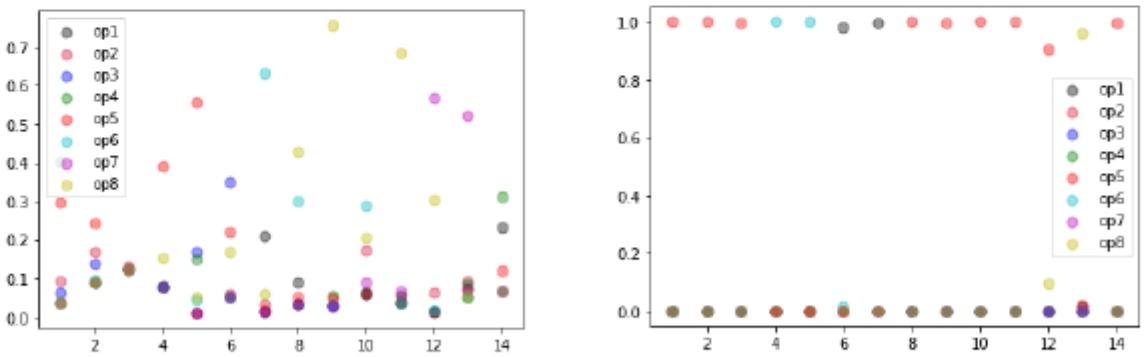
Figure 1. Comparison with initialized feature map and convergent feature map (one epoch training) of intermediate layer. In each sub figure, 14 channels of feature maps and each feature map shares the same size as candidate operations and edges.

- 初始dense and random
- 一个epoch之后sparse



(a) DARTS-N (b) VIM-NAS-N

Figure 2. Anytime architectural weights on the first edge of normal cell on DARTS search space for DARTS and VIM-NAS. 'N' denotes the searched normal cell(best viewed in color).



(a) Initialized architectural distribution (b) Convergent architectural distribution after one epoch training

Figure 3. Contrast of architectural distribution between initialization and convergence after training one epoch.

DARTS space上只训练一个epoch就收敛

与GAN的关系

表示架构分布网络可以看作一个generator（把噪声映射到好的网络架构的分布）

超网可以看作一个discriminator（来区分出performance好的网络架构），没有好的网络架构作为GT

实验结果

Results

DARTS Search Space

CIFAR-10

- One epoch就收敛（within 10 minutes on a single NVIDIA GTX 1080 Ti GPU)）
- top-1 test error of 2.45% and 15.80% on CIFAR-10 and CIFAR-100

Architecture	Top-1 (Test) Error (%)		Params	Cost
	CIFAR-10	CIFAR-100	(M)	(days)
DARTS-V1 [29]	3.00 ± 0.14	17.76^*	3.3	0.4
DARTS-V2 [29]	2.76 ± 0.09	17.54^*	3.3	1
P-DARTS [8]	2.50	16.55^*	3.4	0.3
SNAS [41]	2.85 ± 0.02	-	2.8	1.5
PARSEC [5]	2.81 ± 0.03	-	3.7	1
BayesNAS [53]	2.81 ± 0.04	-	3.4	0.2
DATA (M = 7) [51]	2.79	-	2.9	1
PC-DARTS [42]	2.57 ± 0.07	-	3.6	0.1
ASNG-NAS [1]	2.83 ± 0.14	-	3.9	0.11
SI-VDNAS-C† [37]	2.60 ± 0.05	16.20	2.7	0.8
GDAS [15]	2.93	18.38	3.4	0.21
SDARTS-ADV [7]	2.61 ± 0.02	-	3.3	1.3
SGAS [25]	2.66 ± 0.24	-	3.7	0.25
DARTS- [10]	2.59 ± 0.08	-	3.5	0.4
TE-NAS [6]	2.63 ± 0.06	-	3.8	0.05
VIM-NAS	$\mathbf{2.45 \pm 0.04}$	**15.80**	3.9	**0.007**

Table 1. Comparison with state-of-the-art gradient-based NAS methods for image classification on CIFAR-10/100. For each method, top-1 test error (%), number of parameters (M) and search cost (GPU-days) are evaluated. Here, lower error rate stands for better performance and * indicates that the experiments are conducted by P-DARTS. † denotes that SI-VDNAS-C are the searched convergent cell.

ImageNet

- 用CIFAR-10搜出来的架构放在ImageNet上评估： top-1 error rate of 24.0% and a top-5 error rate of 7.2%
- 在imagenet上搜：top-1error rate of 23.8% and a top-5 error rate of 7.1%

Table 5: Transfer learning results on ImageNet

Architecture	Test Error(%)	Params (M)
NASNet-A [51] *	26.0	5.3
AmoebaNet-A [31] *	25.5	5.1
PNAS [22] *	25.8	5.1
SNAS [43] *	27.3	4.3
DARTS [23] *	26.7	4.7
SDARTS-ADV [6]	25.2	4.8
arch2vec-BO [45] *	25.5	5.2
RANK-NOSH	25.2	5.3

* Results obtained from the arch2vec paper [45].

NAS-Bench-201

所有对比的方法都经过四次不同随机种子的独立实验

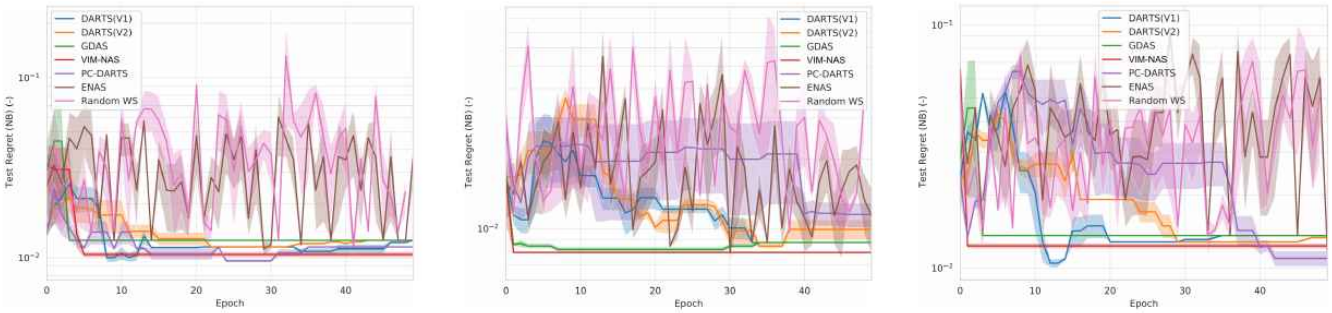
Architecture	Test Error (%)		FLOPS	Search Cost
	Top-1	Top-5	(M)	(GPU-days)
DARTS (2nd) [29]	26.7	8.7	574	1
GDAS [15]	26.0	8.5	581	0.21
PARSEC [5]	26.0	8.4	-	1
PC-DARTS [42]	25.1	7.8	586	0.1
PC-DARTS† [42]	24.2	7.3	597	3.8
P-DARTS [8]	24.4	7.4	557	0.3
DARTS+† [27]	23.9	7.4	582	6.8
DARTS-† [10]	23.8	7.0	467	4.5
FairDARTS-B [11]	24.9	7.5	541	0.4
DSO-NAS-share [52]	25.4	8.4	586	6
SDARTS-ADV [7]	25.2	7.8	-	1.3
SGAS [25]	24.2	7.2	585	0.25
SparseNAS [40]	24.7	7.6	-	1
BayesNAS [53]	26.5	8.9	-	0.2
DATA (M = 7) [51]	24.9	8.1	-	1.5
SI-VDNAS-B [37]	25.3	8.0	577	0.3
TE-NAS [6]	26.2	8.3	-	0.05
TE-NAS† [6]	24.5	7.5	-	0.17
VIM-NAS	**24.0**	**7.2**	627	**0.007**
VIM-NAS†	**23.8**	**7.1**	660	**0.26**

Table 2. Comparison with state-of-the-art gradient-based NAS methods on ImageNet. For each method, top-1 and top-5 test errors (%), FLOPS (M) and search cost (GPU-days) are evaluated. Here, lower error rate stands for better performance and † indicates that the architecture is directly searched on ImageNet.

- 仍然one epoch就收敛（232.51 seconds on a single NVIDIA GTX 1080 Ti GPU）
- 尽管TE-NAS提出了一个training-free的策略，但搜索cost是VIM-NAS六倍

NAS-Bench-1Shot1

- 里面有3个search space
- several epochs (2-5 epochs) in all three search spaces（op分布在一个epoch收敛，但拓扑参数需要几个epoch去收敛）



(a) Search space 1 (b) Search space 2 (c) Search space 3

Figure 4. Anytime test regret on NAS-Bench-1Shot1 (best viewed in color).

Simplified Search Spaces S1-S4

- 一个epoch收敛（within 10 minutes）

Benchmark		DARTS†	R-DARTS†		DARTS†		DARTS-†	Ours†	PC-DARTS‡	SDARTS‡		DARTS-‡	Ours‡
			DP	L2	ES	ADA				RS	ADV		
C10	S1	3.84	3.11	2.78	3.01	3.10	2.68	**2.61**	3.11	2.78	2.73	2.68	**2.61**
	S2	4.85	3.48	3.31	3.26	3.35	3.71	**3.22**	3.02	2.75	2.65	2.63	**2.53**
	S3	3.34	2.93	2.51	2.74	2.59	**2.42**	**2.42**	2.51	2.53	2.49	**2.42**	**2.42**
	S4	7.20	3.58	3.56	3.71	4.84	3.88	**3.55**	3.02	2.93	2.87	2.86	**2.85**
C100	S1	29.46	25.93	24.25	28.37	24.03	22.41	**22.07**	18.87	17.02	16.88	16.92	**16.12**
	S2	26.05	22.30	22.24	23.25	23.52	21.61	**20.90**	18.23	17.56	17.24	**16.14**	16.35
	S3	28.90	22.36	23.99	23.73	23.37	21.13	**21.11**	18.05	17.73	17.12	**15.86**	15.94
	S4	22.85	22.18	21.94	21.26	23.20	21.55	**21.01**	17.16	17.17	**15.46**	17.48	17.39

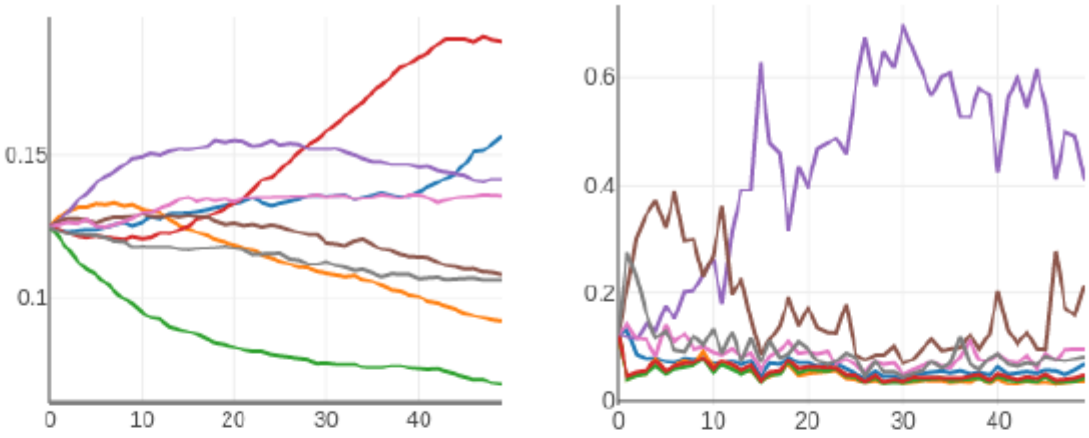
Table 4. Comparison in various search spaces. We report the lowest error rate (%) of 4 found architectures. ‡: under [7, 10] evaluation settings where all models have 20 layers and 36 initial channels. †: under [48] settings where CIFAR-10 models in S2 and S4 have 20 layers and 16 initial channels, and CIFAR-100 models have 8 layers and 16 initial channels.

消融实验

实验使用DARTS搜索空间

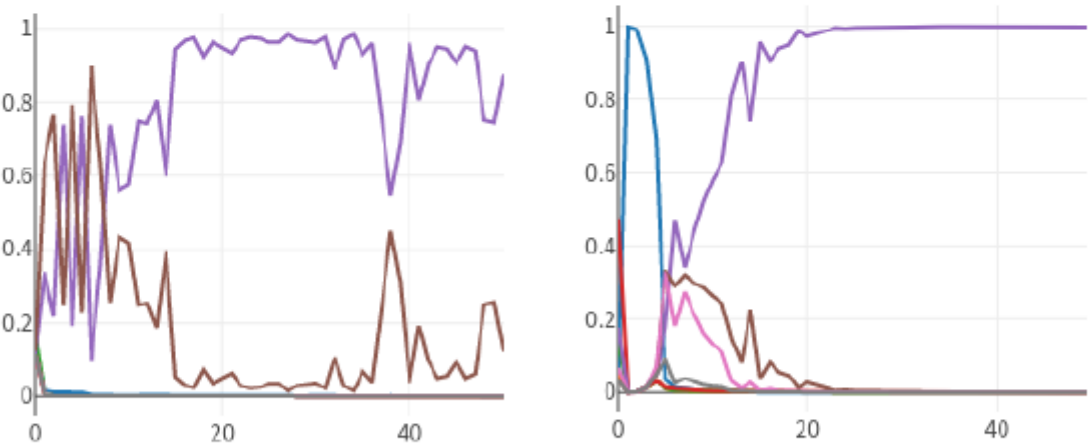
Architectural Network

- 原本设置（一个epoch收敛，快速稳定收敛）
- 单个ConvReLUBN （limited capacity->收敛不稳定）
- 五个块（大网络需要更多epoch去训练->需要几个epoch去收敛）



(a) DARTS-N (b) DARTS†-N

Figure 6. Anytime architectural weights on DARTS search space. 'N' denotes the searched normal cell. DARTS†: DARTS is implemented with high learning rate (0.025) and added noise.

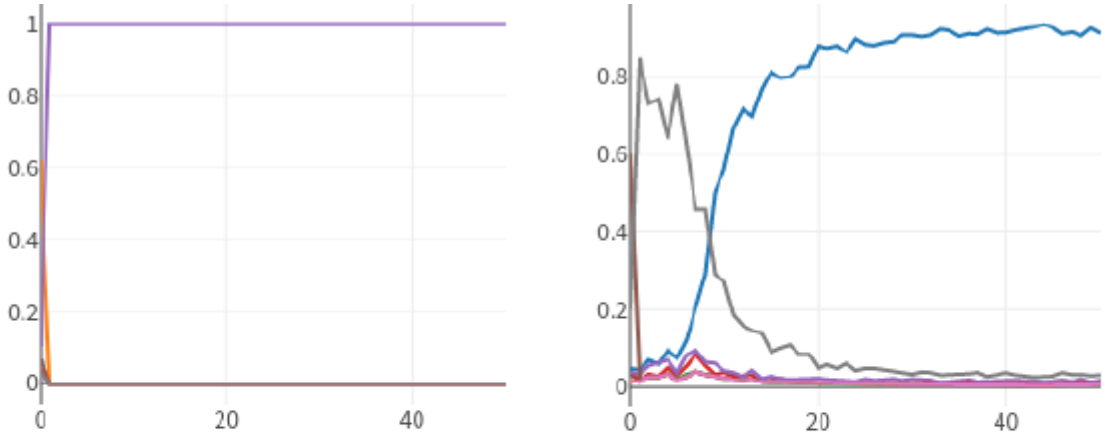


(a) VIM-NAS-S-N (b) VIM-NAS-L-N

Figure 7. Anytime architectural weights on DARTS search space. 'N' denotes the searched normal cell. VIM-NAS-S: VIM-NAS implemented with a small architectural network. VIM-NAS-L: VIM-NAS implemented with a large architectural network.

实验：

- 点估计（VIM-NAS-P）替代分布估计
- 增加方差的参数化（VIM-NAS-Dropout）



(a) VIM-NAS-D-N (b) VIM-NAS-P-N

Figure 9. Anytime architectural weights on DARTS search space. 'N' denotes the searched normal cell. VIM-NAS-D: reformulation of variational dropout NAS with VIM. VIM-NAS-P-N: VIM-NAS implemented with point estimation.

说明：

- 点估计收敛less effective（慢、且有波动）
- VIM-NAS-Dropout 和 VIM-NAS一样快、稳定