Neural Ensemble Search via Bayesian Sampling

Kseniia Petrushina

MIPT, 2024

May 14, 2024

Motivation

Neural Architecture Search

Automate the design of well-performing architectures for different tasks

Neural Network Ensembles

- NAS algorithms select only one single architecture
- NNE achieve an improved performance compared with a single neural network in practice
- NES algorithm based on RS or evolutionary algorithm requires excessive search costs

Background

DARTS

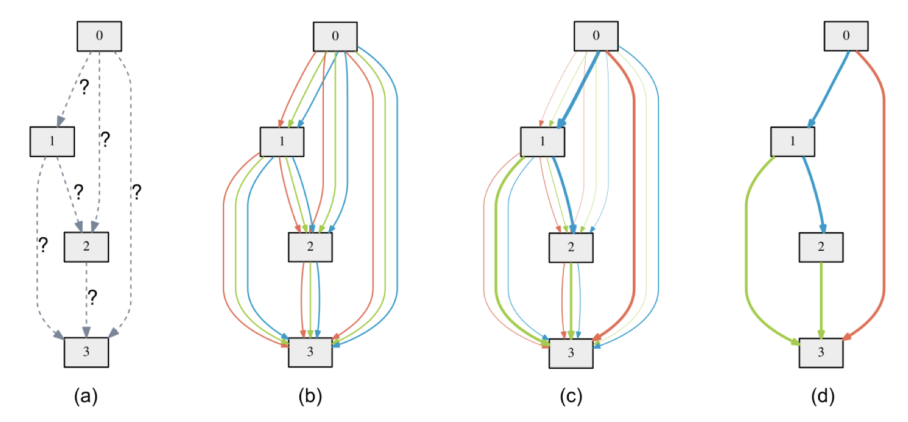

Figure: (a) Unknown operations. (b) Continious relaxation of the search space. (c) Joint optimization of mixing probabilities and network weights. (d) Final architecture.

Background

Stein Variational Gradient Descent

Approximate target distribution $p(\mathbf{x})$ with simple density $q^*(\mathbf{x}) \in \mathcal{Q}$:

$$q^* = \arg\min_{q \in \mathcal{Q}} \{\mathbf{KL}(q \| p) = \mathbb{E}_q[\log(q(\mathbf{x})/p(\mathbf{x}))]\}$$

$q^*(\mathbf{x})$ - set of particles $\{\mathbf{x}_i\}_{i=1}^n$ iteratively updated:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \varepsilon \phi^*(\mathbf{x})$$

$q_{[\varepsilon\phi]}$ - distribution of updated particles, then

$$\phi^* = \arg\max_{\phi \in \mathbb{F}} \left\{ -\frac{d}{d\varepsilon} \mathbf{KL}(q_{[\varepsilon\phi]} \| p) \Big|_{\varepsilon=0} \right\}$$

Background

Stein Variational Gradient Descent

Closed-form solution

$$\phi^*(\cdot) = \mathbb{E}_{x\sim q}[k(\mathbf{x},\cdot)\nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} k(\mathbf{x},\cdot)]$$

Empirical mean

$$\hat{\phi}^*(\mathbf{x}_i) = \frac{1}{n}\sum_{j=1}^{n} k(\mathbf{x}_j,\mathbf{x}_i)\nabla_{\mathbf{x}_j} \log p(\mathbf{x}_j) + \nabla_{\mathbf{x}_j} k(\mathbf{x}_j,\mathbf{x}_i)$$

First term favors particles with higher probabilty dentisy, second term pushes particles away from each other

NES via Bayesian Sampling

Ensemble scheme

$$\mathcal{F}_S(\mathbf{x}, \boldsymbol{\Theta}_S^*) = n^{-1} \sum_{\mathcal{A} \in S} \mathbf{f}_A(\mathbf{x}, \theta_A)$$

NES

$$\min_S \mathcal{L}_{\text{val}}(\mathcal{F}_S(\mathbf{x}, \boldsymbol{\Theta}_S^*)) \tag{1}$$

$$\text{s.t. } \forall \theta_A^* \in \boldsymbol{\Theta}_S^* \quad \theta_A^* = \arg\min_{\theta_A} \mathcal{L}_{\text{train}}(\mathbf{f}_A(\mathbf{x}, \theta_A)). \tag{2}$$

Challenges

1. The enormous number of candidate architectures in the NAS search space (e.g., $\sim 10^{25}$ in the DARTS search space)
2. There are $\sim m^n$ different ensembles given m diverse architectures

Model training of supernet

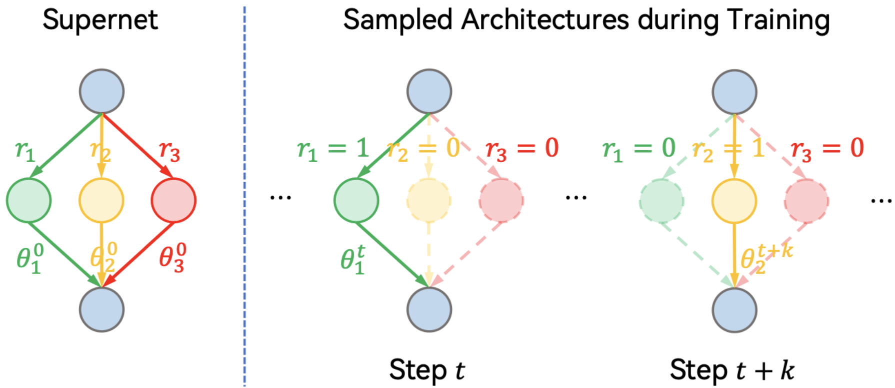

Figure: Model training of supernet. At each step only one architecture is uniformly sampled to update its parameters.

Distribution of architectures

Single-model performance

$\mathcal{D}$ - validation dataset, $p(\mathcal{A})$ and $p(\mathcal{A}|\mathcal{D})$ - prior and posterior distributions of a candidate architecture, $p(\mathcal{D}|\mathcal{A})$ - likelihood

$$p(\mathcal{A}|\mathcal{D}) = p(\mathcal{D}|\mathcal{A})p(\mathcal{A})/p(\mathcal{D}) \propto p(\mathcal{D}|\mathcal{A})$$

Diversity

$\mathcal{L}(\mathbf{f})$ - γ-Lipschitz continuous loss function.

$$\|\mathbf{f}_{\mathcal{A}_1} - \mathbf{f}_{\mathcal{A}_2}\|_2 \geq \gamma^{-1}|\mathcal{L}(\mathbf{f}_{\mathcal{A}_1}) - \mathcal{L}(\mathbf{f}_{\mathcal{A}_2})|$$

$p(\mathcal{A}|\mathcal{D})$ can estimate diversity using $|p(\mathcal{A}_1|\mathcal{D}) - p(\mathcal{A}_2|\mathcal{D})|$

Posterior approximation

Variational distribution $p_\alpha(\mathcal{A})$ approximates $p(\mathcal{A}|\mathcal{D})$:

$$\max_\alpha \mathbb{E}_{\mathcal{A}\sim p_\alpha(\mathcal{A})}[\log p(\mathcal{D}|\mathcal{A})] - \mathbf{KL}[p_\alpha(\mathcal{A})\|p(\mathcal{A})] \tag{3}$$

NES via Bayesian sampling

Algorithm 1 NES via Bayesian Sampling (NESBS)

Input: Iterations T, ensemble size n, a supernet
Train the supernet to get its tuned parameters $\boldsymbol{\theta}^*$
Obtain the posterior distribution $p_{\boldsymbol{\alpha}^*}(\mathcal{A})$ with (3)
for iteration $t = 1, \ldots, T$ **do**
    Sample S_t of size n via Algorithm 2 or 3
    Evaluate estimated $\mathcal{L}_{\text{val}}(\mathcal{F}_{S_t}(\boldsymbol{x}, \boldsymbol{\Theta}^*_{S_t}))$ given $\boldsymbol{\theta}^*$
end for
Select optimum $S^* = \arg\min_{S_t} \mathcal{L}_{\text{val}}(\mathcal{F}_{S_t}(\boldsymbol{x}, \boldsymbol{\Theta}^*_{S_t}))$

Bayesian sampling

Monte-Carlo Sampling (MC)

Sampling a set of architectures from posterior distribution

Algorithm 2 MC Sampling

1: **Input:** Ensemble size n, set $S = \emptyset$, posterior $p_{\boldsymbol{\alpha}^*}(\mathcal{A})$
2: **for** iteration $i = 1, \dots, n$ **do**
3: Sample $\mathcal{A}_i \sim p_{\boldsymbol{\alpha}^*}(\mathcal{A})$
4: $S \leftarrow S \cup \{\mathcal{A}_i\}$
5: **end for**
6: **Output:** S

Bayesian sampling

SVGD with Regularized Diversity (RD)

Adding a term representing the diversity

$$q^* = \arg\min_{q \in \mathcal{Q}} \{\mathbf{KL}(q \| p)\} + n\delta \mathbb{E}_{\mathbf{x},\mathbf{x}' \sim q}[k(\mathbf{x}, \mathbf{x}')]$$

Algorithm 3 SVGD-RD

1: **Input:** Diversity coefficient δ, ensemble size n, iterations L, initial particles $\{\boldsymbol{x}_i^{(0)}\}_{i=1}^n$, posterior $p_{\boldsymbol{\alpha}^*}(\mathcal{A})$, kernel $k(\boldsymbol{x}, \boldsymbol{x}')$, step size $\{\epsilon_l\}_{l=1}^L$

2: **for** iteration $l = 0, \dots, L-1$ **do**

3: Evaluate updates $\widehat{\boldsymbol{\phi}}_l^*(\boldsymbol{x}) = \frac{1}{n} \sum_{j=1}^n \nabla_{\boldsymbol{x}_j^{(l)}} k(\boldsymbol{x}_j^{(l)}, \boldsymbol{x}) - \delta \nabla_{\boldsymbol{x}} k(\boldsymbol{x}_j^{(l)}, \boldsymbol{x}) + k(\boldsymbol{x}_j^{(l)}, \boldsymbol{x}) \nabla_{\boldsymbol{x}_j^{(l)}} \log p_{\boldsymbol{\alpha}^*}$

4: Update particles $\boldsymbol{x}_i^{(l+1)} \leftarrow \boldsymbol{x}_i^{(l)} + \epsilon_l \, \widehat{\boldsymbol{\phi}}_l^*(\boldsymbol{x}_i^{(l)})$

5: **end for**

6: **Output:** $S = \{\mathcal{A}_i\}_{i=1}^n$ derived based on $\{\boldsymbol{x}_i^{(L)}\}_{i=1}^n$

SVGD-RD

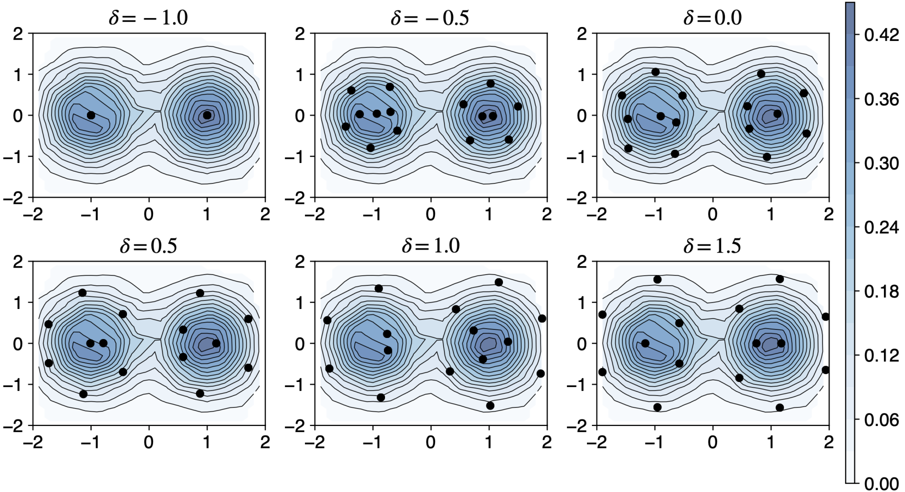

Figure: Impact of δ in SVGD-RD.

Search in NAS-BENCH-201

Architecture(s)	**Test Error (%)**			**Search Cost** (GPU Hours)
	CIFAR-10	**CIFAR-100**	**ImageNet-16-200**	
		Manual design		
ResNet† [He et al., 2016]	6.03	29.14	56.37	-
		NAS algorithms		
ENAS† [Pham et al., 2018]	45.70±0.00	84.39±0.00	83.68±0.00	3.7
DARTS† (2nd) [Liu et al., 2019]	45.70±0.00	84.39±0.00	83.68±0.00	8.3
GDAS† [Dong and Yang, 2019a]	6.49±0.13	29.39±0.26	58.16±0.90	8.0
SETN† [Dong and Yang, 2019b]	13.81±4.63	43.13±7.77	68.10 ±4.07	8.6
RSPS† [Li and Talwalkar, 2019]	12.34±1.69	41.67±4.34	68.86±3.88	2.1
		Ensemble (search) algorithms		
DeepEns [Lakshminarayanan et al., 2017]	5.75	25.27	54.70	-
NES-RS [Zaidi et al., 2021]	5.83±0.33	25.58±0.84	54.34±1.67	5.1
		Our ensemble search algorithm		
NESBS (MC Sampling)	5.76±0.25	25.39±0.69	**53.47**±1.75	**1.1**
NESBS (SVGD-RD)	5.92±0.07	**25.00**±0.17	**52.68**±0.35	**1.2**

Figure: Comparison of architectures selected by different NAS and ensemble (search) algorithms, $n = 3$.

Search in the DARTS space

Architecture(s)	Test Error (%)		Params (M)		Search Cost (GPU Days)	Search Method
	C10	C100	C10	C100		
			NAS algorithms			
NASNet-A [Zoph et al., 2018]	2.65	-	3.3	-	2000	RL
AmoebaNet-A [Real et al., 2019]	3.34	18.93	3.2	3.1	3150	evolution
PNAS [Liu et al., 2018]	3.41	19.53	3.2	3.2	225	SMBO
ENAS [Pham et al., 2018]	2.89	19.43	4.6	4.6	0.5	RL
DARTS [Liu et al., 2019]	2.76	17.54	3.3	3.4	1	gradient
GDAS [Dong and Yang, 2019a]	2.93	18.38	3.4	3.4	0.3	gradient
P-DARTS [Chen et al., 2019]	2.50	-	3.4	-	0.3	gradient
DARTS- (avg) [Chu et al., 2020]	2.59	17.51	3.5	3.3	0.4	gradient
SDARTS-ADV [Chen and Hsieh, 2020]	2.61	-	3.3	-	1.3	gradient
			Ensemble (search) algorithms			
MC DropPath (ENAS)	2.88	16.83	$3.8^{\ddagger}$	$3.9^{\ddagger}$	-	-
DeepEns (ENAS)	2.49	15.04	$3.8^{\ddagger}$	$3.9^{\ddagger}$	-	-
DeepEns (DARTS)	2.42	14.56	$3.3^{\ddagger}$	$3.4^{\ddagger}$	-	-
NES-RS$^{\sharp}$ [Zaidi et al., 2021]	2.50	15.24	$3.0^{\ddagger}$	$3.1^{\ddagger}$	0.7	greedy
			Our ensemble search algorithm			
NESBS (MC Sampling)	**2.41**	14.70	$3.8^{\ddagger}$	$3.9^{\ddagger}$	**0.2**	sampling
NESBS (SVGD-RD)	**2.36**	**14.55**	$3.7^{\ddagger}$	$3.8^{\ddagger}$	**0.2**	sampling

Figure: Comparison of different image classifiers on CIFAR-10/100.

Search in the DARTS space

Architecture(s)	Test Error (%)		Params (M)	$+\times$ (M)
	Top-1	**Top-5**		
NAS algorithms				
NASNet-A	26.0	8.4	5.3	564
AmoebaNet-A	25.5	8.0	5.1	555
PNAS	25.8	8.1	5.1	588
DARTS	26.7	8.7	4.7	574
GDAS	26.0	8.5	5.3	581
P-DARTS	24.4	7.4	4.9	557
SDARTS-ADV	25.2	7.8	5.4	594
Ensemble (search) algorithm				
NES-RS	23.4	6.8	3.9	432
Our ensemble search algorithm				
NESBS (MC Sampling)	**22.3**	**6.2**	4.6	522
NESBS (SVGD-RD)	**22.3**	**6.1**	4.9	562

Figure: Comparison of image classifiers on ImageNet, $n = 3$.

Search in the DARTS space

Method	FGSM		PGD-40		CW		AutoAttack	
	Attack (%)	Defense (%)	Attack (%)	Defense (%)	Attack (%)	Defense (%)	Attack (%)	Defense (%)
On CIFAR-10 Dataset								
DeepEns	-	-	-	-	-	-	-	-
↪ RobNet-free	66.62±0.32	85.25±0.39	41.81±0.80	77.48±0.67	5.74±1.41	86.53±0.50	21.35±0.33	45.51±0.15
↪ ENAS	77.85±0.58	87.94±0.21	59.51±1.13	86.57±0.15	31.36±1.20	85.20±0.77	31.71±0.72	50.96±0.07
↪ DARTS	76.79±0.80	88.21±0.14	57.71±1.65	82.02±0.10	26.90±1.37	82.46±0.35	29.97±1.17	49.67±0.14
NES-RS	79.19±1.39	89.32±0.27	65.59±2.11	85.22±0.41	37.20±4.62	86.75±0.88	35.00±1.15	53.80±0.14
NESBS (MC Sampling)	78.75±1.29	89.15±0.08	63.60±1.87	85.35±0.31	**37.71**±1.97	**86.86**±0.66	**36.02**±0.64	**56.90**±0.17
NESBS (SVGD-RD)	79.12±0.61	**89.86**±0.33	65.53±1.56	85.37±0.38	**38.27**±1.27	86.00±1.10	**37.55**±0.68	**57.15**±0.20
On CIFAR-100 Dataset								
DeepEns	-	-	-	-	-	-	-	-
↪ RobNet-free	36.47±0.25	61.39±0.30	18.18±0.47	52.61±0.13	2.36±0.13	69.44±0.04	7.31±0.35	24.56±0.33
↪ ENAS	46.40±0.37	64.94±0.27	28.87±0.27	56.79±0.25	9.60±0.30	69.43±0.44	11.53±0.47	27.01±0.27
↪ DARTS	46.98±0.57	65.38±0.23	28.78±0.74	57.10±0.04	9.73±0.43	70.15±0.29	11.20±0.40	26.86±0.36
NES-RS	47.10±1.46	65.33±0.36	30.68±1.66	58.80±0.80	9.96±1.45	70.24±0.33	12.01±0.93	27.49±0.34
NESBS (MC Sampling)	**50.69**±1.58	**67.63**±0.05	**33.37**±0.42	**60.36**±0.62	**15.64**±2.83	**71.25**±1.27	**13.11**±1.16	**29.87**±1.17
NESBS (SVGD-RD)	**51.47**±0.40	**66.66**±0.13	**35.02**±0.37	**59.96**±0.18	**16.72**±0.61	**69.88**±0.16	**14.62**±0.55	**31.07**±0.33

Figure: Comparison of adversarial defense among different ensemble (search) algorithms on CIFAR-10/100 under white-box adversarial attacks.

Single-model performances and diverse model predictions

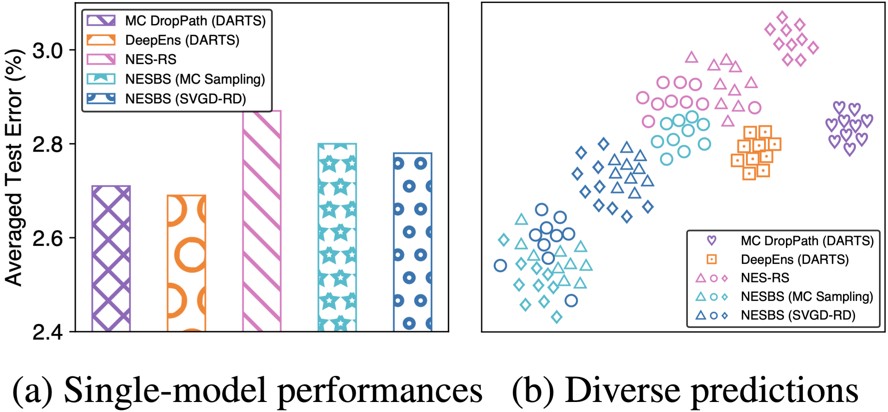

Figure: Qualitative comparison of (a) the single-model performances and (b) the diverse model predictions achieved by different ensemble (search) algorithms with an ensemble size of $n = 3$ on CIFAR-10.

Single-model performances and diverse model predictions

Method	C10		C100	
	ATE	PPD	ATE	PPD
MC DropPath (DARTS)	2.71	0.39	16.68	2.63
DeepEns (DARTS)	**2.69**	2.08	**16.18**	12.45
NES-RS	2.87	2.29	17.20	**14.14**
NESBS (MC Sampling)	2.80	**2.57**	16.70	13.84
NESBS (SVGD-RD)	2.78	2.27	16.50	13.16

Figure: Quantitative comparison of the single-model performances and the diversity of model predictions achieved by different ensemble (search) algorithms with an ensemble size of 3 on CIFAR-10/100.

Conclusion

- Novel neural ensemble search algorithms
- Effectively and efficiently selects well-performing NNE with diverse architectures from a NAS search space
- Achieves improved performances while preserving a comparable search cost
- Boosted search effectiveness and efficiency compared to DeepEns and NES-RS

Literature

1. **Main article** Neural Ensemble Search via Bayesian Sampling.