Summary of A few Papers on: Machine Learning and Cryptography

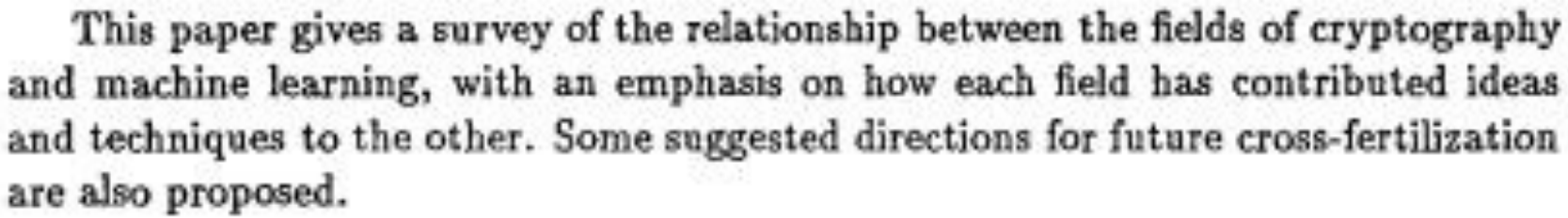
Presenter: Joseph Tobin

Department of Computer Science, University of Virginia

https://qdata.github.io/deep2Read/

Cryptography and Machine Learning (1991)

- Author: Ronald Rivest



This paper gives a survey of the relationship between the fields of cryptography and machine learning, with an emphasis on how each field has contributed ideas and techniques to the other. Some suggested directions for future cross-fertilization are also proposed.

- Look at Kearns [21]

Initial Comparison

- Cryptography produces examples of classes that are hard to learn
  - Pseudo-random functions F_k {0,1}^k -> {0,1}^k
- Secret keys and target functions
  - Secret key equivalent to target function
- Variable length keys and machine learning polynomials of unknown order
- Information known and unknown to attack, create model
  - Know a priori distribution

Initial Comparison

- Cryptography usually wants exact identification of an unknown function
- Machine learning approximates
- Computational complexity
- Information Theory
- Noise as an advantage/disadvantage

Cryptography's impact on Learning Theory

- Valiant showed that work on random functions implied approximately learning class of functions representable by polynomial size boolean circuits is infeasible
  - Focus on identifying which classes of functions are learnable
- Show certain learning problems are computationally intractable
  - Learning theory results on intractability are representation dependent (boolean functions represented a certain way are intractable)
    - 2-term DNF formula consistent with a set of I/O pairs for such a target formula is NP-complete

Cryptography's impact on learning theory

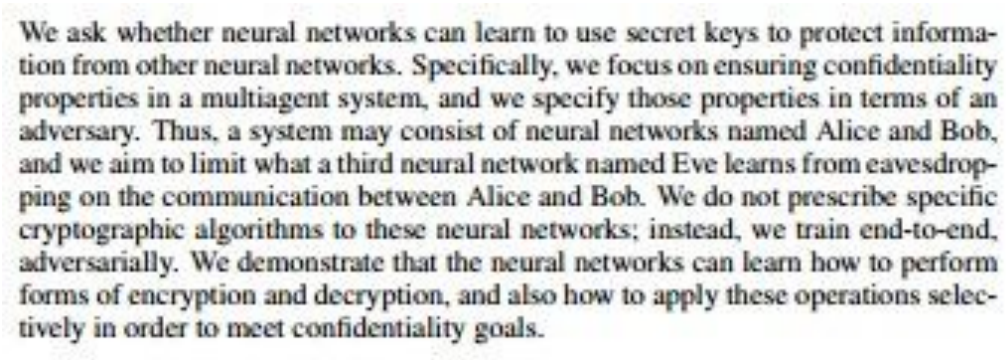
- Representation independent
  - Cryptographic assumptions
- PAC-learing hard for
  - "Small" boolean formulae
  - Class of deterministic finite automata of size at most p(n) and accepts strings of length n
  - Class of threshold circuits over n variables with depth at most d
  - If a machine learning function could learn these, then the algorithm could be used to break one of the cryptographic problems assumed to be hard
- "Prediction-preserving reducibility"
  - Queries asked by learner get translated into chosen-ciphertext requests against Naor-Yung scheme

Learning Theory's impact on Cryptography

- Impact of negative results on learning theory on development of cryptographic schemes
- Learning algorithm can try to infer mapping from plaintext to ciphertext bits (approximately learning 99% of bits)
- Design non-linear feedback shift registers used in cipher-feedback mode
- Application of continuous optimization techniques to discrete learning
- Learning theory leading to better compressive algorithms

Learning to Protect Communications with Adversarial Neural Cryptography (2016)

- Authors: Martin Abadi and David G. Andersen



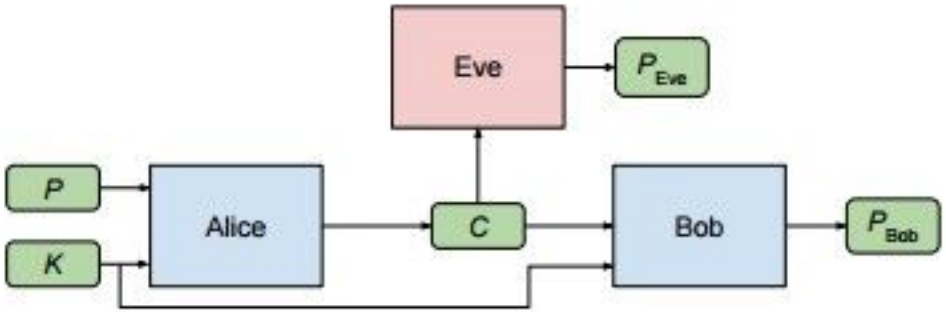
We ask whether neural networks can learn to use secret keys to protect information from other neural networks. Specifically, we focus on ensuring confidentiality properties in a multiagent system, and we specify those properties in terms of an adversary. Thus, a system may consist of neural networks named Alice and Bob, and we aim to limit what a third neural network named Eve learns from eavesdropping on the communication between Alice and Bob. We do not prescribe specific cryptographic algorithms to these neural networks; instead, we train end-to-end, adversarially. We demonstrate that the neural networks can learn how to perform forms of encryption and decryption, and also how to apply these operations selectively in order to meet confidentiality goals.

Introduction

- End to end objective: learn to protect communications in order to satisfy a policy specified in terms of adversary
- Encryption algorithm is said to be secure if no attacker can extract infromation about plaintexts from ciphertexts
- Simple neural nets cannot even computer XOR, but neural networks can learn to proect data from other neural nets by “discovering” encyrption and decryption without being taught
- Previous work: ZooCrypt (dependent on symbolic theorem-proving, not neural networks)

Learning Symmetric Encryption

- Alice and Bob want to communicate securely over public channel (with key K) and Eve wishes to eavesdrop on communications



- Let Alice, Bob, Eve be competing neural nets (inspired by GANs)

Objective Functions

$$L_E(\theta_A, \theta_E, P, K) = d(P, E(\theta_E, A(\theta_A, P, K)))$$

Intuitively, $L_E(\theta_A, \theta_E, P, K)$ represents how much Eve is wrong when the plaintext is P and the key is K. We also define a loss function for Eve over the distribution on plaintexts and keys by taking an expected value:

$$L_E(\theta_A, \theta_E) = \mathbb{E}_{P,K}(d(P, E(\theta_E, A(\theta_A, P, K))))$$

We obtain the "optimal Eve" by minimizing this loss:

$$O_E(\theta_A) = \text{argmin}_{\theta_E}(L_E(\theta_A, \theta_E))$$

Similarly, we define a per-example reconstruction error for Bob, and extend it to the distribution on plaintexts and keys:

$$L_B(\theta_A, \theta_B, P, K) = d(P, B(\theta_B, A(\theta_A, P, K), K))$$
$$L_B(\theta_A, \theta_B) = \mathbb{E}_{P,K}(d(P, B(\theta_B, A(\theta_A, P, K), K)))$$

We define a loss function for Alice and Bob by combining L_B and the optimal value of L_E:

$$L_{AB}(\theta_A, \theta_B) = L_B(\theta_A, \theta_B) - L_E(\theta_A, O_E(\theta_A))$$

Learning Symmetric Encryption

- Alice produces ciphertext C, and Eve and Bob both try to decipher it
  - Bob has advantage because he can use key K to help decrypt
- K, P, C are all floating-point numbers (easier to learn than using bits)
- E learns to minimize the error between P and P_eve
  - Not trying to determine if C “looks random” (could contain metadata)
- Combine A and B loss function to with objective of minimizing the difference between P and P_Bob

Results and Conclusion

- Assume models are neural networks, but alternative models can use reinforcement learning
- Use machine learning to make sense of metadata, traffic analysis, pseudorandom-number generation, integrity checks
-

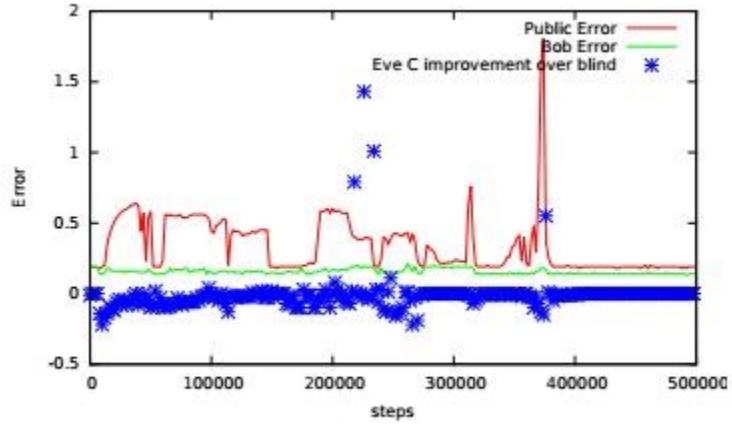


Figure 4: Training to estimate D while hiding C.

Results

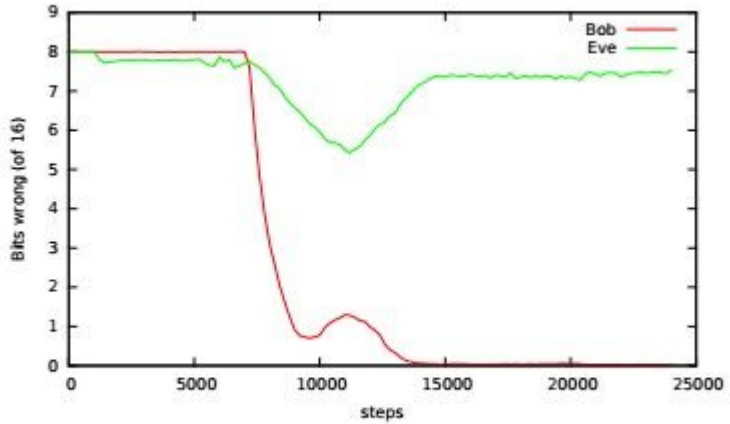
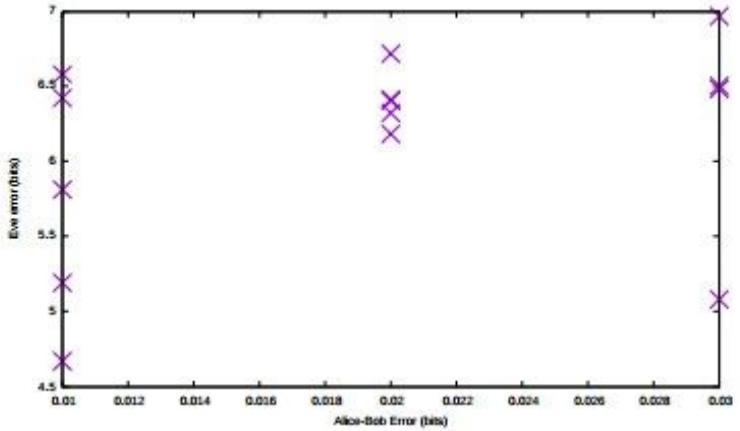


Figure 2: Evolution of Bob's and Eve's reconstruction errors during training. Lines represent the mean error across a minibatch size of 4096.



Results

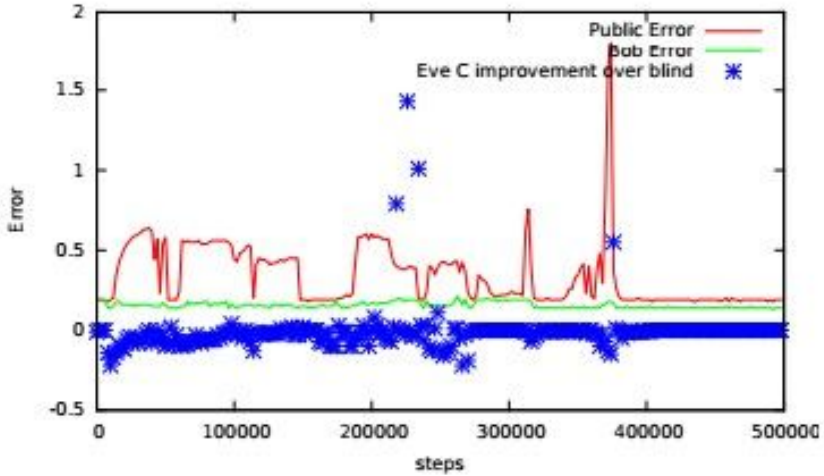


Figure 4: Training to estimate D while hiding C.

Analysis of Neural Cryptography (2002)

- Authors: Alexander Klimov, Anton Mityagin, Adi Shamir

Abstract. In this paper we analyse the security of a new key exchange protocol proposed in [3], which is based on mutually learning neural networks. This is a new potential source for public key cryptographic schemes which are not based on number theoretic functions, and have small time and memory complexities. In the first part of the paper we analyse the scheme, explain why the two parties converge to a common key, and why an attacker using a similar neural network is unlikely to converge to the same key. However, in the second part of the paper we show that this key exchange protocol can be broken in three different ways, and thus it is completely insecure.

Analysis of Neural Cryptography (2002)

- Alice and Bob use “chaotic syncronization” to agree upon key in key exchange protocol
- Genetic attack: create a large population of neural networks and train them as the same inputs as Alice and Bob, only keep ones with similar output
- Geometric attack:
- Probabalistic attack
- Apply attacks to other works?

CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy (2016)

- Authors: Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristen Lauter, Michael Naehrig, John Wernsing

Applying machine learning to a problem which involves medical, financial, or other types of sensitive data, not only requires accurate predictions but also careful attention to maintaining data privacy and security. Legal and ethical requirements may prevent the use of cloud-based machine learning solutions for such tasks. In this work, we will present a method to convert learned neural networks to *CryptoNets*, neural networks that can be applied to encrypted data. This allows a data owner to send their data in an encrypted form to a cloud service that hosts the network. The encryption ensures that the data remains confidential since the cloud does not have access to the keys needed to decrypt it. Nevertheless, we will show that the cloud service is capable of applying the neural network to the encrypted data to make encrypted predictions, and also return them in encrypted form. These encrypted predictions can be sent back to the owner of the secret key who can decrypt them. Therefore, the cloud service does not gain any information about the raw data nor about the prediction it made.

We demonstrate *CryptoNets* on the MNIST optical character recognition tasks. CryptoNets achieve 99% accuracy and can make more than 51000 predictions per hour on a single PC. Therefore, they allow high throughput, accurate, and private predictions.

Summary

- Enable Machine Learning As a Service by allowing model to be trained on encrypted data using homomorphic encryption
- Run as typical model with several changes to adjust for encrypted data
- Other approaches including Multi-Party Computation
- Future work: use GPUs, FPGAs to accelerate computation and find more efficient encoding schemes that allow for smaller parameters

On Lattices, Learning with Errors, Random Linear Codes, and Cryptography (2009)

- Author: Oded Regev

Our main result is a reduction from worst-case lattice problems such as GAPSVP and SIVP to a certain learning problem. This learning problem is a natural extension of the 'learning from parity with error' problem to higher moduli. It can also be viewed as the problem of decoding from a random linear code. This, we believe, gives a strong indication that these problems are hard. Our reduction, however, is quantum. Hence, an efficient solution to the learning problem implies a *quantum* algorithm for GAPSVP and SIVP. A main open question is whether this reduction can be made classical (i.e., non-quantum).

We also present a (classical) public-key cryptosystem whose security is based on the hardness of the learning problem. By the main result, its security is also based on the worst-case quantum hardness of GAPSVP and SIVP. The new cryptosystem is much more efficient than previous lattice-based cryptosystems: the public key is of size $\tilde{O}(n^2)$ and encrypting a message increases its size by a factor of $\tilde{O}(n)$ (in previous cryptosystems these values are $\tilde{O}(n^4)$ and $\tilde{O}(n^2)$, respectively). In fact, under the assumption that all parties share a random bit string of length $\tilde{O}(n^2)$, the size of the public key can be reduced to $\tilde{O}(n)$.

Summary

- Show the reduction of “lattice problem” (used for cryptographic schemes) to a machine learning problem
- ‘Learning from parity with error’ or decoding from a random linear code

$$\langle \mathbf{s}, \mathbf{a}_1 \rangle \approx_\varepsilon b_1 \pmod 2$$
$$\langle \mathbf{s}, \mathbf{a}_2 \rangle \approx_\varepsilon b_2 \pmod 2$$
$$\vdots$$

- Derive public-key cryptosytem whose security is based on hardness of hte learning problem

Differential Privacy and Machine Learning: A Survey and Review (2014)

- Authors: Zhanglong Ji, Zachary C. Lipton, Charles Elkan

The objective of machine learning is to extract useful information from data, while privacy is preserved by concealing information. Thus it seems hard to reconcile these competing interests. However, they frequently must be balanced when mining sensitive data. For example, medical research represents an important application where it is necessary both to extract useful information and protect patient privacy. One way to resolve the conflict is to extract general characteristics of whole populations without disclosing the private information of individuals.

In this paper, we consider differential privacy, one of the most popular and powerful definitions of privacy. We explore the interplay between machine learning and differential privacy, namely privacy-preserving machine learning algorithms and learning-based data release mechanisms. We also describe some theoretical results that address what can be learned differentially privately and upper bounds of loss functions for differentially private algorithms.

Finally, we present some open questions, including how to incorporate public data, how to deal with missing data in private datasets, and whether, as the number of observed samples grows arbitrarily large, differentially private machine learning algorithms can be achieved at no cost to utility as compared to corresponding non-differentially private algorithms.

Summary

- How to train a differentially private model with as little noise as possible?
- Add noise once (if we add multiple times, we divide privacy budget)and occasionally add noise iteratively
- Lower global sensitivity to noise
- Use of public data
-

Aggregative Private Sparse Learning Models Using Multi-Party Computation (Presentation given at SRG)

- Authors: anonymous
- N hospitals want to work together to securely create a machine learning model
- Approach: learn model on local data and then send encrypted parameters to third party
- Third party takes mean of parameters and then returns model back to hospitals

Other papers

- On the use of Recurrent Neural Networks to design Symmetric Ciphers (2008)
  - Similar to deep mind paper
- Power Analysis attack: an approach based on machine learning (2014)
  - Take advantage of high dimensionality to attack model
- Cryptic-Mining: Association Rules Extractions Using Session Log (2015)
  - Similar to deep mind paper

Other papers:

- generation and establishment of cryptographic keys (Ruttor, 2006; Kinzel & Kanter, 2002), and on corresponding attacks (Klimov et al., 2002).