SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient

Lantao Yuy, Weinan Zhangy, Jun Wangz, Yong Yuy

AAAI 2017

Kuhwan Jeong [1]

[1] Department of Statistics, Seoul National University

April, 2018

Introduction

- In GAN a discriminative net D learns to distinguish whether a given data instance is real or not, and a generative net G learns to confuse D by generating high quality data.

- Authors consider the sequence generation procedure as a sequential decision making process.

- The generative model is treated as an agent of reinforcement learning.
  - STATE : generated tokens so far
  - ACTION : next token to be generated
  - REWARD : output of a discriminator D

Sequence Generative Adversarial Nets

- G_θ produces a sequence $Y = (y_1, \ldots, y_T)$.
- In timestep t, the state is $y_{1:(t-1)}$ and the action is the next y_t to select.
- The policy model $G_\theta(y_t|y_{1:(t-1)})$ is stochastic, whereas the state transition is deterministic after an action has been chosen.
- $D_\phi(Y)$ is a probability indicating how likely a sequence Y is from real sequence data.
- D_ϕ is trained by providing positive examples from the real sequence data and negative examples from the synthetic sequences generated from G_θ.

Sequence Generative Adversarial Nets

- The discriminator only provides a reward value for a complete sequence.

- To evaluate the reward for an intermediate state, they apply Monte Carlo search with a roll-out policy G_θ to sample the unknown last $T-t$ tokens.

- The action-value function of a sequence is

$$Q_{\theta,\phi}(s = y_{1:(t-1)}, a = y_t) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y^{(n)}) & \text{for } t < T, \\ D_\phi(y_{1:t}) & \text{for } t = T. \end{cases}$$

- Given $y_{1:T}$, the object function is

$$J(\theta) = \sum_{t=1}^{T} \sum_{a\in\mathcal{Y}} G_\theta(a|y_{1:(t-1)}) Q_{\theta,\phi}(y_{1:(t-1)}, a).$$

Sequence Generative Adversarial Nets

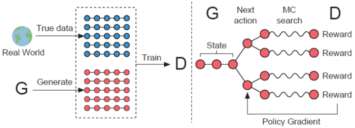

Figure 1: The illustration of SeqGAN. Left: D is trained over the real data and the generated data by G. Right: G is trained by policy gradient where the final reward signal is provided by D and is passed back to the intermediate action value via Monte Carlo search.

Algorithm 1 Sequence Generative Adversarial Nets

Require: generator policy G_θ; roll-out policy G_β; discriminator D_ϕ; a sequence dataset $\mathcal{S} = \{X_{1:T}\}$

Initialize G_θ, D_ϕ with random weights θ, ϕ.
Pre-train G_θ using MLE on $\mathcal{S}$
$\beta \leftarrow \theta$
Generate negative samples using G_θ for training D_ϕ
Pre-train D_ϕ via minimizing the cross entropy
repeat
    for g-steps **do**
        Generate a sequence $Y_{1:T} = (y_1, \dots, y_T) \sim G_\theta$
        for t in $1 : T$ **do**
            Compute $Q(a = y_t; s = Y_{1:t-1})$ by Eq. (4)
        end for
        Update generator parameters via policy gradient Eq. (8)
    end for
    for d-steps **do**
        Use current G_θ to generate negative examples and combine with given positive examples $\mathcal{S}$
        Train discriminator D_ϕ for k epochs by Eq. (5)
    end for
    $\beta \leftarrow \theta$
until SeqGAN converges

- At the beginning of the training, use the MLE to pre-train G_θ.
- After that, the generator and discriminator are trained alternatively.
- RNNs are used for the generator, and CNN is used for discriminator.

Experiments

Table 1: Sequence generation performance comparison. The p-value is between SeqGAN and the baseline from T-test.

Algorithm	Random	MLE	SS	PG-BLEU	SeqGAN
NLL	10.310	9.038	8.985	8.946	**8.736**
p-value	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	

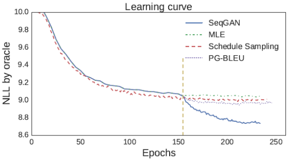

Figure 2: Negative log-likelihood convergence w.r.t. the training epochs. The vertical dashed line represents the end of pre-training for SeqGAN, SS and PG-BLEU.

Table 2: Chinese poem generation performance comparison.

Algorithm	Human score	p-value	BLEU-2	p-value
MLE	0.4165	0.0034	0.6670	$< 10^{-6}$
SeqGAN	**0.5356**		**0.7389**	
Real data	0.6011		0.746	

Table 3: Obama political speech generation performance.

Algorithm	BLEU-3	p-value	BLEU-4	p-value
MLE	0.519	$< 10^{-6}$	0.416	0.00014
SeqGAN	**0.556**		**0.427**	

Table 4: Music generation performance comparison.

Algorithm	BLEU-4	p-value	MSE	p-value
MLE	0.9210	$< 10^{-6}$	22.38	0.00034
SeqGAN	**0.9406**		**20.62**	