# Generative Image Models

**Project Proposal, Group 11, 1 Sep 2016**
Vikas Jain (13788), Nirbhay Modhe (13444)
CS698N: Recent Advances in Computer Vision, Jul–Nov 2016
Instructor: Gaurav Sharma, CSE, IIT Kanpur, India

## 1 Importance of the problem

The problem of designing generative image models is important due to its uses in other fields of computer vision. Learning to generate images efficiently leads to more efficient image representations. Furthermore, a model which learns to generate images given certain parameters (of the desired image) as input, can even generate images of an unseen parameter configuration provided it accurately learned the intrinsic parameter-image relationship.

## 2 Literature review

The generative adversarial network (GAN) by I. Goodfellow et. al. (2014) [2] was the first adversarial approach for learning to generate a distribution of images given as input. It was tested on MNIST and TFD (Toronto Face Detection) datasets, beating the Deep GSN (Generative Stochastic Neworks) by Bengio et. al. [6].

Dosovitsky et. al. (2015) [1] proposed a convolutional neural netowrk model similar to Goodfellow's GAN model [2], for the purpose of reconstructing images of chairs in different configurations. This model allowed for interpolation to generate chair images in previously unseen configurations, which indicated that the network learnt meaningful 3D representations of chairs, instead of memorizing training data.

Most image generation approaches tend to generate the entire image in a single step, and hence they make use of a single latent distribution for conditioning each pixel. However, the SoA paper [3] uses an iterative "refinement" procedure which is inspired by the step by step process of a human drawing an image. Additionally, the SoA paper makes use of the kind of variational autoencoder, which has recently gained popularity for the task of generative modelling (Gregor et. al.[4], Kingma et. al.[5]) .

## 3 Description of the SoA paper

**DRAW: A Recurrent Neural Network For Image Generation:** Gregor et. al. [3] proposed a recurrent encoder-decoder network for image generation, based on the earlier work on variational auto-encoders by Kingma et. al. [5]. The main idea of the work is iterative construction of complex images which combines a novel spatial attention mechanism that mimics the *foveation of the human eye* and a sequential *variational auto-encoding framework*. The encoder network compresses the real images presented during training into *latent codes*. The decoder network *reconstitutes* images after receiving codes. Both networks are implemented as recurrent networks. The encoder network is made privy to the decoder network's output from the preceding time step.

Figure1 shows the the complete architecture of the DRAW network. The recurrent networks in DRAW model run for fix number of $T$ steps. The encoder and equations are as follows:

$$\hat{x}_t = x - \sigma(c_{t-1})$$

$$r_t = read(x, \hat{x}_t, h_{t-1}^{dec})$$

$$h_t^{enc} = RNN^{enc}(h_{t-1}^{enc}, [r_t, h_{t-1}^{dec}])$$

$$z_t \sim Q(Z_t|h_t^{enc})$$

$$h_t^{dec} = RNN^{dec}(h_{t-1}^{dec}, z_t)$$

$$c_t = c_{t-1} + write(h_t^{dec})$$

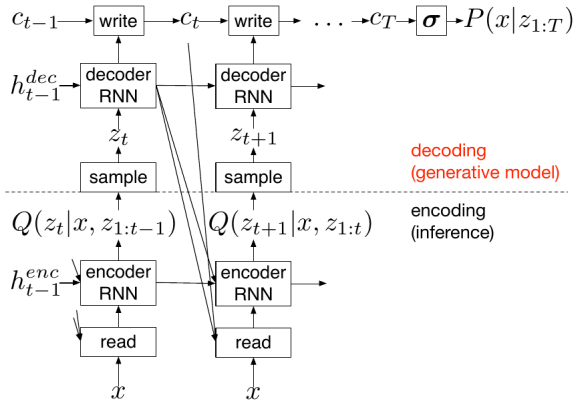


Figure 1: DRAW Network (from [3])

where $\sigma$ is sigmoid function, $x$ is the original input image, $\hat{x}_t$(error image) is the input image at time step $t$ after subtracting the output from the decoder. $h_t^{enc}$ and $h_t^{dec}$ are the outputs of encoder and decoder respectively at time step $t$. $Q$ is gaussian distribution over latent vectors($Z_t$) parametrised using output of the encoder($h_t^{enc}$). Take $read$ and $write$ as black box function as of now (explained next). At each time step $t$, canvas matrix $c_t$ is modified using previous canvas matrix $c_{t-1}$ and $write$ operation.

The final canvas matrix $c_T$ is used to parametrise a model $D(X|c_T)$ of the input data where $D$ is a Bernoulli distribution with means given by $\sigma(c_T)$. To calculate the loss, two types of loss are added *Reconstruction* Loss:

$$\mathcal{L}^x = -logD(x|c_T)$$

and *Latent* Loss:

$$\mathcal{L}^z = \Sigma_{t=1}^T KL(Q(Z_t|h_t^{enc})\|P(Z_t))$$
$$= \frac{1}{2}(\Sigma_{t=1}^T \mu_t^2 + \sigma_t^2 - log\sigma_t^2) - \frac{T}{2}$$

where

$$Q(Z_t|h_t^{enc}) = \mathcal{N}(Z_t|\mu_t, \sigma_t)$$
$$\mu_t = W(h_t^{enc})\ ,\ \sigma_t = exp(W(h_t^{enc}))$$
$$P(Z_t) \sim \mathcal{N}(0, I)$$

The functions $read$ and $write$ in the DRAW model can be implemented in two ways: with *attention* and *without attention*. The implementation without attention is straightforward with $read$ as concatenation of the original input image($x$) and error image($\hat{x}_t$); $write$ as linear transformation of $h_t^{dec}$. The implementation with attention involves 2D gaussian filter of $N \times N$ grid. The attention parameters(parameters of gaussian, stride rate and intensity) are computed using linear transformation of output of the recurrent networks ($h_t^{enc}$ and $h_t^{dec}$). The exact details are out of the scope of this proposal, please see [3] for details.

Once the model is trained, the decoder network can be used to generate images by iteratively sampling latent codes from the prior $P(Z_t)$.

$$\tilde{z}_t \sim P(Z_t)$$
$$\tilde{h}_t^{dec} = RNN^{dec}(\tilde{h}_{t-1}^{dec}, \tilde{z}_t)$$
$$c_t = c_{t-1} + write(\tilde{h}_t^{dec})$$
$$\tilde{x} \sim D(X|\tilde{c}_t)$$

The authors demonstrated the DRAW model on 5 experiments –MNIST classification, generation of MNIST data, MNIST generation with two digits, street view house number generation and generating CIFAR images. All experiments showed promising results.

# 4 Plan for the project

- We wish to reproduce results of the image generation of MNIST digits, as demonstrated in section 4.2 of the paper. The estimated completion date of this task is September 15, 2016. We will also give a demo of the MNIST digit generation, while highlighting the intermediate steps of generation.
- The paper used only an RNN (specifically an LSTM) architecture, along with the attention model. We feel that the integration of convolutional neural networks, both at the encoding and decoding stage, might improve the performance (quality of generated images) and run-time of the network as a whole.
- We will be using the Theano implementation of the paper's code, which may require a GPU for running on the CIFAR dataset, or any other dataset which we might want to include later on. If a GPU is not available, we will restrict ourselves to datasets which can be handled by only CPU (MNIST).
- Nirbhay will be undertaking the following tasks:
  - Implementing the proposed improvements (Convolutional Layer Addition) to the existing code (in Theano).
  - Producing results on the MNIST dataset after the above step.
- Vikas will be undertaking the following tasks:
  - Reproduction of results on MNIST and SVHN (Street View House Numbers) dataset
  - Experimenting with the existing code to identify the importance of each component of the architecture.
  - Verification of the proposed improvements, producing results for this code on the SVHN dataset.

# 5 Expertise of the group

Vikas and Nirbhay both have completed the following courses which will help us in this project: Computer Vision, Machine Learning Tools and Techniques, and Artificial Intelligence Programming. Nirbhay has worked on the following projects:

- Image Colorization as a course project in Computer Vision
- Object detection as a course project in Machine Learning Tools and Techniques

Vikas has done the following relevant projects:

- Learning Relative attributes which used CNNs
- Object classification which used VGG-16 model for feature representation.

Additionally, Vikas and Nirbhay have worked together on a project during the summer of 2015, which was to extend the paper by Dosovitsky et. al.[1] on generative modelling of chairs using CNNs which is closely related to the project.

# References


[1] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[3] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[4] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra. Deep autoregressive networks. *arXiv preprint arXiv:1310.8499*, 2013.

[5] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] E. Thibodeau-Laufer, G. Alain, and J. Yosinski. Deep generative stochastic networks trainable by backprop. 2014.