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 [McM+17], 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.

Proposed FedNAS methods differ in their search spaces, targeted clients and optimize for different trade-offs in the FL setting. This Bachelor thesis offers a comprehensive survey and framework analysis of FedNAS methods.

1.1 Motivation

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

NAS automates the laborious architecture engineering process responsible for so many of the advances made in Deep Learning in recent years. During the early stages of the most recent wave of NAS research since 2017 [cite: RL NAS Zoph et. al], NAS was mostly done with black-box optimization techniques and was computationally very expensive. In the following years, significant effort was directed towards reducing the computational burden with great success. NAS was no longer a technique only affordable to those with access to vast swathes of GPUs, but could now be done feasibly on a single GPU. Accessibility to NAS helped powered a wide range of experimentation and ultimately resulted in techniques for finding architectures with better accuracy, smaller size and faster training convergence. Architectures found via NAS occupied the top spot of several benchmarks for 90% of the time [TODO: cite].

FL on the other hand has become a viable privacy-enhancing choice for Collaborative Distributed Machine Learning [cite]. Ever more valuable training data is collected on edge devices like smartphones, but collecting it in a central location poses a serious privacy risk to the owners of that data. 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]. Several FL frameworks have sprung up and FL is used in production for various purposes [TODO: cite]. It enables practicioners to train models on data that would otherwise be considered inacessible due to privacy concerns.

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 [TODO: cite]. 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 the optimal choice when 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. Allowing architecture search to take place on the clients, where the architecture can be adjusted according to the data and distribution present, may yield architectures better-suited for the task at hand. Architectures could even be personalized to each client's computational resource budget in a second stage.

However, research in NAS methods has focused on a central NAS setting and the methods rely on assumptions that do not hold for the federated setting. Assumptions that do not hold in the FL setting include an abundance of computational resources, homogeinity 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. Consequentially, adapting NAS methods to the FL setting has spawned a wide variety of *FedNAS* methods.

Example: NAS method adapted to deal with non-i.i.d.

For example, naively training a one-shot supernet with a large search space in the cross-device FL setting by having each client train the entire supernet and aggregate gradients would take weeks or months to complete [TODO: make realistic]. Instead, in one FedNAS method [DLF22], researchers have opted to reduce the compute burden on clients by sending randomly sampled subspaces to clients for training and applied novel weight aggregation techniques tailored to aggregating architecture-related weights. In another [TODO: cite FINCH].

[TODO: Example 2]

[TODO: Example 3]

1.2 Research Questions

The variety of FedNAS methods created in the last couple of years poses several problems to the study and use of FedNAS methods which leads us to our research questions.

Most FedNAS methods compare themselves to two or three related methods, but the field has grown so substantially that patterns in the methods have emerged. Instead of comparing to singular other papers, researchers could be comparing their approaches to a range of similar approaches. There is a lack of a holistic view of the field.

(RQ1) Which FedNAS methods exist?

Research into FedNAS is ongoing and FedNAS researchers have a hard time picking the

(RQ2) How can FedNAS methods be compared on a conceptual level?

(RQ3)

A naive approach to creating a taxonomy for FedNAS might consider a cartesian product of the possible FL settings with NAS methods, but this yields a space relatively sparse of practically relevant combinations.

The increasing amount of papers on this topic leads us to our first research question:

Reducing training time of a model stands in conflict with Additionally many FedNAS methods are similar and share

(RQ5) For each of the categores of NAS methods, what problems arise when adopting them in the FL setting?

(RQ3) Which FedNAS method is suited for which FL setting, what categories of trade-offs become apparent?

on its own:(RQ4) Which trade-off pairings have become apparent?

In most cases, making use of NAS for FL requires designing a framework or system with many components suited to the given use case. The variation between different frameworks and approaches is large, but they tend to be composed of a differing combination of universal components. Most of the approaches have some degree of overlap in the components they use to apply NAS to the FL setting, with components depending on the challenges of the targeted use case.

It is unclear what the most widely used components are, how different components are beneficial towards certain goals and how exactly these components should be defined. Identifying these components along with the direction in which they push a system could allow future research to more easily compose existing components into systems for specific use cases. Furthermore, clarity on the segmentation of a system into components could allow reasoning about the comparison of two systems more effectively.

1.3 Methodology

We therefore propose a detailed analysis of the existing approaches to applying NAS to FL in this Bachelor's thesis.

Abbreviations

List of Figures

List of Tables

Bibliography

[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]`.

[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]`.

[McM+17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. "Communication-efficient learning of deep networks from decentralized data." In: *Proceedings of the 20th international conference on artificial intelligence and statistics*. Ed. by Z. J. Singh Aarti. Vol. 54. Proceedings of machine learning research. PMLR, Apr. 2017, pp. 1273–1282.

[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`.

[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]`.

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

[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`.