DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, Daan Wierstra

https://arxiv.org/abs/1502.04623
https://arxiv.org/pdf/1502.04623.pdf

TL;DR

Deep Recurrent Attentive Writer(DRAW) is a neural network architecture for image generation. DRAW networks combine a spatial attention mechanism that mimics the foveation in human eyes, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images.

Introduction

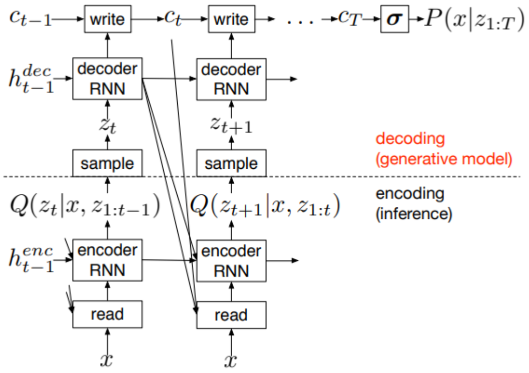
A normal person will naturally recreate a visual scene in a sequential, iterative fashion, reassessing the product after each modification. Most approaches to automatic image generation generate entire scenes at once. They are conditioned on a single latent distribution. DRAW attempts to generate images more naturally in which parts of a scene are created independently from others.

The core of the architecture is a pair of RNNs: an encoder that compresses the real images during training and a decoder that reconstitutes images after receiving the compressed images.

Rather than generating images in a single pass, it iteratively constructs scenes through an accumulation of modifications emitted by the decoder emitted by the decoder, each of which is observed by the encoder.

The DRAW Network

Basic structure similar to that of VAEs: an encoder network determines a distribution over latent codes that capture information about input data; a decoder network receives these samples and uses them to condition its own distribution over images. However, three main things are different. In DRAW, both networks are RNNs, the decoder's outputs are successively added to the distribution, and thirdly, a dynamically updated attention mechanism is used to restrict both the input region observed by the encoder, and the output region modified by the decoder.



RNN^{enc} at time t is the encoder hidden vector h_t^{enc}. Similarly, RNN^{dec} at time t is the encoder hidden vector h_t^{dec}. LSTM architecture is used for its proven track record for handling sequential data.

At each time step t, the encoder receives input from the image and the previous decoder hidden vector. The output of the encoder is used to parameterise the distribution over the latent vector.

Loss Function

Reconstruction loss- The final canvas matrix c_T, is used to parametrize a model $D(X|c_T)$. $D$is a Bernoulli distribution. The canvas matrix is a cumulative matrix produced by adding the output of the decoder on a sample drawn from the latent distribution. The canvas matrix is used to reconstruct the image.

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

Latent loss- for a sequence of latent distributions

$$\mathcal{L}^z = \sum_{t=1}^{T} KL\big(Q(Z_t|h_t^{enc})||P(Z_t)\big)$$

Total network loss

$$\mathcal{L} = \langle \mathcal{L}^x + \mathcal{L}^z \rangle_{z\sim Q}$$

Read and Write Operations

Without Attention

The entire input image is passed to the encoder at every time-step and the decoder modifies the entire canvas matrix. This approach does allow the encoder to focus on only part of the input when creating the latent distribution.

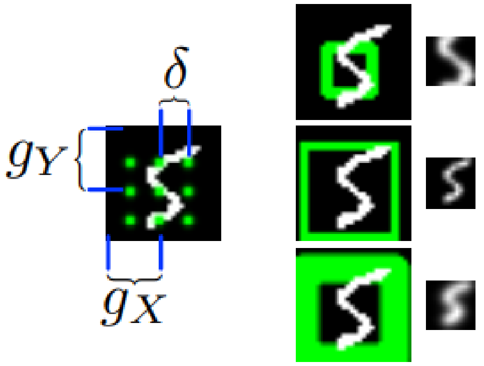


Figure 3. **Left:** A 3×3 grid of filters superimposed on an image. The stride (δ) and centre location (g_X, g_Y) are indicated. **Right:** Three $N \times N$ patches extracted from the image ($N = 12$). The green rectangles on the left indicate the boundary and precision (σ) of the patches, while the patches themselves are shown to the right. The top patch has a small δ and high σ, giving a zoomed-in but blurry view of the centre of the digit; the middle patch has large δ and low σ, effectively downsampling the whole image; and the bottom patch has high δ and σ.

Attention

2-D attention, where an array of 2D Gaussian filters is applied to the image, yielding an image 'patch' of smoothly varying location and zoom. The stride controls the zoom of the patch. The larger the stride, the larger an area of the original image will be visible in the attention patch, but the lower the effective resolution of the patch will be.

The read operation returns the concatenation of two patches from the image and error image.

For the write operation, a distinct set of attention parameters are extracted from the decoder output, the order of transposition is reversed, and the intensity is inverted.

Experiments

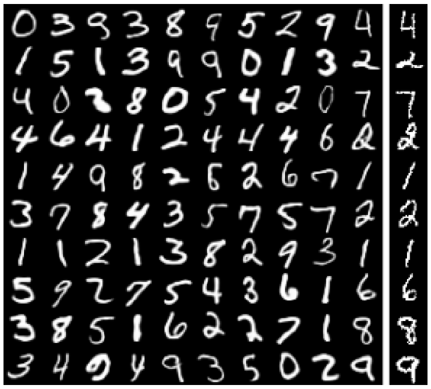


Figure 6. **Generated MNIST images.** All digits were generated by DRAW except those in the rightmost column, which shows the training set images closest to those in the column second to the right (pixelwise L^2 is the distance measure). Note that the network was trained on binary samples, while the generated images are mean probabilities.

Conclusion

DRAW generates highly realistic natural images in a more natural way. The attention mechanism is beneficial not only to image generation, but also to image classification.