adversarial training methods
for semi-supervised text classification

abstract

adversarial training + virtual adversarial training

noise in neural network
input 자체에는 적용X

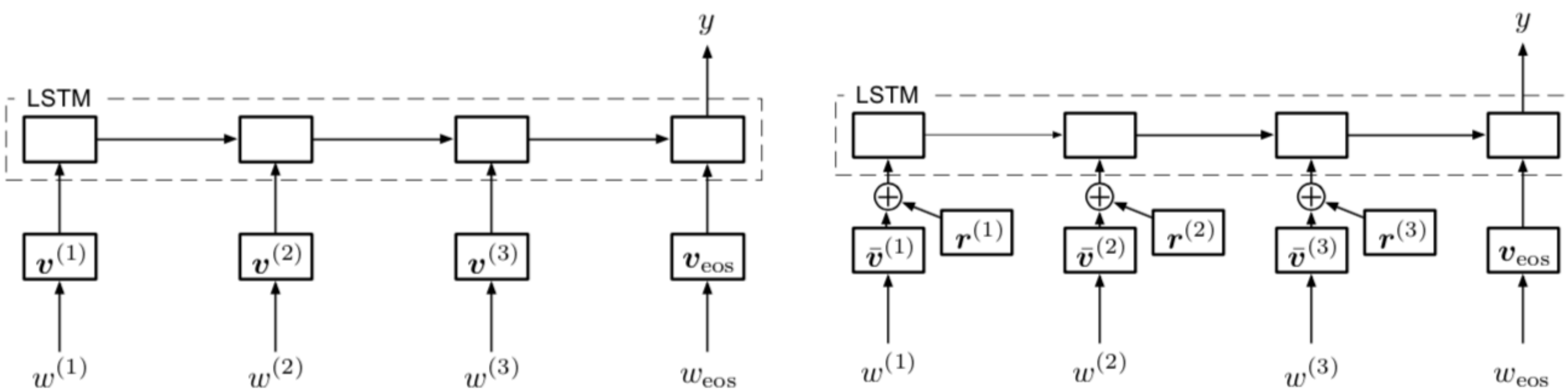
introduction

- adversarial training
  - origin sample / noise sample을 모두 정확하게 구분하는 model 만들기 위한 과정
  - label 필수
- virtual adversarial training
  - unlabeled sample
  - 모델의 regularization
    -> origin sample & noise sample 모두 같은 출력

악의적 input에 대한 방어로 작용

word embedding에 접근할 수 없기 때문에
classifier regularization으로서 분류 기능 안정화를 제시

method



(a) LSTM-based text classification model.

(b) The model with perturbed embeddings.

$$\bar{\boldsymbol{v}}_k = \frac{\boldsymbol{v}_k - \mathrm{E}(\boldsymbol{v})}{\sqrt{\mathrm{Var}(\boldsymbol{v})}} \text{ where } \mathrm{E}(\boldsymbol{v}) = \sum_{j=1}^{K} f_j \boldsymbol{v}_j, \mathrm{Var}(\boldsymbol{v}) = \sum_{j=1}^{K} f_j \left(\boldsymbol{v}_j - \mathrm{E}(\boldsymbol{v})\right)^2$$

method

adversarial training / loss

$$-\log p(y \mid \boldsymbol{x} + \boldsymbol{r}_{\text{adv}}; \boldsymbol{\theta}) \text{ where } \boldsymbol{r}_{\text{adv}} = \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \epsilon}{\arg\min} \log p(y \mid \boldsymbol{x} + \boldsymbol{r}; \hat{\boldsymbol{\theta}})$$

$$\boldsymbol{r}_{\text{adv}} = -\epsilon \boldsymbol{g} / \|\boldsymbol{g}\|_2 \text{ where } \boldsymbol{g} = \nabla_{\boldsymbol{x}} \log p(y \mid \boldsymbol{x}; \hat{\boldsymbol{\theta}}).$$

$$L_{\text{adv}}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^{N} \log p(y_n \mid \boldsymbol{s}_n + \boldsymbol{r}_{\text{adv},n}; \boldsymbol{\theta})$$

virtual adversarial training / loss

$$\mathrm{KL}[p(\cdot \mid \boldsymbol{x}; \hat{\boldsymbol{\theta}}) || p(\cdot \mid \boldsymbol{x} + \boldsymbol{r}_{\text{v-adv}}; \boldsymbol{\theta})]$$

$$\text{where } \boldsymbol{r}_{\text{v-adv}} = \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \epsilon}{\arg\max} \mathrm{KL}[p(\cdot \mid \boldsymbol{x}; \hat{\boldsymbol{\theta}}) || p(\cdot \mid \boldsymbol{x} + \boldsymbol{r}; \hat{\boldsymbol{\theta}})]$$

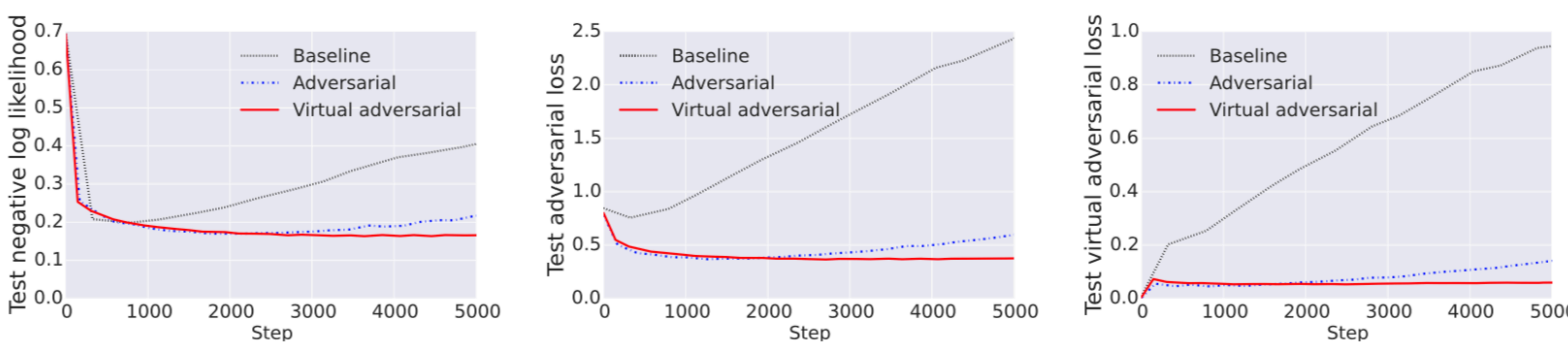
$$\boldsymbol{r}_{\text{v-adv}} = \epsilon \boldsymbol{g} / \|\boldsymbol{g}\|_2 \text{ where } \boldsymbol{g} = \nabla_{\boldsymbol{s}+\boldsymbol{d}} \mathrm{KL}\left[p(\cdot \mid \boldsymbol{s}; \hat{\boldsymbol{\theta}}) || p(\cdot \mid \boldsymbol{s} + \boldsymbol{d}; \hat{\boldsymbol{\theta}})\right]$$

$$L_{\text{v-adv}}(\boldsymbol{\theta}) = \frac{1}{N'} \sum_{n'=1}^{N'} \mathrm{KL}\left[p(\cdot \mid \boldsymbol{s}_{n'}; \hat{\boldsymbol{\theta}}) || p(\cdot \mid \boldsymbol{s}_{n'} + \boldsymbol{r}_{\text{v-adv},n'}; \boldsymbol{\theta})\right]$$

experiment

Table 1: Summary of datasets. Note that unlabeled examples for the Rotten Tomatoes dataset are not provided so we instead use the unlabeled Amazon reviews dataset.

	Classes	Train	Test	Unlabeled	Avg. T	Max T
IMDB	2	25,000	25,000	50,000	239	2,506
Elec	2	24,792	24,897	197,025	110	5,123
Rotten Tomatoes	2	9596	1066	7,911,684	20	54
DBpedia	14	560,000	70,000	–	49	953
RCV1	55	15,564	49,838	668,640	153	9,852



(a) Negative log likelihood (b) $L_{\text{adv}}(\boldsymbol{\theta})$ (c) $L_{\text{v-adv}}(\boldsymbol{\theta})$

experiment

Table 2: Test performance on the IMDB sentiment classification task. * indicates using pretrained embeddings of CNN and bidirectional LSTM.

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**%
Adversarial + Virtual Adversarial	6.09%
Virtual Adversarial (on bidirectional LSTM)	**5.91**%
Adversarial + Virtual Adversarial (on bidirectional LSTM)	6.02%
Full+Unlabeled+BoW (Maas et al., 2011)	11.11%
Transductive SVM (Johnson & Zhang, 2015b)	9.99%
NBSVM-bigrams (Wang & Manning, 2012)	8.78%
Paragraph Vectors (Le & Mikolov, 2014)	7.42%
SA-LSTM (Dai & Le, 2015)	7.24%
One-hot bi-LSTM* (Johnson & Zhang, 2016b)	5.94%

Table 4: Test performance on the Elec and RCV1 classification tasks. * indicates using pretrained embeddings of CNN, and † indicates using pretrained embeddings of CNN and bidirectional LSTM.

Method	Test error rate	
	Elec	RCV1
Baseline	6.24%	7.40%
Adversarial	5.61%	7.12%
Virtual Adversarial	5.54%	7.05%
Adversarial + Virtual Adversarial	**5.40**%	6.97%
Virtual Adversarial (on bidirectional LSTM)	5.55%	6.71%
Adversarial + Virtual Adversarial (on bidirectional LSTM)	5.45%	**6.68**%
Transductive SVM (Johnson & Zhang, 2015b)	16.41%	10.77%
NBLM (Naıve Bayes logisitic regression model) (Johnson & Zhang, 2015a)	8.11%	13.97%
One-hot CNN* (Johnson & Zhang, 2015b)	6.27%	7.71%
One-hot CNN† (Johnson & Zhang, 2016b)	5.87%	7.15%
One-hot bi-LSTM† (Johnson & Zhang, 2016b)	5.55%	8.52%

conclusion

- classification, word embedding에 뛰어난 성과
- 음성, 비디오와 같은 순차적 작업에 적용 가능성

code : https://github.com/tensorflow/models/tree/master/adversarial_text