Rudiments of Neural Cryptography

Francesco Moraglio

University of Torino

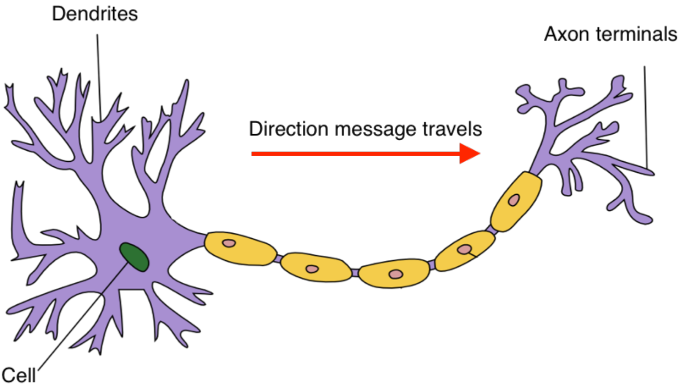
21/07/2020

Overview

1. Neural Networks
2. Genetic Algorithms
3. Neural Cryptography
4. Neural Cryptanalysis
5. Adversarial Neural Cryptography

Neural Networks

Artificial Neural Networks (ANNs) are models of computation based loosely on the way in which the brain is believed to work. A biological neural network consists of interconnected nerve cells, whose bodies are where neural processing takes place.



Artificial Neural Networks

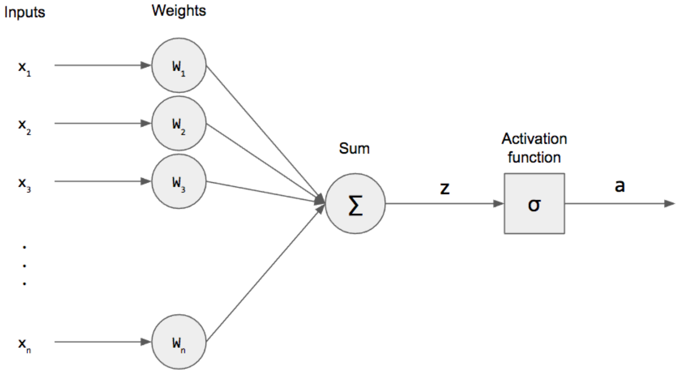
Interconnections between cells are not all equally weighted: this is the key feature modeled by ANNs. Their theoretical principles were firstly formulated in the forties. Among them, a fundamental, widely applied learning principle is the following.

Hebbian Learning Law

When an axon of cell A is near-enough to excite cell B and when it repeatedly and persistently takes part in firing it, then some growth process or metabolic change takes place in one or both these cells such that the efficiency of cell A is increased.

Perceptron: 1

The first complete neural model is called Perceptron and appeared in the late fifties. It serves as a building block to most later models.



Perceptron: 2

The input/output relations of the Perceptron are defined to be

$$z = \sum_i w_i x_i \quad \text{(summation output)}$$
$$y = f_N(z) \quad \text{(cell output)},$$

where w_i is the (adjustable) weight at input x_i. Function f_N is nonlinear and is called activation. Typical activation functions used in ANNs include

- sigmoid function;
- hyperbolic tangent;
- Heaviside step function.

Training

The training of an ANN is the procedure of adjusting its weight. This task can be performed with several techniques. For the simplest architectures, we recall

- **Least Mean Square Training**
- **Gradient Descent Training**

Perceptron: 2

Heading

1. Statement
2. Explanation
3. Example

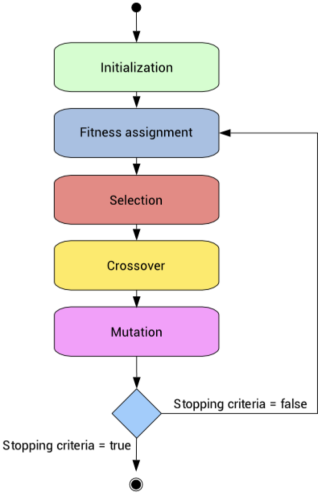
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Genetic Algorithms: 1

Genetic Algorithms (GAs) were invented by John Holland in the 1960s and can be considered a key building block of artificial intelligence. These are stochastic search algorithms that can generate "sufficiently good" solutions to an optimization problem. Nowdays GAs are widely employed in applied science and, in neural cryptography, they can be used to

- optimize network architectures involved in communication;
- perform effective cryptanalitic attacks.

Genetic Algorithms: 2



- The GA is a method for evolving from a population of "chromosomes" (elements of a solution space) to a new population by using an imitation of natural selection together with the genetic-inspired operators of crossover and mutation.
- The main theoretical result on the behavior of GAs is known as Holland Theorem. It implies that the best "building blocks" of the solutions propagate exponentially over time in the population.

Neural Cryptography

- From the nineties on, researchers have made attempts at combining the features of neural networks with cryptography: this field is commonly known as neurocryptography.
- One of the first attempts can be found in [Volná(2000)]. In this work, the author builds a symmetric cryptosystem by making use of neural networks, whose parameters constitute the key of the cipher.
- GAs are used for the optimization of the designed NN topology. Adaptation of the best found network architecture is then finished with BP.

The NNs of Volná

The GA of Volná evolves the architecture of feedforward NNs whose weights are initialized as follows. For every existing connection, three digits are generated and weights are computed as

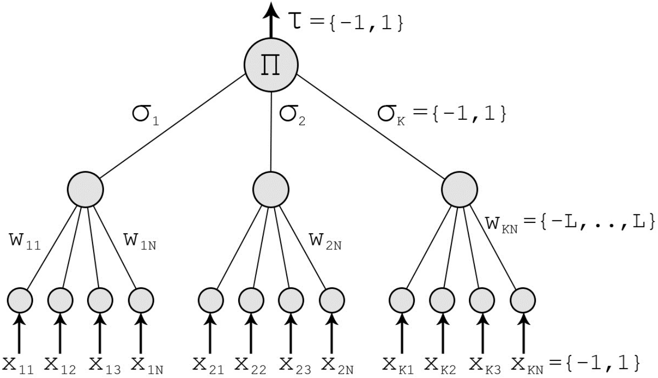
$$w_{ij,kl} = \eta[e_2(e_1 2^1 + e_0 2^0)]; \quad \begin{aligned} i, k &\in \{1, \ldots, L\}, \\ j &\in \{1, \ldots, n_i\}, \\ l &\in \{1, \ldots, n_k\}, \end{aligned}$$

where $w_{ij,kl} = w(x_{ij}, x_{kl})$ is the weight value between the j-th unit in the i-th layer and the l-th unit in the k-th layer and

$$\begin{aligned} \eta &= \text{learning parameter}; \quad \eta \in (0, 1) \\ e_0, e_1 &= \text{random digits} \\ e_2 &= \text{sign bit.} \end{aligned}$$

KKK Key Exchange Protocol

The first complete cryptosystem based on neural network is known in literature as KKK, from the surnames of its inventors [Kanter, Kinzel and Kanter(2001)].
This protocol is based on the synchronization of the weights of two tree parity machines, that represent the participants.



- The two NNs participating in the communication start from private key vectors $E_k(0)$ and $D_k(0)$. Mutual learning from the exchange of public information leads the two nets to develop a common, time dependent key: $E_k(t) = -D_k(t)$. This is then used for both encryption and decryption.
- At each step of the training process (and of encryption/decryption), a common public input vector is needed.
- Sender and recipient send their outputs to each other and in case they do not agree on them, weight are updated according to a Hebbian learning rule.
- As soon as the two NNs are synchronized, so they stay forever.

KKK: Shamir's Insights

In the paper cited before, no mathematical proof of its core principle, synchronization, is given. This result was instead attained in [Klimov, Mityagin and Shamir(2002)]. Such work also contains a section dedicated to the cryptanalysis of KKK. Besides it is robust against attacks based on intercepting the key using the same neural network structure, this protocol can be broken using

- Genetic Attacks;
- Geometric Attacks;
- Probabilistic Attacks.

These results marked the beginning of a long period without any substantial contribution to neural cryptography.

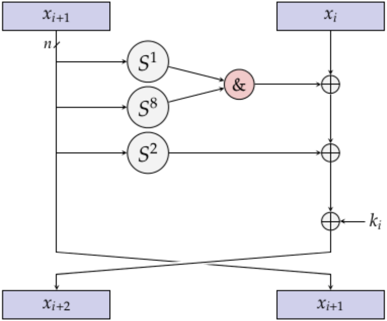
Neural Cryptanalysis

Not only NNs reveal themselves capable of learning how to communicate securely: recent studies show they can be employed to perform efficient cryptanalysis. Consider the family of block ciphers released by NSA in [Beaulieu et al.(2013)], namely

- **Simon**, a cipher efficient in hardware implementations in IoT (Internet of Things) devices;
- **Speck**, the sister algorithm of Simon, optimized for software implementations.

Both algorithms are compositions of the basic functions of modular addition, bitwise rotation and bitwise addition.

Simon



- The attack to Simon cipher consists in recovering the key, given a set of plaintext-ciphertext pairs.
- MLPs are employed; input layer has one neuron per each bit of the plaintext-ciphertext pairs. Output layer has same size of the key; each cell is binary.
- Activation function is chosen to be the Linear Rectifier:

$$y = \sum_j w_j x_j + b.$$

Speck

A more advanced attack was developed against Speck, by making use of convolutional NNs to build a (neural) distinguisher.

- Let $F : \{0,1\}^n \longrightarrow \{0,1\}^m$ be a map (round function). A differential transition for F is a pair

$$(\Delta_{\text{in}}, \Delta_{\text{out}}) \in \{0,1\}^n \times \{0,1\}^m$$

- A differential attack uses nonrandom properties of the output of the cipher when it is being given input data with known difference distribution, i. e. $\mathbb{P}(\Delta_{\text{in}} \rightarrow \Delta_{\text{out}})$.
- Nets are trained to distinguish the output of Speck with given input conditions from random data.

The End

References

- GRAUPE, D. (2007). *Principles of Artificial Neural Networks (2nd Edition)*. Word Scientific Publishing, Singapore.
- MCCULLOCH, W. and PITTS, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics 5, 115-133.
- HEBB, D. O. (1949). *The organization of behavior; a neuropsychological theory*. Wiley, New York.
- ROSENBLATT, F. (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65.
- WIDROW, B. and HOFF, M. E. (1960). *Adaptive Switching Circuits*. IRE WESCON Convention Record, 96-104.

- MInsky, M. and Papert, S. A. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.
- Rumelhart, D., Hinton, G. and Williams, R. (1986). *Learning representations by back-propagating errors*. Nature 323, 533–536.
- Sejnowski, T. J. and Rosenberg, C. R. (1987). *Parallel Networks that Learn to Pronounce English Text*. Complex Systems, 1.
- Kingma, D.P. and Lei Ba, J. (2015). *Adam: a method for stochastic optimization*. CoRR.
- Mitchell, M. (1998). *An introduction to Genetic Algorithms*. MIT Press.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. MIT Press.

- Lauria, F. E. (1990). *On Neurocryptology.* Proceedings of the Third Italian Workshop on Parallel Architectures and Neural Networks, 337-343.

- Volná, E. (2000). *Using Neural Network in Cryptography*. University of Ostrava.

- Volná, E. (1998).*Learning algorithm which learns both architectures and weights of feedforward neural networks.* Neural Network World. Int. Journal on Neural and Mass-Parallel Compo and Inf. Systems.

- Kanter, I., Kinzel, W. and Kanter, E. (2001). *Secure exchange of information by synchronization of neural networks*. Bar Ilan University.

- Klimov, A., Mityagin, A. and Shamir, A. (2002). *Analysis of Neural Cryptography*. Weizmann Institute.

- Abadi, M. and Andersen, D. G. (2016). *Learning to protect communications with Adversarial Neural Cryptography*.

- Coutinho, M., Robson de Oliveira Albuquerque, R., Borges, F. , Villalba, L. J. G. and Kim T. H. (2018). *Learning Perfectly Secure Cryptography to Protect Communications with Adversarial Neural Cryptography*. University of Brasília.

- Jayachandiran, K. (2018). *A Machine Learning Approach for Cryptanalysis*. Rochester Institute of Technology.

- Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B. and Wingers, L. (2013). *The Simon and Speck Families of Lightweight Block Ciphers*. National Security Agency.

- Alani, M. M. (2012). *Neuro-cryptanalysis of DES*. World Congress on Internet Security (WorldCIS-2012)

- Gohr, A. (2019). *Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning*. Bundesamt für Sicherheit in der Informationstechnik (BSI).