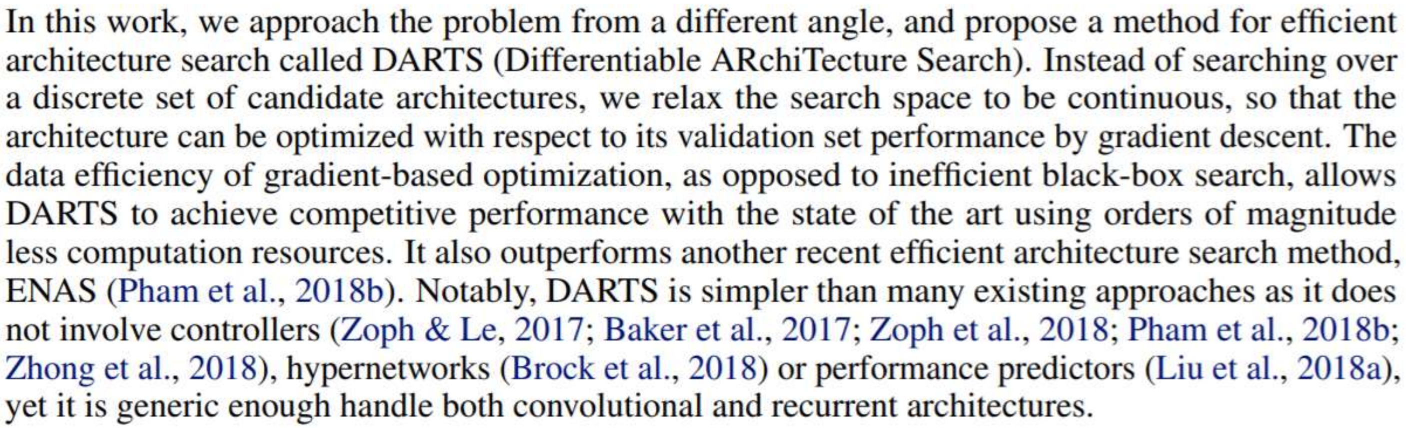
DARTS: DIFFERENTIABLE ARCHITECTURE SEARCH

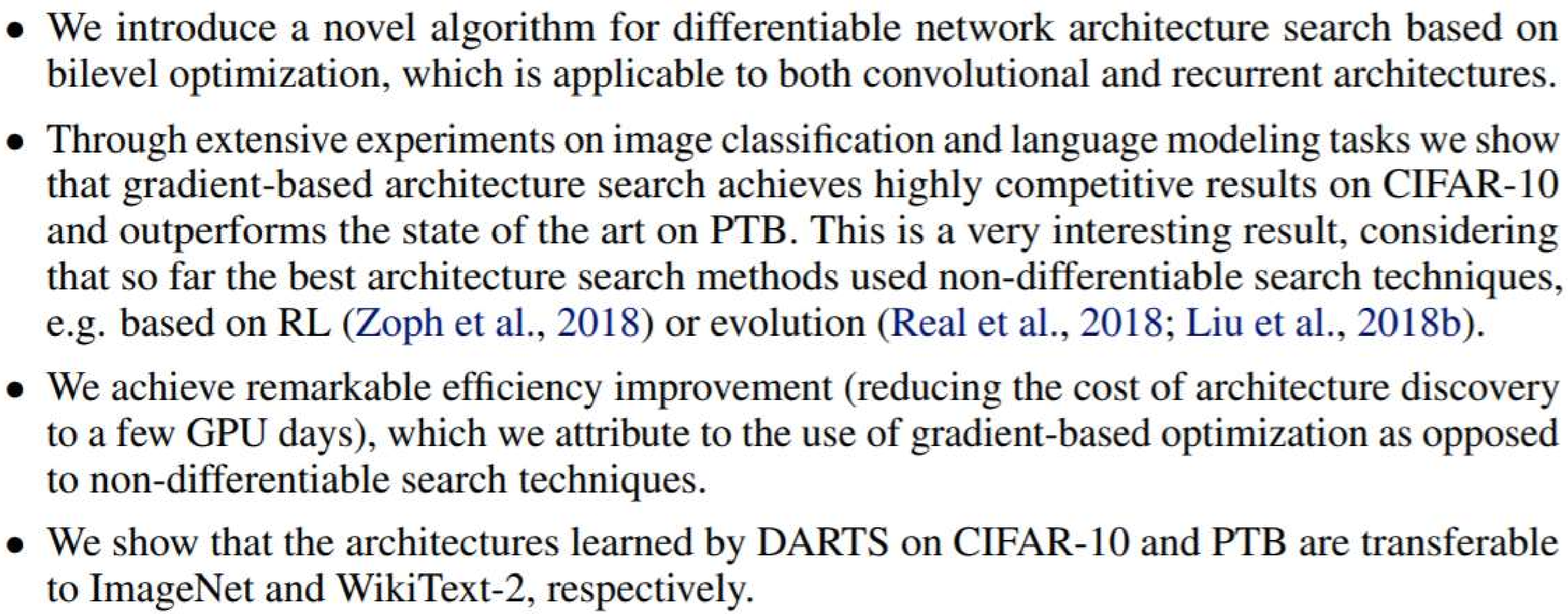


In this work, we approach the problem from a different angle, and propose a method for efficient architecture search called DARTS (Differentiable ARchiTecture Search). Instead of searching over a discrete set of candidate architectures, we relax the search space to be continuous, so that the architecture can be optimized with respect to its validation set performance by gradient descent. The data efficiency of gradient-based optimization, as opposed to inefficient black-box search, allows DARTS to achieve competitive performance with the state of the art using orders of magnitude less computation resources. It also outperforms another recent efficient architecture search method, ENAS (Pham et al., 2018b). Notably, DARTS is simpler than many existing approaches as it does not involve controllers (Zoph & Le, 2017; Baker et al., 2017; Zoph et al., 2018; Pham et al., 2018b; Zhong et al., 2018), hypernetworks (Brock et al., 2018) or performance predictors (Liu et al., 2018a), yet it is generic enough handle both convolutional and recurrent architectures.

特点：不是在一组离散的候选架构上进行搜索，而是将搜索空间松弛为连续的，以便通过梯度下降对架构的验证集性能进行优化。

其他方法：ENAS

Advantages：



- We introduce a novel algorithm for differentiable network architecture search based on bilevel optimization, which is applicable to both convolutional and recurrent architectures.
- Through extensive experiments on image classification and language modeling tasks we show that gradient-based architecture search achieves highly competitive results on CIFAR-10 and outperforms the state of the art on PTB. This is a very interesting result, considering that so far the best architecture search methods used non-differentiable search techniques, e.g. based on RL (Zoph et al., 2018) or evolution (Real et al., 2018; Liu et al., 2018b).
- We achieve remarkable efficiency improvement (reducing the cost of architecture discovery to a few GPU days), which we attribute to the use of gradient-based optimization as opposed to non-differentiable search techniques.
- We show that the architectures learned by DARTS on CIFAR-10 and PTB are transferable to ImageNet and WikiText-2, respectively.

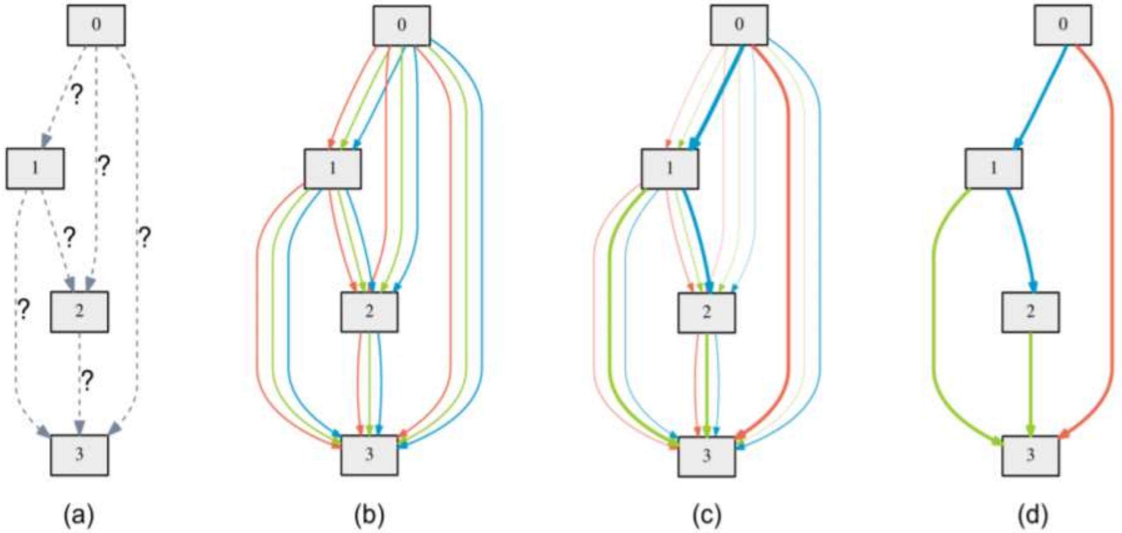


Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

Let $\mathcal{O}$ be a set of candidate operations (e.g., convolution, max pooling, *zero*) where each operation represents some function $o(\cdot)$ to be applied to $x^{(i)}$. To make the search space continuous, we relax the categorical choice of a particular operation to a softmax over all possible operations:

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x) \tag{2}$$

where the operation mixing weights for a pair of nodes (i, j) are parameterized by a vector $\alpha^{(i,j)}$ of dimension $|\mathcal{O}|$. The task of architecture search then reduces to learning a set of continuous variables $\alpha = \{\alpha^{(i,j)}\}$, as illustrated in Fig. 1. At the end of search, a discrete architecture can be obtained by replacing each mixed operation $\bar{o}^{(i,j)}$ with the most likely operation, i.e., $o^{(i,j)} = \text{argmax}_{o \in \mathcal{O}} \ \alpha_o^{(i,j)}$. In the following, we refer to α as the (encoding of the) architecture.

Algorithm 1: DARTS – Differentiable Architecture Search

```
Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge $(i,j)$
while not converged do
    1. Update architecture $\alpha$ by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w,\alpha), \alpha)$
       ($\xi = 0$ if using first-order approximation)
    2. Update weights $w$ by descending $\nabla_w \mathcal{L}_{train}(w,\alpha)$
Derive the final architecture based on the learned $\alpha$.
```

Evaluating the architecture gradient exactly can be prohibitive due to the expensive inner optimization. We therefore propose a simple approximation scheme as follows:

$$\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha) \tag{5}$$

$$\approx \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w,\alpha), \alpha) \tag{6}$$

3 Experiments and Results

Our experiments on CIFAR-10 and PTB consist of two stages, architecture search (Sect. 3.1) and architecture evaluation (Sect. 3.2). In the first stage, we search for the cell architectures using DARTS, and determine the best cells based on their validation performance. In the second stage, we use these cells to construct *larger* architectures, which we train from scratch and report their performance on the test set. We also investigate the transferability of the best cells learned on CIFAR-10 and PTB by evaluating them on ImageNet and WikiText-2 (WT2) respectively.

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10 (lower error rate is better). Note the search cost for DARTS does not include the selection cost (1 GPU day) or the final evaluation cost by training the selected architecture from scratch (1.5 GPU days).

Architecture	**Test Error (%)**	**Params (M)**	**Search Cost (GPU days)**	**#ops**	**Search Method**
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	–	manual
NASNet-A + cutout (Zoph et al., 2018)	2.65	3.3	2000	13	RL
NASNet-A + cutout (Zoph et al., 2018)†	2.83	3.1	2000	13	RL
BlockQNN (Zhong et al., 2018)	3.54	39.8	96	8	RL
AmoebaNet-A (Real et al., 2018)	3.34 ± 0.06	3.2	3150	19	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	19	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	19	evolution
Hierarchical evolution (Liu et al., 2018b)	3.75 ± 0.12	15.7	300	6	evolution
PNAS (Liu et al., 2018a)	3.41 ± 0.09	3.2	225	8	SMBO
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	6	RL
ENAS + cutout (Pham et al., 2018b)*	2.91	4.2	4	6	RL
Random search baseline‡ + cutout	3.29 ± 0.15	3.2	4	7	random
DARTS (first order) + cutout	3.00 ± 0.14	3.3	1.5	7	gradient-based
DARTS (second order) + cutout	2.76 ± 0.09	3.3	4	7	gradient-based

* Obtained by repeating ENAS for 8 times using the code publicly released by the authors. The cell for final evaluation is chosen according to the same selection protocol as for DARTS.

† Obtained by training the corresponding architectures using our setup.

‡ Best architecture among 24 samples according to the validation error after 100 training epochs.

Thanks！