Explore the Possibility of Fine-grained Non-encrypted Distributed MLaaS: an Adversary View

Progress Talk for *CyberSec & Privacy*, Type: Research

Bohan Cui

Nanjing University, Department of Computer Science and Technology

Nov 2024

1. Clarification of Problem and Methodology

2. Filling the Logic Chain

3. First-stage Experiment

4. Challenges and Solutions

5. Next Step

The problem

- Old: IP protection in distributed AI model training
- Our assumption: The model is uncrypted and the pirates **can not** control all the devices. [1] All the devices are untrusted.
- Our goal: To protect the IP of the model, including getting a copy of the **whole** model or get **similar** performance of the model with efforts **lower than** retraining the model.

[1] We do not take **Trusted Execution Environment** into consideration, for it is not practical in our scenario.

The Methodology: Adversary View

- Not the Adversary Analysis in the Complexity Theory
- Like the Repeated Game in Game Theory: We assume there is a pirate who wants to get the model. We update our strategy to protect the model according to the pirate's strategy, so does the pirate.

1. Clarification of Problem and Methodology
2. Filling the Logic Chain
3. First-stage Experiment
4. Challenges and Solutions
5. Next Step

S_0 (S for Service Provider): Our model for this problem

- We assume there exists an untrusted distributed system, which combine the pipeline parallelism and the data parallelism.
- Therefore each of the devices hosts a part of the model.

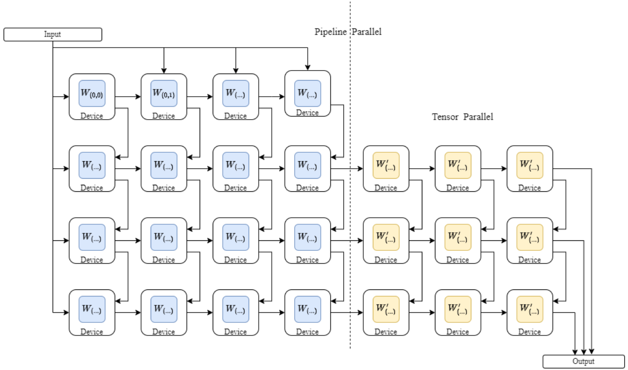


Fig. 1: Model

P_1 (P for Pirate): Naive Attack

- Just acquire part of the model from the devices.
- BUT how much is enough? (Please refer to the next part)

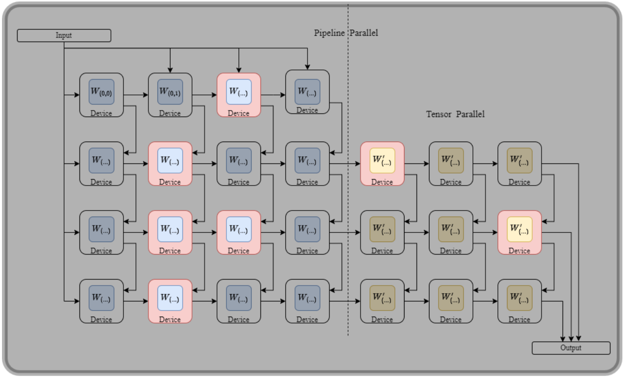


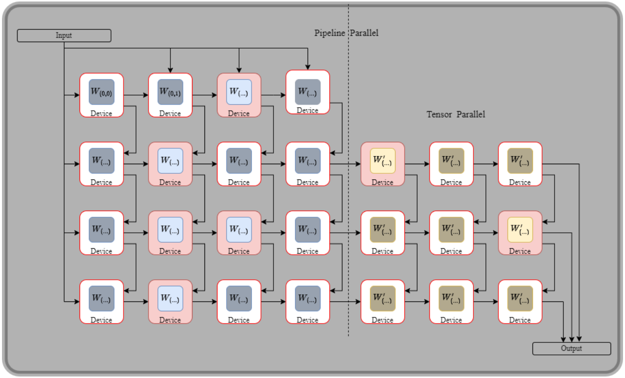
Fig. 2: Naive Attack

S_1: Not so promising: Complexity of the Operators!

- The posibility of putting the parameters to the right place is very low:
- $\frac{1}{A_N^N} = \frac{1}{O(N!)}$
- How about N ? Very large in our scenario. ('Fine-grained')
- How about some of them are not in the right place? (Please refer to the next part)

P_2: More technical Attack

- How could be search in such a large space?
- We could use machine learning to help us.
- Abstraction: Given a function $\mathbf{Val}$, a set of values S, among all the orders of S, find $max(\mathbf{Val}(\mathbf{order}(\mathbf{S})))$.



S_2: Do not be happy too early: Complexity of the Architecture!

- All of them are just floating-point numbers or integers tensors, how could you know which operator they belong to and which layer do they belong to?[2]
- You can not tell the difference between the 'Yellow' and the 'Blue' in the last slide!

[2] Or you may even do not know the what layers the model has.

P_3: Stream Hijacking

- To get the architecture of the model, the pirate can monitor the stream between the devices and the centralized server.
- It is a Pattern Recognition task! [3]

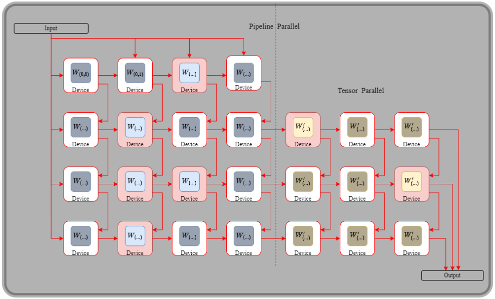


Fig. 4: Take the stream

[3] It could be a self-supervised learning task, and I will work on it.

S_3: Stream Obfuscation

- Padding all the messages to the same length.
- Add some useless messages to the normal stream, in case of being analyzed by the pirate.
- Here is a trade-off between the additional stream and the communication cost.[4]

[4] I will work on a feasible algorithm here.

1. Clarification of Problem and Methodology
2. Filling the Logic Chain
3. First-stage Experiment
4. Challenges and Solutions
5. Next Step

Only get part of the model (in the right position)

- Actually it just a prune task. 256x256 divided and drop
- When the model has 10 percents of the parameters, there are limited loss compared to the original model.

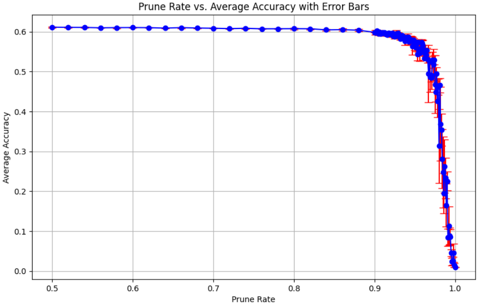


Fig. 5: Exp1

Only get part of the model (not int the right position)

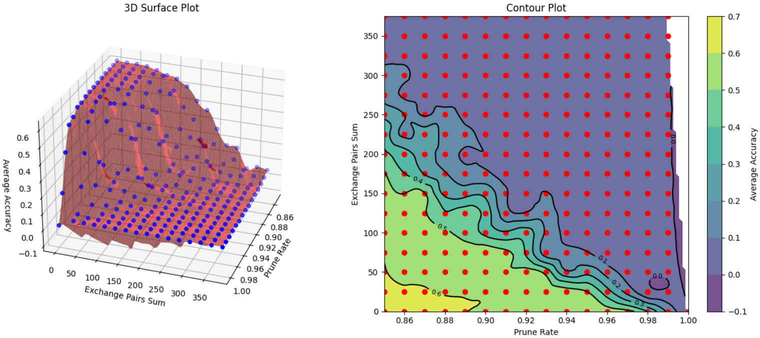


Fig. 6: Exp2

1 Clarification of Problem and Methodology

2 Filling the Logic Chain

3 First-stage Experiment

4 Challenges and Solutions

5 Next Step

Methodology Problems (and maybe my feelings)

- I once chose llama 3.1 as the object of the experiment. And I tried Whisper-3large as the objects too.
- Not Good, too complicated. (Actually I still need to learn the structure of these models from 0.)
- I did not follow the regular logical flow that *from simple to complicated*

Trivial Problems (and maybe my feelings)

- Within the great wall, it is too hard to conduct the experiments on the ML models.
- Most tools are not valid and call for special configurations. (Huggingface-cli, Kaggle, Dataloader, transformer package, Docker...)
- Downloading is too slow, abort some plans for the size. e.g. the ImageNet.

Trivial Problems (and maybe my feelings)

Poor!

Fig. 7: Rubbish

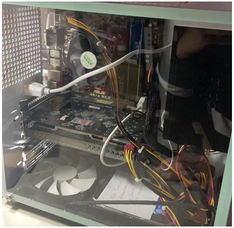


Fig. 8: my 750Ti, old and exhausted

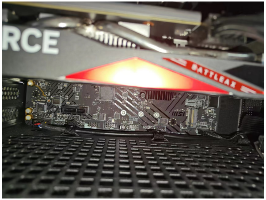


Fig. 9: Ron's 聖女騎士です 4060 saved from the mining farm.

- Actually I have no hardware to do training & inference.
- Thanks to the sponsor of my friend Ron Zhang(THU, Dep of Auto), otherwise I cannot see the results until the universe~~(or, maybe, the NVIDIA)~~ is down.

1 Clarification of Problem and Methodology

2 Filling the Logic Chain

3 First-stage Experiment

4 Challenges and Solutions

5 Next Step

- P_2, P_3 and S_3.
- Something breaks the assumption.
  Callback:
  Our assumption: The model is uncrypted and the pirates **can not** control all the devices. All the devices are untrusted.
  1. S_4: What about part of the model is trusted?
     P_5: We can try it! [5]
  2. P_n: What about most of the devices are pirates'?
     ~~I 'm Sauron! Errrrr~~
     S_{n+1}: ...
- Paper(report, exactly) writing.(have being working on it...)

[5] I will not work on this but just do a Literature Review, for it is an individual domain named model inverse attack.

Thanks!