DATA: Differentiable ArchiTecture Approximation

33nd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada

Jianlong Chang, Xinbang Zhang, Yiwen Guo, Gaofeng Meng, Zhouchen Lin, Shiming Xiang, Chunhong Pan

Differentiable Architecture Search

Intrinsically, the goal in NAS is to find a graph that minimizes the validation loss, where the network weights associated with the architecture α∗are obtained by minimizing the training loss.

$$\min_{\alpha \in \mathcal{A}} \mathcal{L}_{val}(\mathcal{N}(\alpha, w^*)), \quad s.t.\ w^* = \arg\min_{w} \mathcal{L}_{train}(\mathcal{N}(\alpha^*, w))$$

This implies that the essence of NAS is to solve a bi-level optimization problem, which is hard to optimize because of the nested relationship between architecture parameters and network weights. To handle this issue, we parameterize architectures with binary codes, and devote to jointly learning architectures and network weights in a differentiable way.

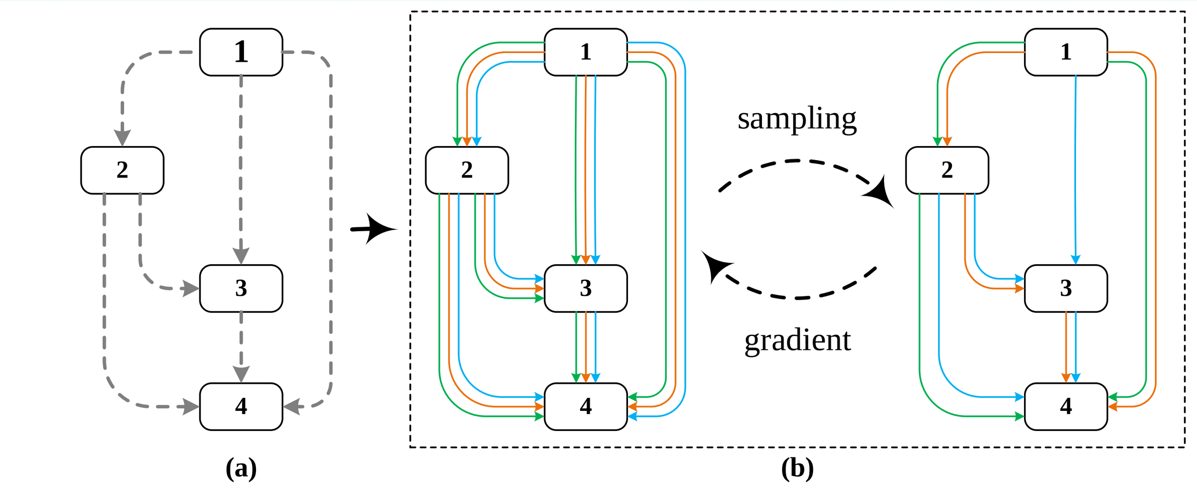
Parameterizing Architectures with Binary Codes

The function in the edge *(i,j)* can be decomposed into a superposition of primitive operations, *i.e.*,

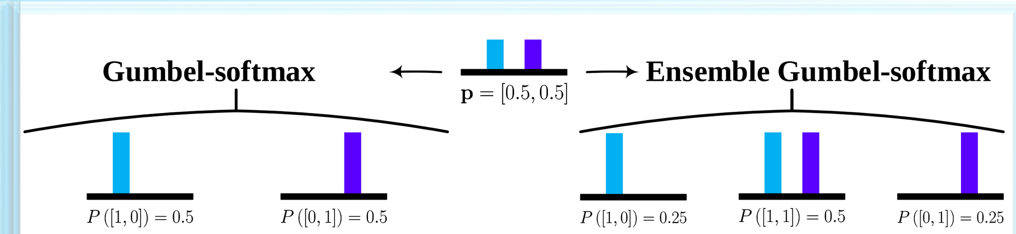
$$o^{(i,j)}(x^{(i)}) = \sum_{k=1}^{K} A_k^{(i,j)} \cdot o_k(x^{(i)}), \quad s.t.\ A_k^{(i,j)} \in \{0,1\},\ 1 \le k \le K,$$

Benefiting from the uniqueness property of our architecture code ***A***, the task of learning an architecture can therefore be converted to learning the optimal binary code ***A***.

From Probability Vectors to Binary Codes - Ensemble Gumbel-Softmax

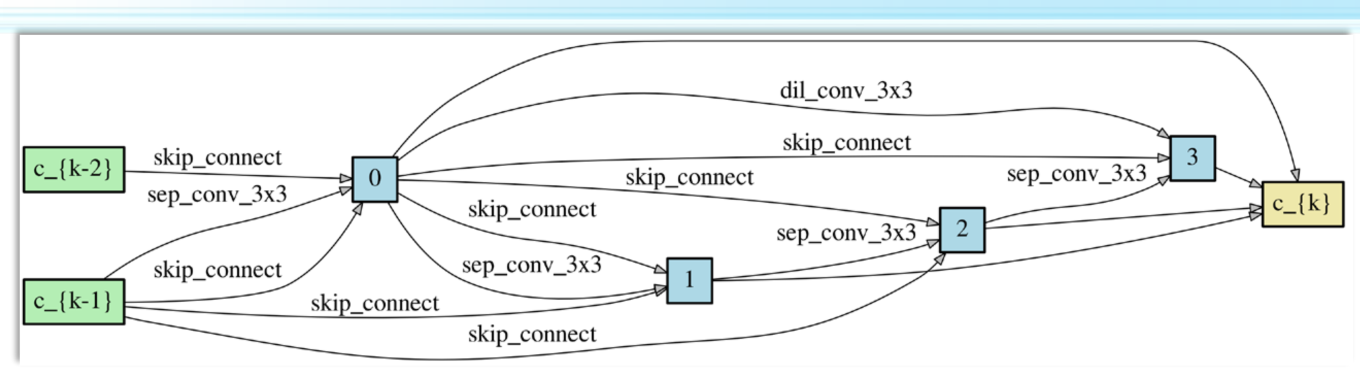


We introduce a binary function f(·) to approach the optimal binary codes with probability vectors, which can be easily obtained in deep models. The function f(·) is formulated with the ensemble Gumbel-softmax. During the forward propagation, with three candidate primitive operations (*i.e.*, green, orange and cyan lines), the binary function f(·) is employed to generate a network in a differentiable manner. During the backward propagation, the standard back-propagation algorithm is utilized to simultaneously calculate the gradients of the both architecture parameters and network weights.

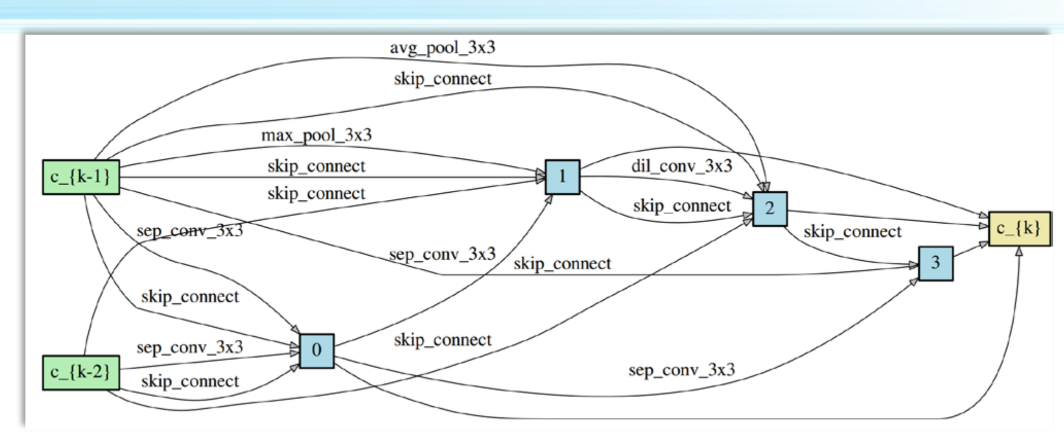


For a probability vector p=[0.5,0.5], Gumbel-Softmax solely pertains to sample only two binary codes with the same probability, *i.e.*, P([1,0])=P([0,1])=0.5. The ensemble Gumbel-Softmax is capable of sampling more binary codes, *i.e.*, [1,0], [1,1] and [0,1]. Furthermore, the probabilities of sampling these binary codes are logical. It is intuitive that the probability of sampling [1,1] is larger than the probabilities of sampling the others since the probabilities in p=[0.5,0.5] are equal.

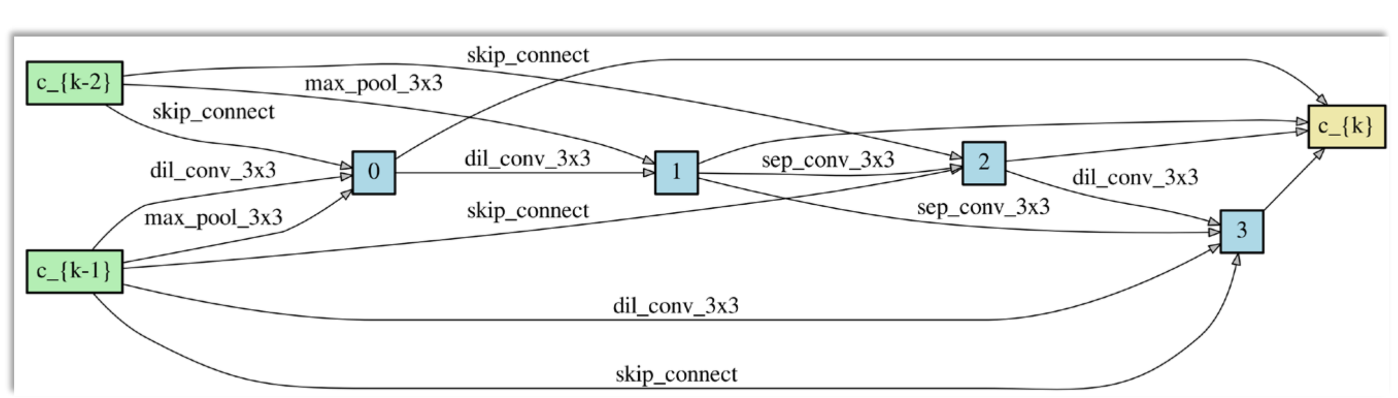
Architectures on CIFAR-10



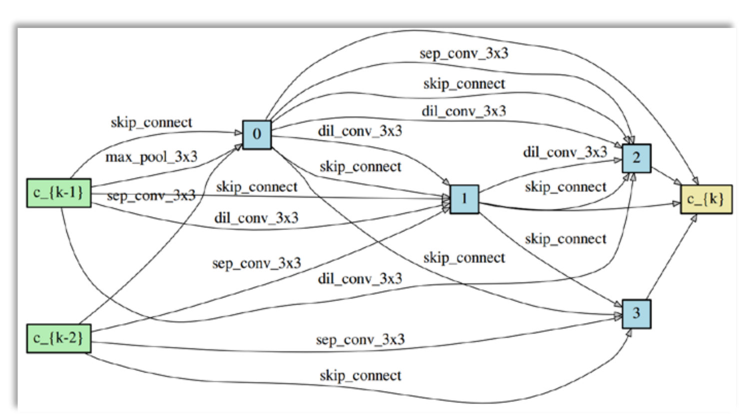
M=4 normal cell



M=7 normal cell



M=4 reduction cell



M=7 reduction cell

Experiments

Table 1 gives the searched architectures and classification results on CIFAR-10, which shows that DATA achieves comparable results with the state-of-the-art with less computation resources.

Table 2 indicates that the cell searched on CIFAR-10 can be smoothly employed to deal with the large-scale classification task.

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10 (lower test error is better)

Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	Ops	Search
DenseNet-BC [22]	3.46	25.6	-	-	manual
PNAS [31]	3.41	3.2	225	8	SMBO
Hierarchical evolution [33]	3.75	15.7	300	6	evolution
AmoebaNet-A [44]	3.34	3.2	3150	19	evolution
AmoebaNet-B + cutout [44]	**2.55**	2.8	3150	19	evolution
NASNet-A + cutout [60]	2.65	3.3	2000	13	RL
ENAS + cutout [42]	2.89	4.6	**0.5**	6	RL
DARTS (1-th order) + cutout [34]	3.00	3.3	1.5	7	gradient-based
DARTS (2-th order) + cutout [34]	**2.76**	3.3	4	7	gradient-based
SNAS + mild + cutout [53]	2.98	2.9	**1.5**	-	gradient-based
SNAS + moderate + cutout [53]	2.85	2.8	**1.5**	-	gradient-based
SNAS + aggressive + cutout [53]	3.10	2.3	**1.5**	-	gradient-based
Random search baseline + cutout	3.29	3.2	4	7	random
DATA ($M=4$) + cutout	2.70	3.2	**1**	7	gradient-based
DATA ($M=7$) + cutout	**2.59**	3.4	**1**	7	gradient-based

Table 2: Comparison with classifiers on ImageNet in the mobile setting (lower test error is better).

Architecture	Test Error (%) Top 1	Test Error (%) Top 5	Params (M)	FLOPs (M)	Search Cost (GPU days)	Search
Inception-v1 [48]	30.2	10.1	6.6	1448	-	manual
MobileNet [20]	29.4	10.5	4.2	569	-	manual
ShuffleNet-v2 2× [36]	25.1	-	∼5	591	-	manual
PNAS [31]	25.8	8.1	5.1	588	∼225	SMBO
AmoebaNet-A [44]	25.5	8.0	5.1	555	3150	evolution
AmoebaNet-B [44]	26.0	8.5	5.3	555	3150	evolution
AmoebaNet-C [44]	**24.3**	**7.6**	6.4	570	3150	evolution
NASNet-A [60]	26.0	8.4	5.3	564	2000	RL
NASNet-B [60]	27.2	8.7	5.3	488	2000	RL
NASNet-C [60]	27.5	9.0	4.9	558	2000	RL
DARTS (on CIFAR-10) [34]	26.7	8.7	4.7	574	4	gradient-based
SNAS (mild constraint) [53]	27.3	9.2	4.3	522	1.5	gradient-based
GDAS [18]	26.0	8.5	5.3	581	**0.21**	gradient-based
DATA ($M=4$)	25.5	8.3	4.9	568	**1**	gradient-based
DATA ($M=7$)	**24.9**	**8.0**	5.0	588	**1**	gradient-based

Table 3 signifies that DATA also is in a position to search recurrent architectures effectively.

Table 4 demonstrates that the transferability is also retentive on recurrent architectures.

Table 3: Comparison with state-of-the-art language models on PTB (lower perplexity is better).

Architecture	Perplexity valid	Perplexity test	Params (M)	Search Cost (GPU days)	Ops	Search
Variational RHN [57]	67.9	65.4	23	-	-	manual
LSTM [40]	60.7	58.8	24	-	-	manual
LSTM + skip connections [38]	60.9	58.3	24	-	-	manual
LSTM + 15 softmax experts [54]	58.1	56.0	22	-	-	manual
DARTS (first order) [34]	60.2	57.6	23	0.5	4	gradient-based
DARTS (second order) [34]	58.1	55.7	23	1	4	gradient-based
ENAS [42]	68.3	63.1	24	0.5	4	RL
Random search baseline	61.8	59.4	23	2	4	random
DATA ($M=4$)	58.3	56.2	23	0.5	4	gradient-based
DATA ($M=7$)	**57.1**	**55.3**	23	0.5	4	gradient-based

Table 4: Comparison with state-of-the-art language models on WT2 (lower perplexity rate is better).

Architecture	Perplexity valid	Perplexity test	Params (M)	Search Cost (GPU days)	Search
LSTM + augmented loss [23]	91.5	87.0	28	-	manual
LSTM + cache pointer [16]	-	68.9	-	-	manual
LSTM [40]	69.1	66.0	33	-	manual
LSTM + skip connections [38]	69.1	65.9	24	-	manual
LSTM + 15 softmax experts [54]	66.0	63.3	33	-	manual
DARTS (searched on PTB) [34]	69.5	66.9	33	1	gradient-based
ENAS (searched on PTB) [42]	72.4	70.4	33	0.5	RL
DATA ($M=4$)	67.3	64.6	33	1	gradient-based
DATA ($M=7$)	**66.5**	**64.2**	33	1	gradient-based

Ablation study

Table 5 means that larger M indicates higher performance, while more parameters will be introduced as M increases.

Table 6 verifies that DATA have more prominent superiority on more complex tasks, not just toy tasks on the tiny datasets, because of a large search space that is proportional to the sampling time M.

Table 5: Sensitivity to number of sampling times on CIFAR-10 (lower test error is better).

Sampling Times (M)	1	2	3	4	5	6	7	8	9
Test Error (%)	2.94	2.95	2.78	2.70	2.72	2.60	2.59	2.50	**2.45**
Params (M)	**2.54**	2.68	2.71	3.24	3.41	3.49	3.44	3.79	3.97

Table 6: Semantic segmentation on the PASCAL VOC-2012 (higher mIOU is better).

Architecture	NASNet [58]	DARTS [32]	DATA ($M=1$)	DATA ($M=4$)	DATA ($M=7$)
mIOU(%)	73.7	73.2	73.4	74.1	**75.6**
Params (M)	12.4	11.8	**10.8**	11.7	12.7

Table 7 shows the stds of DATA and the variances of DATA with different sampling time M.

Table 8 reports the validation errors at the end of search and after architecture derivation without fine-tuning.

Figure (a) illustrates the search progresses of different models. Figure (b) and (c) study the influence of initializations and the contribution of ensemble Gumbel-Softmax.

Table 7: Number of operations on CIFAR-10.

Model	Error (%)	Params (M)
DARTS(k=1)	3.00 ± 0.14	3.30
DARTS(k=2)	3.10 ± 0.12	4.00
DARTS(k=3)	2.95 ± 0.13	5.20
SNAS	2.85 ± 0.02	2.80
DATA(M=1)	2.94 ± 0.09	**2.54**
DATA(M=4)	2.70 ± 0.10	3.24
DATA(M=7)	**2.59** ± **0.09**	3.44

Table 8: Validation error on CIFAR-10.

Model	Search	Child	Gap
DARTS	12.33	45.34	33.01
SNAS	11.46	9.33	2.13
DATA (M=7)	**11.08**	**9.21**	**1.87**

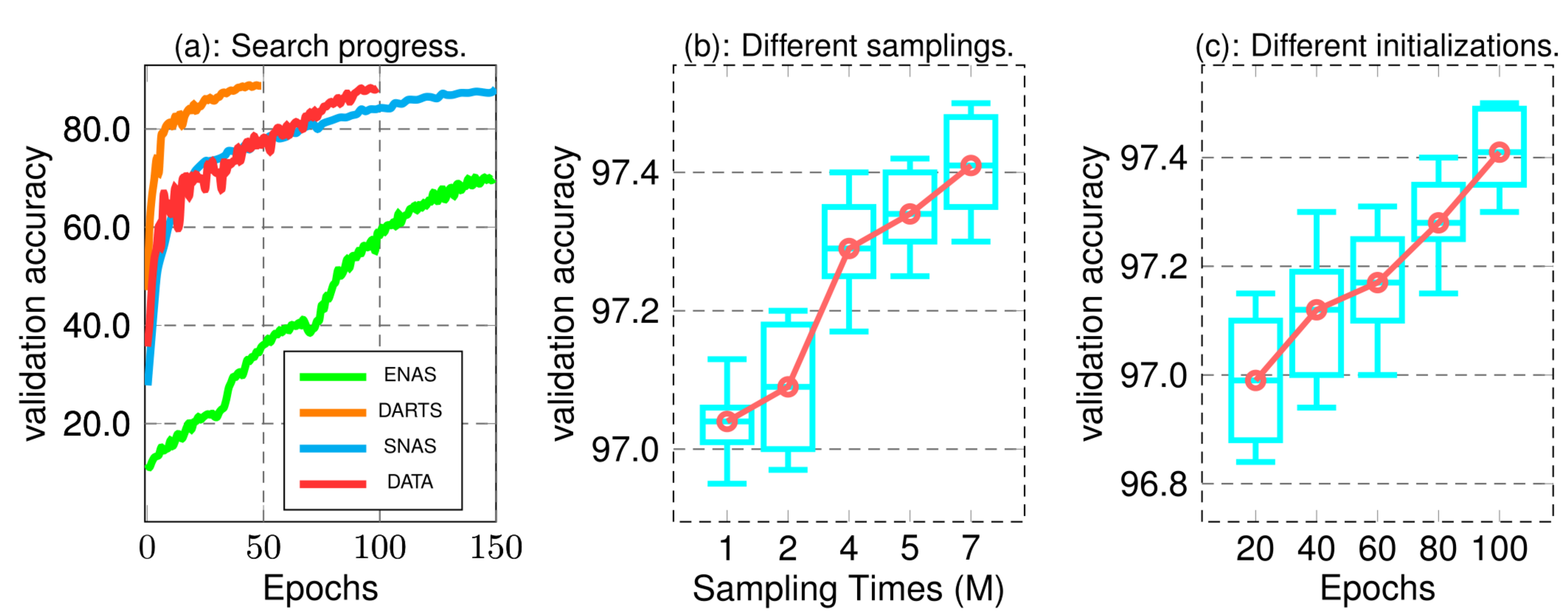


Figure 3: Ablation study.