Discreteness in Generative Models

--- Discrete Sequence Generation

Hao Dong

Peking University
Discreteness in Generative Models  
– Discrete Sequence Generation

• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN
• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN
• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
• Introduction
  • RNN Language Modelling
    • Generating Sentences from a Continuous Space
    • Recap: Inverse RL vs. GAN
  • GAN+RL
    • SeqGAN
    • RankGAN
    • MaskGAN
  • GANs
    • Adversarial Text Generation without Reinforcement Learning
    • GANs for Sequences Generation with the Gumbel-softmax
RNN Language Modelling

• Recap: Recurrent Neural Network

Multiple inputs and multiple outputs
RNN Language Modelling

• Synchronous Many-to-Many

• MLE training

• The output of each step is equal to the input of its next step.

For testing, input “this,” output the entire sentence
RNN Language Modelling

- Limitations
  
  1. No latent space – Representation learning
  
  2. Exposure bias – Long sentence generation problem
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inversed RL vs. GAN
• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN
• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
Generating Sentences from a Continuous Space

• Why Continuous Space

i went to the store to buy some groceries.
i store to buy some groceries.
i were to buy any groceries.
horses are to buy any groceries.
horses are to buy any animal.
horses the favorite any animal.
horses the favorite favorite animal.
horses are my favorite animal.
Generating Sentences from a Continuous Space

• VAE Approach

Input

Hidden state

Output

one to one

one to many

many to one

many to many

asynchronous (Seq2Seq)

Multiple data inputs and multiple data outputs

Generating Sentences from a Continuous Space.
Generating Sentences from a Continuous Space

• VAE Approach: Training
Generating Sentences from a Continuous Space

• VAE Approach: Testing
Generating Sentences from a Continuous Space

• Application

Impute missing words within sentences.

---

but now, as they parked out front and owin stepped out of the car, he could see _ _ _ _ _
True: that the transition was complete. RNNLM: it, ” i said. VAE: through the driver’s door.

you kill him and his _ _
True: men.

RNNLM: ”
VAE: brother.

not surprising, the mothers dont exactly see eye to eye with me _ _ _
True: on this matter.

RNNLM: , i said.
VAE: , right now.

---

Generating Sentences from a Continuous Space.
Generating Sentences from a Continuous Space

• Application

Results from the mean of the posterior distribution and samples from that distribution.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>MEAN</th>
<th>SAMP. 1</th>
<th>SAMP. 2</th>
<th>SAMP. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>we looked out at the setting sun .</td>
<td>they were laughing at the same time .</td>
<td>ill see you in the early morning .</td>
<td>i looked up at the blue sky .</td>
<td>it was down on the dance floor .</td>
</tr>
<tr>
<td>i went to the kitchen .</td>
<td>i went to the kitchen .</td>
<td>i went to my apartment .</td>
<td>i looked around the room .</td>
<td>i turned back to the table .</td>
</tr>
<tr>
<td>how are you doing ?</td>
<td>what are you doing ?</td>
<td>“ are you sure ?</td>
<td>what are you doing ?</td>
<td>what are you doing ?</td>
</tr>
</tbody>
</table>

Generating Sentences from a Continuous Space.
Generating Sentences from a Continuous Space

• Application

Interpolation between two points in the VAE manifold.

“i want to talk to you .”
“i want to be with you .”
“i do n’t want to be with you .”
i do n’t want to be with you .
she did n’t want to be with him .

he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .
Generating Sentences from a Continuous Space

- Limitation

Difficult to generate long sentences
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN

• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN

• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
Recap: Inverse RL vs. GAN

- Reinforcement Learning and Imitation Learning
Recap: Inverse RL vs. GAN

- Reinforcement Learning and Imitation Learning

- Why?

Difficult and even impossible to define reward functions for many environments
Recap: Inverse RL vs. GAN

• Reinforcement Learning and *Imitation Learning*
  
  • Behaviour Cloning
  
  • Inverse RL
Recap: Inverse RL vs. GAN

• Reinforcement Learning and Imitation Learning

• Behaviour Cloning == Supervised Learning

• Given expert’s demonstrations: \((s_1, a_1), (s_2, a_2), \ldots, (s_N, a_N)\)

• Supervised training:
Recap: Inverse RL vs. GAN

• Reinforcement Learning and Imitation Learning

• Behaviour Cloning == Supervised Learning

• Problem

  • Expert only samples from limited observation (states)
    (Solution: Dataset Aggregation)

  • If machine has limited capacity, it may choose the wrong behavior to copy

  • Assume the training and testing data distributions are the same
Recap: Inverse RL vs. GAN

- Reinforcement Learning and Imitation Learning

- Inverse Reinforcement Learning
Recap: Inverse RL vs. GAN

- Reinforcement Learning and Imitation Learning

- Inverse Reinforcement Learning

\[ T = (s_1, a_1, s_2, a_2, \ldots, s_N, a_N) \]

...
Recap: Inverse RL vs. GAN

- Reinforcement Learning and Imitation Learning

- Inverse Reinforcement Learning

\[ \mathcal{F} = \{s_1, a_1, s_2, a_2, \ldots, s_t, a_t\} \]

...
Recap: Inverse RL vs. GAN

- **Inverse RL: Training**

  \[ \sum_{n=1}^{N} R(T_n) > \sum_{n=1}^{N} R(\hat{T}_n) \]

  **Goal:** The expert is always the best

  \[ T = (s_1, a_1, s_2, a_2, \ldots, s_t, a_t) \]
Recap: Inverse RL vs. GAN

- **GAN**
  - Dataset: $x_1 x_2 x_3 \ldots x_N$
  - Generator: $\hat{x}_1 \hat{x}_2 \hat{x}_3 \ldots \hat{x}_N$
  - Discriminator: $\text{real, fake}$

- **Inverse RL**
  - Expert $\pi$: $T_1 T_2 T_3 \ldots T_N$
  - Actor $\hat{\pi}$: $\hat{T}_1 \hat{T}_2 \hat{T}_3 \ldots \hat{T}_N$
  - Reward Function: $\text{real, fake}$
  - “Common” Reinforcement Learning: $\hat{T}$ are not differentiable .. so update the actor via RL
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN

• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN

• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
**SeqGAN:** Sequence Generative Adversarial Nets with Policy Gradient

- **Motivation**
  
  MLE has exposure bias

  MLE tends to move to the mean when there is uncertainty
  “this is a ...” “this is one” ⇐ “a” or “one” which one is correct?

  MLE-free sentence generation using adversarial networks could be better

- **Challenge**

  Sentences are discrete

  Sentence generation is not differentiable

**SeqGAN:** Sequence Generative Adversarial Nets with Policy Gradient. Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu. AAAI. 2016.
SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient

- **Inverse RL**
  - Expert $\pi$  $\mathcal{T}_1 \mathcal{T}_2 \mathcal{T}_3 \ldots \mathcal{T}_N$
  - Actor $\hat{\pi}$  $\hat{\mathcal{T}}_1 \hat{\mathcal{T}}_2 \hat{\mathcal{T}}_3 \ldots \hat{\mathcal{T}}_N$
  - “Common” Reinforcement Learning
  - Expert demonstrations

- **SeqGAN**
  - Dataset $t_1 t_2 t_3 \ldots t_N$
  - Generator $\hat{t}_1 \hat{t}_2 \hat{t}_3 \ldots \hat{t}_N$
  - Discriminator $\hat{t}$  $\hat{\mathcal{D}}$  $\hat{\mathcal{T}}_1 \hat{\mathcal{T}}_2 \hat{\mathcal{T}}_3 \ldots \hat{\mathcal{T}}_N$

$\hat{\mathcal{T}}$ are not differentiable .. so update the actor via RL

$\hat{t}$ are not differentiable .. so update the generator via RL
**SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient**

- **Method Details**

  - The generator is a LSTM or GRU language model
  - Pretrain the generator using MLE before RL training for better initialization
  - The discriminator is a CNN network

---

SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient

- Method Details

```
Algorithm 1 Sequence Generative Adversarial Nets

Require: generator policy $G_\theta$; roll-out policy $G_\beta$; discriminator $D_\phi$; a sequence dataset $S = \{X_{1:T}\}$
1: Initialize $G_\theta, D_\phi$ with random weights $\theta, \phi$.
2: Pre-train $G_\theta$ using MLE on $S$.
3: $\beta \leftarrow \theta$.
4: Generate negative samples using $G_\theta$ for training $D_\phi$.
5: Pre-train $D_\phi$ via minimizing the cross entropy.
6: repeat
7: \hspace{1em} for $g$-steps do
8: \hspace{2em} Generate a sequence $Y_{1:T} = (y_1, \ldots, y_T) \sim G_\theta$.
9: \hspace{2em} for $t$ in $1 : T$ do
10: \hspace{3em} Compute $Q(a = y_t; s = Y_{1:t-1})$ by Eq. (4).
11: \hspace{2em} end for
12: \hspace{1em} end for
13: \hspace{1em} Update generator parameters via policy gradient Eq. (8).
14: \hspace{1em} for $d$-steps do
15: \hspace{2em} Use current $G_\theta$ to generate negative examples and combine with given positive examples $S$.
16: \hspace{2em} Train discriminator $D_\phi$ for $k$ epochs by Eq. (5).
17: \hspace{2em} end for
18: $\beta \leftarrow \theta$.
19: until SeqGAN converges.
```

SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient

• Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Human score</th>
<th>p-value</th>
<th>BLEU-2</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>0.4165</td>
<td>0.0034</td>
<td>0.6670</td>
<td>&lt; 10^{-6}</td>
</tr>
<tr>
<td>SeqGAN</td>
<td><strong>0.5356</strong></td>
<td></td>
<td><strong>0.7389</strong></td>
<td></td>
</tr>
<tr>
<td>Real data</td>
<td>0.6011</td>
<td></td>
<td>0.746</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Chinese poem generation performance comparison.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>BLEU-3</th>
<th>p-value</th>
<th>BLEU-4</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>0.519</td>
<td>&lt; 10^{-6}</td>
<td>0.416</td>
<td>0.00014</td>
</tr>
<tr>
<td>SeqGAN</td>
<td><strong>0.556</strong></td>
<td></td>
<td><strong>0.427</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Obama political speech generation performance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>BLEU-4</th>
<th>p-value</th>
<th>MSE</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>0.9210</td>
<td>&lt; 10^{-6}</td>
<td>22.38</td>
<td>0.00034</td>
</tr>
<tr>
<td>SeqGAN</td>
<td><strong>0.9406</strong></td>
<td></td>
<td><strong>20.62</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Music generation performance comparison.
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN

• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN

• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
RankGAN: Adversarial Ranking for Language Generation

• Motivation

the existing GANs restrict the discriminator to be a binary classifier, limiting their learning capacity for tasks that need to synthesize output with rich structures such as natural language descriptions

so ... we use a Ranker as the discriminator ... improved SeqGAN
RankGAN: Adversarial Ranking for Language Generation

• **Method**

Generator hopes the Ranker rank the fake sentence to the top
... a better reward function for RL training

RankGAN: Adversarial Ranking for Language Generation.
Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, Ming-Ting Sun. NeurIPS. 2017
RankGAN: Adversarial Ranking for Language Generation

• Method

\[
\min_{\theta} \max_{\phi} \mathcal{L}(G_\theta, R_\phi) = \mathbb{E}_{s \sim P_h} \left[ \log R_\phi(s | U, C^-) \right] + \mathbb{E}_{s \sim G_\theta} \left[ \log(1 - R_\phi(s | U, C^+)) \right]
\]

- $U$ is a set of real data for reference
- $C^-$ is a set of randomly sampled fake data
- $C^+$ is a set of randomly sampled real data
- $p_h$ and $G_\theta$ are real and fake distribution

The “Rank” of a data is the similarity with the reference $U$

$y_s$ and $y_u$ are the feature vector of $U$ and $s$ (s is a single data point)

The similarity score of the input sequence $s$ given a reference $u$:

\[
\alpha(s | u) = \text{cosine}(y_s, y_u) = \frac{y_s \cdot y_u}{\|y_s\| \|y_u\|}
\]

The ranking score for the input sequence $s$ given a comparison set $C$:

\[
P(s | u, C) = \frac{\exp(\gamma \alpha(s | u))}{\sum_{s' \in C} \exp(\gamma \alpha(s' | u))}
\]

The ranking score for the input sentence $s$:

\[
R_\phi(s | U, C) = \mathbb{E}_{u \in U} \left[ P(s | u, C) \right]
\]

RankGAN: Adversarial Ranking for Language Generation.
Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, Ming-Ting Sun. NeurIPS. 2017
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN

• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN

• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
MaskGAN: Better Text Generation via Filling in the ____

- **Motivation**

SeqGAN and RankGAN use Policy Gradient for RL training,

... thus we can only have a single reward for the whole sentence

Could we have a reward for each word?

MaskGAN uses Actor Critic for RL training.
MaskGAN: Better Text Generation via Filling in the ____

• Method

Both Generator and Discriminator are Seq2Seq

---

**MaskGAN: Better Text Generation via Filling in the ____**

William Fedus, Ian Goodfellow, Andrew M. Dai. ICLR 2018
MaskGAN: Better Text Generation via Filling in the ____

- **Results**

  - **Conditional Sentence Generation == Inpainting**

    | Ground Truth | MaskGAN                                      | MaskMLE                                    |
    |--------------|----------------------------------------------|--------------------------------------------|
    | Pitch Black was a complete shock to me when I first saw it back in 2000 In the previous years I | Pitch Black was a complete shock to me when I first saw it back in 1979 I was really looking forward | Black was a complete shock to me when I first saw it back in 1969 I live in New Zealand |

  - **Unconditional Sentence Generation == All words are masked**

    | MaskGAN | **Positive**: Follow the Good Earth movie linked Vacation is a comedy that credited against the modern day era yarns which has helpful something to the modern day s best It is an interesting drama based on a story of the famed |

*MaskGAN: Better Text Generation via Filling in the ____.* William Fedus, Ian Goodfellow, Andrew M. Dai. ICLR 2018
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN
• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN
• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
Adversarial Text Generation without Reinforcement Learning

• **Motivation**

  RL training is too slow ...

• **Method: Stage 1**

  The generator is a common Seq2Seq
Adversarial Text Generation without Reinforcement Learning

• Method: Stage 2
• Introduction
  • RNN Language Modelling
  • Generating Sentences from a Continuous Space
  • Recap: Inverse RL vs. GAN
• GAN+RL
  • SeqGAN
  • RankGAN
  • MaskGAN

• GANs
  • Adversarial Text Generation without Reinforcement Learning
  • GANs for Sequences Generation with the Gumbel-softmax
GANs for Sequence Generation with Gumbel-softmax

• **Motivation**
  
  Train like a normal GAN

GANs for Sequences of Discrete Elements with the Gumbel-softmax Distribution.

GANs for Sequence Generation with Gumbel-softmax

- **SeqGAN**
  - Dataset
  - Generator
    - LSTM language model
    - $t_1 \, t_2 \, t_3 \ldots \, t_N$
    - $\hat{t}_1 \, \hat{t}_2 \, \hat{t}_3 \ldots \, \hat{t}_N$
  - Discriminator
    - Policy Gradient
    - LSTM encoder
  - real sentences

- **Gumbel-softmax GAN**
  - Dataset
  - Generator
    - LSTM decoder
    - $z \rightarrow$
    - Gumbel-softmax
    - $\hat{t}_1 \, \hat{t}_2 \, \hat{t}_3 \ldots \, \hat{t}_N$
  - Discriminator
    - LSTM encoder
  - real sentences

GANs for Sequences of Discrete Elements with the Gumbel-softmax Distribution.
GANs for Sequence Generation with Gumbel-softmax

• Limitation
  
  • Gumbel-softmax’s gradient is too large ... poor performance

GANs for Sequences of Discrete Elements with the Gumbel-softmax Distribution.
Discreteness in Generative Models
– Discrete Sequence Generation

• Introduction
  • RNN Language Modelling
  • *Generating Sentences from a Continuous Space* 2016
  • Recap: Inverse RL vs. GAN
• GAN+RL
  • SeqGAN 2017
  • RankGAN 2017
  • MaskGAN 2018
• GANs
  • Adversarial Text Generation without Reinforcement Learning 2018
  • *GANs for Sequences Generation with the Gumbel-softmax* 2016
Thanks