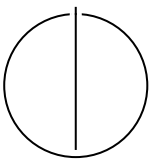
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis, Master's Thesis, ... in Informatics

Thesis title

Author



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

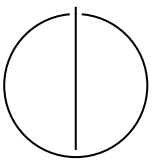
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis, Master's Thesis, ... in Informatics

Thesis title

Titel der Abschlussarbeit

Author:	Author
Examiner:	Supervisor
Supervisor:	Advisor
Submission Date:	Submission date



I confirm that this bachelor's thesis, master's thesis, ... is my own work and I have documented all sources and material used.

Munich, Submission date Author

Acknowledgments

Abstract

Applying techniques from Neural Architecture Search (NAS) to Federated Learning (FL) has been fruitful (remove) in recent years. The combination was identified as a promising research (remove) direction by [Kai+21]. It has yielded methods for finding architectures that deal with the challenges imposed by the FL setting.

Research into NAS has grown rapidly [Whi+23] since it was popularized by [ZL17]; consequently, literature on its application to FL has grown. The last survey on NAS applied to FL compared approaches of four papers [ZZJ21]. Since then, we have identified approximately 50 new papers. This motivates a new systematic survey of the landscape to identify progress and gaps in the literature.

In this thesis, we propose a map of the literature landscape based on the FL challenges they address. We achieve this by systematically evaluating the literature and identifying which challenge it solves.

We refer to the FL challenges described in [ZL17], i.e., non-IID data, limited communication, client heterogeneity, privacy of client data, and break them down into smaller subchallenges — each subchallenge being associated with a pattern in the literature. We include personalized FL [Tan+23] as an additional subchallenge that was not originally posited, but has since drawn the community's attention.

We then analyze how the subchallenges are addressed and focus on the contribution of the used NAS method towards overcoming the subchallenge. For each subchallenge, we keep track of the NAS types used (following [Whi+23], [Avv+25]) and assess whether the underexplored methods are candidates for future research.

Neural Architecture Search

Contents

1 Introduction

Applying Neural Architecture Search (NAS) methods in a Federated Learning (FL) setting is an emerging field of study which we shall call *FedNAS*. In this chapter we provide a motivation for the study of FedNAS and *adaptation techniques* used by FedNAS methods to adapt NAS to the FL setting. Our motivation leads us to our research questions and finally, our proposed methodology to address them.

1.1 Motivation

NAS and FL have independantly made significant progress in recent years and both are increasingly adopted in practice.

The goal of NAS is the automation of the laborious neural network architecture engineering process responsible for so many of the advances made in Deep Learning in recent years [EMH19]. The most recent wave of NAS research was initiated by [ZL17] in 2017. Reinforcement Learning, a black-box optimization technique, was used to show that an automatically searched architecture can outperform handcrafted state-of-the-art architectures for image classification. NAS methods proposed early on were inefficient and a significant amount of research attention to address this improved the situation soon after. Newly developed efficient techniques tended to make use of the internals of the searched architectures (ENAS [Pha+18] and DARTS [LSY19] are prominent examples). These techniques are still widely used today and have helped speed up NAS by orders of magnitude. Faster NAS has enabled more wide-spread adoption and consequently more experimentation for finding architectures with better accuracy, smaller size and faster training convergence. Architectures found via NAS now frequently improve upon state-of-the-art handcrafted architectures (e.g. NASNet [Zop+18], AmoebaNet [Rea+19], MobileNetV3 [How+19], EfficientNetV2 [TL21] and many more).

FL on the other hand has become a viable privacy-enhancing choice for Collaborative Distributed Machine Learning [Jin+24]. Ever more valuable training data is collected on edge devices like smartphones referred to as *clients*. Traditionally, this privacy-sensitive data is either not used at all or it is collected in a central location for training — greatly increasing the risk of exploitation by malicious actors. FL was invented to address this issue and allow the use of decentral training data without the data leaving the device it

was generated on. Instead of performing training at a central location with a cluster of high-end machines, clients in FL train a shared model on their local data and coordinate weight updates to the shared model via a central server. FL "*embodies the principles of focused collection and data minimization, and can mitigate many of the systemic privacy risks and costs resulting from traditional, centralized machine learning*" [Kai+21]. Since its inception in 2016, FL has been adopted for various ML tasks in production systems by organizations like Google [Yan+18], Apple [Ji+25] and Owkin [Old+22].

Inevitably, users of FL expect to make use of existing NAS methods in the FL setting to produce architectures that achieve state-of-the-art accuracy. Apart from the generic benefits, NAS has been identified as a particularily good fit for the FL setting. A large body of work in NAS has focused on finding smaller architectures with reduced inference latency that still have reasonable accuracy [Whi+23]. Such architectures are ideal for deployment on the often resource-constrained clients in the FL setting. [Kai+21] notes that predefined architectures may not be an optimal choice for FL where user-generated data is not visible to model developers. They note that predefined architectures may contain components redundant for specific data sets or perform poorly on non-i.i.d. data (which is prevalant in FL). When the architecture search takes place on the clients, architectures can be adjusted according to the data and distribution present on them, yielding architectures better-suited to the data than architectures a model developer would pick. Dealing with with the distributed, non-i.i.d. data of the FL setting has been a key challenge for FedNAS methods and research into the matter has shown promise [HAA21] [Yao+21] [DLF22] [Liu+24] [Yan+24]. Beyond architectures being tailored towards the data and distribution of each client, they can also be personalized to client's heterogenous computation and bandwidth budgets as shown by [Kha+24] [DLF22] [YL24] [Yua+22].

However, research in NAS methods has focused on a central NAS setting and the methods rely on assumptions that do not hold for the FL setting. These assumptions include an abundance of computational resources, homogeneinity of hardware for training and deployment, high availability of worker nodes, access to the entire dataset, access to the data distribution, etc. [Kai+21]. This makes most centralized NAS methods unfeasable for direct application in the FL setting. Adopting NAS methods in the FL setting requires adapting them to the FL setting, giving rise to what we will call *adaptation techniques*. In the following we briefly illustrate three adaptation techniques.

1. For example, naively training a one-shot supernet with a large search space by having each client train the entire supernet and aggregate weights of the entire supernet works for the cross-silo FL setting [HAA21], but doesn't translate to the cross-device setting. Clients in the cross-device setting generally have less compute resources and such a training scheme would result in detrimental

completion times. Instead, in one FedNAS method [DLF22], researchers have opted to reduce the compute burden on clients by only sampling and training subnets within the training budget of the client.

2. Another problem arises when using standard FedAvg to average the architecture weights for DARTS-based supernets [LSY19]. Clients may tend towards architectures at the opposite ends of a spectrum, but averaging the architecture weights may select for an architecture that is not favored by any client. In [Wei+24] this is addressed by aggregating architecture weights into a probability distribution that can be used to sample likely architectures in the next communication round.

3. Typically, when NAS methods are run in a central environment, they implicitly assume that the machines that NAS is performed on is homogenous and contribute equally to the architecture search. This assumption does not hold in the FL setting where the variance between compute resources of clients can be very large. Consequently, higher-end clients will tend to be used in training more and bias the searched architecture. [ÖÖ25] adapts a NAS method to overcome this problem by by grouping clients into clusters according to their computational speed and network bandwidth. Small models get trained on low-end client clusters and larger models get trained on high-end client clusters.

As shown by the examples above, the conditions of the FL setting break NAS methods in a way that requires novel adaptation techniques. While NAS research has focused on finding ways to perform NAS faster and finding better and smaller architectures, a key challenge of FedNAS methods lies in achieving these goals in spite of the challenges imposed by the FL setting — and adaptation techniques are key. No consolidated body of knowledge exists that can inform researchers on existing adaptation techniques and aid them in designing new techniques. To mend this, we propose a systematic review of adaptation techniques in this thesis.

1.2 Research Questions

Making use of either NAS or FL on its own is a complex task. Combining them introduces even more possible knobs to turn on the system as a whole. There are many NAS methods to choose from and many challenges the FL setting imposes that can be optimized — surmounting in a vast amount of possible FedNAS methods. This leads us to our first research question:

(RQ1) Which FedNAS methods already exist?

The vast amount of FedNAS methods result in a vast array of different adaptation techniques. Each FedNAS method makes use of a set of adaptation techniques to adapt a NAS method to FL. Some FedNAS methods have overlapping sets of adaptation techniques, but most sets of adaptation techniques are disjoint. Nonetheless, most literature does not draw clear boundaries around individual adaptation techniques, leading us to our second research question:

(RQ2) How can we identify and cleanly separate adaptation techniques?

Each adaptation technique pushes the system as a whole in a direction with regards to optimality towards FL challenges. Identifying this direction for each adaptation technique is important to understand potential trade-offs. This leads us to our third research question:

(RQ3) Which FL challenges do the adaptation techniques address at what magnitude?

Adaptation techniques are heavily dependant on the targeted FL challenge and used NAS method. Naturally, this does not allow for most adaptation techniques to work together. Additionally, some adaptation techniques are incompatible due to other reasons [TODO: what other reasons]. We would still like to know which existing adaptation techniques can be combined how, possbily paving the way for creating more powerful FedNAS methods tailored towards specific use cases. This leads us to our third research question:

(RQ4) How can adaptation techniques be composed? How can we choose a set of adaptation techniques based on a use case?

Once we have made it easy to mix and match adaptation techniques and understand in which direction they push the system, we can potentially identify combinations that could improve upon the status quo for selected subsets of FL challenges. Therefore our final research question is:

(RQ5) Which combinations of adaptation techniques are particularily promising for certain sets of challengs in the FL setting?

1.3 Methodology

To tackle RQ1 we propose a systematic literature review with a PRISMA-style table for literature inclusion and exclusion. Once the set of literature to include is fixed, we address RQ2 by performing systematic coding of adaptation techniques according to ... [TODO: cite] and provide a concept definition together with discriminant rules that clearly separate one adaptation technique from others.

For RQ3 we make use of prior work on a taxonomy of FL challenges [Arb+24]. We will go over all adaptation techniques and discuss how the technique works towards, against or has no effect towards overcoming each of the FL challenges. The end result will be a table with the techniques on the y-axis and the FL challenges on the x-axis.

For RQ4 we propose building an is-compatible-with relation on the set of adaptation techniques. We achieve this by cross-examing each adaptation technique with every other one. To reduce the number of comparisons we make use of the taxonomy of NAS methods provided in [Whi+23] and place each adaptation tecnique in one or multiple of the NAS method bins. Only adaptation techniques within the same bin need to be compared.

Finally, we make use of the is-compatible-with relation and our table discussing the FL challenges for each to select a few promising adaptation technique combinations and discuss their potential compared to the state of the art.

Abbreviations

List of Figures

List of Tables

Bibliography

[Arb+24] M. Arbaoui, M.-A. Brahmia, A. Rahmoun, and M. Zghal. "Federated learning survey: A multi-level taxonomy of aggregation techniques, experimental insights, and future frontiers." In: *ACM Trans. Intell. Syst. Technol.* 15.6 (Nov. 2024). ISSN: 2157-6904. DOI: 10.1145/3678182.

[Avv+25] S. S. P. Avval, N. D. Eskue, R. M. Groves, and V. Yaghoubi. "Systematic review on neural architecture search." In: *Artificial Intelligence Review* 58.3 (Jan. 6, 2025), p. 73. ISSN: 1573-7462. DOI: 10.1007/s10462-024-11058-w.

[DLF22] L. Dudziak, S. Laskaridis, and J. Fernandez-Marques. *FedorAS: Federated architecture search under system heterogeneity*. 2022. arXiv: 2206.11239 [cs.LG].

[EMH19] T. Elsken, J. H. Metzen, and F. Hutter. "Neural architecture search: A survey." In: *Journal of Machine Learning Research* 20.55 (2019), pp. 1–21.

[HAA21] C. He, M. Annavaram, and S. Avestimehr. *Towards Non-I.I.D. and invisible data with FedNAS: Federated deep learning via neural architecture search*. 2021. arXiv: 2004.08546 [cs.LG].

[How+19] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le. "Searching for MobileNetV3." In: *2019 IEEE/CVF international conference on computer vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140.

[Ji+25] A. Ji, B. Bandyopadhyay, C. Song, N. Krishnaswami, P. Vashisht, R. Smiroldo, I. Litton, S. Mahinder, M. Chitnis, and A. W. Hill. *Private federated learning in real world application – a case study*. 2025.

[Jin+24] D. Jin, N. Kannengießer, S. Rank, and A. Sunyaev. "Collaborative distributed machine learning." In: 57.4 (Dec. 2024). ISSN: 0360-0300. DOI: 10.1145/3704807.

[Kai+21] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A.

Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. *Advances and Open Problems in Federated Learning*. 2021. arXiv: 1912.04977 [cs.LG].

[Kha+24] A. Khare, A. Agrawal, A. Annavajjala, P. Behnam, M. Lee, H. Latapie, and A. Tumanov. *SuperFedNAS: Cost-efficient federated neural architecture search for on-device inference*. 2024. arXiv: 2301.10879 [cs.LG].

[Liu+24] J. Liu, J. Yan, H. Xu, Z. Wang, J. Huang, and Y. Xu. "Finch: Enhancing federated learning with hierarchical neural architecture search." In: *IEEE Transactions on Mobile Computing* 23.5 (2024), pp. 6012–6026. DOI: 10.1109/TMC.2023.3315451.

[LSY19] H. Liu, K. Simonyan, and Y. Yang. *DARTS: Differentiable architecture search*. 2019. arXiv: 1806.09055 [cs.LG].

[Old+22] M. Oldenhof, G. Ács, B. Pejó, A. Schuffenhauer, N. Holway, N. Sturm, A. Dieckmann, O. Fortmeier, E. Boniface, C. Mayer, A. Gohier, P. Schmidtke, R. Niwayama, D. Kopecky, L. Mervin, P. C. Rathi, L. Friedrich, A. Formanek, P. Antal, J. Rahaman, A. Zalewski, W. Heyndrickx, E. Oluoch, M. Stößel, M. Vančo, D. Endico, F. Gelus, T. Boisfossé, A. Darbier, A. Nicollet, M. Blottière, M. Telenczuk, V. T. Nguyen, T. Martinez, C. Boillet, K. Moutet, A. Picosson, A. Gasser, I. Djafar, A. Simon, Á. Arany, J. Simm, Y. Moreau, O. Engkvist, H. Ceulemans, C. Marini, and M. Galtier. *Industry-scale orchestrated federated learning for drug discovery*. 2022. arXiv: 2210.08871 [cs.LG].

[ÖÖ25] G. Öcal and A. Özgövde. "Network-aware federated neural architecture search." In: *Future Generation Computer Systems* 162 (2025), p. 107475. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2024.07.053.

[Pha+18] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean. "Efficient neural architecture search via parameters sharing." In: *Proceedings of the 35th international conference on machine learning*. Ed. by D. Jennifer and K. Andreas. Vol. 80. Proceedings of machine learning research. PMLR, July 2018, pp. 4095–4104.

[Rea+19] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le. "Regularized evolution for image classifier architecture search." In: *Proceedings of the thirty-third AAAI conference on artificial intelligence and thirty-first innovative applications of artificial intelligence conference and ninth AAAI symposium on educational advances in artificial intelligence*. AAAI'19/IAAI'19/EAAI'19. Honolulu, Hawaii, USA: AAAI Press, 2019. ISBN: 978-1-57735-809-1. DOI: 10.1609/aaai.v33i01.33014780.

[Tan+23] A. Z. Tan, H. Yu, L. Cui, and Q. Yang. "Towards personalized federated learning." In: *IEEE Transactions on Neural Networks and Learning Systems* 34.12 (2023), pp. 9587–9603. DOI: 10.1109/TNNLS.2022.3160699.

[TL21] M. Tan and Q. Le. "EfficientNetV2: Smaller models and faster training." In: *Proceedings of the 38th international conference on machine learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of machine learning research. PMLR, July 2021, pp. 10096–10106.

[Wei+24] X. Wei, G. Chen, C. Yang, H. Zhao, C. Wang, and H. Yue. "EFNAS: Efficient federated neural architecture search across AIoT devices." In: *2024 international joint conference on neural networks (IJCNN)*. 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650653.

[Whi+23] C. White, M. Safari, R. Sukthanker, B. Ru, T. Elsken, A. Zela, D. Dey, and F. Hutter. *Neural architecture search: Insights from 1000 papers*. 2023. arXiv: 2301.08727 [cs.LG].

[Yan+18] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays. *Applied federated learning: Improving google keyboard query suggestions*. 2018. arXiv: 1812.02903 [cs.LG].

[Yan+24] J. Yan, J. Liu, H. Xu, Z. Wang, and C. Qiao. "Peaches: Personalized federated learning with neural architecture search in edge computing." In: *IEEE Transactions on Mobile Computing* 23.11 (2024), pp. 10296–10312. DOI: 10.1109/TMC.2024.3373506.

[Yao+21] D. Yao, L. Wang, J. Xu, L. Xiang, S. Shao, Y. Chen, and Y. Tong. "Federated model search via reinforcement learning." In: *2021 IEEE 41st international conference on distributed computing systems (ICDCS)*. 2021, pp. 830–840. DOI: 10.1109/ICDCS51616.2021.00084.

[YL24] D. Yao and B. Li. "PerFedRLNAS: One-for-all personalized federated neural architecture search." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.15 (Mar. 2024), pp. 16398–16406. DOI: 10.1609/aaai.v38i15.29576.

[Yua+22] J. Yuan, M. Xu, Y. Zhao, K. Bian, G. Huang, X. Liu, and S. Wang. *Federated neural architecture search*. 2022. arXiv: 2002.06352 [cs.LG].

[ZL17] B. Zoph and Q. V. Le. *Neural architecture search with reinforcement learning*. 2017. arXiv: 1611.01578 [cs.LG].

[Zop+18] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. "Learning transferable architectures for scalable image recognition." In: *2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2018, pp. 8697–8710. DOI: 10.1109/CVPR.2018.00907.

[ZZJ21] H. Zhu, H. Zhang, and Y. Jin. "From federated learning to federated neural architecture search: a survey." In: *Complex and Intelligent Systems* 7.2 (Apr. 2021), pp. 639–657. ISSN: 2198-6053. DOI: 10.1007/s40747-020-00247-z.