UDC 004.023+519.876.5

MODULAR SEARCH SPACE FOR AUTOMATED DESIGN OF NEURAL ARCHITECTURE

Radiuk P.M.

Khmelnytskyi National University,
11 Instytuts'ka St., 29016, Ukraine, Khmelnytskyi.
radiukpavlo@gmail.com

МОДУЛЬНИЙ ПРОСТІР ПОШУКУ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ НЕЙРОННОЇ АРХІТЕКТУРИ

Радюк П.М.

Хмельницький національний університет,
29016, Україна, м. Хмельницький, вул. Інститутська, 11.
radiukpavlo@gmail.com

МОДУЛЬНОЕ ПРОСТРАНСТВО ПОИСКА ДЛЯ АВТОМАТИЗИРОВАННОГО ПРОЭКТИРОВАНИЯ НЕЙРОННОЙ АРХИТЕКТУРЫ

Радюк П.М.

Хмельницкий национальный университет,
29016, Украина, г. Хмельницкий, ул. Институтская, 11.
radiukpavlo@gmail.com

Abstract. The past years of research have shown that automated machine learning and neural architecture search are an inevitable future for image recognition tasks. In addition, a crucial aspect of any automated search is the predefined search space. As many studies have demonstrated, the modularization technique may simplify the underlying search space by fostering successful blocks' reuse. In this regard, the presented research aims to investigate the use of modularization in automated machine learning. In this paper, we propose and examine a modularized space based on the substantial limitation to seeded building blocks for neural architecture search. To make a search space viable, we presented all modules of the space as multisectoral networks. Therefore, each architecture within the search space could be unequivocally described by a vector. In our case, a module was a predetermined number of parameterized layers with information about their relationships. We applied the proposed modular search space to a genetic algorithm and evaluated it on the CIFAR-10 and CIFAR-100 datasets based on modules from the NAS-Bench-201 benchmark. To address the complexity of the search space, we randomly sampled twenty-five modules and included them in the database. Overall, our approach retrieved competitive architectures in averaged 8 GPU hours. The final model achieved the validation accuracy of 89.1% and 73.2% on the CIFAR-10 and CIFAR-100 datasets, respectively. The learning process required slightly fewer GPU hours compared to other approaches, and the resulting network contained fewer parameters to signal lightness of the model. Such an outcome may indicate the considerable potential of sophisticated ranking approaches. The conducted experiments also revealed that a straightforward and transparent search space could address the challenging task of neural architecture search. Further research should be undertaken to explore how the predefined knowledge base of modules could benefit modular search space.

Key words: search space, modularization, automl, neural architecture search, genetic algorithm.

Анотація. За минулі роки дослідження підтвердили, що автоматизоване машинне навчання та пошук архітектури нейронної мережі – це неминуче майбутнє для завдань розпізнавання зображень. Крім того, одним із вирішальних аспектів будь-якого автоматизованого пошуку виявився попередньо визначений простір пошуку. Як показали багато обчислювальних досліджень, техніка модуляризації здатна спростити базовий простір пошуку, сприяючи повторному використанню успішних блоків. У зв'язку з цим, ця наукова стаття має на меті дослідити використання модуляризації в автоматизованому машинному навчанні. У цій статті ми пропонуємо та оцінюємо модульований простір, з огляду на істотне обмеження попередньо визначених блоків для пошуку архітектури. Щоб зробити простір пошуку істотним, ми показали всі модулі простору, як багато секторальні мережі. Тому кожну архітектуру в просторі пошуку однозначно описано вектором. У нашому випадку модуль є заздалегідь заданою кількістю параметризованих шарів з інформацією про їхні взаємозв'язки. Ми застосували запропонований модульний простір до генетичного алгоритму та оцінили його на наборах даних CIFAR-10 та CIFAR-100 на основі модулів з еталонного тесту NAS-Bench-201. Щоб розглянути складність простору пошуку, ми випадковим чином відібрали двадцять п'ять модулів та включили їх у базу даних. Загалом, наш підхід знайшов конкурентні архітектури в середньому за 8 GPU годин. Кінцева модель досягла точності перевірки 89,1 % та 73,2 % на наборах даних CIFAR-10 та CIFAR-100 відповідно. Процес навчання зайняв дещо меншу кількість графічних годин порівняно з іншими підходами, а отримана мережа містила менше параметрів, сигналізуючи про легкість моделі. Такий результат вказує на значний потенціал складних підходів до ранжування. Проведені експерименти також виявили, що простий і зрозумілий простір пошуку може бути застосований до складного завдання пошуку нейронної архітектури. Має сенс провести подальші дослідження, щоб вивчити, як попередньо визначена база знань модулів може посприяти модульному простору пошуку.

Ключові слова: простір пошуку, модуляризація; automl, пошук архітектури нейронної мережі, генетичний алгоритм.

Аннотация. За прошедшие годы исследования подтвердили, что автоматическое машинное обучение и поиск архитектуры нейронной сети являются неизбежным будущим для задач распознавания образов. Кроме того, одним из важнейших аспектов любого автоматизированного поиска оказалось предопределенное пространство поиска. Как показали многие исследования, технология модуляризации может упростить основное пространство поиска, способствуя успешному повторному использованию блоков. В связи с этим, представленная научная статья направлена на исследование использования модуляризации в автоматизированном машинном обучении. В данной статье мы предлагаем и оцениваем модульное пространство, основанное на существенном ограничении предопределенных блоков для поиска архитектуры. Для того, чтобы сделать пространство поиска существенным, мы предоставили все модули пространства в виде многоотраслевой сети. Поэтому каждая архитектура внутри пространства поиска может быть однозначно описана отдельным вектором. В нашем случае модуль представляет предопределенное количество параметризованных слоев с информацией об их соотношениях. Мы применили предложенное модульное пространство к генетическому алгоритму и оценили его на наборах данных CIFAR-10 и CIFAR-100 на основе модулей из эталонного теста NAS-Bench-201. Чтобы рассмотреть проблему сложности пространства поиска, мы случайным образом выбрали двадцать пять модулей и включили их в базу данных. В целом, наш подход позволил получить конкурентные архитектуры в среднем за 8 GPU часов. Окончательная модель достигла валидационной точности 89,1% и 73,2% на наборах данных CIFAR-10 и CIFAR-100 соответственно. Процесс обучения потребовал чуть меньше GPU-часов по сравнению с другими подходами, а полученная сеть содержала меньше параметров, сигнализируя о легкости модели. Такой результат может показать значительный потенциал сложных подходов к ранжированию. Проведенные эксперименты также показали, что простое и прозрачное пространство поиска может быть применим к сложной задаче поиска архитектур нейронной сети. Имеет смысл провести дальнейшие исследования для изучения того, каким образом предопределенная база знаний модулей может способствовать модульному пространству поиска.

Ключевые слова: пространство поиска, модуляризация, automl, поиск архитектуры нейронной сети, генетический алгоритм.

Introduction. Over the past decade, the field of automated search and optimization of machine learning models (AutoML) has become widely recognized. Neural architecture search (NAS), the research branch of AutoML, has achieved astonishing results in image recognition and classification tasks [1]. However, conventional NAS techniques rely on a massive amount of processing power and machine memory [2], as numerous architecture candidates must be trained and evaluated during the search process. In this regard, the efficient search task has become an increasing priority in NAS over the past few years [3, 4].

Among other aspects of efficiency, the size and complexity of a search space are the most influential. As recent works [5, 6] prove, even conventionally small and light search spaces can provide decent results comparable to algorithms that require much larger computational resources. Consequently, the open question remains as to how complex and extensive the search space's structure and its objects can be. To tackle this issue, we investigate a search space formed by five modules sampled from NAS-Bench-201 [7] and then combined using regularized evolution [8].

Related works. A cell-based cyclic structure [9] has been a commonly utilized type of search space in NAS. The size of the search space directly depends on the set and combinations of hyperparameters; for instance, the original NASNet configuration comprises 10^{12} architectures. Even in the case of sequential models, all possible parameter combinations within a layer enlarges the total number of possible architectures to an order of 10^{54} [10]. Thus, due to the potential redundancy, the search space's cell-based structure was rejected within the current study. Instead, we delineate the search space by introducing a modular architecture structure to provide a more efficient search.

The concept of modularity has been introduced in other aspects of NAS. For example, Zhang et al. [12] implemented the scalable modules concerning the width and depth of a single. In [13], the authors divided the search space into blocks with fixed external and flexible internal structures. Negrinho et al. [14] presented modules in which each layer of a block receives an array of hyperparameters while defining the search space. In [15], the researchers proposed modules as blocks, each of which comprised the multilevel combination of convolutional and pooling layers.

Overall, the mentioned above studies support the notion that modularization may be an excellent solution for search space. Therefore, we aim to expand previously presented approaches by further reducing the complexity of modular search space.

The problem statement. In this study, we propose and examine a search space based on predefined modules that could efficiently search for optimal architecture. The current application requires minimal modifications and adaptions owing to the straightforward concept of our approach. The modular search space examined in this study was inspired by [14].

Let us consider the issue of NAS for the target dataset $D = \{D_{train}, D_{val}\}$, where D_{train} and D_{val} stand for training and validation datasets, respectively. The target objective can be considered a two-level multipurpose optimization problem. In this case, the function of multiobjective bilevel optimization might be presented as follows

$$a_{opt} = \min_{A \in \mathbf{A}} \left\{ L_{val}\left(A, w^*\right), C\left(A\right) \right\}, \tag{1}$$

subject to

$$w^* \in \arg\min_{w \in \mathbf{W}} \left\{ L_{train}\left(w, a\right) \right\},$$

where a_{opt} stands for the optimized architecture among valid ones A from the search space of all possible architectures $\mathbf{A}$; $C(A)$ is the complexity function of the optimized architecture; w represents the weights of the neural network from the weight space $\mathbf{W}$; L_{train} and L_{val} are loss functions on D_{train} and D_{val}, respectively.

Now, let us consider a NAS procedure like (1), where a combination of modules $m \in \mathrm{M}$ determines a valid architecture $A \in \mathbf{A}$. Let us also assume that each $m \in \mathrm{M}$ is conducted among each other and the structure of the architecture A is defined by a vector

$$\vec{m} = \left(m_1, m_2, \ldots, m_{n-1}, m_n\right),$$

where n limits the number of modules in an architecture.

Therefore, the main goal of the presented study is to find a vector $\vec{m}$ that performs the best for

$$A^*_{\vec{m}} \in \arg\min_{m \in \mathrm{M}} \left\{ F\left(A_{\vec{m}}\left(D_{train}\right), D_{val}\right) \right\},$$

where the vector $\vec{m}$ denotes the architecture $A_{\vec{m}} \in \mathrm{A}$, and each module $m \in \vec{m}$ is stacked consistently.

Modular neural architecture search. The primary aim of NAS implies an efficient selection of well-functioning neural architectures by searching in a sequential search space. In the case of modular search space, all modules are presented as multisectoral networks. Thus, each architecture within the search space can be unequivocally described by a separate vector. In our case, a module is a predefined number of parameterized layers with information about their relationships with each other. Each module represents a possible change in the internal modular search space. The total number of combinations of arrangements is defined as

$$l^c \cdot c!,$$

where l stands for the number of possible layers, and c is the number of layers per module.

Each module's complexity is high and comparable to the complexity of other non-modular search spaces. We represented each module as a randomly generated acyclic graph of layers combined within the search sequence. Fig. 1 reveals the conceptual scheme of the examined modular search space.

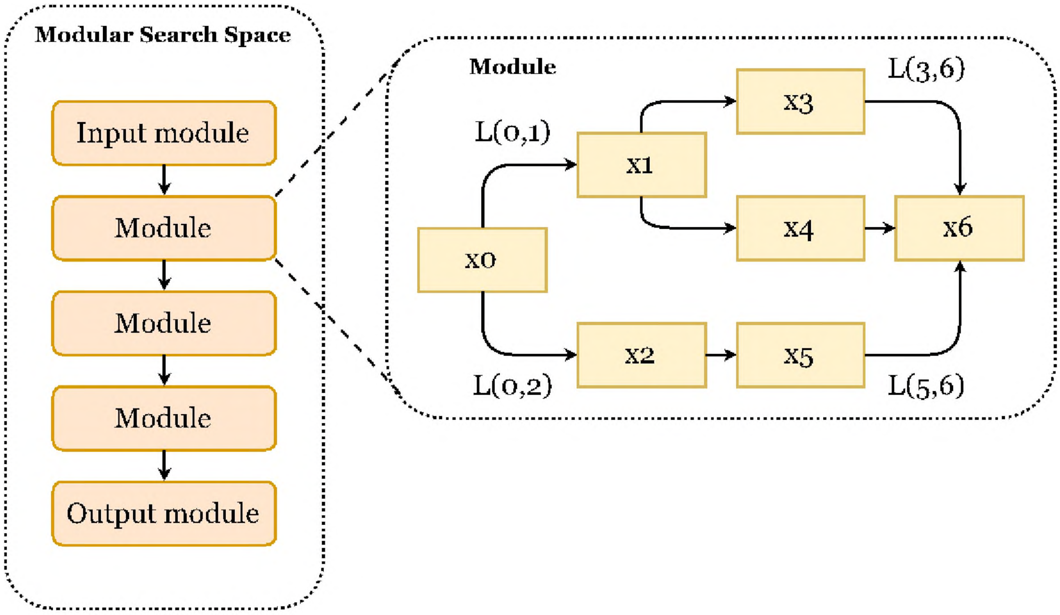

Figure 1 – The group of modules within the evaluated search space

The search problem of the presented study is limited to an existing pool of modules. Each milestone result comprises the previous module's output data, which are then processed in the current module. With this regard, let us consider that the vector $\vec{m}$ specifies the combination of the consistently attached modules. Then, the final architecture gained from $\vec{m}$ can be presented as a composition of modules m_i, $i = \overline{1,n}$:

$$A_{\vec{m}} = m_1 \circ m_2 \circ \ldots \circ m_i \circ \ldots \circ m_{n-1} \circ m_n .$$

The complexity of the examined search space is calculated as follows

$$S = \sum_{i=1}^{n} |M|^i ,$$

where M stands for the set of modules, and n here is the permitted number of sequential combined modules.

As a search strategy, we utilized a regularized genetic algorithm inspired by [8]. The regularized algorithm aimed to search for an optimal architecture regarding the accuracy of the classification. The search space was also limited to the selection of modules and their sequence.

Implementation details.

Datasets. In this work, we investigate the modular search space for the classification task on two well-known benchmark datasets, CIFAR-10 and CIFAR-100 [16].

Module database. The set of modules was based on the NAS-Bench-201 interface presented by [7]. Following the guidelines from [7], the modules were randomly generated via a superordinate parameters matrix. The examined search space was restricted by convolutional layers of sizes of 5×5 and 3×3 [17]. Pooling layers were set of the size of 2×2 [18]. Moreover, the search space was also limited to the modules' size, where each module consisted of seven layers.

Search space complexity. To address the complexity of the search space, we randomly sampled twenty-five modules and included them in the database. The search algorithm was designed to select either five random modules as input or the best five modules ranked based on CIFAR-10 (which is directly available via NAS-Bench-201) limited to the time of 8 GPU hours. The simple structure of the presented search space allowed a straightforward influence of its size.

Algorithm details. In this work, we set up the following parameters of the genetic algorithm: population size of 4 and a sample size of 2, according to [19], and the probability of mutations for regularized evolution of 0.8 [8]. We also utilized an input filter size of 32, 60 training epochs, and a batch size of 64 for each training procedure [20]. According to [7], filter sizes were gradually increased, and dimensions instead – decreased. We also applied the weight sharing technique as instructed in [11] to ensure the efficient training of a mutated child. Fig. 2 depicts the general scheme of modular neural architecture search.

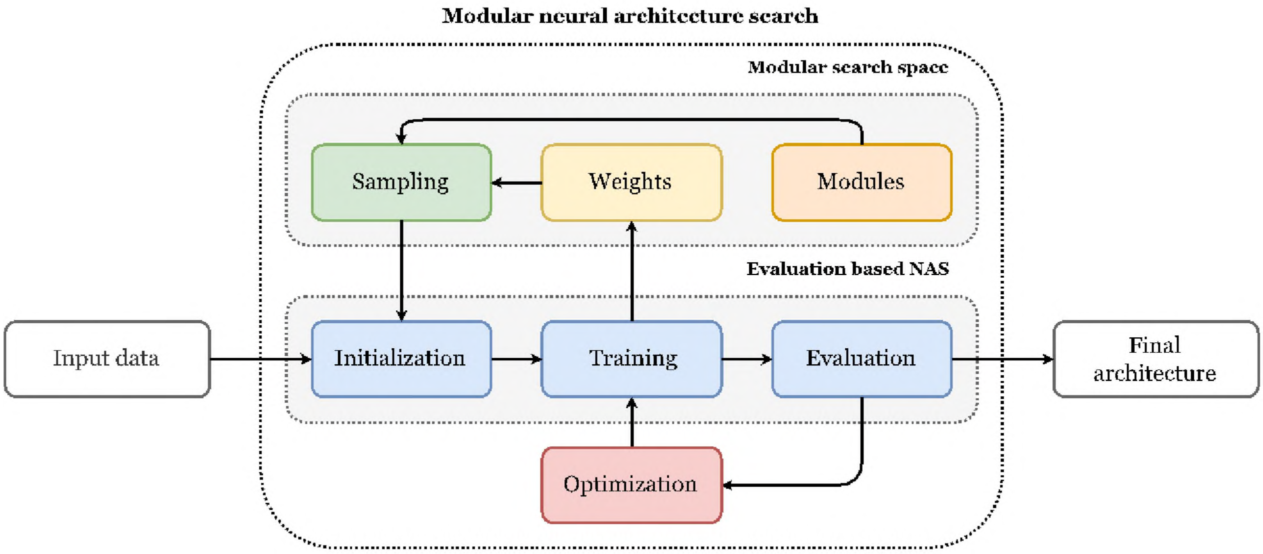

Figure 2 – The conceptual framework of the modular NAS

Results and discussion. The results of the examination based on the random initialization of 5 modules are presented herein. In addition to examining the proposed modular search space, we compared it to the two most recognized approaches in the search space designing. Table 1 shows the results of CIFAR-10.

Table 1 – The results on the CIFAR-10 dataset

Approach	Time, GPU hours	Number of parameters, mil	Validation accuracy
[12]	12,34	19,2	0,937
[13]	9,5	15,71	0,963
[14]	10,17	14,98	0,901
Our modular search space	7,89	9,39	0,891

As seen from Table 1, our approach's validation accuracy, which is 89,1%, is below the competition, yet remains efficient considering the search time of almost 8 GPU hours and the restricted number of modules. Also, it is worth noting the potential impact of small search spaces with learned ratings on performance. Using a few ranked modules can result in increased productivity. Table 2 introduces the computational results of CIFAR-100.

Table 2 – The results on the CIFAR-100 dataset

Approach	Time, GPU hours	Number of parameters, mil.	Validation accuracy
[12]	16,89	25,6	0,768
[13]	14,3	26,11	0,792
[14]	9,25	20,79	0,774
Our modular search space	8,07	18,78	0,732

According to Table 2, the examined modular search achieved the validation accuracy of 73,2%, which was worse compared to the state-of-the-art. Such an outcome could occur due to the limited module choice to 3 paired with the initial filter size of 32, therefore, resulting in restricted architectures. Nevertheless, the training process required slightly fewer GPU hours, and the resulting network contained fewer parameters, which led to the less weight of the model.

Even though modular search space achieved lower classification accuracy than state-of-the-art, it won in computational cost and weight. Besides, the conducted examination demonstrated that the complex and challenging task of NAS could be addressed by a straightforward and easy to understand search space. Direct comparison of accuracy can be considered only as of the first indicator due to a significant reduction in module size, module size limitation, and low parameters in the search algorithm.

Conclusion. The presented study aims to examine the modularization of the search space in NAS. To be specific, we substantially limited the predefined building blocks within modules. The experiments revealed that the small set of random choice and ranked modules could produce efficient architectures in averaged 8 GPU hours. Our genetic NAS approach achieved the validation accuracy of 89.1% and 73.2% on the CIFAR-10 and CIFAR-100 datasets, respectively.

The conducted examination promised the modular approach's potential benefits, such as smooth weight sharing integration and the ranking of available modules. Furthermore, a dynamic set of modules may result in promising architectures for varied tasks, such as medical image classification and segmentation. To search for modules less arbitrarily, the use of meta-learning should also be considered. A reasonable approach might be to create a module database and a knowledge base of module potentials for diverse datasets.

REFERENCES:

1. Elsken, Thomas, et al. "Neural architecture search: A survey." Journal of Machine Learning Research, vol. 20, no. 55, 2019, pp. 1–21.

2. Ren, Pengzhen, et al. A comprehensive survey of neural architecture search: Challenges and solutions. Preprint, submitted 1 June 2020, arXiv.org.

3. Elsken, Thomas, et al. "Efficient multiobjective neural architecture search via Lamarckian evolution." Proceedings of the 7th International Conference on Learning Representations, ICLR-2019, Preprint, submitted 24 April 2018, arXiv.org.

4. Wong, Catherine, et al. "Transfer learning with neural AutoML." Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS-2018, edited by Samy Bengio and Hanna M Wallach, Curran Associates Inc., 2018, pp. 8366–75.

5. Veniat, Tom, and Ludovic Denoyer. "Learning time/memory-efficient deep architectures with budgeted super networks." Proceedings of the 31st 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR-2018, IEEE Inc., 2018, pp. 3492–500, DOI:10.1109/CVPR.2018.00368

6. Tan, Mingxing, and Quoc Le. "EfficientNet: Rethinking model scaling for convolutional neural networks." Proceedings of the 36th International Conference on Machine Learning, edited by Kamalika Chaudhuri and Ruslan Salakhutdinov, ICML-2019, vol. 97, PMLR.org, 2019, pp. 6105–14.

7. Dong, Xuanyi, and Yi Yang. "NAS-Bench-201: Extending the scope of reproducible neural architecture search." Proceedings of the 8th International Conference on Learning Representations, ICLR-2020, Preprint, submitted 15 Jan. 2020, arXiv.org.

8. Real, Esteban, et al. "Regularized evolution for image classifier architecture search." Proceedings of the 2019 AAAI Conference on Artificial Intelligence, IAAI-2019, vol. 33, no. 1, AAAI Press., 2019, pp. 4780–89, DOI:10.1609/aaai.v33i01.33014780.

9. Zoph, B., et al. "Learning transferable architectures for scalable image recognition." Proceedings of the 31st 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR-2018, IEEE Inc., 2018, pp. 8697–710, DOI:10.1109/CVPR.2018.00907.

10.Bello, Irwan, et al. "Neural optimizer search with reinforcement learning." Proceedings of the 34th International Conference on Machine Learning, ICML-2017, edited by Doina Precup and Yee Whye Teh, JMLR.org, 2017, pp. 459–468, DOI:10.5555/3305381.3305429

11.Pham, Hieu, et al. "Efficient neural architecture search via parameters sharing." Proceedings of the 35th International Conference on Machine Learning, ICML-2018, edited by Jennifer Dy and Andreas Krause, vol. 80, JMLR.org, 2018, pp. 4095–104.

12.Zhang, Xingcheng, et al. "PolyNet: A pursuit of structural diversity in very deep networks." Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR-2017, vol. 2017-Janua, IEEE Inc., 2017, pp. 3900–08, DOI:10.1109/CVPR.2017.415.

13.Zhong, Zhao, et al. "Practical block-wise neural network architecture generation." Proceedings of the 31st 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR-2018, IEEE Inc., 2018, pp. 2423–32, DOI:10.1109/CVPR.2018.00257.

14.Negrinho, Renato, et al. "Towards modular and programmable architecture search." Proceedings of the 33rd Conference on Neural Information Processing Systems, NIPS-2019, edited by H Wallach et al., Curran Associates Inc., 2019, pp. 13715–25.

15.Wei, Tao, et al. "Modularized morphing of deep convolutional neural networks: A graph approach." IEEE Transactions on Computers, Preprint, submitted 16 April 2020, IEEE Inc., 2020, DOI:10.1109/TC.2020.2988006.

16.Krizhevsky, Alex. “Learning multiple layers of features from tiny images.”Master’s Thesis, The University of Toronto, 2009. 60 p.

17.Romanuke, Vadim. “An attempt of finding an appropriate number of convolutional layers in CNNs based on benchmarks of heterogeneous datasets.” Electrical, Control and Communication Engineering, vol. 14, no. 1, Sciendo, 2018, pp. 51–57, DOI:10.2478/ecce-2018-0006

18.Romanuke, Vadim. “Appropriate number of standard 2 × 2 max pooling layers and their allocation in convolutional neural networks for diverse and heterogeneous datasets.” Information Technology and Management Science, vol. 20, no. 1, Sciendo, 2017, pp. 12–19, DOI:10.1515/itms-2017-0002

19.Radiuk, Pavlo. “Application of a genetic algorithm to search for the optimal convolutional neural network architecture.” Herald of Khmelnytskyi National University, vol. 281, no. 1, 2020, pp. 7–11, DOI:10.31891/2307-5732-2020-281-1-7-11.

20.Radiuk, Pavlo. “Impact of training set batch size on the performance of convolutional neural networks for diverse datasets.” Information Technology and Management Science, vol. 20, no. 1, Sciendo, 2017, pp. 20–24, DOI:10.1515/itms-2017-0003.

ЛІТЕРАТУРА:

1. Elsken T. Neural architecture search: A survey / T. Elsken, J. H. Metzen, F. Hutter // Journal of Machine Learning Research. – 2019. – Vol. 20, No. 55. – P. 1–21.

2. Ren P. A comprehensive survey of neural architecture search: Challenges and solutions / P. Ren, Y. Xiao, X. Chang // Preprint. – 2020.

3. Elsken T. Efficient multiobjective neural architecture search via Lamarckian evolution / T. Elsken, J. H. Metzen, F. Hutter // Proceedings of the 7th International Conference on Learning Representations, ICLR-2019, New Orleans, LA, USA, 6–9 May 2019. – P. 1–23.

4. Wong C. Transfer learning with neural AutoML / C. Wong, N. Houlsby, Y. Lu // Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS-2018, Montréal, QC, Canada, 2-8 Dec. 2018, Curran Associates Inc. – P. 8366–8375.

5. Veniat T. Learning time/memory-efficient deep architectures with budgeted super networks / T. Veniat, L. Denoyer // Proceedings of the 31st 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR-2018, Salt Lake City, UT, USA, 18–23 June 2018, IEEE Inc. – P. 3492–3500. DOI:10.1109/CVPR.2018.00368

6. Tan M. EfficientNet: Rethinking model scaling for convolutional neural networks / M. Tan, Q. Le // Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, USA, PMLR.org. – P. 6105–6114.

7. Dong X. NAS-Bench-201: Extending the scope of reproducible neural architecture search / X. Dong, Y. Yang // Proceedings of the 8th International Conference on Learning Representations, ICLR-2020, Addis Ababa, Ethiopia, 26 April – 1 May 2020. – P. 1–16.

8. Real E. Regularized evolution for image classifier architecture search / E. Real, A. Aggarwal, Y. Huang // Proceedings of the AAAI Conference on Artificial Intelligence, IAAI-2019, Honolulu, HI, USA, 27 January – 1 February 2019, AAAI Press. – P. 4780–4789. DOI:10.1609/aaai.v33i01.33014780.

9. Zoph B. Learning transferable architectures for scalable image recognition / B. Zoph, V. Vasudevan, J. Shlens // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR-2018, Salt Lake City, UT, USA, 18–23 June 2018. – P. 8697–8710. DOI:10.1109/CVPR.2018.00907.

10.Bello I. Neural optimizer search with reinforcement learning / I. Bello, B. Zoph, V. Vasudevan // Proceedings of the 34th International Conference on Machine Learning, ICML-2017, Sydney, Australia, 7-11 Aug. 2017, JMLR.org. – P. 459–468. DOI:10.5555/3305381.3305429

11.Pham H. Efficient neural architecture search via parameters sharing / H. Pham, M. Guan, B. Zoph // Proceedings of the 35th International Conference on Machine Learning, ICML-2018, Stockholmsmässan, Stockholm Sweden, 11-15 July 2018, JMLR.org. – P. 4095–4104.

12.Zhang X. PolyNet: A pursuit of structural diversity in very deep networks / X. Zhang, Z. Li, C. Loy // Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR-2017, Honolulu, HI, USA, 21-26 July 2017, IEEE Inc. – P. 3900–3908. DOI:10.1109/CVPR.2017.415.

13.Zhong Z. Practical block-wise neural network architecture generation / Z. Zhong, J. Yan, W. Wu // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE Inc. – P. 2423–2432. DOI:10.1109/CVPR.2018.00257.

14.Negrinho R. Towards modular and programmable architecture search / R. Negrinho, M. Gormley, G. J. Gordon // Proceedings of the 33rd Conference on Neural Information Processing Systems, NIPS-2019, Vancouver, BC, Canada, 8-14 Dec. 2019, Curran Associates Inc. – P. 13715–13725.

15.Wei T. Modularized morphing of deep convolutional neural networks: A graph approach / T. Wei, C. Wang, C. W. Chen // IEEE Transactions on Computers, Preprint. – 2020. DOI:10.1109/TC.2020.2988006.

16.Krizhevsky, A. Learning multiple layers of features from tiny images: master’s thesis / A. Krizhevsky // The University of Toronto. – 60 p.

17.Romanuke V. V. An attempt of finding an appropriate number of convolutional layers in CNNs based on benchmarks of heterogeneous datasets / V. V. Romanuke // Electrical, Control, and Communication Engineering. – 2018. – Vol. 14, No. 1. – P. 51–57. DOI:10.2478/ecce-2018-0006.

18. Romanuke V. V. Appropriate number of standard 2 × 2 max pooling layers and their allocation in convolutional neural networks for diverse and heterogeneous datasets / V. V. Romanuke // Information Technology and Management Science. – 2017. – Vol. 20, No. 1. – P. 12–19. DOI:10.1515/itms-2017-0002

19. Radiuk P. M. Application of a genetic algorithm to search for the optimal convolutional neural network architecture / P. M. Radiuk // Herald of Khmelnytskyi National University. – 2020. – Vol. 281, No. 1. – P. 7–11. DOI:10.31891/2307-5732-2020-281-1-7-11.

20. Radiuk P. M. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets / P. M. Radiuk // Information Technology and Management Science. – 2017. – Vol. 20, No. 1. – P. 20–24. DOI:10.1515/itms-2017-0003.

DOI 10.33243/2518-7139-2020-1-1-37-44