SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Adaptation Techniques for using NAS Methods in the FL Setting

Max Coetzee

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Adaptation Techniques for using NAS Methods in the FL Setting

Titel

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

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

Munich, Submission date Max Coetzee

Acknowledgments

Abstract

[TODO: overhaul]

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 [8]. It has yielded methods for finding architectures that deal with the challenges imposed by the FL setting.

Research into NAS has grown rapidly [15] since it was popularized by [18]; consequently, literature on its application to FL has grown. The last survey on NAS applied to FL compared approaches of four papers [17]. 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 [18], 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 [13] 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 [15], [1]) and assess whether the underexplored methods are candidates for future research.

Neural Architecture Search

Contents

1 Introduction

Both Neural Architecture Search (NAS) and Federated Learning (FL) have made significant progress independently in the past decade, and both are increasingly adopted in practice. To benefit from the advantages of NAS methods in FL, researchers have started combining them by using NAS in the FL setting.

Neural Architecture Search (NAS) automates the process of engineeering neural network architectures for Deep Learning application domains [3]. This stands in contrast to the traditional, labourious approach to applying Deep Learning. Traditionally a team of domain experts and Deep Learning experts engineer a well-suited architecutre based on expert knowledge and trial-and-error.

NAS does not only reduce manual effort, but can also be used to find architectures that perform better than architectures humans have designed for specific application domains [19] [12] [6] [14].

Federated Learning (FL) is a machine learning method whereby *clients* collaboratively train a model without sharing their data. Weight updates to the shared model are coordinated by a central *server*.

FL was invented by Google to enable privacy-preserving usage of the increasing volume of privacy-sensitive data stored on edge devices. Before FL was invented, it was typical for distributed privacy-sensitive data to either not be used at all for machine learning or it was collected in a central location for training — creating the risk of a major breach by malicious actors.

Although FL was originally targeted at the usage of distributed data on edge devices, the usage of distributed data silos (at the organisational level) have become another use case. The former is referred to as the *cross-device* FL setting and the latter as the *cross-silo* FL setting. Both cross-device and cross-silo FL have since been adopted for various ML tasks in production systems by organisations like Google [16], Apple [7] and Owkin [10].

By using NAS in the FL setting, practicioners benefit from the generic benefits of NAS mentioned above as well as several benefits specific to the FL setting:

- A large body of work in NAS focuses on finding smaller architectures with reduced inference latency that still have reasonable accuracy [15]. Such lightweight architectures are ideal for deployment on the resource-constrained clients in the cross-device FL setting.

- [8] note that predefined architectures may not be an optimal choice for FL. Since client data is not visible to model developers, a predefined architecture selected by model developers may contain components redundant for generalizing well from certain client data sets.
- Predefined architectures may perform poorly on another prevalent characteristic of the FL setting: data that is not independently and identically distributed (non-i.i.d.).

However, Using NAS in the FL setting is not straightforward. Research on NAS methods has traditionally focused on a centralized setting as opposed to the distributed FL setting. This makes many NAS methods unfeasable for direct application in the FL setting, because NAS methods designed for the centralized setting can make several assumptions about the search process that do not hold in the FL setting (see Table 1). Instead, practicioners need to adapt NAS methods to the FL setting, giving rise to *Federated Neural Architecture Search* [5] (FedNAS) methods.

Assumptions that hold for NAS in the centralized setting, but do not hold for the FL setting, make it challenging to adapt NAS methods to the FL setting. Table 1 illustrates these discreptencies as well as the resulting challenges faced by FedNAS methods.

Practicioners need to choose which set of challenges they are interested in addressing for their particular use case, since overcoming one challenge typically comes at the expense of others. For example, consider a particular client that might have a lot of data that could contribute towards the architecture search, but constantly drops out of communication rounds. Practicioners can choose to prioritize either a) avoiding delays due to stragglers or b) waiting for stragglers to ensure model fairness.

The possible use cases and subsets of challenges to overcome is large and consequentially a growing body of FedNAS methods have been created by practicioners. To this end an overview of FedNAS methods would be useful. why: - overview for researchers to view the landscape FedNAS methods - overview for practicioners looking for existing FedNAS methods - overview for practicioners wanting to create a FedNAS method for their specific problem However, it is not always clear to practicioners if a FedNAS method exists that well-suited for their particular use case and if one doesn't exist - scenarios for which a FedNAS method exists that suits a problem Practicioners use existing FedNAS methods to solve a problem. create new FedNAS methods, either to adapt new NAS methods to the FL setting or to adapt NAS methods in new ways to overcome different sets of challenge classes. To this end, it is useful to use existing literature on FedNAS methods and avoid re-inventing adaptation techniques for overcoming each of the challenge classes. A considerable amount of literature on FedNAS methods has appeared in recent years, resulting in a large number of novel adaptation techniques. Unfortunately, adaptation techniques are scattered throughout the increasingly large volume of FedNAS literature, putting a burden on researchers intersted in re-using them for new FedNAS methods.

There have been prior literature surveys on FedNAS methods [17] [9] [4]. [17] is an early survey that characterises FedNAS methods on the whole. The survey differentiates FedNAS methods into offline vs. online architecture search and single- vs. multi-objective methods. [9] gives a brief overview of the FedNAS landscape at the time, highlighting the major contributions each FedNAs method has made. [4] provides an overview of how multi-objective optimizaiton can be integrated into FL in general and includes sections that discuss how this is done specifically for FedNAS methods.

Prior literature surveys only analyze a fraction of the FedNAS literature. [17] and [9] are limited by the small amount of FedNAS literature available at the time. The volume of proposed FedNAS methods has grown substantially since. [4] only analyzes FedNAS methods that make use of multi-objective optimization (MOO), thereby excluding a large share of the literature.

To deal with the challenges faced by FedNAS methods, FedNAS practicioners employ what we shall call *adaptation techniques* to adapt NAS methods to the FL setting. For example, naively using a supernet-based NAS method in the cross-device FL setting by having each client train the entire supernet, would result in detrimental completion times. This embodies the computational efficiency challenge, and one FedNAS method [2] overcomes it by *adapting* the subnet sampling process of X NAS method, such that only subnets within the client's training budget get selected for training.

- users can pick FedNAS method for their use case - researcher can create new, better FedNAS methods Adaptation techniques are composable design patterns extracted from the FedNAS literature that make it easy to compose new FedNAS methods suited towards new tasks.

None of the prior literature surveys provide a consolidated body of knowledge that can inform researchers on adaptation techniques used by FedNAS methods. [17] and [9] only analyse and summarise FedNAS methods on the whole. [4] analyses how MOO is used within FedNAS methods. The prior surveys do not identify individual adaptation techniques responsible for overcoming sets of challenge classes.

Extracting the adaptation techniques used by FedNAS methods and organising them into a single consolidated body of knowledge, would allow researchers to easily make use of this knowledge to compose new FedNAS methods tailored to overcoming a specific set of challenge classes relevant to them. This leads us to our research question:

(RQ) **How and which challenge classes do adaptation techniques described in the literature overcome?**

To answer our research question, we perform a systematic literature review of adaptation techniques used by FedNAS methods and their effects on overcoming challenge classes. We divide our approach into 5 steps:

1. **Literature Selection:** We follow the guidelines and flow diagrams provided by PRISMA 2020 [11] for inclusion and exclusion of papers and perform forward and backwards citation searching. Each paper contains one or more FedNAS methods.

2. **Adaptation Technique Extraction:** Once the set of included papers is fixed, we analyse each paper individually, extracting the adaptation techniques it uses and summarising them.

3. **Merge Highly-Similar Adaptation Techniques:** We then merge conceptually highly-similar adaptation techniques into a single representative adaptation technique.

4. **Categorise Adaptation Techniques:** After merging, we categorise the adaptation techniques based on conceptual similarity and deliver a taxonomy of adaptation techniques.

5. **Map FL Challenge Types onto Adaptation Techniques:** Next, we discuss how each adaptation technique works towards, against, or does not affect overcoming each of the FL challenge classes and provide a table with an overview as an end result.

Our review organizes the n extracted adaptation techniques into a single taxonomy that gives researchers an overview of the FedNAS landscape through the lens of adaptation techniques. Our discussions on each adaptation technique helps researchers find and choose adaptation techniques relevant to their problem.

In chapter 2 we cover the background required for this thesis and related work. In chapter 3 we describe the method with which we conduct our literature review in detail. In chapter 4 we explain our process of including FedNAS literature and give an overview of the included FedNAS literature. In chapter 5 we present our taxonomy of adaptation techniques and explain the effect of adaptation techniques on challenge classes. In Chapter 6 we conduct a discussion about our work. Chapter 7 contains our conclusion.

Abbreviations

List of Figures

List of Tables

Bibliography

[1] 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.

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

[3] 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.

[4] M. Hartmann, G. Danoy, and P. Bouvry. *Multi-objective methods in Federated Learning: A survey and taxonomy*. 2025. arXiv: 2502.03108 [cs.LG].

[5] 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].

[6] 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.

[7] 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.

[8] 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].

[9] S. Khan, A. Rizwan, A. N. Khan, M. Ali, R. Ahmed, and D. H. Kim. "A multi-perspective revisit to the optimization methods of Neural Architecture Search and Hyper-parameter optimization for non-federated and federated learning environments." In: *Computers and Electrical Engineering* 110 (2023), p. 108867. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2023.108867.

[10] 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].

[11] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting, and D. Moher. "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews." In: *BMJ* 372 (2021). DOI: 10.1136/bmj.n71. eprint: https://www.bmj.com/content/372/bmj.n71.full.pdf.

[12] 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.

[13] 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.

[14] 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.

[15] 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].

[16] 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]`.

[17] 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`.

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

[19] 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`.