Lab4: Conditional VAE for Video Prediction

Student id / name: A113599 / 楊淨富

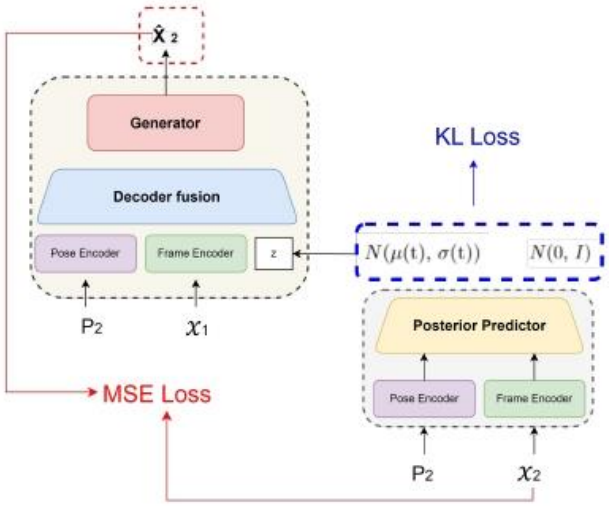
I. Introduction:

實作一個 conditional VAE model 來預測影像，即根據上一幀的資訊預測下一幀。並在最後的 testing data 上，給予第一幀，去預測接下來的 629 幀。並和 label frame(ground truth)計算出 PSNR。

在 VAE 中，主要是希望學習能夠生成 input data 的潛在變數表示(latent variable representation)，並利用 reparameterization 的技巧使得能從中隨機 sample 來進行 back propagation。

II. Implementation Details:

1. How do you write your training protocol



- 總共會有 16 幀作為 training video sequence，即 x1, x2, ..., x16。以及 16 幀作為 conditional signals，即 P1, P2, ..., P16。
- 其中第一幀是用來預測未來 15 幀的。
- 假設目前產生的下一幀為 xˆ2，則會需要
  - Pose image (P2): 即 label feature。
  - Last generated frame: x1
  - Z sample from prior distribution
    - Sample 一個 Gaussian distribution 的 tensor 作為 noise
- 產出下一幀後，就可以計算 mse，會根據 adapt_TeacherForcing 來決定是否用 ground truth(即 x2)或者用自己先前所預測的前一幀來進行比較，即:
  - mse(xˆ2, x2) or mse(xˆ2, x1)
- 但 loss = mse + kld * beta，其中:
  - kld 為 KL divergence，是一種用來衡量兩個機率分佈間差異的指標。它衡量了兩個分布間的相對熵，即一個分布相對於另一個分布的資訊損失。
- 算完 loss 後就進行 back propagrion 和更新參數:
  - loss.backward()

- self.optimizer_step()
  - 會用到 nn.utils.clip_grad_norm_(self.parameters(), 1.0)，主要是用於對模型的梯度進行裁減，防止梯度爆炸的情況。透過將梯度的大小限制在一個合理的範圍內，來緩解這個問題。

2. How do you implement reparameterization tricks

```python
class Gaussian_Predictor(nn.Sequential):
    '''
    接收圖像和label的特徵，用於預測生成圖像分布的高斯參數(均值和變異數)。
    '''
    def __init__(self, in_chans=48, out_chans=96):
        super(Gaussian_Predictor, self).__init__(
            ResidualBlock(in_chans, out_chans//4),
            DepthConvBlock(out_chans//4, out_chans//4),
            ResidualBlock(out_chans//4, out_chans//2),
            DepthConvBlock(out_chans//2, out_chans//2),
            ResidualBlock(out_chans//2, out_chans),
            nn.LeakyReLU(True),
            nn.Conv2d(out_chans, out_chans*2, kernel_size=1)
        )

    def reparameterize(self, mu, logvar):
        '''
        從均值(mu)和變異數的對數(logvar)中採樣生成潛在變量，以便進行反向傳播和優化。
        '''
        # Sample from the posterior distribution using the reparameterization trick
        std = torch.exp(logvar/2)
        eps = torch.randn_like(std) # return a same size tensor as std, sample from Normal distribution
        return mu + eps * std

    def forward(self, img, label):
        # Predict the posterior distribution parameters
        feature = torch.cat([img, label], dim=1)
        parm = super().forward(feature)
        mu, logvar = torch.chunk(parm, 2, dim=1)
        # z是模型使用reparameterization trick從給定的均值(mu)和對數變異數(logvar)中生成的潛在變數。
        # 這個潛在變數通常被視為模型在隱藏空間中的表示。
        # VAE中，z可以看作是從隱藏空間中隨機生成的一個點，用於生成數據。
        z = self.reparameterize(mu, logvar)

        return z, mu, logvar
```

- 在一些生成模型中，需要對潛在變數（latent variables）進行 sample，然後使用這些樣本生成數據。但是，直接從潛在分布中 sample 可能會使反向傳播變得困難，因為梯度無法有效地通過隨機過程進行傳播。
- Reparameterization（重參數化）:從潛在分布中生成樣本，同時允許對樣本進行反向傳播（backpropagation）以更新模型參數。作法是從一個固定的分布（通常是標準正態分布）中 sample，然後進行線性變換和平移，得到我們想要的分布。
- reparameterize()這個 function 從平均(mu)和對數變異數(logvar)中 sample，生成潛在變量(z)。這個過程可以通過將標準常態分布中的隨機樣本（eps）進行線性變換和平移得到。因為這個過程可微，所以梯度可以正確傳播，使訓練過程中可以對模型更新參數。
- 重參數化技巧的核心思想是將隨機取樣的過程分解為兩個部分，其中一部分透過模型參數來表達隨機性，而另一部分是從實際分佈中獲取數值，使得整個生成過程可以與反向傳播結合，實現模型的有效訓練和優化。

3. How do you set your teacher forcing strategy

```python
def teacher_forcing_ratio_update(self):
    if self.current_epoch % self.tfr_sde == 0:
        self.tfr -= self.args.tfr_d_step
        self.tfr = max(self.tfr, 0.0)
```

```python
# Teacher Forcing strategy
parser.add_argument('--tfr',          type=float, default=1.0,  help="The initial teacher forcing ratio")
parser.add_argument('--tfr_sde',      type=int,   default=10,   help="The epoch that teacher forcing ratio start to decay")
parser.add_argument('--tfr_d_step',   type=float, default=0.1,  help="Decay step that teacher forcing ratio adopted")
parser.add_argument('--ckpt_path',    type=str,   default=None, help="The path of your checkpoints")
```

- 一開始的 tfr ratio 是 1.0，即 100%使用 ground truth。並且每經過 tfr_sde(default = 10)個 epoch 後，tfr 就會減少 tfr_d_step(default = 0.1)。

```python
if adapt_TeacherForcing:
    mse += self.mse_criterion(out, img[i].to(out.device)) # X2_hat vs ground truth
else:
    mse += self.mse_criterion(out, decoded_frame_list[-1].to(out.device)) # X2_hat vs prev pred frame
```

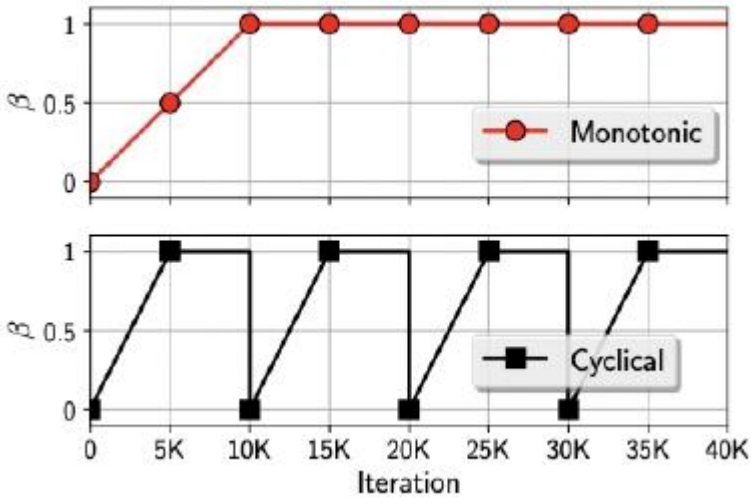
- 如果 adapt_TeacherForcing 是 True，則使用 ground truth 圖片與生成的圖片計算 MSE，即使用真實的輸入來指導模型的訓練。
- 如果 adapt_TeacherForcing 是 False，則使用先前預測的圖片與生成的圖片計算 MSE。不使用真實的輸入，而使用模型自身生成的先前圖片作為輸入，這種策略可以使模型更有能力處理自身生成的結果。

4. How do you set your kl annealing strategy

```python
class kl_annealing():
    def __init__(self, args, current_epoch=0):
        self.cur_iter = 0

        self.type = args.kl_anneal_type
        self.cycle = args.kl_anneal_cycle
        self.ratio = args.kl_anneal_ratio

        self.total_iter = args.num_epoch * args.train_vi_len
        self.beta_list = np.ones(self.total_iter)

        if(self.type == 'Cyclical'):
            self.frange_cycle_linear(self.total_iter, n_cycle=self.cycle, ratio=self.ratio)

        elif(self.type == 'Monotonic'):
            self.cycle= 1
            self.frange_cycle_linear(self.total_iter, n_cycle=self.cycle, ratio=0.25)

        elif(self.type == 'None'):
            self.beta_list = np.zeros(self.total_iter)

    def update(self):
        self.cur_iter += 1

    def get_beta(self):
        return self.beta_list[self.cur_iter]

    def frange_cycle_linear(self, n_iter, start=0.0, stop=1.0,  n_cycle=1, ratio=1):
        period = n_iter / n_cycle
        step = (stop - start) / (period * ratio)

        for c in range(n_cycle):
            v, i = start, 0
            while v <= stop and (int(i + c * period) < n_iter):
                self.beta_list[int(i + c * period)] = v
                v += step
                i += 1
```

- KL 散度(KL divergence)用於評估兩個分布之間的差異。在 VAE 中，希望學習的潛在變量分布（通常是高斯分布）與已知的標準分布（例如，標準高斯分布）之間的 KLD 盡量接近零。這意味著學習的潛在變量分布應該盡量接近標準高斯分布，這有助於更好地探索潛在空間。

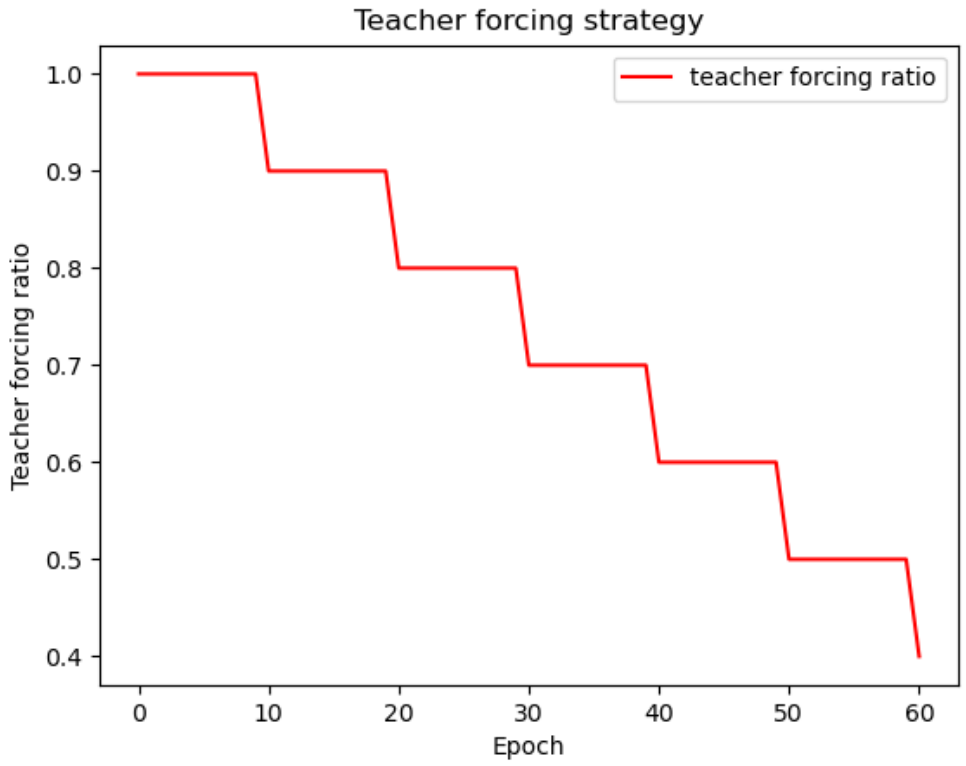
- 如果在訓練的早期就強制 KLD 為零，模型可能會過早地將潛在變量分布縮小到接近標準高斯分布，這可能會導致模型丟失有用的特徵。KL annealing 最主要就是避免這種情況。
- KL annealing 的想法是在訓練過程中逐步增加 KLD 的權重，從而逐漸將潛在變量分布限制在較接近標準高斯分布的範圍內。這可以使模型在早期學習到更多的特徵，然後在訓練後期越來越多地遵循標準高斯分布。這樣的過程有助於更好地平衡生成數據的多樣性和學習有用特徵之間的關係。



- Cyclical: 根據 cycle 數決定每個「波」，根據 ratio 決定每個 cycle 內的每一步(每次 period)應該走多少量。
- Monotonic: 全部就是一個 cycle，走到 1 時，後面的 nparray 全部都是 1。

III. Analysis & Discussion:

1. Plot Teacher forcing ratio

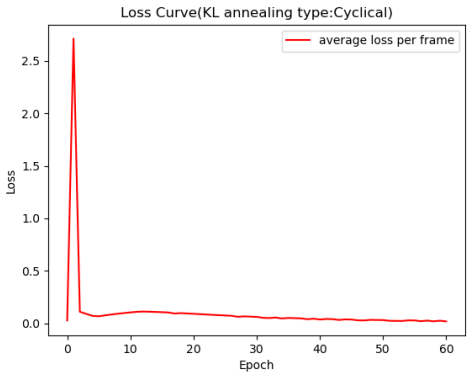


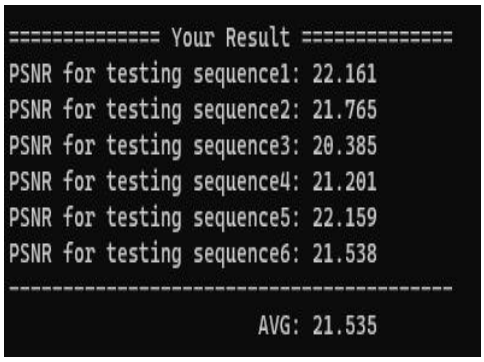
- 我的 teacher forcing ratio 初始值是 1.0，代表 100%使用 ground truth，然後每個 10epochs，則 teacher forcing ratio 減少 0.1。
- Teacher forcing ratio 剛開始需要設置比較大的初始值是因為剛開始產出的 frame 品質比較差，所以拿來計算 mse 時先以 ground truth 為主會比較容易讓 loss 穩定。

2. Plot the loss curve while training with different settings.

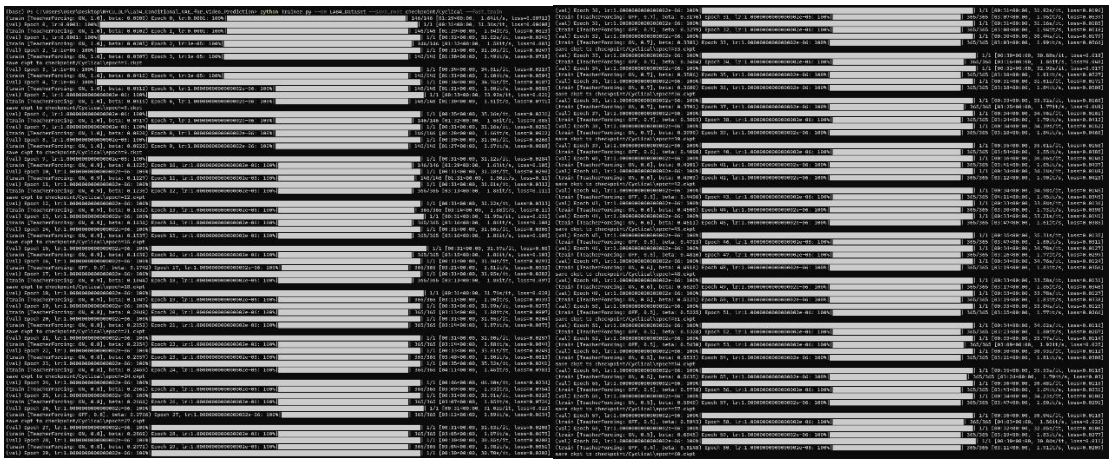
   Analyze the difference between them.

- With KL annealing (Cyclical)

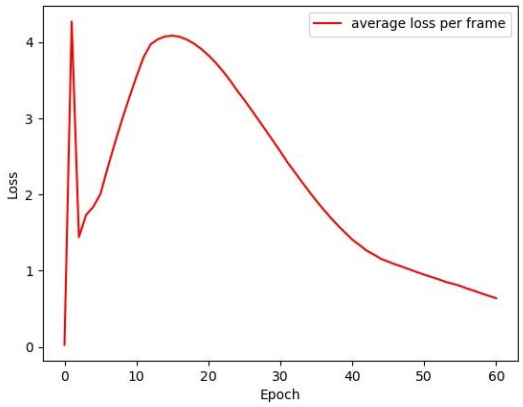




  - 我的 loss 計算方式是 average loss per frame，只有 epoch=1 時數值很高，理由可能是第一次產出的幀的品質還相當爛，但之後平均每幀的 loss 便在 0.1-0.01 左右。

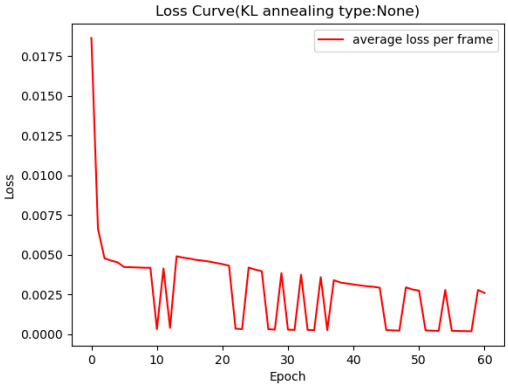


- With KL annealing (Monotonic)




```
============== Your Result ==============
PSNR for testing sequence1: 22.020
PSNR for testing sequence2: 21.741
PSNR for testing sequence3: 20.351
PSNR for testing sequence4: 21.242
PSNR for testing sequence5: 22.078
PSNR for testing sequence6: 21.498
----------------------------------------
                        AVG: 21.488
```

- Without KL annealing

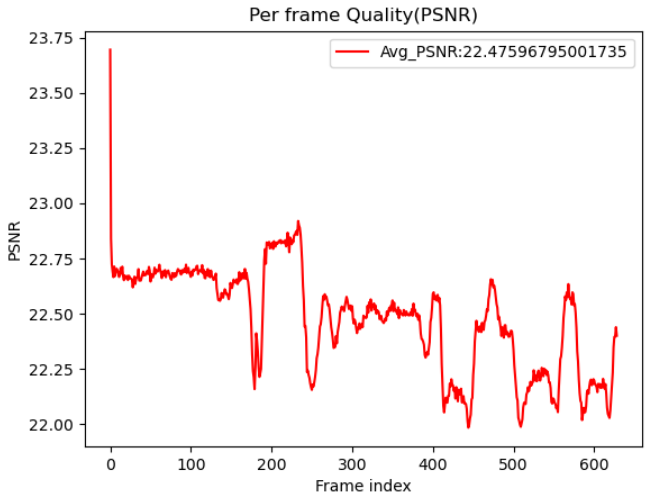



```
============== Your Result ==============
PSNR for testing sequence1: 23.212
PSNR for testing sequence2: 22.677
PSNR for testing sequence3: 21.352
PSNR for testing sequence4: 22.014
PSNR for testing sequence5: 23.476
PSNR for testing sequence6: 23.161
----------------------------------------
                        AVG: 22.648
```

- 最好的一次 PSNR(Cyclical)

- Anaylze the difference between them
  - 整體而言，Cyclical/ Monotonic/ None 三者皆在 epoch=1 時到達最高點，之後 Cyclical 是不斷在低點徘徊，Monotonic 則又再爬升至一個高點，後面才逐漸穩定減少 loss，而 None 則是有在震盪。就各自的 PSNR 而言，我的實驗結果反而是 Without KL annealing 表現最好，而 Cyclical 和 Monotonic 兩者差不多。KL annealing 並未對我的 model 產生正面的影響，有可能是我的退火參數(退火速率、最大 KLD 等等)並未調整好，也有可能是過度限制了 model 的生成，或者 epoch 太少。
  - Cyclical KL Annealing 在訓練過程中週期性地增加和降低 KLD 的權重。這可以有助於在不同階段適當地平衡失真和正規化，有助於 model 更好地適應不同的數據分佈和特徵，讓 model 更靈活。
  - Monotonic KL Annealing 從訓練開始時就增加 KLD 的權重，但不進行週期性變化。這種策略可能更穩定，但在某些情況下無法適應數據的變化。由於 KLD 的增加可能會導致模型陷入 local minima，可能需要調整相關參數來確保 training model 的穩定性。

3. Plot the PSNR-per frame diagram in validation dataset



- 這張圖的 parameters 如下:
  - KL annealing type: Cyclical
  - Epoch: 61

4. Derivate conditional VAE formula

f. Derivation of Conditional VAE (Reference ; EM algorithm L13 P.23)

- The chain rule of probability

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

Given c →

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

- We next introduce an arbitrary distribution $q(Z|c)$ on both sides and integrate over Z

$$\int q(Z|c)\log p(X|c;\theta)dZ = \int q(Z|c)\log p(X,Z|c;\theta)dZ - \int q(Z|c)\log p(Z|X,c;\theta)dZ$$

$$= \int q(Z|c)\log p(X,Z|c;\theta)dZ - \int q(Z|c)\log q(Z|c)dZ$$

$$+ \int q(Z|c)\log q(Z|c)dZ - \int q(Z|c)\log p(Z|X,c;\theta)dZ$$

$$= L(X,c,q,\theta) + KL(q(Z|c)\,||\,p(Z|X,c;\theta))$$

where

$$L(X,c,q,\theta) = \int q(Z|c)\log p(X,Z|c;\theta)dZ - \int q(Z|c)\log q(Z|c)dZ$$

$$KL(q(Z|c)\,||\,p(Z|X,c;\theta)) = \int q(Z|c)\log\frac{q(Z|c)}{p(Z|X,c;\theta)}dZ$$

- Since the KL divergence is non-negative, $KL(q\,||\,p) \geq 0$, it follows that

$$\log p(X|c;\theta) \geq L(X,c,q,\theta)$$

with equality if and only if

$$q(Z|c) = p(Z|X,c;\theta)$$

In other words, $L(X,c,q,\theta)$ is a lower bound on $\log p(X|c;\theta)$

$$L(X,c,q,\theta) = \int q(Z|c)\log p(X,Z|c;\theta)dZ - \int q(Z|c)\log q(Z|c)dZ$$

$$= \int q(Z|c)\log p(X|Z,c;\theta)dZ + \int q(Z|c)\log p(Z|c)dZ$$

$$- \int q(Z|c)\log q(Z|c)dZ$$

$$= E_{z\sim q(Z|X,c;\phi)}\log p(X|Z,c;\theta) + E_{z\sim q(Z|X,c;\phi)}\log p(Z|c)$$

$$- E_{z\sim q(Z|X,c;\phi)}\log q(Z|X,c;\theta)$$

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