Adversarial Learning

NLP-SC
田植良
2017.7.18

Outline

- GAN
- Adversarial learning in NLP
  - Adversarial Examples
    - Properties of NN
    - Adversarial and Virtual Adversarial Learning
    - Adversarial and Virtual Adversarial Learning in NLP
  - GAN in NLP
    - SeqGAN

Generative Adversarial Nets(GANs)

2014 NIPS
Goodfellow
Montreal, Canada
citation: 931

GANs

- Goal
  - “estimating generative models via an adversarial process” by author
  - Improve on generative model and/or discriminative model
- Idea
  - G: generative model $G(\boldsymbol{z};\theta_g)$
    - z (random noise) -> x (data)
    - Capture data distribution, fool the discriminative model
  - D: discriminative model $D(\boldsymbol{x};\theta_d)$
    - x (data(real data & generated data)) -> P (possibility of real data)
    - Estimate the probability: real data vs generated data
  - Loss function

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

GANs

- Experiments
  - Discriminative model
    - Test set: real data(positive) generated data(nagetive)
    - Metric: log-likelihood

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [5]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

  - Generative model
    - Ability to capture data distribution

GANs

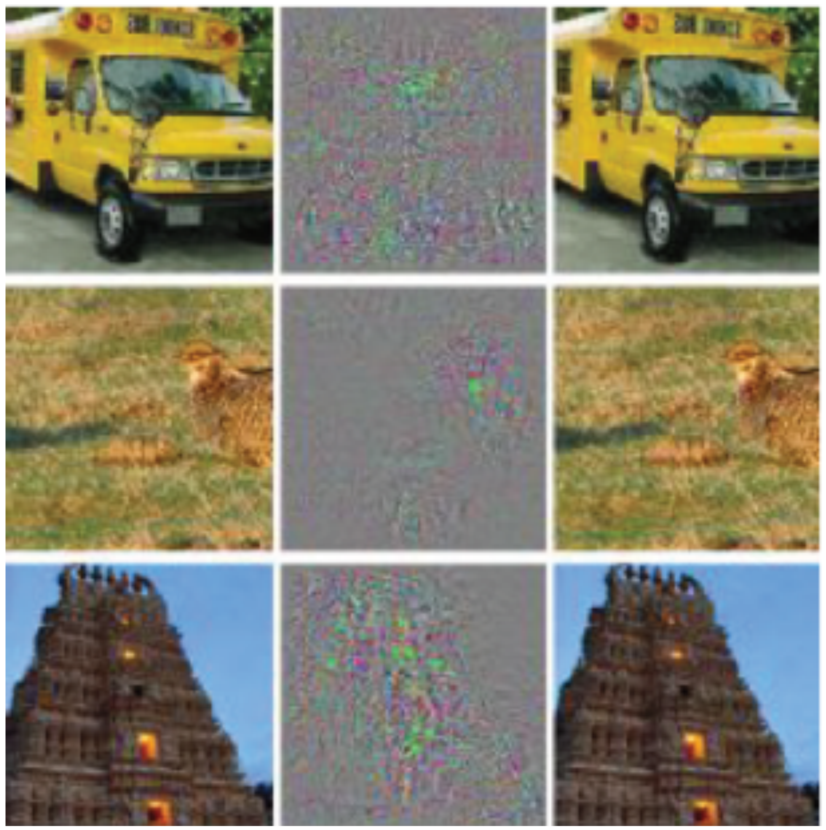
- Discuss
  - Min-max two-player game
  - Exist an unique solution (if capacities of G and D are enough)
    - G: capture the data distribution totally
    - D: can not work ($P \equiv 1/2$)
- Future work (selected)
  - Conditional generative model
    - more practical
  - Semi-supervised learning
    - improve the discriminative model when limited labeled data is available

Outline

- GAN
- Adversarial learning in NLP
  - Adversarial Examples
    - Properties of NN
    - Adversarial and Virtual Adversarial Learning
    - Adversarial and Virtual Adversarial Learning in NLP
  - GAN in NLP
    - SeqGAN

Properties of NN

- About
  - Intriguinting properties of neural networks
  - Christian Szegedy, Google Inc, arkiv 2013
- Contributions (selected)
  - Blind spots in nerual networks



1	great	great
2	decent	decent
3	$\times$bad	excellent
4	excellent	nice
5	Good	Good
6	fine	$\times$bad
7	nice	fine
8	interesting	interesting
9	solid	entertaining
10	entertaining	solid

  - For deep neural networks, the smoothness assumption that underlies many kernel methods does not hold.

(Virtual) Adversarial Training

- About
  - Distributional Smoothing with Virtual Adversarial Training
  - Miyato(Tokyo University), Goodfellow(Google), 2016 ICLR
- Idea
  - Adversarial training(Supervise): F(x)->y, F(x+r)->y   r:perturbation
  - Virtual Adversarial training(Semi-supervise): F(x)->y,   sim(F(x'), F(x'+r))
- Loss function
  - Adversarial

$$r_{adv}^{(n)} = \arg\min_{r}\{p(y^{(n)}|x^{(n)} + r^{(n)}); ||r||_2 \leq \epsilon\}$$

$$J = \frac{1}{N}\sum_{n=1}^{N}\log(p(y^{(n)}|x^{(n)},\theta)) + \lambda\frac{1}{N}\sum_{n=1}^{N}p(y^{(n)}|x^{(n)} + {r_{adv}}^{(n)},\theta)]$$

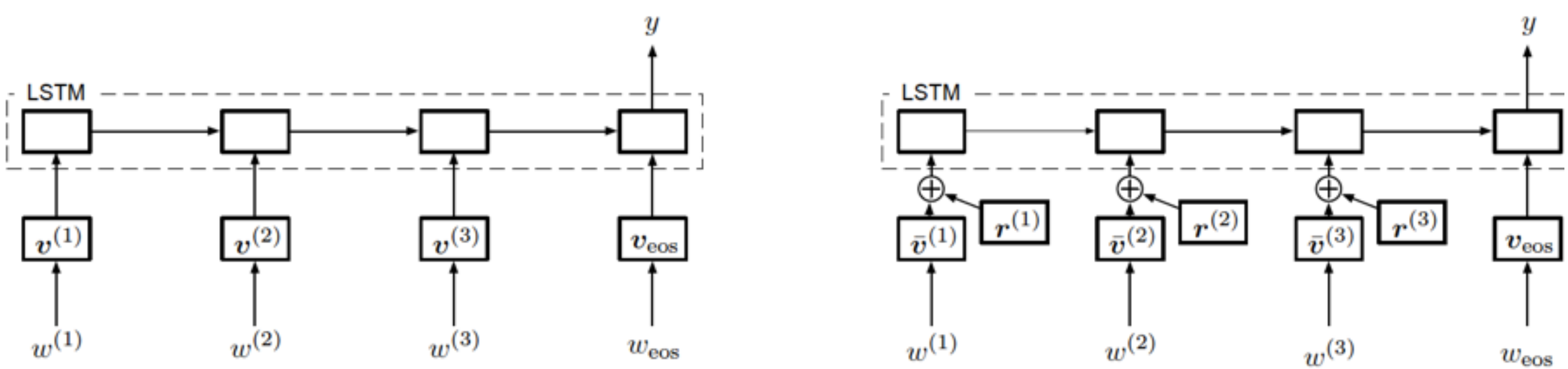
  - Virtual Adversarial

$$r_{v-adv}^{(n)} = \arg\max_{r}\{KL[p(y|x^{(n)},\theta)||p(y|x^{(n)} + r^{(n)},\theta)]; ||r||_2 \leq \epsilon\}$$

$$J = \frac{1}{N}\sum_{n=1}^{N}\log(p(y^{(n)}|x^{(n)},\theta)) - \lambda\frac{1}{N}\sum_{n=1}^{N}KL[p(y|x^{(n)},\theta)||p(y|x^{(n)} + {r_{v-adv}}^{(n)},\theta)]$$

(Virtual) Adversarial in NLP

- About
  - Adversarial Training Methods for Semi-Supervised Text Classification
  - Miyato(Tokyo University), Goodfellow(Google), 2017 ICLR
- Idea
  - Apply (virtual) adversarial training on NLP (text classification)
  - Get adversarial examples by perturbing embeddings

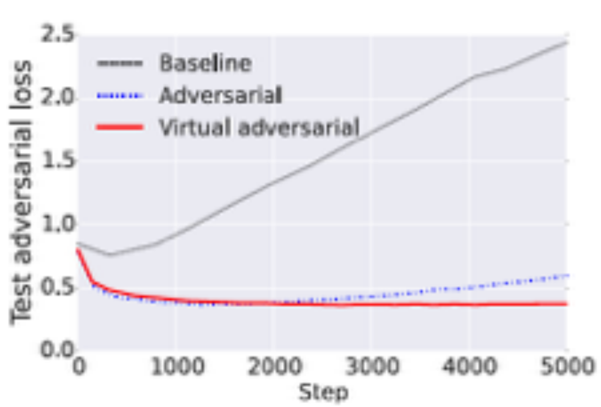


(Virtual) Adversarial in NLP

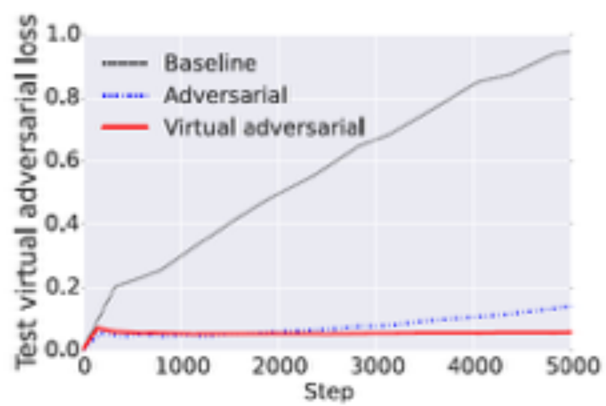
- Experiment & Analysis
  - Task: text classification
  - Accuracy on Classication

Method	Test error rate
Baseline (without embedding normalization)	7.33%
Baseline	7.39%
Random perturbation with labeled examples	7.20%
Random perturbation with labeled and unlabeled examples	6.78%
Adversarial	6.21%
Virtual Adversarial	**5.91%**

  - Performance on test data with noise



(b) $L_{\mathrm{adv}}(\boldsymbol{\theta})$



(c) $L_{\mathrm{v\text{-}adv}}(\boldsymbol{\theta})$

  - Performance on "blind spots"

	'*good*'			
	Baseline	**Random**	**Adversarial**	**Virtual Adversarial**
1	great	great	decent	decent
2	decent	decent	great	great
3	×bad	excellent	nice	nice
4	excellent	nice	fine	fine
5	Good	Good	entertaining	entertaining
6	fine	×bad	interesting	interesting
7	nice	fine	Good	Good
8	interesting	interesting	excellent	cool
9	solid	entertaining	solid	enjoyable
10	entertaining	solid	cool	excellent

Summary on Adversarial Examples

- Adapt to NLP
- Similiar to GANs
  - Min-max two-player game
  - Product new samples to improve the discriminative model
    - GAN: z -> G model -> G(z) -> D model -> D(G(z))
    - Adv Exam: x -> D model -> r+x -> D model -> D(x+r)
- Different with GANs
  - Optimization: GANs joint optimization D(G(z)))
  - Applications: GANs only adapt to real-value data (eg: image)
  - Goal: GANs output: G & D
  - Role of Samples:
    - GANs: capture data distribution -> act as nagetive samples
    - Adv Exam: capture blind spots -> act as positive samples

Outline

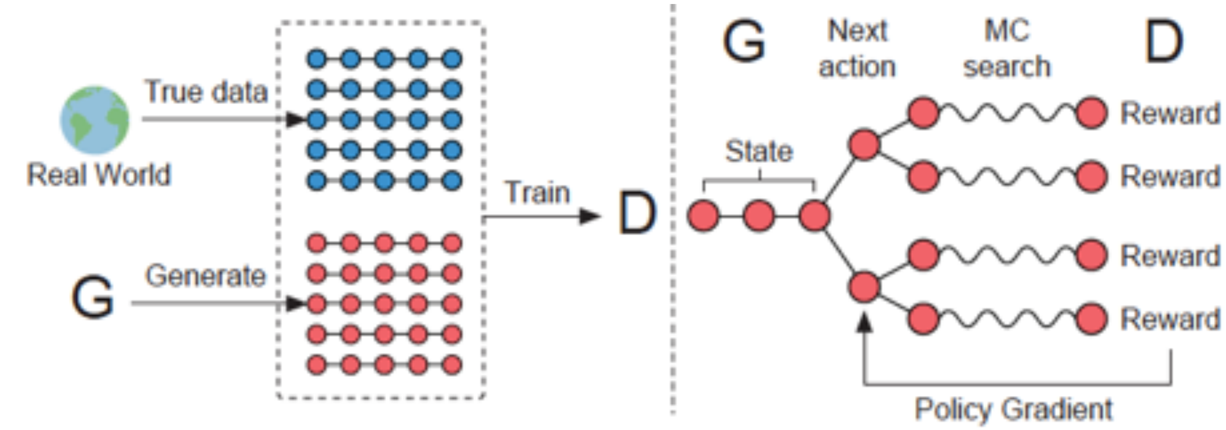
- GAN
- Adversarial learning in NLP
  - Adversarial Examples
    - Properties of NN
    - Adversarial and Virtual Adversarial Learning
    - Adversarial and Virtual Adversarial Learning in NLP
  - GAN in NLP
    - SeqGAN

SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient

2017 AAAI

Lantao Yu, Weinan Zhang(Shanghai Jiao Tong)
Jun Wang(UCL, UK)

SeqGAN



- Goal
  - Adapt GAN to NLP
    - GAN requires real-value data
    - Discriminative model can only assess a complete sequence
- Idea
  - Real-value data: RL: G(policy network) D(reward function)
  - Discriminate a Sequence: complete the rest by Monte Carlo search
- Model
  - RL
    - Given reward function Q, maximize the expectation of reward on G

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

    - Re-train reward function Q, minimize log-likelihood on D

$$Y_{1:T} = (y_1, \ldots, y_t, \ldots, y_T) \qquad Q_{D_\phi}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\phi(Y_{1:T})$$

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

SeqGAN

- Model
  - Monte Carlo search
    - Given current sequence, get future outcome (similiar to Chess)

$$Y_{1:T} = (y_1, \ldots, y_t, \ldots, y_T)$$

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

G_β same as generator, but can use a simplified version

    - Rewrite the reward function Q

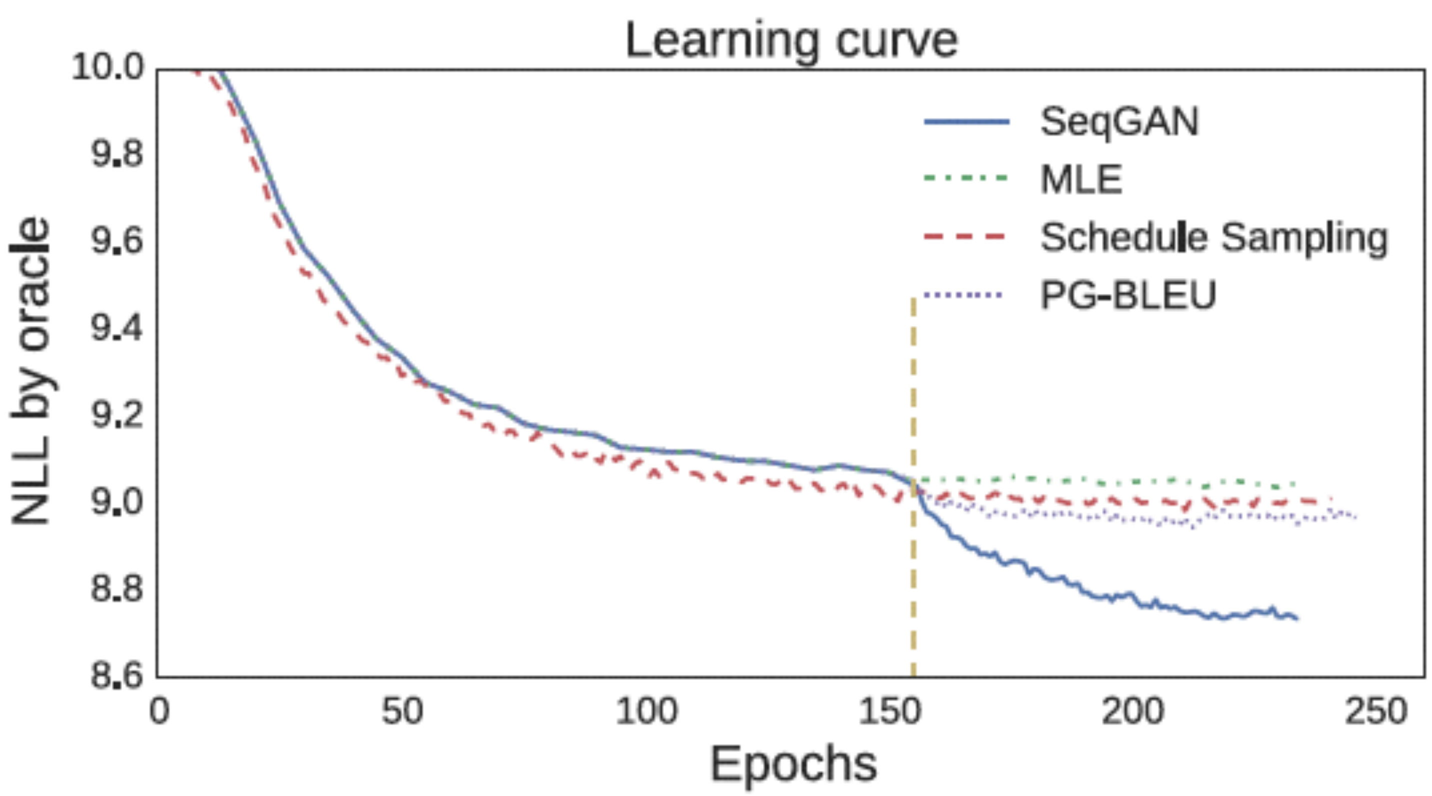
$$Q_{D_\phi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) =$$
$$\begin{cases} \frac{1}{N} \sum_{n=1}^{N} D_\phi(Y_{1:T}^n),\ Y_{1:T}^n \in \mathrm{MC}^{G_\beta}(Y_{1:t}; N) & \text{for} \quad t < T \\ D_\phi(Y_{1:t}) & \text{for} \quad t = T, \end{cases}$$

SeqGAN

- Experiments
  - G: RNN D: CNN
  - Evaluation Metric: evaluate G by language model
    - The expectation of LM score on G\theta

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

  - Result



Some Thoughts

- Effects
  - For G: GANs bring a better loss function / reward
    - Ground-truth are sometimes not suitable (eg: dialog systems / MT)
  - For D: GANs bring the better nagetive samples
    - Discriminative models are common on industry.
    - Data may be the most improtant.
    - Stronger nagetive samples for ranking/classification tasks
    - More random nagetive samples for retrieval tasks ?
    - More controlled samples for specific tasks
- GAN in NLP
    - Generate embedding as example in GAN
    - x: embedding; $x = G(z)$; $D(x)$

Q & A