NCTU DLP
Lab5-Conditional sequence-to-sequence VAE

TA 陳璽存

April 27, 2021

Outline

- Lab Objective
- Important Date
- Lab Description
- Scoring Criteria

Lab Objective

- In this lab, you need to implement a conditional seq2seq VAE for English tense conversion.
- Tense conversion (4 tenses)
  - E.g. ‘access’ to ‘accessing’, or ‘accessed’ to ‘accesses’
- Generative model
  - Gaussian noise + tense -> access, accesses, accessing, accessed

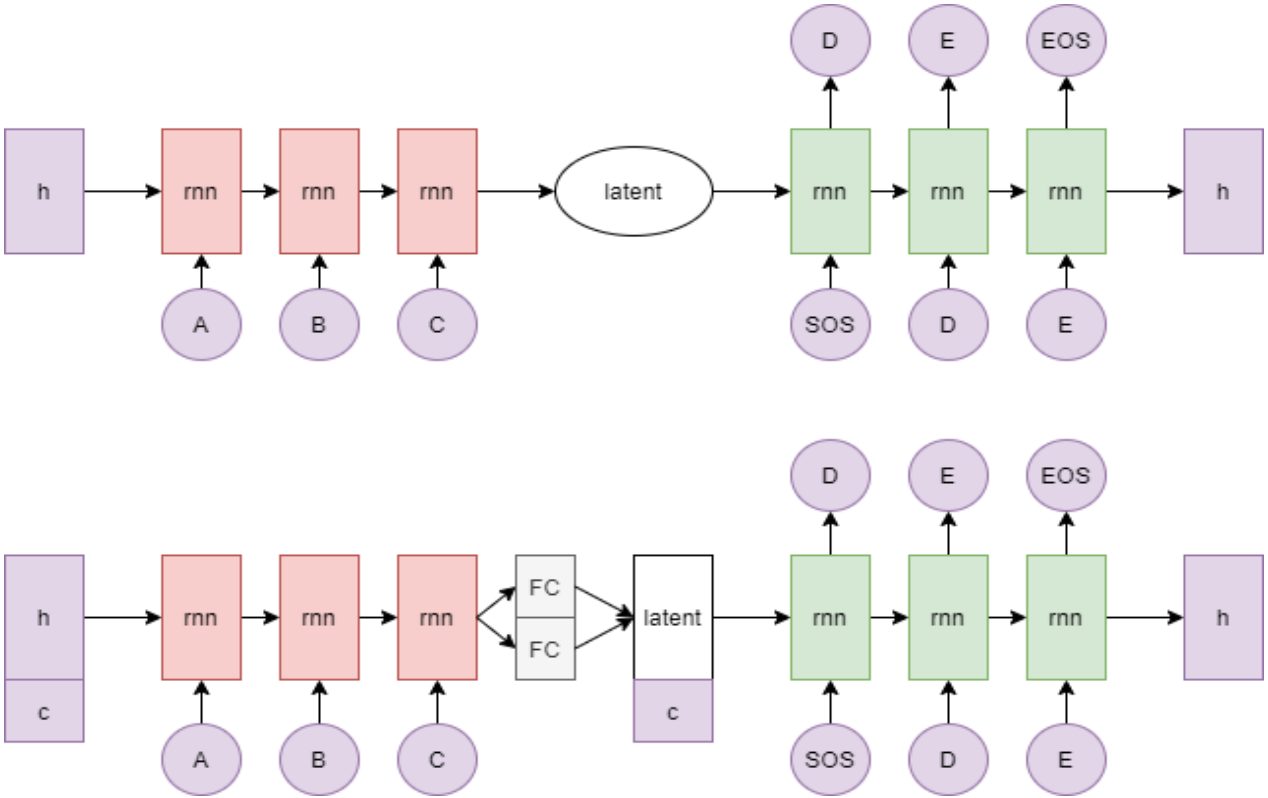
Important Date

- Experiment Report Submission Deadline: 5/11 11:59 a.m
- Demo date: 5/11
- Zip all files in one file
  - Report (.pdf)
  - Source code
- name it like 「DLP_LAB4_yourID_name.zip」
    - ex: 「DLP_LAB4_309551009_陳璽存 .zip」

Lab Description

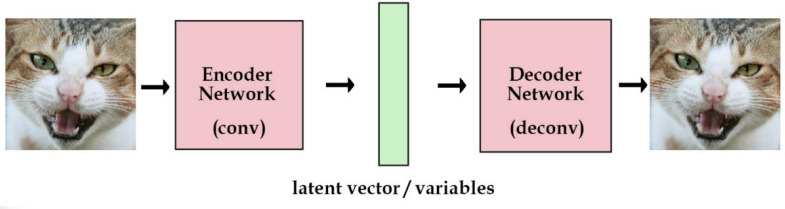
- To understand CVAE
  - Reparameterization trick
  - Log variance
  - KL lost annealing
  - Condition

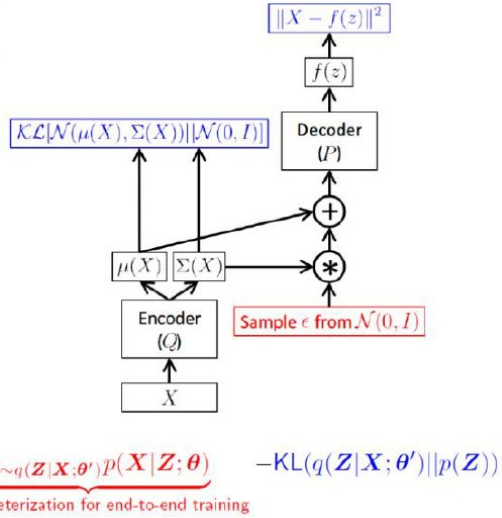
Lab Description - architecture



Lab Description - VAE

- VAE objective: reconstruction and generation



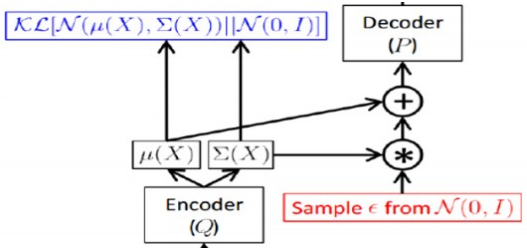


$$\mathcal{L}(\mathrm{X}, \mathrm{q}, \theta) = E_{Z\sim q(Z|X;\phi)} \log p(X|Z;\theta) - KL(q(Z|X;\phi)||p(Z))$$

where $q(Z|X;\phi)$ is considered as encoder and $p(X|Z;\theta)$ as decoder.

Lab Description - CVAE

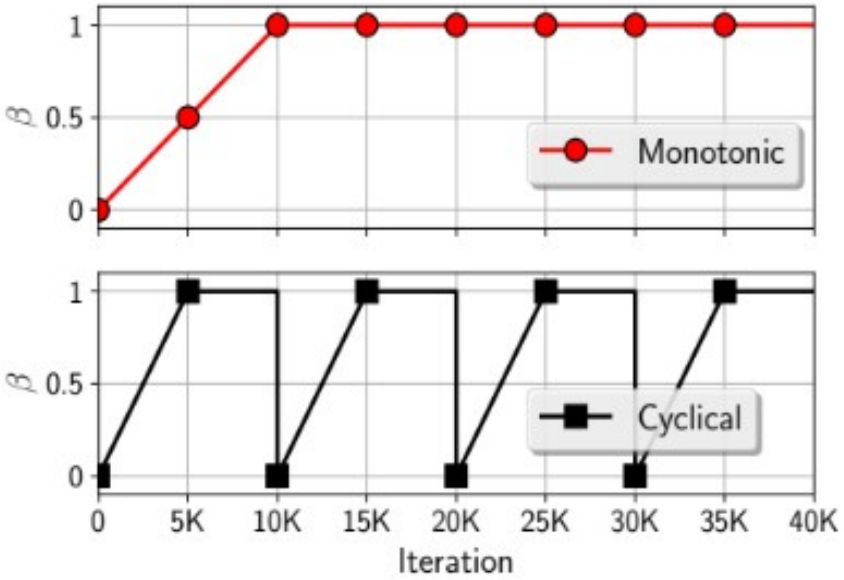
- Reparameterization trick



- Log variance
  - Output should be log variance (not variance)
- Condition
  - Simply concatenate to the hidden_0 and z
  - Embed your condition to high dimensional space (or simply use one-hot)

Lab Description - CVAE

- KL cost annealing
  - Initially set your KL weight to 0
  - Maximum value is 1



Lab Description – Other details

- The encoder and decoder must be implemented by LSTM.
- You should not adopt attention mechanism
- The loss function is nn.CrossEntropyLoss().
- The optimizer is SGD
- Adopt BLEU-4 score function in NLTK.
  - Average 10 testing scores
- Adopt Gaussian_score() to compute the generation score
  - Random sample 100 noise to generate 100 words with 4 different tenses (totally 400 words)
  - 4 words should exactly match the training data.

Lab Description – Requirements

- Modify encoder, decoder, and training functions
- Implement evaluation function, dataloader, and reparameterization trick.
- Adopt teacher-forcing and kl loss annealing in your training processing.
- Plot the crossentropy loss, KL loss, and BLEU-4 score curve

```
['bear', 'bears', 'bearing', 'bear']
['sit', 'sits', 'intervening', 'intervened']
['characterize', 'characterizes', 'characting', 'characterized']
['chide', 'chides', 'chiding', 'chided']
['cite', 'cites', 'citing', 'cited']
['explain', 'festoons', 'festoring', 'festooned']
['back', 'backs', 'backsliding', 'backslid']
['cide', 'cides', 'ciding', 'cided']
['survey', 'surrenders', 'surveying', 'surrendered']
['wet', 'wets', 'wetting', 'chew']
Gaussian score : 0.35
```

```
input:flared
target:flare
prediction:flare

input:functioning
target:function
prediction:furnish

input:functioning
target:functioned
prediction:functioned

input:healing
target:heals
prediction:heals

Average BLEU-4 score : 0.8319248477410198
```

Hints

- Training method
  - Input the word with the tense and the output should also be the same word.
  - Convert each character to a number (dictionary)
- Model weights
  - Strongly recommend you save your model weights during training
- Teacher forcing ratio and KL weight
  - Influential to the performance of model
  - You can first set your KL weight to 0 to see whether your model works.

Scoring Criteria

- Report (50%)
  - Introduction(5%)
  - Derivation of CVAE(5%)
  - Implementation details(15%)
    - Describe how you implement your model. (e.g. dataloader, encoder, decoder, etc)
    - Specify the hyperparameters (KL weight, teacher forcing ratio, etc.)
    - Notice: You must prove that your text generation is produced by Gaussian noise (paste/screenshot your code)
  - Results and discussion(25%)
    - Plot the loss and KL loss curve while training and discuss the results. (5%)
    - Plot the BLEU-4 score of your testing data while training and discuss the result. (20%)
    - Notice: This part mainly focuses on your discussion, if you simply just paste your results, you will get a low score

Scoring Criteria

- Demo(50%)
  - Capability of tense conversion on testing data. (20%)
    - score >= 0.7 ---- 100%
    - 0.7 > score >= 0.6 ---- 90%
    - 0.6 > score >= 0.4 ---- 80%
    - score < 0.4 ---- 0%
  - Capability of word generation. (Gaussian noise + tense) (10%)
    - score = Gaussian_score() (100 words with 4 tenses)
    - score >= 0.3 ---- 100%
    - 0.3 > score >= 0.2 ---- 90%
    - 0.2 > score >= 0.05 ---- 80%
    - Otherwise ---- 0%
  - Questions (20%)

Implement Hints

- Sampling
  - torch.randn(*size)
    ex : torch.randn(*size) * exp(0.5 * var) + mean
  - torch.randn_like(input)

- Pytorch seq2seq tutorial
  https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html