PRE-PRESENTATION

Checking projector. . .

Indian Institute of Information Technology, Nagpur

भारतीय सूचना प्रौद्योगिकी संस्थान, नागपुर

Final Year Project

Mentor: Dr. Nidhi Lal

LEARNING TO PROTECT COMMUNICATIONS WITH ADVERSARIAL NEURAL CRYPTOGRAPHY

Shreyash H. Turkar

BT17CSE026

PROBLEM STATEMENT

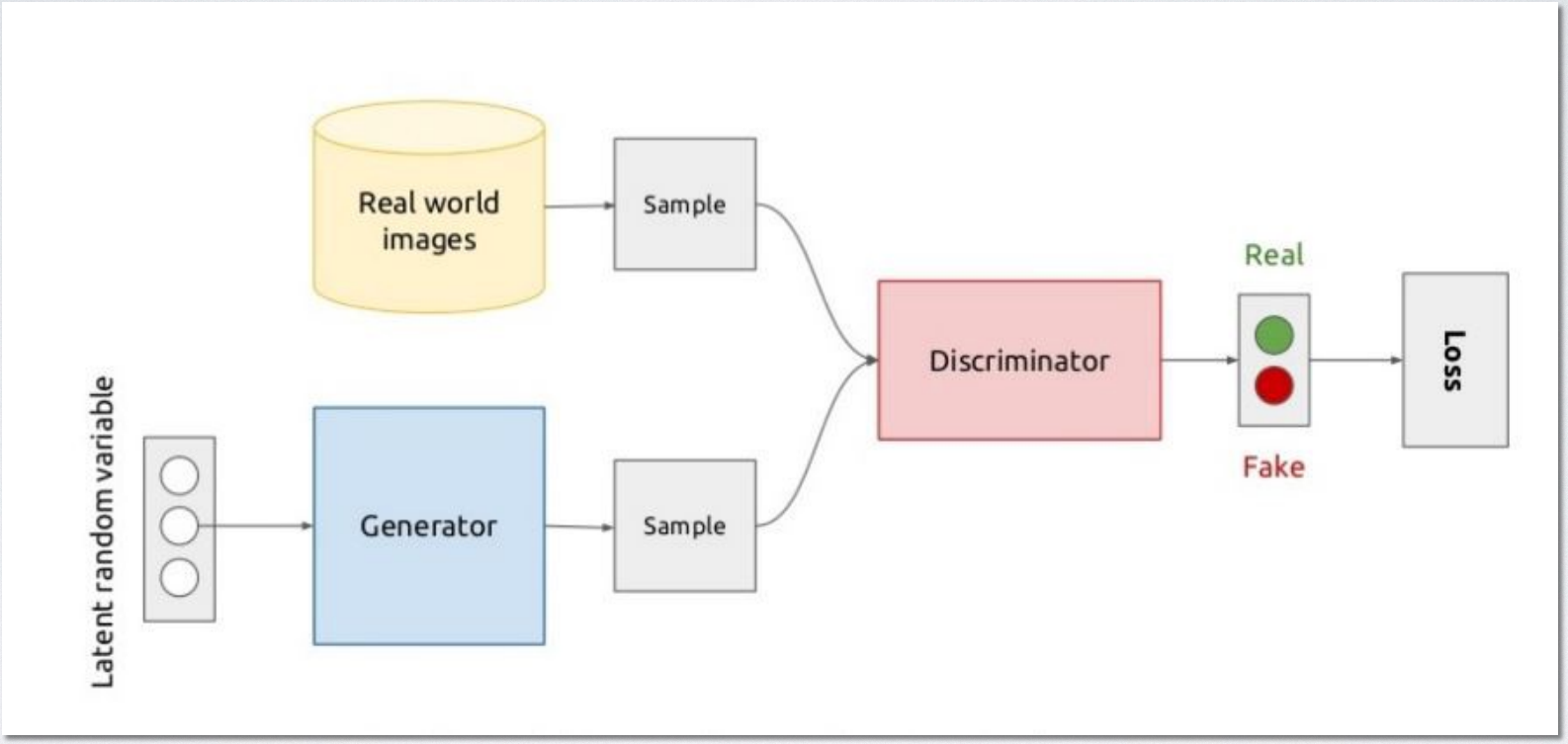
We here discuss whether we can employ neural nets to secure our communication channels. We take classic Alice, Bob and Eve example where Alice is trying to send a message to Bob and Eve is eavesdropping. Instead of using rigid symmetric encryption algorithms like AES, DES, we will create an adversarial neural network where Alice will be trained to encrypt using a shared key, Bob will be trained to decrypt using a shared key and Eve will be trained to reconstruct the message without a shared key.

SYMMETRIC ENCRYPTION

Symmetric crypto-system can be easily explained by a classic example of three people named Alice, Bob and Eve. Both Alice and Bob hold shared a secret key. Alice wants to send a secret message to Bob which she encrypts using their shared secret key to create cipher-text, Alice sends this cipher-text to Bob through a channel, presumably insecure. Bob utilises the same shared secret key to decrypt. When cipher-text was intercepted by Eve but she can't decrypt it without the shared secret key.

A classic scenario involving: Alice, Bob and Eve. Alice and Bob require to communicate securely, and Eve envies to snoop. We start with a particularly simple instance of this scenario, depicted in 1, in which Alice wishes to send a single confidential message P to Bob. The message P is input to Alice. Alice generates an output C. Bob and Eve take C to determine P. PBob and PEve, are what they compute respectively. Alice and Bob have a share a secret key K as a lead over Eve.

GANS


Real world images
Sample
Real
Discriminator
Loss
Latent random variable
Generator
Sample
Fake

DISCRIMINATOR GETS FOOLED

OR

GENERATOR WAS UNABLE TO FOOL

IF
Discriminator gets fooled

The discriminator is optimised by the loss to work better at distinguishing real vs fake samples

Else IF
Generator was unable to fool

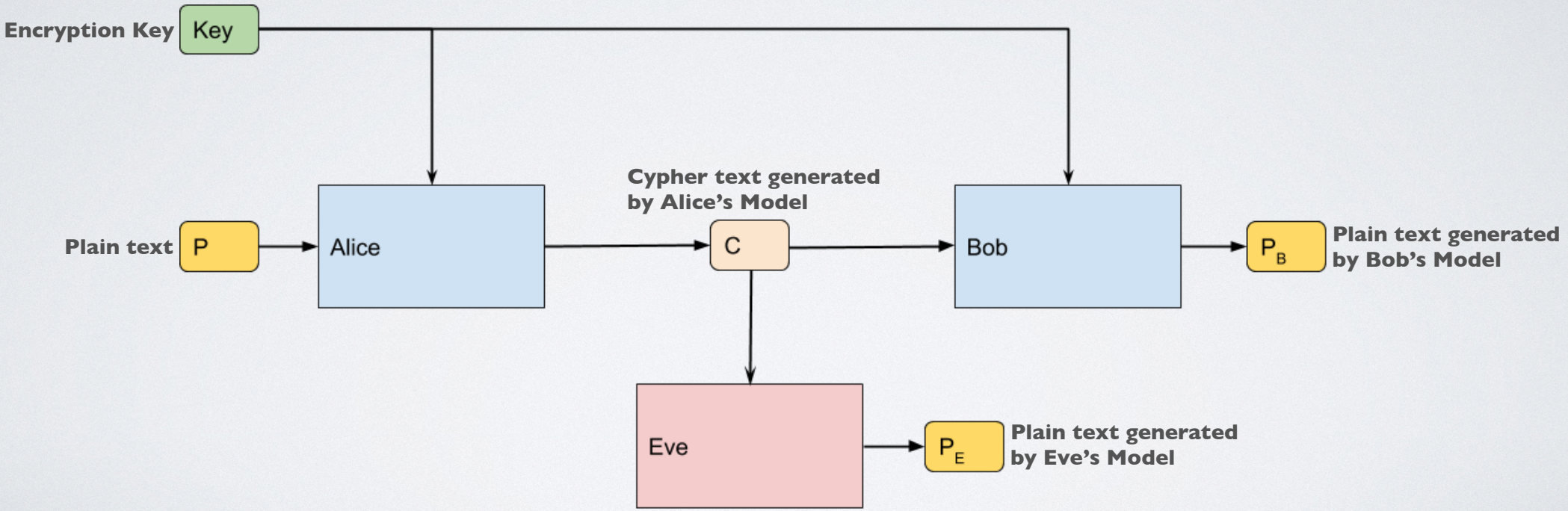
The generator is optimised by the loss to create better samples

INTEGRATING NEURAL NETWORK INTO CRYPTO- SYSTEM

We train three neural networks Alice, Bob and Eve whose jobs are as follows:

- Alice takes n-bit message + n-bit key and calculates n-bit cypher-text as an output
- Bob takes n-bit cypher-text created by Alice + n-bit key and computes original n-bit message as output.
- Eve takes n-bit cypher-text created by Alice and computes original n-bit message as output.

BLUEPRINT OF THE MODEL



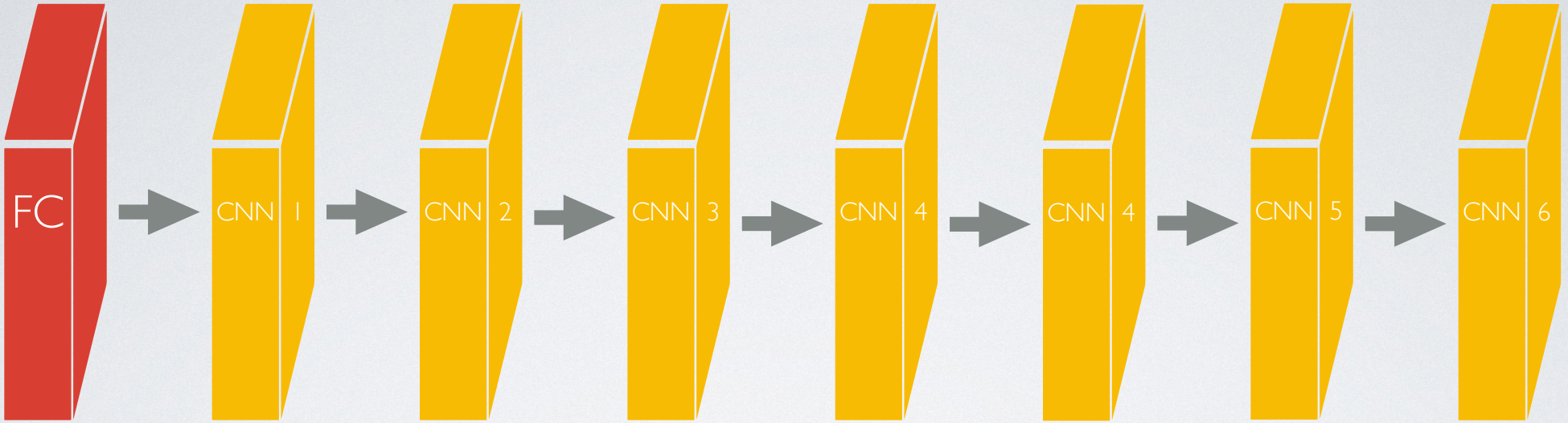
Rather than a rigid method, we chose the following architecture. It has a front fully-connected layer, where the number of outputs and inputs are equal. n-bits of plaintext and n-bits key are fed into this layer. (2n for Alice and Bob, n for Eve) This layer does not assure mixing between the key and the plaintext bits.

THE ARCHITECTURE OF MODEL

In final layer to bring the values back to range to map to binary we use tanh as activation function. Network of Alice and Bob are identical, whereas the network Eve network takes only the cipher-text as input explaining first $N \times 2N$ FC layer. We train at learning rate of .0008.

Input is processed by a dense layer using relu as an activation function. Following convolutional layers are described in terms of their window size, input depth, and output depth, stride—the amount by which the window is shifted and activation function as [4,1,1,1,leaky_relu], [2,1,2,2,leaky_relu], [1,2,1,1,leaky_relu], [1,1,4,1,sigmoid], [1,4,1,1,relu] and [1,1,1,1,tanh].

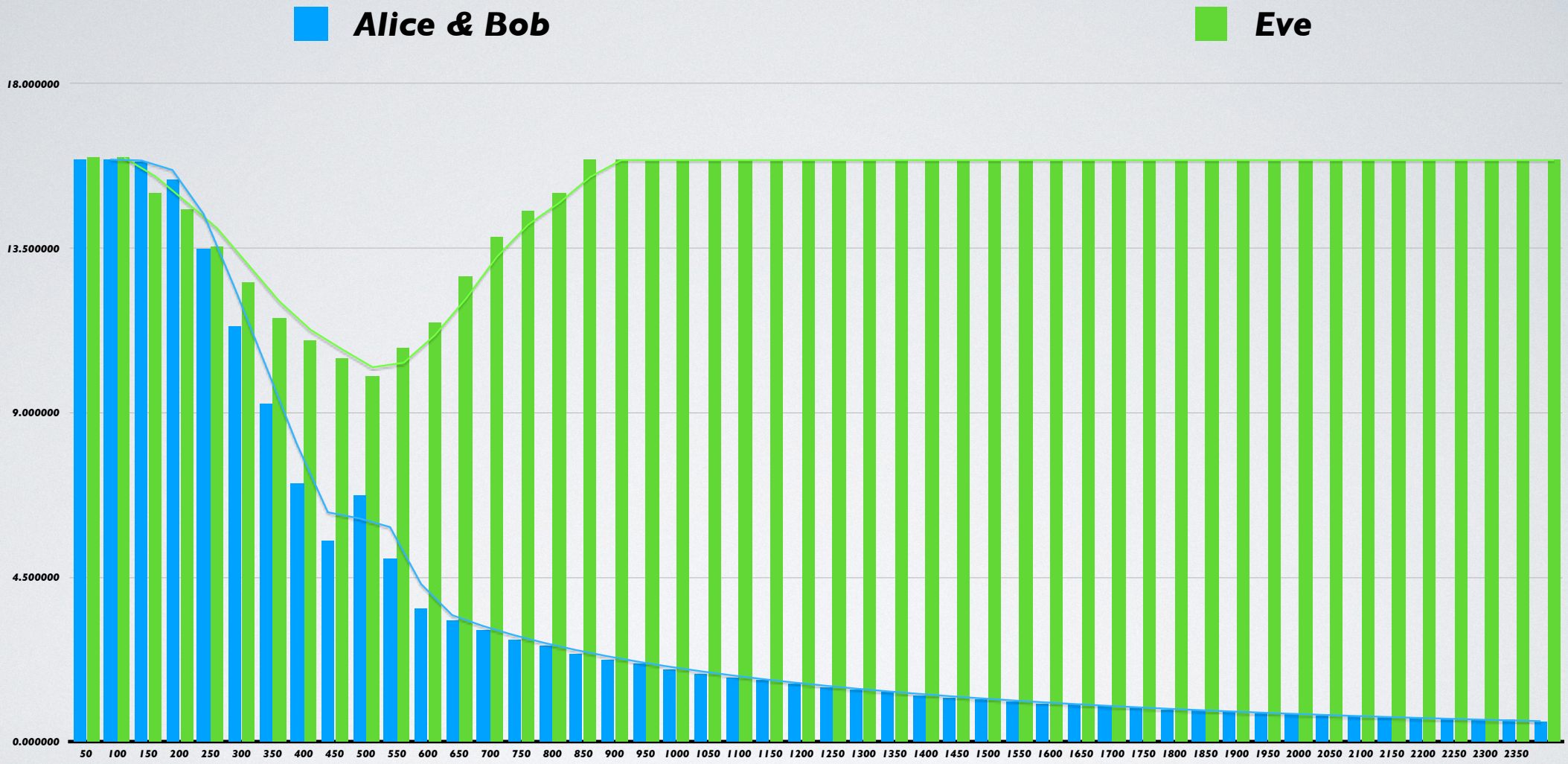
INPUT			Input	
			W:	32
			H:	1
			D:	1
			Parameters:	0
			Activation Size:	32
FC	Input		Output	
	W:	32	W:	32
	H:	1	H	0
	D:	1		1
			Parameters:	64
			Activation Size:	32
CNN 1	Input		Output	
	W:	32	W:	32
	H:	0	H:	0
	D:	1	D:	4
	(Filters) K:	4	Parameters:	8
	(Filter Dimensions) F:	1	Activation Size:	0
	(Stride) S:	1		
	(Padding) P:	0		
CNN 2	Input		Output	
	W:	32	W:	16
	H:	0	H:	0
	D:	4	D:	2
	(Filters) K:	2	Parameters:	34
	(Filter Dimensions) F:	2	Activation Size:	0
	(Stride) S:	2		
	(Padding) P:	0		
CNN 3	Input		Output	
	W:	16	W:	16
	H:	0	H:	0
	D:	2	D:	1
	(Filters) K:	1	Parameters:	3
	(Filter Dimensions) F:	1	Activation Size:	0
	(Stride) S:	1		
	(Padding) P:	0		
CNN 4	Input		Output	
	W:	16	W:	16
	H:	0	H:	0
	D:	1	D:	4
	(Filters) K:	4	Parameters:	8
	(Filter Dimensions) F:	1	Activation Size:	0
	(Stride) S:	1		
	(Padding) P:	0		
CNN 5	Input		Output	
	W:	16	W:	16
	H:	0	H:	0
	D:	4	D:	1
	(Filters) K:	1	Parameters:	5
	(Filter Dimensions) F:	1	Activation Size:	0
	(Stride) S:	1		
	(Padding) P:	0		
CNN 6	Input		Output	
	W:	16	W:	16
	H:	0	H:	0
	D:	1	D:	1
	(Filters) K:	1	Parameters:	2
	(Filter Dimensions) F:	1	Activation Size:	0
	(Stride) S:	1		
	(Padding) P:	0		



32x1						
W:	32	32	16	16	16	16	16
D:	1	4	2	1	1	4	1
Kernel Size:	4	2	1	4	4	1	1
(Filters) F:	1	2	1	1	1	1	1
(Stride) S:	1	2	1	1	1	1	1
(Padding) P:	0	0	0	0	0	0	0

RESULTS

The figure shows Eve's, Bob's reconstruction error vs the number of training steps for 16-bits. The initial error rate of Eve and Bob are 16-bits and start reducing. Around 350 Eve's error rate is reduced to 8-bits and after 360 it starts increasing and around 800 it remains constant at 16-bits and the error rate of Bob is constant at 1 after 2150 and after 2300 is decreased to 0.


Alice & Bob
Eve
18.000000
13.500000
9.000000
4.500000
0.000000
50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 1450 1500 1550 1600 1650 1700 1750 1800 1850 1900 1950 2000 2050 2100 2150 2200 2250 2300 2350

REFERENCES

- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp. 2672–2680, 2014a. http://papers.nips.cc/paper/5423-generative-adversarial-nets

- Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Info-GAN: Interpretable representation learning by information maximizing generative adversarial nets. CoRR, abs/1606.03657, 2016. URL https://arxiv.org/abs/1606.03657

- Pengtao Xie, Misha Bilenko, Tom Finley, Ran Gilad-Bachrach, Kristin E. Lauter, and Michael Naehrig. Crypto-nets: Neural networks over encrypted data. CoRR, abs/1412.6181, 2014. URL http://arxiv.org/abs/1412.6181

- Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. CoRR, abs/1605.07736, 2016. URL http://arxiv.org/abs/1605.07736

- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. CoRR, abs/1606.03498, 2016. URL https://arxiv.org/abs/1606.03498

- Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24,2016, volume 48 of JMLR Workshop and Conference Proceedings, pp. 201–210. JMLR.org, 2016. URL http://jmlr.org/proceedings/papers/v48/gilad-bachrach16.html

- Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. CoRR, abs/1602.02672, 2016a. URL http://arxiv.org/abs/1602.02672.

- Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. CoRR, abs/1605.06676, 2016b. URL http://arxiv.org/abs/1605.06676.

- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franc¸ois Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. CoRR, abs/1505.07818, 2015. URL http://arxiv.org/abs/1505.07818.

- Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, volume 48 of JMLR Workshop and Conference Proceedings, pp. 201–210. JMLR.org, 2016. URL http://jmlr.org/proceedings/papers/v48/gilad-bachrach16.html.

- Shafi Goldwasser and Silvio Micali. Probabilistic encryption. J. Comput. Syst. Sci., 28(2):270– 299, 1984. doi: 10.1016/0022-0000(84)90070-9. URL http://dx.doi.org/10.1016/ 0022-0000(84)90070-9.

- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Process- ing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp. 2672–2680, 2014a. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.

BY

SHREYASH H. TURKAR

BT17CSE026