BPNAS: Bayesian Progressive Neural Architecture Search

Hyunwoong Chang[1,*], Anirban Samaddar[2,*], Sandeep Madireddy[2,†]

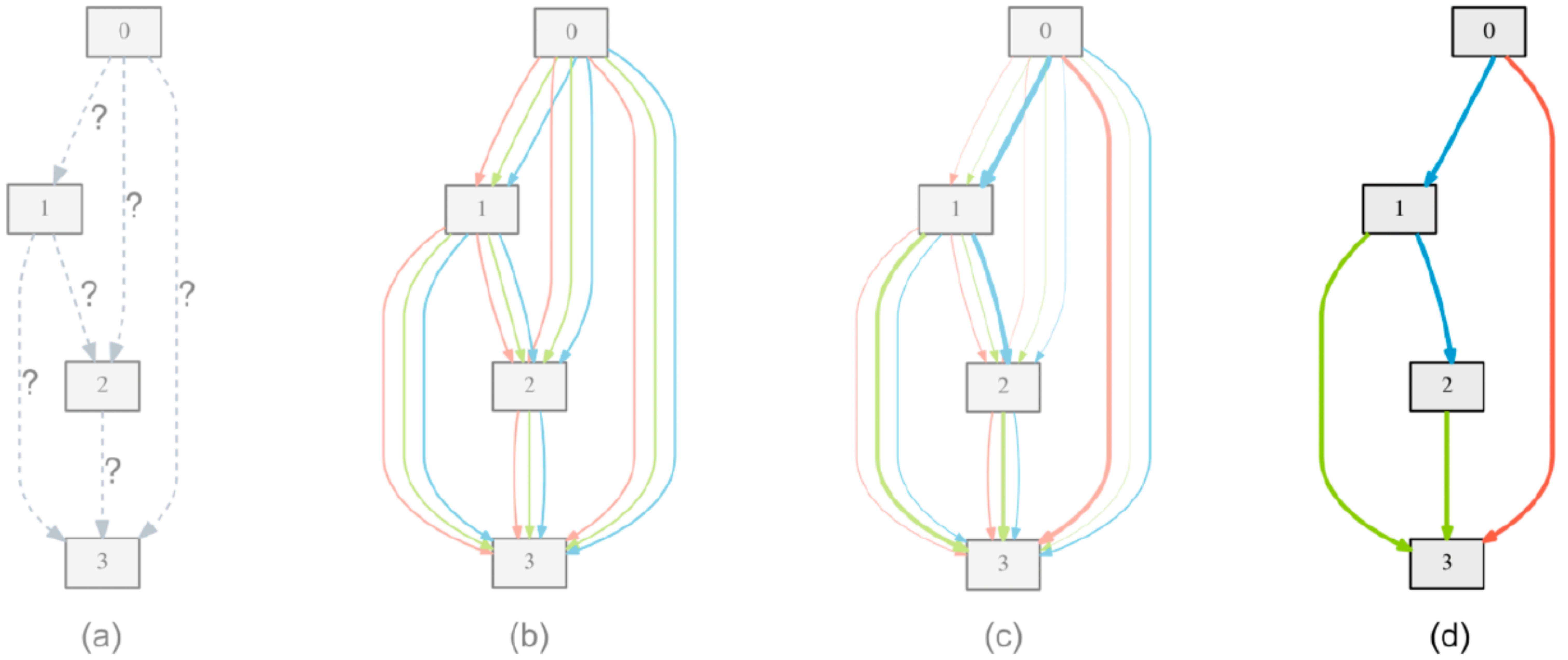
[1] Texas A&M University, [2] Argonne National Laboratory,
[*] Equal contribution, [†] Corresponding author

Anirban Samaddar

Introduction

- Objective of NAS is to automate designing of neural networks
- Traditional NAS methods use evolutionary algorithms and reinforcement learning (heavy computation burden)
- Differentiable NAS[1] is an attractive alternative that incorporates differentiability in the search process hence increase efficiency

Differentiable NAS



A cell is a directed acrylic graph consisting N (=4 here) nodes with unknown operations (convolution, skip-connect etc.)

A super-network is assumed where output of each node is an weighted average of K operations

The weights are learned as a part of the training

An architecture is selected by choosing the operators with the maximum weights

Fig.1: Schematic of differentiable NAS[1]

Differentiable NAS

- The differentiable NAS objective function:

$$\min_{\alpha} \mathscr{L}_{\mathrm{val}}(\alpha, W^*) \ \mathrm{s.t.}\ W^* = \arg\min_{W} \mathscr{L}_{\mathrm{train}}(\alpha, W)$$

  α, W are the weights of the operators and the weights (and biases) of the network

- The loss is differentiable w.r.t the parameters and therefore they can be learned with existing gradient based tools

A Bayesian Framework

- We propose a unified Bayesian framework for the architecture search —

$$\pi(\alpha, W, \Gamma) \propto \pi(\alpha, W \mid \Gamma)\pi(\Gamma)$$

$$\pi(\alpha, W \mid \Gamma) \propto e^{-\mathscr{L}(\alpha(\Gamma), W(\Gamma) \mid \Gamma)}$$

$$\pi(\Gamma) \propto \prod_{m=1}^{n_{edge}} \pi(\gamma_m) = \prod_{m=1}^{n_{edge}} e^{-c|\gamma_m|} 1_{\{|\gamma_m|>0\}}(\gamma_m)$$

n_{edge} is the number of edges, and Γ, a binary matrix, is introduced to select from the space of architectures

- The prior for Γ is set to ensure that it enforces a mode where every edge has one operator ($|\gamma_m| = 1$)

BPNAS algorithm

- Our goal is to learn– $(\hat{\alpha}^{\text{MAP}}, \hat{W}^{\text{MAP}}, \hat{\Gamma}^{\text{MAP}}) = \arg\max_{\alpha, W, \Gamma} \pi(\alpha, W, \Gamma)$
- Difficult to find this MAP estimate from the joint distribution of discrete and continuous random variable
- We devise an algorithm to prune out less important operators based on the weight matrix α

BPNAS algorithm

Bayesian Progressive architecture search (BPNAS):

Input: An initial super-network ($\Gamma = \mathbf{1}$) with architecture parameters α , weights W, posterior distribution $\pi(W, \alpha, \Gamma)$, and a threshold δ

While $||\gamma_m||_0 > 1$ for some k:

Update W, α by stochastic gradient descent

If $||\theta(\alpha_m) - \frac{\mathbf{1}_{||\gamma_m||_0}}{||\gamma_m||_0}||_2 \geq \delta$ holds:

Select $(l, m) = argmin_{\{(l', m'):\Gamma_{l'm'}=1\}} \alpha_{l'm'}$ and set $\Gamma_{l'm'} = 0$

Reset α

Output: An architecture Γ with a single operation for each edge.

- We chose $\theta(\alpha_m) \sim Dirichlet(\alpha_m)$

Architecture Ensemble

- Our Bayesian framework enables efficient method to sample architectures from the posterior
- Upon convergence to an architecture $\hat{\Gamma}$, we restart the algorithm N times by randomly sampling an edge s and set $\gamma_s = \mathbf{1}$
- We perform this (in parallel) to build an ensemble of architectures

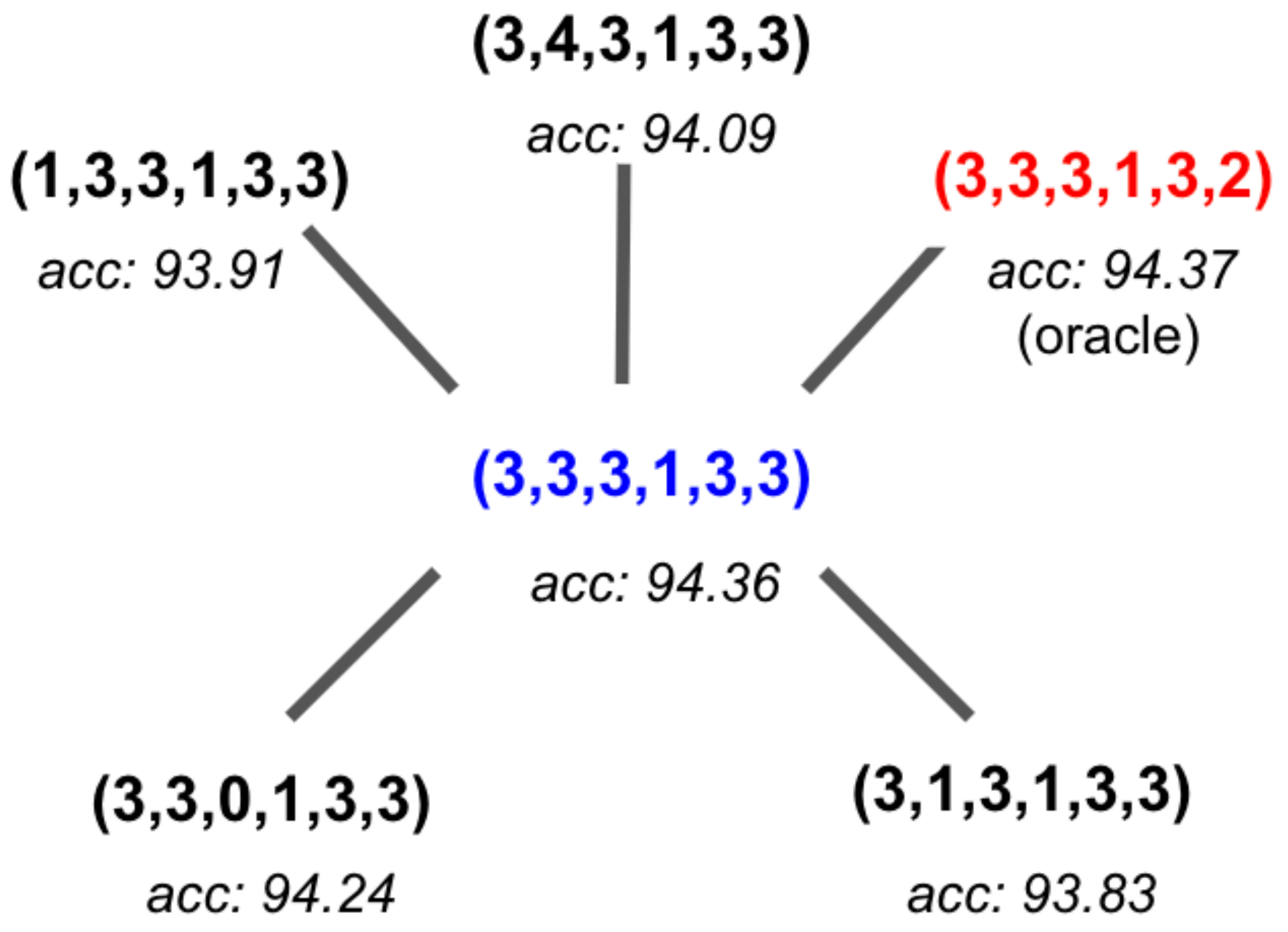


Fig.2: An ensemble in NAS-Bench-201[2]

Results

Architecture search

Methods	CIFAR-10		CIFAR-100		ImageNet	
	Test	Epochs	Test	Epochs	Test	Epochs
ResNet	93.97	100	70.86	100	43.63	100
DrNAS[3]	**94.36 ± 0.0**	100	71.00 ± 1.3	100	**46.34 ± 0.0**	100
BPNAS	94.18 ± 0.3	**63**	**73.40 ± 0.2**	**50**	**46.34 ± 0.0**	**26**
Optimal	94.37	-	73.51	-	47.31	-

- On CIFAR-10 and ImageNet dataset, BPNAS performs similarly to the state-of-the-art while outperforming on CIFAR-100
- Since BPNAS used pruning it converges to the final architecture faster (~70% less epochs on ImageNet)

Results

Architecture Ensemble

Methods	CIFAR-10	CIFAR-100	ImageNet
NES-RS[4]	94.17 ± 0.3	74.42 ± 0.8	45.66 ± 1.7
NESBS[5]	94.08 ± 0.1	75.00 ± 0.2	47.32 ± 0.4
BPNAS	**95.10 ± 0.3**	**77.47 ± 1.0**	**50.27 ± 0.5**

- On all datasets, BPNAS ensemble outperforms all the state-of-the-art architecture ensemble algorithms

Thank you!

References

1. Liu, H., K. Simonyan, and Y. Yang (2018). “Darts: Differentiable architecture search”. In: arXiv preprint arXiv:1806.09055.

2. Dong, X. and Y. Yang (2020). “Nas-bench-201: Extending the scope of reproducible neural architecture search”. In: arXiv preprint arXiv:2001.00326.

3. Chen, X., R. Wang, M. Cheng, X. Tang, and C.-J. Hsieh (2020). “Drnas: Dirichlet neural architecture search”. In: arXiv preprint arXiv:2006.10355.

4. Zaidi, S., A. Zela, T. Elsken, C. C. Holmes, F. Hutter, and Y. Teh (2021). “Neural ensemble search for uncertainty estimation and dataset shift”. In: Advances in Neural Information Processing Systems 34, pp. 7898–7911.

5. Shu, Y., Y. Chen, Z. Dai, and B. K. H. Low (2022). “Neural ensemble search via Bayesian sampling”. In: Uncertainty in Artificial Intelligence. PMLR, pp. 1803–1812.

6. Zoph, B., V. Vasudevan, J. Shlens, and Q. V. Le (2018). “Learning transferable architectures for scalable image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8697–8710.