Hyper Heuristic Cryptography with Mixed Adversarial Nets

Author: Aly Shmahell
Supervisor: Prof. Giovanni De Gasperis

July 15, 2018

University of L'Aquila

Introduction

What This Thesis Is About

Neural Cryptography:

applying stochastic methods to get neural nets to achieve cryptographic functionality.

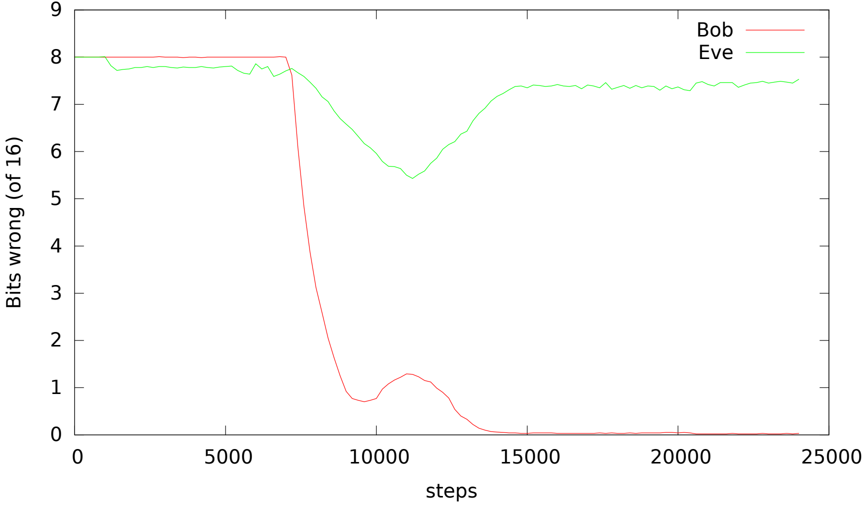
Basis For This Thesis:

a recent paper released in 2016 from Google Brain [1].

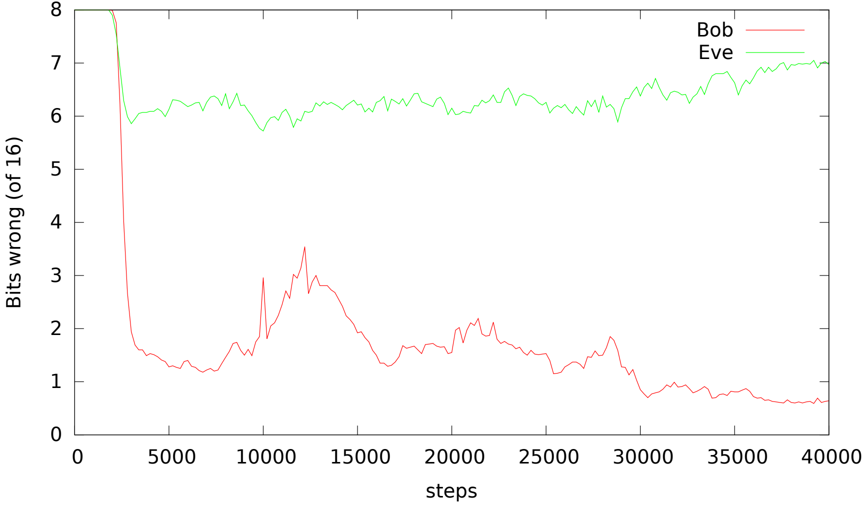
How The Thesis Extends Its Basis:

by focusing on increasing confidentiality of communication, while minimizing loss of information integrity.

Symmetric Results From Google Brain - For Comparison



Asymmetric Results From Google Brain - For Comparison



Justification For Neural Cryptography

Neural Cryptography Is Viable:

convolutional nets can construct local spatial relations in data.

Neural Cryptanalysis Is Viable:

fully connected layers can detect global spatial relations in data.

Neural Cryptography Can Be Fast:

convolutional nets share weights using their filters.

Neural Cryptography Is Evolved, Not Patched:

using adversary in training evolves weights which serves to tweak the cryptographic functionality.

What This Thesis Adds To The Research Pool

A Prototype Blueprint:

for a software-engineered neural crypto-system.

An Analysis Of How Neural Components Work:

when the objective is to achieve cryptographic functionality.

An Enhancement In The Neural Structures:

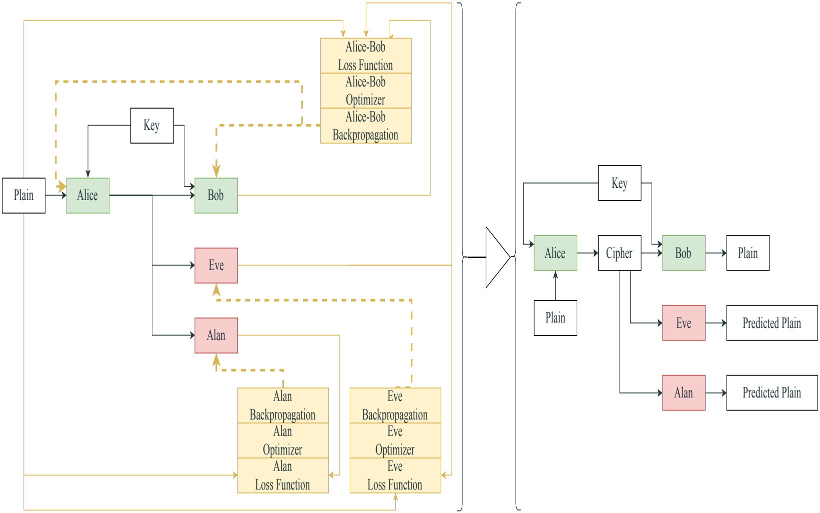
which yields a boost in cryptographic robustness.

Transfer Learning:

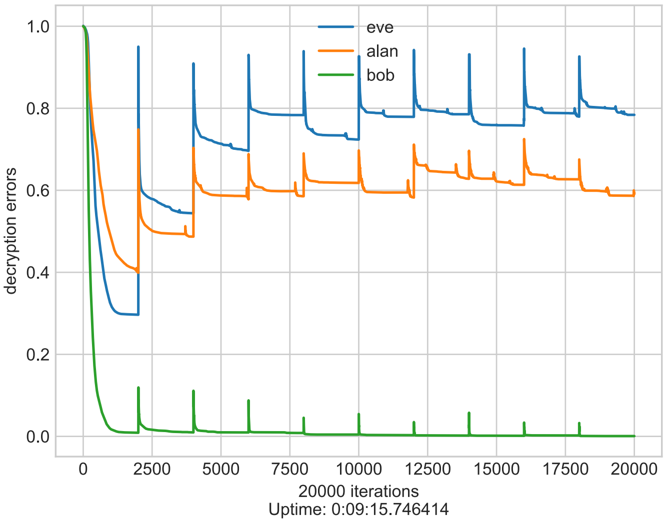
to get symmetric neural cryptography on par with asymmetric neural cryptography.

Experiments & Results

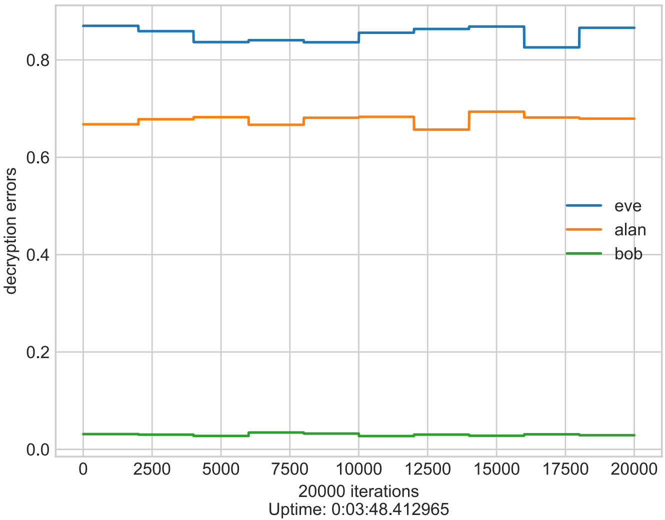
Symmetric Scheme



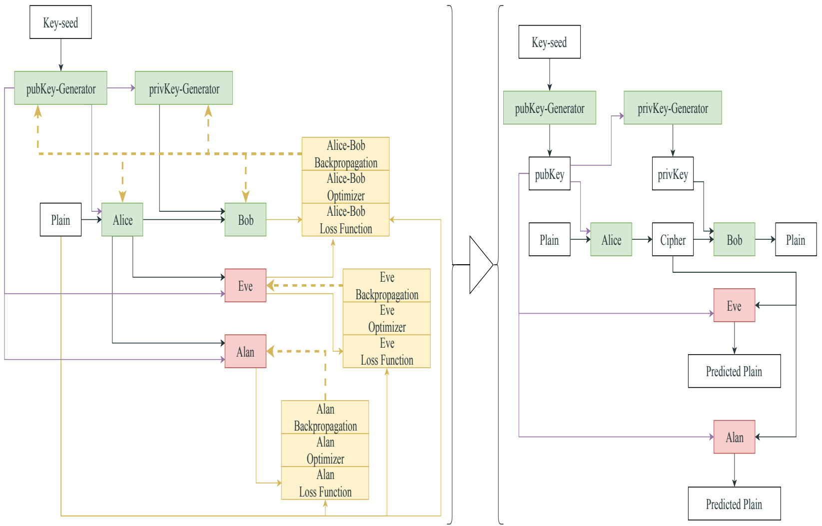
Thesis Results - Symmetric Training



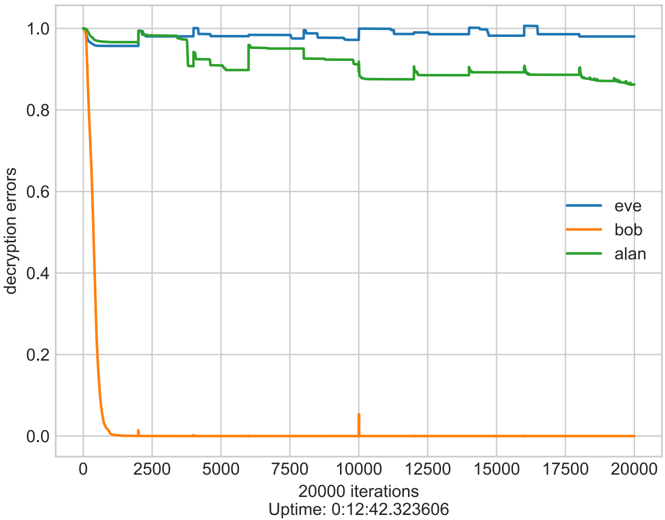
Thesis Results - Symmetric Testing



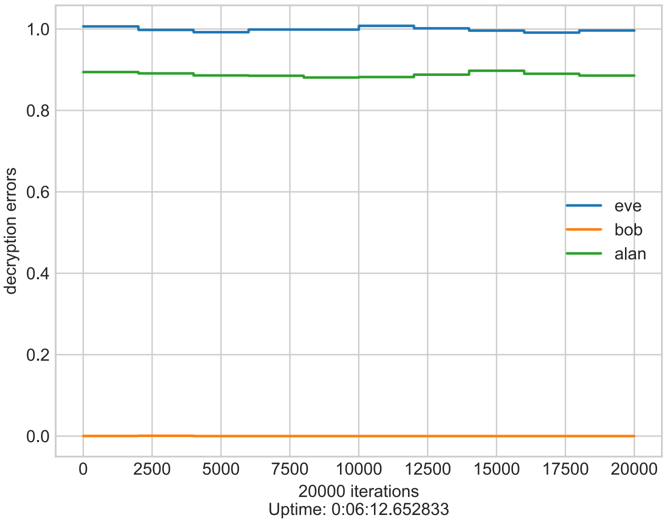
Asymmetric Scheme



Thesis Results - Asymmetric Training



Thesis Results - Asymmetric Testing

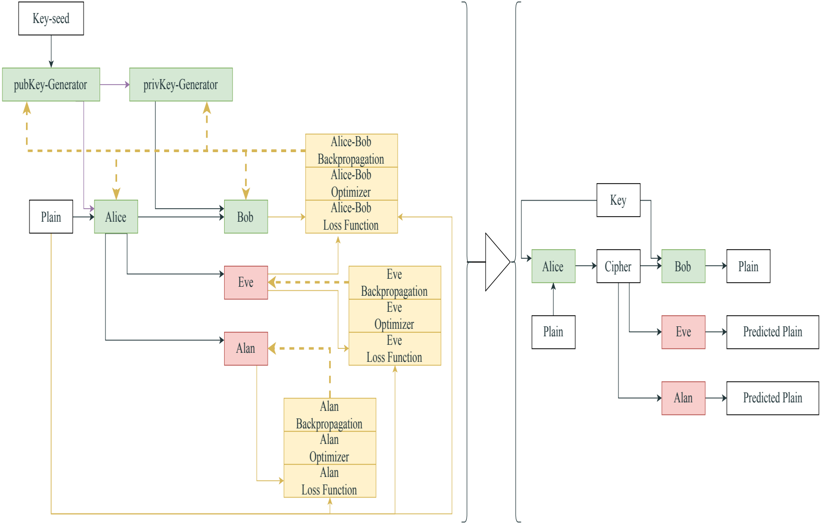


Transfer Learning

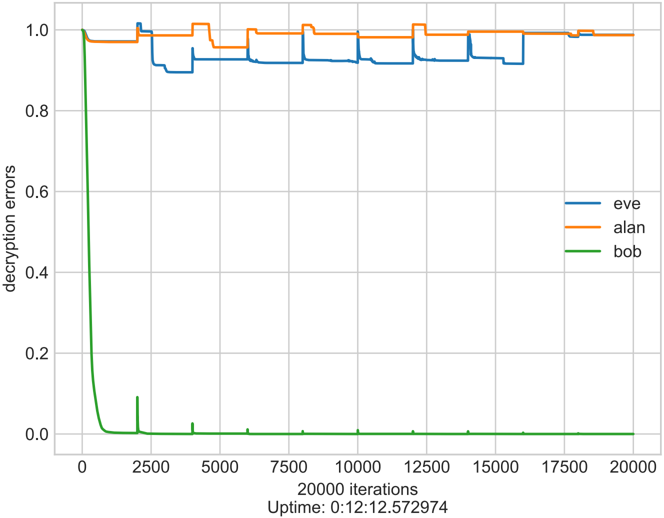
Transfer Learning:

Given a source domain D_S and learning task T_S , a target domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$, or $T_S \neq T_T$. [2]

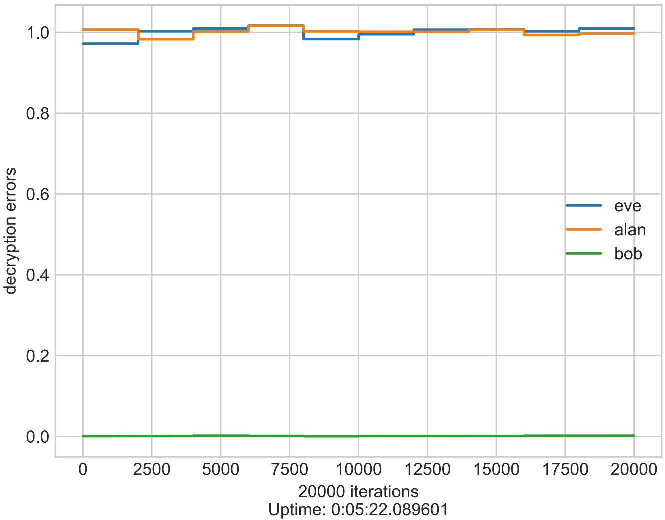
Hybrid Scheme



Thesis Results - Hybrid Training

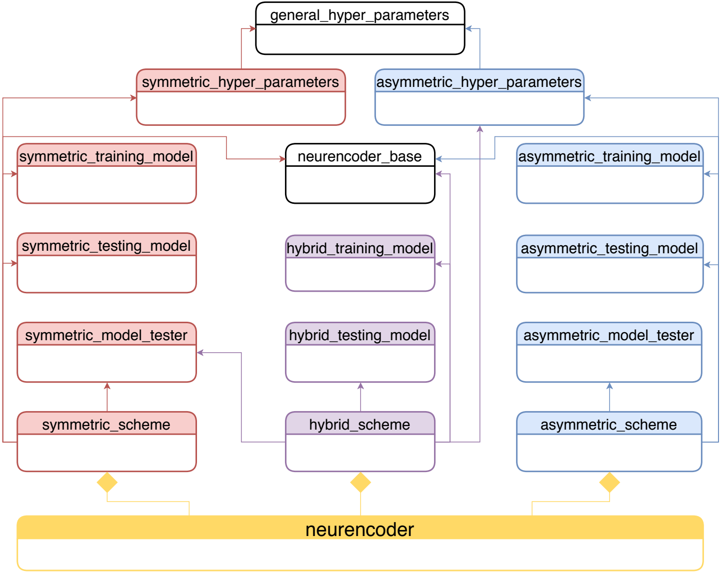


Thesis Results - Hybrid Testing



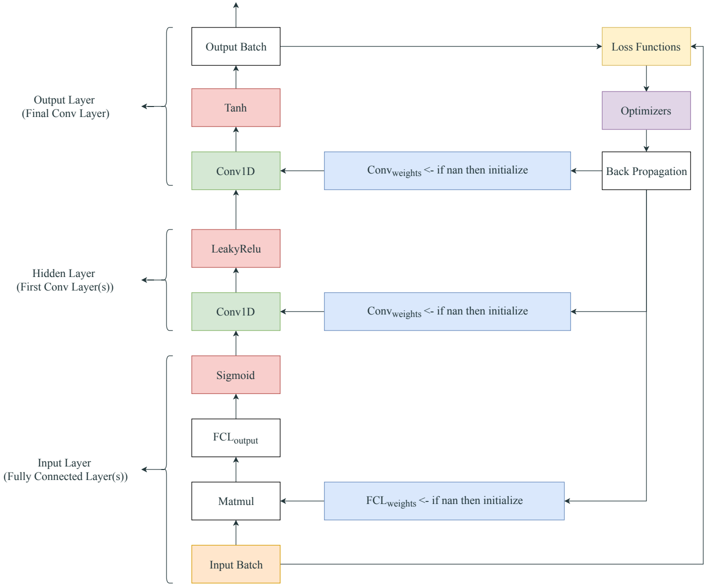
Implementation

Class Diagram



Appendix: How I Chose My Activation Functions

Dummy Net Example



Activation Function Combinations

Different Options:

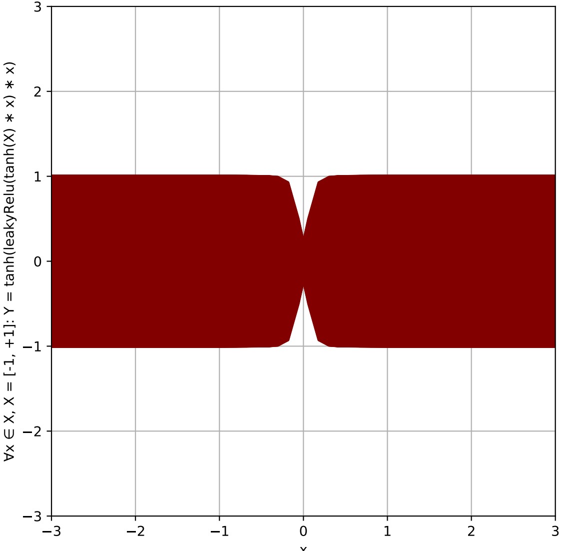
- $"Sigmoid \to LeakyRealu \to Sigmoid"$.
- $"Tanh \to LeakyRealu \to Tanh"$.
- $"Sigmoid \to LeakyRealu \to Tanh"$.

The Empirically Reliable Choice:

$$"Sigmoid \to LeakyRealu \to Tanh".$$

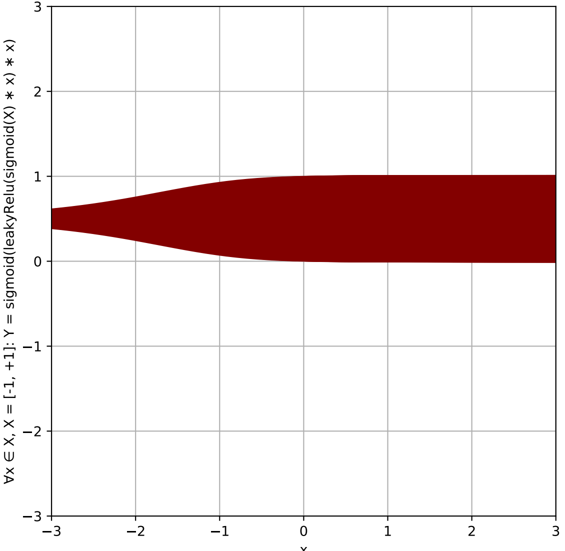
Activation Function Combinations - Numerical Analysis

"Tanh → LeakyRealu → Tanh"



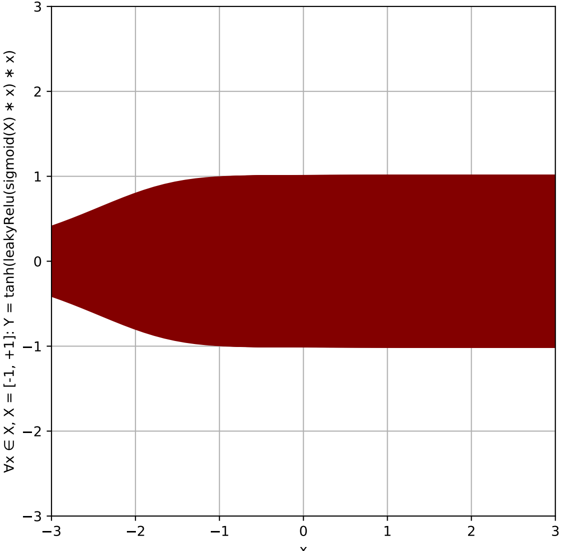
Activation Function Combinations - Numerical Analysis

"Sigmoid → LeakyRealu → Sigmoid".



Activation Function Combinations - Numerical Analysis

"Sigmoid $\rightarrow$ LeakyRealu $\rightarrow$ Tanh".



References

[1] Martín Abadi and David G. Andersen. Learning to protect communications with adversarial neural cryptography. *CoRR*, abs/1610.06918, 2016.

[2] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.