International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Neural Architecture Search in Classical and Quantum Computers: A Survey

Suhas Poornachandra[1], Prapulla S B[2]

[1]Student, Dept. of Computer Science and Engineering, RV College of Engineering
[2]Assistant Professor, Dept. of Computer Science and Engineering, RV College of Engineering

---

Abstract - *Neural Architecture Search (NAS) is a technique used to partially automate the designing of deep learning architectures. In recent years, NAS has delivered architecture designs that have greater accuracy than any human-designed models. NAS also gives promising results when used to reduce the latency of inference run and the size of the deep learning model without affecting the accuracy significantly. This survey discusses and analyses the different approaches used for the NAS, existing problems, and proposed solutions. Additionally, this paper discusses some predictions on the future of NAS methods and the use cases in the future. This paper also throws light upon improvements that will be added by the use of quantum computers.*

Key Words: Neural Architecture Search, Automation of Deep Learning, Search space design, Search strategy, Performance evaluation, Quantum Neural Architecture Search

1. INTRODUCTION

Deep learning methods extract features from unstructured data such as text, image and audio. Which makes them highly successful in tasks such as machine translation, image and speech recognition. This success has created huge demand for architecture engineering, which has resulted in creation of more complex architectures designed manually. However deep learning techniques are computationally intensive and model designing requires high level domain knowledge. Deep learning models designed by NAS have outperformed models designed by humans, object detection [10], image classification [10,11] and semantic segmentation [12]. NAS is a subfield of AutoML whose goal is to completely automate the machine learning starting from data collection to predicting outputs.

There has been an increased effort in research towards NAS within last two years. This resulted because of the realization that it is possible to achieve significant gain in the performance of deep learning methods by making slight modifications to the present architecture. This realization meant that the reduced performance by a model is because of the slightly different model architecture. These variables which needs to be modified can be activation function of a layer, kernel size of convolution network, optimal skip connections in the network, dropout rate, number hidden units of a LSTM etc. As it can be observed that these modifications are minor but finding the exact place of modification and the amount of modification is a laborious and error prone task. Hence NAS is used to automate this process, on basic level it can be said that NAS is a search algorithm over the set of variables that define the architecture of a neural network. Meaning that NAS is employed to find out the best architecture from the pool of architectures which have different combinations of the variables as defined above. From this search space NAS will pick out architecture which has the best performance. By this definition we can define NAS as a system of three parts: search space, search strategy and performance estimation strategy [1].

- Search space: Search space defines all the architectures that can be represented in principle.
- Search strategy: The search strategy details how to explore the search space.
- Performance estimation strategy: Refers to the estimation of the performance of deep learning model. Simplest one is the sample training and validation of the architecture on the given dataset.

This review is divided into several sections, Section 2 discusses about the different search spaces that have been proposed, Section 3 looks at the search strategies proposed till now, Section 4 looks at the performance calculation strategies with a look at the multi object search, Section 5 discusses about the use cases where NAS will be used in the future, Section 6 discusses about the improvements that can be added by the quantum computing to NAS and Section 7 concludes the discussion with the outlook on future directions.

2. ARCHITECTURE SEARCH SPACE

Neural Architecture Search Space can be defined as the set of all architectures which can be a feasible solution of the given NAS method. Looking at the Search Space for Convolutional Neural Networks (CNNs) it can be divided into two categories global search space and cell-based search space. Whereas for Recurrent Neural Networks (RNNs) a different kind of search space needs to be used. This section discusses about all these search spaces.

Cell based search space means that architecture is composed of the repetition of the fixed structures called cells. Zoph et al in [10] was the first to use this kind of search space his search space is popularly known as NASnet search space. NASnet search space consists of two kinds of normal cell and reduction cell, former doesn't reduce the reduce the spatial maps and latter is designed to reduce the spatial map. This

design dominates the cell-based search space. Zongh et al in [13] uses pooling layer instead of reduction cell to reduce the spatial dimension. Both of these search space did not have skip connections between cells but there were skip connections inside the cell. While the architecture remains the same across the cells of the architecture hyperparameters inside the cell can be changed for each cell. Global search space gives freedom to select operations of the complete architecture. Baker et al [14] explored this search space which had chain structure search space which did not have skip connections. Zoph and Lee in [15] defined a version of this space with arbitrary skip connections.

Cell based search spaces are more popular in the NAS community because of their success and transferability of the cells across different datasets. But Tan et al [16] states that diversity of layer is necessary to achieve both high accuracy and low latency. Hence, we can conclude that selection of search space is dependent on the researcher and their requirement.

As for search space of RNN is concerned no researcher has explored this part except Zoph and Lee in [15]. They are the only researchers who have defined a search space for the RNN. They defined a search space where each recurrent cell is composed of hidden state, cell state and input for that step. The definition of search space plays a major role in the success of the NAS method. While increase in the hyperparameters and architectural parameters for selection broadens the architectures available for the selection it also comes at a cost of increased exploration time. This increased exploration time can cause the delay in results and may need extra resource for proper execution. Hence it is advised that the researcher choose a search space which has less hyperparameters available for tuning.

3. ARCHITECTURE SEARCH STRATEGY

As mentioned in section 1 NAS is a search algorithm on the Architecture Search Space. Making the search strategy principal part of NAS responsible for achieving success. But this search strategy is an example of black box optimization problem. Many different search strategies are proposed to explore the neural architecture search space including random search, Bayesian optimization, Evolutionary methods, Reinforcement Learning, Surrogate Model based search, One shot architecture search, Monte Carlo Tree search and gradient based methods.

Bayesian Optimization was the earliest successful search strategies in NAS, Domhan et al [17] created state of the art performance on cifar-10 without data augmentation. Mendoza et al [18] formed the world's first automatically tuned neural network which won competition data sets against human experts. But after this BO has not been used much by the researchers because typical BO toolboxes are based on Gaussian processes and focus on lowdimensional continuous optimization problems.

Evolutionary methods were first proposed by Real et al in [19] based on tournament selection for both survivor and parent selection. Shortly after this Xiu and Yullie in [20] proposed the use of genetic algorithm over a more structured search space. Real et al in [21] created AmoebaNet-b and AmoebaNet-c with evolutionary algorithms on NASnet search space it created new record on classification for both CIFAr-10 and ImageNet dataset. These evolutionary algorithms used for NAS are sometimes also referred as neuro evolutionary algorithms.

Reinforcement learning is the most popular search technique among researchers for NAS. As we see in [10,14,17,23] RL has been very successful in identifying architectures which perform very well compared to the human designed architectures. Tan et al in [16] demonstrated that RL was successful in multi objective search also.

Elsken et al in [24] proposed a hill climbing algorithm which moves greedily in the direction of architectures performing better than previous without requiring any other exploration techniques.

Liu et al in [25] propose a direct gradient-based optimization. Here authors create the operation to be selected as a convex combination of all operators which can fit in that space. Now this equation for selection of operation is differentiable and this loop is continued till a discrete operation is obtained.

Comparing the different search techniques based on the performance of architecture detected by the search methods is not a good idea because all these search methods are applied on different search space an architecture which has very good performance detected by RL method might not even be present in the search space of evolutionary methods. Recently NAS bench 101[26] and NAS bench 201[27] tried to solve this problem by using a database of trained architecture and applying search method on that database as a search space. Results obtained by their research conclude that there isn't much difference between RL and evolutionary search algorithms. Real et al in [21] also compared RL, Evolutionary search and Random search. Concluding that RL and evolutionary search perform equally well in terms of test accuracy but evolutionary search finds smaller models and models with lesser inference latency. Whereas both methods outperform Random Search at all stages.

4. PERFORMANCE CALCULATION

Performance calculation of the architectures is the bottle neck for the speed up of NAS methods. Classical performance

calculation of architecture is to train the dataset on the architecture and obtain validation accuracy. But the training of single architecture on given dataset takes a lot of time based on the size of dataset and number of epochs used for training. This is the reason why we see NAS applied on smaller datasets like CIFAR-10 and penn tree bank [10]. With the increase in the size of datasets the resources and time required also increases. There are many methods used to reduce this bottle neck to achieve speed.

Estimating the performance of architecture based on the lesser number of epochs, subset of the dataset, downscaled models and downscaled data. It is assumed that any architecture performing well on the above condition will perform well on the complete dataset. This approach is used in [10,21,28,29,30 and 31]

Previous approach just assumed that performance will extrapolate as expected for bigger datasets and models but some researchers estimated the performance based on extrapolation methods on initial data on performance. This approach is used in [32,33,34 and 35].

One more approach is to use models which inherit weights from a parent model. That is instead of training model from the scratch it is warm started with weights inherited from a parent model. This approach was followed in [19,22,24,36 and 37]

One shot model or weight sharing approach is also used in some cases. One shot model approach first trains a parent model which is a combination of all neural architectures present in the search space, meaning that NAS searches for the model which are only sub graphs of the parent model. Now since the whole model is trained, weights of sub graph are taken from the parent graph. In this only one model needs to be trained which reduces the performance calculation time by a large margin. This approach is used in [25,38,39,40 and 41]

All of the above approaches have their own limitations, many researchers argue that these methods are not a correct way to measure the performance of a model. That is also true because all these methods are an estimation of final performance. Some models might be lagging in the initial calculations and might perform well in later stages. And while training the models common hyperparameters are used for all models, meaning that hyperparameters are not optimized for the given model, which in turn affects the performance of the model. But these methods have achieved good results with less resources utilized. Hence researchers are using the above methods to overcome the requirement of large amount of GPU hours needed for NAS when models are trained from scratch for performance calculation.

5. APPLICATION OF NAS

It is established that NAS outperforms the human designed models in performance, size and inference latency. But its limitations of large resource requirement to arrive at the given results hasn't made them a preferred choice when the size of dataset is very large. And there is a research gap for NAS used for applications other than image classification. Some researchers have taken up initiative to fill up this gap. Notable among them are Image Restoration [43], Semantic segmentation [12], Transfer learning [42], Machine Translation [31] and RNN [15,44] for language and music modelling.

One more direction in which NAS has produced promising results is multi objective performance. In this performance of the model will be a combination of accuracy and number of model parameters, number of floating-point operations, device specific latency. In this regard some model compression techniques inspired by NAS are proposed to reduce the size of the model without altering the accuracy of the model. Elsken et al in [24] proposed an evolutionary algorithm where the objective functions are divided into cheap to evaluate objective functions (number of parameters and inference time) and expensive to evaluate objective functions (validation accuracy). Tan et al in [16] combined the accuracy and inference time into a single equation for objective function. Where accuracy is directly proportional and inference time is inversely proportional to the objective function. Model compression techniques inspired by NAS are proposed in the following works of [45,46 and 47]

NAS has a great potential to provide good results when applied for device specific architecture design. Which is already proved in [16,40]. Now a days we can see a vast variety of hardware designs. And we can expect a significant reduction in latency if neural network is designed for the specific hardware devices. Which is a humanely impossible task making NAS an inevitable choice for this task. While applying NAS for one device the architecture design and weights can be saved as done in [26,27] which can be reused when applying NAS to other devices. Resource requirements would be justified when the function is optimizing the network for more than a certain number of devices.

6. IMPROVEMENTS WITH QUANTUM COMPUTING

Quantum computing even though has promised the exponential speedup of execution of certain algorithms. From the day Shor proposed his algorithm for prime factorization researchers have been coming up with new propositions and algorithms using quantum computers. Recent findings have shown that even fields of Machine learning and AI can achieve exponential speed up with the help of quantum computing. Quantum computation researchers hope to find more quantum algorithms demonstrating significant speedup over classical algorithms.

The design of QNNs [3] is one such problem where classical computers can help quantum research. On the other hand, the AI community believes that quantum computation shows significant potential for solutions to currently intractable problems [9].

Quantum Reinforcement learning [8] and Quantum Evolutionary Algorithms [7] proposed have promised an exponential speedup in terms of mathematical computations. Quantum Reinforcement learning has proved to produce a good tradeoff between exploration and exploitation with the help of quantum superposition for representing the states i.e. in the case NAS representing architecture search space. Meaning that with the proper arrangement of qubits one can fit all the architectures in the search space as a single qubit array. Of course, these qubits will be combination of the architectures. By this search action will be reduced to finding the combination of qubit which has higher performance.

Apart from these some research has been focused on application of quantum computing specifically for NAS. Silva et al in [5] proposed a quantum algorithm which can predict the performance of the classical neural network architectures. This algorithm can speed up the NAS algorithm by reducing the time required for performance calculation as neural network's performance can be predicted without training. Szwarcman et al in [6] proposed a quantum inspired neural architecture search based on the quantum inspired evolutionary algorithms. This algorithm is designed to be executed in classical computers by using the techniques applied in quantum computation. Santos et al in [4] proposed a quantum algorithm which can possibly train the neural network in superposition. i.e. training many neural network architectures in a single run. He also proposed the method for evaluating and selection of neural network architectures. But as all quantum computing algorithms these algorithms also need to be verified in the quantum computers which might take many years for practical implementation.

7. CONCLUSION

With the above findings the question raises can computers design neural network architectures better than humans? Can NAS replace the neural network architecture designers? In the current scenario the answer will be no. But what happens in the future? What happens when Quantum Computers are practically implemented? Will the answer still remain NO?

The answer is a bit complicated for the question. As the title says NAS is a search algorithm meaning that it finds the best architecture in the given search space. NAS cannot create a new architecture design. It sure does find the best architecture in the given search space faster than humans. But it cannot go beyond the search space provided. The only advantage of NAS would be faster search results than human counterparts. It is true that computers can do the repetitive tasks faster than humans. But if NAS can provide better results than human designed architectures then it is efficient for project developers to use NAS. Given the fact that NAS also has the capability to find architectures which have higher accuracy and still have smaller size and lesser inference timing. NAS can also tune the architecture for efficiency in the given architecture. Due to all these reasons it is no doubt that NAS will have upper hand compared to neural network designers.

Neural network designers in the future should focus on designing the architecture search space rather than single architecture design. The research needs to be more focused on creating novel architecture designs like CNN, RNNs etc. Research gap in the field of representing RNNs and other deep learning models in the architecture search space. Even research in the field of search strategy needs to be explored. The advancements in the performance calculation strategy for estimating the performance of architectures needs to be tested and validated.

REFERENCES

[1] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. arXiv preprint arXiv:1808.05377, 2018.

[2] Martin Wistuba, Ambrish Rawat, and Tejaswini Pedapati. A survey on neural architecture search. Technical report, arXiv preprint arXiv:1905.01392, 2019

[3] M. Schuld, I. Sinayskiy, F. Petruccione: The quest for a Quantum Neural Network, Quantum Information Processing 13, 11, pp. 2567-2586 (2014)

[4] Dos Santos, Priscila G. M. et al. "Quantum Enhanced Cross-Validation for Near-Optimal Neural Networks Architecture Selection." International Journal of Quantum Information 16.08 (2018): 1840005

[5] A. J. da Silva and R. L. de Oliveira. Neural networks architecture evaluation in a quantum computer. In 6th Brazilian Conference on Intelligent System, pages 163–168, MG, 2017. IEEE.

[6] D. Szwarcman, D. Civitarese and M. Vellasco, "Quantum-Inspired Neural Architecture Search," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8852453.

[7] M. B. R. Vellasco, A. V. A. Cruz, and A. G. Pinho, "Quantum inspired evolutionary algorithms applied to neural modeling," IEEE World Conference on Computational Intelligence, Plenary and Invited Lectures, pp. 125–150, 2010.

[8] Daoyi Dong et al. "Quantum Reinforcement Learning." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38.5 (2008)

[9] Mingsheng Ying, "Quantum computation, quantum theory and AI", Artificial Intelligence, Volume 174, Issue 2,2010

[10] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for

scalable image recognition. In Conference on Computer Vision and Pattern Recognition, 2018.

[11] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Aging Evolution for Image Classifier Architecture Search. In AAAI, 2019.

[12] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. arXiv preprint, 2019.

[13] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 2423–2432, 2018. doi: 10.1109

[14] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[15] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[16] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. CoRR, abs/1807.11626, 2018

[17] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), 2015.

[18] H. Mendoza, A. Klein, M. Feurer, J. Springenberg, and F. Hutter. Towards AutomaticallyTuned Neural Networks. In International Conference on Machine Learning, AutoML Workshop, June 2016.

[19] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V. Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 2902– 2911, International Convention Centre, Sydney, Australia, 06–11 Aug 2017.

[20] Lingxi Xie and Alan L. Yuille. Genetic CNN. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 1388–1397. IEEE Computer Society, 2017.

[21] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Aging evolution for image classifier architecture search. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA, 2019.

[22] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Simple and Efficient Architecture Search for Convolutional Neural Networks. In NIPS Workshop on Meta-Learning, 2017.

[23] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2423–2432, 2018.

[24] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Efficient multi-objective neural architecture search via lamarckian evolution. In International Conference on Learning Representations, 2019.

[25] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In International Conference on Learning Representations, 2019.

[26] Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. arXiv preprint, 2019.

[27] X. Dong and Y. Yang, "NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search," in ICLR, 2020

[28] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: bandit-based configuration evaluation for hyperparameter optimization. In International Conference on Learning Representations, 2017.

[29] Arber Zela, Aaron Klein, Stefan Falkner, and Frank Hutter. Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. In ICML 2018 Workshop on AutoML (AutoML 2018), 2018.

[30] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 1436–1445, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[31] Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design RNA. In International Conference on Learning Representations, 2019.

[32] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Freeze-thaw bayesian optimization. 2014.

[33] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), 2015.

[34] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. In Aarti Singh and Jerry Zhu, editors, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54 of Proceedings of Machine Learning Research, pages 528–536, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR

[35] Bowen Baker, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Accelerating Neural Architecture Search using Performance Prediction. In NIPS Workshop on Meta-Learning, 2017.

[36] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In Association for the Advancement of Artificial Intelligence, 2018.

[37] Han Cai, Jiacheng Yang, Weinan Zhang, Song Han, and Yong Yu. Path-Level Network Transformation for Efficient Architecture Search. In International Conference on Machine Learning, June 2018.

[38] Shreyas Saxena and Jakob Verbeek. Convolutional neural fabrics. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 4053–4061. Curran Associates, Inc., 2016.

[39] Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. SMASH: one-shot model architecture search through hypernetworks. In NIPS Workshop on Meta-Learning, 2017.

[40] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In International Conference on Learning Representations, 2019.

[41] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: stochastic neural architecture search. In International Conference on Learning Representations, 2019.

[42] David R. So, Chen Liang, and Quoc V. Le. The evolved transformer. arXiv preprint, 2019.

[43] Masanori Suganuma, Mete Ozay, and Takayuki Okatani. Exploiting the potential of standard convolutional autoencoders for image restoration by evolutionary search. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 4771–4780, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[44] Aditya Rawal and Risto Miikkulainen. From Nodes to Networks: Evolving Recurrent Neural Networks. In arXiv:1803.04439, March 2018.

[45] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: automl for model compression and acceleration on mobile devices. In Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII, pages 815–832, 2018. doi: 10.1007/978-3-030-01234-2\ 48.

[46] Anubhav Ashok, Nicholas Rhinehart, Fares Beainy, and Kris M. Kitani. N2N learning: Network to network compression via policy gradient reinforcement learning. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018

[47] Shengcao Cao, Xiaofang Wang, and Kris M. Kitani. Learnable embedding space for efficient neural architecture compression. In Proceedings of the International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA, 2019