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

Chapter 1

Project Goals

1.1 Problem

1.2 Proposed Solution

To investigate possible optimisations of the swarm learning algorithm on simple problems, whilst adding real world constraints incrementally.

To begin, a basic form of swarm learning will be implemented. It will be extremely simple, and used as a basic proof of concept. Following this, real world constraints such as bandwidth limits, unevenly distributed features, and sparsely connected networks of agents, will be added sequentially. After the addition of each new constraint, mitigations and testing will be done to minimise the effect of the constraint. This will result in a robust framework of algorithms and improvements for swarm learning that could be applied to many real world problems.

1.3 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 en-*

vironment 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.

1.4 Risk Assessment

1.4.1 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.

1.4.2 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.

1.4.3 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 Report of Literature Search

2.1 Literature Review

2.1.1 A survey on federated learning [1]

- Introduction to fed learning
- Understand basic concept of fed learning
- Had potential use cases for fed learning

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

- Introduced to swarm learning
- Showed benefits of sward/federated learning over conventional central learning
- Helped me to understand how swarm learning can improve federated learning

2.1.3 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)

2.1.4 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

2.1.5 Multi-Center Federated Learning [5]

-

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

-

2.1.7 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

Plan of Remaining Work

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.