Stochastic and evolutionary structure selection methods

MIPT

2023

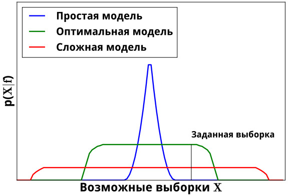
Model selection

First level: select optimal parameters:

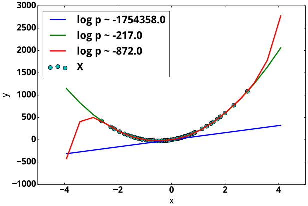
$$\mathrm{w} = \arg\max \frac{p(\mathfrak{D}|\mathrm{w})p(\mathrm{w}|\mathrm{h})}{p(\mathfrak{D}|\mathrm{h})},$$

Second level: select model optimizing Evidence:

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



Model selection scheme



Example

Structure selection: one-layer network

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

$$\text{Model: } \mathbf{f}(\mathbf{x}) = \mathbf{softmax}\left((\mathbf{w}_0^{1,2})^{\mathsf{T}}\mathbf{f}_1(\mathbf{x})\right), \quad \mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to [0,1]^{|\mathbb{Y}|}, \quad \mathbf{x} \in \mathbb{R}^n.$$

$$\mathbf{f}_1(\mathbf{x}) = \gamma_0^{0,1}\mathbf{g}_0^{0,1}(\mathbf{x}) + \gamma_1^{0,1}\mathbf{g}_1^{0,1}(\mathbf{x}),$$

where $\mathbf{w} = [\mathbf{w}_0^{0,1}, \mathbf{w}_1^{0,1}, \mathbf{w}_0^{1,2}]^{\mathsf{T}}$ — parameter matrices, $\{\mathbf{g}_{0,1}^0, \mathbf{g}_{0,1}^1, \mathbf{g}_{1,2}^0\}$ — generalized-linear functions, alternatives of layers of the network.

$$\begin{tikzcd}
\mathbf{f}_0(\mathbf{x}) = \mathbf{x} \arrow[r, bend left, "\gamma_0^{0,1}\mathbf{g}_0^{0,1}(\mathbf{x}) = \gamma_0^{0,1}\boldsymbol{\sigma}\left((\mathbf{w}_0^{0,1})^{\mathsf{T}}\mathbf{x}\right)"] \arrow[r, bend right, "\gamma_1^{0,1}\mathbf{g}_1^{0,1}(\mathbf{x}) = \gamma_1^{0,1}\boldsymbol{\sigma}\left((\mathbf{w}_1^{0,1})^{\mathsf{T}}\mathbf{x}\right)"'] & \mathbf{f}_1(\mathbf{x}) \arrow[r, "\gamma_0^{1,2}\mathbf{g}_0^{1,2}(\mathbf{x}) = \gamma_0^{1,2}\mathbf{softmax}\left((\mathbf{w}_0^{1,2})^{\mathsf{T}}\mathbf{x}\right)"] & \mathbf{f}_2(\mathbf{x})
\end{tikzcd}$$

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 $\mathrm{g}^{j,k} = [\mathrm{g}_0^{j,k}, \dots, \mathrm{g}_{K^{j,k}}^{j,k}]$ with length of $K^{j,k}$;
3. for each vertex $v \in V$: a differentiable aggregation function $\mathbf{agg}_v$.
4. a function $\mathrm{f} = \mathrm{f}_{|V|-1}$:

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

   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

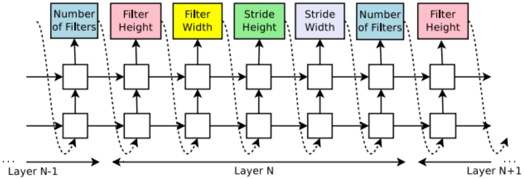
A *parametric set of models* $\mathfrak{F}$ is a graph (V, E) with a set of primitive functions $\{\mathrm{g}^{j,k}, (j, k) \in E\}$ and aggregation functions $\{\mathbf{agg}_v, v \in V\}$.

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

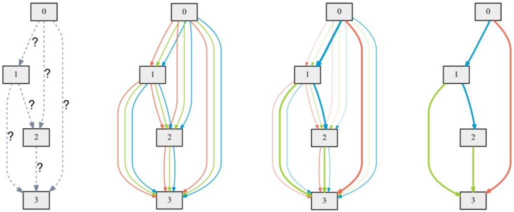
The structure is selected using controllet.
The optimization of model parameters is conducted in a loop of structure selection.



DARTS

The model is a multigraph, where edges $[g^e]$ correspond to submodels, vertices $f_v(x)$ are the results of submodels:

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



ENAS

Two-stage optimization:

1. Parameter optimization
   - Fix controller parameters
   - At each optimization step sample a structure $\mathbf{\Gamma}$
   - Optimize model parameters: $\mathrm{w} = \arg\min L$
2. Controller optimization
   - Fix model parametersw
   - Optimize controller parameters.

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

Why does stochasticity work?

Scheme is a genotype or a subset of genotypes, which are encoded by binary string with mask in the format $[001 * 1]$. Mask symbol $*$ tells us that the corresponding feature is not important in this genotype.

Holland's schema theorem

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

where h is a scheme, N is an amount of schemes, p is a probability of scheme destroy w.r.t. genetic operators, Q is a quality.

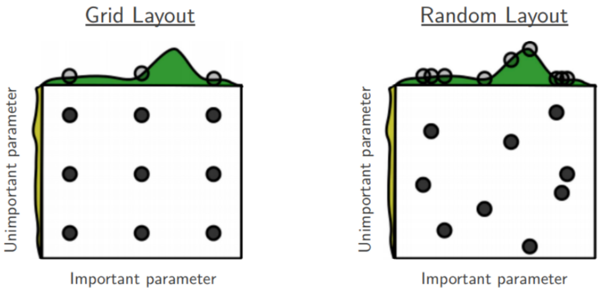
- Schemes with larger mask are more vitable;
- Schemes with larger mask and quality better than average are more vitable.

Naive structure search

Variants:

- Grid search;
- Random search;

Both methods suffer from curse of dimensionality. The random search is more effective if the search space is degenerate.



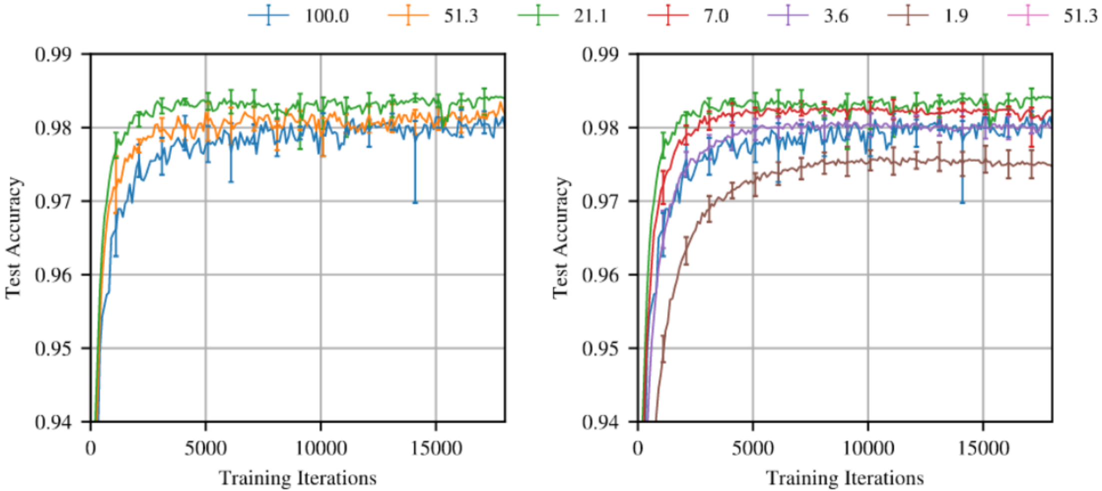
Bergstra et al., 2012

Lottery ticket

The Lottery Ticket Hypothesis, Frankle et al., 2019

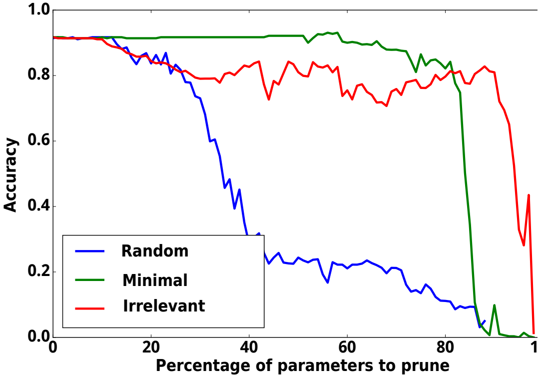
A randomly-initialized, dense neural network contains a subnet- work that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.

Lottery ticket: MNIST

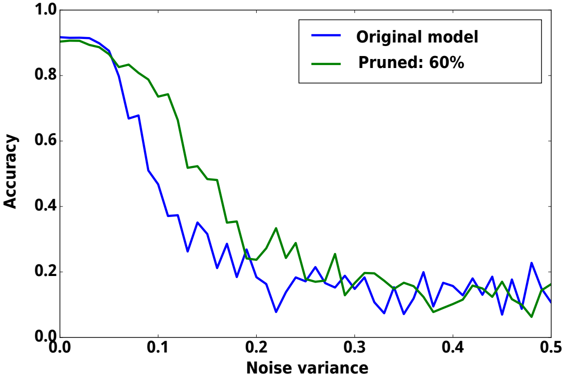


Model structure selection challenge

Data likelihood does not change with removing redundant parameters.



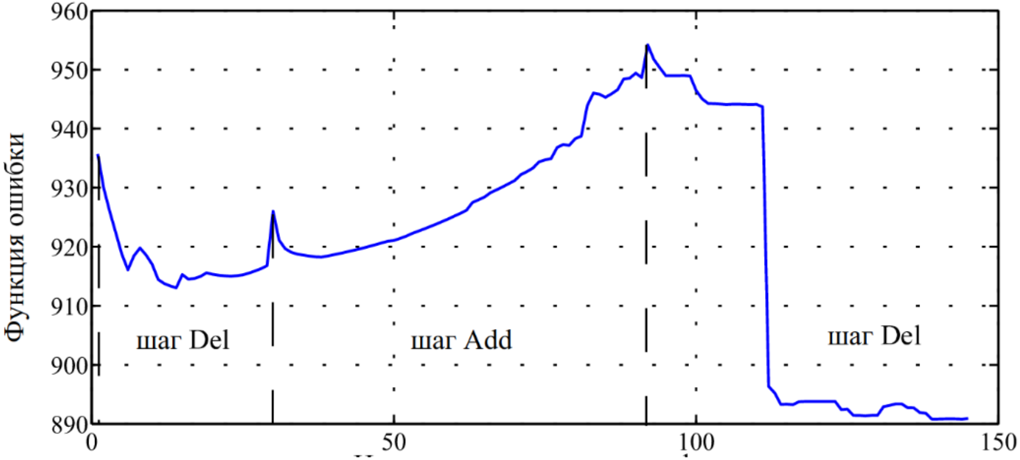
Redundancy of model parameters



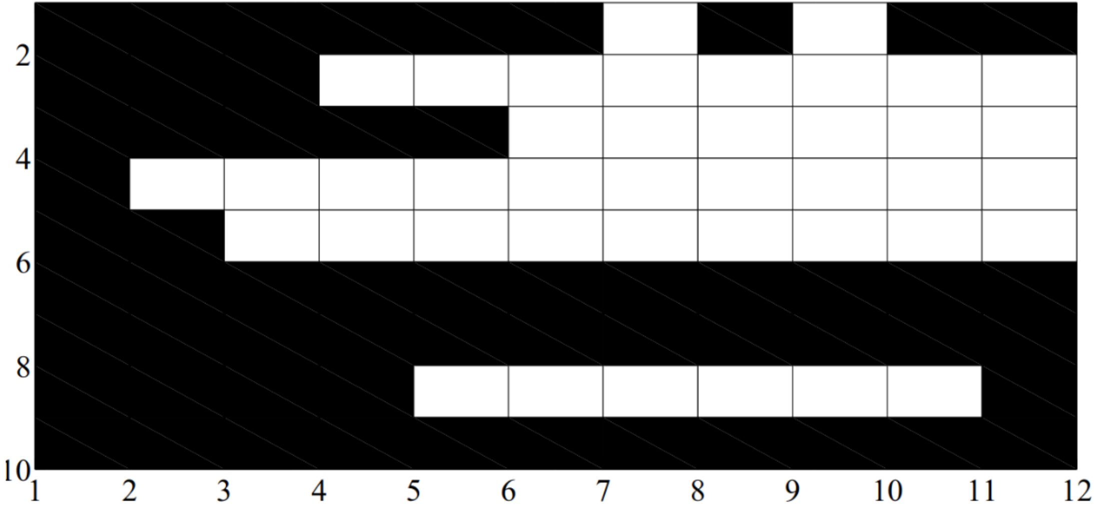
Model robustness

Deep learning models have implicitly redundant complexity.

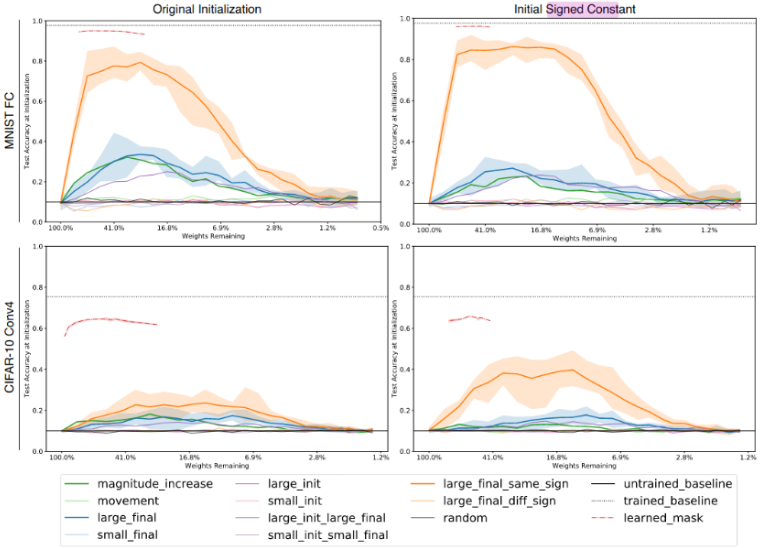
AddDel: Popova, Strijov, 2015



AddDel



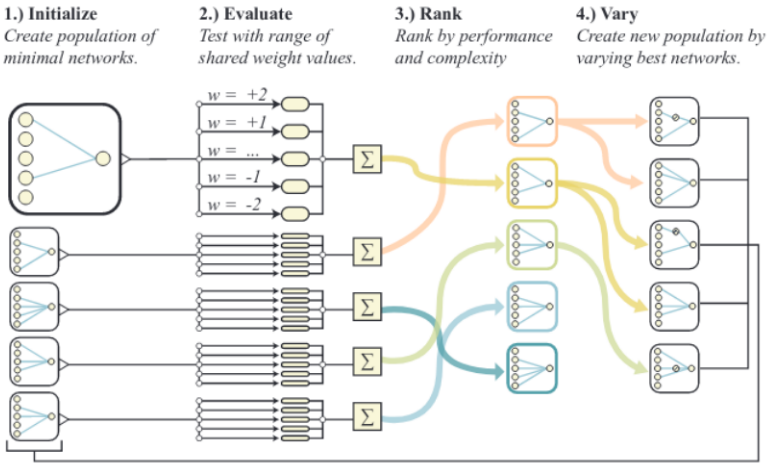
Weight agnostic NN, Gaier et al., 2019



Weight agnostic NN, Gaier et al., 2019

Model f is a MLP with constant parameters and a structure $\mathbf{\Gamma}$ that assigns connections between neurons and activations.
No backprop, only evolutionary algorithms!

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

WANN

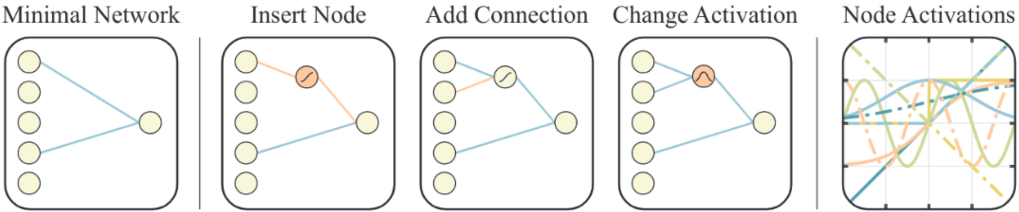


Figure 3: *Operators for Searching the Space of Network Topologies*
Left: A minimal network topology, with input and outputs only partially connected.
Middle: Networks are altered in one of three ways. *Insert Node*: a new node is inserted by splitting an existing connection. *Add Connection*: a new 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

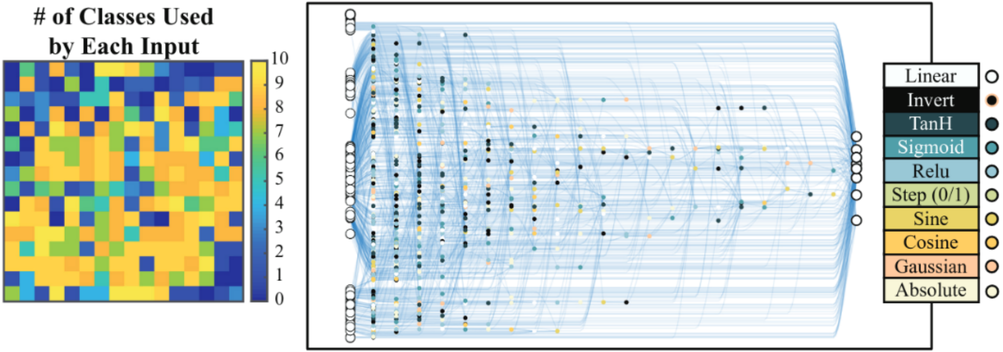


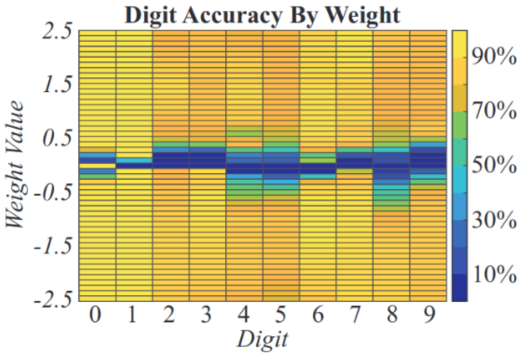
Figure 7: *MNIST classifier network (1849 connections)*

WANN

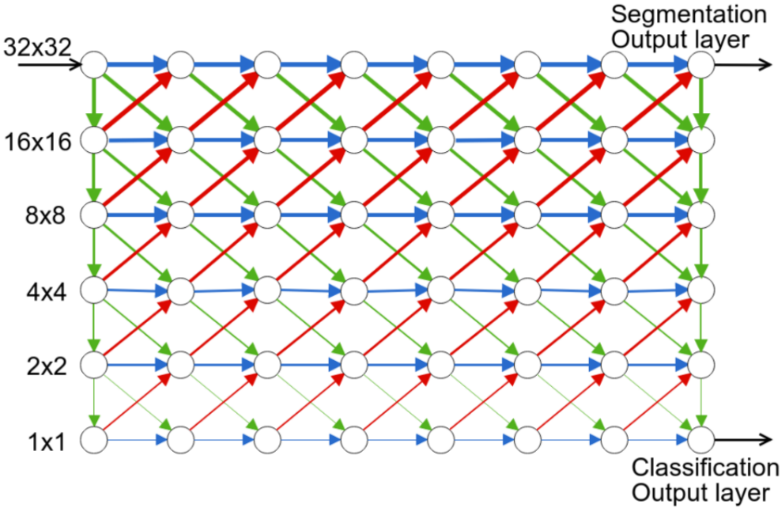
WANN	Test Accuracy
Random Weight	*82.0% ± 18.7%*
Ensemble Weights	*91.6%*
Tuned Weight	*91.9%*
Trained Weights	*94.2%*

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

Figure 6: *Classification Accuracy on MNIST.*



Supernetworks



PathNet

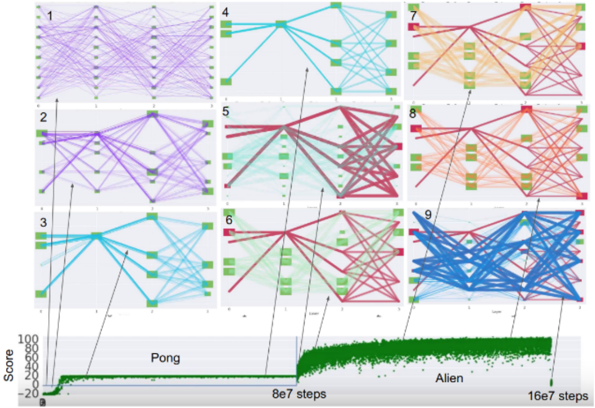
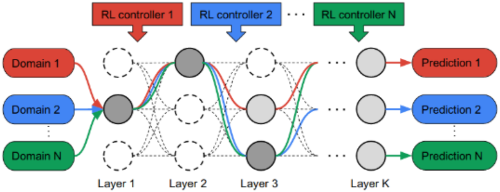


Figure 1: A population of randomly initialized pathways (purple lines in Box 1) are evolved whilst learning task A, Pong. At the end of training, the best pathway is fixed (dark red lines in Box 5) and a new population of paths are generated (light blue lines in Box 5) for task B. This population is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

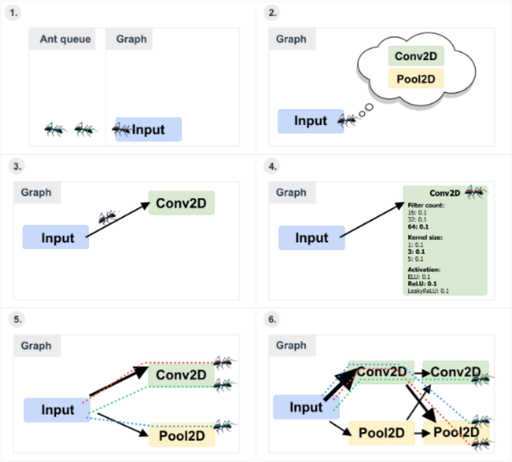
Multipath

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

References

- Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. – New York : springer, 2006. – T. 4. – No. 4. – C. 738.
- Бахтеев О. Ю. Байесовский выбор субоптимальной структуры модели глубокого обучения, диссертация
- Scheme theorem: 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. – C. 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. – T. 32.
- Gaier A., Ha D. Weight agnostic neural networks //Advances in neural information processing systems. – 2019. – T. 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. – C. 3492-3500.
- Fernando C. et al. Evolution Channels Gradient Descent in Super Neural Networks //arXiv preprint arXiv:1701.08734. – 2017. – C. 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. – C. 119-130.