Lab4 CVAE For Video Prediction

1. Introduction

   這次 LAB 透過 CVAE 的方法實作 video prediction, Dataset 使用的是 bair robot pushing, 約有 44000 個 sequence 的 robot pushing motions, 每個 sequence 包含 30 個 frames,在訓練 VAE 的 encoder, decoder 時會增加額外的 condition 來做訓練並觀察能否有更好的結果。

   此外透過 reparameterization trick, teacher forcing, KL annealing,透過調整後兩者的參數來觀察訓練結果

2. Derivation of CVAE

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

$$\Rightarrow \int q(Z|C)\log p(X|C;\theta)\,dz$$

$$= \int q(Z|C)\log p(X,Z|C;\theta)\,dz - \int q(Z|C)\log P(Z|X,C;\theta)\,dz$$

$$\Rightarrow \log P(X|C;\theta) = L(X,q,\theta|C) + KL(q(Z|C) \,\|\, P(Z|X,C;\theta))$$

where $L(X,q,\theta|C) = \int q(Z|C)\log p[X,Z|C;\theta)\,dz - \int q(Z|C)\log q(Z|C)\,dz$

$$KL(q(Z|C) \,\|\, P(Z|X,C;\theta)) = \int q(Z|C)\log\frac{q(Z|C)}{P(Z|X,C;\theta)}\,dz$$

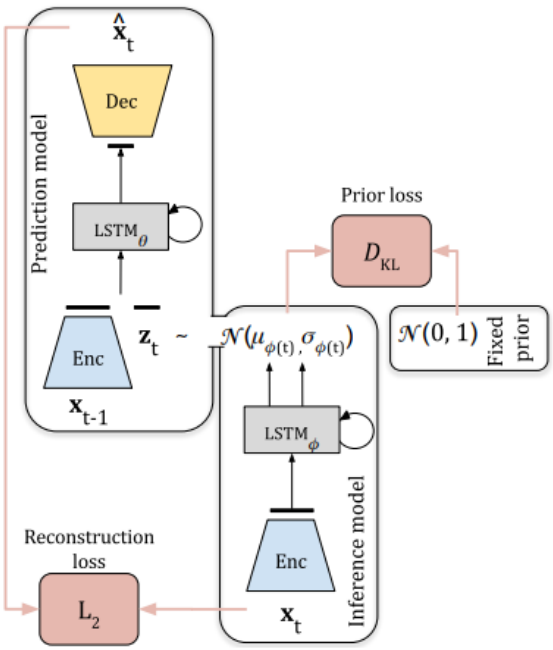
$q(Z|C) \Rightarrow q(Z|X,C;\theta')$

$$\log p(X|C;\theta) - KL(q(Z|X,C;\theta') \,\|\, P(Z|X,C;\theta)) = L(X,q,\theta|C)$$

$$L(X,q,\theta|C) = E_{Z\sim q(Z|X,C;\theta')}\log P(X|Z,C;\theta) + E_{Z\sim q(Z|X,C;\theta')}\log P(Z|C) - E_{Z\sim q(Z|X,C;\theta)}\log q(Z|X,C;\theta)$$

$$= E_{Z\sim q(Z|X,C;\theta')}\log P(X|Z,C;\theta) - KL(q(Z|X,C;\theta') \,\|\, P(Z|C)) \quad \#$$

3. Implementation details
   a. Describe how you implement your model.



(a)

模型架構如 paper[3]中 Stochastic Video Generation with a Learned Prior 的圖，左上 prediction model(也就是 frame predictor)是原本的 sample code 的 lstm 加上額外的 condition，如果有 condition 就將其 concat z,再一起 forward

Condition 的部分修改如下(constructor)

```
class lstm(nn.Module):
    def __init__(self, input_size, output_size, hidden_size, n_layers, batch_size, device, conditional = True):
        super(lstm, self).__init__()
        self.device = device
        self.conditional = conditional
        if self.conditional:
            self.input_size = input_size + 12
        else:
            self.input_size = input_size
        self.output_size = output_size
        self.hidden_size = hidden_size * 2
        self.batch_size = batch_size
        self.n_layers = n_layers
        self.embed = nn.Linear(input_size, hidden_size)
        self.embed_cond = nn.Linear(7, hidden_size)
        self.lstm = nn.ModuleList([nn.LSTMCell(hidden_size, hidden_size) for i in range(self.n_layers)])
        self.output = nn.Sequential(
                nn.Linear(hidden_size, output_size),
                nn.BatchNorm1d(output_size),
                nn.Tanh())
        self.hidden = self.init_hidden()
```

forward 的部分修改如下(concat embedded and embeddedcond)

```
def forward(self, input, cond = None):
    embedded = self.embed(input)
    embeddedcond = self.embed_cond(cond)

    h_in = torch.cat((embedded, embeddedcond), dim = 1)
    for i in range(self.n_layers):
        self.hidden[i] = self.lstm[i](h_in, self.hidden[i])
        h_in = self.hidden[i][0]

    return self.output(h_in)
```

Reparameterization trick

參考 sample code [sample_code]

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

z = z_mean + sigma*epsilon where sigma = exp(z_logvar/2)

Dataset 取得 seq 和 csv 的內容, seq 的部分是利用 PIL 套件來做讀取, Reshape, csv 的部分則是使用 pandas

b. Describe the teacher forcing

一. Main idea:

最初為了解決 RNN 在訓練迭代前期的預測能力非常弱，常常不能給出好的生成結果，因為前面的某個 unit 產生很差的結果必會影響後面一大堆的 unit 的學習成效。而 teacher forcing 就是在學習時直接使用 ground truth 而非使用上一個 state 的輸出做為下一個 state 的對應輸入。

二. Benefit:

使用 teacher forcing 的好處能夠在訓練時矯正模型的預測，避免序列生成的過程誤差進一步放大，能夠加快模型收斂速度，使模型訓練時更加快速平穩

三. Drawback:

Teacher forcing 依賴標籤數據，在訓練過程中，模型會有較好的效果，但在測試的時候因為無法得到 ground truth，若測試 dataset 及訓練 dataset 的分布相差太多的話(cross-domain)，模型的 performance 反而會變很差

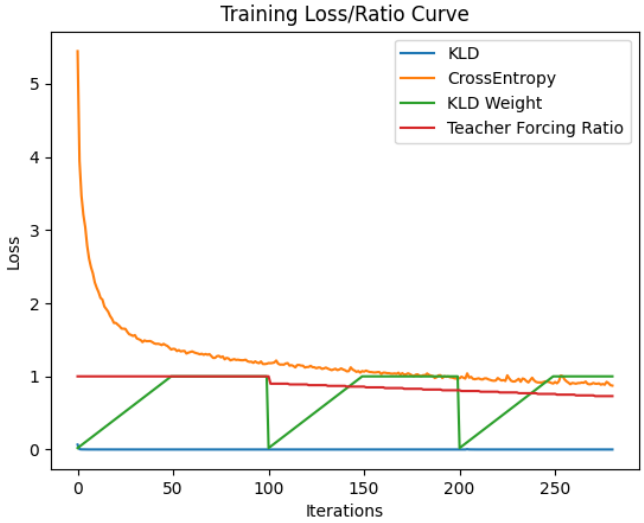
直觀上我的理解: 老師死板的丟答案給學生，學生反而失去獨立思考能力,若是看過答案的題目分數好而且很快就會(訓練過程快)，一遇到沒看過答案的題目反而考超很爛

4. Result and Discussion

a. Show your results of video prediction

A. Output the prediction at each time step

b. Plot the losses ratios.



c. Average PSNR

```
AVG PSNR: 26.567917361680376
Best Validate PSNR: 26.859208290384647
```

(Train 完之後發現忘記存用截圖的)

d. Discuss the results according to your setting of teacher forcing ratio, KL weight, and learning rate.

我的 learning rate 設成 0.002, teacher forcing 是 1.0, KL weight 設成 0.0001, 使用 cyclical 的時候 PSNR 可以達到 26.85 左右,
調整 KL weight 跟 cyclical 的參數僅會小小影響結果
然而調整 tfr 到 0.8 或 0.9 卻會大大影響結果, PSNR 掉到 18 以下, 可以看的出來 teacher forcing 對於 RNN 來說還是很重要的一個 method。