Adversarial Feature Matching for Text Generation

Presenter: Yizhe Zhang

Joint work with: Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, Lawrence Carin

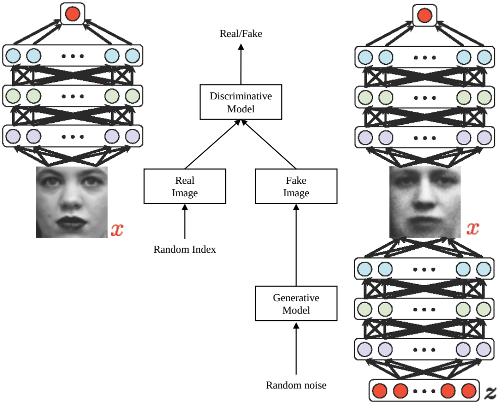
Duke University

August 9, 2017

Background

Generative Adversarial Networks

- A game between:
  - Discriminative model D
  - Generative model G
- G: trained to maximize the probability of D making a mistake
- D: trained to estimate the probability that a sample came from data distribution rather than G



Motivation & contributions

- **Motivation:** Generate realistic-looking text via adversarial training.
- **Difficulties:** (due to discrete nature of text)
  - Synthetic data is not directly differentiable.
  - Transitions in text are less smooth than in images. $\rightarrow$ mode collapsing.
- **Our approach:**
  - Discretization approximations using *Gumbel-softmax*.
  - Ameliorating mode-collapsing issue via *feature moment matching*.

- We specify an LSTM generator to translate a latent code vector, $\boldsymbol{z}$, into a synthetic sentence $\tilde{s}$.
- All other words in the sentence are sequentially generated using the RNN, *based on previously generated words*, until the end-sentence symbol is generated.

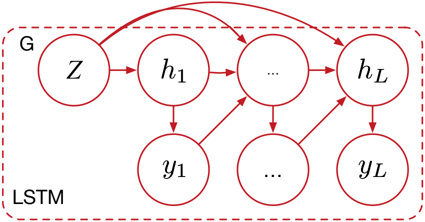


Figure: LSTM generator

- *argmax* operation is not differentiable.
- We consider a *Gumbel-softmax* approach to approximate argmax operation .

$$\boldsymbol{y}_{t-1} = \mathbf{W_e}\text{softmax}(\mathbf{V}\boldsymbol{h}_{\mathbf{t-1}} \odot \mathbf{1}/\tau)\,. \tag{1}$$

where $\odot$ represents the element-wise product. $\mathbf{W_e} \in \mathbb{R}^{\mathbf{k}\times\mathbf{V}}$ is a word embedding matrix. $\mathbf{V}$ is a weight matrix. Note that when $\tau \to 0$, this approximation approaches argmax operation.

- CNNs weight each word equally and are empirically better at abstracting features particularly with long sentences.
- A sentence is represented as a matrix $\mathbf{X} \in \mathbb{R}^{\mathbf{k} \times \mathbf{T}}$, followed by a convolution operation.
- A max-over-time pooling operation is then applied.

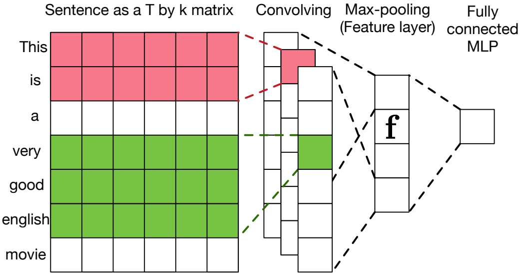


Figure: CNN discriminator

Overview

- The adversarial game is the following:
- $D(\cdot)$ attempts to select informative sentence features.
- $G(\cdot)$ aims to match these features.
- Features are selected according to **syn/real discrimination ability**, **latent code reconstruction** and **moment matching precision**.

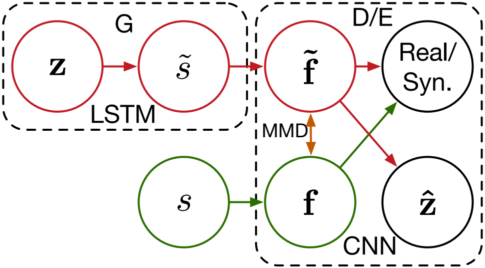


Figure: Model scheme of TextGAN.

Feature moment matching (for G)

- Optimization schemes:

$$\mathcal{L}_G = \mathcal{L}_{MMD^2}$$
$$\mathcal{L}_D = \mathcal{L}_{GAN} + \lambda_r \mathcal{L}_{recon} - \lambda_m \mathcal{L}_{MMD^2}$$

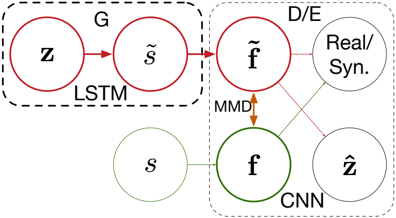
- For G, consider a moment matching loss over *feature vector* using *maximum mean discrepancy* (MMD).

$$\begin{aligned}\mathcal{L}_{MMD^2} &= ||\mathbb{E}_{x\sim\mathcal{X}}\phi(x) - \mathbb{E}_{y\sim\mathcal{Y}}\phi(y)||^2_{\mathcal{H}} \\ &= \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{x'\sim\mathcal{X}}[k(x,x')] \\ &+ \mathbb{E}_{y\sim\mathcal{Y}}\mathbb{E}_{y'\sim\mathcal{Y}}[k(y,y')] - 2\mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y\sim\mathcal{Y}}[k(x,y)].\end{aligned} \tag{2}$$

- With a Gaussian kernel , minimizing the MMD objective $\Leftrightarrow$ minimizing *all order of moments* of two empirical distributions.

Feature moment matching (for G)

- Vanilla GAN: D independently judge each syn/real data.
- The MMD loss for G: match distributions, enforce diversity.
- The gradient signal back-propagated from feature layer is more direct.

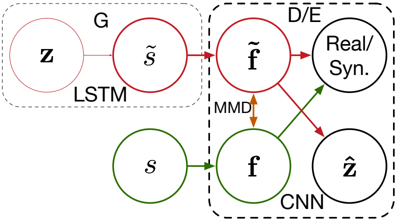


Feature moment matching (for D)

- Optimization schemes:

$$
\begin{aligned}
\mathcal{L}_G &= \mathcal{L}_{MMD^2} \\
\mathcal{L}_D &= \mathcal{L}_{GAN} + \color{blue}{\lambda_r \mathcal{L}_{recon}} - \color{red}{\lambda_m \mathcal{L}_{MMD^2}} \\
\mathcal{L}_{GAN} &= -\mathbb{E}_{s\sim\mathcal{S}} \log D(s) - \mathbb{E}_{z\sim p_z} \log[1 - D(G(z))] \\
\mathcal{L}_{recon} &= ||\hat{\boldsymbol{z}} - \boldsymbol{z}||^2 ,
\end{aligned}
$$

- The reconstruction loss in D : select the most *representative (information-preserving)* features.
- The MMD loss in D: select the most *challenging* features.



Pre-training strategy

- For G, pretrained by using sequence-to-sequence language model.
- For D, *permutation training* strategy: randomly swap two words to construct a *tweaked* sentence counterpart.

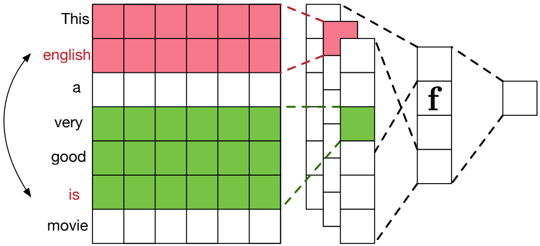


Figure: Permutation training

Variants

- Problems: a minibatch of data points is not densely sampled in feature space with high dimension (900).
- Variants:
  - **(MMD-L)**: Mapping feature space to lower dimension (by D).
  - **(MM)**: Use *accumulated batches*, match *first-order* moment .

$$L_G = (\boldsymbol{\mu}_s - \boldsymbol{\mu}_r)^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_r)$$

  - **(CM)**: Use accumulated batches, match *first-order* and *second-order* moment, which can be interpreted as an lower-bound of JSD between two MVNs:

$$L_G = \text{tr}(\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_r^{-1}\boldsymbol{\Sigma}_s) + (\boldsymbol{\mu}_s - \boldsymbol{\mu}_r)^T(\boldsymbol{\Sigma}_s^{-1} + \boldsymbol{\Sigma}_r^{-1})(\boldsymbol{\mu}_s - \boldsymbol{\mu}_r)$$

$\boldsymbol{\Sigma}_{(s/r)}$ and $\boldsymbol{\mu}_{(s/r)}$ are (accumulated) covariance matrix and mean vector for syn/real feature vector.

Empirical evaluation

- **Dataset:** 0.5M Arxiv sentences + 0.5M BookCorpus sentences .
- **Evaluation:** Kernel density estimation (KDE) .
- **Evaluation:** Corpus-level BLEU score .
- Compared with baseline auto-encoder, variational auto-encoder and seqGAN [Yu et. al. 2016]

Experimental Result

Generated text

- Produce novel phrases by imagining concept combinations. (b)
- In general, the synthetic sentences seem syntactically reasonable.
- However, the semantic meaning is less well preserved with long sentences. (e)

Table: Sentences generated by textGAN.

a	we show the joint likelihood estimator (in a large number of estimating variables embedded on the subspace learning) .
b	this problem achieves less interesting choices of convergence guarantees on turing machine learning .
c	in hidden markov relational spaces , the random walk feature decomposition is unique generalized parametric mappings.
d	i see those primitives specifying a deterministic probabilistic machine learning algorithm .
e	i wanted in alone in a gene expression dataset which do n't form phantom action values .
f	as opposite to a set of fuzzy modelling algorithm , pruning is performed using a template representing network structures .

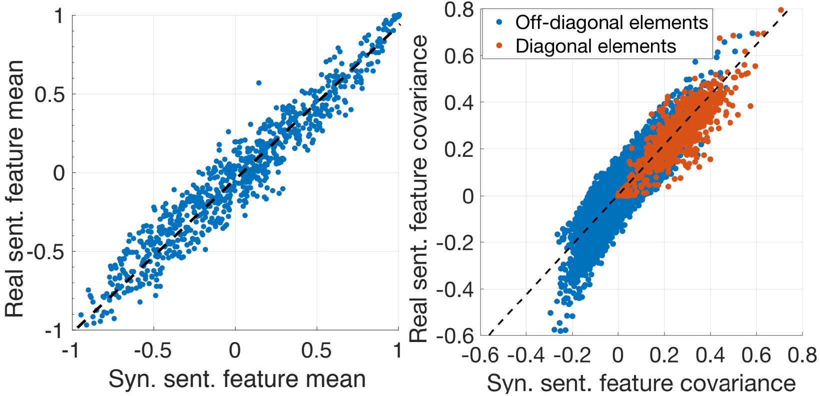


Figure: Moment matching comparison. Left: expectations of latent features from real *vs.* synthetic data. Right: elements of covariance matrix for real and synthetic data, respectively.

Experimental Result

Sentence transition

Table: Intermediate sentences produced from linear transition between two points.

	textGAN	AE
A	our methods apply novel approaches to solve modeling tasks .	
-	our methods apply novel approaches to solve modeling .	our methods apply to train UNK models involving complex .
-	our methods apply two different approaches to solve computing .	our methods solve use to train) .
-	our methods achieves some different approaches to solve computing .	our approach show UNK to models exist .
-	our methods achieves the best expert structure detection .	that supervised algorithms show to UNK speed .
-	the methods have been different related tasks .	that address algorithms to handle) .
-	the guy is the minimum of UNK .	that address versions to be used in .
-	the guy is n't easy tonight .	i believe the means of this attempt to cope .
-	i believe the guy is n't smart okay?	i believe it 's we be used to get .
-	i believe the guy is n't smart .	i believe it i 'm a way to belong .
B	i believe i 'm going to get out .	

- Higher BLEU, lower KDE is better.

Table: Quantitative results using BLEU-2,3,4 and KDE.

	BLEU-4	BLEU-3	BLEU-2	KDE(nats)
AE	0.01 ± 0.01	0.11 ± 0.02	0.39 ± 0.02	2727 ± 42
VAE	0.12 ± 0.06	0.40 ± 0.06	0.61 ± 0.07	2025 ± 25
seqGAN	0.04 ± 0.04	0.30 ± 0.08	0.67 ± 0.04	2019 ± 53
textGAN(MM)	0.09 ± 0.04	0.42 ± 0.04	0.77 ± 0.03	1823 ± 50
textGAN(CM)	0.12 ± 0.03	0.49 ± 0.06	0.84 ± 0.02	1686 ± 41
textGAN(MMD)	$\mathbf{0.13\pm0.05}$	0.49 ± 0.06	0.83 ± 0.04	1688 ± 38
textGAN(MMD-L)	0.11 ± 0.05	$\mathbf{0.52\pm0.07}$	$\mathbf{0.85\pm0.04}$	$\mathbf{1684\pm44}$

Conclusion

- We introduced a novel approach for text generation using adversarial training
- We discussed several techniques to alleviate practical issues when training GAN on text domain.
- We demonstrated that the proposed model delivers superior performance compared to related approaches.

Q&A

paper: https://arxiv.org/abs/1706.03850

code: https://github.com/dreasysnail/textGAN_public

poster: #89 Wednesday

- CNNs weight each word equally and are empirically better at abstracting features particularly with long sentences.
- A sentence is represented as a matrix $\mathbf{X} \in \mathbb{R}^{\mathbf{k} \times \mathbf{T}}$, by concatenating its word embeddings as columns.
- A convolution operation involves a filter $\mathbf{W_c} \in \mathbb{R}^{\mathbf{k} \times \mathbf{h}}$, applied to a window of h words to produce a new feature.
- A max-over-time pooling operation is then applied to the feature map.

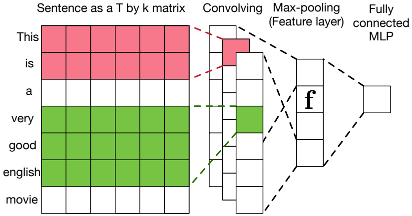


Figure: CNN discriminator