Final Year Project Specification

Session 2017-2018

Marcello De Bernardi

20/10/2017

Contents

1 Project aims

The aim of the project is to explore how effectively neural networks can learn cryptographic functions - in order to bootstrap an end-to-end encrypted communication protocol that ensures confidentiality - by developing an adversarial neural network model to be trained and evaluated.

The minimum project aim is to independently reproduce the findings presented by Abadi and Andersen in their paper CITATION HERE, which demonstrated that adversarially trained neural networks are capable of learning "how to perform forms of encryption end decryption, and also how to apply these operations selectively in order to meet confidentiality goals."

Moreover, if possible, the project aims to move beyond the exploratory work carried out by Abadi and Andersen, as well as further work by other teams PASQUALE LINKS GO HERE, and explore ways in which neural networks may be trained to learn more effectively in this domain, or to produce more convincing results.

2 Methodology

The project The initial phase of the project will consist of background reading in machine learning (neural networks) and cryptography (private key encryption), as well as a review of the research literature on the subject. Further learning will include familiarization with the Python programming language as well as the TensorFlow API.

Initial implementation efforts will go towards building a simple adversarial neural network system, as well as designing a quantitative evaluation method for the nets' performance. The interim report and risk assessment will be written based on the results obtained from this prototype.

The prototype will be folled by a larger implementation, using more complex neural network architectures.

Once the implementation is largely finalized, the model will be trained on the HPC cluster to generate result data. Refinements may be made to the model at 5. Write report

3 Project Milestones

1. Specification 2. Initial implementation for testing 3. Interim report 4. Finalized implementation 5. Training data 6. Project report

4 Required knowledge, skills, tools, and resources

The project requires background knowledge in the areas of machine learning and cryptography, with a particular emphasis on generative adversarial models and private-key encryption.

The main skills involved are Python programming (particularly in procedural and object-oriented styles), as well as working with the TensorFlow API and other supporting toolsets. In addition, an understanding of neural network architecture will be required.

A variety of free and proprietary development tools will be used. The implementation will be written in Python, and make heavy use of TensorFlow, an open-source software library for machine learning. TensorFlow provides a supporting tool, TensorBoard, which allows for visualization of TensorFlow models. Most of the development will be done in Atom, a modular and highly customizable text editor. One of the primary appeals of Atom lies in a package called Hydrogen, which allows the user to interactively run snippets of code while inspecting variables or generating visual graphs. This is similar to the popular Jupyter Notebook.

The software to be developed is expected to be shorter than 500 lines of code, so no major resources will be required during the development phase.

The training phase will require more computational power than is available on a commodity PC or laptop, so an access request to the Queen Mary University of London HPC will be made.

5 Timeplan