Adversarial Neural Cryptography

Artificial Intelligence 2018/2019

Lorenzo Veronese, 852058

Tuesday 04/02/2020

Università Ca' Foscari Venezia

Learning Symmetric Encryption

System Organization

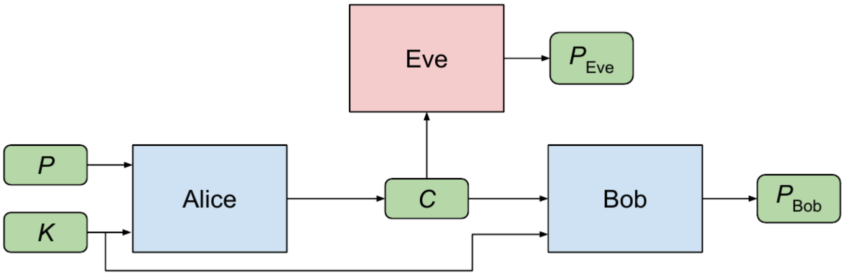


Figure 1: Alice, Bob, and Eve, with a symmetric cryptosystem.

Objectives and Loss Function

Eve	• Reconstruct P accurately (minimize $d(P_{eve}, P)$)
Alice / Bob	• Communicate clearly (minimize $d(P_{bob}, P)$) • Hide communication from Eve

- Differently from the common objectives of the adversaries of GANs, is not a goal for Eve to distinguish C from a random value drawn from some distribution.
- We want to train Alice and Bob jointly to communicate succesfully and defeat Eve without a pre-specified notion of what cryptosystem they might discover for this purpose.
- We want Alice and Bob to defeat the best possible version of Eve, rather than a fixed one.

Objectives and Loss Function

$$A(\theta_A, P, K) \qquad B(\theta_B, C, K) \qquad E(\theta_E, C)$$

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

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

- Alice and Bob want to minimize Bob's reconstruction error and to maximize the reconstruction error of the "optimal Eve".

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

Objectives and Loss Function

"In practice [...] our training method cuts a few corners and incorporates a few improvements with respect to the high-level description of the objectives."

- "optimal Eve" is approximated by alternating Eve and Alice and Bob training.
- In the training of Alice and Bob, we do not maximize Eve's error.
  - If we did, and made Eve completely wrong, then Eve could be completely right in the next iteration by simply flipping all output bits!
  - Generally, the goal is to minimize the mutual information between Eve's guess and the real plaintext.
    - Make Eve produce answers indistinguishable from a random guess.

$$L_{AB} = \text{Bob L1 error} + \frac{(N/2 - \text{Eve L1 error})^2}{(N/2)^2}$$

Objectives and Loss Function

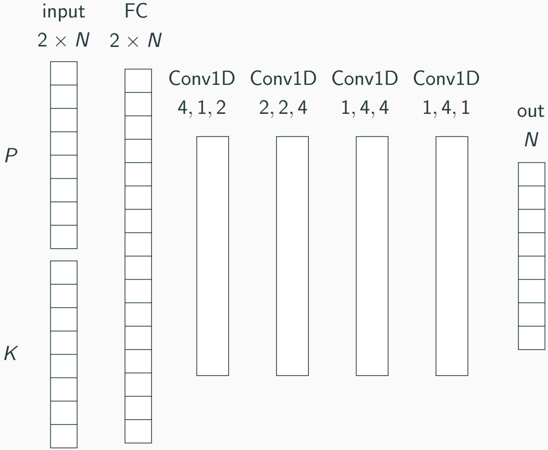
"In practice [. . .] our training method cuts a few corners and incorporates a few improvements with respect to the high-level description of the objectives."

- "optimal Eve" is approximated by alternating Eve and Alice and Bob training.
- In the training of Alice and Bob, we do not maximize Eve's error.
  - If we did, and made Eve completely wrong, then Eve could be completely right in the next iteration by simply flipping all output bits!
  - Generally, the goal is to minimize the mutual information between Eve's guess and the real plaintext.
    - Make Eve produce ans… guess.

This is minimized when half of the bits are wrong and half are right

$$L_{AB} = \text{Bob L1 error} + \frac{(N/2 - \text{Eve L1 error})^2}{(N/2)^2}$$

Network Architecture



- *"mix & transform"* architecture
- Architecture that is *succificient* to learn functions such as XOR
- FC layer enables mixing between P and K
- Conv layers lean functions on groups of bits
    - *learn* which bits to combine

Experimental Evaluation

N_{bits}	16
Mini-batch Size	4096
Training Rounds	1 A/B for every 2 of E

- A training is successful if the networks cross an accuracy threshold
  - 0.05 bits of reconstruction error for Alice ad Bob
  - 1-2 bits more than random guessing for Eve
- After successful training of the networks we reset Eve and train it from scratch 5 times, recording the best result achieved by any Eve.
  - If the retrained Eve obtains substantial advantage the solution is non-robust.

Effects of Different Loss Functions

"The best formulation [of the loss function] remains an open question"

- Define a per-bit loss function for Alice/Bob and Eve

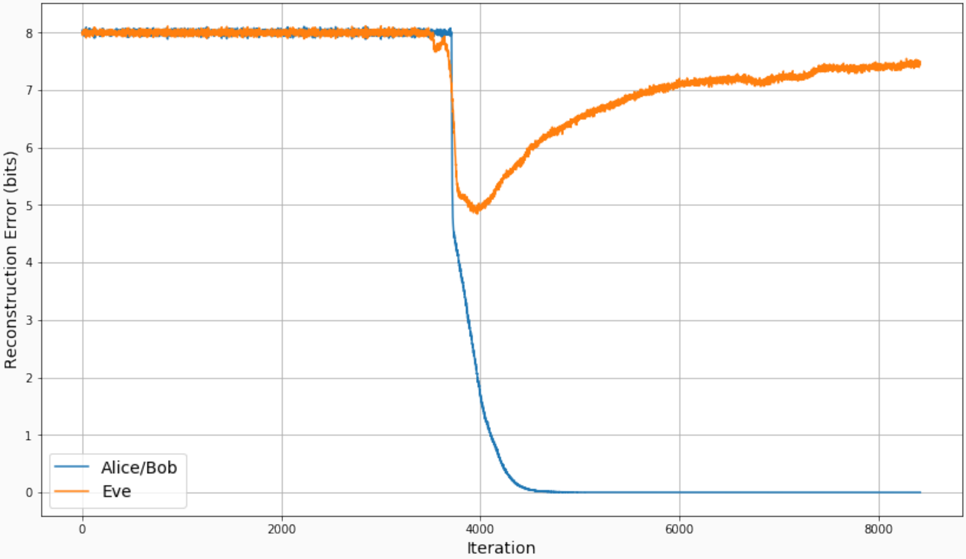
$$L_n = \frac{1}{N}\sum_{i}^{N} |P_{ni} - P_i| \qquad \boxed{0 \leq L_n \leq 2}$$

$$L_{AB} = L_B + (1 - L_E)^2$$

minimized when half of the bits are wrong

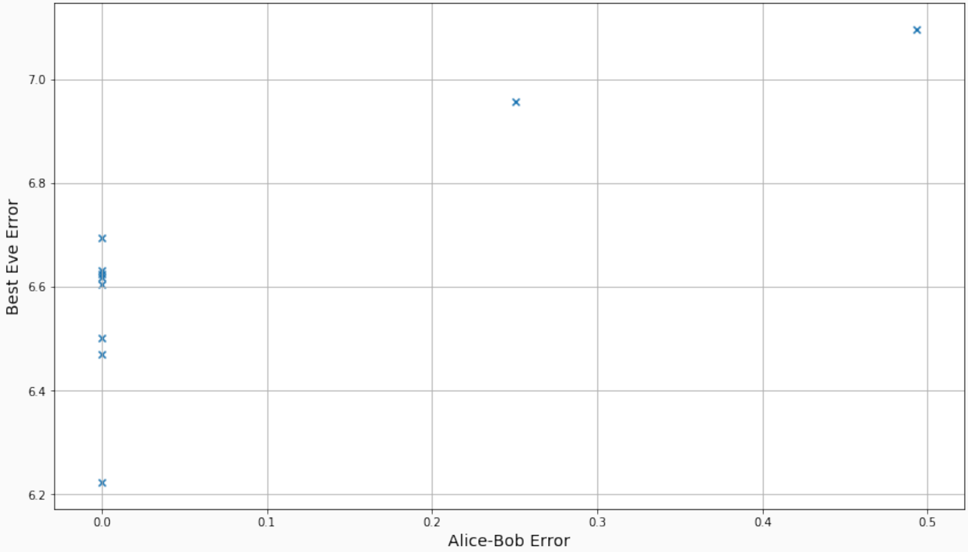
- In my experiments, the training is less unstable and the rate of convergence is improved.

Results

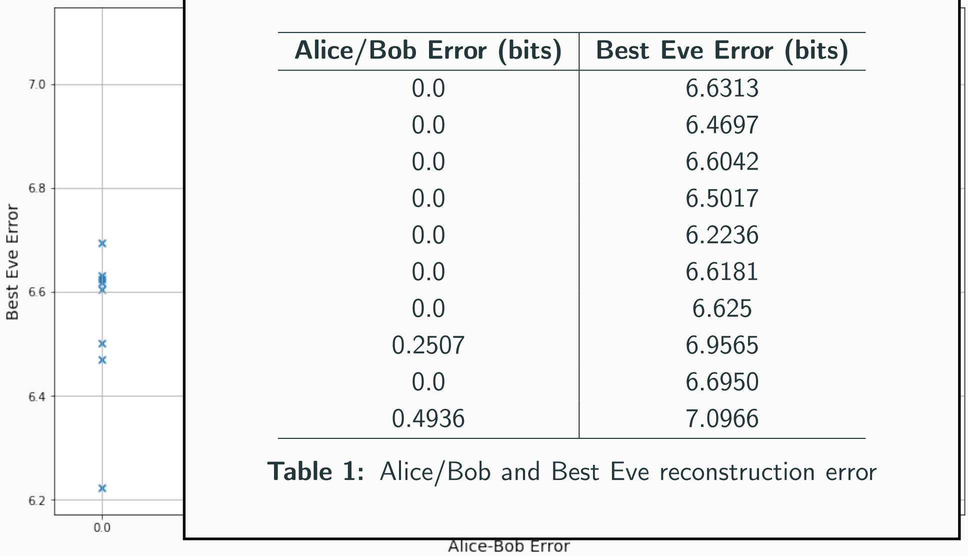


Reconstruction Error (bits)
Iteration
0
1
2
3
4
5
6
7
8
0
2000
4000
6000
8000
Alice/Bob
Eve

Results

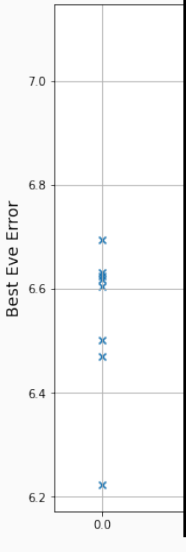


Results



Alice/Bob Error (bits)	Best Eve Error (bits)
0.0	6.6313
0.0	6.4697
0.0	6.6042
0.0	6.5017
0.0	6.2236
0.0	6.6181
0.0	6.625
0.2507	6.9565
0.0	6.6950
0.4936	7.0966

Table 1: Alice/Bob and Best Eve reconstruction error



Alice/Bob Error (bits)	Best Eve Error (bits)
0.0	6.6313
0.0	6.4697
0.0	6.6042
0.0	6.5017
0.0	6.2236
0.0	6.6181
0.	
0.	

Table 1: Ali

- The training was successful 8 out of 10 times
- The most effective retrained version of Eve did not perform better that 6.22/16 bits wrong

Notes on Neural "Encryption"

$$P = \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} K = \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} A(P,K) = \begin{pmatrix} -0.2108 \\ -0.4068 \\ 0.5591 \\ 0.5356 \\ -0.5933 \\ 0.5051 \\ 0.5608 \\ -0.4685 \\ 0.5518 \\ -0.3741 \\ 0.7064 \\ -0.6699 \\ 0.5502 \\ -0.6388 \\ 0.5095 \\ 0.6827 \end{pmatrix} B(A(P,K),K) = \begin{pmatrix} -0.9982 \\ 0.9986 \\ -0.9991 \\ -0.9953 \\ 0.9994 \\ -0.9992 \\ -0.9993 \\ -0.9986 \\ 0.9992 \\ 0.9995 \\ -0.9961 \\ -0.9981 \\ -0.9992 \\ 0.9990 \\ 0.9981 \\ -0.9973 \end{pmatrix} E(A(P,K)) = \begin{pmatrix} -0.9996 \\ -0.9998 \\ -0.9999 \\ 0.9998 \\ -0.9998 \\ -0.9998 \\ -0.9998 \\ 1.0000 \\ 0.9999 \\ -0.9998 \\ 1.0000 \\ -0.9956 \\ -0.9999 \\ -0.9998 \\ -0.9998 \\ 1.0000 \end{pmatrix}$$

Notes on Neural "Encryption"

- The ciphertext is plaintext and key dependent
- Changing a single bit of the key changes multiple outputs
- Outputs are floating point numbers, so the learned algorithm is not XOR but some mapping between the two spaces
- We are training against an adversary that is strictly less complex that A/B. Moreover A and B know which algorithm E is using.

Improving Eve (Eve++)

> What happens if you substantially increase the complexity of Eve [. . .]? There are several reasonable options for trying to make Eve stronger.[1]

- Eve++Layers has two additional convolutional layers.
- Eve++RandomKey has exactly the same shape and size as Bob, but receives random inputs instead of key material.

[1] https://openreview.net/forum?id=S1HEBe_Jl&noteId=rkyzxEDQe

Best Eve Performance

Training	Validation	Alice/Bob Error (bits)	Best Eve Error (bits)
Eve	Eve++Layers	0.0	6.6704
		0.0	6.6086
Eve++Layers	Eve++Layers	0.0000	6.5488
		0.0002	6.7205
Eve	Eve++RandomKey	0.0000	6.5842
		0.4819	6.8489
Eve++RandomKey	Eve++RandomKey	0.0000	6.2371
		0.0000	6.4241

Table 2: Alice/Bob and Best Eve loss and reconstruction error

Best Eve Performance

Training	Validation	Alice/Bob Error (bits)	Best Eve Error (bits)
Eve	Eve++Layers	0.0 0.0	6.6704 6.6086
Eve++Layers	Eve++Layers	0.0000	6.5488
Eve	Eve		
Eve++RandomKey	Eve		

Table 2: A

- The retrained more capable Eve is not more effective than the old version, reaching a 6.23/16 best error.
- It seems that there is no difference in training with the improved Eve
- The extra inputs given to Eve++Random did not give her substantial advantage

Asymmetric Encryption

System Organization

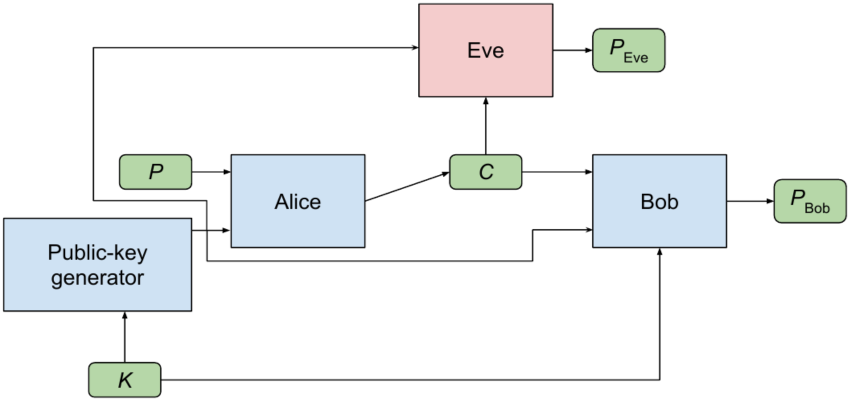
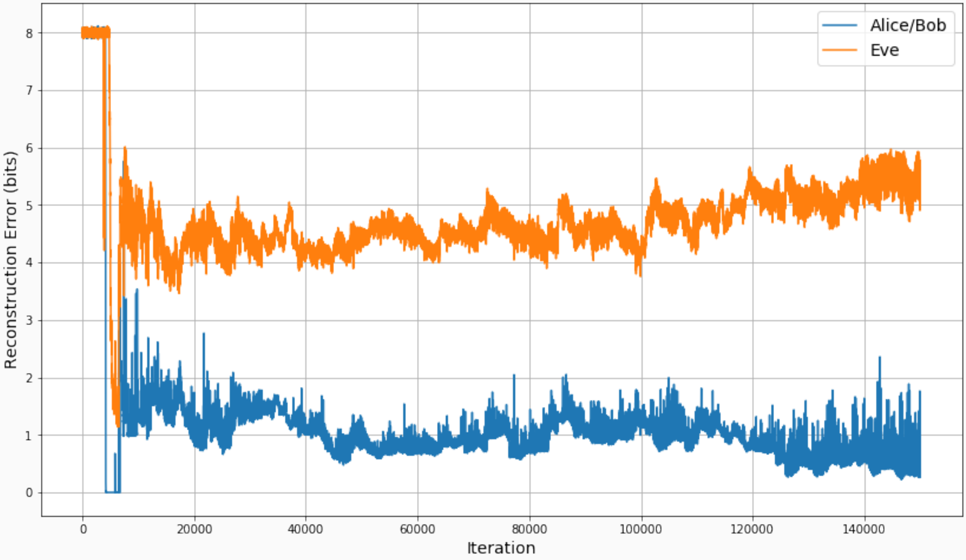


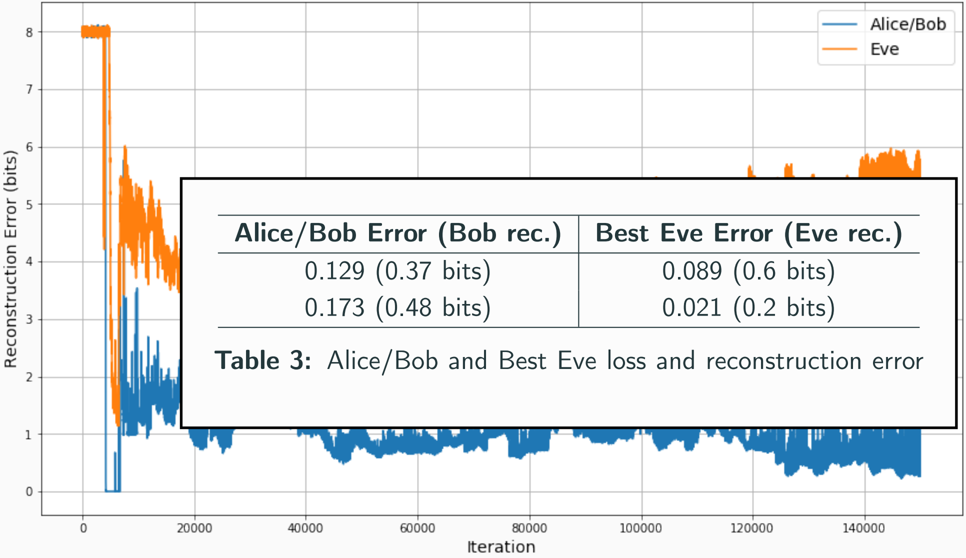
Figure 5: Alice, Bob, and Eve, with an asymmetric cryptosystem.

Results



Alice/Bob
Eve
Reconstruction Error (bits)
Iteration
0
1
2
3
4
5
6
7
8
0
20000
40000
60000
80000
100000
120000
140000

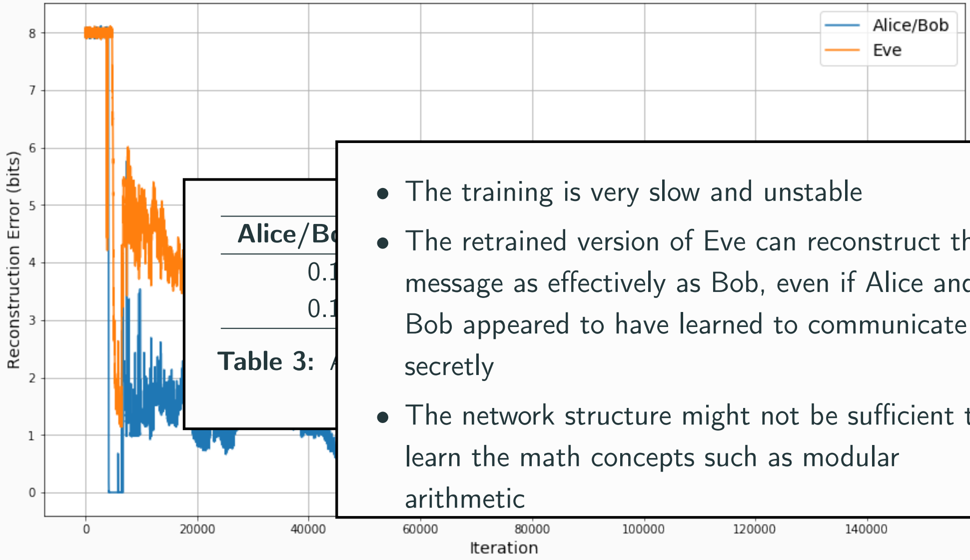
Results



Alice/Bob Error (Bob rec.)	Best Eve Error (Eve rec.)
0.129 (0.37 bits)	0.089 (0.6 bits)
0.173 (0.48 bits)	0.021 (0.2 bits)

Table 3: Alice/Bob and Best Eve loss and reconstruction error

Results



Alice/Bo
0.1
0.1

Table 3: A

- The training is very slow and unstable
- The retrained version of Eve can reconstruct the message as effectively as Bob, even if Alice and Bob appeared to have learned to communicate secretly
- The network structure might not be sufficient to learn the math concepts such as modular arithmetic

References

[AA16] Martín Abadi, David G. Andresen (Google Brain)
Learning to Protect Communications with Adversarial Neural Cryptography
https://arxiv.org/abs/1610.06918