Universiteit van Amsterdam

Projectplan

The quest to finding the best SNNI

16 november 2022

Student:
Jorit Prins
12862789

Supervisor:
Zoltan Mann

Cursus:
Afstudeerproject

1 Relevance

The recent rise in Big Data increased the data exchange on the internet. With more and more computer resources available, researchers quickly started to utilise the possibilities of machine learning (ML) to analyse the data. Techniques like neural networks (NN) are promising ways to scientific breaktroughs. Machine learning has a wide varity of applications for classification such as traffic analysis, image recognition, intrusion detection, spam detection, medical or genomics predictions, financial predictions and face recognition (Dowlin e.a. 2017; Gilad-Bachrach e.a. 2016; He e.a. 2015; Islam e.a. 2011)

The use of ML typically consists of two phases: training and inference. In the first phase a NN is trained by feeding an extensive dataset to find the best parameters. NNs used for machine learning have to be maintained, evaluated and the training phase is often a tedious and time exhausting process. In the inference phase an input is applied to the trained NN. Because of the time consuming process of creating a NN, machine learning as a service (MLaaS) became popular. In MLaaS, a company offers a pre-trained NN to the clients. Now, clients only need to worry about the inference phase.

However, MLaaS offers great threats to privacy. To train the model as accurately as possible a NN needs access to a large amount of precise data from clients, which may consist of sensitive information. Thus, clients may be reluctant to provide the NNs with their data. Other features, irrelevant to the prediction task, could also be derived from this data (Nasr, Shokri en Houmansadr 2019). On the inference phase, input from the client to the NN can also be confidential. On the other hand, owners of a NN could be worried that an adversary could steal (parameters of) their (often costly) NN. Furthermore, the result of the NN could also be confidential resulting in the need to retain this information from unauthorized parties.

A typical MLaaS situation consist of two parties: the client holding an input x and a company holding neural network *f*. For this research we will focus on the inferrence part. The client wants to know the applied input *f(x)* while keeping the sensitive contents of x and the result *f(x)* private from the company. The company wants to hold the intelectual property f private while still giving the opportunity to the client to use *f*. The secure neural network inference (SNNI) problem entails calculating the applied input *f(x)* while still holding all the above security requirements.

No general implementation of an SNNI has been widely accepted to the authors knowledge. Rapid progress in this area has made it hard to get a good overview of technological advances. Mann et al. Mann22 has summarized several proposed approaches for SNNI. However, these approaches are often proof-of-concept and are not thoroughly tested. Moreover, the performance is often only tested on basic measures like efficiency or privacy and tests are on a basic setup (laptop to local server).

Metrics like power usage are important in the internet of things (IoT), where clients are constrained to use devices with limited performance. This limits the applicability of certain approaches that put burden on the client, and the server side should be able to handle many requests.

2 Research question(s) and Method

To contribute to the prior research in this area, this paper will focus on testing one (or more) already existing open source implementations, for example ABY2.0 (Patra e.a. 2020), Chameleon (Riazi e.a. 2018), Cheetah (Huang e.a. 2022), CrypTFlow2 (Kumar e.a. 2019) and Delphi (Mishra e.a. 2020) on their power usage.

To select an existing approach we need to set our constraints: the approach should be available for everyone and there should already be an existing implementation that is open source. To select the approach to test further research lies is in the following research question:

> *RQ1: what is needed to make a SNNI approach good and realistic (e.g. setup and computing power)?*

RQ1 will be answered with an extensive literature research. After this literature research we can also answer the following question:

> RQ2: what are sufficient metrics to test a SNNI approach (e.g. power usage)?

and

> RQ3: how can we test these metrics on our SNNI

We can start selecting our desired implementation once we have answerd *RQ1.* As soon as we have selected our SNNI approach, we can start downloading the SNNI and making it work at our own setup. Once we have got it all working we can start testing the implementation. We test the implementation with the answers found on research question *RQ2.* A small literature search can be done on what databases to use for our testing puroposes, resulting in the following question:

> RQ3: what databases should we use for our testing?

3 Schedule

(will come once research questions are formulated good enough =¿ see email)

Referenties

Dowlin, Nathan e.a. (2017). „Manual for Using Homomorphic Encryption for Bioinformatics". In: *Proceedings of the IEEE*, 1–16. DOI: 10.1109/jproc.2016.2622218. URL: https://doi.org/10.1109/jproc.2016.2622218.

Gilad-Bachrach, Ran e.a. (20–22 Jun 2016). „CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy". In: *Proceedings of The 33rd International Conference on Machine Learning*. Red. door Maria Florina Balcan en Kilian Q. Weinberger. Deel 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, p. 201–210.

He, Kaiming e.a. (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. DOI: 10.48550/ARXIV.1502.01852. URL: https://arxiv.org/abs/1502.01852.

Huang, Zhicong e.a. (aug 2022). „Cheetah: Lean and Fast Secure Two-Party Deep Neural Network Inference". In: *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, p. 809–826. ISBN: 978-1-939133-31-1. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/huang-zhicong.

Islam, Naveed e.a. (sep 2011). „Application of homomorphism to secure image sharing". In: *Optics Communications* 284.19, p. 4412–4429. DOI: 10.1016/j.optcom.2011.05.079.

Kumar, Nishant e.a. (2019). *CrypTFlow: Secure TensorFlow Inference*. DOI: 10.48550/ARXIV.1909.07814. URL: https://arxiv.org/abs/1909.07814.

Mishra, Pratyush e.a. (aug 2020). „Delphi: A Cryptographic Inference Service for Neural Networks". In: *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, p. 2505–2522. ISBN: 978-1-939133-17-5. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/mishra.

Nasr, Milad, Reza Shokri en Amir Houmansadr (mei 2019). „Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning". In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. DOI: 10.1109/sp.2019.00065.

Patra, Arpita e.a. (2020). *ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation*. Cryptology ePrint Archive, Paper 2020/1225. https://eprint.iacr.org/2020/1225. URL: https://eprint.iacr.org/2020/1225.

Riazi, M. Sadegh e.a. (2018). *Chameleon: A Hybrid Secure Computation Framework for Machine Learning Applications*. DOI: 10.48550/ARXIV.1801.03239.