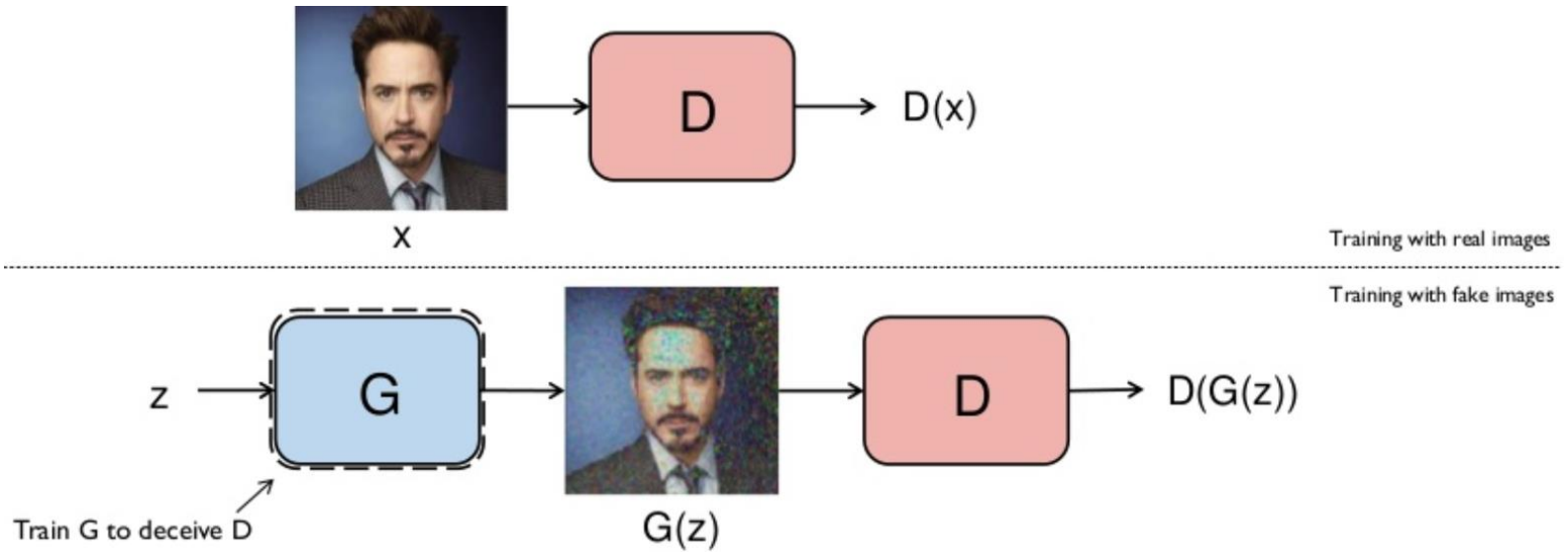
Contents

Unit 01 | GAN



https://www.youtube.com/watch?v=odpjk7_tGY0

GAN 의 objective 함수

Sample x from real data distribution

Sample latent code z from Gaussian distribution

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

D should maximize v(D,Z)

Maximum when D(x)=1

Maximum when D(G(z))=0

https://www.youtube.com/watch?v=odpjk7_tGY0

GAN 의 objective 함수

G is independent of this part

Sample latent code z from Gaussian distribution

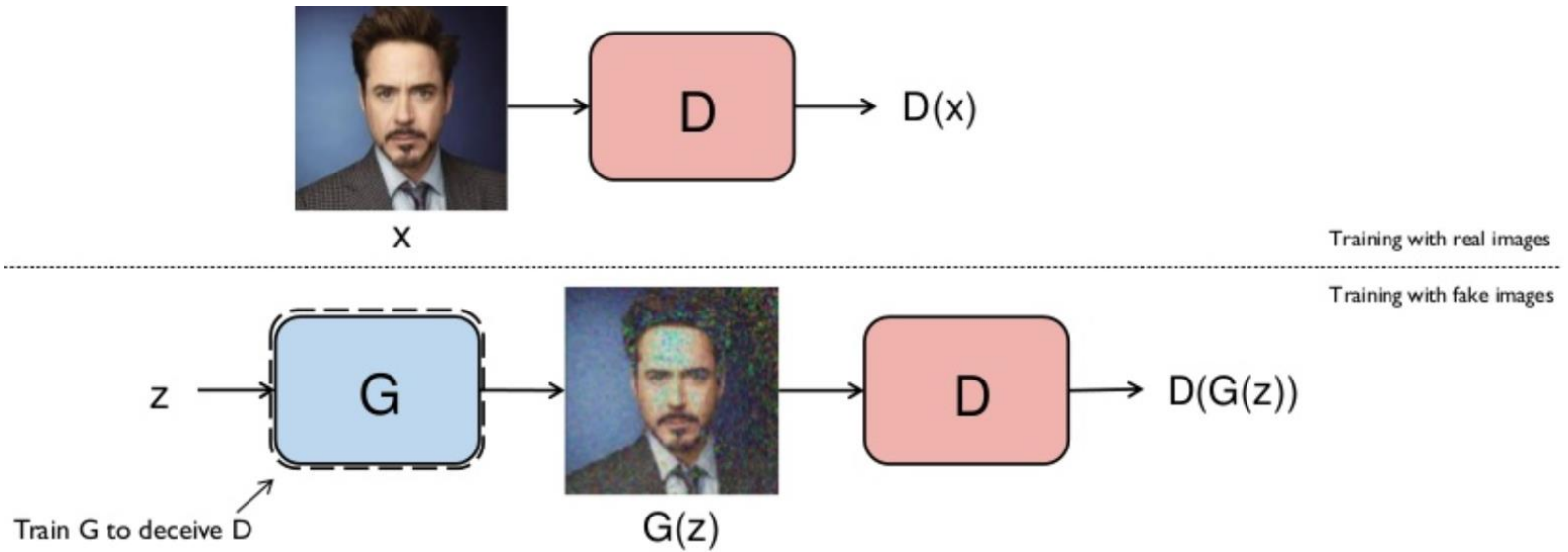
$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

G should minimize v(D,Z)

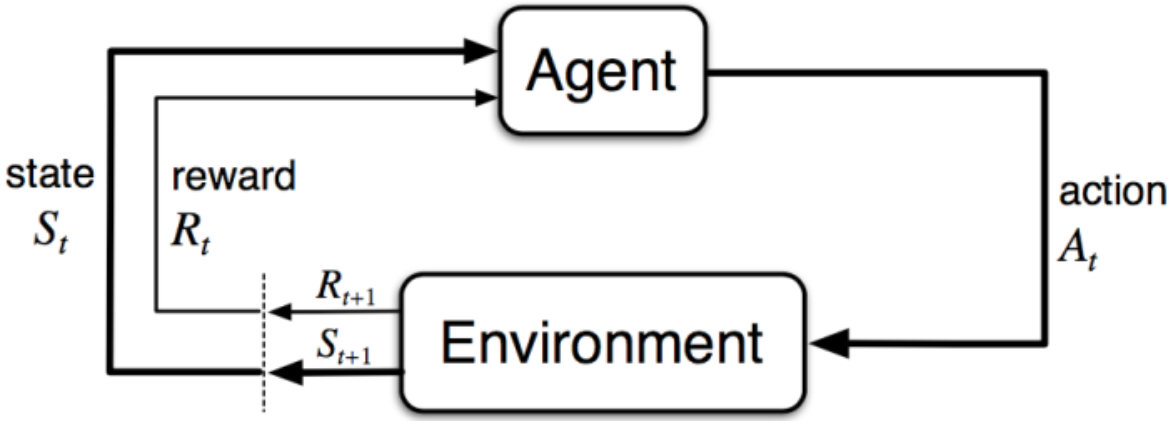
Minimum when D(G(z))=1

https://www.youtube.com/watch?v=odpjk7_tGY0

Unit 01 | GAN



https://www.youtube.com/watch?v=odpjk7_tGY0



Objective : Finding an optimal policy which maximizes the expected sum of rewards

https://www.youtube.com/watch?v=dw0sHzE1oAc

State-Value Function (미래 보상들의 총합,γ :discount factor)

$$V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s]$$

Action-Value Function

$$Q_\pi(s,a) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s, A_t = a$$

GAN의 한계점

- G로 부터 생성된 output이 D에서 G로 넘어갈 때 gradient 업데이트에 어려움이 있다.
- D는 완전한 문장에 대해서만 판단할 수 있다.

⇒ seqGAN으로 해결

- G를 강화학습의 stochastic policy로 정의하고 바로 gradient policy update 가능하게 함으로써 generator differentiation problem 우회한다.
- 완전한 문장에 대한 D의 판단 점수가 RL의 reward가 되며, MC-search를 통해 중간 ' state-action 단계로 reward 전달.

RNN을 이용한 접근법 : previous observed tokens이 주어졌을 때, training sequence의 각 true token의 log predictive likelihood 최대화 하는 학습 방법

$$\max_{\theta} \frac{1}{|D|} \sum_{Y_{1:T} \in D} \sum_{t} \log[G_{\theta}(y_t|Y_{1:t-1})]$$

-> 하지만 MLE 접근법은 inference 단계에서 exposure bias 문제가 발생한다.

-> 하지만 MLE 접근법은 inference 단계에서 exposure bias 문제가 발생한다.
: model이 sequence를 반복적으로 생성할 때 예측된 token들로 부터 다음 token을 예측하는데 이 때 예측된 token들이 training 데이터에 존재 하지 않을 수 있다.

Training

Update the model as follows:

$$\max_{\theta} \mathbb{E}_{Y \sim p_{\text{true}}} \sum_{t} \log G_\theta(y_t | Y_{1:t-1})$$

The **real** prefix

Inference

When generating the next token y_t , sample from:

$$G_\theta(\hat{y}_t | \hat{Y}_{1:t-1})$$

The **guessed** prefix

GAN을 이용한 sequence generation의 문제점

1. GAN은 continuous data 생성을 위해 디자인된 모델이므로 discrete token으로 구성된 sequence generation에 적용하기 어렵다.
   : D의 gradient loss로 G는 조금씩 generated value를 바꾼다. 이 때 limited dictionary space에서 slight change와 대응하는 token이 아마 존재 하지 않을 것이기 때문에 direct gradient가 불가능

2. GAN은 완전한 문장에 대해 score/loss를 줄 수 있다.
   : 부분적으로 생성된 현재 sequence와 완전한 미래 sequence의 score 균형을 맞추기 어렵다.

Unit 02 | Introduction

⇒ 2가지 문제를 해결하기 위해서

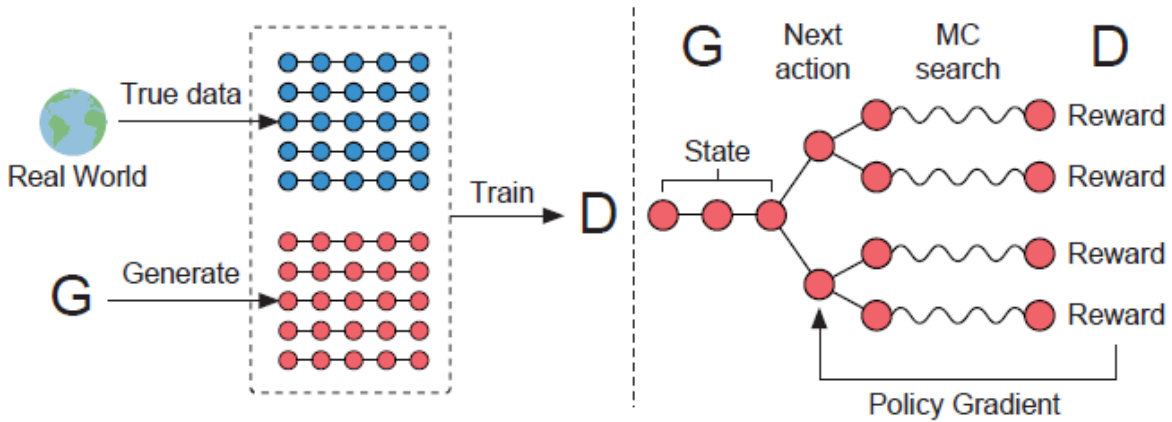
- Sequence generation procedure을 sequential decision making process로 두고 G(generative model)를 강화학습의 agent로 정의한다. State는 generated token $Y_{1:t-1}$ =$(y_1, \cdots, y_{t-1})$, action 은 next token to be generated(y_t), reward= $D_\emptyset(Y_{1:T})$ 로 두었다.

- D를 이용해 sequence 를 평가하고 G의 학습의 평가 지표로 사용.

- GAN에서 discrete output 에 대해 gradient 가 G로 전달되지 못하는 문제점 해결을 위해 G를 stochastic parameterized policy로 정의 하고, MC-search 를 policy gradient에 이용해 state-action 추정. 직접적으로 policy 인 G를 학습시켜 기존 GAN에서 발생하는 differentiation difficulty 피한다.

Unit 03 | Sequence Generative Adversarial Nets

Real world structed data가 주어졌을 때 강화학습을 기반으로 candidate tokens에 속하는 y에 대해 $Y_{1:T} = (y_1, \cdots, y_t)$, 를 생성하는 G_θ와 G가 생성한 완전한 문장 $Y_{1:T}$ 가 real sequence data 일 확률을 출력하는 $D_\emptyset$ 의 parameter 학습



SeqGAN via policy Gradient

1. Objective Function of generative model

한 문장 내에서 reward가 없다면 G는 start state s_0에서 완전한 문장에 대한 reward (R_T)를 최대화 해서 sequence generation 할 것임. 이 때 reward는 $D_\emptyset$로부터 온 것.

$$J(\theta) = \mathbb{E}[R_T|s_0, \theta] = \sum_{y_1 \in \mathcal{Y}} G_\theta(y_1|s_0) \cdot Q_{D_\phi}^{G_\theta}(s_0, y_1), \qquad (1)$$

Action-value function

$$Q_{D_\phi}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\phi(Y_{1:T}). \qquad (2)$$

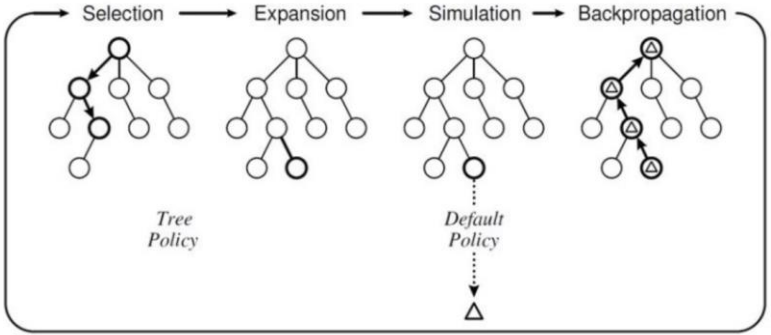
SeqGAN via policy Gradient

⇒ 중간 state의 Q-function을 평가하기 위해서 roll-out policy(G_B, 모르는 $T-t$ 개의 $token$들을 $sampling$) 를 활용한 MC-search를 사용한다.

⇒ 즉 MC-search를 통해 unknown last T-t tokens를 sampling하고 그것을 바탕으로 $D_{\emptyset}(Y_{1:t-1})$ 를 계산해 중간의 reward를 얻는 과정을 통해 이전 token과 관련되고 future reward를 최대화하는 token을 생성 할 수 있도록 학습한다.

SeqGAN via policy Gradient

Monte-Carlo Tree Search란?
의사 결정을 위한 체험적 탐색 알고리즘으로, 어떻게 움직이는 것이 가장 유망한 것인가를 분석하면서 검색 공간에서 무작위 추출에 기초한 탐색 트리를 확장하는데 중점을 둔다.



SeqGAN via policy Gradient

roll-out policy(G_B)를 기반으로 N번 samplin된 N-time MC-search 표현 식

$$\left\{Y_{1:T}^1, \ldots, Y_{1:T}^N\right\} = \mathrm{MC}^{G_\beta}(Y_{1:t}; N), \qquad (3)$$

이를 반영한 **Objective Function of Generator**

$$Q_{D_\phi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) = \qquad (4)$$

$$\begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n),\ Y_{1:T}^n \in \mathrm{MC}^{G_\beta}(Y_{1:t}; N) & \text{for} \quad t < T \\ D_\phi(Y_{1:t}) & \text{for} \quad t = T, \end{cases}$$

SeqGAN via policy Gradient

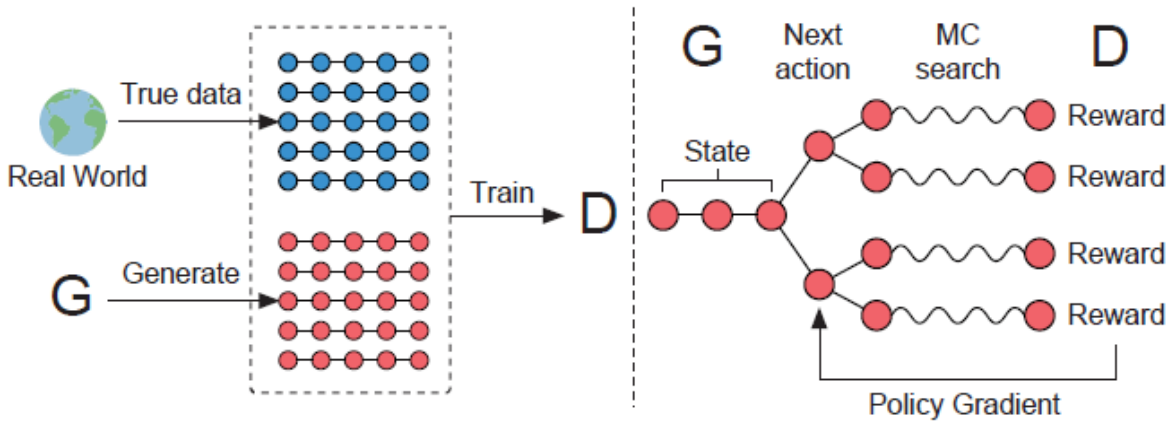
2. Objective Function of Discriminative Model

기존의 GAN 과 동일한 objective function 을 이용해 학습.

$$\min_{\phi} -\mathbb{E}_{Y\sim p_{\text{data}}}[\log D_\phi(Y)] - \mathbb{E}_{Y\sim G_\theta}[\log(1 - D_\phi(Y))]. \quad (5)$$

Unit 03 | Sequence Generative Adversarial Nets

Real world structed data가 주어졌을 때 강화학습을 기반으로 candidate tokens에 속하는 y에 대해 $Y_{1:T} = (y_1, \cdots, y_t)$, 를 생성하는 G_θ와 G가 생성한 완전한 문장 $Y_{1:T}$ 가 real sequence data 일 확률을 출력하는 $D_\emptyset$ 의 parameter 학습



SeqGAN via policy Gradient

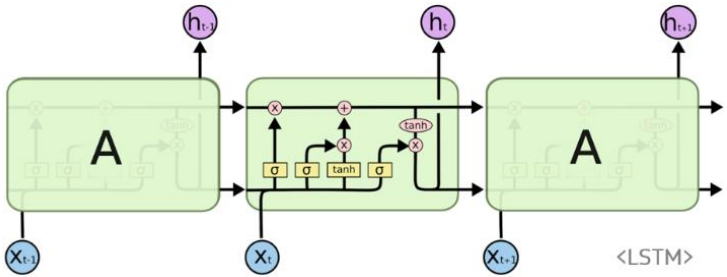
Algorithm 1 Sequence Generative Adversarial Nets

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 $Y_{1:T} = (y_1, \ldots, y_T) \sim G_\theta$
    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

The Generative Model for Sequences.

⇒ G는 LSTM cell 을 사용.



$$p(y_t|x_1,\ldots,x_t) = z(\boldsymbol{h}_t) = \text{softmax}(\boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}_t), \qquad (10)$$

The Discriminative Model for Sequences.

⇒ D는 CNN을 사용.

https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/

Evaluation Metric

Oracle 이 G의 training data set 을 제공해주고, G의 행동에 대한 평가해준다.

$$\mathrm{NLL}_{\mathrm{oracle}} = -\mathbb{E}_{Y_{1:T}\sim G_\theta}\left[\sum_{t=1}^{T}\log G_{\mathrm{oracle}}(y_t|Y_{1:t-1})\right], \qquad (13)$$

Training Setting

1. Real data distribution (G_{Oracle})을 Normal distribution 으로 LSTM 의 parameter를 초기화 하고, 이것을 이용해 10000sequences (training set S) 생성한다.

2. 1.에서 얻은 데이터와 G 에서 얻은 데이터로 D의 데이터를 구성하고 kernel size(1~T), number of kernel(100~200), dropout, L2-regularizaion으로 CNN을 구성한다.

3. Random token generation, MLE trained LSTM (G_θ) , scheduled sampling, PC-BLEU 4 개의 generative model을 seqGAN과 비교한다.

Result

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	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	

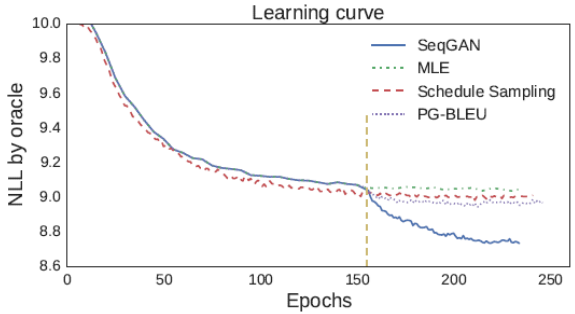


Figure 2: Negative log-likelihood convergence w.r.t. the training epochs. The vertical dashed line represents the end of pre-training for SeqGAN, SS and PG-BLEU.

Result

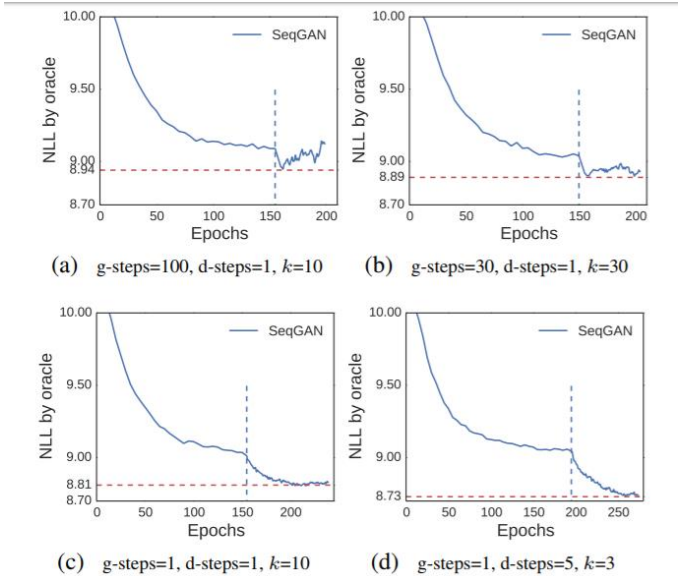


Figure 3: Negative log-likelihood convergence performance of SeqGAN with different training strategies. The vertical dashed line represents the beginning of adversarial training.

Real-world Scenarios

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**	

Obama Speech Text Generation

- when he was told of this extraordinary honor that he was the most trusted man in america
- but we also remember and celebrate the journalism that walter practiced a standard of honesty and integrity and responsibility to which so many of you have committed your careers. it's a standard that's a little bit harder to find today
- i am honored to be here to pay tribute to the life and times of the man who chronicled our time.

Human

- i stood here today i have one and most important thing that not on violence throughout the horizon is OTHERS american fire and OTHERS but we need you are a strong source
- for this business leadership will remember now i can't afford to start with just the way our european support for the right thing to protect those american story from the world and
- i want to acknowledge you were going to be an outstanding job times for student medical education and warm the republicans who like my times if he said is that brought the

Machine