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

Research into NAS has grown rapidly [13] since it was popularized by [15]; consequently, literature on its application to FL has grown. The last survey on NAS applied to FL compared approaches of four papers [14]. 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 [15], 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 [11] 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 [13], [2]) 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 [4]. 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 [16] [10] [6] [12].

Federated Learning (FL) is a machine learning method whereby clients collaboratively train a model without sharing their data. Clients train a shared model on their local data and coordinat weight updates to the shared model via a central server. FL was originally invented by Google to enable the usage of the increasing volume of privacy-sensitive data stored on edge devices, but distributed data silos (at the organisational level) containing privacy-sensitive data 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. Traditionally, this kind of distributed privacy-sensitive data was either not 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.

By using NAS in the FL setting, researchers not only benefit from the generic benefits of NAS mentioned above, but from several advantages specific to 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 [13]. Such lightweight architectures are ideal for deployment on the resource-constrained clients in the cross-device FL setting.

- [7] 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. NAS methods described in the literature traditionally focus on a centralized setting as opposed to the distributed FL setting. When used in a centralized setting, NAS methods can assume i) that worker nodes will have high availability, ii) that the entire training dataset can be accessed, iii) that the distribution of the training data can be inferred, etc. These assumptions do not hold for the FL setting, making most NAS methods unfeasable for direct application in the FL setting. Instead, researchers need to adapt NAS methods to the FL setting, giving rise to what we shall call *FedNAS* methods.

Adapting NAS methods to the FL setting poses a set of challenges similar to the challenges faced by researchers making use of FL in general. We shall make use of a prior work [1] that organizes these challenges into seven classes of challenges:

- **Heterogeneity**: dealing with hardware and data heterogeneity
- **Fairnes**: mitigating bias caused by more performant devices contributing more to the trained gradients
- **Communication Efficiency**: dealing with the high network latency and low bandwith of edge devices
- **Computation Efficiency**: hard to train ML models on low end edge devices
- **Client selection**: clients contribute different amounts of information towards training the shared model, it's hard to select the right clients for a communication round and select clients that will be available for the next communication round
- **Security**:
- **Privacy**:
- **Fault-Tolerance**: clients are not always available and stragglers need to be dealt with

Each FedNAS method employs several *adaptation techniques* dependant on i) the type of NAS method it adapts and ii) the class of challenges the FedNAS method aims to overcome. 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 class, and one FedNAS method [3] 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.

(RQ) Which adaptation techniques are described in the literature and which challenge classes do they overcome in what way?

There have been prior surveys on FedNAS methods [14] [8] [5], but none of them have been exhaustive and - exhaustive - focus on composable units of learning targeted towards certain challenge classes

provide a consolidated body of knowledge that can inform researchers on existing adaptation techniques. [14] is an early work that characterises FedNAS methods on the whole, but not the individual adaptation techniques used by them. [14] was also limited by the small amount of literature available at the time. [8] only analyzes FedNAS methods on the whole as well. Further, [8] focuses only on FedNAS methods that use NAS and Hyperparameter Optimization together and so a large part of the FedNAS literature is not analyzed. [5] is a recent survey that looks at the application of multi-objective optimization methods in Federated Learning in general, not FedNAS specifically. Since only multi-objective optimization methods are considered, a large part of the FedNAS literature is left out as well.

Researchers constantly create new FedNAS methods, either i) to adapt new NAS methods to the FL setting or ii) to adapt NAS methods in new ways to overcome different sets of challenge classes. In this pursuit, they hope to use existing literature on FedNAS methods and avoid re-inventing patterns for overcoming each of the challenge classes. Unfortunately, . on adaptation techniques for overcoming the challenge classes relevant to their use case. Unfortunately, adaptation techniques are scattered throughout the increasingly large volume of FedNAS literature, leading us to our research question:

We propose taking a perspective at the adaptation technique level

- existing literature doesn't break down FedNAS methods into more composable adaptation techniques

A considerable amount of literature on FedNAS methods has appeared in recent years, resulting in a large number of novel adaptation techniques, but they are scattered throughout the literature. This leads us to our research question:

To mend this, we perform a systematic review of adaptation techniques in this thesis.

Literature Selection: To tackle the first part of RQ1, we follow the guidelines and flow diagrams provided by PRISMA 2020 [9] for inclusion and exclusion of papers and perform forward and backwards citation searching. Each paper presents one or more FedNAS methods.

Adaptation Technique Extraction: Once the set of included papers is fixed, we tackle the second part of RQ1 by analysing each paper individually, extracting the adaptation techniques it uses and summarising them.

Categorise Adaptation Techniques: For the first part of RQ2, we cluster adaptation techniques based on conceptual similarities and deliver a taxonomy of adaptation techniques.

Define FL Challenge Types: For the second part of RQ2, we must first define a classification of FL challenges. To this end, we repurpose a prior classification [1] of recent research advances in FL (as shown in **??**). [1] suggests that newly proposed FL methods in FL research enhance one or more *evaluation metrics*. We argue that their aforementioned evaluation metrics are the result of finding a measure for how effectively an FL challenge has been addressed. Therefore, the clustering of evaluation metrics into research directions is akin to clustering FL challenges into types of FL challenges.

Mapping FL Challenge Types onto Adaptation Technique Clusters: Next, we review all clusters of adaptation techniques from step 3, and discuss how each adaptation technique cluster works towards, against, or does not affect overcoming each of the FL challenge types defined in step 4.

Compose promising FedNAS methods: For RQ3, we propose building an "is-compatible-with" relation on the set of adaptation technique clusters. We achieve this by cross-examining each adaptation technique cluster with every other one. We then utilise the "is-compatible-with" relation, together with our discussion from step 5, to identify groups of adaptation technique clusters that can be composed into FedNAS methods, which could potentially shift the Pareto frontier with respect to the optimality of addressing FL Challenge types.

Abbreviations

List of Figures

List of Tables

Bibliography

[1] 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.

[2] 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.

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

[4] 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.

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

[8] 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`.

[9] 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`.

[10] 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`.

[11] 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`.

[12] 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.

[13] 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]`.

[14] 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`.

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

[16] 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`.