GAN for Text Generation

Yahui Liu

Tencent AI Lab

yahui.cvrs@gmail.com

July 3, 2018

Overview

1. Introduction
   - Decision Theory
   - Seq2Seq & MLE
   - GAN for Text Generation
   - Solutions
2. MaskGAN
   - Task
   - Method
   - Experiments
3. Summerization

Decision Theory

Three approaches of solving decision problems[1]:

- **Generative models**: inference problem of determining the class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ for each class $p(\mathcal{C}_k)$ (explicitly or implicitly model the distribution of inputs and outputs)
- **Discriminative models**: solve the inference problem of determining the posterior class probabilities $p(\mathcal{C}_k|\mathbf{x})$
- **Discriminant function**: maps each input $\mathbf{x}$ directly onto a class label

[1] Bishop C M, et al. Pattern recognition and machine learning, 2006

Generative Model

Generative model is to approximate a data distribution as closely as possible. Yet, the following fundamental question is still largely open[1]:

Given a parametric family of models, how should we measure (and optimize) ***closeness*** *between the model distribution and the data distribution?*

Two predominant paradigms

- Maximum Likelihood Estimation (MLE)
- Adversarial Training (GAN)

[1] https://ermongroup.github.io/blog/flow-gan/

MLE

Basic

- The idea is to pick the parameters that maximize the probability (likelihood) of observing the data
- It's statistically efficient under certain conditions, and a variety of probabilistic models are learned by directly optimizing likelihood or its approximations
- RBMs, autoregressive models, VAEs, etc.

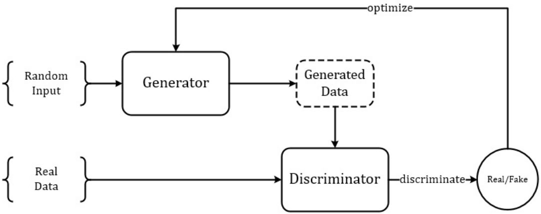
Issue

Evaluation and/or optimization of the likelihood of a model may not always be tractable, such as:

- RBMs: approximating gradients of log-likehood using MCMC
- VAEs: introducing variational approximations to the posterior

GAN

Framework



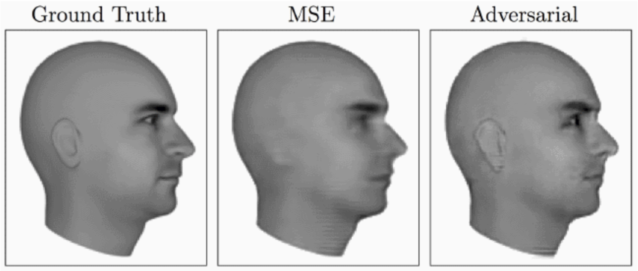
Formulation[1]

$$\min_G \max_D V(G, D) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[\log D(\mathbf{x})] + \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\mathbf{x}}))] \quad (1)$$

[1] Goodfellow I J, et al. Generative Adversarial Nets, NIPS 2014

Why GAN?

- Semi-supervised learning: limited labeled data is available[1]
- In pixel space, loss functions such as mean-squared error (MSE) and maximum likehood are unstable to slight deformations[2]

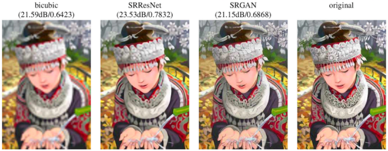


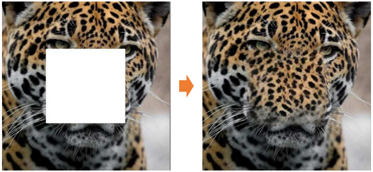
1 Goodfellow I J, et al. Generative Adversarial Nets, NIPS 2014

2 Lotter W, et al. Unsupervised learning of visual structure using predictive generative networks, ICLR 2016

Why GAN?

- Better for multi-modal outputs tasks: some promising applications, such as super-resolution[1], image inpainting[2], neural style transfer[3], adversarial examples[4], etc.





[1] Ledig C, et al. Photo-realistic single image super-resolution using a generative adversarial network, 2017

[2] Yang C, et al. High-resolution image painting using multi-scale neural patch synthesis, CVPR 2017

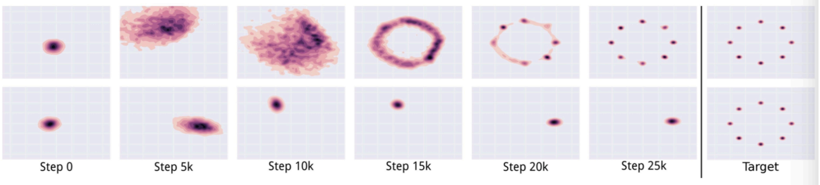
[3] Zhu J, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks, ICCV 2017

[4] Elsayed G F, et al. Adversarial examples that fool both human and computer vision, 2018

GAN

Issue

- Training instability[1] [2]: vanishing gradients on the generator (JS, KL)
- Mode collapse/dropping[3] [4]: lack of diversity, tend to generate high-frequency patterns or their combinations



[1] Martin A, et al. Wasserstein GAN, 2017
[2] Martin A, et al. Towards Principled Methods for Training Generative Adversarial Networks, ICLR 2017
[3] Metz L, et al. Unrolled generative neural networks, ICLR 2017
[4] William F, et al. MaskGAN: better text generation via filling in the ___, ICLR 2018
[5] https://zhuanlan.zhihu.com/p/25071913

GAN

Issue

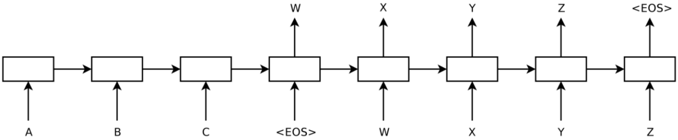
- GANs require differentiation through the visible units, and thus cannot model discrete data

GANs require differentiation through the visible units, and thus cannot model discrete data, while VAEs require differentiation through the hidden units, and thus cannot have discrete latent variables[1].

[1] Goodfellow I J, et al. Generative Adversarial Nets, NIPS 2014

Seq2Seq & MLE

Basic Model[1]



Objective

Maximum likehood estimation

$$\ell(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) = \prod_{t=1}^{T'} p(y_t | y_{1:t-1}, \mathbf{x}, \boldsymbol{\theta}) \tag{2}$$

[1] Sutskever I, et al. Sequence to sequence learning with neural networks, NIPS 2014

Seq2Seq & MLE

Dialogue is a task with *1-n* relationship between <q,a>

- Exposure bias[1] [2]: the model generates a sequence iteratively and predicts next token conditioned on its previously predicted ones that may be never observed in the training data
- Loss-evaluation mismatch[3]: **word**-level loss in training v.s. **sequence**-level evaluation metrics in testing (BLEU, human evaluation, etc.)

[1] Bengio S, et al. Scheduled sampling for sequence prediction with recurrent neural network, NIPS 2015

[2] Yu L, et al. SeqGAN: sequence generative adversarial nets with policy gradient, ACL 2017

[3] Sam W, et al. Sequence-to-sequence learning as beam-search optimization, 2016

Seq2Seq & MLE

Dialogue is a task with *1-n* relationship between <q,a>

- Automatic evaluation: no metrics to evaluate the quantity of information in a
- MLE optimizes the model to generate a from the expectation over all possible a, weighted by each a's likehood

$$\mathcal{L}(\mathbb{C}, \boldsymbol{\theta}) = \frac{1}{|\mathbb{C}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{C}} \ell(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) \tag{3}$$

GAN for Text

Ian Googfellow[1]:

If you output the word "penguin", you can't change that to "penguin + .001" on the next step, because there is no such word as "penguin + .001". You have to go all the way from "penguin" to "ostrich".

Since all NLP is based on discrete values like words, characters, or bytes, no one really knows how to apply GANs to NLP yet.

In principle, you could use the REINFORCE algorithm, but REINFORCE doesn't work very well, and no one has made the effort to try it yet as far as I know.

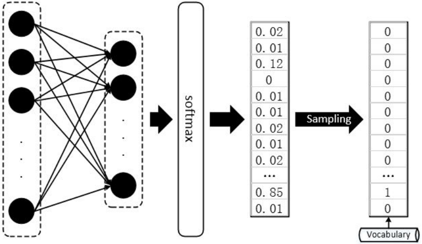
I see other people have said that GANs don't work for RNNs. As far as I know, that's wrong; in theory, there's no reason GANs should have trouble with RNN generators or discriminators. But no one with serious neural net credentials has really tried it yet either, so maybe there is some obstacle that comes up in practice.

BTW, VAEs work with discrete visible units, but not discrete hidden units (unless you use REINFORCE, like with DARN/NVIL). GANs work with discrete hidden units, but not discrete visible units (unless, in theory, you use REINFORCE). So the two methods have complementary advantages and disadvantages.

[1] https://www.reddit.com/r/MachineLearning/comments/40ldq6/generative_adversarial_networks_for_text/

GAN for Text

Embeddings	sparse features	⇒	dense features
RNNs	feature sequences	⇒	dense features
Softmax	dense features	⇒	discrete predictions



Non-differentiable Sampling

$$[softmax(\mathbf{h})]_i = \frac{exp(\mathbf{h}_i)}{\sum_{j=1}^{K=1} exp(\mathbf{h}_j)}, \quad i = 1, \cdots, d. \tag{4}$$

$$\mathbf{y} = one\_hot\left(\arg\max_i(h_i)\right) \tag{5}$$

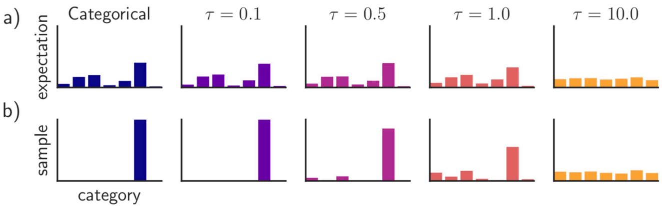
Gumbel Softmax

Gumbel distribution

$$g = -\log(-\log(u)), \;\; u \sim Uniform(0,1) \tag{6}$$

Gumbel Softmax[1] [2]

$$\mathbf{y} = softmax\left(\frac{1}{\tau}(\mathbf{h}+\mathbf{g})\right) \tag{7}$$



[1] Jang E, et al. Categorical reparameterization with gumbel-softmax, ICLR 2017

[2] Kusner M, et al. GANS for sequences of discrete elements with the gumbel-softmax distribution, NIPS 2016

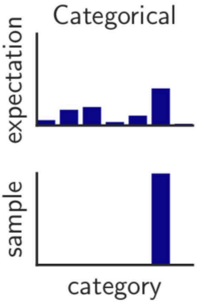
Gumbel Enough?

Unfortunately, it's not!

- Inputs: one-hot vectors
- Outputs: dense (float) vectors

The discriminator is cheating:

- Discriminator gets better, the gradient of the generator vanishes (can be improved[1] [2])



[1] Martin A, et al. Wasserstein GAN, 2017

[2] Gulrajani I, et al. Improved Training of Wasserstein GANs, NIPS 2017

WGAN & WGAN-GP

Formulation[1] [2]

$$\min_{G} \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[D(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}})] \tag{8}$$

We were unable to produce comparable results with the standard GAN objective, though ***we do not claim that doing so is impossible****.*

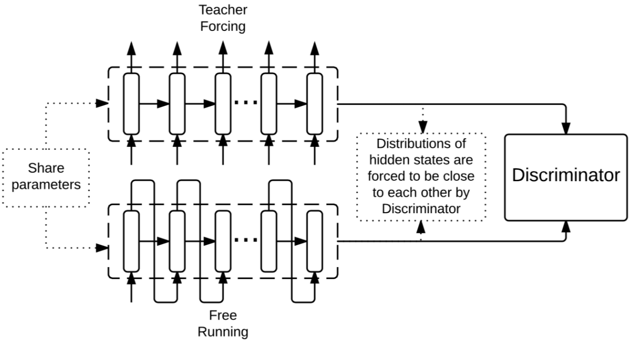
WGAN with gradient penalty (1D CNN)	
Busino game camperate spent odea	Solice Norkedin pring in since
In the bankaway of smarling the	ThiS record (31.) UBS) and Ch
SingersMay , who kill that imvic	It was not the annuas were plogr
Keray Pents of the same Reagun D	This will be us , the ect of DAN
Manging include a tudancs shat "	These leaded as most-worsd p2 a0
His Zuith Dudget , the Denmbern	The time I paid0a South Cubry i
In during the Uitational questio	Dour Fraps higs it was these del
Divos from The ' noth ronkies of	This year out howneed allowed lo
She like Monday , of macunsuer S	Kaulna Seto consficutes to repor

1. Martin A, et al. Wasserstein GAN, 2017
2. Gulrajani I, et al. Improved Training of Wasserstein GANs, NIPS 2017

Professor Forcing

Discriminator[1]

The classifier discriminates between hidden states from sampling model and teacher forcing model

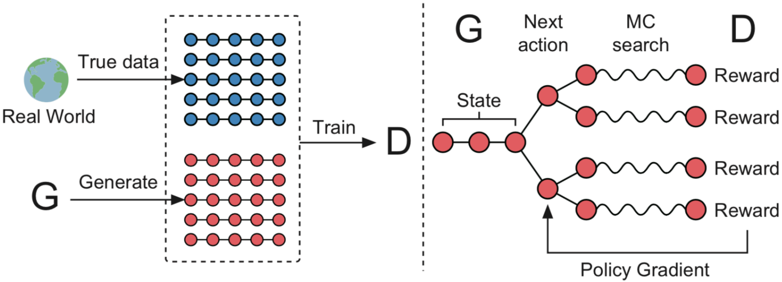


[1] Lamb A, et al. Professor forcing: a new algorithm for training recurrent networks, NIPS 2016

REINFORCE

Policy Gradient

The score of current sequences being human-generated ones assigned by the D is used as a reward for the G, which is trained to maximize the expected reward of G



[1] Yu L, et al. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, AAAI 2017

REINFORCE

Formulation[1] [2]

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x})}(Q_{+}(\{\mathbf{x}, \mathbf{y}\}|\boldsymbol{\theta})) \tag{9}$$

Reward for every generation step:

$$\nabla J(\boldsymbol{\theta}) \approx \sum_{t} (Q_{+}(\mathbf{x}, Y_t) - b(\mathbf{x}, Y_t)) \nabla \log p(y_t|\mathbf{x}, Y_{1:t-1}) \tag{10}$$

- Generator: $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x})$, a standard Seq2Seq
- Discriminator: machine-generated (Q_{-}) or human-generated (Q_{+}), a binary classifier

[1] Li J, et al. Adversarial learning for neural dialogue generation, EMNLP 2017

[2] Yu L, et al. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, AAAI 2017

Task

Goal

Infill the missing portions of a body of text are deleted or redacted:

- Part body of text is redacted: conditional language model, $p(\mathbf{x}|\mathbf{z}, \mathbf{m}(\mathbf{x}))$
- Entire body of text is redacted: language model, $p(\mathbf{x}|\mathbf{z})$

where $\mathbf{z} \sim N(0, 1)$

Example

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

[1] William F, et al. MaskGAN: better text generation via filling in the ___, ICLR 2018

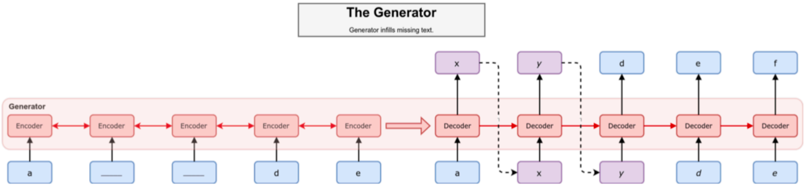
Formulation

Generator

$$p(\hat{x}_1, \cdots, \hat{x}_T | \mathbf{m}(\mathbf{x})) = \prod_{t=1}^{T} p(\hat{x}_t | \hat{x}_1, \cdots, \hat{x}_{t-1}, \mathbf{m}(\mathbf{x})) \tag{11}$$

$$G(x_t) \equiv p(\hat{x}_t | \hat{x}_1, \cdots, \hat{x}_{t-1}, \mathbf{m}(\mathbf{x}))$$

Architecture



Formulation

Discriminator

$$D_\phi(\tilde{x}_t|\tilde{x}_{0:T}, \mathbf{m}(\mathbf{x})) = p(\tilde{x}_t = x_t^{real}|\tilde{x}_{0:T}, \mathbf{m}(\mathbf{x})) \tag{12}$$

Example (Discrminator Code)

```
if is_training:
  rnn_out *= output_mask
# Prediction is linear output for Discriminator.
pred = tf.contrib.layers.linear(rnn_out, 1, scope=vs)
  ...
dis_loss_real = tf.losses.sigmoid_cross_entropy(
  real_labels, real_predictions, weights=missing)
dis_loss_fake = tf.losses.sigmoid_cross_entropy(
  targets_present, fake_predictions, weights=missing)
```

[1] https://github.com/tensorflow/models/tree/master/research/maskgan

Formulation

Reward

$$r_t \equiv \log D_\phi(\tilde{x}_t | \tilde{x}_{0:T}, \mathbf{m}(\mathbf{x})) \tag{13}$$

$$R_t = \sum_{s=t}^{T} \gamma^s r_s \tag{14}$$

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_G[R_t] = (R_t - b_t) \nabla_{\boldsymbol{\theta}} \log G_{\boldsymbol{\theta}}(\hat{x}_t) \tag{15}$$

where $\gamma =$0.8835659

Notion

$$A(a_t, s_t) = Q(a_t, s_t) - V(s_t) \tag{16}$$

where action $a_t \equiv \hat{x}_t$, state $s_t \equiv \hat{x}_1, \cdots, \hat{x}_{t-1}$

Reinforce Objective

Example (Code)

```
# Cumulative Discounted Returns.
cumulative_rewards = []
for t in xrange(FLAGS.sequence_length):
  cum_value = tf.zeros(shape=[FLAGS.batch_size])
  for s in xrange(t, FLAGS.sequence_length):
    cum_value += missing_list[s] *
                 np.power(gamma, (s - t)) * rewards_list[s]
  cumulative_rewards.append(cum_value)
  ...
# mean squared error
critic_loss = create_critic_loss(cumulative_rewards,
                                 estimated_values,
                                 present)
```

[1] https://github.com/tensorflow/models/tree/master/research/maskgan

Suggestions

Long sequences

Increment the maximum sequence length to $T + 1$ and continue training, capture dependencies over shorter sequences before moving to longer dependencies

Large vocabularies

Generating a reward only on the sampled token $\Rightarrow$ using the full information of the generator distribution

Conditional Sampling

Ground Truth	**the next day 's show <eos> interactive telephone technology has taken a new leap in <unk> and television programmers are**
MaskGAN	the next day 's show <eos> interactive telephone technology has taken a new leap in its retail business <eos> a
	the next day 's show <eos> interactive telephone technology has long dominated the <unk> of the nation 's largest economic
	the next day 's show <eos> interactive telephone technology has exercised a N N stake in the u.s. and france
MaskMLE	the next day 's show <eos> interactive telephone technology has taken a new leap in the complicate case of the
	the next day 's show <eos> interactive telephone technology has been <unk> in a number of clients ' estimates mountain-bike
	the next day 's show <eos> interactive telephone technology has instituted a week of <unk> by <unk> <unk> wis. auto

No quantitative evaluation.

Unconditional Sampling

We claim that validation perplexity alone is not indicative of the quanlity of text generated by a model.

Model	Perplexity of IMDB samples under a pretrained LM
MaskMLE	273.1 ± 3.5
MaskGAN	108.3 ± 3.5

Preferred Model	Grammaticality %	Topicality %	Overall %
LM	15.3	19.7	15.7
MaskGAN	59.7	58.3	58.0
LM	20.0	28.3	21.7
MaskMLE	42.7	43.7	40.3
MaskGAN	49.7	43.7	44.3
MaskMLE	18.7	20.3	18.3
Real samples	78.3	72.0	73.3
LM	6.7	7.0	6.3
Real samples	65.7	59.3	62.3
MaskGAN	18.0	20.0	16.7

Preferred model	Grammaticality %	Topicality %	Overall %
LM	32.0	30.7	27.3
MaskGAN	41.0	39.0	35.3
LM	32.7	34.7	32.0
MaskMLE	37.3	33.3	31.3
MaskGAN	44.7	33.3	35.0
MaskMLE	28.0	28.3	26.3
SeqGAN	38.7	34.0	30.7
MaskMLE	33.3	28.3	27.3
SeqGAN	31.7	34.7	32.0
MaskGAN	43.3	37.3	37.0

Failures – Mode Collapse

Mode collapse across various n-gram levels.

It may manifest as grammatical, inanely repetitive phrases:

It is a very funny film that is very funny It s a very funny movie and it s charming

Model	% Unique bigrams	% Unique trigrams	% Unique quadgrams
LM	40.6	75.2	91.9
MaskMLE	43.6	77.4	92.6
MaskGAN	38.2	70.7	88.2

Failures – Matching Syntax at Boundaries

The MaskGAN architecture often struggles to produce syntactically correct sequences when there is a hard boundary where it must end:

Cartoon is one of those films me when I first saw it back in 2000

Failures – Loss of Global Context

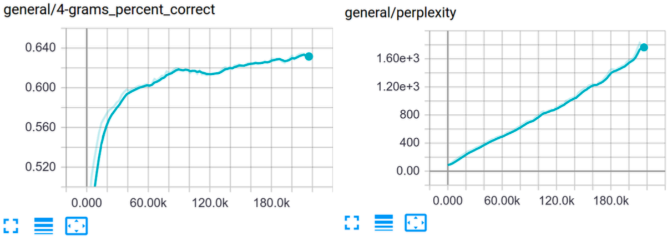
The produced samples often can lose global coherence:

This movie is terrible The plot is ludicrous The title is not more interesting and original This is a great movie

Lord of the Rings was a great movie John Travolta is brilliant

Failures – Exploration of n-gram Metrics

MaskGAN models that led to improvements of a particular n-gram metric at the extreme expense of validation perplexity.



(a) 4-gram (b) Perplexity

Status

Key issues

- Discrete elements break the differentiability, leading researchers to either avoid the issue and reformulate the problem, work in the continous domain or to consider RL methods (Gumbel, Professor forcing, REINFORCE)
- Training instability (Wassertein GANs)

Limitation

We can follow these previous methods to apply GANs for text generation and dialogue generation, but with less than linear returns (improvements):

- Generator: Seq2Seq-based model, not a real good language model.
- Discriminator: a binary classifier, if it really works, the problem of automatic evaluation has been solved.

Thanks!