How to build privacy and security into deep learning models

Yishay Carmiel
@YishayCarmiel
The evolution of AI
AI has evolved a lot over the last few years

- Speech Recognition
- Computer Vision
- Machine Translation
- Natural Language Processing
- Reinforcement Learning
AI Applications are evolving

Alexa / Google Home

Autonomous driving

Machine Translation

Google Duplex
Data Privacy is evolving as well

- GDPR
- Facebook and Cambridge Analytica
- Data privacy regulations
Can they work together?
If AI is the new software, how can we protect it?
The Evolution of Security solutions

- Desktop Applications / Security
- Mobile Applications / Security
- Cloud Applications / Security
- AI Applications / Security
Why is it interesting?
Moving into the cloud – Cloud is not trustable

OpenAI Blog – AI and Compute
Sharing data and models

• How can multiple parties share data?

• How can multiple parties work together in the data ↔ Model structure
Attacks in the Physical world
DeepFake and Neural Voice Cloning
Privacy and Stability of models
Privacy and memorization

• Can a neural network remember data or expose data that is was train on?

• In various Machine Learning applications we need to make sure model does not remember or can expose data.
  • Medical records: personal medical information
  • Transaction information: SSN and Credit Cards
  • Sensitive imagery data

• It is able to reconstruct data from a NN model through API’s

• How can we evaluate privacy of an algorithm?
Memorization

• Nicholas Carlini et al. *The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets*

• Introducing the notion of memorization, evaluating if a NN can remember information

• Introducing a metric to evaluate privacy of NN.

• Other works to evaluate privacy of NN:
  • Model stealing: trying to reconstruct the model parameters
  • Attack that attempts to learn aggregate statistics about the training data, potentially revealing private information
Differential Privacy
Differential Privacy (DP)

• Differential privacy is a framework for evaluating the guarantees provided by a mechanism that was designed to protect privacy

• Introducing randomness to a learning algorithm

• Making it hard to tell which behavioral aspects of the model defined by the learned parameters came from randomness and which came from the training data

• One method for DP on NN is PATE (Private Aggregation of Teacher Ensembles) Papernot, Goodfellow et al Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data
Differential Privacy (DP)

- Partition the data into multiple sets, train multiple teacher networks
- Each inference is based on multiple teacher voting + random noise

Privacy and machine learning: two unexpected allies?
TensorFlow Privacy

• TensorFlow framework for differential privacy

• Main idea is based adding random noises to the gradient:
  • Differentially Private Stochastic Gradient Descent (DP-SGD)
  • Martin Abadi et al Deep Learning with Differential Privacy (10/2016)
  •

• Every optimizer can be replaced with a DP optimized
  • AdamOptimizer → DPAdamGaussianOptimizer
  • The DP optimizer has 3 more parameters to support DP

• For more information: https://github.com/tensorflow/privacy/blob/master/tutorials/walkthrough/walkthrough.md
Getting Started

Blog Post:

https://github.com/tensorflow/privacy/tree/master/tutorials

Code:

https://github.com/tensorflow/privacy
https://github.com/tensorflow/models/tree/master/research/differential_privacy/pate
Machine Learning on Private Data
Machine Learning Workflow

- **Extraction**
  - Raw Data
  - Features and Labels

- **Training**
  - Training Set
  - Validation Set
  - Test Set
  - Machine Learning
  - Model

- **Inference**
  - Features
  - Production Model
  - Predicted Labels
Training on Private Data
Train on Private Data – Data Protection

- **Edge Devices data export:** Prevent data going out of the edge device
  - Mobile Devices
  - Sensors (IoT)

- **Sharing data without exposing it:** Multiple sources want to achieve a common goal without exposing data content. i.e. Common goal – train a NN model

- Preventing data reconstruction
Train on Private Data Techniques

**Federated learning:**
Training data on edge devices without exporting data from the device

**SMP (Secure Multi-Party) Training**
When multiple parties want to achieve a common goal (model) without sharing the data with each other

**Encryption protocols**
Due to the security aspects of that, Federated learning and SMP involve advanced encryption protocols, maintaining the mathematical calculations.

**Neural Based Differential Privacy**
Techniques for training without exposing data through model attacks.
Federated Learning
Federated Learning

- Multiple devices are working together to create a single model
- A copy of the model is downloaded into the device
- Device calculates on model update
- The server calculates the overall average

\[
\min_{w \in \mathbb{R}^d} f(w) \quad \text{where} \quad f(w) \equiv \frac{1}{n} \sum_{i=1}^{n} f_i(w). \tag{1}
\]

\[
f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad \text{where} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w).
\]

H. Brendan McMahan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data
Federated Learning – Secure aggregation

- Aggregation – The centralized system needs the average of all the updates
- Security - This needs to be done in a secured manner without sharing updates with different parties
- Secure Aggregation Encryption protocol:
  - In order to calculate the overall average without sharing data a dedicated encryption protocol is used.
  - Keith Bonawitz et al Practical Secure Aggregation for Privacy-Preserving Machine Learning

Keith Bonawitz et al Practical Secure Aggregation for Privacy-Preserving Machine Learning
Federated Learning – Encryption and limitations

- **Limitations:**
  - Model Size
  - Differential Privacy, data is not really protected
  - Communication between devices and server

Google AI Blog – Federated Learning
Secure Training – Open Sources

- OpenMined is an open source for secured machine learning
  - https://www.openmined.org/
- TF Federated, federated learning using TensorFlow
  - https://github.com/tensorflow/federated
Inference on encrypted data
Inference on Private Data

• Sharing or disclosing the data is an issue, inference without data disclosure is a natural solution

• On premise solutions are challenging, organization ideally can move their machine learning inference into the cloud

• Prevents from model disclosure
Encryption methods for secure calculation

Multi-Party Computation (MPC)
MPC is a way by which multiple parties can compute some function of their combined secret input without any party revealing anything more to the other parties about their input other than what can be learnt from the output.

Secret Sharing
A set of methods for distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types, of shares are combined together; individual shares are of no use on their own.
Encryption methods for secure calculation

**Garbled Circuits**
Cryptographic protocol that enables two-party secure computation in which two mistrusting parties can jointly evaluate a function over their private inputs without the presence of a trusted third party.

**Homomorphic encryption**
A form of encryption that allows computation of cipher texts

- Partially Homomorphic Encryption: A cryptosystem that supports specific computation on ciphertexts
- Fully Homomorphic Encryption (FHE): A cryptosystem that supports arbitrary computation on ciphertexts

\[
\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = x_1^e x_2^e \mod m = (x_1 x_2)^e \mod m = \mathcal{E}(x_1 \cdot x_2)
\]

\[
\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{r_1^m})(g^{r_2^m}) \mod m^2 = g^{r_1+r_2}(r_1 r_2)^m \mod m^2 = \mathcal{E}(x_1 + x_2)
\]

Unpadded RSA

Paillier
Problems and limitations

Encryption calculation is still a very slow process, very impractical at this stage

**Optimization Techniques**

- Polynomial approximation of neural network activation functions
- FHE or HE optimization
- Optimization on the encryption protocol
  - Neural Network based optimization
  - SPDZ protocol optimization
  - SS optimization
  - Secure tensor operation optimization

**Limitations**

- All evaluation are on simple or classical NN topologies and not recent ones
- No tangible use cases, most work is theoretical or basic CV tasks (MNIST, CIFAR)
- Calculation is still slow compared to non-encrypted techniques
Privacy preserving inference Open Source

HElib – Homomorphic Encryption library
https://github.com/shaih/HElib

TinyGrable - a full implementation of Yao’s Grabled Circuit (GC) protocol
https://github.com/esonghori/TinyGarble

TF – Encrypted
https://github.com/mortendahl/tf-encrypted

OpenMined.org
https://github.com/OpenMined/
Adversarial Attacks and Deep Fakes
The trust model

Data Owners
The owners or trustees of the data/environment that the system is deployed within.

Service Owners
Construct the system and algorithms, e.g., the authentication service software vendors.

Customers
Consumers of the service the system provides, e.g., the enterprise users.

Outsiders
May have explicit or incidental access to the systems, or may simply be able to influence the system inputs.

Trust Model
A trust model, assigns a level of trust to each party within that deployment. Any party can be trusted, untrusted, or partially trusted. i.e. trusted to perform or not perform certain actions.
Adversarial Capabilities

Inference Phase Attacks
White Box Attacks: The adversary has some information about the model or its original training data
• Can be disfurnished further based on the information used: model architecture, model parameters, training data, or combinations of these.
• The adversary exploits this information to evaluate where the model is vulnerable

Black Box Attacks: Assume no knowledge about the model
• The adversary in these attacks use information about the setting or past inputs to infer model vulnerability

Training Phase Attacks
Attempt to learn, influence or corrupt the model itself
• Altering the training data by inserting adversarial inputs into the existing training data (injection)
• Altering the data collection process by direct attacks via untrusted data collection component
Adversarial Goals

Confidentiality and Privacy
Attacks are with the respect of model and data
- The model or its hyperparameters can be considered confidential, for example financial markets
- ML models have tendency to memorize information on the data, an attack can try reconstruct the data or some high level statistics on the data.
- Example: Reconstruction SSN or Credit cards from a language model trained on private data

Integrity and availability
The goal is to induce model behavior as chosen by the adversary, attempting to control model outputs
- ML confidence can be targeted
- Supervised task— wrong classifier or noise with high confidence
- Unsupervised task – Meaningless feature representation
- Example: Forcing an ADAS system to miss detect traffic sign
Integrity Attacks
What is an adversarial attack?

- Subtly modifying an original image in such a way that the changes are almost undetectable to the human eye.

- The modified image is called an adversarial image, and when submitted to a classifier is misclassified.

Ian Goodfellow et al: Explaining and Harnessing Adversarial Examples

Alexey Kurakin et al: A Adversarial examples in the physical world
The basic idea of attacks

Modifying the image
Modify the image towards the direction of the gradient of the loss function with respect to the input image

One-Shot Attacks
The attacker takes a single step in the direction of the gradient

Iterative Attacks
Multiple steps in the direction of the gradient
Attacks in the Physical world

- In oppose to classical cyber attacks, neural networks attacks can be done in the physical world.

- Specific printed patches or stickers in unique places can fool machine learning systems.

- Open the attacks to much broader range of attackers, no a-priori knowledge


- Tom B. Brown et al: Adversarial Patch (05/2018)
Attacks on Q&A and LM systems

- Attacks can fool even hybrid vision and NLP systems
Good References on adversarial attacks


Getting Started:

https://github.com/IBM/adversarial-robustness-toolbox
Neural Networks that fool us
DeepFake and Neural Voice Cloning
The risks with neural networks foolers

Experts Bet on First Deepfakes Political Scandal

Researchers wager on a possible Deepfake video scandal during the 2018 U.S. midterm elections

By Jeremy Hsu

DARPA is funding new tech that can identify manipulated videos and ‘deepfakes’

Taylor Hatmaker  @tayhatmaker / Apr 30, 2018
In Summary

• There are 3 main interesting aspects of AI and Privacy:
  1. Privacy preserving machine learning

  2. How to apply machine learning on private data
     • Training
     • Inference

  3. Fooling neural networks
     • Adversarial attacks on neural networks
       • Confidentiality and privacy attacks
       • Integrity attacks

• Neural networks that fool us
A new field
A new field in AI

- Neural Network Design
- Reinforcement Learning
- Human Language Understanding
- Computer vision

Deep Learning

Security

Encryption
- Secure Computations
- Crypto Networks

Machine Learning

AI Security

Information Security
Thank you!

Yishay Carmiel
@YishayCarmiel