Text Generation with Deep Variational GAN

Mahmoud Hossam[1], **Trung Le**[1], **Michael Papasimeon**[2], **Viet Huynh**[1], **Dinh Phung**[1]
[1]Faculty of Information Technology, Monash University
Clayton, VIC 3800
[2]School of Computing and Information Systems, The University of Melbourne
Parkville, VIC 3052
[1]{mhossam, trunglm, viet.huynh, dinh.phung}@monash.edu
[2]michael.papasimeon@unimelb.edu.au

Abstract

Generating realistic sequences is a central task in many machine learning applications. There has been considerable recent progress on building deep generative models for sequence generation tasks. However, the issue of mode-collapsing remains a main issue for the current models. In this paper we propose a GAN-based generic framework to address the problem of mode-collapse in a principled approach. We change the standard GAN objective to maximize a variational lower-bound of the log-likelihood while minimizing the Jensen-Shanon divergence between data and model distributions. We experiment our model with text generation task and show that it can generate realistic text with high diversity.

1 Introduction

Realistic sequence generation is one of the most important tasks in machine learning. In many applications such as natural language processing, music synthesis, biological sequences design, robotics, and dynamical systems modeling, a model that is able to learn in an unsupervised manner from data is crucial. Recently, there has been a considerable amount of work on developing deep generative models to generate discrete sequences of text using adversarial learning [13, 14, 1, 2]. However, two fundamental problems emerge in these models; non-differentiability of discrete data and mode-collapsing – the lack of ability to capture various modes of data.

Current approaches to learning sequence generation usually focus on tackling the non-differentiabiliy obstruction, and can be grouped into two main approaches. The first group of models makes use of reinforcement learning techniques like policy gradients, to overcome the non-differentiability with discrete data. In these architectures, the discriminator networks (fake/real binary classifiers) are disconnected from the generator networks (the main networks), and policy gradients are used to estimate an error signal from the discriminator [13, 2]. The main issue with these techniques is the high variance of the gradients estimations [1]. In addition, they do not incorporate latent space learning, which allows learning higher representations of the data. The other group of models still employ a fully differentiable GAN [3] network, but they make use of Gumbel Softmax trick [14, 1] to overcome the non-differentiability problem.

However, the issue of mode-collapsing has not been addressed yet in a principled way in all of the aforementioned models. In this paper, we present a principled way to alleviate the mode collapsing problem of sequence generation models and apply our framework to text generation task. While our work is demonstrated with discrete data, it can be straightforwardly adopted for continuous data.

In practice, GANs usually suffer from "mode-collapsing" problem [10, 12, 4], where the generator learns to map several different z values to the same output point, relying on a few modes from the data distribution. This causes GAN to be incapable of generating diverse samples from the

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

given latent codes. In this paper, we propose a new framework called Adversarial Auto-regressive Networks (ARN) to address the mode collapsing problem in a principled way when generating sequences using adversarial training. The main highlights of our model include: (i) the capability of learning to generate sentences either from random latent space or conditioned on the first token; (ii) using standard back-propagation with relaxation instead of policy gradients; (iii) overcoming mode-collapsing issues and achieving high diversity scores.

2 Adversarial Autoregressive Networks (ARN)

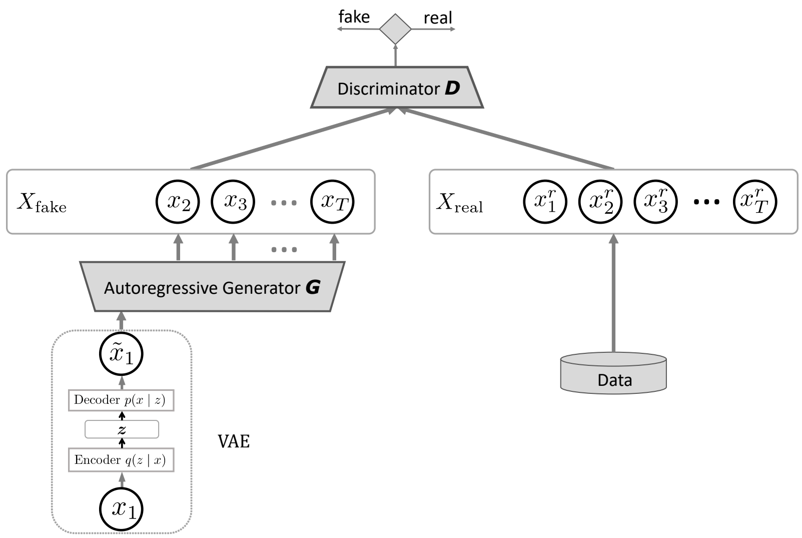

Figure 1: Proposed Adversarial Autoregressive Network

Adversarial Autoregressive Network is mainly built using an autoregressive generator, like RNN or LSTM, trained in a GAN framework for sequence generation. However, in order to learn a latent space that can be used to control the sequence generation, we employ a variational autoencoder at the first token, x_1 (Figure 1). Below we discuss in details the derivation and intuition behind our model.

Model definition. A sample X in our setting is defined as a sequence of T tokens denoted by $X = [x_1, x_2, ..., x_T]$, where we assume that all samples have length T. For our autoregressive model with model parameters θ, the log-likelihood can be written as:

$$\log\, p_G(X \mid \theta) = \sum_{i=2}^{T} \log\, p_G(x_i | \, h_{i-1}, \theta) \, + \, \log\, p_G(x_1 \mid \theta),$$

This is the default neural autoregressive model formulation. Now we start introducing an adversarial learning framework for this model by introducing a latent variable z to the autoregressive model, where we rewrite $\log\, p(x_1 \mid \theta)$ as marginalization over the z:

$$\log\, p_G(x_1 \mid \theta) = \log \sum_{z} p_G(x_1,\, z \mid \theta) \geqslant -I_{KL}(q(z \mid x_1, \phi) \,||\, p(z)) + \mathbb{E}_{q(z|x_1,\phi)}[\log\, p_G(x_1 | \, z, \theta)], \quad (1)$$

where I_{KL} is Kullback–Leibler divergence, $q(z \mid x_1, \phi)$ is an approximation of the posterior $p\,(z \mid x_1, \theta)$ and $p(z)$ is a prior distribution to z. The right hand side of Eq. (1) is a lower bound for $\log\, p_G(x_1 \mid \theta)$. We can then write $\log\, p_G(X \mid \theta)$ in terms of a lower bound as:

$$\log\, p_G(X \mid \theta) \geqslant \sum_{i=2}^{T} \log\, p_G(x_i | \, h_{i-1}, \theta) - I_{KL}(q(z \mid x_1, \phi) \,\|\, p(z)) + \mathbb{E}_{q(z|x_1,\phi)}[\log\, p_G(x_1 | \, z, \theta)]. \quad (2)$$

We propose to incorporate adversarial learning to autoregressive sequential model in a principled way. One generator $G\,(z)$ and one discriminator $D\,(X)$ are employed to create a game like in GAN while the task of the discriminator is to discriminate true data and fake data and the task of the generator is

Table 1: BLEU, FC and Diversity scores

	IMDB Reviews					
	BLEU-2 %	BLEU-3 %	FC-2 %	FC-3 %	Diversity-2 %	Diversity-3 %
SeqGAN	**86.38**	**54.19**	11.00	11.98	21.14	49.98
Ours (decoded x_1)	69.04	30.70	**14.28**	**12.04**	**31.51**	**64.21**
Ours (noise)	69.23	30.66	**14.23**	**12.11**	**31.10**	**64.13**

to generate fake data that maximally make the discriminator confused. In addition, the generator G is already available which departs from a noise $z \sim p_z$, uses the conditional distribution $p\left(x_1 \mid z, \theta\right)$ to generate x_1, and follows the autoregressive model to consecutively generate $x_{2:T}$. We come with the following minimax problem:

$$\max_G \min_D \left[\mathbb{E}_{X\sim p_d}\left[\log\, p_G\left(X \mid \theta\right)\right] - \mathbb{E}_{X\sim p_d}\left[\log\, D\left(X\right)\right] - \mathbb{E}_{z\sim p_z}\left[\log\left[1 - D\left(G\left(z\right)\right)\right]\right]\right], \quad (3)$$

where the generator G consists of the decoder $p\left(x_1 \mid z, \theta\right)$, the autoregressive model, hence G is parameterized by (θ, ϕ), and $\log\, p_G\left(X \mid \theta\right)$ is substituted by its lower bound in Eq. (2).We can theoretically prove that the minimax problem in Eq. (3) is equivalent to the following optimization problem (see the proof in Appendix):

$$\min_G\, I_{KL}\left(P_d \,||\, P_G\right) + I_{JS}\left(P_d \,||\, P_G\right), \quad (4)$$

where I_{JS} is Jenshen-Shannon divergence and P_G is the generative distribution. The optimization problem in Eq. (4) reveals that at the Nash equilibrium point the generative distribution P_G is exactly the data distribution P_d, thus overcoming the mode-collapse issue caused by original GAN formulation.

To train our model, we alternatively update G and D with relevant terms. We note that in the optimization for updating G regarding $\log\, p_G\left(X \mid \theta\right)$, we maximize its lower bound in Eq. (2) instead of the likelihood function. In addition, to overcome the non-differentiability of discrete data in our model, we use Gumbel Softmax relaxation [5, 9].

3 Experiments

We train our model for text generation task using the IMDB movie reviews dataset [8]. The dataset consists of 100,000 reviews extracted from IMDB website. The training set is divided into three groups; positive, negative (12,500 reviews each) and unlabeled reviews (50,000), and the testing set is 12,500 positive and negative each. In all of our experiments, we use sentence length of 20 words and vocabulary size of 10000. We use 500 dimensional embedding vectors, and 500 units hidden layer for LSTM Generator and 350 units for VAE.

We evaluate both the quality and mode-collapse of the generated sentences. To evaluate quality, we use BLEU score [11], which is commonly used in machine translation to compare the quality of candidate translations compared to the ground-truth reference. To evaluate the mode-collapse, we use two n-gram based scores inspired by [2], namely the *Diversity* and "*Feature Coverage* (*FC*)" scores.

Diversity Score We define the diversity as the ability to generate sentences with diverse n-grams that are not necessarily found in the test set. We measure this by computing the percentage of unique n-grams generated by the model relative to the number of all n-grams generated by the model.

Feature Coverage (*FC*) Score The *FC* score is used to measure how well the model covers all features (n-grams) of the data. The score is computed as the percentage of unique n-grams generated by the model that is found in the test set, relative to the number of all n-grams generated by the model.

It is important to notice that *FC* score doesn't necessarily correlate with BLEU score, as it computes only the percentages of "unique" n-grams that match with the test set. This means that unlike BLEU score, *FC* score is affected by the diversity of the generated sentences, where the higher the unique n-grams count, the higher the *FC* score, and vice versa. Sentences can be generated from our model in two ways; by starting from a real first word through the decoder, or from noise input through z. We compare our model to SeqGAN [13] and report the results in Table 1 for 2-grams and 3-grams of all scores.

Table 2: Generated sample sentences

Our Model (starting from decoded x_1)
i had more interest for this movie . even though they rightly watched it once . it's neither entertaining nor
he just watched this show , one of the funniest moments of <UNK> or plot . i watched this and
victor is a zombie documentary of its own hype that <UNK> puzzles on many of my favorites . and every
i couldn't relate to this movie for the first time only because time i watched the director whose friend i
Our Model (starting from noise z)
first movie was a letdown of a film . the first 30 minutes of it was ok , until i
this movie is not only horrible as a bad movie , not because it has been based on history .
probably don't know i am a fan of horror movies , it was a little while ago and i think
another is one of the worst movies , ranking up there . the script is <UNK> by the film's predecessor
the first review that i can't figure out what's a <UNK> about this film when it came out in 1972
this most awaited movie and it seems to be the worst film ever . good plot , the storyline was
holy movie is just fun . it's not to do justice . the first monkey meets this hoping to be
this film grabbed the attention to the plot with standard <UNK> , and this never even saw a story with
what i saw this episode too , i thought it was awesome for several great actors . <br / science
although movie <UNK> if not to forget with a british tradition . i understand why the ideas presented well enough
this 1985 cult film " animal " was a <UNK> of a couple tells the most of youth , but
this disappointing . . . richard murray was a landmark in many two so far , such a pretentious crap
people misguided . i found way almost no battlestar <UNK> to complain about this movie (and like a <UNK>
this superbly finished the <UNK> , i was very excited about martin carter and scott he ran the tv series
this movie is really very sweet here . it reminded me of the dvd both brought us that devil's <UNK>
if just watched this movie when i was around 15 years ago and although it looks like it was boring

We can see that "Feature-Coverage" score is very similar to the baseline, while the Diversity score is higher. This suggests that our model is capable of learning the same overall features (n-grams) of the data, yet it is also able to generate more diverse samples out of these learned features. To qualitatively evaluate the "quality" of this diverse output, we show generated samples with average or low BLEU scores in Table 3. We can see that low BLEU sentences generally still have grammatical structure and sometimes semantically meaningful, suggesting that the model does not learn random or gibberish modes when they do not match directly with data. In addition, we show general samples from the output in Table 2.

While the BLEU scores are lower than the baseline, yet the *FC* and Diversity scores are higher. For text generation tasks, BLEU is computed relative to the whole test corpus. This means that BLEU score can increase significantly on the expense of output diversity, when few generated sentences match highly with test corpus but are repeated very frequently. Therefore, we see that it is essential that text generative models are evaluated for both quality and diversity in a unified manner[2].

4 Conclusion

In this paper we presented a sequential deep generative model to generate sequences based on a principled approach to address the mode collapse problem. We applied the model to text generation task and showed that the model can generate grammatically and semantically meaningful sentences with high diversity.

In future work, we will investigate the latent space learning of the model, aiming to learn smooth transitions between sentence styles or sentiments. We intend to improve the latent learning using recent methods such as learned similarity metric [7] that include all of the input tokens, and incorporate the ability to control output sentences conditioned on true data through the encoder. We will also fine tune the model to achieve better results, design more comprehensive quality/diversity measures, and compare with recent sequential generative models [14, 1].

Acknowledgments

We gratefully acknowledge the partial support from the Defence Science and Technology (DST) Group for this work.

References

[1] Liqun Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. Adversarial text generation via feature-mover's distance. In *NIPS*, 2018.

[2] William Fedus, Ian Goodfellow, and Andrew M. Dai. MaskGAN: Better text generation via filling in the. In *International Conference on Learning Representations*, pages 1–15, 2018.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: Training generative adversarial nets with multiple generators. In *International Conference on Learning Representations*, 2018.

[5] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. *ArXiv e-prints*, abs/1611.01144, 2016.

[6] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *The 2nd International Conference on Learning Representations (ICLR)*, 2013.

[7] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *ArXiv e-prints*, abs/1512.09300, 2015.

[8] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150, 2011.

[9] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *ArXiv e-prints*, abs/1611.00712, 2016.

[10] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, pages 1–25, 2016.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318, 2002.

[12] Ben Poole, Alexander A Alemi, Jascha Sohl-Dickstein, and Anelia Angelova. Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*, pages 1–8, 2016.

[13] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. pages 1–11, 2017.

[14] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *ICML*, 2017.

Appendix

Generative Adversarial Networks (GAN)

The basic idea of Generative adversarial networks (GAN) [3] is the adversarial training between two players. The goal of the first player, the generator G , is to get very good at generating data that is very close to the real data that comes from real distribution $p_d(x)$. The goal of the second player, the discriminator D, is to distinguish real data from fake data generated by the generator. The standard GAN objective to optimize is the minimax game between D and G:

$$\min_G \max_D \; \mathbb{E}_{x\sim p_d} \log\, D(x) \,+\, \mathbb{E}_{z\sim p_z} \log\left(1 - D(G(z))\right) \tag{5}$$

where z is the random noise input to G, and p_z is the prior distribution of the z. After the training is finished, the generator is used to generate data from any random input z.

Table 3: Low/average BLEU samples

BLEU-3	Sample sentences
00.00%	a creative , well-made comedy that focuses into a scary film by peter jackson , are heading for <UNK>
00.00%	this direction struck me , that's ! this dreadful zombie movie , generally and curtis are terrific but doesn't .
00.00%	but effort , rural accent must be developed from joke a video <UNK> this new version starts ! some say
11.11%	okay exceptionally documentary about steve hayes (who doesn't even destroy to the <UNK> ! that doesn't act better than
11.11%	sorry you're a fan , might even like your a spoiler as a master of shakespeare ; a little girl
11.11%	the wish i can relate to seeing many kind of kills the other well-made film adaptations than the godfather documentary
22.22%	wow hitchcock's prince meets thriller of the modern relationships in <UNK> terror and <UNK> above <UNK> the war film and
27.77%	the fine ending is , not even close to " princess <UNK> , " that it'd be extremely cool .
27.77%	this is an excellent , unbelievable , rather colorful show . brian <UNK> did never become a straight to part
33.33%	brilliant carpenter's halloween is actually my favorite films of my life . it made thriller a few laughs of new
33.33%	seriously imdb many previous reviews posted here about this movie due to a pair of art killings & <UNK> combined
33.33%	i haven't been such a clever flick , which is neither so <UNK> a mystery . the acting is somewhere
33.33%	so is a lovely film by any <UNK> . it makes a dumb decision to play fairly typical thing .
38.33%	wow surprised how wanted to see this . i thought about two of steven johnson are superb . however ,
44.44%	the tense 1931 melodrama is one of the finest entries to all ideas . while this is a great anime
44.44%	i believe i know things i can say , you always come out of my time and the way max
44.44%	the movie is a piece of crap . a group of guy lives off the shelf who meets shrek ,
50.00%	this izzard is a one-in-a-million comic genius ! many stars in this film ? <br / ><br / <UNK> the

Variational Autoencoder (VAE)

Variational Autoencoders (VAEs) [6] approximate the maximum log-likelihood and can be trained using gradient decent. VAEs are trained to maximize a variational lower bound L on log-likelihood:

$$L(x;\,\theta) = \mathbb{E}_{z\sim q(z|x)}[\log p_{model}(x|z)] - D_{KL}[q(z|x)\|p_{model}(z)]$$

where $q(z|x)$ is a posterior and $p_{model}(z)$ is a prior distributions for latent variable z. The first term is the data reconstruction likelihood. The second term works a regularizer to make $q(z|x)$ and $p_{model}(z)$ close to each other. $p_{model}(z)$ can be chosen as $\mathcal{N}(0,\,I)$. $p_{model}(x|z)$ is the decoder, modeled as a neural network that resembles reconstruction of x from z sampled from the learned $q(z|x)$.

Proof of final objective function

Consider this optimization problem:

$$\max_G \min_D \left[\mathbb{E}_{X\sim p_d}\left[\log\, p_G\left(X \mid \theta\right)\right] - \mathbb{E}_{X\sim p_d}\left[\log\, D\left(X\right)\right] - \mathbb{E}_{z\sim p_z}\left[\log\left[1 - D\left(G\left(z\right)\right)\right]\right]\right] \tag{6}$$

Given a generator G, the optimal $D^*\left(G\right)$ is determined as:

$$D_G^*\left(X\right) = \frac{p_d\left(X\right)}{p_G\left(X\right) + p_d\left(X\right)}$$

where $p_G\left(X\right)$ is the distribution induced from $G\left(X\right)$ where $X \sim p_d\left(X\right)$.

Substituting D_G^* back to Eq. (6), we obtain the following optimization problem regarding G:

$$\max_G \left(\mathbb{E}_{p_d}\left[\log\, p_G\left(X\right)\right] - I_{JS}\left(P_d\|P_G\right)\right) \tag{7}$$

The objective function in Eq. (7) can be written as

$$\begin{aligned}\mathbb{E}_{p_d}\left[\log\, p_G\left(X\right)\right] - I_{JS}\left(P_d\|P_G\right) &= -I_{JS}\left(P_d\|P_G\right) - I_{KL}\left(P_d\|P_G\right) - \mathbb{E}_{p_d}\left[\log\, p_d\right] \\ &= -\,I_{JS}\left(P_d\|P_G\right) - I_{KL}\left(P_d\|P_G\right) + \text{const}\end{aligned}$$

Therefore, the optimization problem in Eq. (7) is equivalent to:

$$\min_G \left(I_{JS}\left(P_d\|P_G\right) + I_{KL}\left(P_d\|P_G\right)\right)$$

At the Nash equilibrium point of this game, we hence obtain: $p_G\left(X\right) = p_d\left(X\right)$.