Understanding and Exploring the Network with Stochastic Architectures

Zhijie Deng, Yinpeng Dong, Shifeng Zhang, Jun Zhu

Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Lab, Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University

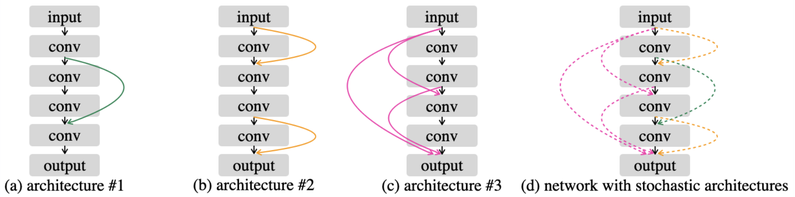
NEURAL INFORMATION PROCESSING SYSTEMS

Motivation and introduction

❑ **There is an emerging trend to train a network with stochastic architectures to enable various architectures to be plugged and played during *inference*.**

❑ **However, the existing investigation is highly entangled with neural architecture search (NAS), limiting its widespread use across scenarios.**

❑ **In this work, we decouple the training of a network with stochastic architectures (NSA) from NAS and provide a first systematical investigation on it as a stand-alone problem.**



(a) architecture #1 (b) architecture #2 (c) architecture #3 (d) network with stochastic architectures

❑ **The stochasticity over the network architecture is promising:**

- ➢ It works as a stochastic regularization to regularize deep models from *co-adaptation* and *over-fitting*, in a more *structured* style than standard stochastic regularizations applied locally.
- ➢ The trained weight-sharing network can adopt diverse architectures, seen or even *unseen* during training, to perform inference, enabling us to leverage the expressivity of various architectures with training *only one* set of weights.
- ➢ The predictions provided by different architectures can be further assembled or used to calculate uncertainty estimates, making the prediction model more *accurate*, *robust*, and *calibrated*.

❑ **The setup of our empirical analyses:**

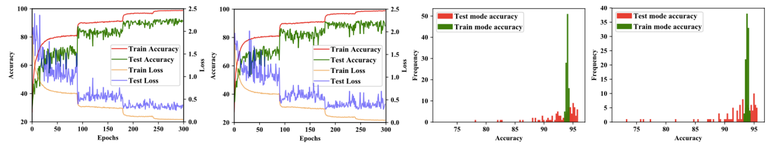
- ➢ The training objective for weight updating – the expected empirical risk w.r.t. the variable architecture:

$$L(\mathbf{w}) = \mathop{\mathbb{E}}_{\boldsymbol{\alpha}\sim p(\boldsymbol{\alpha})}\left[\frac{1}{n}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}} -\log p(y_i|\mathbf{x}_i;\mathbf{w},\boldsymbol{\alpha})\right] \approx \frac{1}{|\mathcal{B}|}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{B}} -\log p(y_i|\mathbf{x}_i;\mathbf{w},\boldsymbol{\alpha}),\ \boldsymbol{\alpha}\sim p(\boldsymbol{\alpha})$$

- ➢ Test principles: the validation performance of a specific architecture α_0; the ensemble performance of T architectures
- ➢ The *training* architecture distribution $p(\alpha)$: Following the wiring view of architecture, we fix a set of layers, and activate skip-connections using the Erdo˝s-Rényi model to draw S architectures from the whole space. $p(\alpha)$ *is a uniform distribution over the S architecture samples*.

Training/test disparity of NSA

❑ **On CIFAR-10, the learning curves are plotted in (a)-(b), and the histograms for the validation accuracy of 100 random architectures (seen during training) tested under different modes are shown in (c)-(d):**

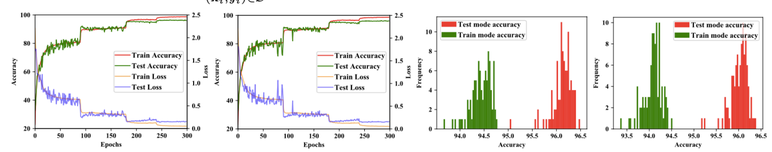


(a) $S = 500$ (b) $S = 5000$ (c) $S = 500$ (d) $S = 5000$

- The test loss and test accuracy curves show severe instability.

A key root – *Using the same architecture sample for the whole mini-batch shifts the features of various data towards similar directions, resulting in high-variance batch statistics during training.*

Solution – NSA-i: draw instance specific architecture samples for weight training: $L^*(\mathbf{w}) = \frac{1}{|\mathcal{B}|}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{B}} -\log p(y_i|\mathbf{x}_i;\mathbf{w},\boldsymbol{\alpha}_i),\quad \boldsymbol{\alpha}_i\sim p(\boldsymbol{\alpha}), i=1,...,|\mathcal{B}|$



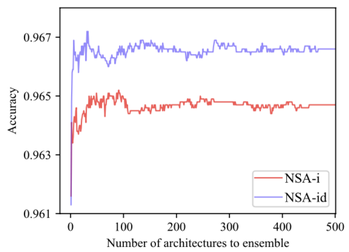
(a) $S = 500$ (b) $S = 5000$ (c) $S = 500$ (d) $S = 5000$

★ Test-mode BN induces notably better validation accuracy, implying the weaknesses of training-mode BN.

Mode collapse of diverse architectures

❑ **Can the predictive diversity from diverse architectures still be held given only a set of shared weights in NSA?**

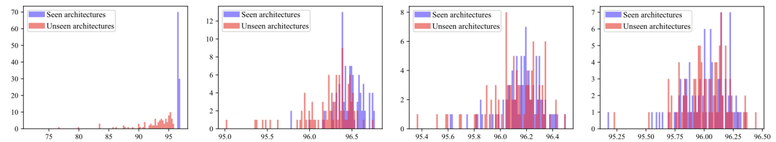
- A measure of predictive diversity: ensemble performance gain
- The gain of NSA-i is limited
- **Solution – NSA-id**: deploy an extra set of architecture dependent weights
- Where to deploy?
  - Aggregation module
  - Affine transformation in BN
- More obvious gain with negligible extra weights



Generalization capacity to unseen architectures

❑ **Can the trained NSA generalize to unseen architectures for broader exploration?**

- The histograms for the validation accuracy of 100 architectures seen during training vs. those for 100 unseen architectures (tested on NSA-i):



(a) $S = 5$ (b) $S = 50$ (c) $S = 500$ (d) $S = 5000$

➢ These results validate the generalization capacity of NSA, perhaps because the shared weights learn common structures of the architectures. As shown, we can train a NSA with a suitable number of architectures (e.g., [500, 5000]) to conjoin architecture *generalization* and *accuracy*.

Applications of NSA

❑ ***Model ensemble with stochastic architectures*** **(image classification)**

Method	# params	CIFAR-10		CIFAR-100	
		Test error (%) ↓	ECE ↓	Test error (%) ↓	ECE ↓
WRN-28-10 [49]	36.5M	4.00	-	19.25	-
DenseNet-BC [14]	25.6M	3.46	-	17.18	-
ENAS + CutOut [30]	4.6M	2.89	-	-	-
DARTS + CutOut [22]	3.4M	2.83	-	-	-
WRN-28-10†	39.5M	2.93	0.0140	16.75	0.0672
WRN-28-10†, MC dropout	39.5M	3.23	0.0107	17.16	0.0454
Average of individuals	39.5M	2.97	0.0153	17.02	0.0446
NSA-id	39.6M	**2.75**	**0.0032**	**16.44**	**0.0212**

❑ ***Uncertainty Estimation*** **(out-of-distribution detection)**

- adopt the mutual information between the prediction of incoming data and the variable architecture as the uncertainty measure

Method	OOD	PGD1-2-1		PGD2-3-1		PGD3-4-1	
	AUC ↑	Acc. ↑	AUC ↑	Acc. ↑	AUC ↑	Acc. ↑	AUC ↑
WRN-28-10†, MC dropout	0.935	0.622	0.735	0.345	0.694	0.183	0.564
NSA-id	**0.970**	**0.630**	**0.737**	**0.401**	**0.705**	**0.263**	**0.618**

Conclusion

❑ **We reveal un-identified training & test properties of NSA.**

❑ **We observe two issues and propose two solutions.**

❑ **We further provide valuable insights on how to train a NSA, hopefully benefiting NAS.**

❑ **We apply NSA into three appropriate scenarios.**