VRAE

Y.-B. Hong

Task

- Batch size code 일반화
- VRAE sentence generation
- 코드 정리

Contents

- VRAE
- Char-rnn 과 비교

    → In model

    → In train

    → In inference

- VRAE sentence generation

Contents

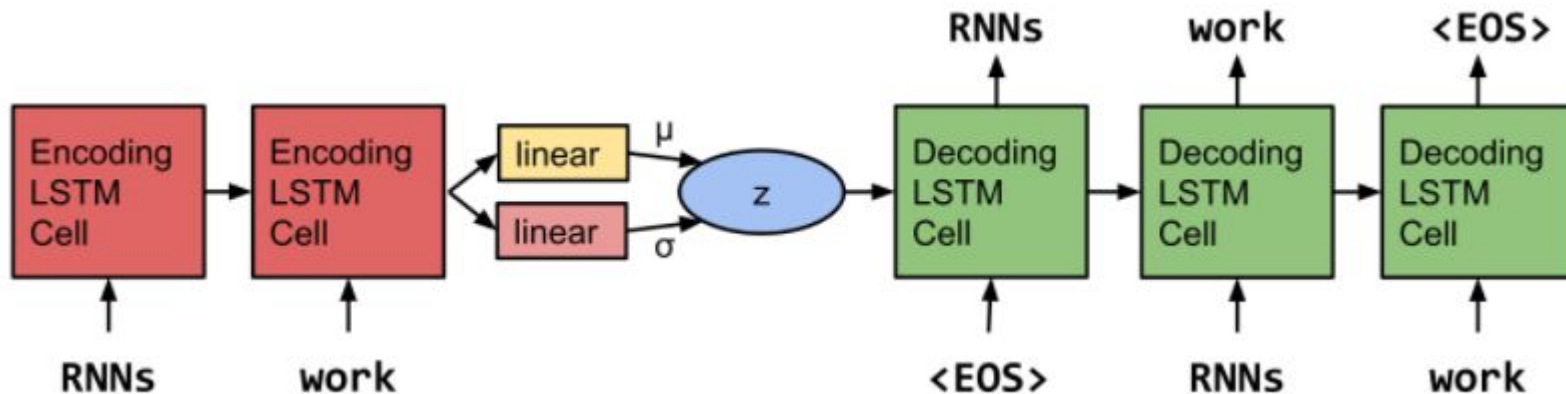- <mark>VRAE</mark>

- Char-rnn 과 비교

    → In model

    → In train

    → In inference

- VRAE sentence generation

VRAE

- 표준 반복 신경망 언어 모델(RNNLM)은 한 번에 한 단어씩 문장을 생성하며 명시적인 전역 문장 표현에서 작동하지 않는다.
- 본 연구에서는 전체 문장의 분산 잠재 표현을 통합하는 RNN-based variational autoencoder generative model을 소개하고 연구한다.

Mar/12/2021

Contents

- VRAE
- <mark>Char-rnn 과 비교</mark>

    → In model

    → In train

    → In inference

- VRAE sentence generation

Char-rnn 과 비교 (In model)

- VRAE

    → Embedding layer

    → Packed_pad sequence

    → RNN, LSTM, GRU layer

    → Pad_pack sequence

    → Linear function

- Char rnn

    → Embedding layer

    → Packed_pad sequence

    → RNN, LSTM, GRU layer

    → Pad_pack sequence

    → Linear function

Char-rnn 과 비교 (In train)

- VRAE

    → **(NLL loss + KL_loss * KL_weight)/batch_size**

    → **Mask 사용**

    → **Clip_grad_norm 사용 X**

- Char rnn

    → NLL loss만 사용

    → Mask 사용 X

    → Clip_grad_norm 사용

Char-rnn 과 비교 (In Inference)

- VRAE

  → **Random z1, z2**

  → **numpy.linspace(Start = 0, 100, nums)**

  → **array([0, 25, 50, 75, 100])**

  → **numpy.linspace(Start = z1, z2, step + 2)**

  ⇒ **Z**

- Char rnn

  → Multinomial

  → 상위 proportion을 나타내는 Char 중 하나를 고름

Char-rnn 과 비교 (In Inference of VRAE)

- VRAE

  → model에서

  → **hidden** = self.latent2hidden(z)

  ⇒ self.latent2hidden = nn.Linear(latent size, hidden size * hidden_factor)

  ⇒ self.hidden_factor = (2 if bidirectional else 1) * num_layers

  → embedding layer 통과 후

  → output, hidden = self.decoder_rnn(embed(embedding layer통과), **hidden**)

  ⇒ self.decoder_rnn = rnn(embedding_size, hidden_size, num_layers=num_layers, bidirectional=self.bidirectional, batch_first=True)

Contents

- VRAE
- Char-rnn 과 비교

  → In model

  → In train

  → In inference

- VRAE sentence generation

Sequence generation of VRAE

```
python3 inference.py -c $CHECKPOINT -n $NUM_SAMPLES
```

the company said it will be n with the exception of the company
but the company said it will be n with the exception of the company ' s shares outstanding
but the company said that the company ' s n n and n n
but the company ' s n n in the past two years ago
but the company ' s n n in the past two years ago
but in the past few years ago that the company ' s n n
but in the past few years ago that they were n't disclosed
but in the past few years ago that they were n't disclosed
but in a statement that they were n't aware of the $ n million in the past few weeks
but in a statement that they were n't paid by the end of the past few weeks