Neural Architecture Search

...

prezentowane przez Mateusza Olko

DARTS: Differentiable Architecture Search

- Rozwiązanie prostsze od innych (na przykład bez kontrolerów)
- Do rekurencji i do konwolucji
- Wydajne
- Można transferować na bardziej skomplikowane zbiory danych
- Wyniki konkurencyjne do SOTA

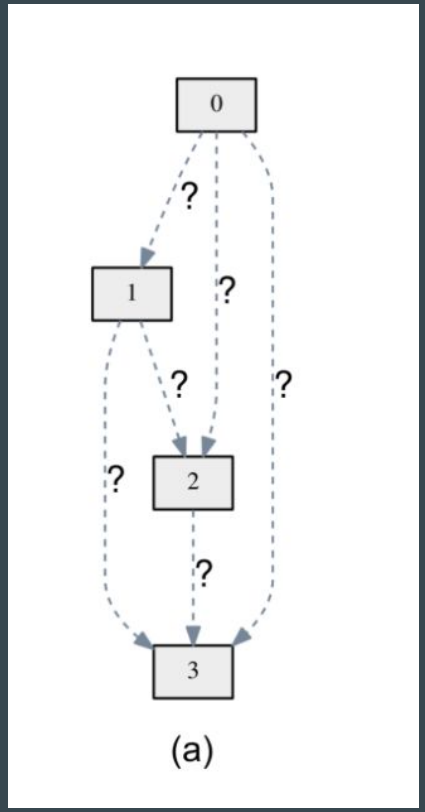
DARTS: Differentiable Architecture Search

Szukamy komórki podstawowej, która będzie budulcem naszej sieci, która:

- jest acyklicznym grafem z N ponumerowanymi wierzchołkami
- każdy wierzchołek to ukryta reprezentacja danych wejściowych
- każda krawędź (i, j) jest związana z jakąś operacją (np. konwolucja), która modyfikuje reprezentacja z i i przekazuje wynik jako wejście do j.
- każdy wierzchołek jest połączony ze wszystkimi poprzednimi
- ma dwa wierzchołki wejściowe i jeden wyjściowy

$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)})$$

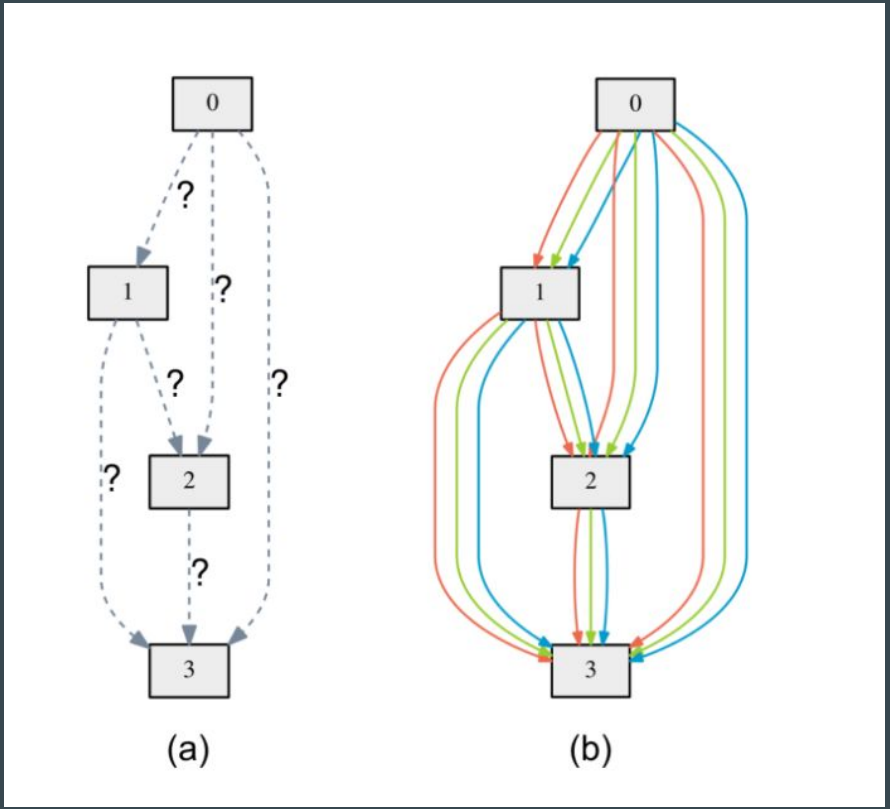
DARTS: Differentiable Architecture Search



Relaksacja przestrzeni

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

Relaksacja przestrzeni



Trening

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ \text{s.t.} \quad & w^*(\alpha) = \text{argmin}_w \ \mathcal{L}_{train}(w, \alpha) \end{aligned}$$

$$\begin{aligned} & \nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ \approx & \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha) \end{aligned}$$

Trening

Algorithm 1: DARTS – Differentiable Architecture Search

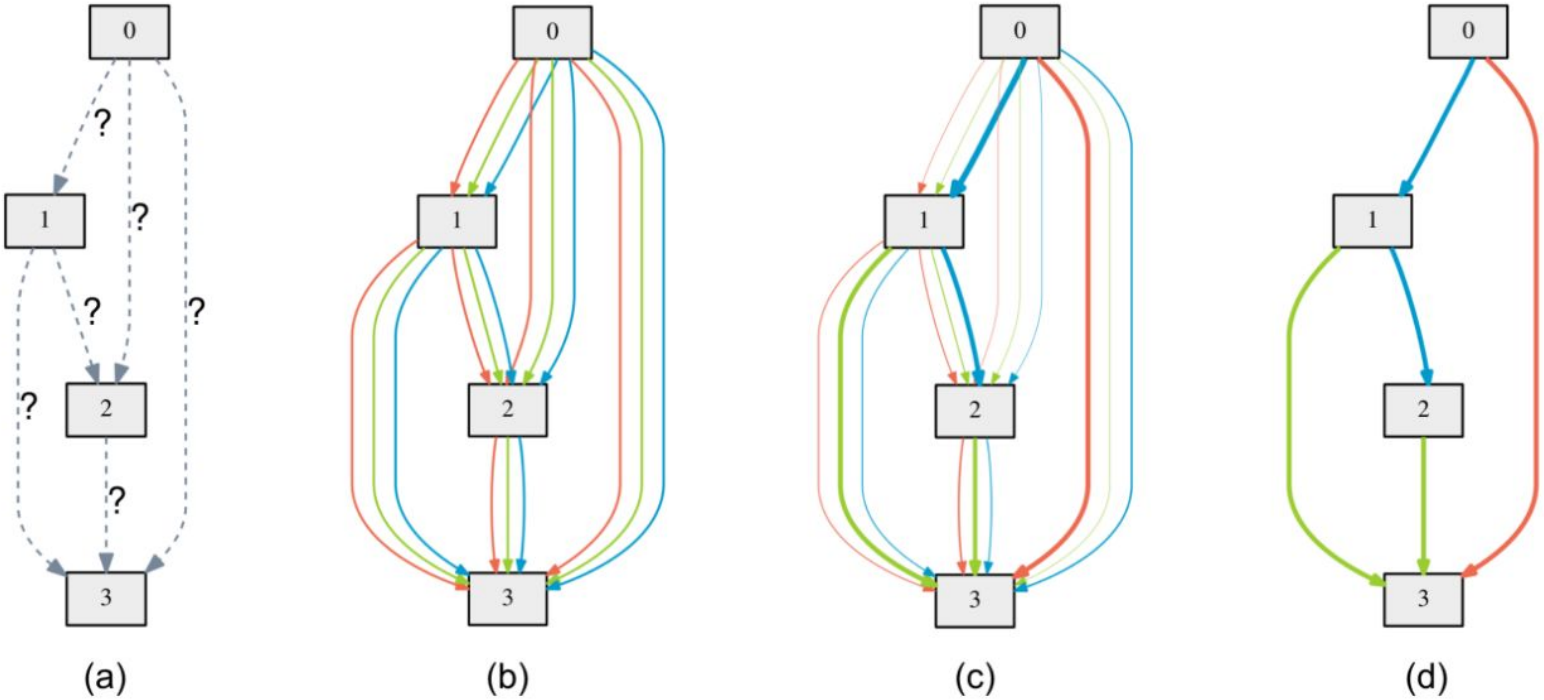
Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge (i, j)

while *not converged* **do**

1. Update architecture α 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 α.

Trening



Eksperymenty

Eksperymenty

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.

Eksperymenty

Table 2: Comparison with state-of-the-art language models on PTB (lower perplexity 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 (3 GPU days).

Architecture	Perplexity		Params (M)	Search Cost (GPU days)	#ops	Search Method
	valid	test				
Variational RHN (Zilly et al., 2016)	67.9	65.4	23	–	–	manual
LSTM (Merity et al., 2018)	60.7	58.8	24	–	–	manual
LSTM + skip connections (Melis et al., 2018)	60.9	58.3	24	–	–	manual
LSTM + 15 softmax experts (Yang et al., 2018)	58.1	56.0	22	–	–	manual
NAS (Zoph & Le, 2017)	–	64.0	25	1e4 CPU days	4	RL
ENAS (Pham et al., 2018b)*	68.3	63.1	24	0.5	4	RL
ENAS (Pham et al., 2018b)†	60.8	58.6	24	0.5	4	RL
Random search baseline‡	61.8	59.4	23	2	4	random
DARTS (first order)	60.2	57.6	23	0.5	4	gradient-based
DARTS (second order)	58.1	55.7	23	1	4	gradient-based

* Obtained using the code (Pham et al., 2018a) publicly released by the authors.

† Obtained by training the corresponding architecture using our setup.

‡ Best architecture among 8 samples according to the validation perplexity after 300 training epochs.

Eksperymenty

Table 3: Comparison with state-of-the-art image classifiers on ImageNet in the mobile setting.

Architecture	Test Error (%)		Params	$+\times$	Search Cost	Search
	top-1	top-5	(M)	(M)	(GPU days)	Method
Inception-v1 (Szegedy et al., 2015)	30.2	10.1	6.6	1448	–	manual
MobileNet (Howard et al., 2017)	29.4	10.5	4.2	569	–	manual
ShuffleNet $2\times$ ($g = 3$) (Zhang et al., 2017)	26.3	–	$\sim$5	524	–	manual
NASNet-A (Zoph et al., 2018)	26.0	8.4	5.3	564	2000	RL
NASNet-B (Zoph et al., 2018)	27.2	8.7	5.3	488	2000	RL
NASNet-C (Zoph et al., 2018)	27.5	9.0	4.9	558	2000	RL
AmoebaNet-A (Real et al., 2018)	25.5	8.0	5.1	555	3150	evolution
AmoebaNet-B (Real et al., 2018)	26.0	8.5	5.3	555	3150	evolution
AmoebaNet-C (Real et al., 2018)	24.3	7.6	6.4	570	3150	evolution
PNAS (Liu et al., 2018a)	25.8	8.1	5.1	588	$\sim$225	SMBO
DARTS (searched on CIFAR-10)	26.7	8.7	4.7	574	4	gradient-based

Eksperymenty

Table 4: Comparison with state-of-the-art language models on WT2.

Architecture	Perplexity valid	Perplexity test	Params (M)	Search Cost (GPU days)	Search Method
LSTM + augmented loss (Inan et al., 2017)	91.5	87.0	28	–	manual
LSTM + continuous cache pointer (Grave et al., 2016)	–	68.9	–	–	manual
LSTM (Merity et al., 2018)	69.1	66.0	33	–	manual
LSTM + skip connections (Melis et al., 2018)	69.1	65.9	24	–	manual
LSTM + 15 softmax experts (Yang et al., 2018)	66.0	63.3	33	–	manual
ENAS (Pham et al., 2018b)† (searched on PTB)	72.4	70.4	33	0.5	RL
DARTS (searched on PTB)	71.2	69.6	33	1	gradient-based

† Obtained by training the corresponding architecture using our setup.

Eksperymenty

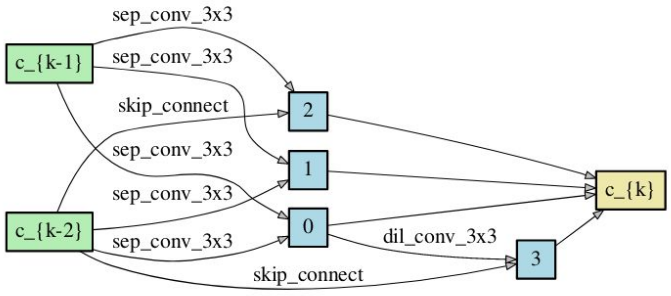


Figure 4: Normal cell learned on CIFAR-10.

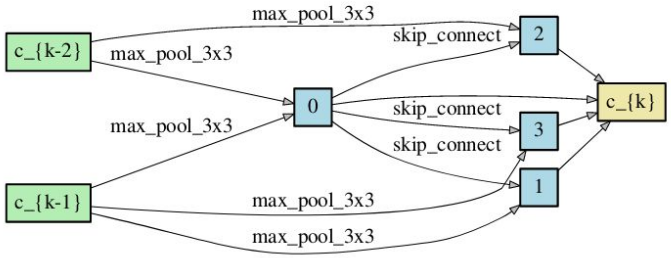


Figure 5: Reduction cell learned on CIFAR-10.

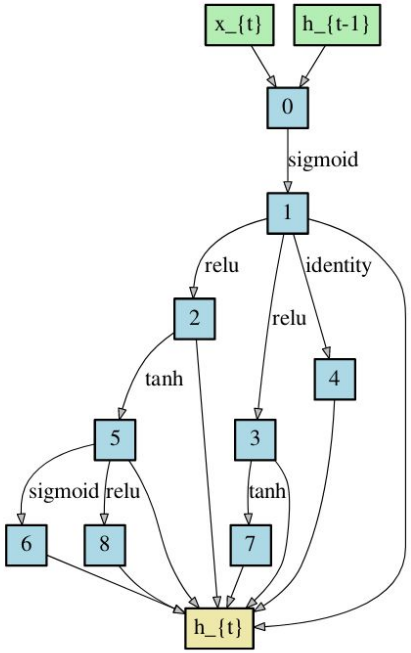


Figure 6: Recurrent cell learned on PTB.

Ale można by lepiej

- Trenować architekturę sieci, a nie tylko komórki
- Rozwiązać trudniejsze zadanie
- Zachować wydajność i jakość

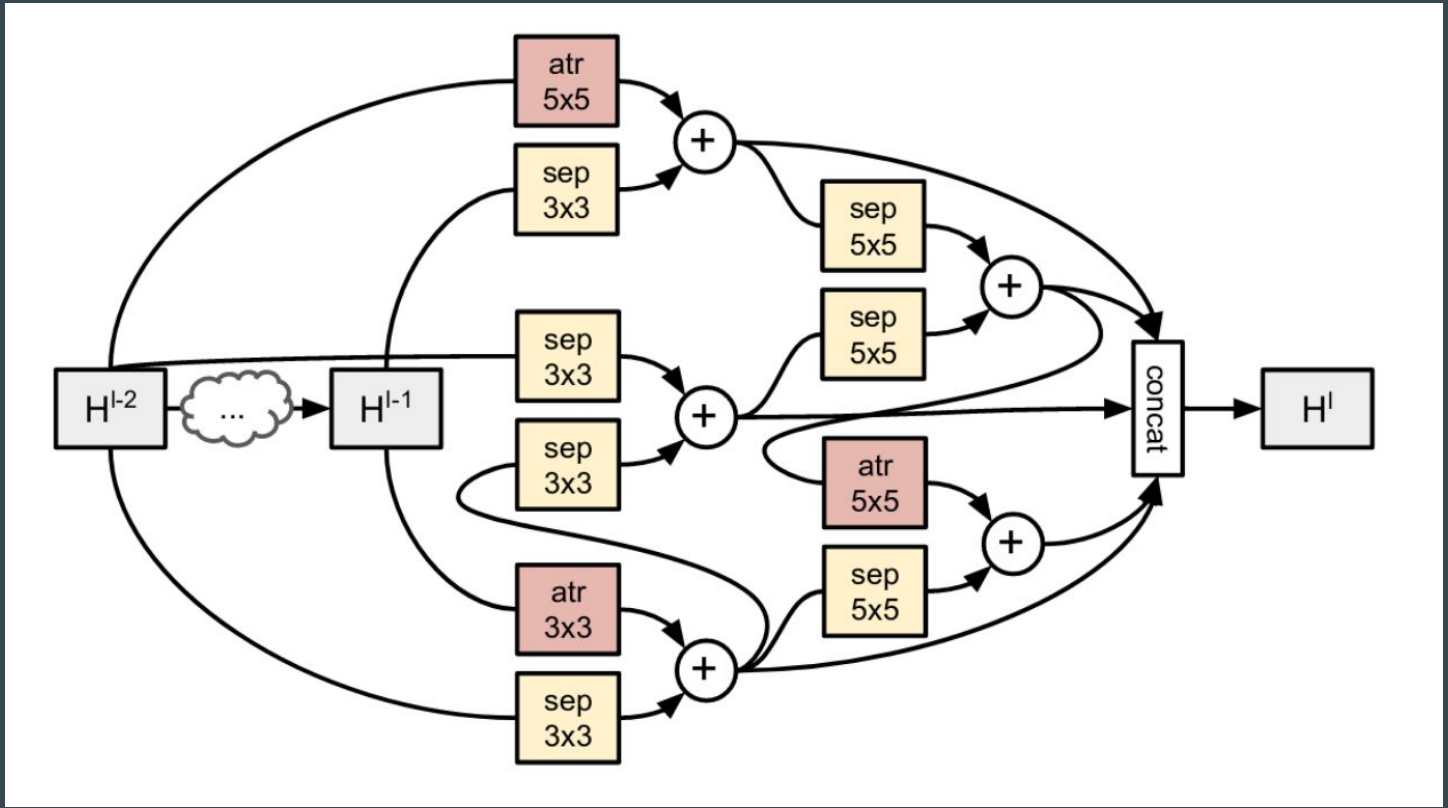
Ale można by lepiej

- Trenować architekturę sieci, a nie tylko komórki
- Rozwiązać trudniejsze zadanie
- Zachować wydajność i jakość

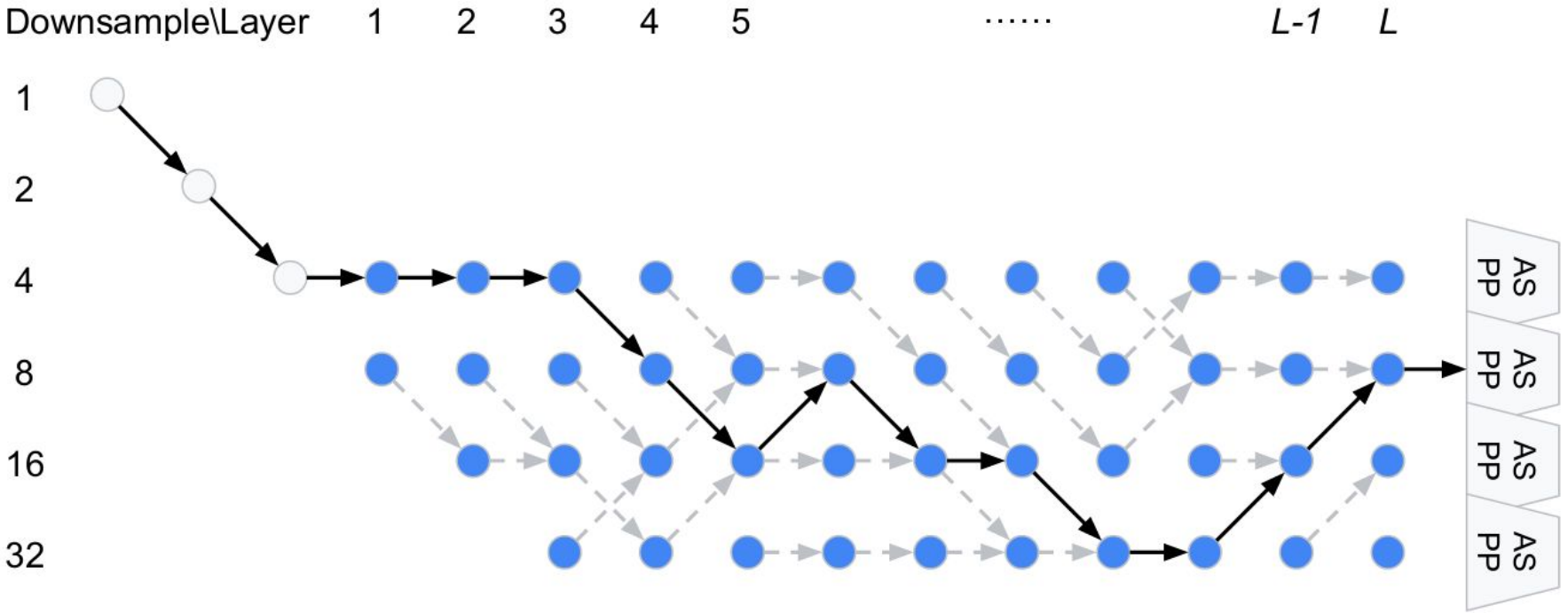
Auto-DeepLab:
Hierarchical Neural Architecture Search for Semantic Image Segmentation

Chenxi Liu[1]*, Liang-Chieh Chen[2], Florian Schroff[2], Hartwig Adam[2], Wei Hua[2],
Alan Yuille[1], Li Fei-Fei[3]
[1]Johns Hopkins University [2]Google [3]Stanford University

Przestrzeń wyszukiwań bloku



Przestrzeń wyszukiwań sieci



Relaksacja

We reuse the continuous relaxation described in [49]. Every block's output tensor H_i^l is connected to all hidden states in $\mathcal{I}_i^l$:

$$H_i^l = \sum_{H_j^l \in \mathcal{I}_i^l} O_{j\to i}(H_j^l) \tag{1}$$

In addition, we approximate each $O_{j\to i}$ with its continuous relaxation $\bar{O}_{j\to i}$, defined as:

$$\bar{O}_{j\to i}(H_j^l) = \sum_{O^k \in \mathcal{O}} \alpha_{j\to i}^k O^k(H_j^l) \tag{2}$$

where

$$\sum_{k=1}^{|\mathcal{O}|} \alpha_{j\to i}^k = 1 \qquad \forall i, j \tag{3}$$

$$\alpha_{j\to i}^k \geq 0 \qquad \forall i, j, k \tag{4}$$

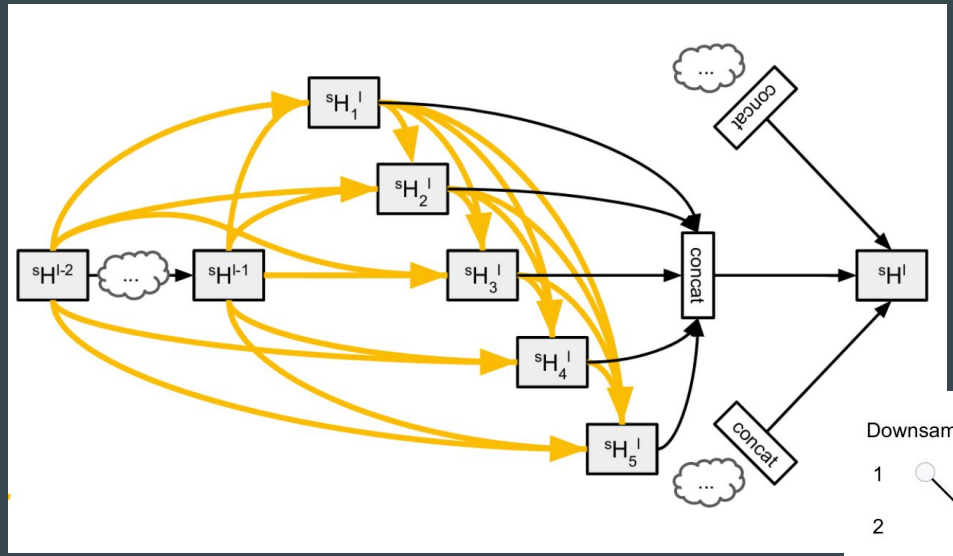
$$\begin{aligned} {}^{s}H^{l} =& \beta^{l}_{\frac{s}{2}\to s}\text{Cell}({}^{\frac{s}{2}}H^{l-1}, {}^{s}H^{l-2}; \alpha) \\ &+ \beta^{l}_{s\to s}\text{Cell}({}^{s}H^{l-1}, {}^{s}H^{l-2}; \alpha) \\ &+ \beta^{l}_{2s\to s}\text{Cell}({}^{2s}H^{l-1}, {}^{s}H^{l-2}; \alpha) \end{aligned} \tag{6}$$

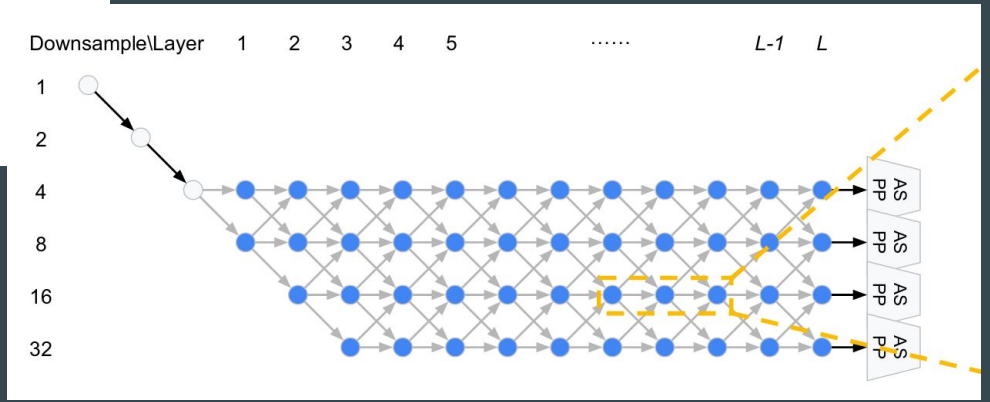
where $s = 4, 8, 16, 32$ and $l = 1, 2, \ldots, L$. The scalars β are normalized such that

$$\beta^{l}_{s\to\frac{s}{2}} + \beta^{l}_{s\to s} + \beta^{l}_{s\to 2s} = 1 \qquad \forall s, l \tag{7}$$

$$\beta^{l}_{s\to\frac{s}{2}} \geq 0 \quad \beta^{l}_{s\to s} \geq 0 \quad \beta^{l}_{s\to 2s} \geq 0 \qquad \forall s, l \tag{8}$$

Relaksacja


$^sH_1^l$
$^sH_2^l$
$^sH_3^l$
$^sH_4^l$
$^sH_5^l$
$^sH^{l-2}$
$^sH^{l-1}$
$^sH^l$
concat
concat
concat
...


Downsample\Layer
1
2
3
4
5
.....
L-1
L
1
2
4
8
16
32
AS PP
AS PP
AS PP
AS PP

Referencje

Prezentację przygotowano na podstawie dwóch publikacji. Pochodzą z nich również wszystkie grafiki na slajdach.

- https://arxiv.org/abs/1806.09055
- https://arxiv.org/abs/1901.02985

Dziękuję za uwagę