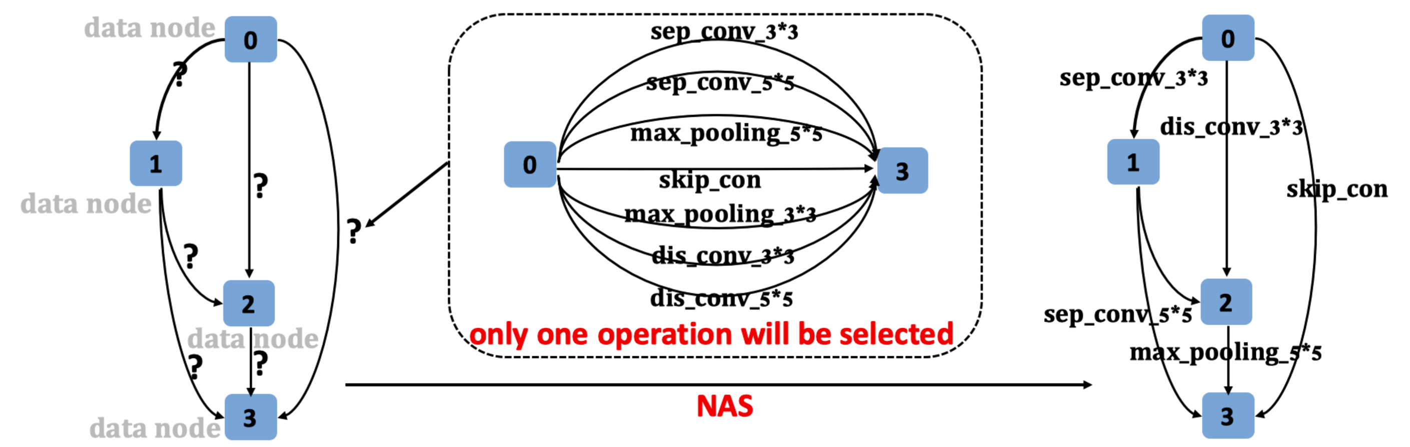
Theory-Inspired Path-Regularized Differential Network Architecture Search

Pan Zhou, Caiming Xiong, Richard Socher, Steven C.H. Hoi
Salesforce Research
{pzhou, cxiong, rsocher, shoi}@salesforce.com

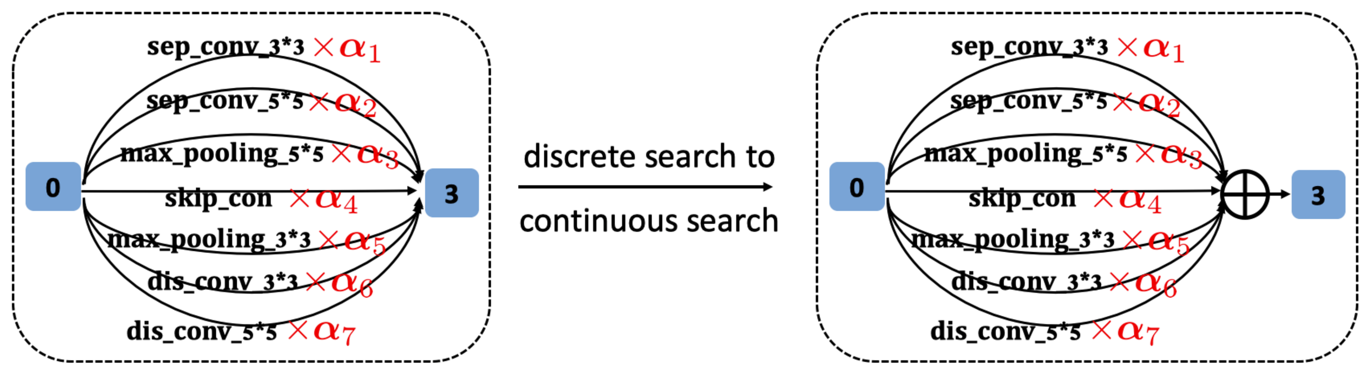
homepage

Problem Setup

What is NAS (network architecture search)? it aims to automatically select a proper operation from an operation set for each edge in a dense graph.



DARTS converts discrete operation selection into continuously weighting a set of operations.

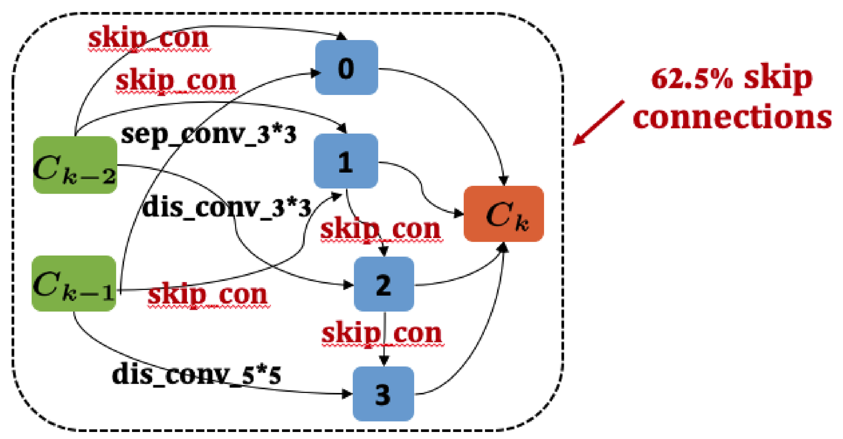


discrete search space: $\{\alpha \mid \alpha_i \in \{0,1\}, \sum_i \alpha_i = 1\}$ hard to optimize, RL or EA

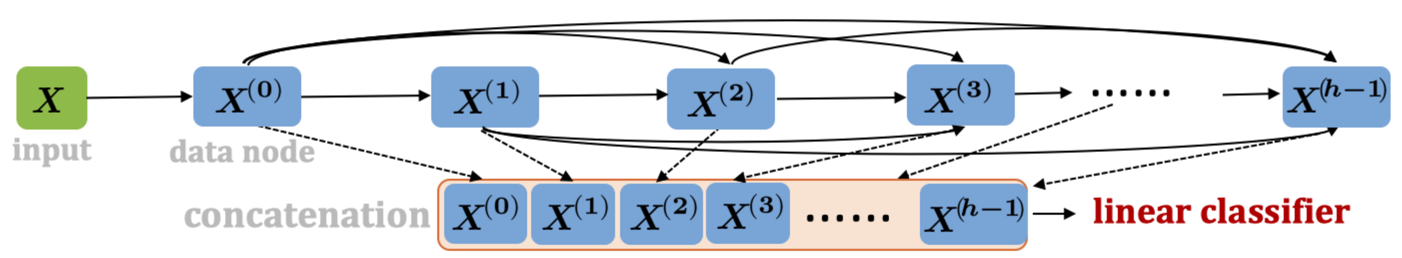
continuous search space: $\{\alpha \mid \alpha_i \in [0,1], \sum_i \alpha_i = 1\}$ easy to optimize, gradient descent

Observations: dominated skip connections in the architectures selected by DARTS

Problem:
- 1) why DARTS prefers to select so many skip connections?
- 2) how to avoid the dominated skip connections?



Formulations of DARTS



Dense directed graph via connecting current node with all previous nodes

$$X^{(l)} = \sum_{s=0}^{l-1} \left[\alpha_{s,1}^{(l)} \text{zero}(X^s) + \alpha_{s,2}^{(l)} \text{skip}(X^s) + \alpha_{s,3}^{(l)} \text{conv}(W_s^{(l)}; X^{(s)})\right] \in \mathbb{R}^{m\times p}$$

where $\alpha_{s,1}^{(l)}, \alpha_{s,2}^{(l)}$ and $\alpha_{s,3}^{(l)}$ respectively denote the weights of zero operation $\text{zero}(X^s)=0$, skip connection $\text{skip}(X^s) = X^s$ and convolution $\text{conv}(W_s^{(l)}; X^{(s)})$.

Prediction by feeding the features of all nodes into a linear classifier

$$u_i = \sum_{s=0}^{h-1} \langle W_s, X_i^{(s)} \rangle \in \mathbb{R}$$

DARTS Model:

$$\min_\alpha F_{\text{val}}(W^*(\alpha), \alpha), \quad \text{s.t.} \quad W^*(\alpha) = \text{argmin}_W F_{\text{train}}(W, \alpha),$$

optimize architecture parameter α on validation data; optimize network parameter W on training data

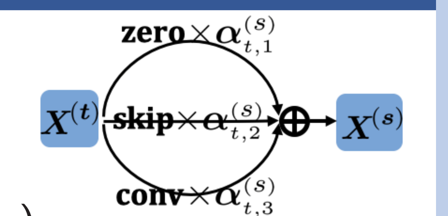
where the squared loss $F(W,\beta) = \frac{1}{2n}\sum_{i=1}^n (u_i - y_i)^2$.

Main Results

Gradient descent for optimization:

inner optimization: $W_s^{(l)}(k+1) = W_s^{(l)}(k) - \eta \nabla_{W_s^{(l)}(k)} F_{\text{train}}(W, \alpha)$

outer optimization: $\alpha_s^{(l)}(k+1) = \alpha_s^{(l)}(k) - \eta \nabla_{\alpha_s^{(l)}(k)} F_{\text{val}}(W_s^{(l)}(k+1), \alpha)$



Convergence of inner optimization. Under proper assumptions, for inner problem, the gradient descent algorithm can enjoy linear convergence rate:

$$F_{\text{train}}(W(k+1), \alpha) \le (1-\lambda) F_{\text{train}}(W(k), \alpha) \quad (\forall k \ge 1),$$

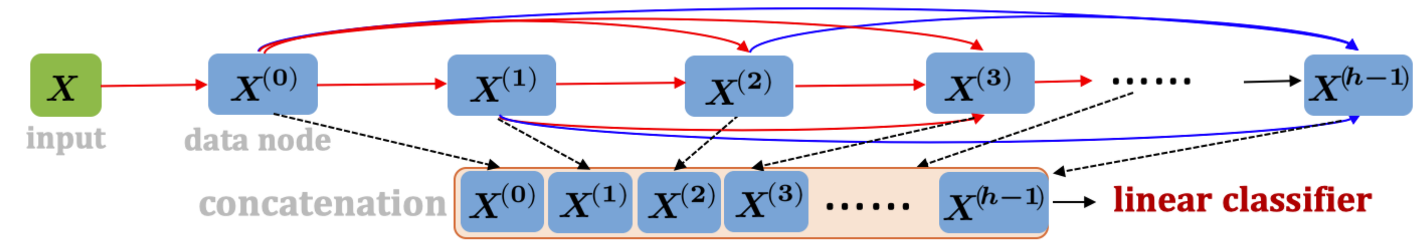
where $\lambda = c\eta \sum_{s=0}^{h-2} (\alpha_{s,3}^{(h-1)})^2 \prod_{t=0}^{s-1} (\alpha_{t,2}^{(s)})^2$ in which $\alpha_{t,2}^{(s)}$ and $\alpha_{t,3}^{(s)}$ respectively denote weights of skip and convolution connections, is a constant c and η is learning rate.

Remark: convergence rate $1-\lambda$ relies on weight $\alpha_{t,2}^{(S)}$ of skip connects more heavily

weights of convolutions which connect the last node $X^{(h-1)}$

$$\lambda = c\eta \sum_{s=0}^{h-2} \lambda_s \quad \text{with} \quad \lambda_s = (\alpha_{s,3}^{(h-1)})^2 \prod_{t=0}^{s-1} (\alpha_{t,2}^{(s)})^2$$

weights of skip connections which do not connect the last node $X^{(h-1)}$



Analysis to outer optimization:
- when skip connections have larger weights, the validation loss can decrease faster, since $\mathbb{E}[F_{\text{train}}(W), \alpha)] = \mathbb{E}[F_{\text{val}}(W), \alpha)]$.
- as all types of operations between two nodes share a softmax distribution

$$\alpha_{t,i}^{(s)} = \frac{\exp(\beta_{t,i}^{(s)})}{\sum_{i=1}^3 \exp(\beta_{t,i}^{(s)})} \longrightarrow \sum_i \alpha_{t,i}^{(s)} = 1,$$

if weight of skip connection becomes larger, other weights become smaller.

Conclusion:
- outer level optimization increases weights of skip connections and reduce other weights
- posterior pruning preserves most skip connections, prunes most other operations

Solution to Alleviate Unfair Competition among Operations

Natural solution: independent gate $g_{t,i}^{(s)}$ (Bernoulli distribution) for each operation

Issue 1 of independent gate. By using independent gate, increasing $g_{t,i}^{(s)}$ of any operations can provably reduce or maintain the validation loss.

Solution to Issue 1: group-structured sparsity regularization on gates
- Step 1. use Gumbel trick to produce an approximate Bernoulli variable u

$$u = \frac{\exp\left(\frac{v_1 + \ln v_2}{\tau}\right)}{1 + \exp\left(\frac{v_1 + \ln v_2}{\tau}\right)} \approx \text{Bernoulli}(v_2),\ v_1 \sim \text{Uniform}(0,1),\ v_2 = \frac{\exp(\beta_{t,i}^{(s)})}{1+\exp(\beta_{t,i}^{(s)})}$$

- Step 2. rescale to $[a,b]$ $(a<0, b>1)$, feed into hard threshold gate

$$g_{t,i}^{(s)} = a + (b-a)u, \quad g_{t,i}^{(s)} = \min(1, \max(0, g_{t,i}^{(s)}))$$

$$g_{t,i}^{(s)} = \begin{cases} 0, & \text{if } u \in (0, \frac{-a}{b-a}] \text{ (sparse)}, \\ g_{t,i}^{(s)}, & \text{if } u \in (\frac{-a}{b-a}, \frac{1-a}{b-a}], \\ 1, & \text{if } u \in (\frac{1-a}{b-a}, 1], \end{cases} \qquad \mathbb{P}(g_{t,i}^{(s)} \neq 0) = \Theta\left(\beta_{t,i}^{(s)} - \tau \ln \frac{-a}{b}\right)$$

gate activation probability; sigmoid

Solutions to Alleviate Unfair Competition among Operations

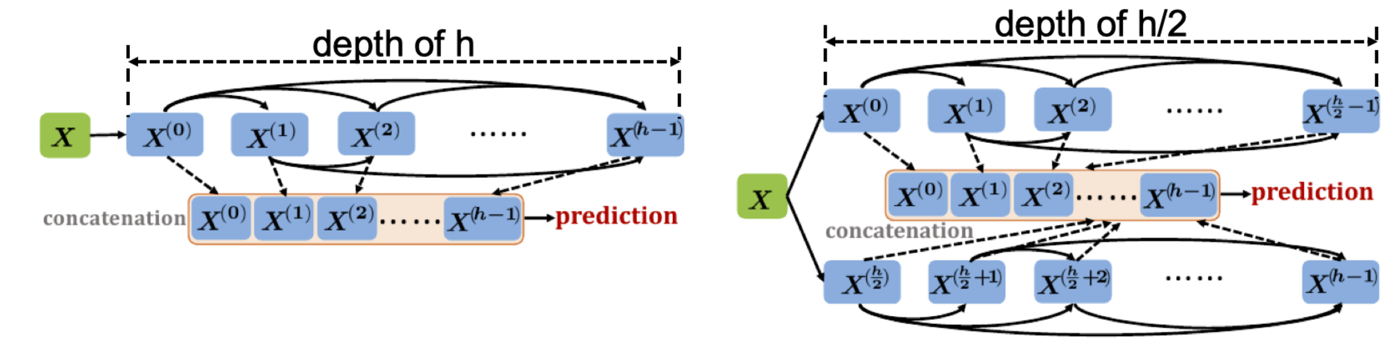
- Step 3. divide operations in network into skip connection and non-skip connection groups, compute their average gate activation probabilities:

$$\mathcal{L}_{\text{skip}}(\beta) = \zeta \sum_{l=1}^{h-1}\sum_{s=0}^{l-1} \Theta\left(\beta_{s,t_{\text{skip}}}^{(l)} - \tau \ln \frac{-a}{b}\right), \quad \mathcal{L}_{\text{non-skip}}(\beta) = \frac{\zeta}{r-1}\sum_{l=1}^{h-1}\sum_{s=0}^{l-1}\sum_{1\le t\le r, t\neq t_{\text{skip}}} \Theta\left(\beta_{s,t}^{(l)} - \tau \ln \frac{-a}{b}\right),$$

- Step 4. penalize $\mathcal{L}_{\text{skip}}(\beta)$ and $\mathcal{L}_{\text{non-skip}}(\beta)$ independently to avoid competition between skip connection and other type operations.

Issue 2 of independent gate: searching algorithm prefers to select shallow networks due to their faster convergence rate over deep ones.

Convergence Comparison. With proper assumptions, shallow network B can converge faster than the deep network A



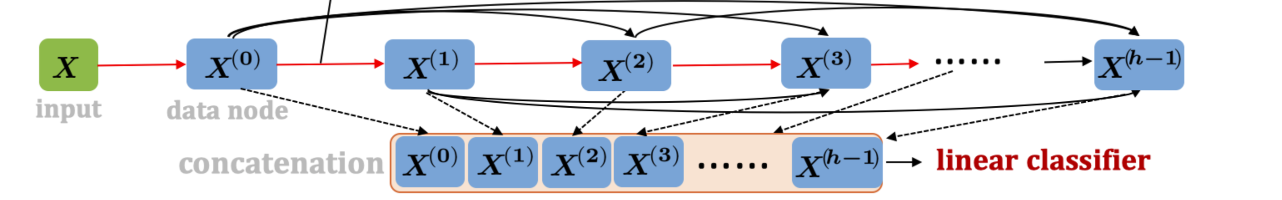
(a) Deep network A (b) Shallow network B

Solution to Issue 2: path-depth-wise regularization
- Step 1. compute probability that all neighboring nodes are connected via parameterized operations

$$\mathcal{L}_{\text{path}}(\beta) = \prod_{l=1}^{h-1} \mathbb{P}_{l,l+1}(\beta) = \prod_{l=1}^{h-1} \sum_{O_t \in \mathcal{O}_p} \Theta(\beta_{l,t}^{(l+1)} - \tau \ln \frac{-a}{b}).$$

corresponding activation probability of parameterized operations

connected by learnable parameterized operations, e.g. various types of convolutions



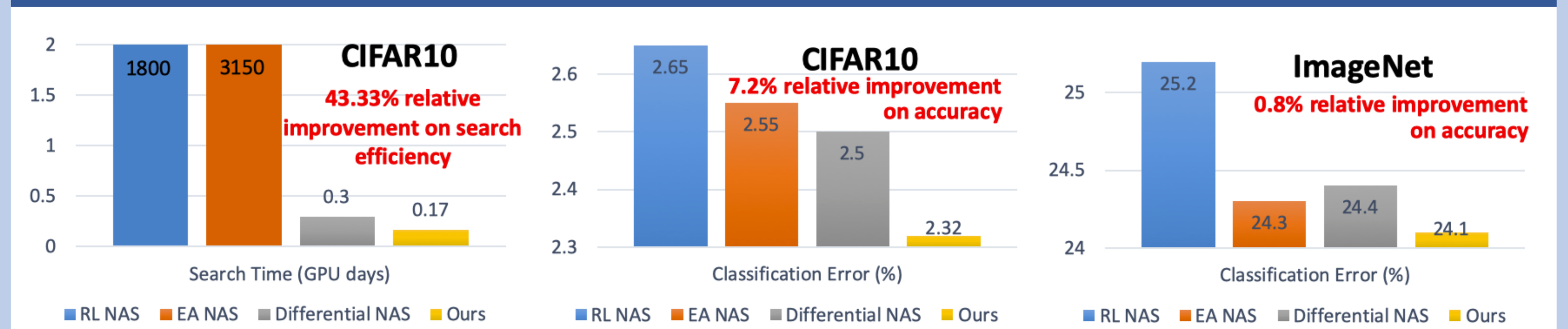
- Step 2. encourage deeper model via penalizing small $\mathcal{L}_{\text{path}}(\beta)$

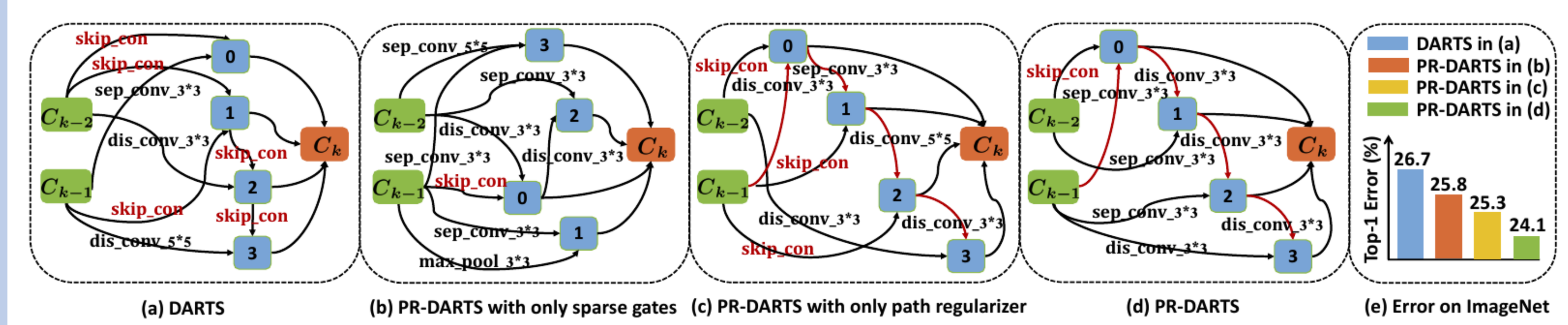
Final Model: $\min_\beta F_{\text{val}}(W^*(\beta), \beta) + \lambda_1 \mathcal{L}_{\text{skip}}(\beta) + \lambda_2 \mathcal{L}_{\text{non-skip}}(\beta) - \lambda_3 \mathcal{L}_{\text{path}}(\beta),$

group sparse; deeper model

s.t. $W^*(\beta) = \text{argmin}_W F_{\text{train}}(W, \beta)$

Experiments





(a) DARTS (b) PR-DARTS with only sparse gates (c) PR-DARTS with only path regularizer (d) PR-DARTS (e) Error on ImageNet