Technische Universität München

School of Computation, Information and Technology
Department of Informatics

Exposé — Bachelor's Thesis

Adaptation Techniques for using NAS Methods in the FL Setting

Author:	Max Coetzee
Supervisor:	M.Sc. Nick Henze
Advisor:	Dr.-Ing. Niclas Kannengießer
Date:	tbd.

Contents

1 Problem Statement

Engineering the architecture of a neural network for a Deep Learning application is traditionally done by a team experts via a process of trial and error. To reduce the amount of manual labour involved in this process, researchers invented *Neural Architecture Search* (NAS) [2] methods and improved them over the past decade. NAS methods employ diverse strategies to automatically search for a neural network architecture for a given Deep Learning application.

Independantly, but in parallel to NAS, researchers developed a distributed machine learning approach called *Federated Learning* (FL) [11] in response to growing concerns about data privacy. In FL, *clients* collaboratively train a model without sharing their local data. This enhances the privacy of clients' data by ensuring model trainers can not view clients' data and client data is not collected at a central location where a single breach could expose the data of all clients.

Engineering neural network architectures in FL is as time-consuming (if not more so) as in centralised Deep Learning, therefore researchers have started investigating the use of NAS methods in FL [4] [1] [10], creating *Federated Neural Architecture Search methods* (FedNAS methods) [4]. Additionally, NAS methods provide an alternative to selecting a fixed architecture upfront — a so-called *prededfined architecture*. Predefined architectures can lead to slow training convergence and poorly performing models in FL, because model developers can view neither the clients' data nor, typically, client's hardware capabilities. Model developers may therefore select a predefined architecture that contains components irrelevant for generalising well from client data sets or select an architecture that trains slowly on some clients. Work has already been done that shows the use of NAS methods in FL can mitigate these issues [5] [12] [14].

Despite the potential benefits of using NAS in FL, it is non-trivial to do so. Developing NAS methods is a time-consuming, error-prone process in a centralised environment and even more so in FL, so researchers try to make use of existing work as much as possible. Since, NAS research has focused on a centralised setting, most NAS methods developed thus far are infeasible for direct application in FL, because they can make several assumptions about the search process that do not hold in the FL setting. These assumption discrepancies result in *challenges* for using centralised NAS methods in FL, and developers have created *adaptation techniques* for overcoming them. The majority of FedNAS methods developed so far use centralised NAS methods with the help of adaptation techniques.

For example, one challenge arises from the fact that centralised NAS can assume worker nodes are computationally powerful, whereas clients in most FL settings are not. Since most centralised NAS methods place large computational burdens on worker nodes, practitioners using these NAS methods in FL without modification would experience detrimental search completion times. To combat this, FedNAS developers have created adaptation techniques to reduce the computational burden on individual clients in FedNAS methods [1] [13] [6]. This typically involves reducing the overall computational work and splitting it up into smaller units, which presents a challenge, as the implementation of the resulting FedNAS method tends to be complex.

The subset of challenges faced by FedNAS methods depends on the specific FL setting, which can differ in many parameters [8]. The literature identifies two major classes of FL settings: the *cross-device* class, wherein clients are edge devices, and the *cross-silo* class, wherein clients are entire organisations, but even within these classes, there is significant variation in the setting parameters. Each FL setting violates centralised NAS assumptions to a different extent, making some challenges more relevant to them than others. For example, for FL settings in the cross-silo class, clients can be expected to be equipped with GPUs, making the challenge described above less relevant.

FedNAS developers who design new FedNAS methods for a specific FL setting must decide which challenges to prioritise and which adaptation techniques to implement for their chosen NAS method. However, informing these design decisions currently requires extensive, manual reviewing of a fragmented literature, because many FedNAS papers do not state the targeted FL setting parameters and assumptions clearly, nor do they clearly link techniques with the addressed challenges. As a result, developers struggle to assess the transferability of existing methods to their setting and risk selecting ineffective or select techniques for addressing a challenge that are known to worsen another. This slows down the development of new FedNAS methods.

The literature on adaptation techniques is fragmented, and FedNAS methods often lack clarity regarding the targeted FL setting and the challenges they address. As a result, extending and re-using existing techniques remains difficult. This poses a problem for FedNAS developers, since they need to trade off which challenges to address for their targeted FL setting without a clear overview of adaptation techniques that would be useful for that setting. Prior literature surveys [15] [9] [3] summarise FedNAS methods on the whole, but do not dissect them in a manner that allows FedNAS developers to decide on the parts they wish to re-use. To aid the development of new FedNAS methods, we set out to answer our research question:

What challenges arise from different FL settings for FedNAS methods, and which adaptation techniques address them in the literature?

To tackle our research question, we conduct a systematic literature review of papers that present FedNAS methods. We employ grounded theory and the methodology from [7]. For our review, we consider papers that modify NAS methods in response to the FL setting.

We define a set of fine-grained parameters to characterise the targeted FL setting of each FedNAS method based on observations of varying setting parameters in the literature. With the help of this characterisation, we identify the violated centralised NAS assumptions and catalogue the challenges that arise from them. Next, we extract unrefined adaptation techniques from the FedNAS methods and iteratively refine and merge them to obtain a set of collectively exhaustive adaptation techniques. We analyse how each adaptation technique works towards, against, or does not affect each challenge, and present our findings in the form of a discussion for each adaptation technique, as well as an overview table.

Our review aims to support the creation of new FedNAS methods by developers. By identifying the source of challenges and elaborating on them, we provide clarity on the expected challenges for a targeted FL setting. Based on the expected challenges, FedNAS developers can use our overview of adaptation techniques to guide the design of new FedNAS methods and determine whether to re-use existing techniques, extend them, or develop new ones.

2 Objectives

What is the work intended to achieve? Make it easier to incorporate existing adaptation techniques into newly designed FedNAS methods.

Test for achievement of objective: How long does it take to design a FedNAS method with and without the overview of this thesis. - generate knowledge that is useful for creating new FedNAS methods - - organise FedNAS literature in a way such FedNAS developers can easily re-use techniques

The primary objective of this thesis is to support the design of new FedNAS methods by providing a structured overview of (i) challenges that arise when applying Neural Architecture Search (NAS) in different Federated Learning (FL) settings and (ii) adaptation techniques proposed in the literature to address these challenges.

From this primary objective, the following sub-objectives are derived (cf. the objective decomposition style in the example exposé:

1. **Derive a characterization of FL settings relevant to FedNAS:** Define a set of FL setting parameters that are reported in the FedNAS literature or can be inferred from experimental setups and that plausibly influence which NAS assumptions are violated.

2. **Identify assumption discrepancies and resulting challenges:** Systematically identify which assumptions of centralised NAS methods are violated under which

FL setting parameters and derive a catalogue of FedNAS challenges grounded in the reviewed papers.

3. **Extract and conceptualize adaptation techniques:** Extract technique candidates from FedNAS papers and iteratively refine them into a coherent set of adaptation techniques that is mutually exclusive and collectively exhaustive at the chosen abstraction level.

4. **Relate techniques to challenges:** Analyse how each adaptation technique addresses, aggravates, or does not affect each challenge, and identify trade-offs reported or implied in the literature.

5. **Provide practitioner-oriented artefacts:** Produce (a) an overview table mapping *FedNAS methods* to the challenges they address via their techniques and (b) an overview table mapping *adaptation techniques* to challenge effects to support technique selection and method design.

Verifiability / success criteria. The objective is considered achieved if the resulting artefacts (i) cover all included FedNAS methods, (ii) allow each included method to be decomposed into a set of adaptation techniques, and (iii) provide an explicit, traceable mapping from techniques to challenges (with paper references) such that a reader can justify reuse decisions without re-reviewing the full literature corpus.

3 Explanation of Terms

Neural Architecture Search (NAS)

Traditionally, neural network architectures are designed by a team of domain and DL experts. In NAS, the architecture is automatically determined refers to the automated process of neural network architectures that achieve high performance for a given task, dataset, and constraints. A NAS method is commonly described by its search space, search strategy, and performance estimation strategy.

Federated Learning FL is a distributed learning paradigm in which multiple clients collaboratively train a model under the coordination of a server while keeping training data local to the clients (see also the FL definition used in the Problem Statement).

FL setting and FL setting parameters An *FL setting* denotes the concrete environment in which FL (and here: FedNAS) is executed (e.g., cross-device vs. cross-silo, number of clients, client availability, hardware heterogeneity, bandwidth constraints, degree of non-IID data). *FL setting parameters* are the specific attributes used to describe and compare such settings in a reproducible manner. In this thesis, the parameter set is derived from recurring variations observed in the FedNAS literature.

Federated Neural Architecture Search (FedNAS) method A FedNAS method is a NAS method that is designed to operate in an FL setting, i.e., it searches for architectures while respecting FL constraints such as decentralised data and communication limits.

Centralised NAS assumptions and assumption discrepancies. *Centralised NAS assumptions* are implicit or explicit prerequisites many NAS methods rely on (e.g., access to centrally pooled data, powerful search workers, synchronous evaluation). *Assumption discrepancies* are mismatches between these prerequisites and the realities of a given FL setting.

FedNAS challenge A FedNAS challenge is a concrete problem that arises when assumption discrepancies prevent directly applying a centralised NAS method in an FL setting (e.g., excessive client-side compute demand, communication overhead, unstable search dynamics under partial participation).

Adaptation technique An adaptation technique is any modification to a NAS method that is explicitly motivated by the FL setting (or by a FedNAS challenge) and is intended to make the search feasible, efficient, or effective in that setting. This thesis conceptualises adaptation techniques at a level where they can be reused as design building blocks across methods.

4 Research Approach

We take a qualitiative research approach in the style of the CDML paper. - qualitative - iterative conceptualisation

- qualitative - iterative conceptualisation

The thesis follows a qualitative, iterative research approach in the form of a systematic literature review combined with grounded-theory-inspired coding. The goal is to derive a conceptual model that links FL setting parameters, violated NAS assumptions, resulting challenges, and adaptation techniques.

4.1 Data Collection (Literature Selection)

We collect and select papers that propose or evaluate FedNAS methods. The inclusion and exclusion procedure follows a structured screening process inspired by PRISMA-style selection reporting (identification, screening, eligibility, inclusion). The final corpus consists of FedNAS papers that include one or more explicit adaptations of NAS to an FL setting.

Inclusion criteria (examples). A paper is included if it (i) proposes a method that performs architecture search in an FL setting or (ii) presents a substantive modification of a NAS method specifically motivated by FL constraints.

Exclusion criteria (examples). A paper is excluded if it (i) performs only hyperparameter optimisation without architecture search, (ii) applies a fixed architecture in FL without searching, or (iii) does not provide sufficient methodological detail to extract adaptations.

Data Analysis (Coding and Conceptualisation) The analysis proceeds in iterative steps (open and axial coding), aligned with the staged and explicit structure exemplified in the attached proposal:

1. **FL setting characterisation:** For each FedNAS method, extract reported FL setting parameters. If parameters are not stated, infer them conservatively from the experimental setup and clearly mark them as inferred.

2. **Challenge derivation via assumption discrepancies:** Identify which centralised NAS assumptions do not hold under the extracted setting parameters and code the resulting challenges. Challenges are refined iteratively until they form a stable catalogue.

3. **Unrefined adaptation technique extraction (open coding):** Extract technique candidates as any modification explicitly motivated by the FL setting or by a FedNAS challenge.

4. **Adaptation technique conceptualisation (axial coding):** Merge and refine technique candidates into adaptation techniques based on mechanism similarity. The goal is a coherent set of techniques at a reusable abstraction level.

5. **Technique–challenge mapping (axial coding):** For each technique, code its effect on each challenge as *mitigates*, *aggravates*, or *no clear effect*. Where the literature is ambiguous, the coding records uncertainty explicitly.

6. **Synthesis into artefacts:** Produce (i) a taxonomy of adaptation techniques, (ii) a method-to-challenges overview derived from each method's techniques, and (iii) a technique-to-challenges overview to support design decisions.

Traceability and rigor. To ensure transparency, the thesis maintains an audit trail that links each coded technique, challenge, and setting parameter to supporting passages in the source papers. Where interpretations are required (e.g., inferred setting parameters), the thesis distinguishes clearly between reported and inferred information, reflecting the writing guideline that clear communication is the goal.

5 Structure

1. **Introduction** (3 pages)

2. **Background** (6 pages)
   - 2.1 Neural Architecture Search
   - 2.2 Federated Learning
   - 2.3 Federated Neural Architecture Search
3. **Method** (5 pages)
   - 3.1 Method and Literature Selection
   - 3.2 Reviewed Literature
4. **FedNAS Challenges** (10 pages):
   - 4.1 Parameters of FL settings relevant to FedNAS
   - 4.2 Assumption Discrepancies between NAS and FedNAS
   - 4.3 FedNAS Challenges
5. **Adaptation Techniques** (25 pages)
   - 5.1 Adaptation Technique 1
   - 5.2 Adaptation Technique 2
   - 5.3 ...
   - 5.4 Adaptation Technique 20
   - 5.5 Overview
6. **Discussion** (2 pages)
7. **Conclusion** (1 page)

6 Expected Results

- challenges for FedNAS methods based on FL system parameters - catalog of adaptation techniques - bibliography of FedNAS methods

7 Open Issues and Problems

- finding the correct abstraction level for adaptation techniques

References

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

[2] 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.

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

[4] 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]`.

[5] M. Hoang and C. Kingsford. "Personalized Neural Architecture Search for Federated Learning." In: *1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021)* ().

[6] C. Huo, J. Jia, T. Deng, M. Dong, Z. Yu, and D. Yuan. "NASFLY: On-device split federated learning with neural architecture search." In: *2024 IEEE international symposium on parallel and distributed processing with applications (ISPA)*. 2024, pp. 2184–2190. DOI: `10.1109/ISPA63168.2024.00298`.

[7] 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`.

[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] 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.

[11] 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 S. Aarti and Z. Jerry. Vol. 54. Proceedings of machine learning research. PMLR, Apr. 2017, pp. 1273–1282.

[12] E. Mushtaq, C. He, J. Ding, and S. Avestimehr. *SPIDER: Searching personalized neural architecture for federated learning*. 2021. arXiv: 2112.13939 [cs.LG].

[13] 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.

[14] 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.

[15] 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.