Electronics and Computer Science Faculty of Engineering and Physical Sciences University of Southampton

Josh Pattman

12 December 2022

Investigating Optimisations Of Swarm Learning With Respect To Real World Challenges

Project Supervisor: Mohammad Soorati Second Examiner: ?

A project progress report submitted for the award of
Computer Science with Artificial Intelligence

Abstract

UP TO 200 WORDS

Contents

1 Project Goals **4**
- 1.1 Problem . . . 4
- 1.2 Proposed Solution . . . 4
- 1.3 Goals . . . 5
- 1.4 Focussing On Project Goals . . . 6
- 1.5 Risk Assessment . . . 6
  - 1.5.1 Personal Issues . . . 6
  - 1.5.2 Hardware Failure - Local Computer . . . 7
  - 1.5.3 Hardware Failure - Iridis 5 . . . 7

2 Background and Literature Review **9**
- 2.1 Background Research . . . 9
  - 2.1.1 Keras and Python . . . 9
  - 2.1.2 Iridis 5 . . . 9
- 2.2 Literature Review . . . 9
  - 2.2.1 A survey on federated learning [1] . . . 9
  - 2.2.2 Swarm Learning for decentralized and confidential clinical machine learning [2] . . . 10
  - 2.2.3 Federated learning in Robotic and Autonomous Systems [3] . . . 10
  - 2.2.4 Decentralized Federated Learning: A Segmented Gossip Approach [4] . . . 10
  - 2.2.5 Multi-Center Federated Learning [5] . . . 10
  - 2.2.6 FedBN: Federated Learning On Non-IID Features via Local Batch Normalization [6] . . . 11
  - 2.2.7 Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach [7] 11

3 Report on Technical Progress **12**
- 3.1 Implementation 1 - Basic MNIST classifier . . . 12
- 3.2 Implementation 2 - Simple swarm learning . . . 12

3.3 Implementation 3 - Reduced dataset swarm learning 13

4 Project Planning **14**

4.1 Planned Phases of the Project 14

4.2 Completed Work 15

4.3 Remaining Work 15

Chapter 1

Project Goals

1.1 Problem

Currently, there is more data stored around the world than ever before. However, there are also many recent regulations that prevent the data from being used by machine learning algorithms. In many cases, this is because the data owners are not comfortable or able to send the data to a central server for processing and training. The result of this is that either each data owner has to train their own inferior models, or no models are trained at all due to the lack of data.

In addition to this, the world also has more electronic devices than ever before. As these devices become more and more powerful, using each device to train a machine learning model on a small amount of data becomes a possibility. However, these devices will still not outperform a fast computer alone.

1.2 Proposed Solution

Swarm learning is a type of machine learning which uses multiple devices or agents to train a machine learning model. It is similar to federated learning in this repect. However, swarm learning is a completely decentralised algorithm, where the agents decide how to train. This contrasts federated learning which requires a central server to work. One of the main features of swarm learning is that data never has to be shared amongst the agents in the learning network, which is excellent for the protection of private data. For this reason, swarm learning seems to fit the problem well, but there is another issue with swarm learning: it is still a relatively new algorithm and it faces many real world challenges that have only been adressed in isolation by existing research.

1.3 Goals

To investigate possible optimisations of the swarm learning algorithm on simple problems, whilst adding real world constraints incrementally, and eventaully arriving at a robust swarm learning algorithm that adresses many real world issues in one.

1. Create a working implementation on a simple dataset of swarm learning
2. Incrementally add real world problems to the swarm learning algorithm
3. Mitigate the effects of the problem on the accuracy of the algorithm by implementing either novel ideas or methods from existing literature.
4. Ensure that the end product is as reproducable as possible such that it can be re-implemented for specific use cases.

Problems

Below are some real world problems with swarm learning. This is not a comprehensive list and more problems could be researched if nescesary.

- **Unevenly distributed features/local bias** - This is where feature in the dataset are not distributed evently between agents. For instance, when classifying mnist, one agent may see more 7s and one may see more 2s.
- **Sparsely connected networks** - This refers to a situation where each agent does not have direct communication with every other agent, but instead can only communicate with a few neighbors.
- **Data transfer limits** - This is a situation where an agent may not be able to send an entrire neural network over the internet, due to connection speed.
- **Low processing power agents** - This situation is where agents have much lower processing power than a standard machine that one may train a model on. For example, agents may not have accsess to a GPU.

1.4 Focussing On Project Goals

The plan for this project went through multiple iterations before the final plan was formulated:

1. The initial plan was to *control a swarm of drones to detect objects such as natural disasters or people needing help, whilst also improving accuracy of the model over time*
2. After discussion, the plan was changed to be *perform edge processing and distributed object detection on many camera perspectives of an environment to decide where disasters are happening, whilst learning to improve the model over time*
3. Following the initial research phase, the plan was narrowed to *explore ways to optimise swarm learning for distributed detection of a simple abstract object*
4. Finally, after the second phase of research, the settled upon plan became *Investigate possible optimisations of the swarm learning algorithm.* This is the current plan.

The shift of focus away from object detection and towards swarm learning is mainly for two reasons:

1. The author finds the swarm learning aspect of the project to be the most interesting part, especially after researching the subject
2. A general framework of improvements and algorithms on swarm learning is much more useful to the real world than an implementation on a simple simulation.

1.5 Risk Assessment

1.5.1 Personal Issues

Description

This risk entails all personal issues which cause the author to be unable to do work, such as illness.

Risk Calculations

Severity (1-5): 3
Likelihood (1-5): 3
Overall Risk (1-25): **9**

Mitigation

As many sections and modules as possible from the codebase will be designed to have minimal requirements from other sections. This means that, even if the author is unable to work for a period of time, some less important sections can be skipped with minimal effect on the reset of the project.

1.5.2 Hardware Failure - Local Computer

Description

This risk entails a failure on the authors local computer of any kind, such as a graphics card or storage breakage.

Risk Calculations

Severity (1-5): 4
Likelihood (1-5): 2
Overall Risk (1-25): **8**

Mitigation

To mitigate storage based failures, the project will be regularly backed up to *GitHub*. If a core component of the work computer breaks, the author has access to a personal laptop and the *Zepler Labs*. The deep learning environment along with dependencies is backed up to the authors *Google Drive* in the form of a docker image, so that switching to a new computer would be a smooth process.

1.5.3 Hardware Failure - Iridis 5

Description

This risk entails a failure on the *Iridis 5 Compute Cluster* which prevents it from being accessed by the author.

Risk Calculations

Severity (1-5): 5
Likelihood (1-5): 1
Overall Risk (1-25): **5**

Mitigation

Iridis 5 will play a key role in this project when simulating large numbers of agents at once. However, it is possible to simulate lower numbers (around 10) agents at the same time on the authors local machine with a basic dataset. This could be a temporary solution if *Iridis 5* went down for a short time. However, for a more permanent solution, funding may be acquired from the university to run the project on a cluster of *AWS* servers, as the author has some experience in that field.

Chapter 2

Background and Literature Review

2.1 Background Research

2.1.1 Keras and Python

What i did to learn keras and python and prepare myself

2.1.2 Iridis 5

What i did to learn about using iridis 5 and prep

2.2 Literature Review

2.2.1 A survey on federated learning [1]

This paper is a good introductory piece to the field of federated learning. It covers a wide range of topics, without going into too much detail, which was useful to direct the author into reading more specific papers. The paper also has a basic introduction to how federated learning works, and also some potential use cases for federated learning. All in all, this paper helped point the author in the right direction for further research.

2.2.2 Swarm Learning for decentralized and confidential clinical machine learning [2]

Despite the fact that this paper focusses on a medical use case, it is a good introductory piece to the topic of swarm learning. A theme of the paper is privacy, and how it can affect the training speed of models. The paper points out that due to privacy legislation, sharing data between hospitals is not possibly, which can mean that training machine learning models does not reach an optimal outcome. However, a solution is proposed in the form of swarm learning: effectively a decentralised form of federated learning. It is shown that swarm learning can outperform the models trained at individual hospitals without communication.

The paper also gives a good high level summary of how the swarm learning algorithm works. This was particularly useful in giving the author a basic understanding of swarm learning to prepare for further research.

2.2.3 Federated learning in Robotic and Autonomous Systems [3]

- Introduced to using federated learning in robotics
- Real world reasons that federated learning is useful in the field of robotics
- Horizontal/vertical federated learning
- Brief look at federated object detection
- Practical challenges of federated (model upload time, etc)

2.2.4 Decentralized Federated Learning: A Segmented Gossip Approach [4]

- Problem of bandwidth for federated learning
- Had a very cool novel idea that every step each node only needs to pull a segment of the network. This reduces bandwidth and should deffo be somthing to try

2.2.5 Multi-Center Federated Learning [5]

-

2.2.6 FedBN: Federated Learning On Non-IID Features via Local Batch Normalization [6]

-

2.2.7 Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach [7]

-

Chapter 3

Report on Technical Progress

3.1 Implementation 1 - Basic MNIST classifier

A simple *mnist* classifier was built using *Keras* in *Python.* There was only one agent which trained on the dataset, and the main reason for this was to get a baseline for the swarm learning to compete against.

3.2 Implementation 2 - Simple swarm learning

Using *Implementation 1* as a base, a swarm of agents was created. Each agent had access to the whole dataset for training. Every agent also could communicate with every other agent. The agents acted in a loop, where they would each first train for one epoch on their copy of the training set, and then would average their models weights with all of their neighbours weights. This implementation was run on the authors local computer, so could not run more than 5 agents without serious performance problems.

All 5 agents did mange to reach the same level of accuracy on the test set as *Implementation 1.* However, the agents took more time and total epochs to reach this accuracy.

The agents seemed to reach the final accuracy in fewer training steps if the agents training loops were offset by even intervals of time. The author hypothesizes that this may be because when the agents training is synced, some of the agents may skip the just completed training when requesting updates, which can cause training epochs to effectively be lost. This effect needs further investigation.

3.3 Implementation 3 - Reduced dataset swarm learning

Building on *Implementation 2*, this implementation was mainly focussed around reducing the amount of data each agent gets to train on. 2000 samples were selected randomly for each agent from the training set, and these never changed. When the agents were not allowed to communicate, their test accuracy almost always stayed below 90 percent. However, when the agents shared their networks, they achieved much higher test accuracies. Interestingly, the improvement in accuracy after the 90 percent point slowed down significantly.

Chapter 4

Project Planning

4.1 Planned Phases of the Project

The project is split up into a number of phases, each of which must be completed sequentially. During these phases the interim report and final report will be developed as a skeleton in parralell.

1. Research - This phase consists of finding and reading existing literature. It is at this time that the project idea is developed fully.
2. Simple Implementation - In this phase, a proof of concept implementation will be deveolped. It will entail a simple swarm learning algorithm learning on a basic dataset, without any complex problems. This implementation will be build upon in further phases.
3. Interim Report - This phase is focussed on filling out and finalising the interim report.
4. Main development - In this phase, the iterative development of the swarm learning algorithm will occur. This is actually a repetition of four sub-phases:
    (a) Further research on real-world problem
    (b) Implement real-world problem and test current solution
    (c) Implement mitigations and test / evaluate
    (d) Document any discoveries and evaluations of mitigations
5. Final report - In this phase, the final report will be written up.
6. Housekeeping - This is the final phase wihch is used for any extra jobs that were not able to be completed in other phases.

4.2 Completed Work

INSERT GANT CHART HERE

4.3 Remaining Work

INSERT GANT CHART HERE

Bibliography

[1] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[2] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz, S. Ktena, F. Tran, M. Bitzer, S. Ossowski, N. Casadei, C. Herr, D. Petersheim, U. Behrends, F. Kern, T. Fehlmann, P. Schommers, C. Lehmann, M. Augustin, J. Rybniker, J. Altmüller, N. Mishra, J. P. Bernardes, B. Krämer, L. Bonaguro, J. Schulte-Schrepping, E. De Domenico, C. Siever, M. Kraut, M. Desai, B. Monnet, M. Saridaki, C. M. Siegel, A. Drews, M. Nuesch-Germano, H. Theis, J. Heyckendorf, S. Schreiber, S. Kim-Hellmuth, P. Balfanz, T. Eggermann, P. Boor, R. Hausmann, H. Kuhn, S. Isfort, J. C. Stingl, G. Schmalzing, C. K. Kuhl, R. Röhrig, G. Marx, S. Uhlig, E. Dahl, D. Müller-Wieland, M. Dreher, N. Marx, J. Nattermann, D. Skowasch, I. Kurth, A. Keller, R. Bals, P. Nürnberg, O. Rieß, P. Rosenstiel, M. G. Netea, F. Theis, S. Mukherjee, M. Backes, A. C. Aschenbrenner, T. Ulas, A. Angelov, A. Bartholomäus, A. Becker, D. Bezdan, C. Blumert, E. Bonifacio, P. Bork, B. Boyke, H. Blum, T. Clavel, M. Colome-Tatche, M. Cornberg, I. A. De La Rosa Velázquez, A. Diefenbach, A. Dilthey, N. Fischer, K. Förstner, S. Franzenburg, J.-S. Frick, G. Gabernet, J. Gagneur, T. Ganzenmueller, M. Gauder, J. Geißert, A. Goesmann, S. Göpel, A. Grundhoff, H. Grundmann, T. Hain, F. Hanses, U. Hehr, A. Heimbach, M. Hoeper, F. Horn, D. Hübschmann, M. Hummel, T. Iftner, A. Iftner, T. Illig, S. Janssen, J. Kalinowski, R. Kallies, B. Kehr, O. T. Keppler, C. Klein, M. Knop, O. Kohlbacher, K. Köhrer, J. Korbel, P. G. Kremsner, D. Kühnert, M. Landthaler, Y. Li, K. U. Ludwig, O. Makarewicz, M. Marz, A. C. McHardy, C. Mertes, M. Münchhoff, S. Nahnsen, M. Nöthen, F. Ntoumi, J. Overmann, S. Peter, K. Pfeffer, I. Pink, A. R. Poetsch, U. Protzer, A. Pühler, N. Rajewsky, M. Ralser, K. Reiche, S. Ripke, U. N. da Rocha, A.-E. Saliba, L. E. Sander, B. Sawitzki, S. Scheithauer, P. Schiffer, J. Schmid-Burgk, W. Schneider, E.-C.

Schulte, A. Sczyrba, M. L. Sharaf, Y. Singh, M. Sonnabend, O. Stegle, J. Stoye, J. Vehreschild, T. P. Velavan, J. Vogel, S. Volland, M. von Kleist, A. Walker, J. Walter, D. Wieczorek, S. Winkler, J. Ziebuhr, M. M. B. Breteler, E. J. Giamarellos-Bourboulis, M. Kox, M. Becker, S. Cheran, M. S. Woodacre, E. L. Goh, J. L. Schultze, C.-. A. S. (COVAS), and D. C.-. O. I. (DeCOI), "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, pp. 265–270, Jun 2021.

[3] Y. Xianjia, J. P. Queralta, J. Heikkonen, and T. Westerlund, "Federated learning in robotic and autonomous systems," *Procedia Computer Science*, vol. 191, pp. 135–142, 2021. The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 16th International Conference on Future Networks and Communications (FNC), The 11th International Conference on Sustainable Energy Information Technology.

[4] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," 2019.

[5] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning," *CoRR*, vol. abs/2005.01026, 2020.

[6] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," 2021.

[7] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 3557–3568, Curran Associates, Inc., 2020.