Student ID : 20194293

Name : Go, Kyeong Ryeol

[AI 502] DRAW: A Recurrent Neural Network For Image Generation

1. Paper Summary

Most approaches to automatic image generation aim to generate entire scene at once. However, this is very different from the way that human draw, paint or recreate a visual scene which is done in a sequential and iteration manner. This paper made a novel network architecture named as 'The Deep Recurrent Attentive Writer (DRAW)' where the parts of a scene are created independently from others and the approximate sketches are successively refined.

It consists of a pair of RNN where the encoder determines a distribution over latent codes $\mathrm{Q}(\mathrm{Z_t}|h_t^{enc})$ and the decoder receives samples from the code distribution and uses them to condition its own distribution over image. Here, they used LSTM due to its strength in handling long-range dependencies and the latent distributions were assumed to follow the diagonal Gaussian distribution so that the gradient of a function of the distribution parameters can be easily obtained through reparameterization trick. Furthermore, the encoder is privy to the decoder's previous outputs and the decoder's outputs successively updates the canvas matrix that ultimately generate the data. As a result, the DRAW network iteratively computes the following equations for $\mathrm{t} = 1, \dots, \mathrm{T}$ and this can be optimized by maximizing the variational upper bound on the log-likelihood just as Variational Autoencoder $\mathcal{L} = < -\log D(x|c_T) + \sum_{t=1}^{T} KL(Q(Z_t|h_t^{enc})||p(Z_t) >_{z\sim Q}$;

$$\hat{\mathrm{x}}_\mathrm{t} = x - \sigma(c_{t-1}) \qquad r_t = read\left(x_t, \hat{\mathrm{x}}_\mathrm{t}, \mathrm{h}_{\mathrm{t-1}}^{dec}\right) \qquad \mathrm{h}_\mathrm{t}^{enc} = RNN^{enc}\left(h_{t-1}^{enc}, [r_t, h_{t-1}^{dec}]\right)$$

$$z_t \sim Q(Z_t|h_t^{enc}) \qquad h_t^{dec} = RNN^{dec}\left(h_{t-1}^{dec}, z_t\right) \qquad c_t = c_{t-1} + write\left(h_t^{dec}\right)$$

If $read\left(x_t, \hat{\mathrm{x}}_\mathrm{t}, \mathrm{h}_{\mathrm{t-1}}^{dec}\right) = [\mathrm{x}, \hat{x_t}]$ and $\mathrm{write}\left(\mathrm{h}_\mathrm{t}^{dec}\right) = W(h_t^{dec})$, it does not allow the encoder to focus on only the particular region of the input when creating the latent distribution and the decoder to modify only a part of the canvas vector. Therefore, the author proposed to utilize the selective attention mechanism in these operations where all five attention parameters $\left(\tilde{g}_X, \tilde{g}_Y, \log\sigma^2, \log\tilde{\delta}, \log\gamma\right)$ are dynamically determined at each time step via a linear transformation of the decoder output. Then, the horizontal and vertical filter-bank matrices $\mathrm{F_X}[i,a] = \frac{1}{Z_X}\exp\left(-\frac{(a-\mu_X^i)^2}{2\sigma^2}\right)$, $\mathrm{F_Y}[j,b] = \frac{1}{Z_Y}\exp(-\frac{(b-\mu_Y^i)^2}{2\sigma^2})$ are computed where $\mu_X^i = \mathrm{g_X} + \left(i - \frac{N}{2} - 0.5\right)\delta$, $\mu_Y^i = \mathrm{g_Y} + \left(j - \frac{N}{2} - 0.5\right)\delta$. (Here, (i,j) is a point in the attention patch, (a,b) is a point in the input image and $\mathrm{Z_x}, Z_Y$ are normalization constants.) Finally, 'read' and 'wrtie' operation can be specified as follows;

$$read\left(x_t, \hat{\mathrm{x}}_\mathrm{t}, \mathrm{h}_{\mathrm{t-1}}^{dec}\right) = \gamma[F_Y x F_X^T, F_Y \hat{x} F_X^T] \qquad write\left(h_t^{dec}\right) = \frac{1}{\hat{\gamma}}\widehat{F_Y^T} W(h_t^{dec})\widehat{F_X}$$

In the experiment, it was shown that with attention mechanism, the model constructs the digit by tracing the lines, while without attention, it progressively sharpens a blurred image in a global way. However, both of them performs comparably to other recent generative models such as DARN, NADE and DBMS. Furthermore, it turns out that the two-dimensional differentiable attention mechanism embedded in DRAW is beneficial not only to image generation but also to image classification.

2. Discussion

Here I would like to offer 2 discussion points. To begin with, how can we impose the disentanglement constraint on latent codes? When an image is generated, the latent samples are iteratively sampled from the prior. If some degree of the disentanglement is guaranteed, the model can understand better about the data distribution so that we can generate an image with specific pre-determined settings by carefully choosing the latent samples. Next, how can we deliberately control the regularization effect? Since there is a lot of parameters in the models, it is vulnerable to overfitting. KL divergence plays a role as a regularization term in the loss function, but we can not directly control its effect while training. I suggest to introduce an additional parameter β that can scale the KL divergence term just like $\beta - \mathrm{VAE}$.