L²NAS: Learning to Optimize Neural Architectures via Continuous-Action Reinforcement Learning

Keith G. Mills[1], Fred X. Han[2], Mohammad Salameh[2], Seyed Saeed Changiz Rezaei[2], Linglong Kong[1], Wei Lu[2], Shuo Lian[3], Shangling Jui[3] and Di Niu[1]

[1]University of Alberta [2]Huawei Technologies Canada Co., Ltd. [3]Huawei Kirin Solution, Shanghai, China

Neural Architecture Search (NAS) has achieved remarkable results in deep neural network design.

Recent NAS literature has popularized the use of differentiable search methods, which apply fixed updates to architecture hyperparameters.

However, concerns have been raised about the suitability of gradient descent-based search methods in the NAS context.

We propose L²NAS, an algorithm for NAS that learns to update architecture hyperparameters based on the distribution of high-performing architectures in the search history.

Experiments show that L²NAS achieves state-of-the-arts results on several search spaces like NAS-Bench-201 and Once-for-All.

Moreover, L²NAS policies are generalizable and can be transferred across different training datasets, minimizing per-dataset fine-tuning.

PROBLEM DEFINITION

Formulate NAS as a *black-box* optimization:

- Let χ be the search space of all valid architectures.
- Let x be an architecture in χ.
- Let $S(.)$ be a real-valued evaluation function.
- Goal: Optimize $S(x)$.

Relax discrete x to continuous domain for optimization:

- Let α be a continuous representation of x,
- Let *Discretize* be a many-to-one mapping, $x = Discretize(\alpha)$.
- Result: We can optimize $S(Discretize(\alpha))$ by querying the discrete architecture but optimizing the update rule for x in a continuous domain with a gradient-based approach.

This problem definition is robust and can generalize to a variety of different search spaces.

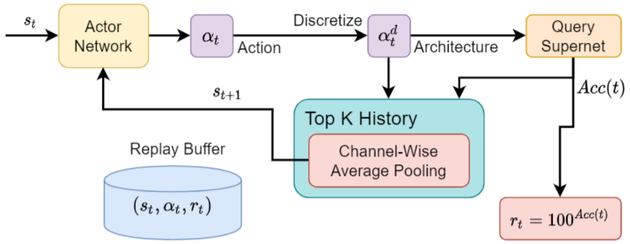

Figure 1: A High-Level Illustration of L²NAS.

OUR METHOD

L²NAS is based on Deep Deterministic Policy Gradient (DDPG), a continuous-action RL algorithm based in the actor critic framework.

Specifically, at step t, the actor network μ will take the current state s_t as input and produce a continuous architecture representation α_t as an action.

$$\alpha_t = \mu(s_t) + z_t,$$

This action is then discretized into a single architecture, which is then evaluated by the environment to calculate the reward r_t.

The environment also keeps track of the top-K best architectures observed, ranked by accuracy. After checking if the new architecture belongs in the top-K and adding it if so, the next state is computed as the channel-wise average of the top-K history. That is, s_{t+1} is the sample distribution of high-performance architectures.

An experiential replay buffer stores state-action-reward tuples for later use by the actor and critic. The critic network Q uses the *check* loss function from quantile regression to learn the τ-th quantile (e.g., 0.9, 0.95) distribution of high-performance architectures:

$$\rho_\tau(x) = x(\tau - \mathbf{1}(x \leq 0)), \quad \mathcal{L}_{\text{Critic}} = \frac{1}{|B_R|} \sum_{i \in B_R} \rho_\tau(r_i - Q(\alpha_i))$$

The actor network is updated based on the teachings of the critic, completing the step,

$$\mathcal{L}_{\text{Actor}} = \frac{1}{|B_R|} \sum_{i \in B_R} Q(\mu(s_i)).$$

BENCHMARK EXPERIMENTAL RESULTS

We first test L²NAS on NAS-Bench-201:

- 15,625 different architectures.
- 3 datasets: CIFAR-10, CIFAR-100 and ImageNet-16-120; a downsampled subset of ImageNet.

On average, our method:

- Obtains state-of-the-art, consistent results on CIFAR-10 and ImageNet-16-120.
- Is the only scheme to obtain an average test accuracy greater than 47% on ImageNet-16-120.
- Only needs to query a small fraction (<1000 architectures) of the dataset.

	CIFAR-10		CIFAR-100		ImageNet-16-120	
Method	**Valid [%]**	**Test [%]**	**Valid [%]**	**Test [%]**	**Valid [%]**	**Test [%]**
DARTS [22]	39.77 ± 0.00	54.30 ± 0.00	15.03 ± 0.00	15.61 ± 0.00	16.43 ± 0.00	16.32 ± 0.00
ENAS [26]	37.51 ± 3.19	53.89 ± 0.58	13.37 ± 2.35	13.96 ± 2.33	15.06 ± 1.95	14.84 ± 2.10
GDAS [11]	89.89 ± 0.08	93.61 ± 0.09	71.34 ± 0.04	70.70 ± 0.30	41.59 ± 1.33	41.71 ± 0.98
GAEA [19]	–	94.10 ± 0.29	–	**73.43 ± 0.13**	–	46.36 ± 0.00
RS [12]	90.93 ± 0.36	93.70 ± 0.36	70.93 ± 1.09	71.04 ± 1.07	44.45 ± 1.10	44.57 ± 1.25
REA [12]	91.19 ± 0.31	93.92 ± 0.30	71.81 ± 1.12	71.84 ± 0.99	45.15 ± 0.89	45.54 ± 1.03
REINFORCE [12]	91.09 ± 0.37	93.85 ± 0.37	71.61 ± 1.12	71.71 ± 1.09	45.05 ± 1.02	45.24 ± 1.18
BOHB [12]	90.82 ± 0.53	93.61 ± 0.52	70.74 ± 1.29	70.85 ± 1.28	44.26 ± 1.36	44.42 ± 1.49
arch2vec-RL [39]	91.32 ± 0.42	94.12 ± 0.42	73.12 ± 0.72	73.15 ± 0.78	46.22 ± 0.30	46.16 ± 0.38
arch2vec-BO	91.41 ± 0.22	94.18 ± 0.24	**73.35 ± 0.32**	73.37 ± 0.30	46.34 ± 0.18	46.27 ± 0.37
L²NAS-500	91.36 ± 0.19	94.11 ± 0.16	72.47 ± 0.74	72.69 ± 0.58	46.23 ± 0.28	46.74 ± 0.39
L²NAS-1k	**91.47 ± 0.15**	**94.28 ± 0.08**	73.02 ± 0.52	73.09 ± 0.35	**46.58 ± 0.08**	**47.03 ± 0.27**
True Optimal	*91.61*	*94.37*	*73.49*	*73.51*	*46.77*	*47.31*

We also test L²NAS on ImageNet using Once-for-All (OFA), based on MobileNetV3.

Two sizes of models:

- OFA
- OFA_{Large}

L²NAS finds better architectures in both settings.

Table 3: Comparison of L²NAS with other state-of-the-art architectures on ImageNet.

Architecture	Top-1 Acc. [%]	Top-5 Acc. [%]	MACs [M]
PC-SDARTS	75.7	92.6	–
P-SDARTS	75.8	92.8	–
PC-GAEA	76.0	92.7	–
MBv2	74.7	–	585
ProxylessNAS	75.1	92.5	320
MBv3-L 0.75	73.3	–	155
MBv3-L 1.0	75.2	–	219
EfficientNet-B0	77.1	93.9	390
EfficientNet-B1	79.1	94.4	700
EfficientNet-B2	80.1	94.9	1000
Cream-S	77.6	93.3	287
Cream-M	79.2	94.2	481
Cream-L	80.0	94.7	604
OFA	76.0	–	230
OFA_{Large}	79.0	94.5	595
L²NAS	**77.4**	**93.4**	**467**
L²NAS_{Large}	**79.3**	**94.6**	**618**

TRANSFERABILITY

We also perform a study on *transferability,*

- Pre-train search agent on one dataset
- Perform search on a completely different dataset.
- Different accuracy range for each dataset.
- Implement a special reward function to normalize agent performance.

$$r_t = \frac{100^{Acc(\alpha_t^d)/Acc(Env)}}{100} - 1,$$

Where $Acc(Env)$ is the maximum accuracy observed for on either the source or target dataset domains.

- Source domain: CIFAR-10.
- Target domain: CIFAR-100 and ImageNet-32-120.
- Use supernet from the DARTS search space.

Table 5: Transferability results for L²NAS for CIFAR-100 and ImageNet. 'Direct' means directly searching on CIFAR-100 and ImageNet using the procedure in Section 4.2.

CIFAR-100	Test Acc. [%]	Params [M]	GPU days
DARTS 1st	82.37	3.3	1.5
DARTS 2nd	82.65	3.3	4.0
L²NAS-Direct	82.24 ± 0.19	3.5	1.0
L²NAS-Transfer	82.97 ± 0.29	4.0	0.1
ImageNet	**Top-1/Top-5 [%]**	**Params [M]**	**Search Cost**
DARTS 2nd	73.1/91.3	4.7	4.0
L²NAS-Direct	74.8/92.2	4.9	1.0
L²NAS-Transfer	75.4/92.5	5.4	0.1

Results:

- L²NAS search policies that are pre-trained and transferred outperform DARTS on CIFAR-100.
- They also outperform L²NAS architectures that are *directly* searched on CIFAR-100 and ImageNet, e.g., search without transferability.
- Transferability approach greatly reduces the search time on the target domain.
- Less time spent tuning hyperparameters on the target dataset domain saves resources.