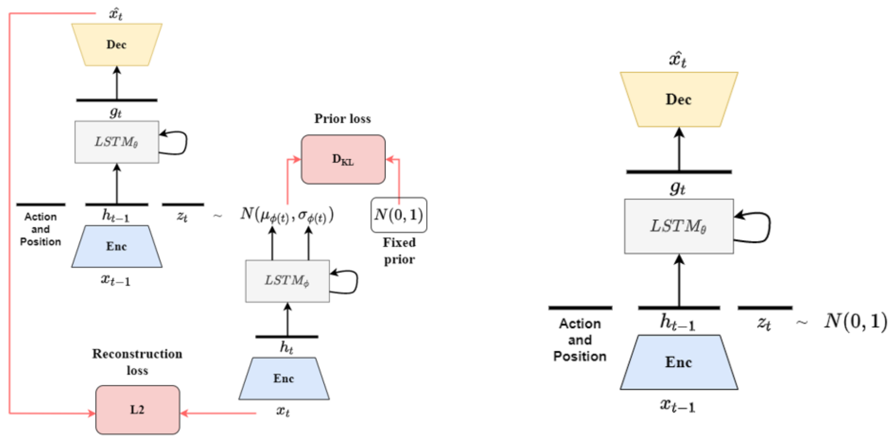
DL Lab4 Report - Conditional VAE for Video Prediction

匯出pdf之後排版變得有點醜 😥😥
助教不介意的話 可以到HackMD網址閱讀 謝謝～～～
https://hackmd.io/@ZdXM6gDQSTGTtZGr5jTo4Q/H1-BICJC5 (https://hackmd.io/@ZdXM6gDQSTGTtZGr5jTo4Q/H1-BICJC5)

Introduction

In this lab, we need to implement a conditional VAE for video prediction. The model should be able to do prediction based on past frames. For example, when we input frame xt−1 to the encoder, it will generate a latent vector ht−1. Then, we will sample zt from fixed prior. Eventually, we take the output from the encoder and zt with the action and position as the input for the decoder and we expect that the output frame should be next frame ˆxt.



(a) Training procedure (b) generating procedure

Requirments

- Implement a conditional VAE model
- Plot the training loss and PSNR curves during training
- Make videos or gif images for test result
- Output the prediction at each time step

Dataset: bair robot pushing small dataset

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.

Derivation of CVAE

> $log\, p(X|c;\theta) = log\, p(X,Z|c;\theta) - log\, p(Z|X,c;\theta)$

> $\int q(Z|X,c;\phi) log\, p(X|c;\theta) dZ$
> $= \int q(Z|X,c;\phi) log\, p(X,Z|c;\theta) dZ - \int q(Z|X,c;\phi) log\, p(Z|X,c;\theta) dZ$
> $= \int q(Z|X,c;\phi) log\, p(X,Z|c;\theta) dZ - \int q(Z|X,c;\phi) log\, q(Z|X,c;\phi) dZ + \int q(Z|X,c;\phi) log\, q(Z|X,c;\phi) dZ - \int q(Z|X,c;\phi) log\, p(Z|X,c;\theta) dZ$
> $= L(X,c,q,\theta) + \int q(Z|X,c;\phi) log \frac{q(Z|X,c;\phi)}{p(Z|X,c;\theta)} dZ$
> $= L(X,c,q,\theta) + KL(q(Z|c;\phi)||p(Z|X,c;\theta))$
> $\Rightarrow L(X,c,q,\theta) = log\, p(X|c;\theta) - KL(q(Z|c;\phi)||p(Z|X,c;\theta))$

> $L(X,c,q,\theta)$
> $= \int q(Z|X,c;\phi) log\, p(X,Z|c;\theta) dZ - \int q(Z|X,c;\phi) log\, q(Z|X,c;\phi) dZ$
> $= E_{Z\sim q(Z|X,c;\phi)} log\, p(X,Z|c;\theta) - E_{Z\sim q(Z|X,c;\phi)} log\, q(Z|X,c;\phi)$
> $= E_{Z\sim q(Z|X,c;\phi)} log\, p(X|Z,c;\theta) + E_{Z\sim q(Z|X,c;\phi)} log\, p(Z|c) - E_{Z\sim q(Z|X,c;\phi)} log\, q(Z|X,c;\phi)$
> $= E_{Z\sim q(Z|X,c;\phi)} log\, p(X|Z,c;\theta) - KL(q(Z|X,c;\phi)||p(Z|c))$
> $\Rightarrow L(X,c,q,\theta) = E_{Z\sim q(Z|X,c;\phi)} log\, p(X|Z,c;\theta) - KL(q(Z|X,c;\phi)||p(Z|c))$

Implementation details

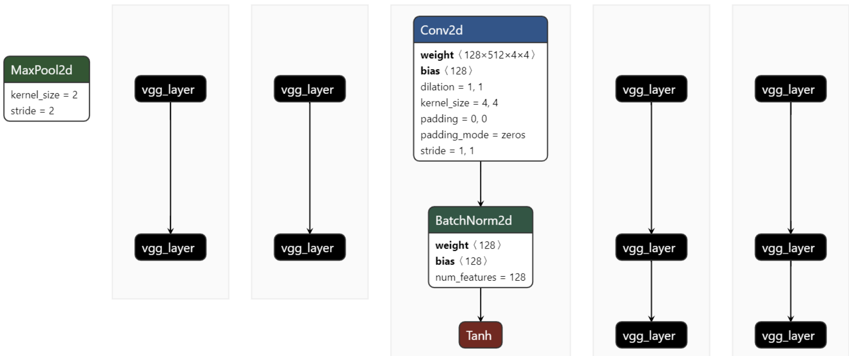
Describe how you implement your model

- dataloader

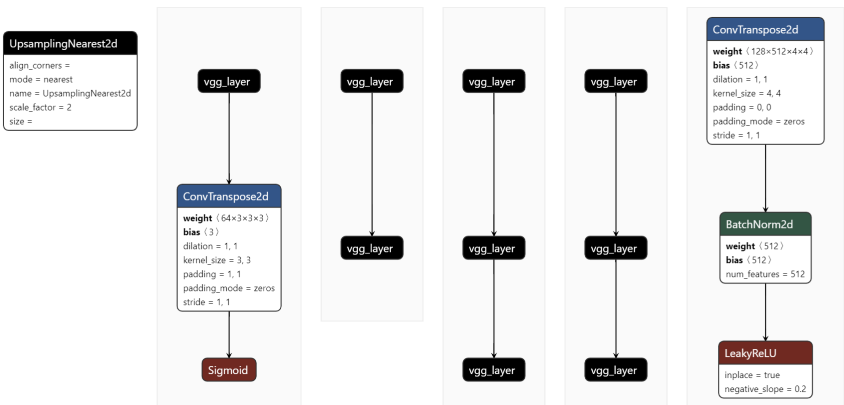
  使用torch.utils.data內建的DataLoader

```
train_loader = DataLoader(train_data,
                          num_workers=args.num_workers,
                          batch_size=args.batch_size,
                          shuffle=True,
                          drop_last=True,
                          pin_memory=True)
```

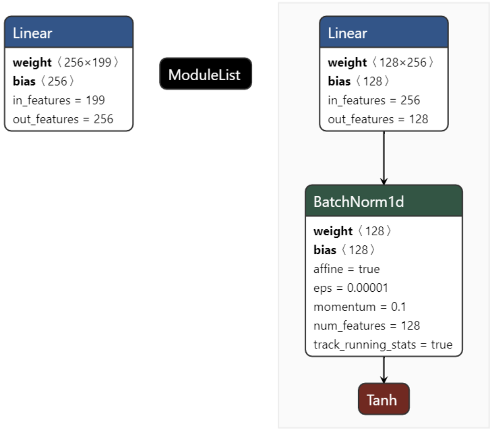
- encoder



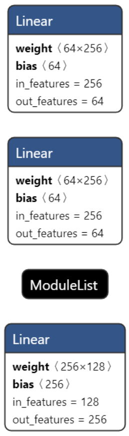
- decoder



- frame predictor



- posterior



Describe the teacher forcing

- main idea

  在training model過程中，不使用上一個state的輸出作為下一個state的輸入，而是直接使用training dataset的ground truth作為下一個state的輸入。也就是說，在training過程的time t，使用training dataset的ground truth y(t)，作為下一個time step的輸入x(t+1)，而不是使用model生成的輸出^x(t)。

- benefits

  free running會因爲生成的錯誤結果，導致後續的learning都受到不好的影響，model偏離正軌，會導致learning速度變慢，也變得不穩定。使用teacher forcing可以避免這種情形。

- drawbacks

  因爲過於依賴ground truth，在training過程中，model會有較好的效果，但是在測試的時候因爲不能得到ground truth的支持，所以如果目前生成的sequence在training過程中有很大不同，model就會變得脆弱。也就是說，這種

model的cross-domain能力會更差，如果testing dataset與training dataset來自不同的領域，model的performance就會變差。

```
1  def pred(validate_seq, validate_cond, modules, args, device):
2      gen_seq = []
3  
4      validate_seq = validate_seq.transpose_(0, 1)
5      validate_cond = validate_cond.transpose_(0, 1)
6  
7      h_seq = [modules['encoder'](validate_seq[i]) for i in range(args.n_past)]
8      modules['frame_predictor'].hidden = modules['frame_predictor'].init_hidden()
9      gen_seq.append(validate_seq[0])
10     x_in = validate_seq[0]
11 
12     for i in range(1, args.n_past+args.n_future):
13         z_t = torch.tensor(np.random.normal(0, 1, args.batch_size*64), device=devi
14         if args.last_frame_skip or i < args.n_past:
15             h, skip = h_seq[i-1]
16         elif i < args.n_past:
17             h, _ = h_seq[i-1]
18         if i < args.n_past:
19             h = modules['frame_predictor'](torch.cat([validate_cond[i], h, z_t], 1
20         else:
21             h, _ = modules['encoder'](gen_seq[-1])
22             h = modules['frame_predictor'](torch.cat([validate_cond[i], h, z_t], 1
23             x_in = modules['decoder']([h, skip])
24         gen_seq.append(x_in)
25 
26     return gen_seq
```

可以看到第21行的gen_seq[-1]，表示是使用previous predicted frame to predict next frame。

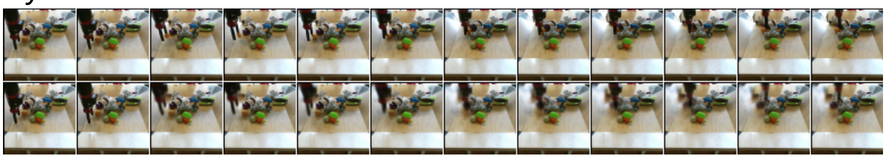
Results and discussion

Show your results of video prediction

- Make videos or gif images for test result

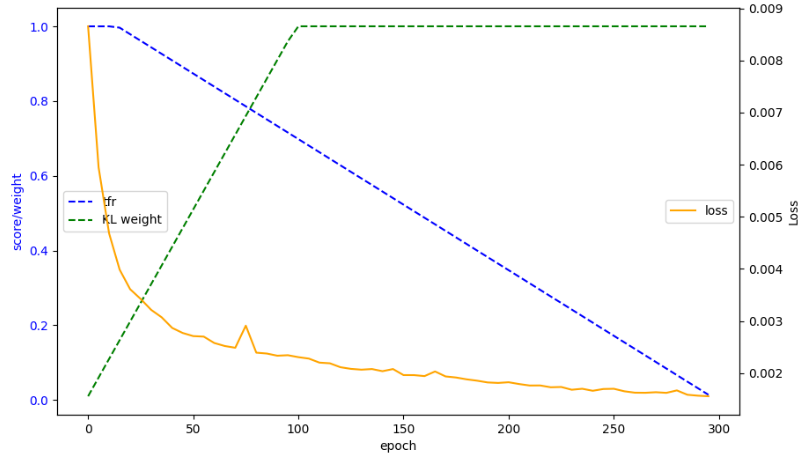
1. Monotoic
   
2. Cyclical
   

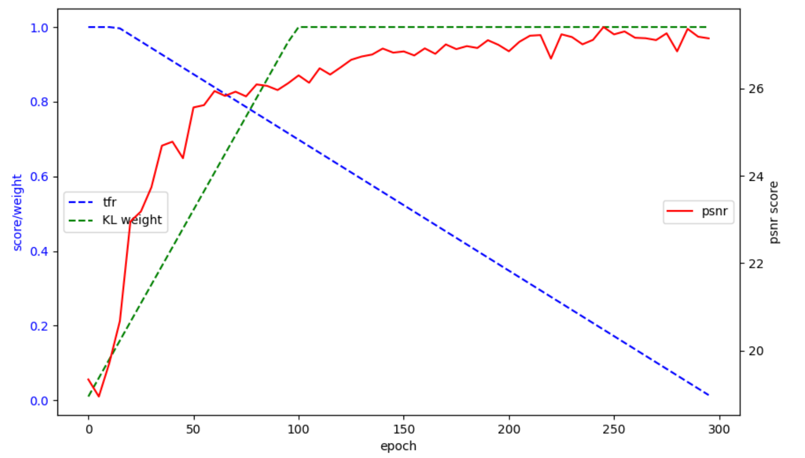
- Output the prediction at each time step (up: ground truth / down: prediction)

1. Monotonic
   
2. Cyclical
   

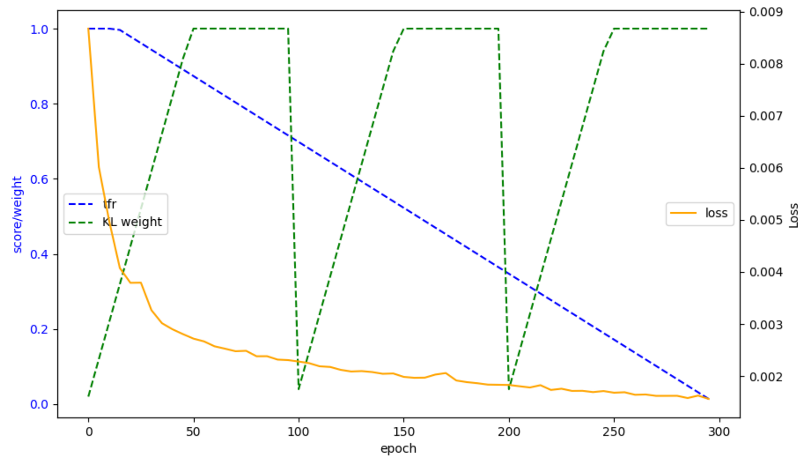
Plot the losses, average PSNR and ratios during training

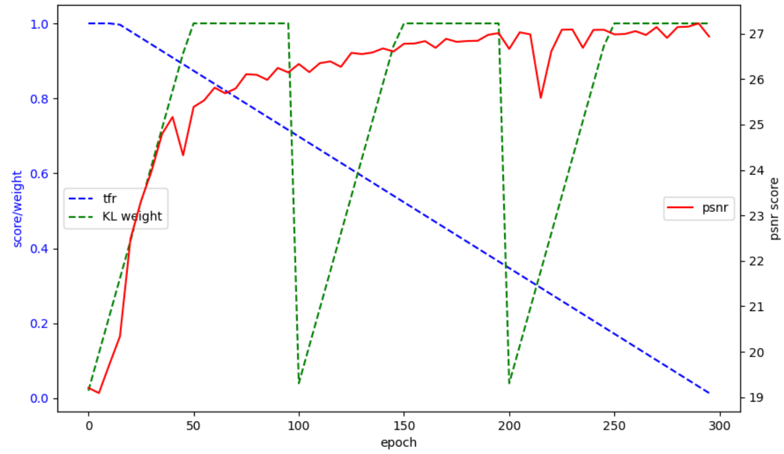
1. Monotonic





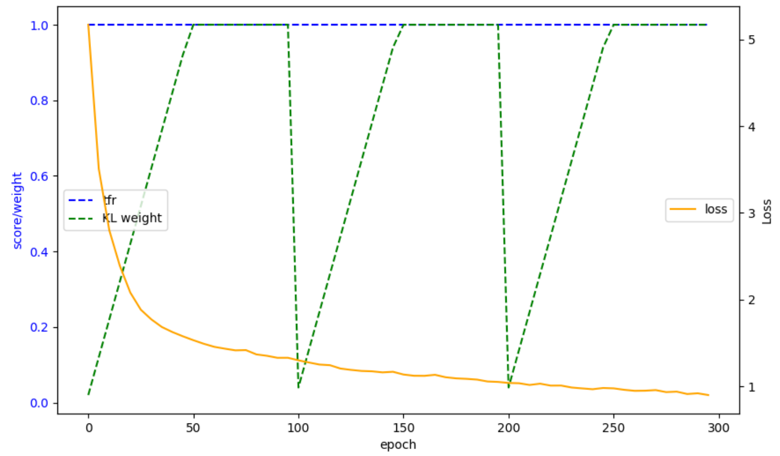
2. Cyclical

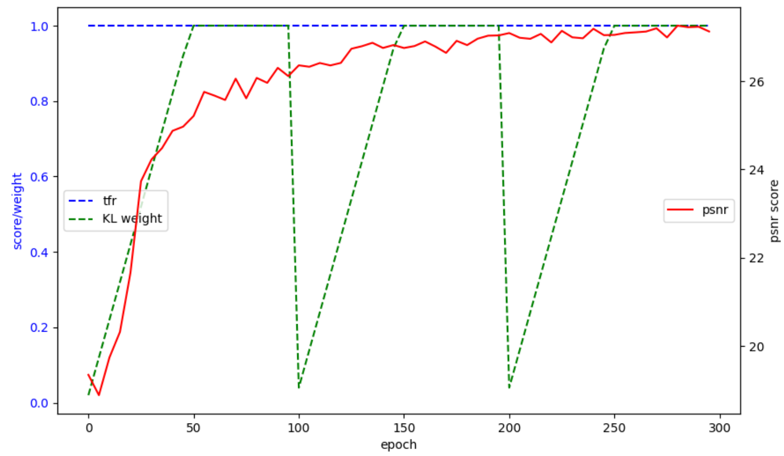


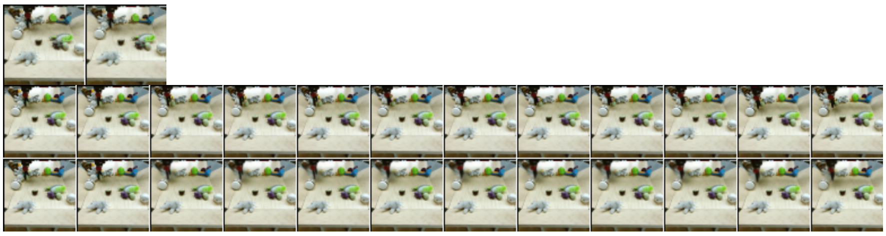


Discuss the results according to your settings

- teacher forcing ratio
  我最後設定的teacher forcing rate是從第15個epoch開始從1線性遞減到0。
  實作過程中，我也有試過teacher forcing rate從頭到尾都不下降的方式，結果如下圖，其實我覺得看起來跟上面有作遞減的沒有什麼太大的差異。







- KL weight
  這次實作了monotonic跟cyclical兩種方式，我設定的kl anneal ratio為0.5，kl anneal cycle為3，但我覺得從圖上看起來這兩種方法也沒有什麼太大的差異。

test PSNR

monotonic	cyclical	cyclical & tfr不遞減
26.93835	26.95898	27.09415

- Hyperparameter
  learning rate = 0.002
  batch size = 12
  optimizer = adam
  epoch = 300
  epoch size = 600
  random seed = 1