Adversarial Training Methods for Semi-Supervised Text Classification

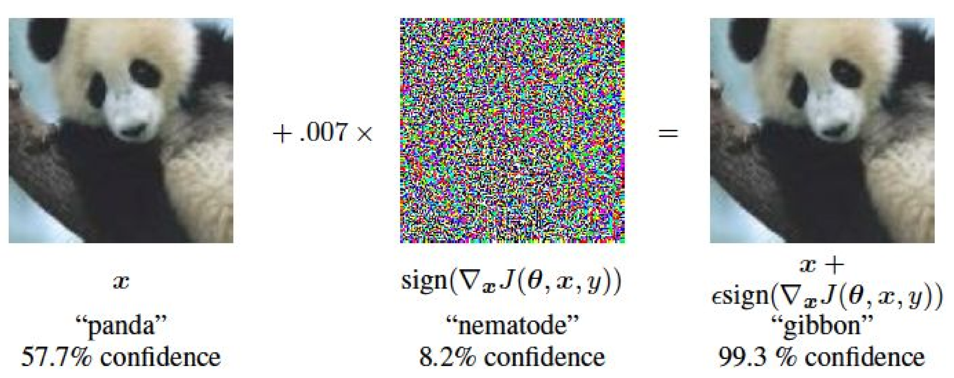
Takeru Miyato[(†)](Kyoto University, Google Brain), Andrew M. Dai(Google Brain), Ian Goodfellow[(*)](OpenAI)

takeru.miyato@gmail.com, adai@google.com, ian@openai.com

[(†)]present affiliation is Preferred Networks, Inc. and ATR cognitive mechanisms laboratories
[(*)]present affiliation is Google Brain

Adversarial examples and (Virtual) Adversarial training

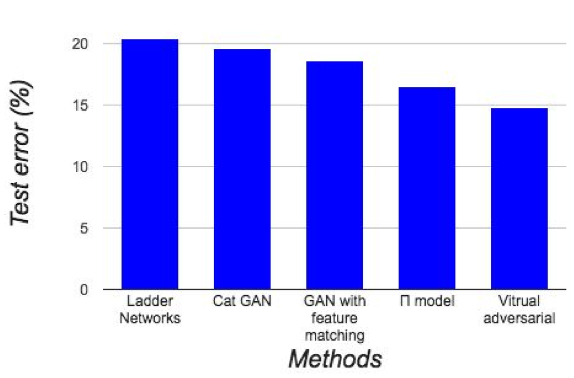
- *Adversarial examples* exist in the state of the art models on image domain.



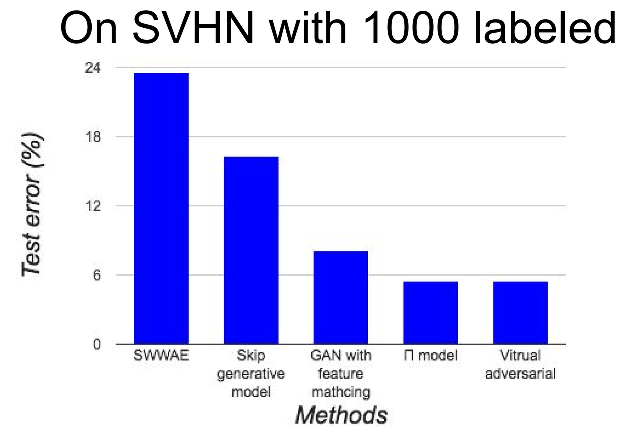
[Goodfellow, et al 2015]

- Training models to be robust to the adversarial examples (*Adversarial training*) improves generalization performance (Goodfellow et al 2015, Miyato et al 2016).
- *Virtual Adversarial Training* (Miyato et al, 2016) can be applied to **semi-supervised learning** tasks and achieves good performances on image classification tasks.

On CIFAR-10 with 4000 labeled



On SVHN with 1000 labeled



Adversarial training on text

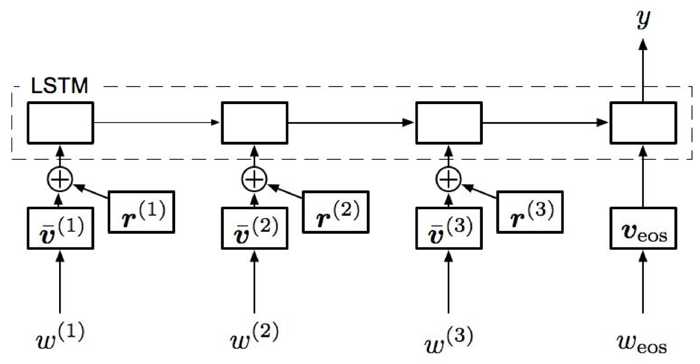
- Abundant unlabeled examples in text domain.
- Training on text can take a very long time (e.g. recurrent models).
  - (Virtual) Adversarial training requires little tuning of hyperparameters.
- **In our work, we applied adversarial and virtual adversarial training to semi-supervised text classification.**
  - **Achieved state of the art performance.**

Model

We use a simple LSTM model for text classification and define adversarial perturbations on its word embeddings, instead of the sequence of words.

- A sequence of T words : $\{w^{(t)}|t=1,\dots,T\}$
- Label : y
- Embedding matrix : $\boldsymbol{V} \in \mathcal{R}^{(K+1)\times D}$
- Normalized embeddings:

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



Proposed (Virtual) adversarial loss on the text classification model

Just add the below losses to the neg. log-likelihood!

- **Adversarial loss**

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

where

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

($\hat{\theta}$: a constant set to the current parameters
$\boldsymbol{s}$: a concatenation of a sequence of word embedding vectors $[\bar{v}^{(1)}, \dots, \bar{v}^{(T)}]$)

- **Virtual Adversarial loss**

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

(N' is the number of both labeled and unlabeled examples)

where

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

Experiments

- We first do pretraining following Dai and Le [2015] (recurrent language model).
  - This procedure is important to (virtual) adversarial training, because the perturbations on the initialized embeddings can be interpreted as the perturbations of "semantics".
- Dataset (4 semi-supervised and 1 supervised dataset):

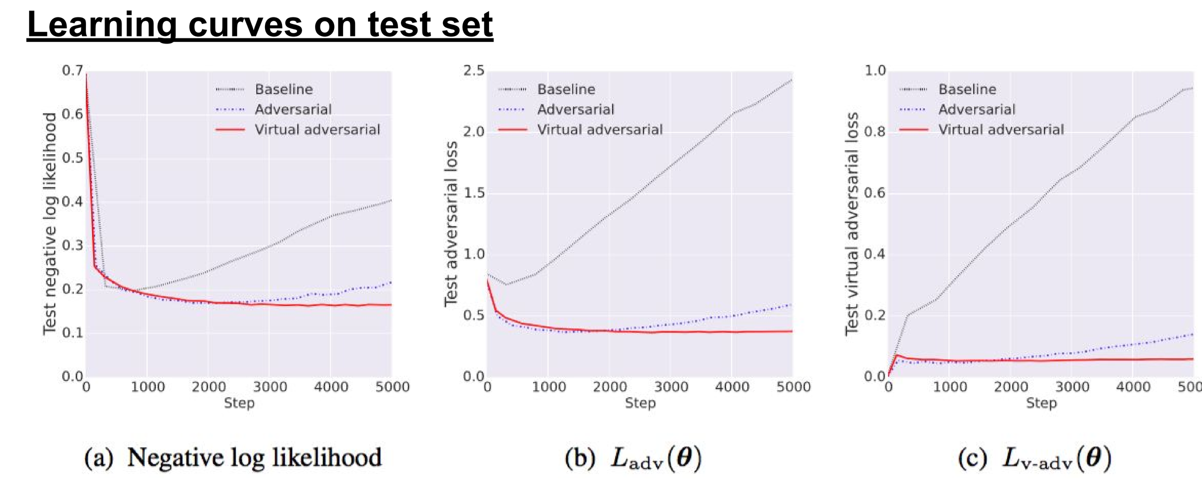
	Classes	Train	Test	Unlabeled	Avg. T	Max T
IMDB	2	25,000	25,000	50,000	239	2,506
Elec	2	24,792	24,897	197,025	110	5,123
Rotten Tomatoes	2	9596	1066	7,911,684	20	54
DBpedia	14	560,000	70,000	–	49	953
RCV1	55	15,564	49,838	668,640	153	9,852

- We optimized dropout rate on embeddings and norm constraint ε on adversarial and virtual adversarial training with each validation set.
- We used the method with only embedding dropout as the baseline.

Results on IMDB

- Virtual adversarial training achived comparable performance with the state of the art semi-supervised method.

Learning curves on test set



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

Test performance on IMDB

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%

- (Virtual) Adversarial training improve word embeddings in quality.
  - Adversarial training ensures that the meaning of a sentence cannot be inverted via a small change.

10 top nearest neighbors to 'good' and 'bad' with the trained word embeddings

	'good'				'bad'			
	baseline	**random**	**adversarial**	**virtual adversarial**	**baseline**	**random**	**adversarial**	**virtual adversarial**
1	great	great	decent	decent	terrible	terrible	terrible	terrible
2	decent	decent	great	great	awful	awful	awful	awful
3	×bad	excellent	nice	nice	horrible	horrible	horrible	horrible
4	excellent	nice	fine	fine	×good	×good	poor	poor
5	Good	Good	entertaining	entertaining	Bad	poor	BAD	BAD
6	fine	×bad	interesting	interesting	BAD	BAD	stupid	stupid
7	nice	fine	Good	Good	poor	Bad	Bad	Bad
8	interesting	interesting	excellent	cool	stupid	stupid	laughable	laughable
9	solid	entertaining	solid	enjoyable	Horrible	Horrible	lame	lame
10	entertaining	solid	cool	excellent	horrendous	horrendous	Horrible	Horrible

On other semi-supervised datasets (Elec, RCV1 and Rotten Tomatoes)

- Our proposed method achieved state of the art performance on both datasets.
  - However, The performance with virtual adversarial training was worse than the baseline on Rotten Tomatoes. We speculate that Virtual adversarial loss on unlabeled examples would overwhelm the supervised loss, and this would cause "wrong label" propagation.

Test performance on Elec and RCV1

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%

Test performance on Rotten Tomatoes

Method	Test error rate
Baseline	17.9%
Adversarial	16.8%
Virtual Adversarial	19.1%
Adversarial + Virtual Adversarial	**16.6%**
NBSVM-bigrams[29]	20.6%
CNN*[12]	18.5%
AdaSent*[32]	16.9%
SA-LSTM † [4]	16.7%

On supervised datasets (DBpedia)

- Achieved state of the art performace with virtual adversarial training.

Test performance on DBpedia

Method	Test error rate
Baseline (without embedding normalization)	0.87%
Baseline	0.90%
Random perturbation	0.85%
Adversarial	0.79%
Virtual Adversarial	**0.76%**
Bag-of-words(Zhang et al., 2015)	3.57%
Large-CNN(character-level) (Zhang et al., 2015)	1.73%
SA-LSTM(word-level)(Dai & Le, 2015)	1.41%
N-grams TFIDF (Zhang et al., 2015)	1.31%
SA-LSTM(character-level)(Dai & Le, 2015)	1.19%
Word CNN (Johnson & Zhang, 2016a)	0.84%

Conclusion

- Adversarial and virtual adversarial training are good regularizers for text classification tasks and achieved state of the art performance.
- With tuning of the additional hyperparameter ε , we can improve over the baseline and achieve state of the art performance.

References

- *Explaining and Harnessing Adversarial Examples*, I. Goodfellow, J. Shlens and C. Szegedy. ICLR 2015.
- *Virtual Adversarial Training : a Regularization Method for Supervised and Semi-supervised Learning*, T. Miyato, S. Maeda, M. Koyama and S. Ishii. arXiv preprint arXiv:1704.03976, 2017.
- *Distributional Smoothing with Virtual Adversarial Training*, T. Miyato, S. Maeda, M. Koyama, K. Nakae and S. Ishii. ICLR 2016.
- *Semi-supervied Sequence Learning*, A. M. Dai and Q. V. Le, NIPS 2015.