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 Center

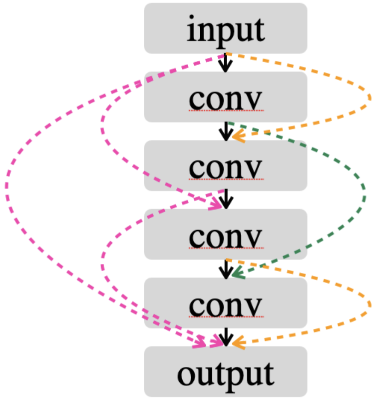
Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University, Beijing, 100084 China

Contact: dzj17@mails.tsinghua.edu.cn

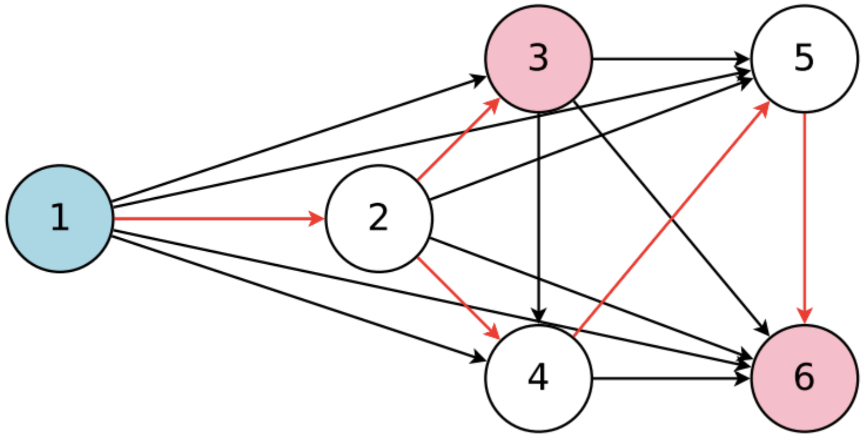
The Network with Stochastic Architectures

There is an emerging trend to train a network with stochastic architectures ***(NSA)*** to enable various architectures to be plugged and played during inference. This is also known as the weight sharing technique, popular used in **neural architecture search (NAS)**.

Despite widespread adoption in NAS, the property/pros/cons of such networks are unexplored, motivating us to perform a first systematical investigation on it as a stand-alone problem.



Stochastic architectures in a wiring view



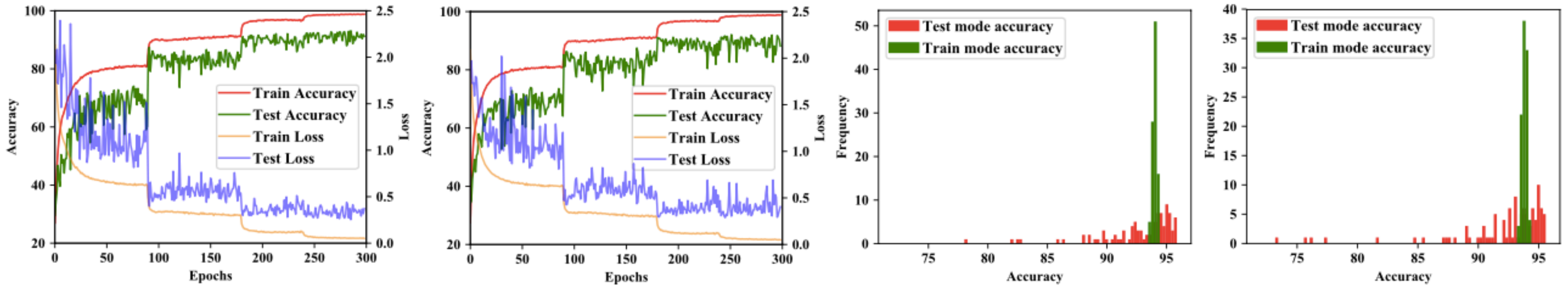
Stochastic architectures in a sub-graph view
Figure from Pham et al. (2018)

Training/test Disparity

- Training principle (expected empirical risk *w.r.t.* the variable architecture)

$$L(\mathbf{w}) \approx \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{B}} -\log p(y_i|\mathbf{x}_i; \mathbf{w}, \boldsymbol{\alpha}), \ \ \boldsymbol{\alpha} \sim p(\boldsymbol{\alpha})$$

- Test principle $\mathcal{A}(\boldsymbol{\alpha}_0) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{val}}} \mathbb{I}\big(\arg\max_y p(y|\mathbf{x}_i; \mathbf{w}, \boldsymbol{\alpha}_0) = y_i \big)$
- $p(\boldsymbol{\alpha})$ for training: a uniform distribution over *S* architectures sampled by the Erdős-Rényi (ER) model with 0.3 connection probability

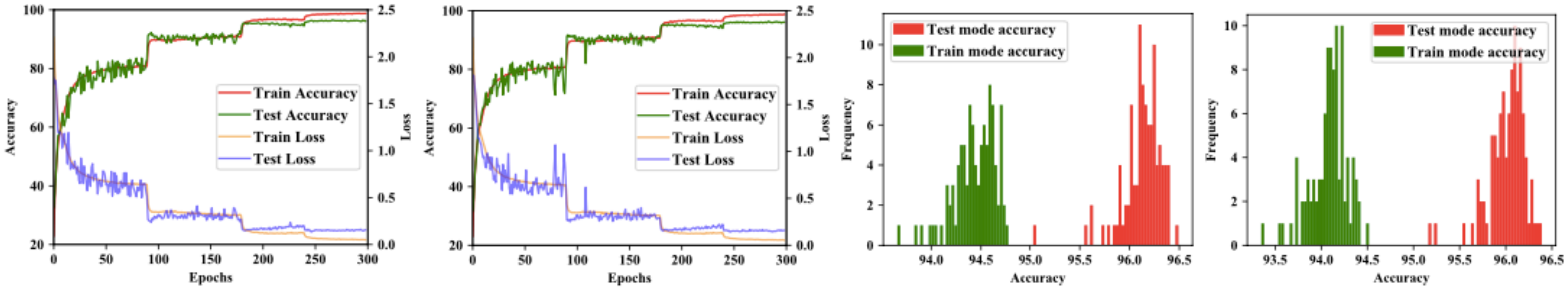


(a) $S = 500$ (b) $S = 5000$ (c) $S = 500$ (d) $S = 5000$

Training/test Disparity

NEURAL INFORMATION PROCESSING SYSTEMS

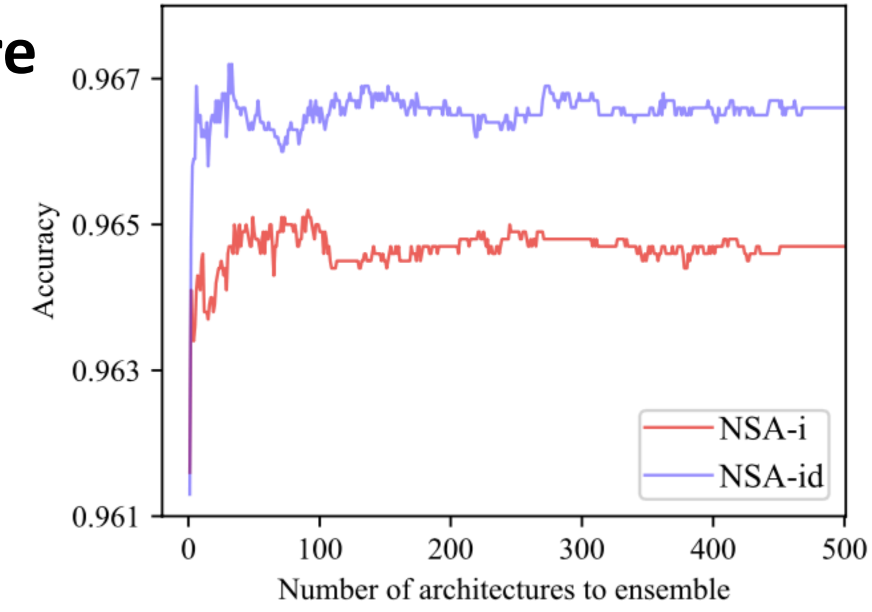
- Typically, the training and test disparity of a DNN model is caused by the train/val inconsistency of BN
- We identify the batch statistics of naïve NSA have high variance because the whole mini-batch **shares the same sampled architecture**
- As a solution, we advocate using *i.i.d* architectures for different instances during training



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

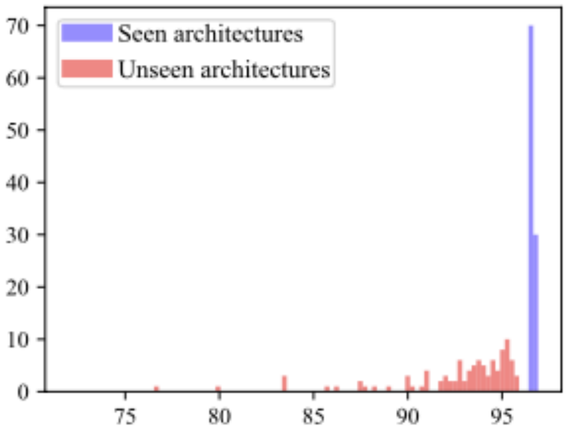
Mode Collapse of Diverse Architectures

- We further concern *'Do diverse architectures behave diversely given shared weights?'*
- Ensemble accuracy gain as a measure of **architecture behaviour diversity**
- NSA-i (trained with instance-wise architectures) shows limited ensemble performance gain (mode collapse)
- ***Augmenting the network with architecture-dependent weights*** alleviates this issue (see NSA-id)

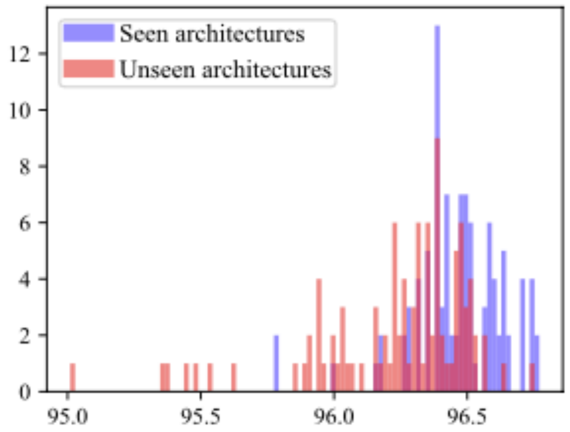


Generalization Capacity to Unseen Architectures

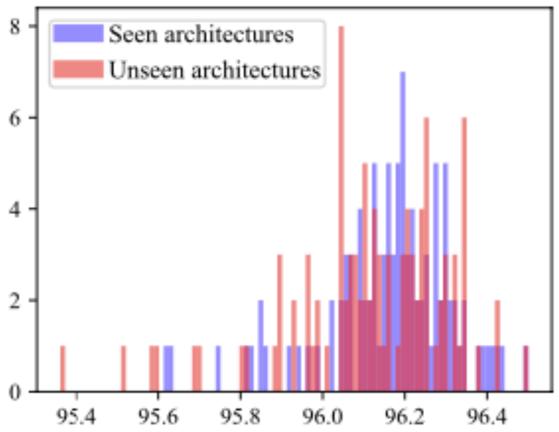
- We next concern ‘Can NSA trained under a limited architecture space generalize to unseen architectures in the broad, raw architecture space?’
- We calculate the test accuracy of 200 randomly sampled architectures (100 seen vs. 100 unseen during training) based on the NSA-i models trained under various S
- We plot the test accuracy histograms



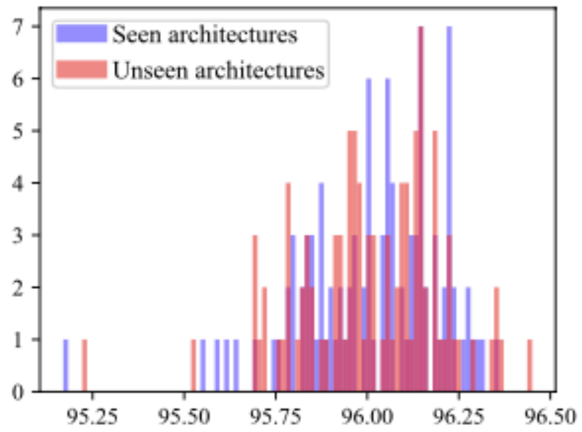
(a) $S = 5$



(b) $S = 50$



(c) $S = 500$



(d) $S = 5000$

Applications of NSA

- Model ensemble; uncertainty estimation; etc.

Method	# params	CIFAR-10 Test error (%) ↓	CIFAR-10 ECE ↓	CIFAR-100 Test error (%) ↓	CIFAR-100 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**

Method	OOD AUC ↑	PGD1-2-1 Acc. ↑	PGD1-2-1 AUC ↑	PGD2-3-1 Acc. ↑	PGD2-3-1 AUC ↑	PGD3-4-1 Acc. ↑	PGD3-4-1 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**

Code available at https://github.com/thudzj/NSA (Scan the QR code for this URL).

Thanks