Energy-Based GANs (EBGANs)

Definition, case study and results

Achraf Guenounou

Lab Neural Network

14-02-2025

References

Paper referenced for this presentation is “*ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS*” by Junbo Zhao, Michael Mathieu and Yann LeCun

(Check this for the math)

Article referenced for this presentation is “Review — EBGAN: Energy-Based Generative Adversarial Network (GAN)” by Sik-Ho Tsang

(Check this for results)

(click on the violet text to be redirected to the reference)

Published as a conference paper at ICLR 2017

ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS

Junbo Zhao, Michael Mathieu and Yann LeCun
Department of Computer Science, New York University
Facebook Artificial Intelligence Research
{jakezhao, mathieu, yann}@cs.nyu.edu

ABSTRACT

We introduce the “Energy-based Generative Adversarial Network” model (EBGAN) which views the discriminator as an energy function that attributes low energies to the regions near the data manifold and higher energies to other regions. Similar to the probabilistic GANs, a generator is seen as being trained to produce contrastive samples with minimal energies, while the discriminator is trained to assign high energies to these generated samples. Viewing the discriminator as an energy function allows to use a wide variety of architectures and loss functionals in addition to the usual binary classifier with logistic output. Among them, we show one instantiation of EBGAN framework as using an auto-encoder architecture, with the energy being the reconstruction error, in place of the discriminator. We show that this form of EBGAN exhibits more stable behavior than regular GANs during training. We also show that a single-scale architecture can be trained to generate high-resolution images.

1 INTRODUCTION

1.1 ENERGY-BASED MODEL

The essence of the energy-based model (LeCun et al., 2006) is to build a function that maps each point of an input space to a single scalar, which is called “energy”. The learning phase is a data-driven process that shapes the energy surface in such a way that the desired configurations get assigned low energies, while the incorrect ones are given high energies. Supervised learning falls into this framework: for each X in the training set, the energy of the pair (X, Y) takes low values when Y is the correct label and higher values for incorrect Y's. Similarly, when modeling X alone within an unsupervised learning setting, lower energy is attributed to the data manifold. The term *contrastive sample* is often used to refer to a data point causing an energy pull-up, such as the incorrect Y's in supervised learning and points from low data density regions in unsupervised learning.

1.2 GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) have led to significant improvements in image generation (Denton et al., 2015; Radford et al., 2015; Im et al., 2016; Salimans et al., 2016), video prediction (Mathieu et al., 2015) and a number of other domains. The basic idea of GAN is to simultaneously train a discriminator and a generator. The discriminator is trained to distinguish *real* samples of a dataset from *fake* samples produced by the generator. The generator uses input from an easy-to-sample random source, and is trained to produce fake samples that the

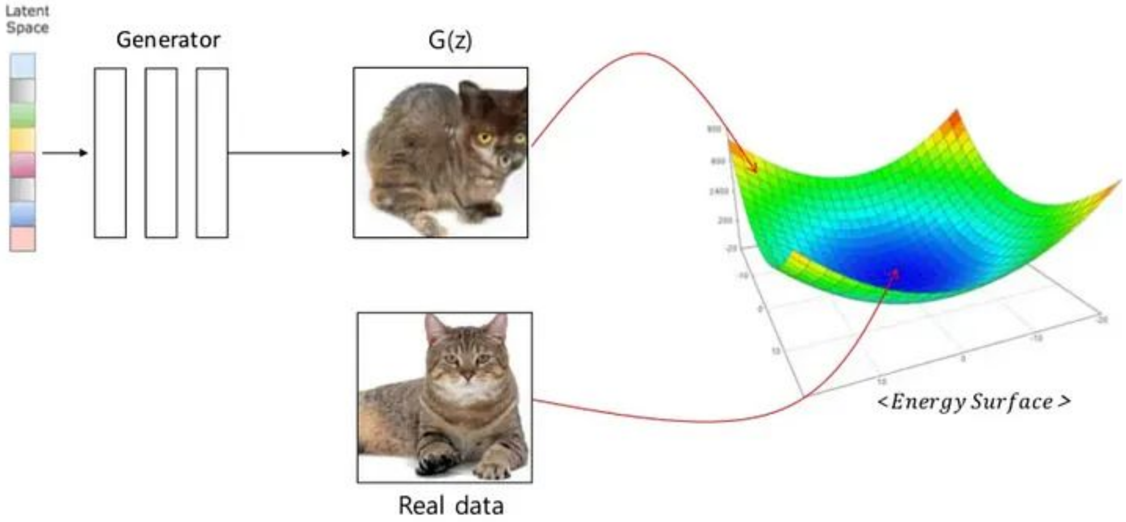
Energy-Based GANs

Definition, Components & Training Process

What is an EBGAN?

Definition

An Energy-Based Generative Adversarial Network (EBGAN) is a GAN variant where the discriminator acts as an energy function, evaluating images based on their reconstruction error



Components of EBGAN

Generator (G)

Produces fake images, aiming to reduce their energy score by making them more realistic

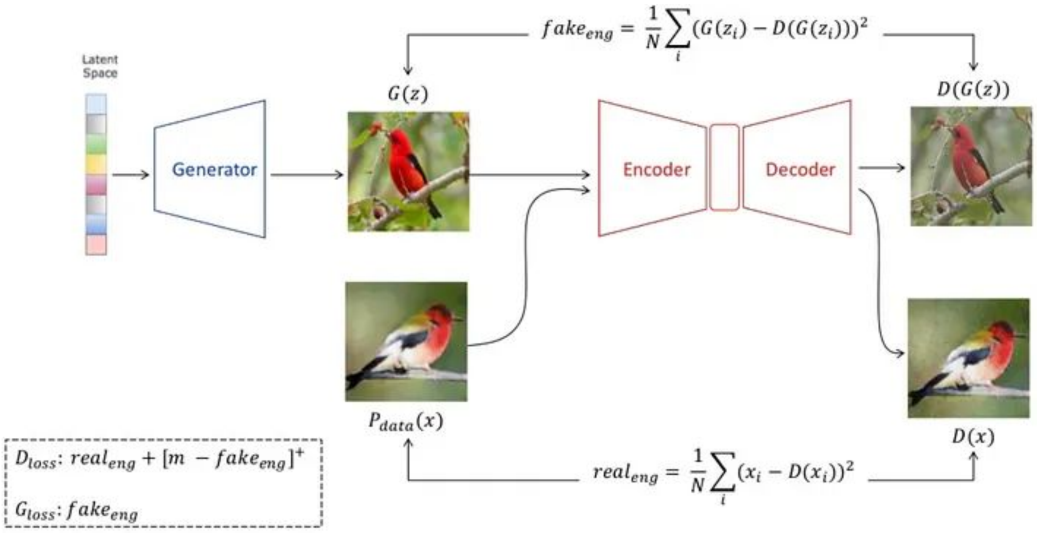
Discriminator (D)

Works as an **autoencoder** that assigns low energy to real images and high energy to fake ones, focusing on how well they can be reconstructed

Training process

How it works

1. Discriminator assigns low energy to real images and high energy to fake ones
2. Generator learns to produce images that result in lower reconstruction loss (fooling the discriminator)
3. Over time, the generator produces realistic images with low energy



In this type of networks **mode collapse** may occur → Use a **Pulling-away Term Regularization**

EBGAN-PT

Pulling-away Term Regularization

Problem:

Autoencoders might learn an identity function, assigning zero energy to everything

Solution

Introduce a repelling regularizer (Pulling-away Term - PT) to prevent clustering in few modes

How it works:

1. It measures the cosine similarity among all generated images features S in a minibatch
2. It will add a high penalty if there are too similar

```
Epoch 1, D Loss: 0.0041, G Loss: 0.0270
Epoch 2, D Loss: 0.0231, G Loss: 0.0022
Epoch 3, D Loss: 0.0191, G Loss: 0.0008
Epoch 4, D Loss: 0.2639, G Loss: 0.0001
Epoch 5, D Loss: 0.2627, G Loss: 0.0000
Epoch 6, D Loss: 0.2677, G Loss: 0.0001
Epoch 7, D Loss: 0.3596, G Loss: -0.0000
Epoch 8, D Loss: 0.3423, G Loss: -0.0000
Epoch 9, D Loss: 0.3653, G Loss: -0.0000
Epoch 10, D Loss: 0.3284, G Loss: 0.0001
```

Example of mode collapse that occurred during the training

$$f_{PT}(S) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \left(\frac{S_i^{\mathsf{T}} S_j}{\|S_i\| \|S_j\|} \right)^2 .$$

Benefits and limitations

Benefits	Limitations
More stable training than traditional GANs	Computationally Expensive
Suited for semi-supervised learning	Can suffer from mode collapse
Energy-Based Loss	Complex structure

EBGAN on CIFAR-10 Dataset

Test and results

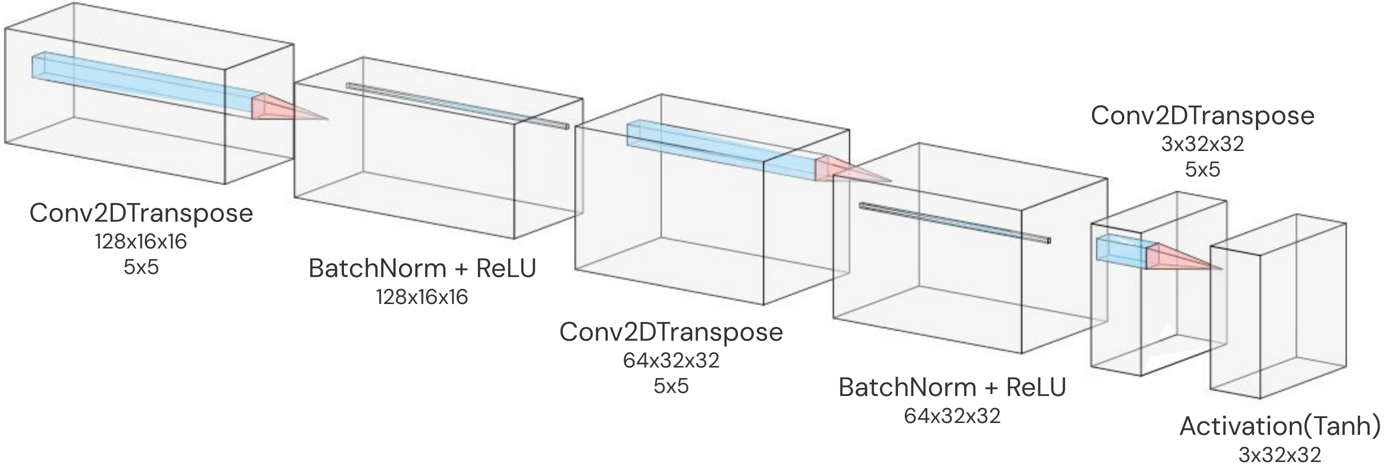
Model Architecture

Generator

Layer	Depth (Channels)	Height	Width	Filter Height	Filter Width
Input (Random Noise)	100	-	-	-	-
Input (Class Info)	10	-	-	-	-
Dense Layer	8192 (reshaped)	8	8	-	-
Reshape	128	8	8	-	-
Conv2DTranspose	128	16	16	5	5
BatchNorm + ReLU	128	16	16	-	-
Conv2DTranspose	64	32	32	5	5
BatchNorm + ReLU	64	32	32	-	-
Conv2DTranspose (Output)	3 (RGB)	32	32	5	5
Activation (Tanh)	3 (RGB)	32	32	-	-

Model Architecture

Generator



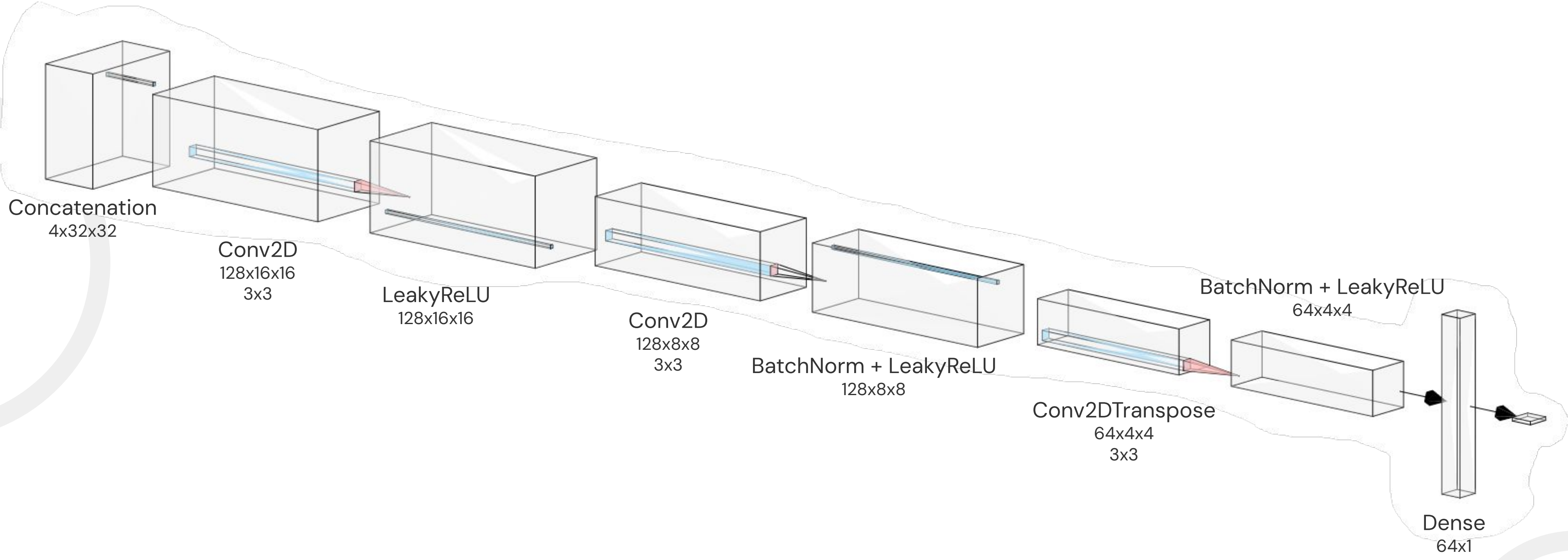
Model Architecture

Discriminator

Layer	Depth (Channels)	Height	Width	Filter Height	Filter Width
Input (Image)	3 (RGB)	32	32	-	-
Input (Class Info)	10	-	-	-	-
Dense Layer (Reshape)	1	32	32	-	-
Concatenation	4	32	32	-	-
Conv2D	128	16	16	3	3
LeakyReLU	128	16	16	-	-
Conv2D	128	8	8	3	3
BatchNorm + LeakyReLU	128	8	8	-	-
Conv2D	64	4	4	3	3
BatchNorm + LeakyReLU	64	4	4	-	-
Flatten	1D	-	-	-	-
Dense	64	-	-	-	-
Dense (Output)	1 (Real/Fake)	-	-	-	-

Model Architecture

Discriminator



Model Parameters

Parameter	Value
Optimization Algorithm	Adam
Learning Rate	0.0001
Beta_1	0.5
Epochs	200
Batch Size	256
Noise Vector Size	100
Discriminator Loss	Energy-Based Loss
Generator Loss	Minimize Energy Score

Training Data
CIFAR-10 Dataset

Dataset Overview:

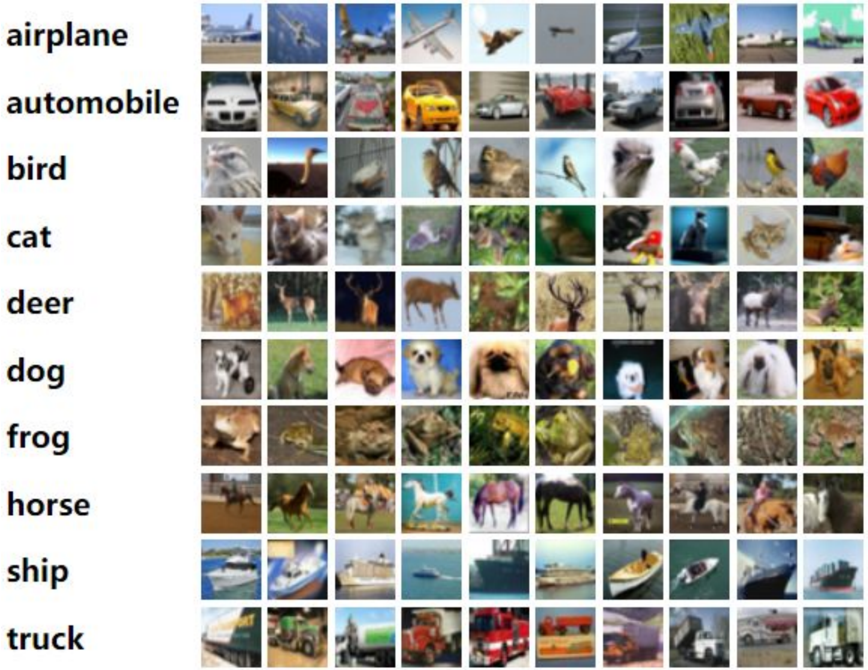
- 60,000 images (32×32×3),
- 10 classes

Splitting

- **Training:** 50,000 images (80%)
- **Testing:** 10,000 images (20%)

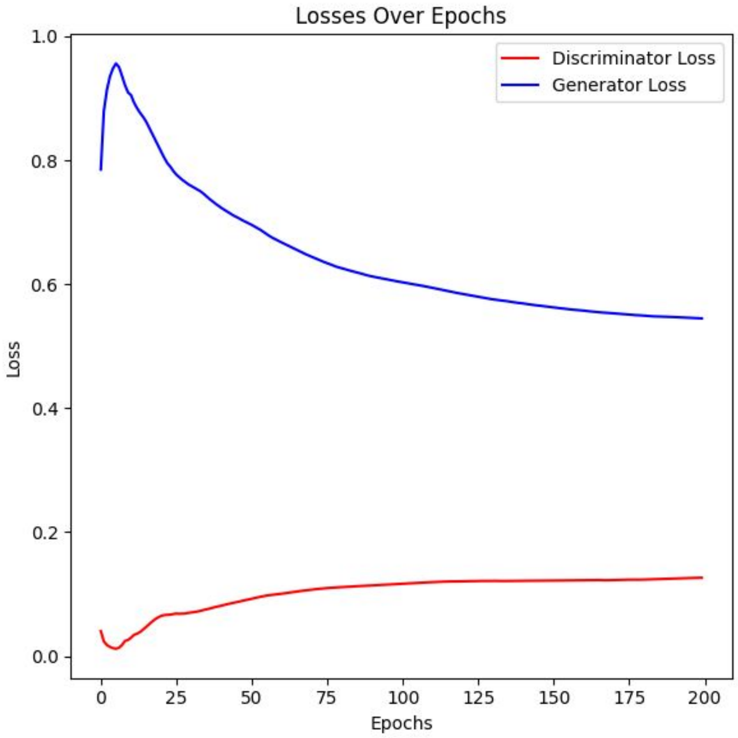
Preprocessing:

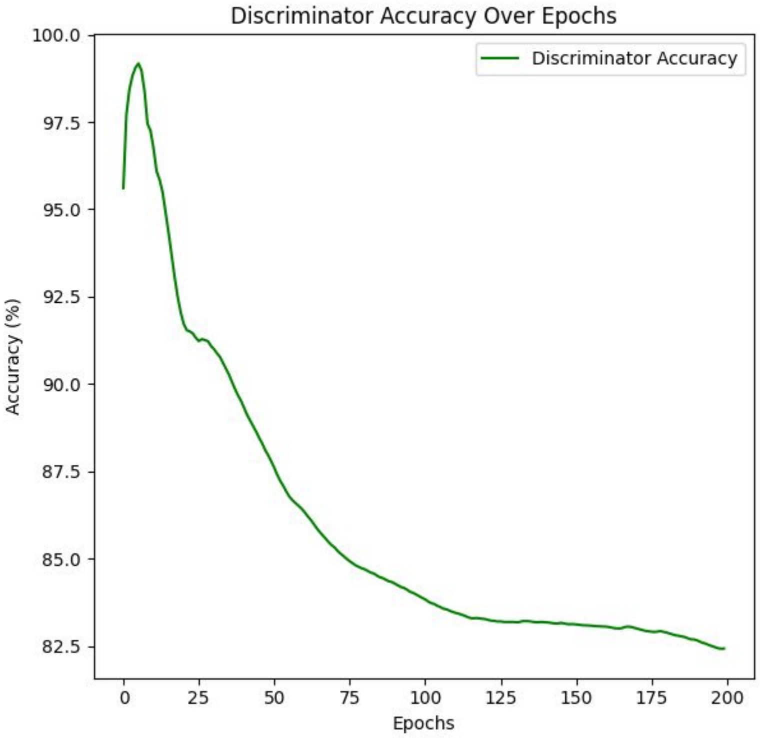
- Normalization
- Reshaping



Model Performance

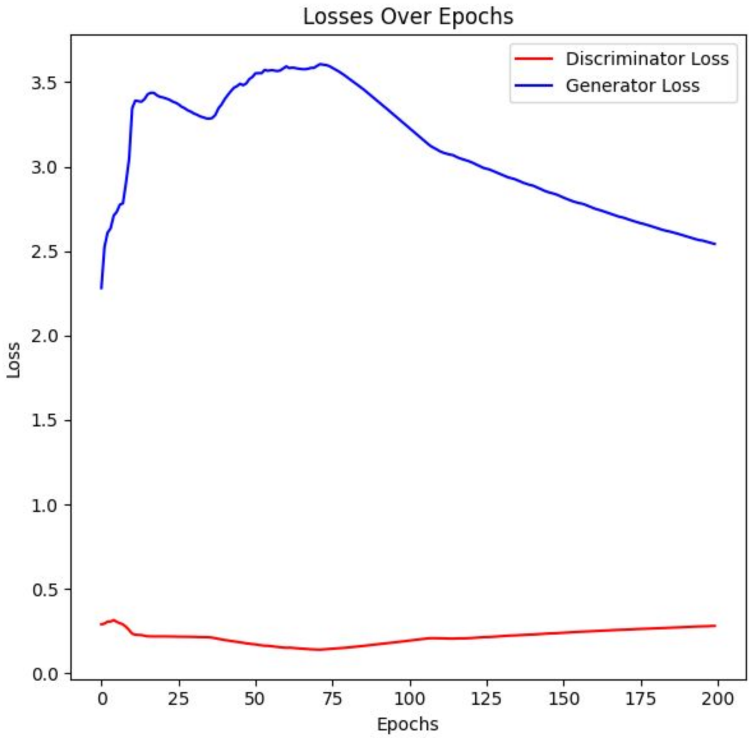
Training Results EBGAN

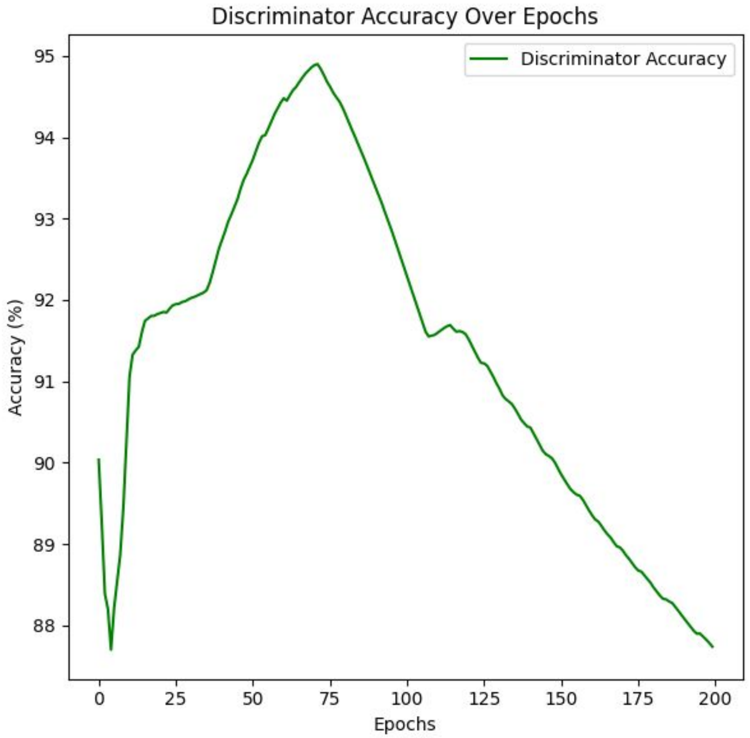




Model Performance

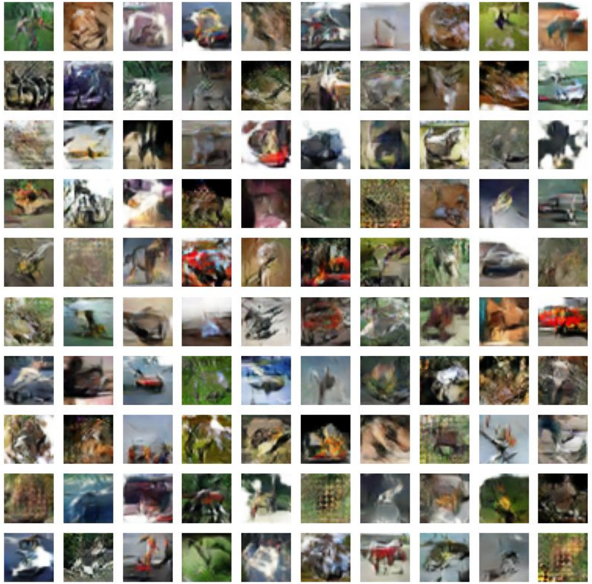
Training Results CGAN



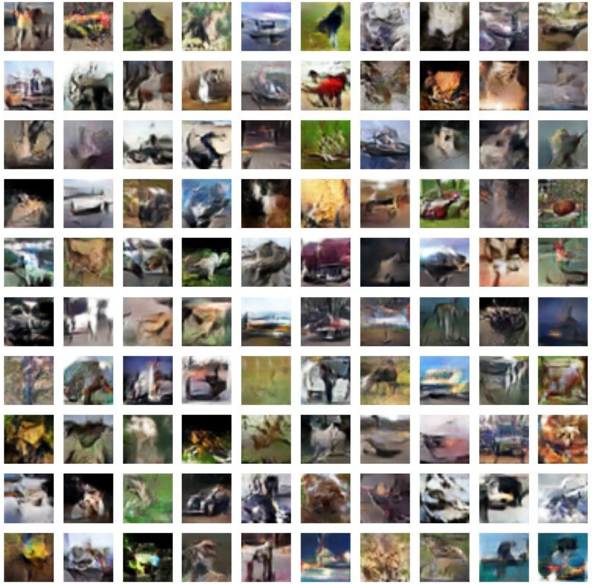


Sample Outputs

Generated Images



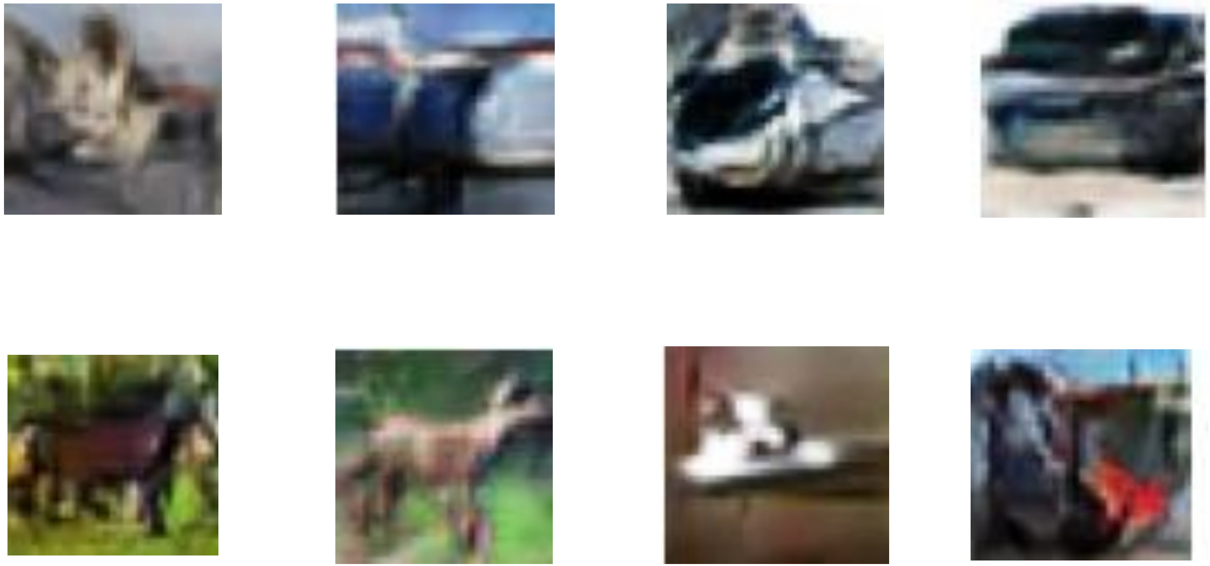
EBGAN
Energy function
Without normalization



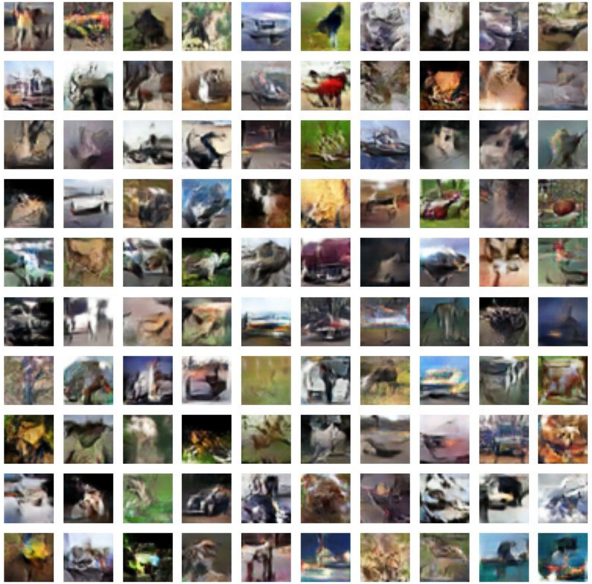
EBGAN
Energy function
With normalization

Sample Outputs

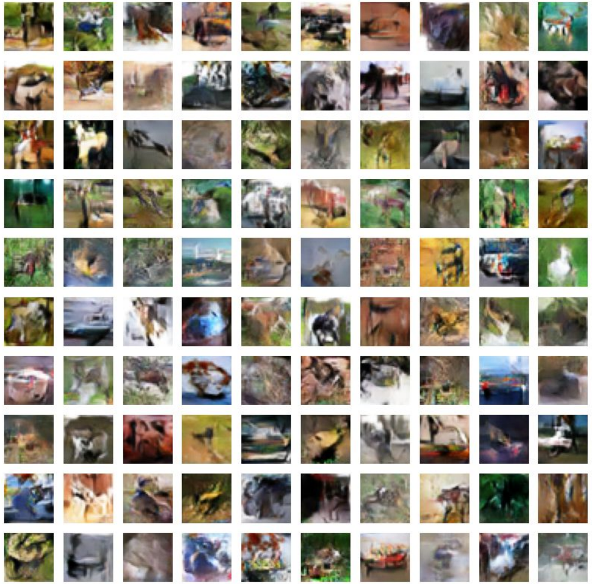
Generated Images



Sample Outputs
Generated Images



EBGAN
With normalization



CGAN
Binary Crossentropy

Conclusion

Summary & Future Improvements

Summary

- Minimizes energy scores of generated samples
- More stable training than traditional GANs
- Suited for semi-supervised learning

Improvements

- Implement a fully functional autoencoder structure
- Test with other datasets
- Test in other domains like anomaly detection and [intrusion detection]

Thank you!

Q&A