Нелинейная цифровая обработка сигналов

DARTS

Differentable architecture search

Власов Роман, старший инженер Huawei
Афонин Андрей, инженер Huawei

HUAWEI

18.08.2020

Сириус
Образовательный центр

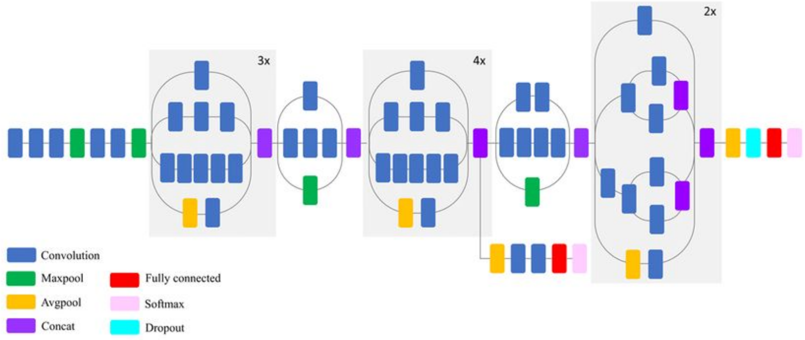
Объект исследования

$$f(x, y) = x^2 + xy + (x + y)^2$$

HUAWEI

Объект исследования

Нейронные сети



Объект исследования

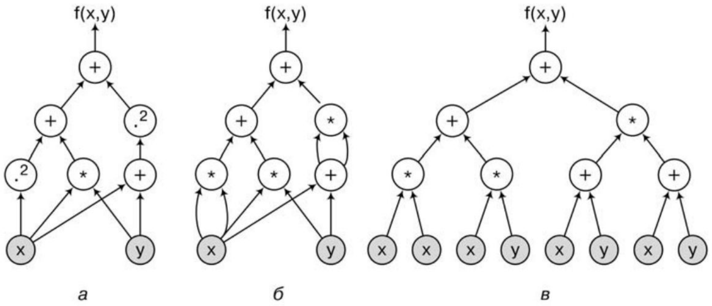
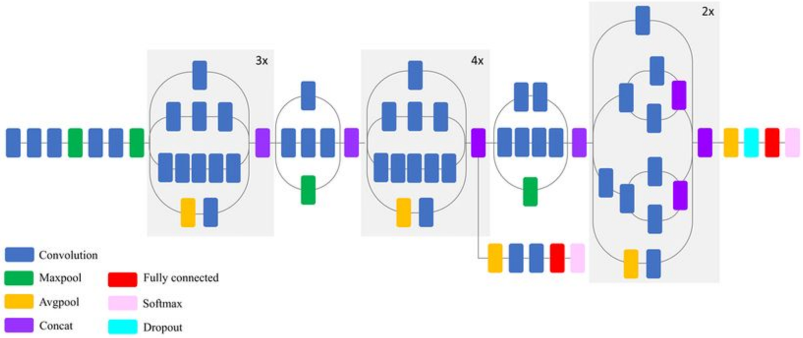


Рис. 2.7. Графы вычислений для функции $f(x, y) = x^2 + xy + (x + y)^2$:
а — с использованием функций $+$, $\times$ и ^{2}; *б* — с использованием функций $+$ и $\times$;
в — в виде дерева с использованием функций $+$ и $\times$

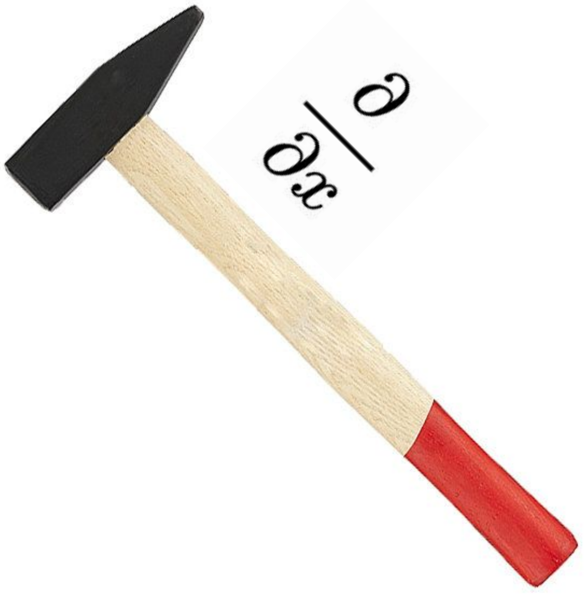


HUAWEI

D for Differentional

Поиск архитектуры

Поиск параметров



Арсенал

Методы оптимизации:

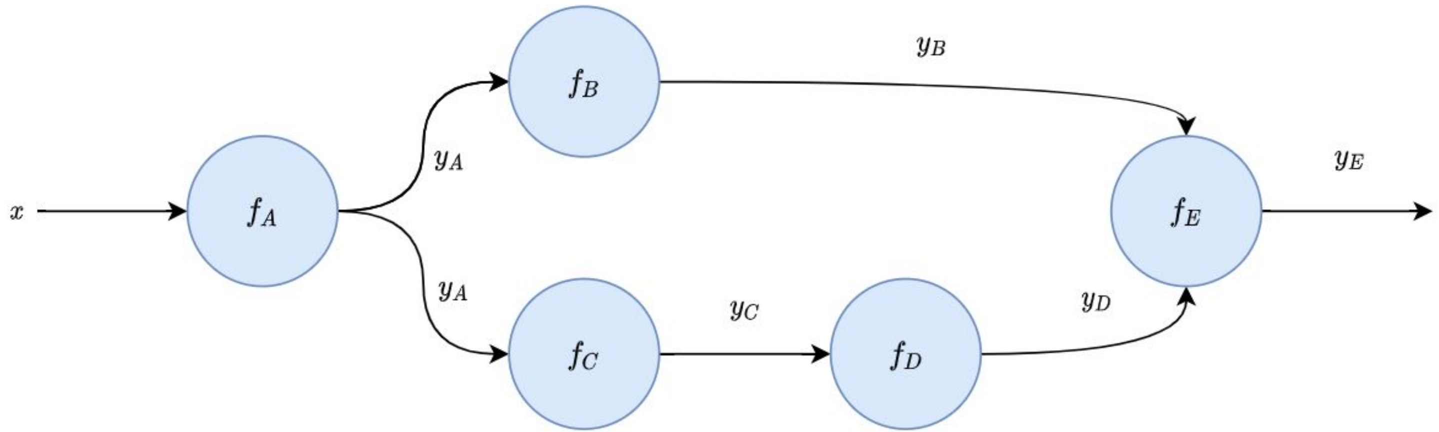
- Методы первого порядка
  - SGD
  - Momentum
  - Adam
  - Adagrad
  - …
- Квазиньютоновские методы
  - BFGS
  - L-BFGS
  - Levenberg-Marquardt
  - …

Фреймворки (+ GPU):

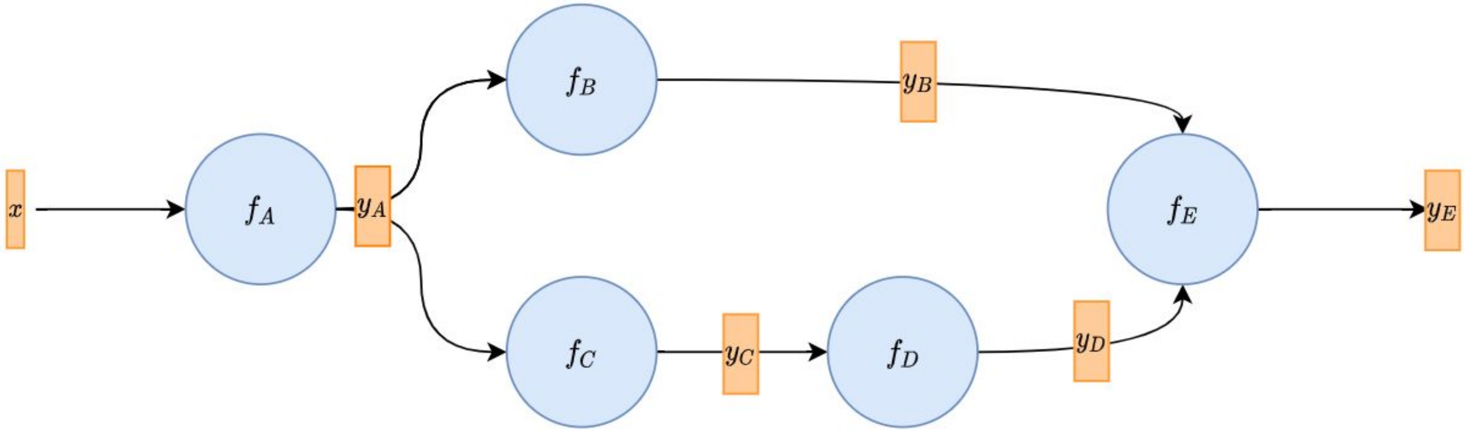
- Tensorflow
- PyTorch
- Caffe
- …

$$\frac{\partial}{\partial x}$$

Представление данных

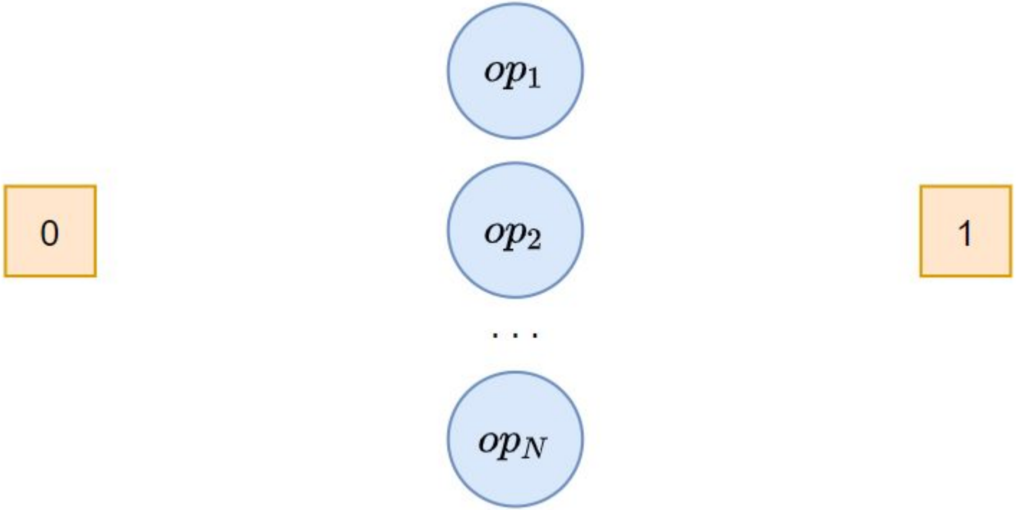


Представление данных



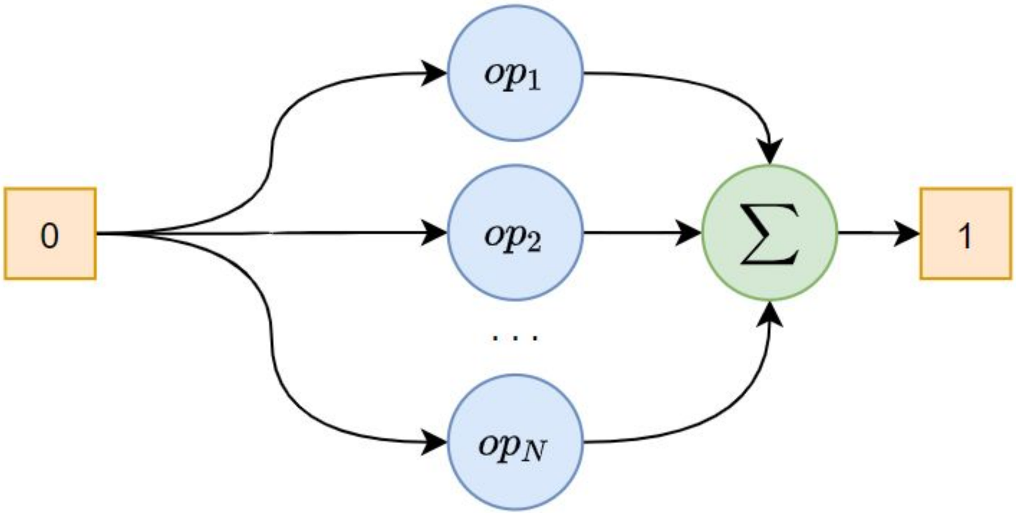
Ячейка DARTS

Ячейка DARTS



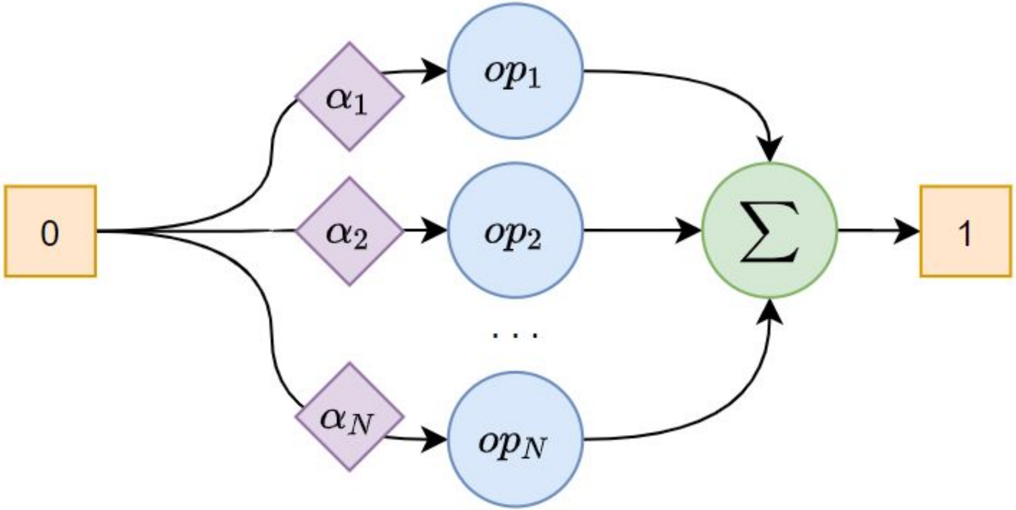
HUAWEI

Ячейка DARTS

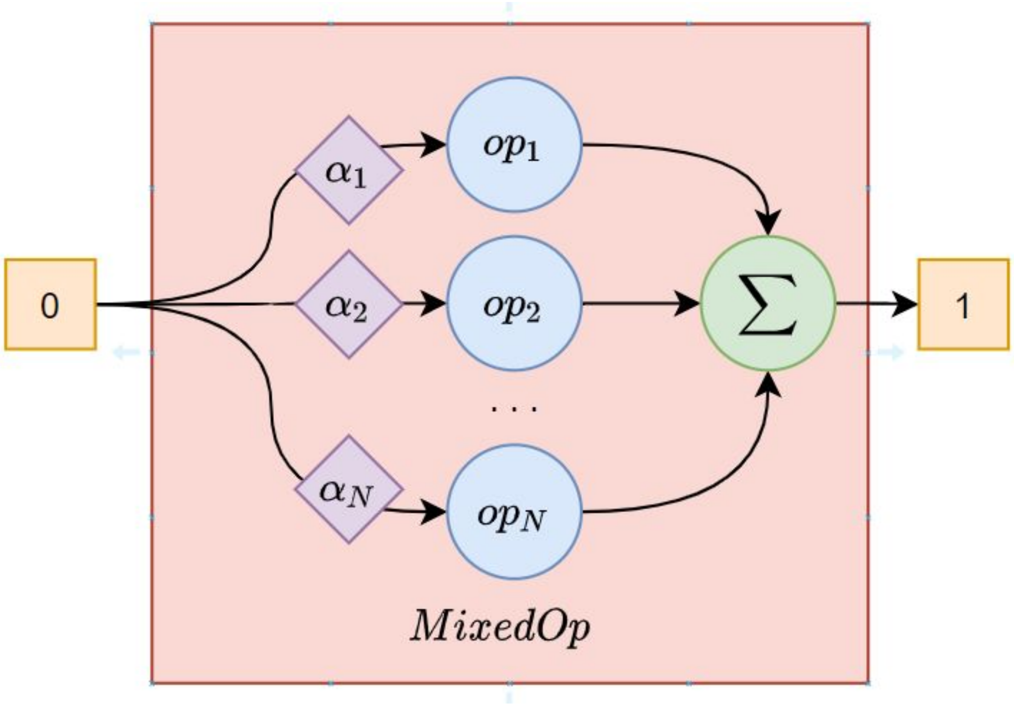


HUAWEI

Ячейка DARTS

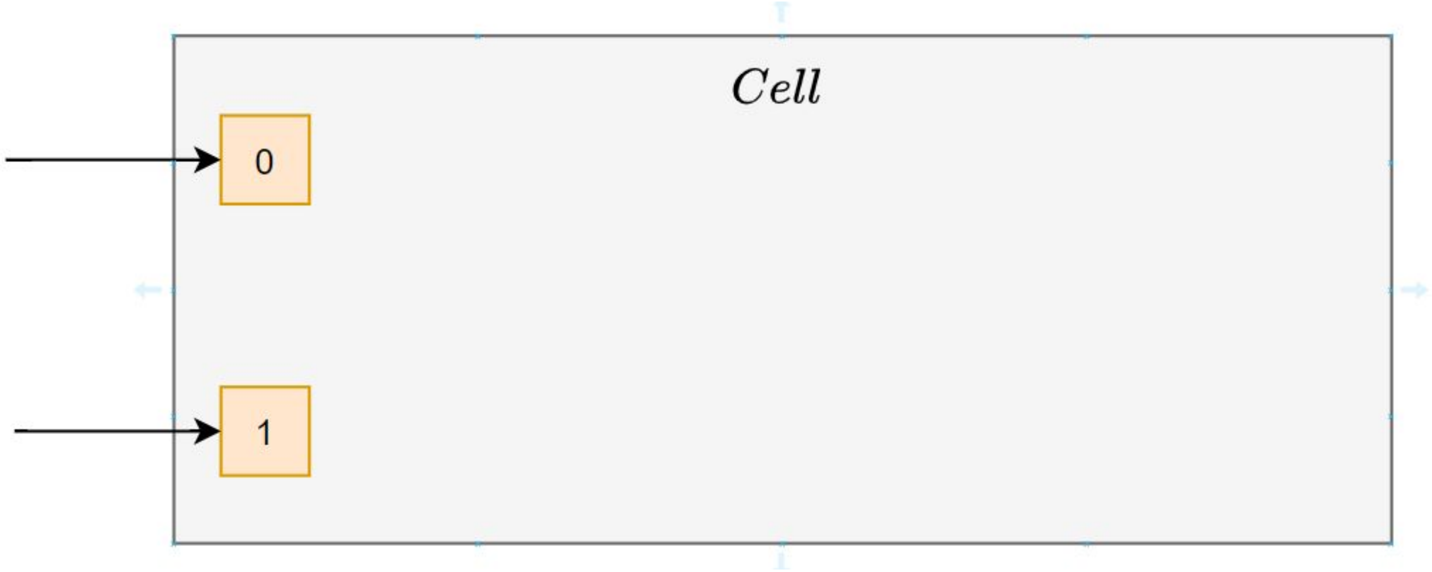


Ячейка DARTS

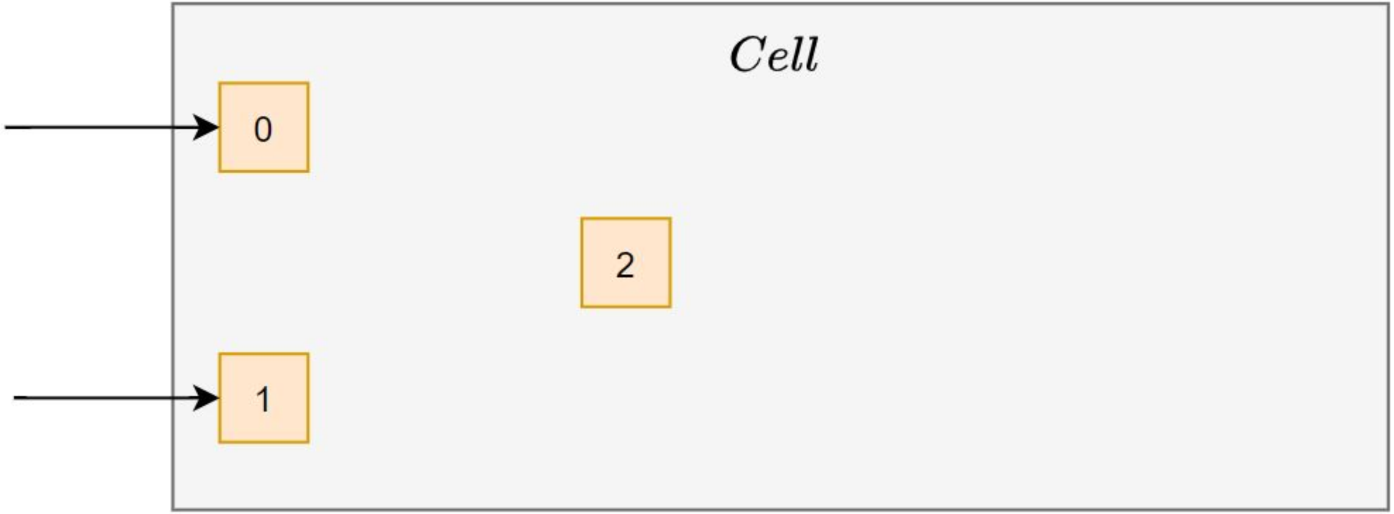


Ячейка DARTS

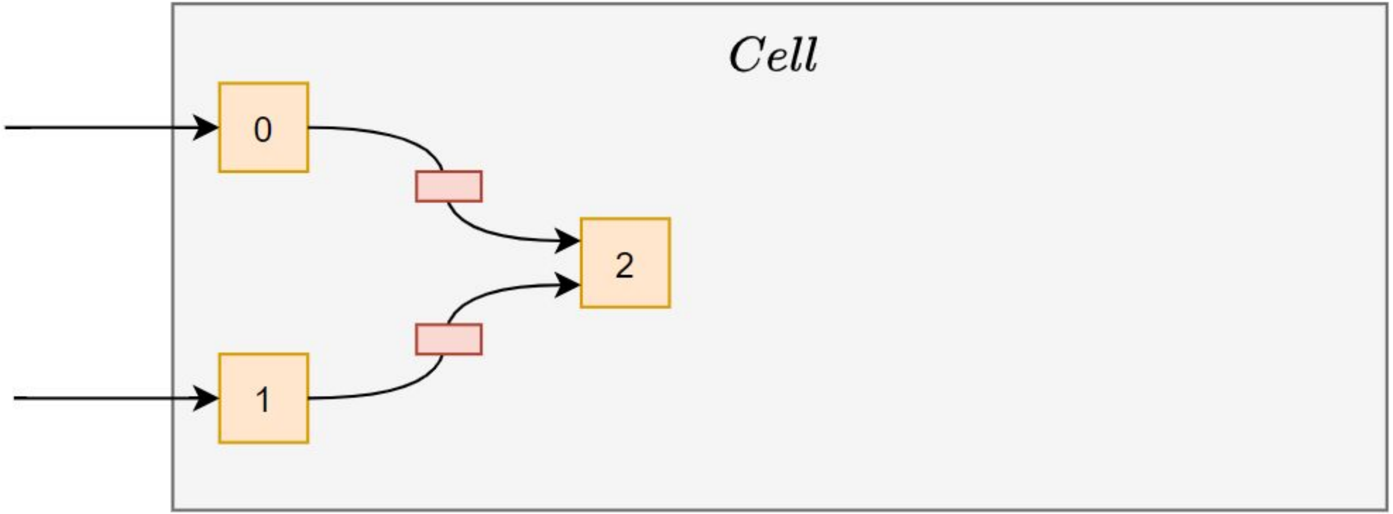
Ячейка DARTS



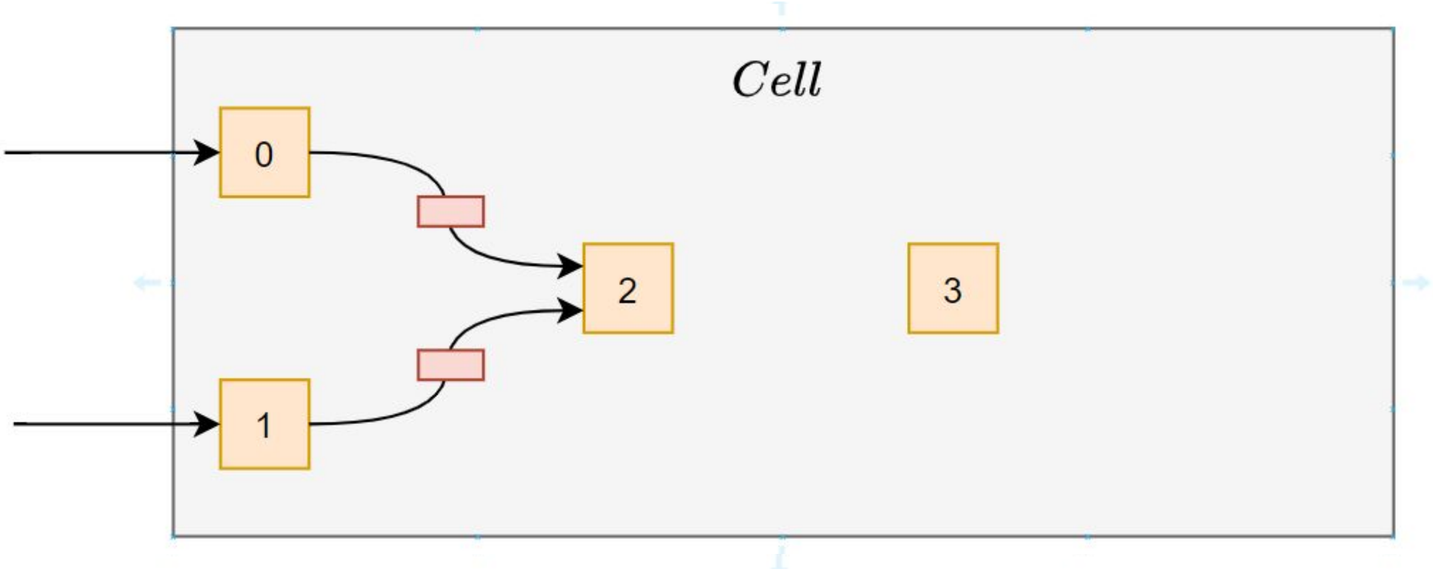
Ячейка DARTS



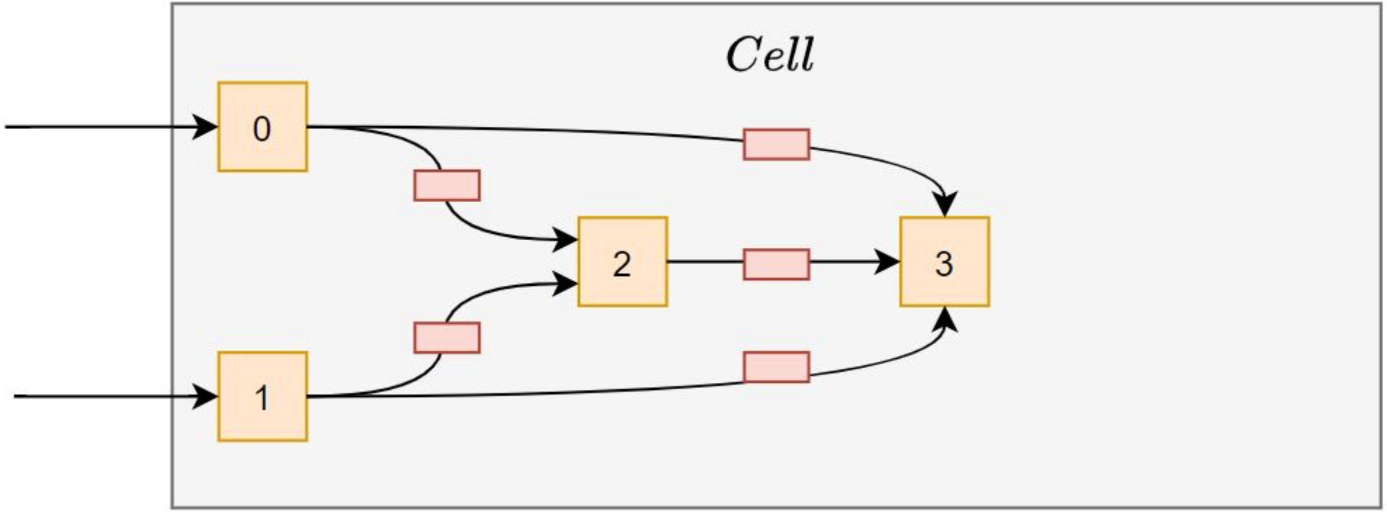
Ячейка DARTS



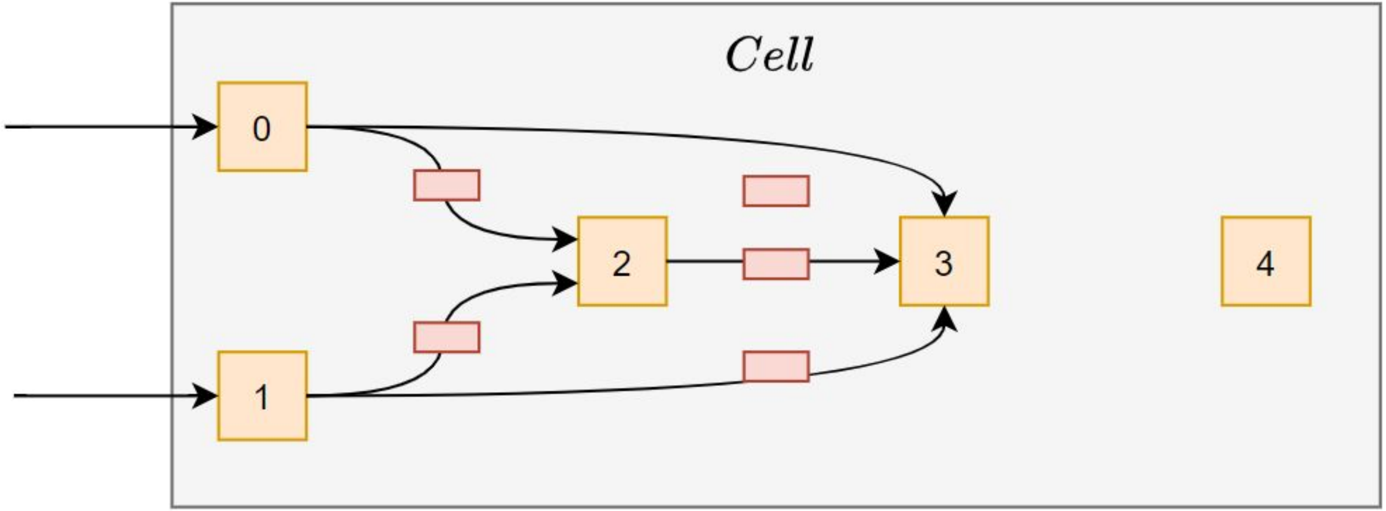
Ячейка DARTS



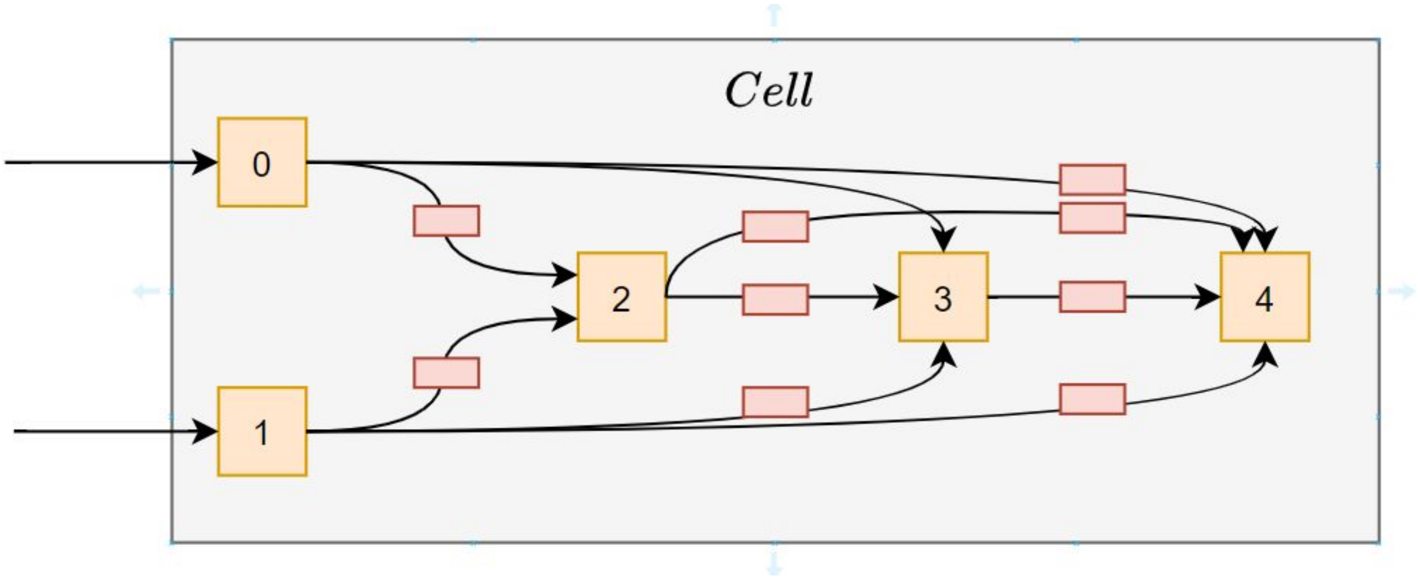
Ячейка DARTS



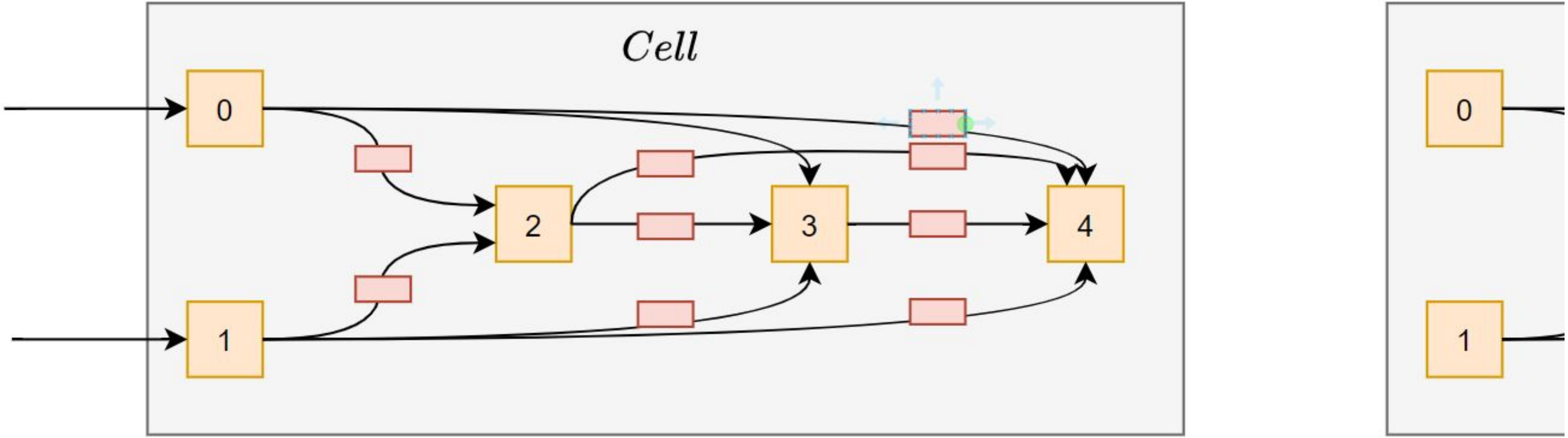
Ячейка DARTS



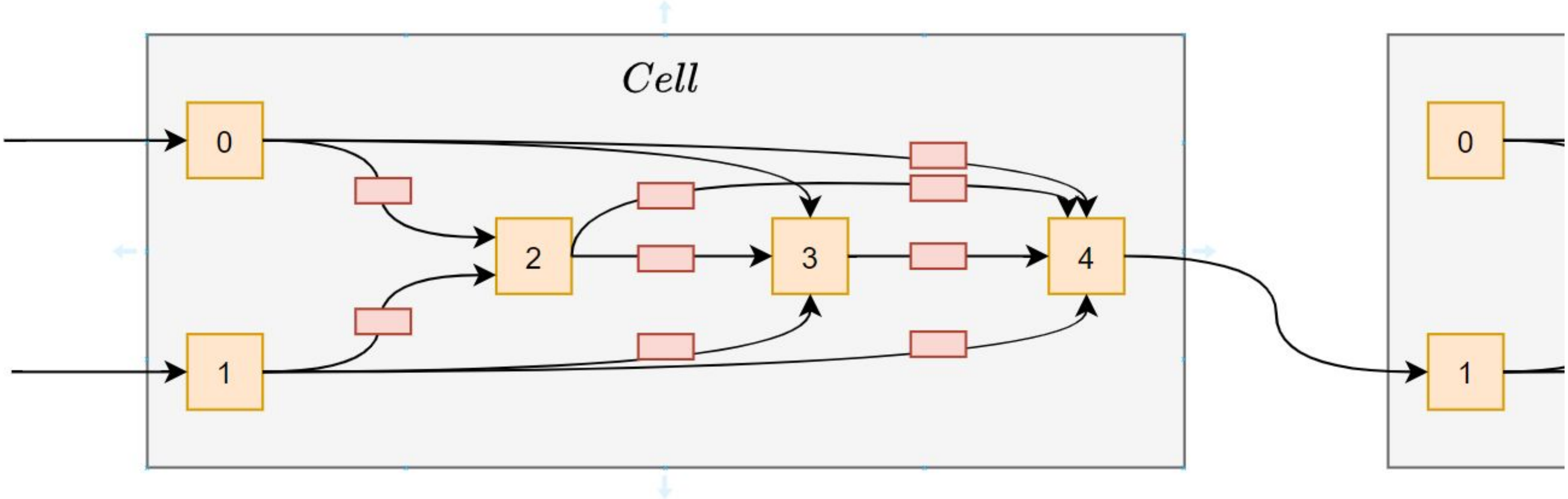
Ячейка DARTS



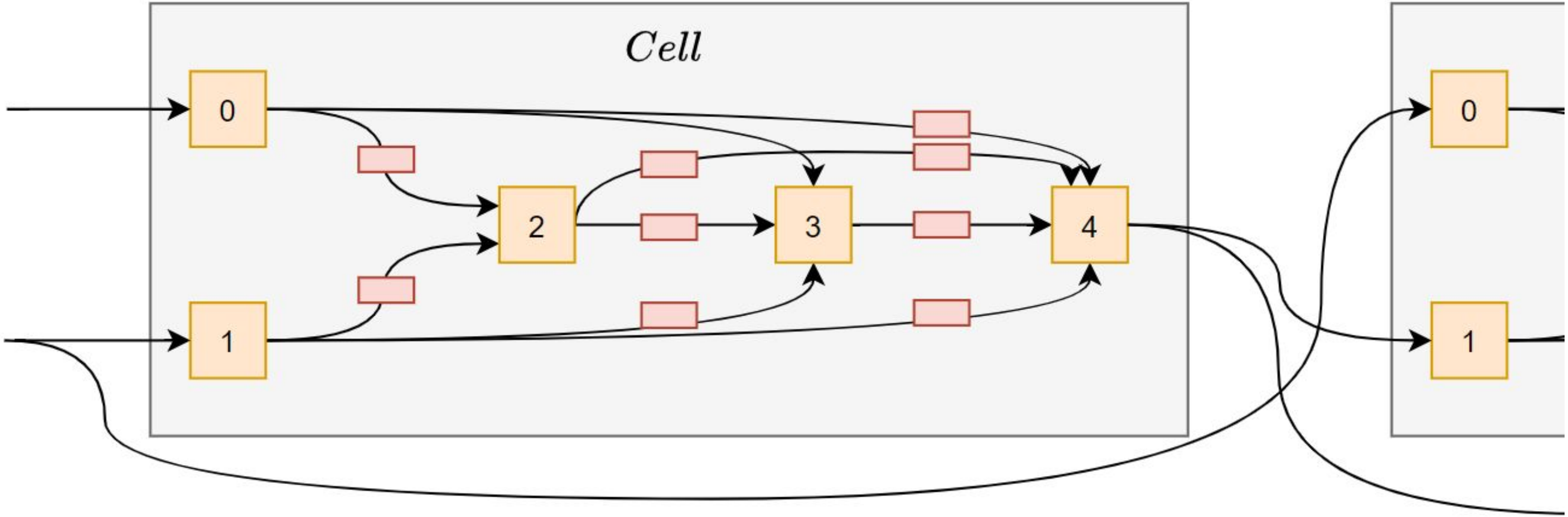
Ячейка DARTS



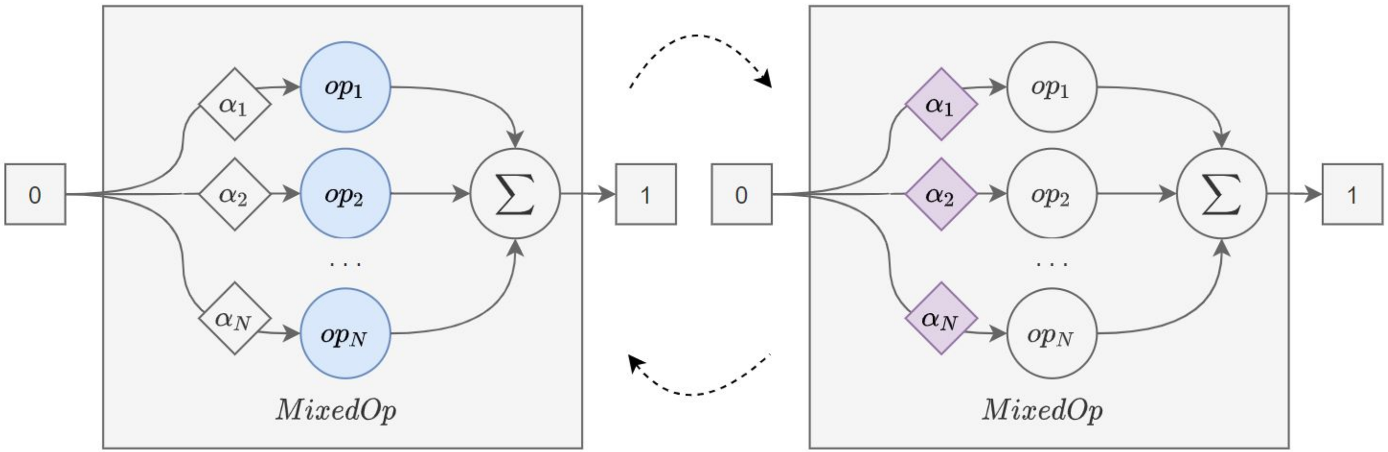
Ячейка DARTS



Ячейка DARTS



Обучение



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 $\nabla_w \mathcal{L}_{train}(w, \alpha)$

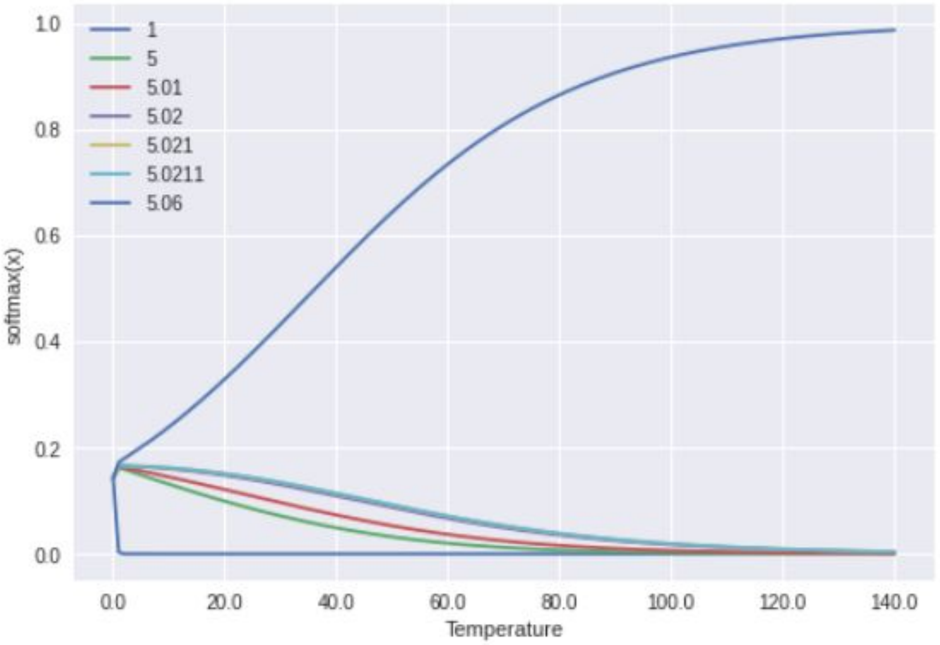
Derive the final architecture based on the learned α.

https://arxiv.org/pdf/1806.09055.pdf

Выбор операций

Softmax:

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



Параметры алгоритма

- Пространство поиска (словарь)
- Количество step-ов внутри ячейки
- Количество слоев (ячеек)
- Параметры обучения

Параметры алгоритма

- **Пространство поиска (словарь)**
- Количество step-ов внутри ячейки
- Количество слоев (ячеек)
- Параметры обучения

CNN

```
OPS = {
  'none' ,
  'avg_pool_3x3' ,
  'max_pool_3x3' ,
  'skip_connect' ,
  'sep_conv_3x3' ,
  'sep_conv_5x5' ,
  'sep_conv_7x7' ,
  'dil_conv_3x3' ,
  'dil_conv_5x5' ,
  'conv_7x1_1x7' : lambda C, stride, affine: nn.Sequential(
    nn.ReLU(inplace=False),
    nn.Conv2d(C, C, (1,7), stride=(1, stride), padding=(0, 3), bias=False),
    nn.Conv2d(C, C, (7,1), stride=(stride, 1), padding=(3, 0), bias=False),
    nn.BatchNorm2d(C, affine=affine)
    ),
}
```

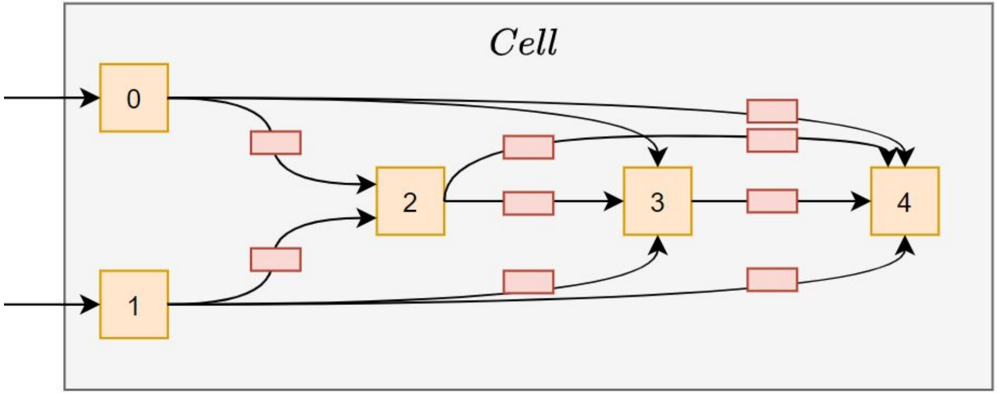
DPD

```
OPS = {
  'none' ,
  'Polynomial' ,
  'Delay' ,
  'FIR' ,
  'FIR-POLY-FIR' ,
}
```

Параметры алгоритма

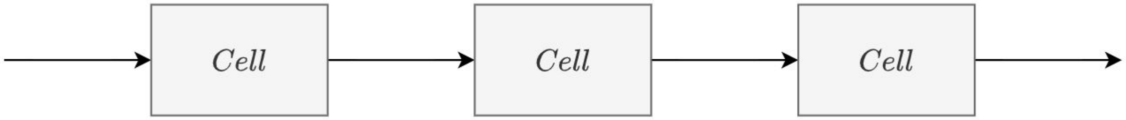
- Пространство поиска (словарь)
- **Количество step-ов внутри ячейки**
- Количество слоев (ячеек)
- Параметры обучения

$$N_{op} = \sum_{i=1}^{N_{step}} i + 2$$



Параметры алгоритма

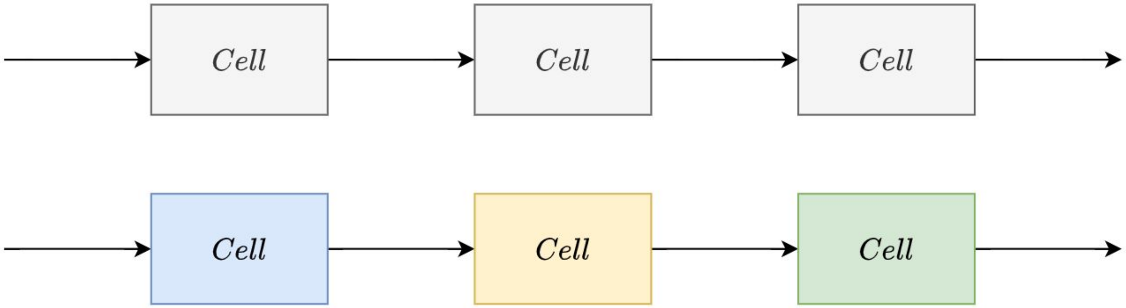
- Пространство поиска (словарь)
- Количество step-ов внутри ячейки
- **Количество ячеек**
- Параметры обучения



HUAWEI

Параметры алгоритма

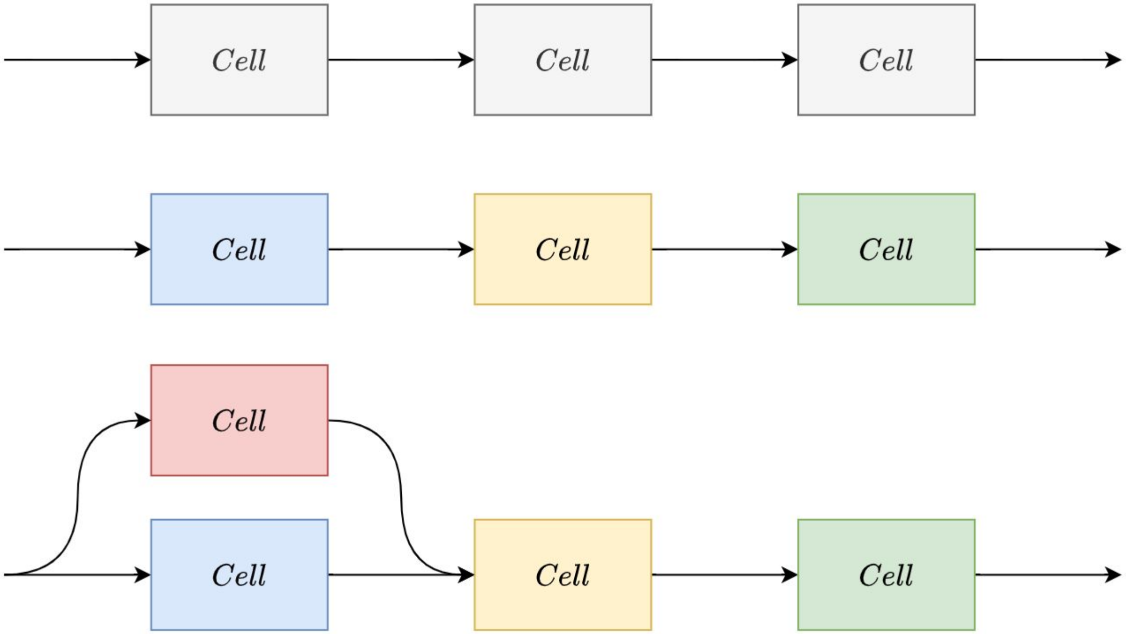
- Пространство поиска (словарь)
- Количество step-ов внутри ячейки
- **Количество ячеек**
- Параметры обучения



HUAWEI

Параметры алгоритма

- Пространство поиска (словарь)
- Количество step-ов внутри ячейки
- **Количество ячеек**
- Параметры обучения



Параметры алгоритма

- Пространство поиска (словарь)
- Количество step-ов внутри ячейки
- **Количество ячеек**
- Параметры обучения

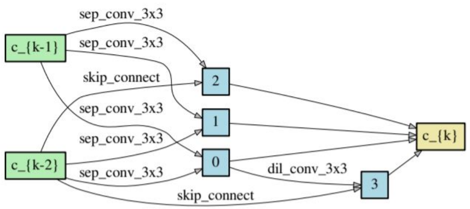


Figure 4: Normal cell learned on CIFAR-10.

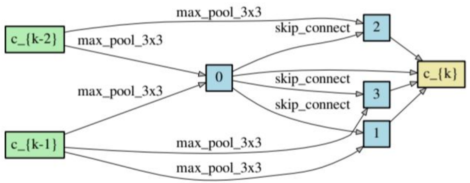
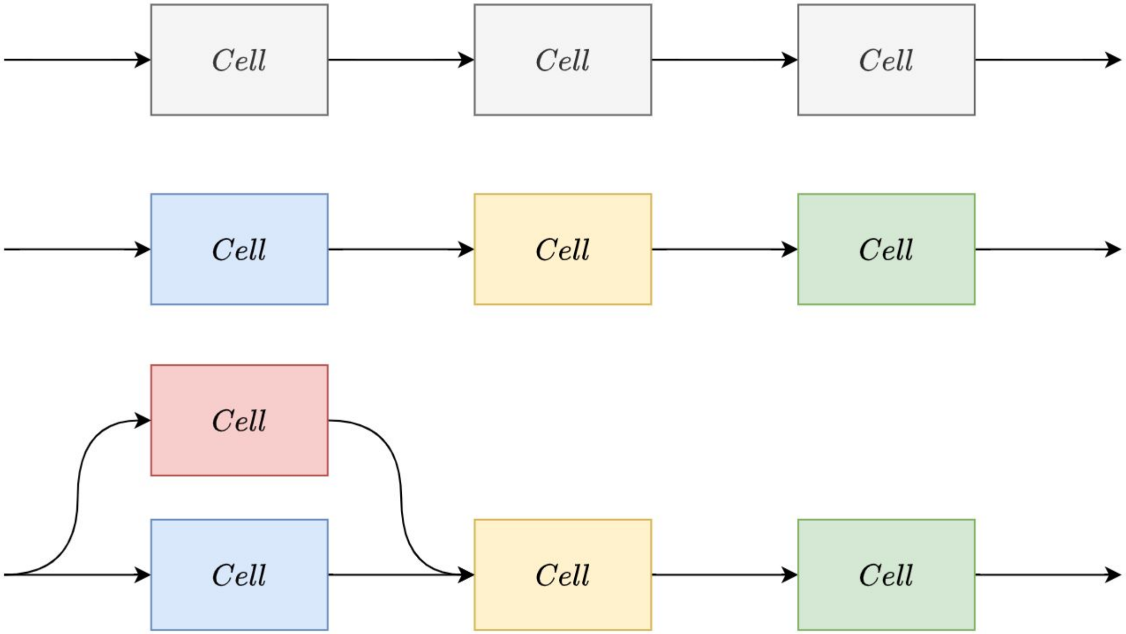


Figure 5: Reduction cell learned on CIFAR-10.



Параметры алгоритма

- Пространство поиска (словарь)
- Количество step-ов внутри ячейки
- Количество ячеек
- **Параметры обучения**

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

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

Applying chain rule to the approximate architecture gradient (equation 6) yields

$$\nabla_\alpha \mathcal{L}_{val}(w', \alpha) - \xi \nabla^2_{\alpha,w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \tag{7}$$

where $w' = w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha)$ denotes the weights for a one-step forward model. The expression above contains an expensive matrix-vector product in its second term. Fortunately, the complexity can be substantially reduced using the finite difference approximation. Let ϵ be a small scalar[2] and $w^\pm = w \pm \epsilon \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$. Then:

$$\nabla^2_{\alpha,w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \approx \frac{\nabla_\alpha \mathcal{L}_{train}(w^+, \alpha) - \nabla_\alpha \mathcal{L}_{train}(w^-, \alpha)}{2\epsilon} \tag{8}$$

Evaluating the finite difference requires only two forward passes for the weights and two backward passes for α, and the complexity is reduced from $O(|\alpha||w|)$ to $O(|\alpha| + |w|)$.

First-order Approximation When $\xi = 0$, the second-order derivative in equation 7 will disappear. In this case, the architecture gradient is given by $\nabla_\alpha \mathcal{L}_{val}(w, \alpha)$, corresponding to the simple heuristic of optimizing the validation loss by assuming the current w is the same as $w^*(\alpha)$. This leads to some speed-up but empirically worse performance, according to our experimental results in Table 1 and Table 2. In the following, we refer to the case of $\xi = 0$ as the first-order approximation, and refer to the gradient formulation with $\xi > 0$ as the second-order approximation.

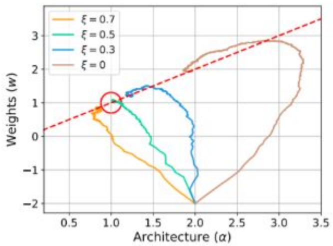
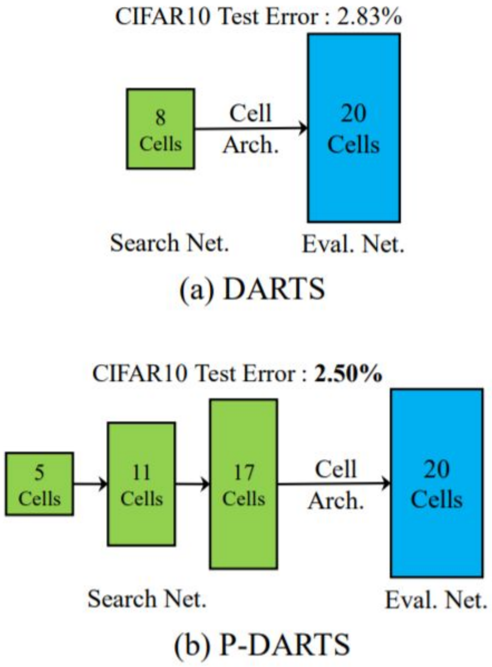


Figure 2: Learning dynamics of our iterative algorithm when $\mathcal{L}_{val}(w, \alpha) = \alpha w - 2\alpha + 1$ and $\mathcal{L}_{train}(w, \alpha) = w^2 - 2\alpha w + \alpha^2$, starting from $(\alpha^{(0)}, w^{(0)}) = (2, -2)$. The analytical solution for the corresponding bilevel optimization problem is $(\alpha^*, w^*) = (1, 1)$, which is highlighted in the red circle. The dashed red line indicates the feasible set where constraint equation 4 is satisfied exactly (namely, weights in w are optimal for the given architecture α). The example shows that a suitable choice of ξ helps to converge to a better local optimum.

HUAWEI

Progressive-DARTS



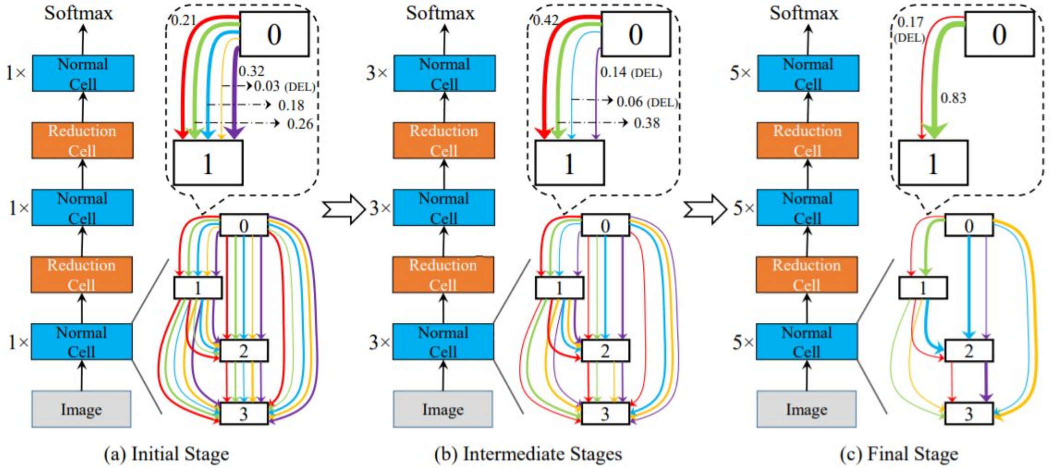
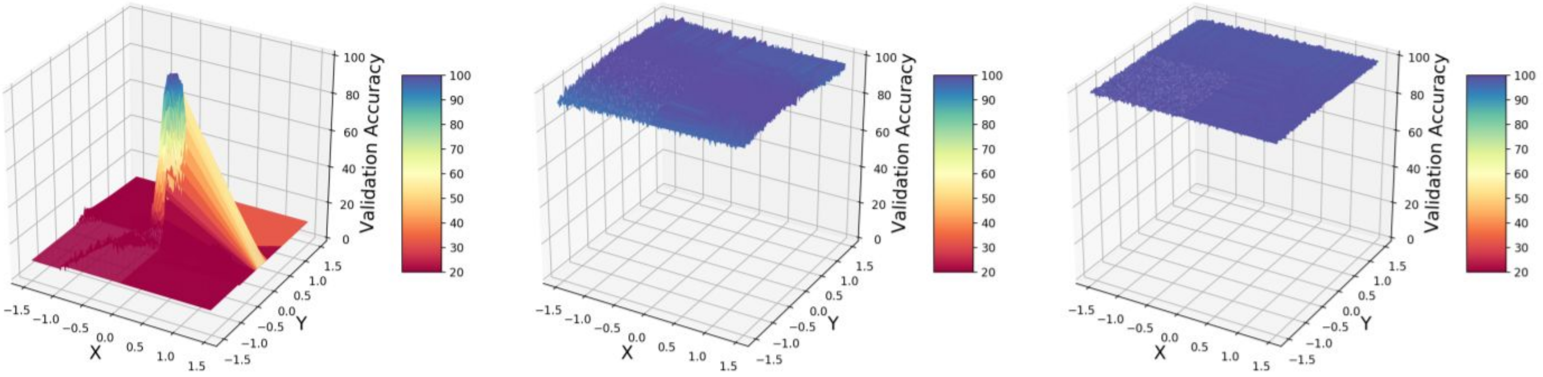


Figure 2: The overall pipeline of P-DARTS (best viewed in color). For simplicity, only one intermediate stage is shown, and only the normal cells are displayed. The depth of the search network increases from 5 at the initial stage to 11 and 17 at the intermediate and final stages, while the number of candidate operations (shown in connections with different colors) is shrunk from 5 to 3 and 2 accordingly. The lowest-scored ones at the previous stage are dropped (the scores are shown next to each connection). We obtain the final architecture by considering the final scores and possibly additional rules.

Smooth-DARTS



(a) DARTS (b) SDARTS-RS (c) SDARTS-ADV

Figure 1: The landscape of validation accuracy regarding the architecture weight A on CIFAR-10. The X-axis is the gradient direction $\nabla_A L_{valid}$, while the Y-axis is another random orthogonal direction (best viewed in color).

- Randomly Smoothed
- Adversarial (minimize worst-case around)

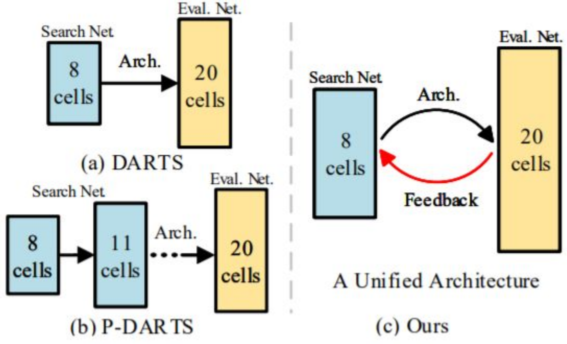
$$\min_A L_{\text{val}}(\bar{w}(A), A), \text{ s.t.} \tag{2}$$

$$\text{SDARTS-RS: } \bar{w}(A) = \arg\min_w E_{\delta \sim U_{[-\epsilon,\epsilon]}} L_{\text{train}}(w, A + \delta)$$

$$\text{SDARTS-ADV: } \bar{w}(A) = \arg\min_w \max_{\|\delta\| \leq \epsilon} L_{\text{train}}(w, A + \delta),$$

HUAWEI https://arxiv.org/pdf/2002.05283.pdf

Cyclic DARTS



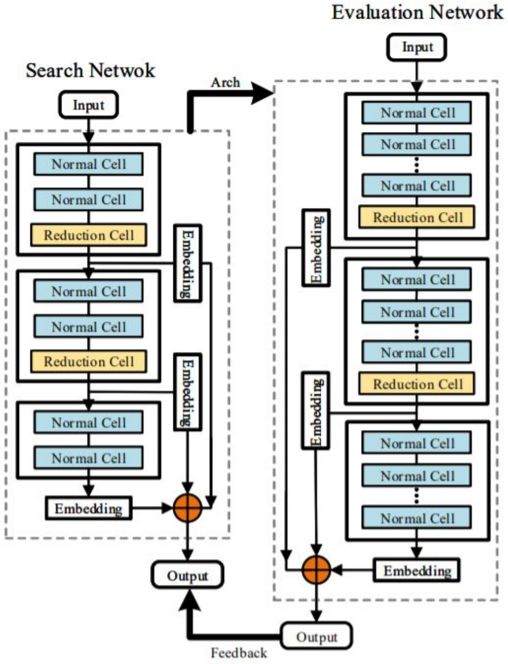


Figure 2: Illustration of the proposed cyclic DARTS. Our model contains two networks, the search network(left) and the evaluation network(right). The *Embedding* module maps each stage feature map to a one-dimensional vector.

Algorithm 1 Cyclic DARTS

Input: The *train* and *val* data, search and evaluation iterations S_S and S_E, update iterations S_U, architecture hyperparameter α, and weights w_S and w_E for search S-Net and evaluation E-Net.

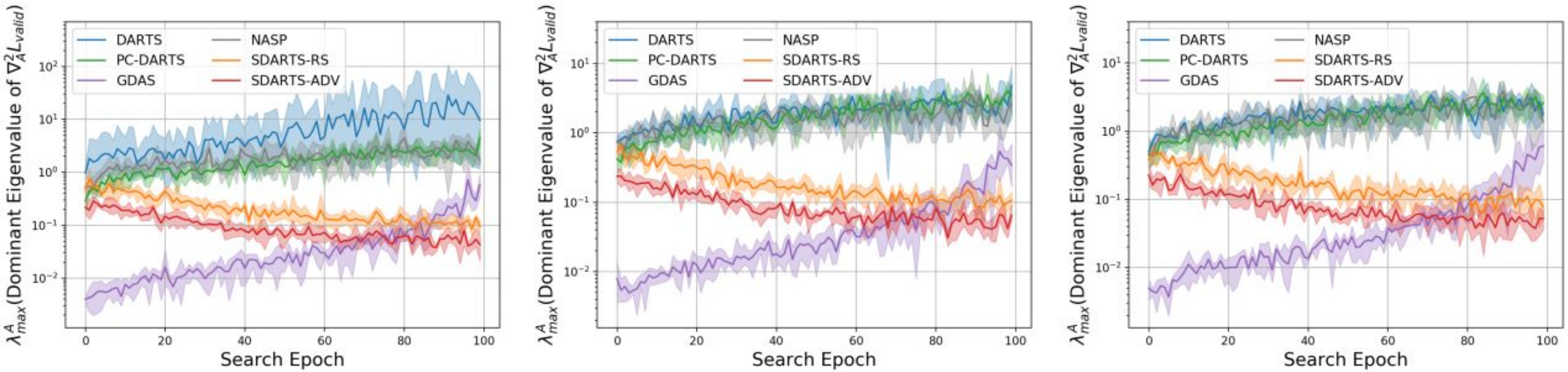
Output: Evaluation network.

```
Initialize α with randoms
Initialize w_S
for each search step i ∈ [0, S_S] do
    if i Mod S_U = 0 then
        Discretize α to ᾱ by selecting the top-k
        Generate E-Net with ᾱ
        for each evaluation step j ∈ [0, S_E] do
            Calculate L^E_val according to Eq.(5)
            Update w_E
        end for
    end if
    Calculate L^S_val, L^E_val and L^{S,E}_val according to Eq.(6)
    Jointly update α and w_E
    Calculate L^S_train according to Eq.(4)
    Update w_S
end for
```

https://arxiv.org/pdf/2006.10724.pdf

HUAWEI

Smooth-DARTS



(a) Space 1 (b) Space 2 (c) Space 3

Figure 5: Trajectory (mean ± std) of the Hessian norm on NAS-Bench-1Shot1 (best viewed in color).

Анализ стабильности

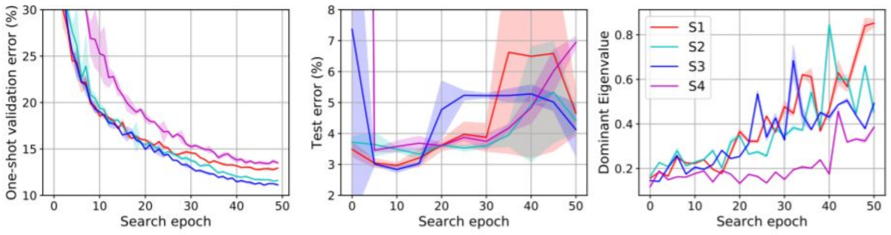


Figure 3: *(left)* validation error of search model; *(middle)* test error of the architectures deemed by DARTS optimal *(right)* dominant eigenvalue of $\nabla^2_\alpha \mathcal{L}_{valid}$ throughout DARTS search. Solid line and shaded areas show mean and standard deviation of 3 independent runs. All experiments conducted on CIFAR-10.

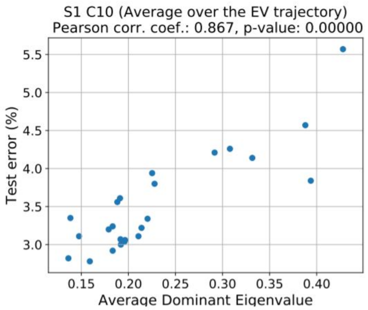
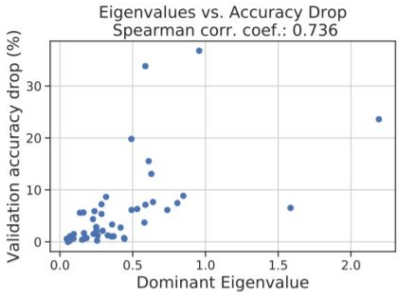
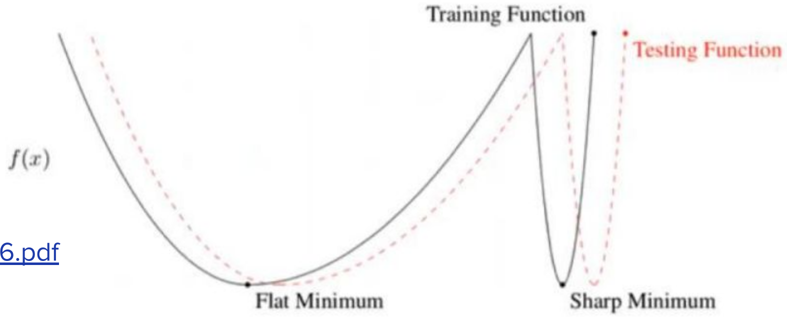


Figure 4: Correlation between dominant eigenvalue of $\nabla^2_\alpha \mathcal{L}_{valid}$ and test error of corresponding architectures.



(b)

Figure 5: (a) Hypothetical illustration of the loss function change in the case of flat vs. sharp minima. (b) Drop in accuracy after discretizing the search model vs. the sharpness of minima (by means of λ^α_{max}).



- Остановка поиска при увеличении собственных значений гессиана
- L2-регуляризация

https://arxiv.org/pdf/1909.09656.pdf

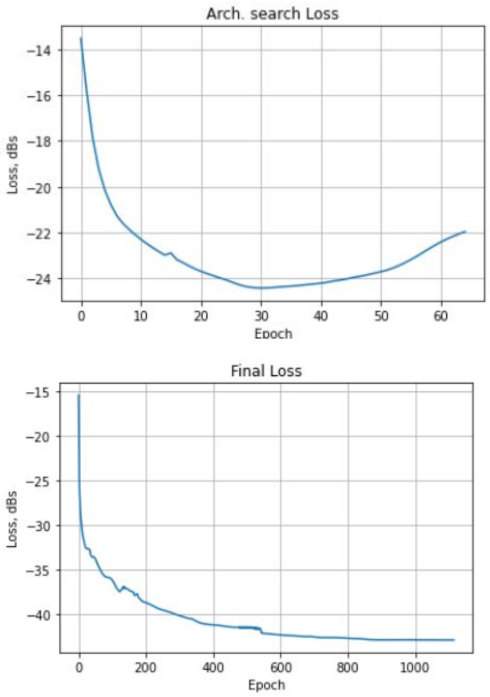
HUAWEI

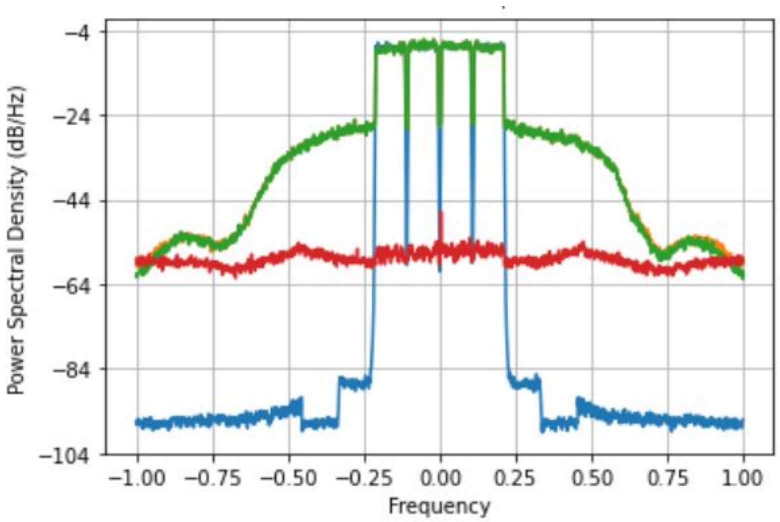
Сравнение

Table 3: Comparison with state-of-the-art architectures on CIFAR10 and CIFAR100. † We use the same search space as DARTS[39].

Architecture	Test Top-1 Acc. (%)		Param	Search Cost	Search
	CIFAR10	CIFAR100	(M)	(GPU days)	Method
Wide ResNet [66]	96.20	82.70	36.5	-	manual
ResNeXt-29, 16x64d [61]	96.42	82.69	68.1	-	manual
DenseNet-BC [27]	96.52	82.82	25.6	-	manual
NASNet-A [69]	-	97.35	3.3	1800	RL
AmoebaNet-B [48]	97.45	-	2.8	3150	evolution
PNAS [37]	96.59	-	3.2	225	SMBO
ENAS [47]	97.11	-	4.6	0.5	RL
NAONet [42]	96.82[1]	84.33	10.6	200	NAO
SNAS (moderate) [62]	97.15 ± 0.02	-	2.8	1.5	gradient
ProxylessNAS [5]	97.92	-	5.7	4	gradient
Random†	96.75 ± 0.18	-	3.4	-	-
DARTSV1 [39]†	97.00 ± 0.14	82.24	3.3	1.5	gradient
DARTSV2 [39]†	97.24 ± 0.09	82.46	3.3	4.0	gradient
PDARTS [6]†	97.50	83.45	3.4	0.3	gradient
PCDARTS [63]†	97.43 ± 0.06	-	3.6	0.1	gradient
FairDARTS [10]†	97.41 ± 0.14	-	3.8	0.1	gradient
CDARTS(Ours)†	97.52 ± 0.04	84.31	3.8	0.3	gradient

DARTS for DPD

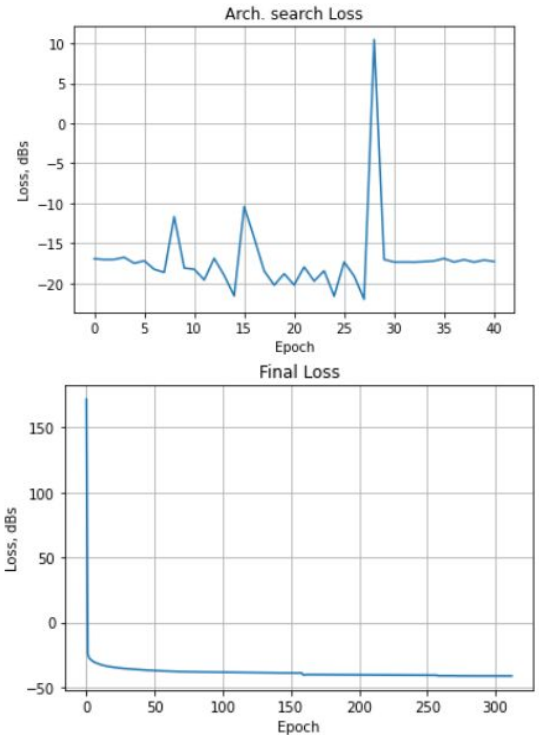


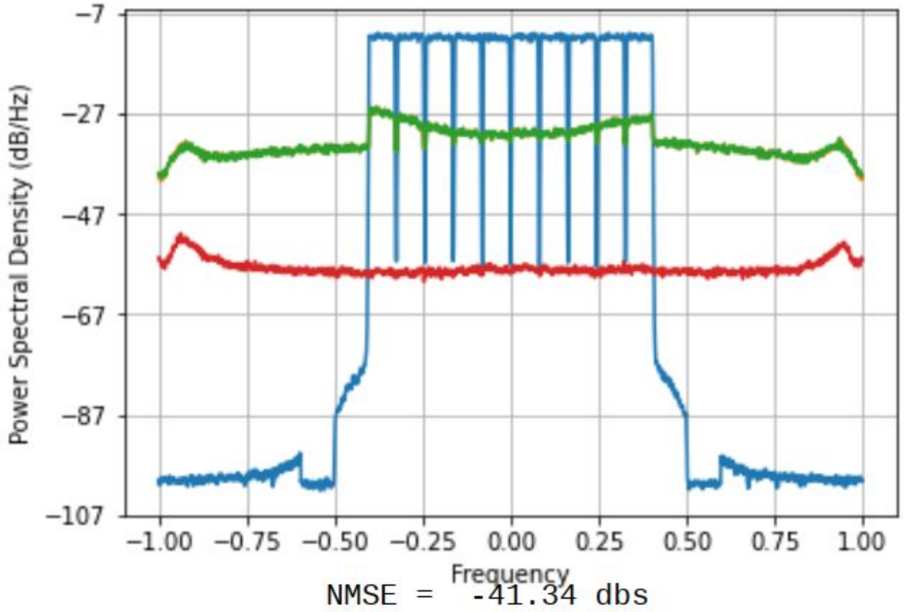


36 FIR, 11 POLY

NMSE = -42.9dbs

DARTS for DPD





Спасибо за внимание