Generative Image Models

DRAW: Deep Recurrent Attention Writer by K Gregor et al.

CS698N Final Project Presentation

Nirbhay Modhe Vikas Jain

Department of Computer Science
IIT Kanpur

November 11, 2016

Outline

1. Introduction
   - Problem Statement
   - DRAW Model
2. Work Done
   - Analysis of the DRAW model
   - Latent Space Analysis
   - Proposed Models
3. What's new?
4. Contributions
5. Who did what?

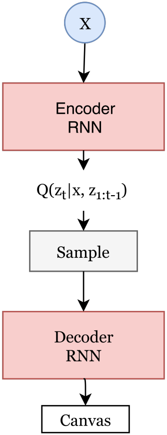
Problem Statement

- Learning generative model for images.
- Inspired from DRAW: A Recurrent Neural Nerwork for Image Generation by Gregor et al. (Google DeepMind)

DRAW Model

Existing Work

- *Encoder RNN* : Compresses real images presented during training into **latent codes** distribution.
- *Decoder RNN* : Reconstitutes images from sampled codes.
- Encoder network at every time step is made aware of the decoder output of the previous time step.
- Training Time ∼ 30 minutes (MNIST)



Outline

1. Introduction
   - Problem Statement
   - DRAW Model
2. Work Done
   - Analysis of the DRAW model
   - Latent Space Analysis
   - Proposed Models
3. What's new?
4. Contributions
5. Who did what?

Analysis of the DRAW model

	Avg.	σ	**Max.** $(\leq)$
DRAW(Attention)	643	264	1370
DRAW(No-Attention)	644	268	1337
DRAW(T=1)	653	290	1724
DRAW(T=2)	698	276	1402
DRAW(T=5)	796	302	1853
DRAW(no-privy[1])	809	298	2546
DRAW(only error image)	648	302	1853

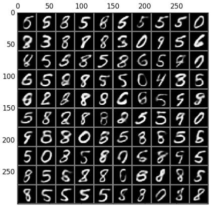
It can be observed that **DRAW model with attention** is performing better than the other models which is the proposed model of the *DRAW* paper.
Qualitative Results – in midterm report.

[1] decoder output is not made privy to the encoder

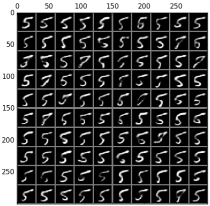
Latent Space Analysis

Effect of each time step latent code

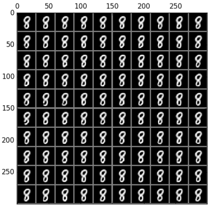
To study the effect of each time step latent code. For $T = 10$:



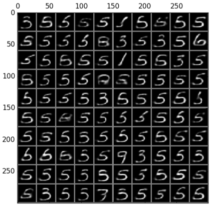
First fixed



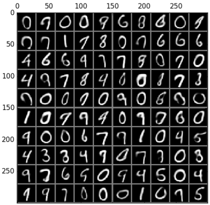
First 2 fixed



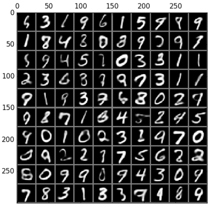
First 9 fixed



Last 9 fixed



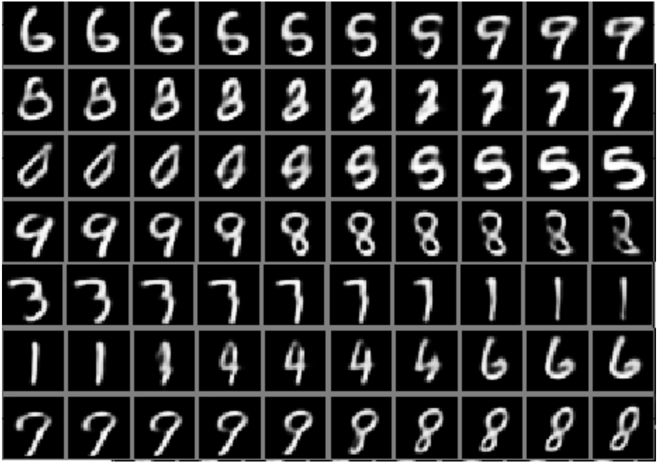
Last 8 fixed



Last fixed

Latent Space Analysis

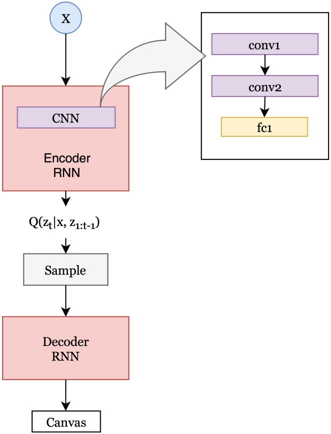
Interpolation in latent space



Proposed Models

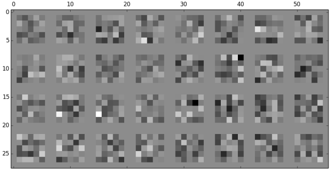
Model I - Each Step Convolution

- Introduced Convolutional layers in the Encoder network.
- Encoder reads both the input image and cnn features as input at each time step.
- Training Time $\sim$ 3 hr (MNIST)

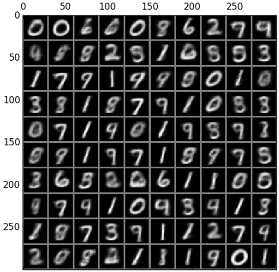


Proposed Models

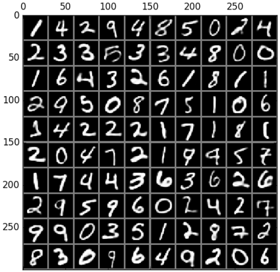
Model I - Each Step Convolution



Learned 1st layer kernels (5 × 5 × 32)



Model II

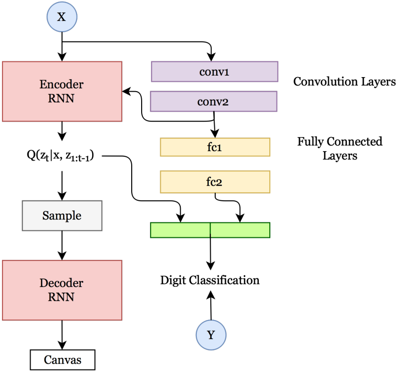


DRAW

Proposed Models

Model II - Supervised Encoder

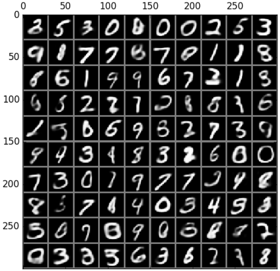
- Supervision in the form of a digit classification network
- Interconnected with the encoder network at two places.
- Training Time ∼ 3 hr (MNIST)



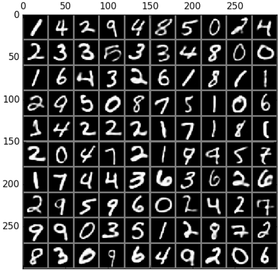
Proposed Models

Model II - Supervised Encoder

Learned 1st layer kernels (5 × 5 × 16)



Model II

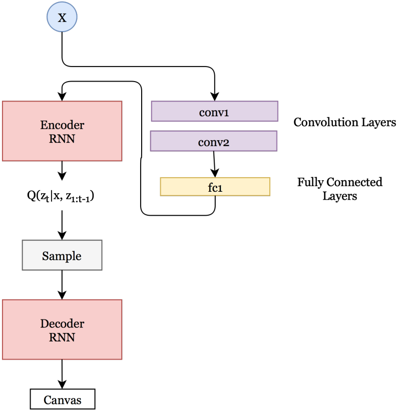


DRAW

Proposed Models

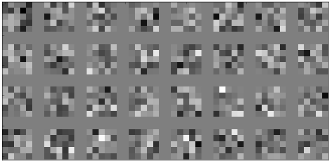
Model III - Convolutional Encoder

- Encoder is fed only the cnn features of the input image.
- CNN parameters updated using reconstruction and latent loss.
- Training Time ∼ 3 hr (MNIST)

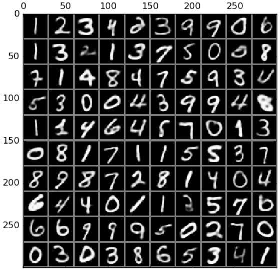


Proposed Models

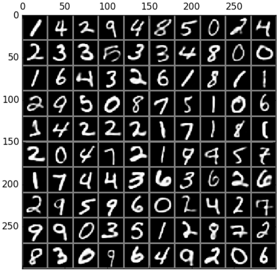
Model III - Convolutional Encoder



Learned 1st layer kernels (5 × 5 × 32)



Model III



DRAW

Proposed Models

Calculating Model Error

We calculated the negative log-likelihood of 1000 generated images for each of the model shown in the table below.

	Average Error	**Standard Deviation**	**Maximum Error** ($\leq$)
DRAW	820	327	1989
Model I - Each Step Convolution	656	267	1429
Model II - Supervised Encoder	791	295	1823
Model III - Convolutional Encoder	625	255	1637

Outline

1 Introduction
- Problem Statement
- DRAW Model

2 Work Done
- Analysis of the DRAW model
- Latent Space Analysis
- Proposed Models

3 What's new?

4 Contributions

5 Who did what?

What's new?

Things not covered in *DRAW*:

- We analyzed the network parameters of *DRAW* model.

What's new?

Things not covered in *DRAW*:

- We analyzed the network parameters of *DRAW* model.
- We analyzed the latent space of the encoded images. This deems as an important step which was missing and produced interesting observations.

What's new?

Things not covered in *DRAW*:

- We analyzed the network parameters of *DRAW* model.
- We analyzed the latent space of the encoded images. This deems as an important step which was missing and produced interesting observations.
- We proposed and trained 3 models incorporating CNN in the original *DRAW* model:
  - Model I - Each Step Convolution
  - Model II - Supervised Encoder
  - Model III - Convolutional Encoder

  Each model shows better performance than the original *DRAW* model.

Outline

1. Introduction
   - Problem Statement
   - DRAW Model
2. Work Done
   - Analysis of the DRAW model
   - Latent Space Analysis
   - Proposed Models
3. What's new?
4. Contributions
5. Who did what?

Contributions

Our contributions on top of the existing code

- Implemented **stochastic data generation** part of the paper.
- Added an interface for the **SVHN dataset** to be given as input to the draw network.
- Added **convolution and deconvolution** wrappers for **training the proposed models I, II and III**.
- Implemented the evaluation phase to **calculate the negative log-likelihood of the generated images**.
- Added new sampling functionalities to **visualize the latent space**.
- **Visualizing and plotting kernels** of the learned CNN.

Outline

1. Introduction
   - Problem Statement
   - DRAW Model
2. Work Done
   - Analysis of the DRAW model
   - Latent Space Analysis
   - Proposed Models
3. What's new?
4. Contributions
5. Who did what?

Who did what?

Task	**P**	**M**	**E**	
Reproduction of results on MNIST and SVHN	✓	✓	✓	VN
Experimenting with parameters	✓	✓	✓	VN
Evaluation using negative log likelihood	✗	✓	✓	V
Model I - Each Step Convolution	✓	✓	✓	VN
Model II - Supervised Encoder	✗	✗	✓	N
Model III - Convolutional Encoder	✗	✗	✓	V
Latent Space Analysis	✗	✗	✓	N

Table: Work proposed(P), completed at midterm(M) and at endterm(E), along with the member-wise distribution. **V** represents Vikas, **N** represents Nirbhay, **VN** represents joint contribution by both members.

Thank you!