DLP lab5

Conditional VAE for Video Prediction

311551170 林琨堯

Introduction

這次的 lab 需要我們實作 conditional VAE，根據之前的畫面(x_{t-1})，來預測一部影片接下來的畫面(x_t)

Derivation of CVAE

EM algo.

$$\log p(x|c;\theta) = \log p(x,z|c;\theta) - \log p(z|x,c;\theta)$$

Arbitary distribution $q(z|c)$:

$$\int q(z|c)\log p(x|c;\theta)dz = \int q(z|c)\log p(x,z|c;\theta)dz - \int q(z|c)\log p(z|x,c;\theta)dz$$

$$= \int q(z|c)\log p(x,z|c;\theta)dz - \int q(z|c)\log q(z|c)dz$$

$$= \int q(z|c)\log q(z|c)dz - \int q(z|c)\log p(z|x,c;\theta)dz$$

$$= L(x,c,q,\theta) + KL(q(z|c)||p(z|x,c;\theta))$$

$$L(x,c,q,\theta) = \int q(z|c)\log p(x,z|c;\theta)dz - \int q(z|c)\log q(z|c)dz$$

$$KL(q(z)||p(z|x,c;\theta)) = \int q(z|c)\log\frac{q(z)}{p(z|x;\theta)}dz$$

又 $L(x,c,q,\theta)$ 為 $\log p(x|c;\theta)$ ($\because$ KL divergence ≥ 0)

$$\Rightarrow L(x,c,q,\theta) = \int q(z|c)\log p(x,z|c;\theta)dz - \int q(z|c)\log q(z|c)dz$$

$$= \int q(z|c)\log p(x|z,c;\theta)dz + \int q(z|c)\log p(z|c)dz - \int q(z|c)\log q(z|c)dz$$

$$= E_{z\sim q(z|x,c;\theta')}\log p(x|z,c;\theta) + E_{z\sim q(z|x,c;\theta')}\log p(z|c) - E_{z\sim q(z|x,c;\theta')}\log q(z|x,c;\theta)$$

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

Implementation details

1. Describe how you implement your model
   - Encoder:
     利用 vgg net 實作 encoder，會 down-sample 五次，產生 1*1 的 latent code。
   - Decoder:
     同樣利用 vgg net 實作 decoder，會 up-sample 五次，讓 latent code 回到 64*64，以此預測原本 input 的樣子。
   - lstm:
     把 encoder 的 output 嵌入 latent vector，之後數入 decoder。
   - gaussian lstm:

學習 gaussian distribution，作為產生 latent code 的一部份。

- Re-parameterization trick:

```
def reparameterize(self, mu, logvar):
    std = torch.exp(0.5*logvar)
    eps = torch.randn_like(std)
    return mu + eps*std
```

- Train:
  利用 teacher forcing ratio 把 groundtruth 輸入 latent code，然後預測下一個 frame，並且會逐漸縮小 tfr，使模型朝向"使用這次預測的結果來預測下一個 frame"的方向發展。

```
def train(x, cond, modules, optimizer, kl_anneal, args):
    modules['frame_predictor'].zero_grad()
    modules['posterior'].zero_grad()
    modules['encoder'].zero_grad()
    modules['decoder'].zero_grad()
    mse_criterion = nn.MSELoss()

    # initialize the hidden state.
    modules['frame_predictor'].hidden = modules['frame_predictor'].init_hidden()
    modules['posterior'].hidden = modules['posterior'].init_hidden()
    mse = 0
    kld = 0
    use_teacher_forcing = True if random.random() < args.tfr else False
    x = x.permute(1, 0, 2, 3 ,4)
    cond = cond.permute(1, 0, 2)
    h_seq = [modules['encoder'](x[i]) for i in range(args.n_past+args.n_future)]
    for i in range(1, args.n_past + args.n_future):
        h_target = h_seq[i][0]
        if args.last_frame_skip or i < args.n_past:
            h, skip = h_seq[i-1]
        else:
            h = h_seq[i-1][0]

        z_t, mu, logvar = modules['posterior'](h_target)
        h_pred = modules['frame_predictor'](torch.cat([cond[i-1], h, z_t], 1))
        x_pred = modules['decoder']([h_pred, skip])
        mse += mse_criterion(x_pred, x[i])
        kld += kl_criterion(mu, logvar, args)
        if not use_teacher_forcing :
            h_seq[i] = modules['encoder'](x_pred)

    beta = kl_anneal.get_beta()
    loss = mse + kld * beta
    loss.backward()

    optimizer.step()

    return loss.detach().cpu().numpy() / (args.n_past + args.n_future), mse.detac
```

2. Describe the teacher forcing

- 使 tfr 線性衰減，當訓練到最後一個 epoch 時，tfr 會到達整體的最小值。

```
if epoch >= args.tfr_start_decay_epoch:
    ### Update teacher forcing ratio ###
    slope = (1.0 - args.tfr_lower_bound) / (args.niter - args.tfr_start_decay_epoch)
    tfr = 1.0 - (epoch - args.tfr_start_decay_epoch) * slope
    args.tfr = min(1, max(args.tfr_lower_bound, tfr))
print(f'Teacher ratio: {args.tfr}')
```

Results and discussion (30%)

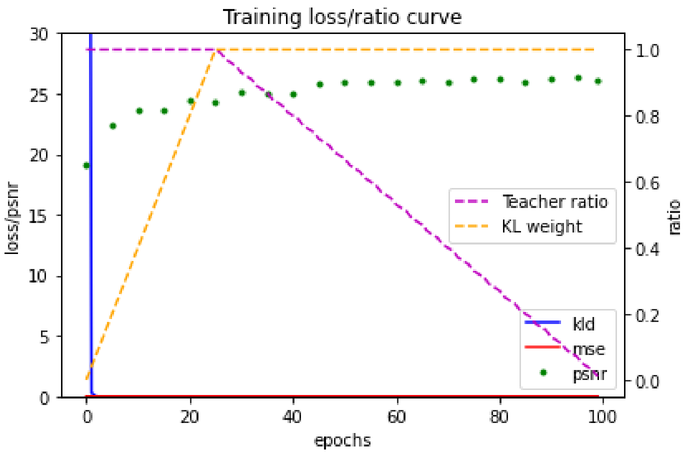
- Show your results of video prediction (10%)
  - (a) Make videos or gif images for test result (select one sequence)

    Non-cyclical test

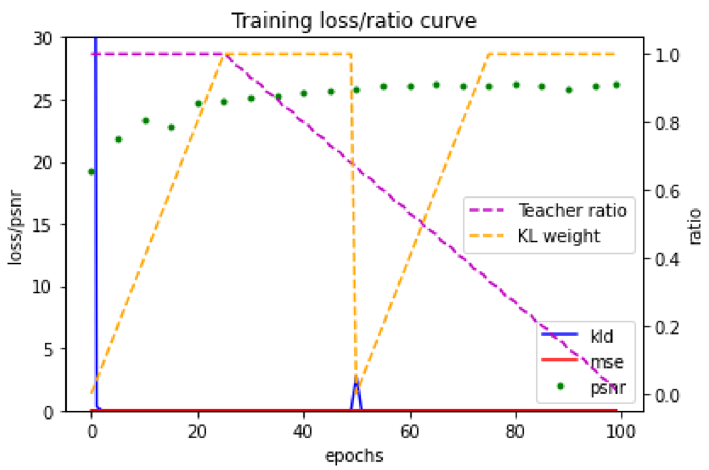
    Cyclical test
  - (b) Output the prediction at each time step (select one sequence)

- Plot the KL loss and PSNR curves during training (5%)

  Train without cyclical:



Train with cyclical (2 cycle):



- Discuss the results according to your setting of teacher forcing ratio, KL weight, and learning rate.
  - Hyper-parameter:
    epoch=100, learning rate=0.002, batch size=24, epoch size = 500
  - Teacher forcing ratio:
    前 25 個 epochs，tfr 保持為 1，目的是用 groundtruth 讓 model 建立正確的 encoder-decoder，之後才以線性的方式逐漸降低 tfr，達到”讓 model 使用 predicted frame 來預測下一個 frame”的目標。
  - KL annealing:
    一開始，因為 KL loss 太大，會誤導 loss，所以把 beta 設為 0，使 loss 專注在 mse loss 上。而 KL annealing 週期性的微調 model，使其得到比較好的 psnr。

Without cyclical

```
epoch: 96 | loss: 0.00165 | mse loss: 0.00162 | kld loss: 0.00003 | tfr: 0.05
epoch: 97 | loss: 0.00166 | mse loss: 0.00163 | kld loss: 0.00003 | tfr: 0.04
epoch: 98 | loss: 0.00171 | mse loss: 0.00168 | kld loss: 0.00003 | tfr: 0.03
epoch: 99 | loss: 0.00163 | mse loss: 0.00160 | kld loss: 0.00003 | tfr: 0.01
====================== validate psnr = 26.10038 ========================
```

Cycle = 2

```
epoch: 96 | loss: 0.00161 | mse loss: 0.00161 | kld loss: 0.00001 | tfr: 0.05
epoch: 97 | loss: 0.00183 | mse loss: 0.00181 | kld loss: 0.00001 | tfr: 0.04
epoch: 98 | loss: 0.00169 | mse loss: 0.00168 | kld loss: 0.00000 | tfr: 0.03
epoch: 99 | loss: 0.00162 | mse loss: 0.00161 | kld loss: 0.00000 | tfr: 0.01
====================== validate psnr = 26.25150 ========================
```