Lab5 - Conditional VAE For Video Prediction

Hong-Sheng, Xie

- Deadline: May 10, 2022 11:59 a.m.
- Demo Date: May 10, 2022
- Format: Experimental report (.pdf) and source code (.py). Zip all files in one file and name it like DLP_LAB5_yourID_name.zip .

1 Lab Description

In this lab, you need to implement a conditional VAE for video prediction. VAE [5] has been applied to many computer vision tasks such as super-resolution, compression. Specifically, your model should be able to do prediction based on past frames. For example, when we input frame x_{t-1} to the encoder, it will generate a latent vector h_{t-1}. Then, we will sample z_t from fixed prior. Eventually, we take the output from the encoder and z_t with the action and position (the condition) as the input for the decoder and we expect that the output frame should be next frame $\hat{x_t}$ (see in Fig. 1b).

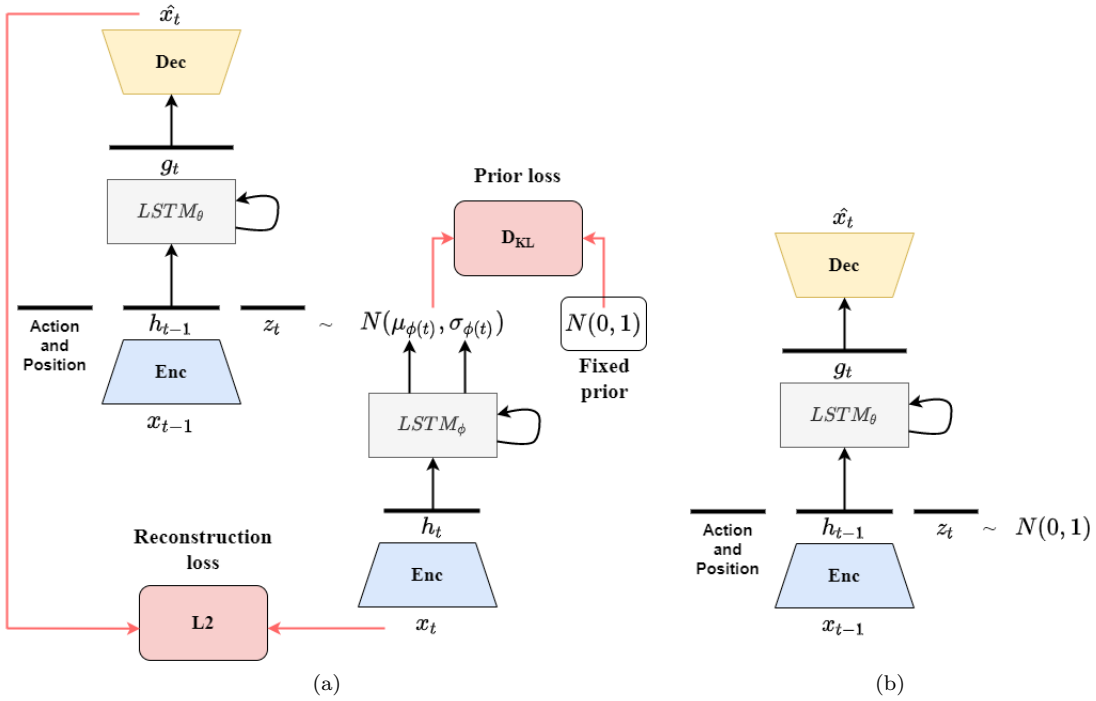

Figure 1: The illustration of overall framework. (a) Training procedure (b) Generation procedure

2 Requirements

- **Implement a conditional VAE model**
  - Modify training functions
  - Implement dataloader, teacher forcing, KL annealing, and reparameterization trick
- **Plot the traning loss and PSNR curves during training**
  - Teacher forcing ratio
  - KL annealing schedules (two methods)
- **Make videos or gif images for test result (select one sequence)**
  - Example (https://reurl.cc/VjOpWN)
- **Output the prediction at each time step (select one sequence)**

Figure 2: Prediction at each time step

3 Implementation Details

3.1 Variational Autoencoder

Recall that the loss function of VAE

$$L(X, q, \theta) = E_{z\sim q(Z|X;\phi)}\log p(X|Z;\theta) - KL(q(Z|X;\phi)||p(Z)) \tag{1}$$

where $q(Z|X;\phi)$ is considered as encoder and $p(X|Z;\theta)$ as decoder.
To train the model end-to-end, we adopt the reparameterization trick. (see in Fig. 3). The output of reparameterization trick should be log variance instead of variance directly.

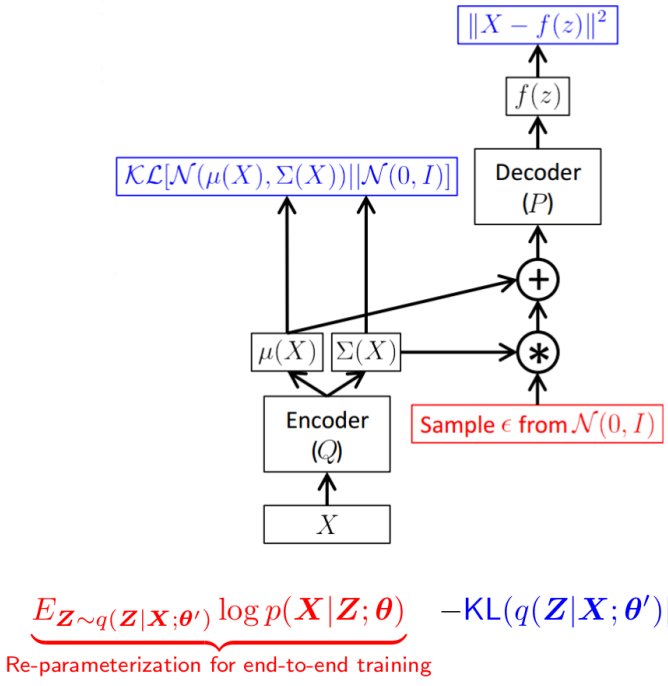

Figure 3: The illustration of reparameterization trick.

3.2 Conditional VAE

The objective function of conditional VAE is formulated as

$$L(X, c, q, \theta) = E_{z\sim q(Z|X,c;\phi)}\log p(X|Z, c; \theta) - KL(q(Z|X, c; \phi)||p(Z|c)) \tag{2}$$

where both the encoder $q(Z|X, c; \phi)$ and the decoder $p(X|Z, c; \theta)$ need to take c as part of their input. There several ways to add the conditional part to your VAE model. In the figure of model architecture, we concatenate the condition part with the latent vector z as input of decoder. Before the concatenation, we construct condition embeddings via projection. You can adopt nn.Embedding and decide the size of your condition embeddings. You can also try to convert your condition into one-hot vector.

3.3 KL cost annealing

We add a variable weight to the KL term in the loss function. We initially set the weight to 0. The maximum value is 1. Fig. 4 shows monotonic and cyclical annealing method. You should adopt these two methods and compare their results.

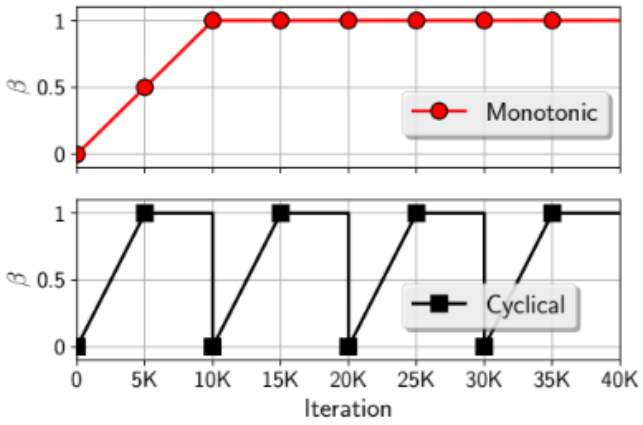

Figure 4: The design of KL annealing schedule.

3.4 Other implementation details

- The model is trained by optimizing the variational lower bound. 3

$$\begin{aligned} max\ L_{\theta,\phi}(\mathbf{x}_{1:T}) &= \sum_{t=1}^{T}[E_{q_\phi(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})} log\ p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) - \beta D_{KL}(q_\theta(\mathbf{z}_t|\mathbf{x}_{1:t})||p(\mathbf{z}))] \\ \Longleftrightarrow max &\sum_{t=1}^{T}[||\mathbf{x}_t - \hat{\mathbf{x}}_t||_2^2 - \beta D_{KL}(q_\theta(\mathbf{z}_t|\mathbf{x}_{1:t})||p(\mathbf{z}))] \end{aligned} \tag{3}$$

  where β is the trade-off between minimizing frame prediction error and fitting the prior.

- Adopt PSNR to evaluate the perceptual quality. (provided in the sample code.)
- If the dimensions between layers are mismatched, you can adopt extra fully-connected layers to transform the channle size.
- Your KL annealing schedules only follow the similar notions of monotonic and cyclical methods; that is, it is not necessary to implement the exactly the same methods.

- Hyper-parameters and model setting.
  - RNN hidden size: 256
  - Latent size(z): 128
  - Latent size(g): 64
  - KL weight(β): $0 \sim 1$
  - Teacher forcing ratio: $0 \sim 1$
  - Learning rate: 0.002

4 Derivation of Conditional VAE

Derive the objective function of conditional VAE 2. Start from the EM algorithm (L13, page 23)

5 Dataset Descriptions

We use bair robot pushing small dataset [4] to train CVAE. This data set contains roughly 44,000 sequences of robot pushing motions, and each sequence include 30 frames. In addition, it also contains action and end-effector position for each time step. You can download the dataset by the following command.

5.1 For Linux

pip install gdown
gdown https://drive.google.com/uc?id=1t8iOfBUYDFQH65RNWIgsEYenAo-HktAu
unzip processed_data.zip

5.2 For Windows

Click the following link and download it: https://reurl.cc/GoyZGd

6 Scoring Criteria

1. Report (60%)
   - Introduction (5%)
   - Derivation of CVAE (Please use the same notation in Fig.1a)(10%)
   - Implementation details (15%)
     - Describe how you implement your model (encoder, decoder, reparameterization trick, dataloader, etc.). (10%)
     - Describe the teacher forcing (including main idea, benefits and drawbacks.) (5%)
   - Results and discussion (30%)
     - Show your results of video prediction (10%)
       (a) Make videos or gif images for test result (select one sequence)
       (b) Output the prediction at each time step (select one sequence)
     - Plot the KL loss and PSNR curves during training (5%)
     - Discuss the results according to your setting of teacher forcing ratio, KL weight, and learning rate. Note that this part mainly focuses on your discussion, if you simply just paste your results, you will get a low score. (15%)

2. Demo (40%)

   - Capability of video prediction. (20%)
     Your model should use two past frames to predict the next ten frames (Although each sequence contains 30 frames, we only test the model on first 12 frames of each sequence) .
     score = PSNR (Average your score on testing dataset)

Accuracy	Grade
score $\geq$ 25	100%
25 $>$ score $\geq$ 24	90%
24 $>$ score $\geq$ 23	80%
23 $>$ score $\geq$ 22	70%
22 $>$ score $\geq$ 21	60%
21 $>$ score $\geq$ 20	50%
score $<$ 20	0%

   - Questions (20%)

3. Extra (30%)

   (a) Implement learned prior by referring to [3] (10%)

   (b) Implement hierarchical structure by referring to [6] (10%)

   (c) Implement conditional convolution by referring to [2] (10%)

7 Useful Hints

1. You need to do the lab as early as possible because you may take more than two days to train the model.
2. You should know how traditional VAE works [1] before you start to build the model.
3. Studying [3] is very helpful for you to implement the conditional VAE for video prediction.
4. The teacher forcing ratio and KL weight are very important for training this model and significantly influence the performance.

References

[1] Vae reference code. `https://github.com/pytorch/examples/tree/master/vae`.

[2] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable Rate Deep Image Compression With a Conditional Autoencoder. *arXiv e-prints*, page arXiv:1909.04802, Sept. 2019.

[3] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *CoRR*, abs/1802.07687, 2018.

[4] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections, 2017.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[6] Zhihui Lin, Chun Yuan, and Maomao Li. Haf-svg: Hierarchical stochastic video generation with aligned features. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 991–997. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.