DRAW: A Recurrent Neural Network For Image Generation

mingzailao

2016-9-11

Outline

Introduction

Core

The core of the DRAW architecture is a pair of recurrent neural networks:

- an encoder network that compresses the real images presented during training.
- a decoder that reconstitutes images after receiving codes.

where the loss function is a variational upper bound on the log-likelihood of the data.

Introduction

Type

It belongs to the family of variational auto-encoders(hybrid of deep learning and variational inference; generative model)

Difference

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

Introduction

Partial Glimpses

An obvious correlate of generating images step by step is the ability to selectively attend to parts of the scene while ignoring others.

The DRAW Network

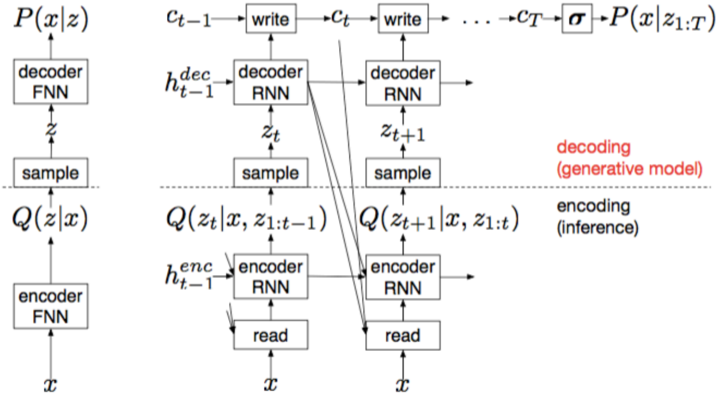

Figure: The Structure of the Net(left Conventional Variational Auto-Encoder; right:DRAW Network)

Network Architecture

Notation

- RNN^{enc} : the function enacted by the encoder network at a single time-step.
- RNN^{dec} : the function enacted by the decoder network at a single time-step.
- h_t^{enc} : the encoder hidden vector at time t.
- h_t^{dec} : the decoder hidden vector at time t.
- $b = W(a)$: a linear weight matrix with bias from the vector a to the vector b.

Network Architecture

Feedforward

For each image x presented to the network, c_0, h_0^{enc}, h_0^{dec} are initialised to learned biases, then:

$$
\begin{aligned}
\hat{x}_t &= x - \sigma(c_{t-1}) \\
r_t &= read(x_t, \hat{x}_t, h_{t-1}^{dec}) \\
h_t^{enc} &= RNN^{enc}(h_{t-1}^{enc}, [r_t, h_{t-1}^{dec}]) \\
z_t &\sim Q(Z_t | h_t^{enc}) \\
h_t^{dec} &= RNN^{dec}(h_{t-1}^{dec}, z_t) \\
c_t &= c_{t-1} + write(h_t^{dec})
\end{aligned}
$$

Network Architecture

Note 1

For $z_t \sim Q(Z_t|h_t^{enc})$, in this we use a diagonal Gaussian $\mathcal{N}(Z_t|\mu_t, \sigma_t)$:

$$\begin{aligned} \mu_t &= W(h_t^{enc}) \\ \sigma_t &= exp(W(h_t^{enc})) \end{aligned}$$

Note 2

For the read and write operation, we will explain it later.

Network Architecture

Tensor Shape

Tensor	Shape
x	$(Batch\_size, B * A)$
$\hat{x}$	$(Batch\_size, B * A)$
r_t	$(Batch\_size, 2 * N * N)$
h_t^{enc}	$(Batch\_size, enc\_size)$
z_t	$(Batch\_size, z\_size)$
h_t^{dec}	$(Batch\_size, dec\_size)$
$write(h_t^{dec})$	$(Batch\_size, B * A)$

Network Architecture

Reconstruction Loss $\mathcal{L}^x$

The final canvas matrix c_T is used to parameterise a model $D(X|c_T)$ of the input data. Like we chose Guassian for latent distribution, we chose Bernoulli distribution for reconstruction, the means of the Bernoulli distribution is $\sigma(c_T)$.

Reconstruction Loss $\mathcal{L}^x$

The reconstruction loss $\mathcal{L}^x$ is defined as the negative log probability of x under D:

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

Network Architecture

Latent Loss $\mathcal{L}^z$

The latent loss $\mathcal{L}^z$ for a sequence of latent distributions $Q(Z_t|h_t^{enc})$ is defined as the summed KL-divergence of some latent prior $P(Z_t)$ from $Q(Z_t|h_t^{enc})$:

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

Chose $P(Z_t)$: a standard Gaussian with mean zero and standard deviation one,

$$\mathcal{L}^z = \frac{1}{2}(\sum_{t=1}^{T} \mu_t^2 + \sigma_t^2 - \log \sigma_t^2) - \frac{T}{2}$$

Network Architecture

Loss

$$\mathcal{L} = < \mathcal{L}^x + \mathcal{L}^z >_{z \sim Q} \tag{1}$$

Network Architecture

Stochastic Data Generation

An image $\tilde{x}$ can be generated by a DRAW network by iteratively picking latent samples $\tilde{z}_t$ from the prior P, then running the decoder to update the canvas matrix $\tilde{c}_t$. After T repetitions of this process the generated image is a sample from $D(X|\tilde{c}_T)$:

$$
\begin{aligned}
\tilde{z}_t &\sim P(z_t) \\
\tilde{h}_t^{dec} &= RNN^{dec}(\tilde{h}_{t-1}^{dec}, \tilde{z}_t) \\
\tilde{c}_t &= \tilde{c}_{t-1} + write(\tilde{h}_t^{dec}) \\
\tilde{x} &\sim D(X|\tilde{c}_T)
\end{aligned}
$$

Read and Write Operations

- Reading and Writing Without Attention
- Selective Attention Model
- Reading and Writing With Attention

Reading and Writing Without Attention

Reading and Writing Function

Without Partial Glimpses:

$$\begin{aligned} read(x, \hat{x}_t, h^{dec}_{t-1}) &= [x, \hat{x}_t] \\ write(h^{dec}_t) &= W(h^{dec}_t) \end{aligned}$$

Selective Attention Model

Notations

- (g_X, g_Y) : The grid centre.
- δ : stride
- μ_X^i, μ_Y^j : mean location of the filter at row i, column j in the patch

$$
\begin{aligned}
\mu_X^i &= g_X + (i - N/2 - 0.5)\delta \\
\mu_Y^j &= g_Y + (j - N/2 - 0.5)\delta
\end{aligned}
$$

- σ^2 : variance
- γ : a scalar intensity that multiplies the filter response

Selective Attention Model

Get Parameters

Given an $A \times B$ input image x, all five attention parameters are dynamically determined at each time step via a linear transformation of the decoder output h^{dec}:

$$
\begin{aligned}
(\tilde{g}_X, \tilde{g}_Y, log\sigma^2, log\tilde{\delta}, log\gamma) &= W(h^{dec}) \\
g_X &= \frac{A+1}{2}(\tilde{g}_X + 1) \\
g_Y &= \frac{B+1}{2}(\tilde{g}_Y + 1) \\
\delta &= \frac{max(A, B) - 1}{N - 1}\tilde{\delta}
\end{aligned}
$$

Selective Attention Model

Get Filterbank

The horizontal and vertical filterbank matrices F_X (tensor shape : $N \times A$) and F_Y (tensor shape : $N \times B$):

$$
\begin{aligned}
F_X[i,a] &= \frac{1}{Z_X} exp(-\frac{(a-\mu_X^i)^2}{2\sigma^2}) \\
F_Y[j,b] &= \frac{1}{Z_Y} exp(-\frac{(b-\mu_Y^j)^2}{2\sigma^2})
\end{aligned}
$$

Reading and Writing With Attention

Reading and Writing Function

$$read(x, \hat{x}_t, h_{t-1}^{dec}) = \gamma[F_Y x F_X^T, F_Y \hat{x}_t F_x^T]$$

For the write operation, a distinct set of attention parameters $\hat{\gamma}$, $\hat{F}_X$, $\hat{F}_Y$ are extracted from h_t^{dec}, the order of transposition is reversed and the intensity is inverted:

$$\begin{aligned} w_t &= W(h_t^{dec}) \\ write(h_t^{dec}) &= \frac{1}{\hat{\gamma}} \hat{F}_Y^T w_t \hat{F}_X \end{aligned}$$

- w_t : the $N \times N$ writing patch emitted by h_t^{dec}

Next article

About variational auto-encoder

Auto-Encodeing Variational Bayes