Distributional Smoothing with Virtual Adversarial Training,
Adversarial Training Methods for Semi-Supervised Text Classification

Miyato et al., ICLR 2016, 2017

Outline

- Semi-supervised learning
  - Small set of labeled + large set of unlabeled data
  - Supervised + unsupervised learnings
    - using pretrained word embeddings is implicitly semi-supervised learning
  - High performance with few labeled data
    - SOTA on IMDB, BERT-finetune, 200 vs 25000 labeled $\sim$ 95.8%
  - Consistency regularization and data augmentation
- Adversarial examples
- Virtual adversarial examples (this work)
- (Virtual) adversarial embeddings (this work)
  - one of few data augmentation methods for NLP,
  - internal rather external, no changes in context words
  - can improve performances of many NLP tasks, already text classification and relation extraction
- Benefits of (virtual) adversarial embeddings (this work)
  - encouraging generalization
  - Improving semantics of word embeddings while learning the task
- Experimental results on IMDB dataset

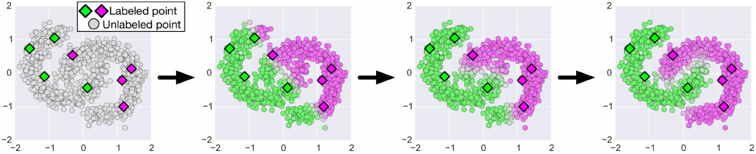
Consistency Regularization and Data Augmentation

- Consistency regularization

$$L(\boldsymbol{x}_l, y_l; \theta) = \mathcal{H}(q(y_l|\boldsymbol{x}_l),\ p(y|\boldsymbol{x}_l; \theta))$$

$$L_c(\boldsymbol{x}_{\mathrm{ul}}; \theta) = \mathrm{KL}(p(y|\boldsymbol{x}_{\mathrm{ul}}; \theta),\ p(y|\boldsymbol{x}_{\mathrm{ul}}^{+}; \theta))$$

  - regularize the model to produce similar output distributions for both original input $\boldsymbol{x}$ and augmented input $\boldsymbol{x}^+$ ($\boldsymbol{x}^+$ has to be "close" to $\boldsymbol{x}$ in input space)
  - labels are not needed for this regularizer term $\rightarrow$ it suits semi-supervised learning
  - intuitively,
    - ★ the model assumes pseudo-labels for unlabeled data and force its augmentation to match its output distribution
    - ★ labels are propagated from labeled to unlabeled examples



- Two ways of generating $\boldsymbol{x}^+$
  - Model's internal stochasticity: Π-Model, Mean Teacher, ICT

Adversarial Examples

- Adversarial examples:

$$L(\boldsymbol{x}_l, y_l; \theta) = \mathcal{H}(q(y_l|\boldsymbol{x}_l),\ p(y|\boldsymbol{x}_l; \theta))$$
$$\boldsymbol{\delta} = \mathrm{argmax}_{\boldsymbol{\delta} \in \mathcal{S}} L(\boldsymbol{x}_l + \boldsymbol{\delta}, y_l; \theta)$$

  - find perturbation $\boldsymbol{\delta}$ that changes the model's prediction, i.e., maximize the model's loss on given labeled input x_l
    - ★ $\boldsymbol{x}_{\mathsf{adv}} = \boldsymbol{x}_l + \boldsymbol{\delta}$ is the resulted adversarial example
  - control perturbation so that the adversaries are "close" to original example by a constraint $\mathcal{S}$, e.g., l_2-ball
    - ★ adversarial examples fit in the consistency regularization approach

- Approximate solution ($\mathcal{S} = l_2$): gradient descent

$$\boldsymbol{\delta} = -\epsilon \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}, \text{ where } \boldsymbol{g} = \nabla_x L(\boldsymbol{x}_l, y_l; \theta)$$

Virtual Adversarial Examples (this work)

- Adversarial examples

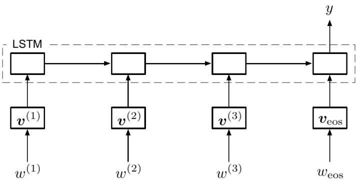
$$L(\boldsymbol{x}_l, y_l; \theta) = \mathcal{H}(q(y_l|\boldsymbol{x}_l),\ p(y|\boldsymbol{x}_l; \theta))$$
$$\boldsymbol{\delta} = \mathrm{argmax}_{\boldsymbol{\delta} \in \mathcal{S}} L(\boldsymbol{x}_l + \boldsymbol{\delta}, y_l; \theta)$$

- Virtual adversarial examples

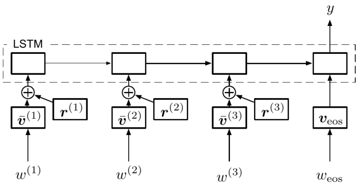
$$L_c(\boldsymbol{x}_{\mathrm{ul}}; \theta) = \mathrm{KL}(p(y|\boldsymbol{x}_{\mathrm{ul}}; \theta),\ p(y|\boldsymbol{x}_{\mathrm{ul}}^{+}; \theta))$$
$$\boldsymbol{\delta} = \mathrm{argmax}_{\boldsymbol{\delta} \in \mathcal{S}} L_c(\boldsymbol{x}_{\mathrm{ul}} + \boldsymbol{\delta}; \theta)$$

  - similar to adversarial examples, find the perturbation δ that maximizes the consistency regularizer L_c instead
  - labels are not needed
  - $\boldsymbol{x}^{+} = \boldsymbol{x}_{\mathsf{vadv}} = \boldsymbol{x}_{ul} + \boldsymbol{\delta}$ is the resulted virtual adversarial example

(Virtual) Adversarial Embeddings (this work)



(a) LSTM-based text classification model.

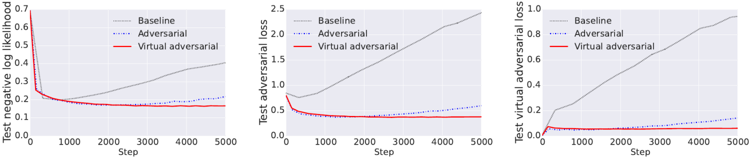


(b) The model with perturbed embeddings.

- (Virtual) Adversarial Examples
  - Adversarial perturbation: small changes to real-valued inputs
- (Virtual) Adversarial Embeddings for NLP
  - issue: discrete inputs, series of high-dimensional one-hot vectors $\rightarrow$ cannot directly compute perturbation
  - solution: compute perturbation on word embeddings instead of discrete word inputs

Benefits of (Virtual) Adversarial Embeddings

- Encouraging generalization



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

  - test loss consistently stays small

Benefits of (Virtual) Adversarial Embeddings

- Improving semantics of word embeddings during learning the task

	'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

  - baseline is strongly influenced by the grammatical structure of language, but not by the semantics of the task.
    - ★ e.g., "bad" appears in the list of nearest neighbors of "good"
  - (virtual) adversarial examples improve the word semantics
    - ★ e.g., "bad" no longer appears in the 10 top nearest neighbors to "good", falling to the 19th nearest neighbor
    - ★ (virtual) adversarial examples ensures that the meaning of a sentence cannot be inverted via a small change, so these words with similar grammatical role but different meaning become separated.

Experimental Results on IMDB Dataset

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%

Thank you !