Project Requirements

Securing Machine Learning Models from Adversarial Attacks

Yishay Asher & Steve Gutfreund
Supervisor: Assi Barak, Research Director at BIU Cyber Center

November 2018

1 Reconstruction of the article

In the article[1], O.Taran, S.Rezaeifer and S.Voloshynovskiy claim that most of the deep learning architectures are vulnerable to adversarial examples.
Their purpose is to question the security of deep neural networks (DNN).
For that, they used two well-known algorithms to create adversarial examples:

- Fast Gradient Sign Method (FGSM)
- an attack proposed by N.Carlini and D.Wagner (CW)

They created two different architectures for DNN classifiers which are trained on the MNIST and Fashion-MNIST datasets.
In order to secure their models against the attacks they used cryptographic tools. They used a random permutation based on the secret key k on every example.
Our goal is to reconstruct the above while relating to different possible encryption function modes:

- ECB
- CBC
- CTR

We'll analyze our results and compare them with the ones of the article.
In case one of the encryption modes failed to secure the model (i.e. we still get a too high classification error rate on the adversarial examples), we will explain the reason for that and show why the model is vulnerable.

[1] `https://arxiv.org/pdf/1809.01715.pdf`
Bridging machine learning and cryptography in defence against adversarial attacks

2 Experimenting with models trained on more complex datasets

We are interested to question the security of more complicated models. We'll train a model on the CIFAR-10 dataset and check if it's secure. If we get a positive answer, we might try some other dataset as well.

3 Alternative encryption function

We would like to consider using different encryption function and see what security they can provide on the models described before.

4 Develop alternative encryption methods based on Kerckhoffs's second cryptographic principle

According to the Kerckhoff's second principle[2], the classification algorithm should be known to all, only the key itself should be secret.
Based on this, we will try to come up with some encryption methods in order to secure the machine learning models.
We might try the following variants:

- encrypt the input
- encrypt the output of one (or many) of the layers
- encrypt the parameters of the layers (like the weight matrices)

[2] https://en.wikipedia.org/wiki/Kerckhoffs%27s_principle