Стохастические и генетические алгоритмы выбора структуры

Московский Физико-Технический Институт

2022

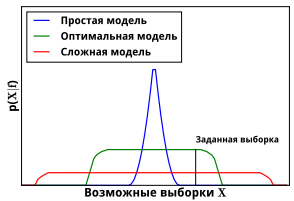
Выбор модели: связанный байесовский вывод со структурой

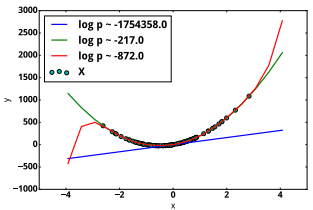
Первый уровень: выбираем оптимальные параметры:

$$\boldsymbol{w} = \arg\max \frac{p(\mathfrak{D}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{h})}{p(\mathfrak{D}|\boldsymbol{h})},$$

Второй уровень: выбираем модель, доставляющую максимум обоснованности модели. Обоснованность модели ("Evidence"):

$$p(\mathfrak{D}|\boldsymbol{h}) = \int_{\boldsymbol{w}} p(\mathfrak{D}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{h})d\boldsymbol{w}.$$





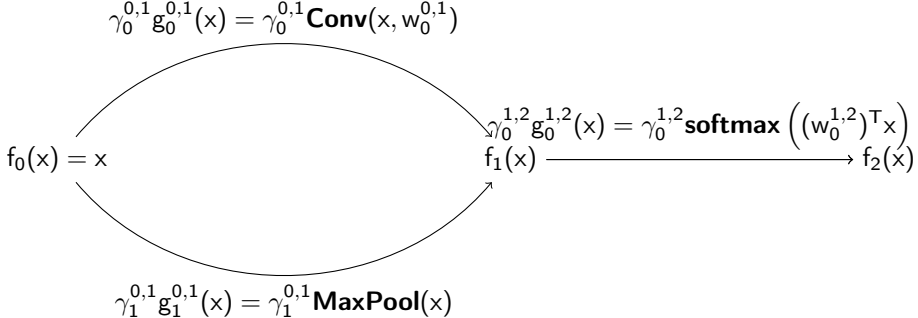
Structure selection: neural architecture search space

The model f is defined by the **structure** $\mathbf{\Gamma} = [\gamma^{0,1}, \gamma^{1,2}]$.

$$\text{Model: } f(x) = \textbf{softmax}\left((w_0^{1,2})^T f_1(x)\right), \quad f(x) : \mathbb{R}^n \to [0,1]^{|\mathbb{Y}|}, \quad x \in \mathbb{R}^n.$$

$$f_1(x) = \gamma_0^{0,1} g_0^{0,1}(x) + \gamma_1^{0,1} g_1^{0,1}(x),$$

where $w = [w_0^{0,1}, w_0^{1,2}]^T$ — parameter matrices, $g_{0,1}^0$ is a convolution,$g_{0,1}^1$ is a pooling operation, $g_{1,2}^0$ is a generalized-linear function.



Deep learning model structure as a graph

Define:

1. acyclic graph (V, E);
2. for each edge $(j, k) \in E$: a vector primitive differentiable functions $\mathrm{g}^{j,k} = [\mathrm{g}_0^{j,k}, \ldots, \mathrm{g}_{K^{j,k}}^{j,k}]$ with length of $K^{j,k}$;
3. for each vertex $v \in V$: a differentiable aggregation function $\mathbf{agg}_v$.
4. a function $\mathrm{f} = \mathrm{f}_{|V|-1}$:

$$\mathrm{f}_v(\mathrm{w}, \mathrm{x}) = \mathbf{agg}_v \left(\{ \langle \boldsymbol{\gamma}^{j,k}, \mathrm{g}^{j,k} \rangle \circ \mathrm{f}_j(\mathrm{x}) | j \in \mathrm{Adj}(v_k) \} \right), v \in \{1, \ldots, |V| - 1\}, \quad \mathrm{f}_0(\mathrm{x}) = \mathrm{x} \tag{1}$$

that is a function from $\mathbb{X}$ into a set of labels $\mathbb{Y}$ for any value of $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$.

Definition

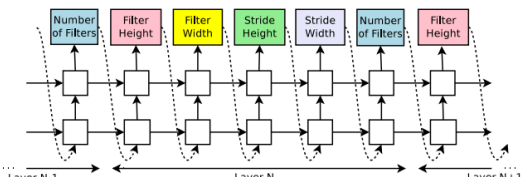
A *parametric set of models* $\mathfrak{F}$ is a graph (V, E) with a set of primitive functions $\{\mathrm{g}^{j,k}, (j, k) \in E\}$ and aggregation functions $\{\mathbf{agg}_v, v \in V\}$.

Statement

A function $\mathrm{f} \in \mathfrak{F}$ is a model for each $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$.

Neural Architecture Search with Reinforcement Learning

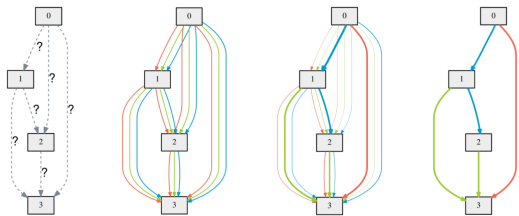
Структура выбирается контроллером. В цикле выбора структуры производится полная оптимизация параметров модели.



DARTS

Модель — мультиграф, где ребра $[g^e]$ соответствуют подмоделям, а вершины $f_v(x)$ — результату действия подмоделей на выборку.
Результат применения подмоделей:

$$f_v = \langle \gamma, \textbf{softmax}([g^e(x)]) \rangle.$$



ENAS

Двухэтапная оптимизация:

1. Оптимизация параметров
   - Фиксируем параметры контроллера
   - На каждом шаге сэмплируем структуру $\mathbf{\Gamma}$
   - Производим оптимизацию параметров $\boldsymbol{w} = \arg\min L$
2. Оптимизация параметров контроллера
3. Оптимизация параметров
   - Фиксируем параметры модели $\boldsymbol{w}$
   - Оптимизируем параметры контроллера

ENAS vs DARTS

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

Почему работает генетика и стохастика?

Схема — генотип или подмножество генотипов, задающийся бинарной строкой с маской вида: $[001 * 1]$, где маска $*$ указывает, что нам не важен данный признак.

Теорема схем

$$N(h, t+1) \geq N(h, t) \frac{Q(h, t)}{\mathrm{E}_h Q(h, t)} (1 - p),$$

h — схема , $N(h, t)$ — количество схем h на шаге t, p — вероятность уничтожения схемы под действием генетических операторов.

- Схемы с большей маской более живучи
- Схемы с большей маской и качеством выше среднего более живучи

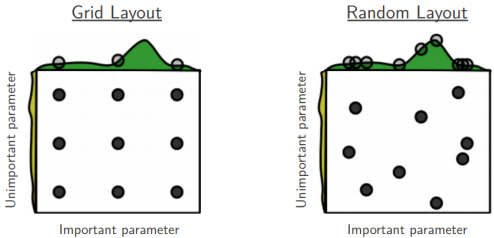
Наивные методы поиска структуры

Варианты:

- Поиск по решетке;
- Случайный поиск.

Оба метода страдают от проклятия размерности.

Случайный поиск может быть более эффективным, если пространство гиперпараметров вырождено.



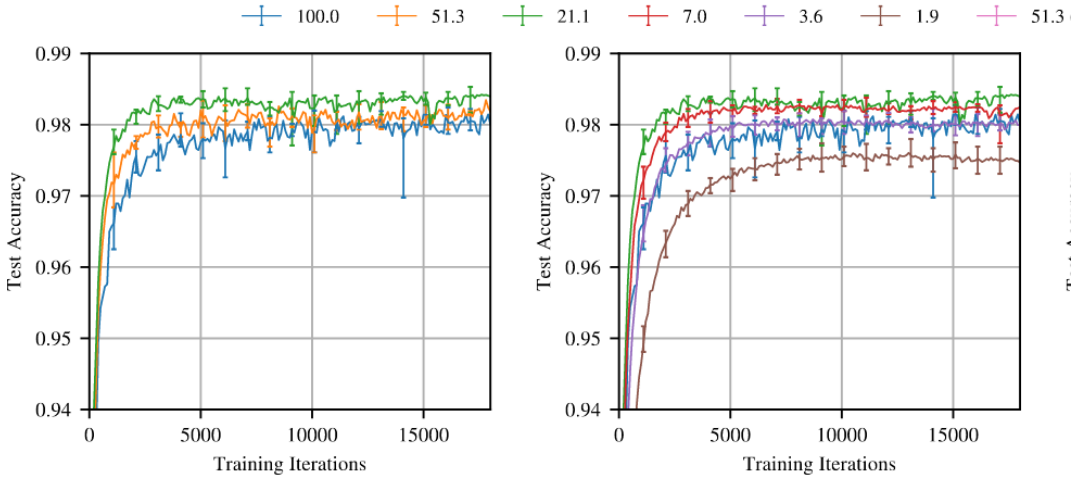
Bergstra et al., 2012

Lottery ticket

The Lottery Ticket Hypothesis, Frankle et al., 2019

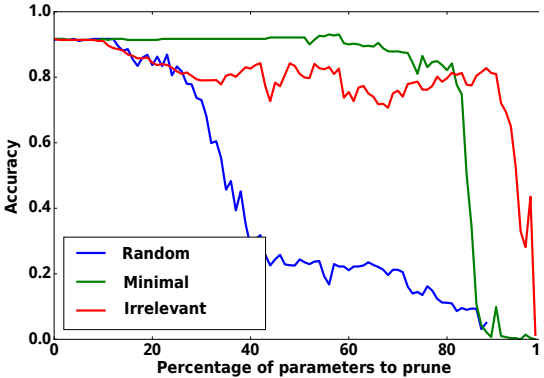
A randomly-initialized, dense neural network contains a subnet- work that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.

Lottery ticket: MNIST

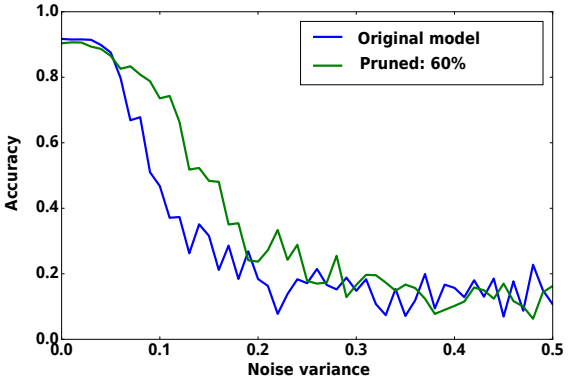


Model structure selection challenge

Data likelihood does not change with removing redundant parameters.



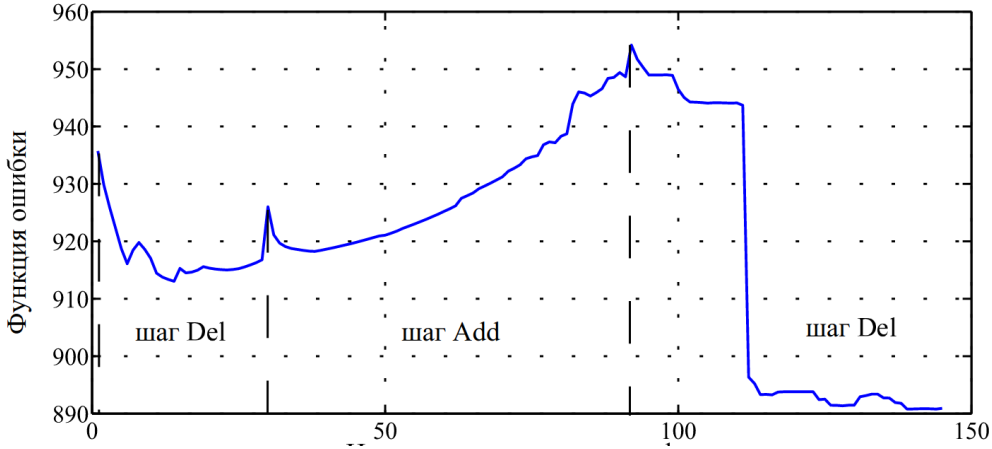
Redundancy of model parameters



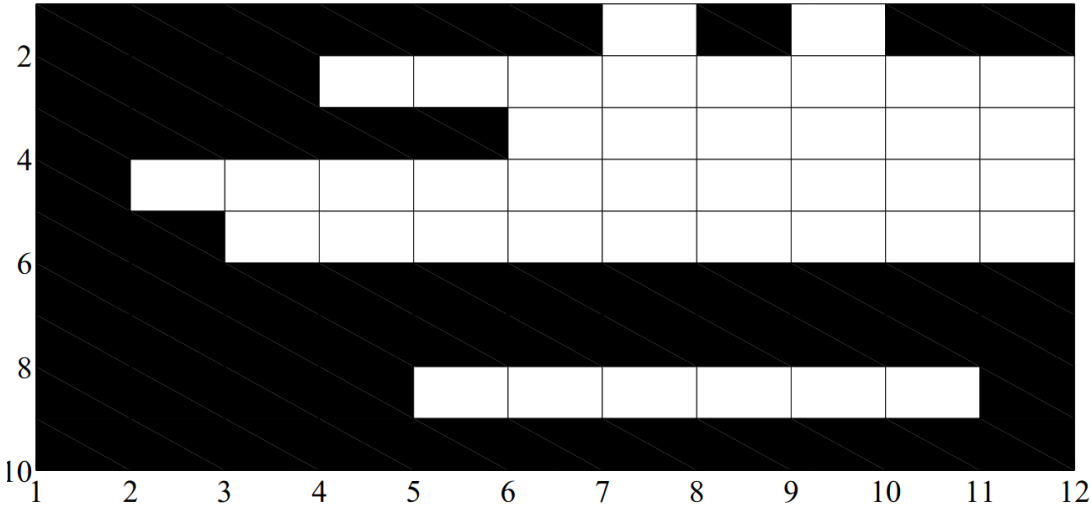
Model robustness

Deep learning models have implicitly redundant complexity.

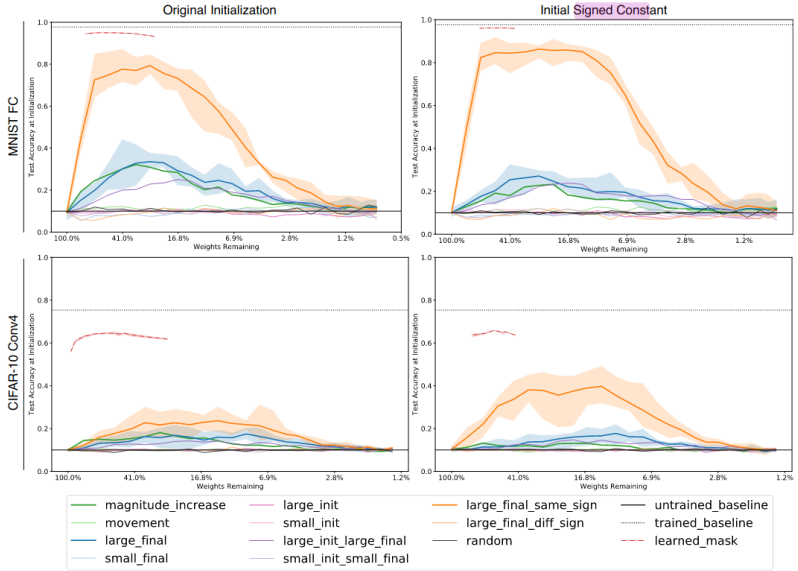
AddDel: Попова, Стрижов, 2015



AddDel: Попова, Стрижов, 2015



Lottery ticket: Zhou et al., 2019

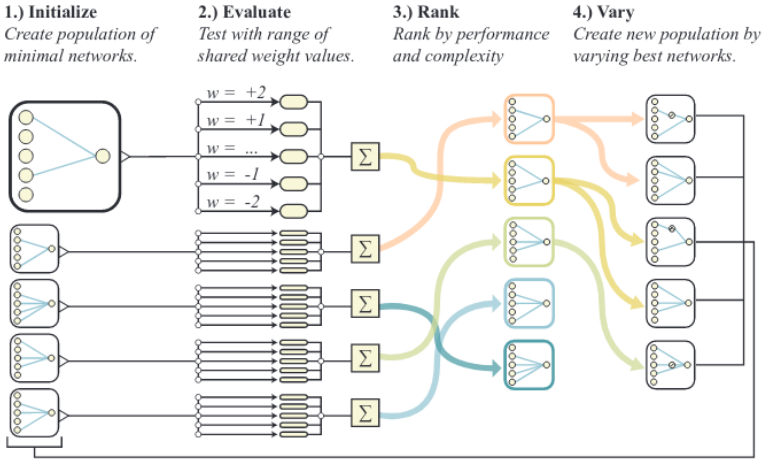


Weight agnostic NN, Gaier et al., 2019

Модель $\boldsymbol{f}$ — нейронная сеть с константными параметрами и структурой $\boldsymbol{\Gamma}$, задающей связи между нейронами и активации.

Никакого бэкпропа, только эволюционные алгоритмы!

WANN


1.) Initialize
Create population of minimal networks.
2.) Evaluate
Test with range of shared weight values.
3.) Rank
Rank by performance and complexity
4.) Vary
Create new population by varying best networks.
w = +2
w = +1
w = ...
w = -1
w = -2
Σ
Σ
Σ
Σ
Σ

WANN

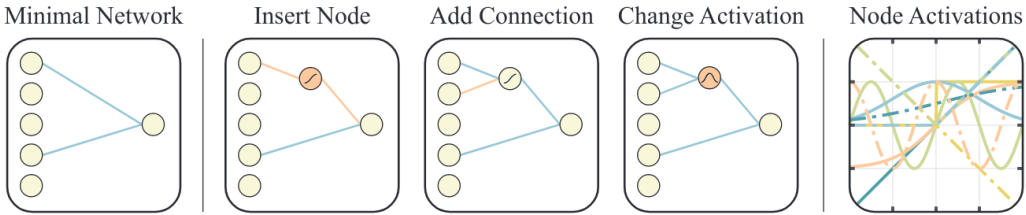


Figure 3: *Operators for Searching the Space of Network Topologies*
Left: A minimal network topology, with input and outputs only partially connected.
Middle: Networks are altered in one of three ways. *Insert Node*: a new node is inserted by splitting an existing connection. *Add Connection*: a new connection is added by connecting two previously unconnected nodes. *Change Activation*: the activation function of a hidden node is reassigned.
Right: Possible activation functions (linear, step, sin, cosine, Gaussian, tanh, sigmoid, absolute value, invert (i.e. negative linear), ReLU) shown over the range $[2, 2]$.

WANN

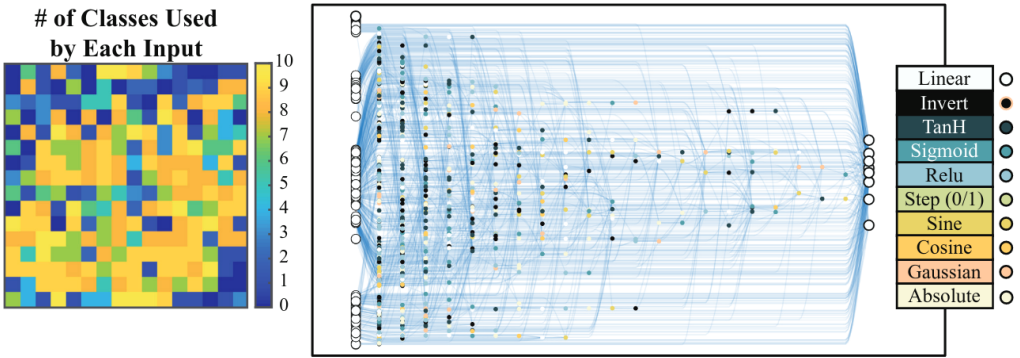


Figure 7: *MNIST classifier network (1849 connections)*

WANN

WANN	Test Accuracy
Random Weight	*82.0% ± 18.7%*
Ensemble Weights	*91.6%*
Tuned Weight	*91.9%*
Trained Weights	*94.2%*

ANN	Test Accuracy
Linear Regression	*91.6%* [62]
Two-Layer CNN	*99.3%* [15]

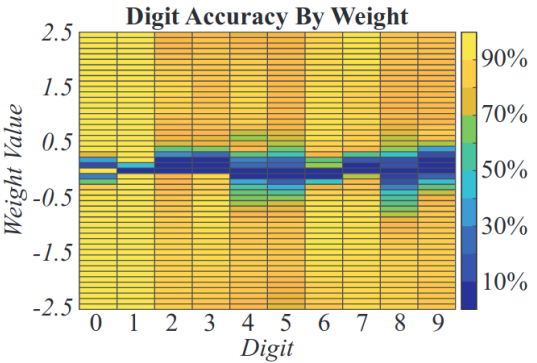
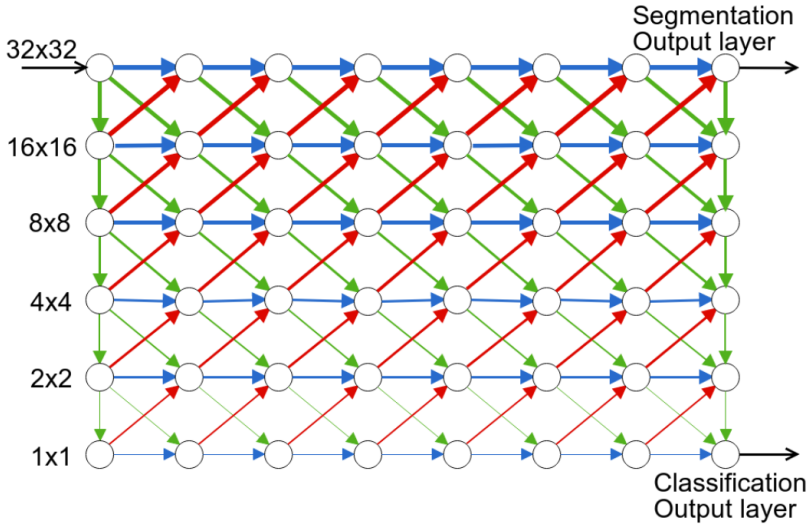


Figure 6: *Classification Accuracy on MNIST.*

Суперсети


32x32
16x16
8x8
4x4
2x2
1x1
Segmentation
Output layer
Classification
Output layer

PathNet

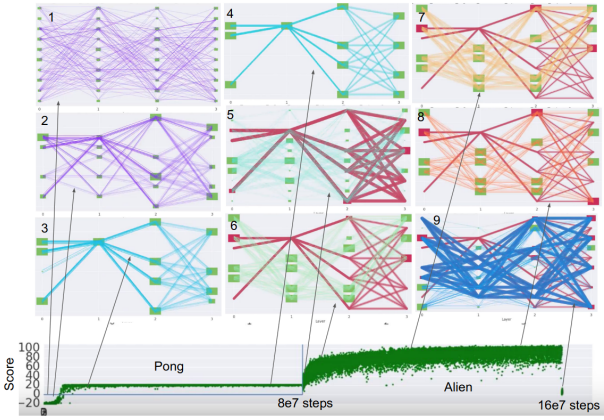
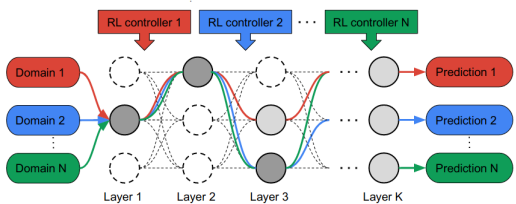


Figure 1: A population of randomly initialized pathways (purple lines in Box 1) are evolved whilst learning task A, Pong. At the end of training, the best pathway is fixed (dark red lines in Box 5) and a new population of paths are generated (light blue lines in Box 5) for task B. This population is then trained on Alien and the optimal pathway that is evolved on Alien is subsequently fixed at the end of training, shown as dark blue lines in Box 9.

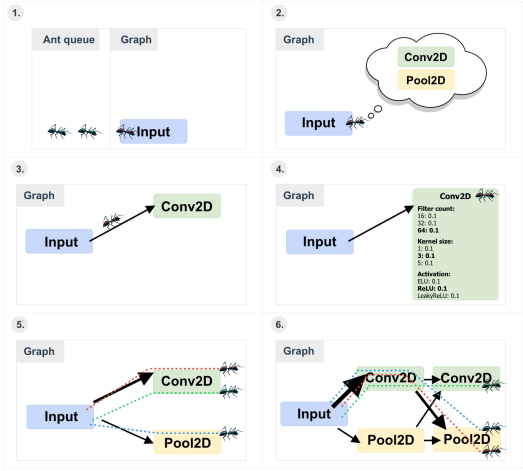
MultiPath

Algorithm 1: Multi-path neural architecture search

Result: Multi-path network
Initialize RLControllers;
Initialize super-network from the search space;
for $Epochs = 1 : MaxEpochs$ **do**
  for $i = 1 : DomainCount$ **do**
    Sample one path for Domain[i] to form model;
    Run model on validation set to get Reward[i];
    Run model on training set to get TrainLoss[i];
  end
  Backprop the joint TrainLoss to update model coefficients in super-network;
  for $i = 1 : DomainCount$ **do**
    Update RLControllers[i] with Reward[i] using REINFORCE;
  end
end



DeepSwarm


1.
Ant queue
Graph
Input
2.
Graph
Conv2D
Pool2D
Input
3.
Graph
Input
Conv2D
4.
Graph
Input
Conv2D
Filter count:
16: 0.1
32: 0.1
64: 0.1
Kernel size:
1: 0.1
3: 0.1
5: 0.1
Activation:
ELU: 0.1
ReLU: 0.1
LeakyReLU: 0.1
5.
Graph
Input
Conv2D
Pool2D
6.
Graph
Input
Conv2D
Conv2D
Pool2D
Pool2D

Литература

- Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. – New York : springer, 2006. – Т. 4. – №. 4. – С. 738.
- Бахтеев О. Ю. Байесовский выбор субоптимальной структуры модели глубокого обучения, диссертация
- Теорема схем: https://engineering.purdue.edu/~sudhoff/ee630/Lecture03.pdf
- Zoph B., Le Q. V. Neural architecture search with reinforcement learning //arXiv preprint arXiv:1611.01578. – 2016.
- Liu H., Simonyan K., Yang Y. Darts: Differentiable architecture search //arXiv preprint arXiv:1806.09055. – 2018.
- Pham H. et al. Efficient neural architecture search via parameters sharing //International conference on machine learning. – PMLR, 2018. – С. 4095-4104.
- Frankle J., Carbin M. The lottery ticket hypothesis: Finding sparse, trainable neural networks //arXiv preprint arXiv:1803.03635. – 2018.
- Zhou H. et al. Deconstructing lottery tickets: Zeros, signs, and the supermask //Advances in neural information processing systems. – 2019. – Т. 32.
- Gaier A., Ha D. Weight agnostic neural networks //Advances in neural information processing systems. – 2019. – Т. 32.
- Попова М.С., Стрижов В.В. Построение сетей глубокого обучения для классификации временных рядов // Системы и средства информатики, 2015, 25(3) : 60-77.
- Veniat T., Denoyer L. Learning time/memory-efficient deep architectures with budgeted super networks //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – С. 3492-3500.
- Fernando C. et al. Evolution Channels Gradient Descent in Super Neural Networks //arXiv preprint arXiv:1701.08734. – 2017. – С. 1-16.
- Go A. M. et al. Multi-path Neural Networks for On-device Multi-domain Visual Classification. – 2021.
- Byla E., Pang W. Deepswarm: Optimising convolutional neural networks using swarm intelligence //UK Workshop on Computational Intelligence. – Springer, Cham, 2019. – С. 119-130.