CS698N Midterm Presentation

Generative Image Models

Nirbhay Modhe   Vikas Jain

Department of Computer Science
IIT Kanpur

CS698N, Monsoon 2016

Outline

Problem Statement

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

Outline

DRAW Model

Essential Components

- *Encoder* : Compresses real images presented during training into **latent codes**.
- *Decoder* : Reconstitutes images after from codes during training, samples a **latent distribution** during generation.

DRAW Model

Essential Components

- *Encoder* : Compresses real images presented during training into **latent codes**.
- *Decoder* : Reconstitutes images after from codes during training, samples a **latent distribution** during generation.
- A pair of **Recurrent Neural Networks** for both the encoder and decoder networks.

DRAW Model

Essential Components

- *Encoder* : Compresses real images presented during training into **latent codes**.
- *Decoder* : Reconstitutes images after from codes during training, samples a **latent distribution** during generation.
- A pair of **Recurrent Neural Networks** for both the encoder and decoder networks.
- Encoder network at every time step is made aware of the decoder output of the previous time step.

The Draw Network

Architecture

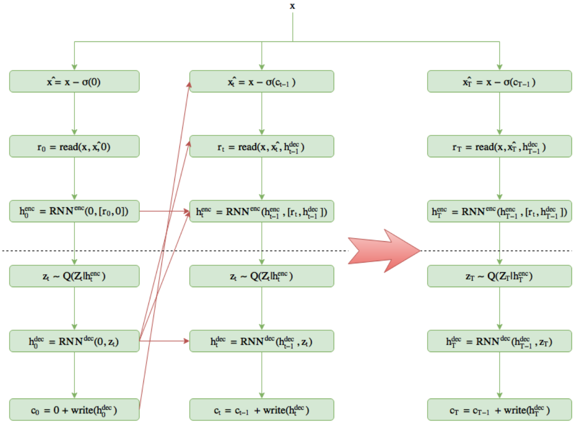


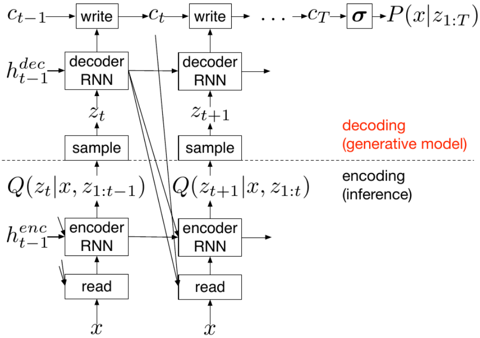
Figure: DRAW Network Architecture

The Draw Network

Equations

Decoder Equations:

- Input latent code z_t
- $z_t \sim Q(Z_t | h_t^{enc})$
- $h_t^{dec} = RNN^{dec}(h_{t-1}^{dec}, z_t)$
- $c_t = c_{t-1} + write(h_t^{dec})$
- c_t is canvas matrix



Outline

Experimenting with the DRAW Model

- Importance of attention in read and write operations.
- Effect of allowing the encoder to see previous decoder output.
- Varying the number of time steps - in particular the use of a single time step.
- Removing the original image from the read function output - only using the error image $\widehat{x_t} = x - \sigma(c_{t-1})$

Outline

Modifications to the Network

Convolution and Devoncolution layers

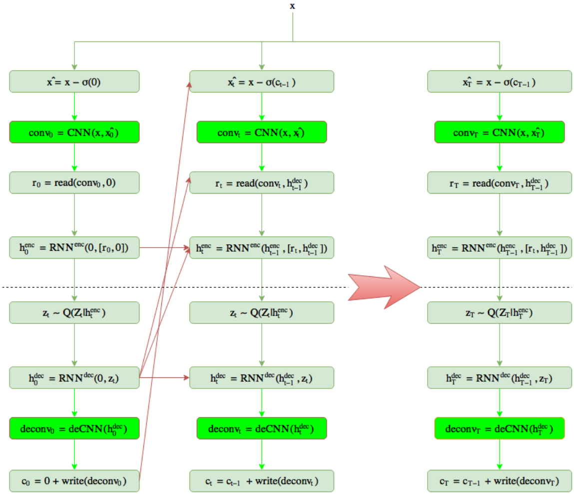


Figure: Modified DRAW Network

Convergence of DRAW network

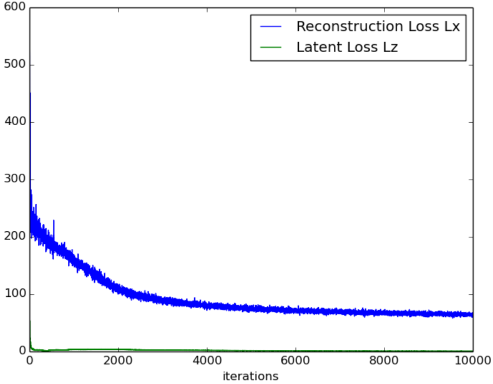


Figure: The original DRAW network convergence graph

Convergence of modified network

Initial Model Convergence

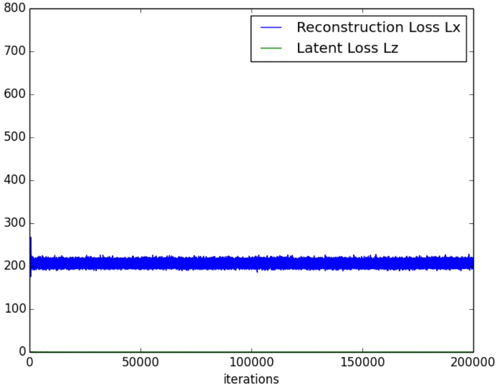


Figure: Using only convolution filter map as output of read operation

Convergence of modified network

Subsequent Changes

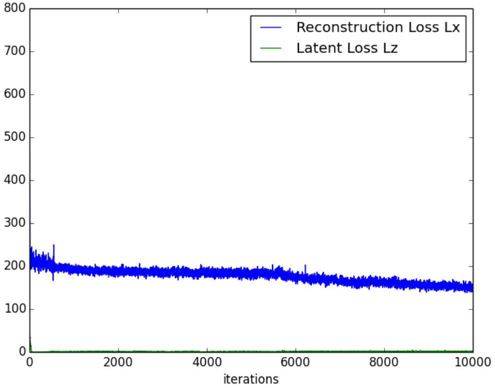


Figure: Concatenating convolution filter map with the image

Convergence of modified network

Visualising the convolution filter maps

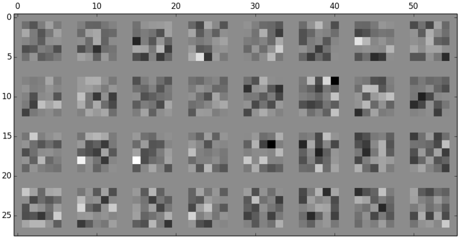


Figure: Filter maps for first convolution layer before the read operation

Outline

Analysis of all models

	Average	σ	Max ($\leq$)
DRAW(Attention)	**643.09**	**264.67**	1370.03
DRAW(No-Attention)	644.61	268.30	**1337.80**
DRAW(T=1)	653.31	290.03	1724.63
DRAW(T=2)	698.17	276.37	1402.27
DRAW(T=5)	796.44	302.79	1853.58
DRAW(no-privy[1])	809.61	298.31	2546.65
DRAW(error img)	648.66	302.79	1853.58

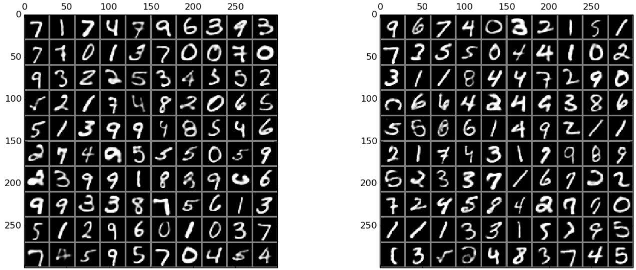
Table: Negative log likelihood error over 1000 generated images. Calculated using respect to closest image (L2 distance) in the MNIST test set.

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

Outline

Generated Images

Varying Attention

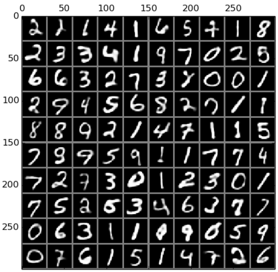


(a) DRAW - Attention   (b) DRAW - No Attn.

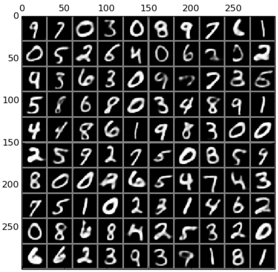
Figure: Generated images using the original DRAW model

Generated Images

Error image and decoder privy



(a) DRAW - ErrorImg

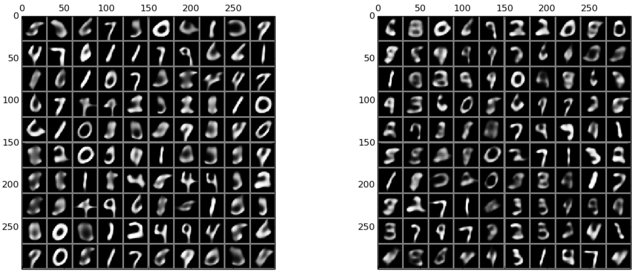


(b) DRAW - No Privy

Figure: Generated images using the original DRAW model

Generated Images

Varying time steps

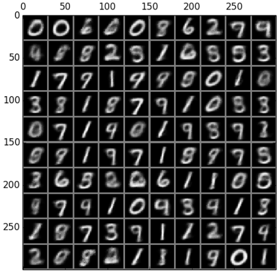


(a) DRAW - T1 (b) DRAW - T2

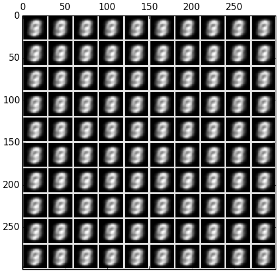
Figure: Generated images using the original DRAW model

Generated Images

Adding convolution layers to read



(a) CONV - Concat



(b) Conv - No Concat

Figure: Generated images using the filter maps of two convolution layers before the read operation with (a) and without (a) concatenation with the source image

Outline

Future Work

- Replacing spatial attention with different known mechanisms.

Future Work

- Replacing spatial attention with different known mechanisms.
- Studying possible changes to the autoencoder and latent code distribution, inspired by the recent attempts at improving the variational bayes autoencoders (Burda et. al., 2015 - Importance Weighted Autoencoders).

Future Work

- Replacing spatial attention with different known mechanisms.
- Studying possible changes to the autoencoder and latent code distribution, inspired by the recent attempts at improving the variational bayes autoencoders (Burda et. al., 2015 - Importance Weighted Autoencoders).
- Testing our models on more complicated datasets such as SVHN, CIFAR-10.