Adversarial Training with Contrastive Learning in NLP

Daniela N. Rim, DongNyeong Heo, Heeyoul Choi

Francisco Javier Sáez Maldonado

March 15, 2022

Máster en Ciencia de Datos

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Introduction

- **Task**: Language modeling (LM) and Neural Machine Translation (NMT).

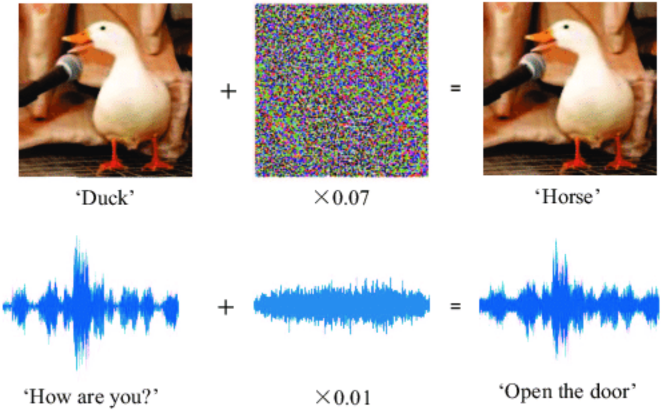
Introduction

- **Task**: Language modeling (LM) and Neural Machine Translation (NMT).
- **Goal**: Improve models so that they are semantically more robust:

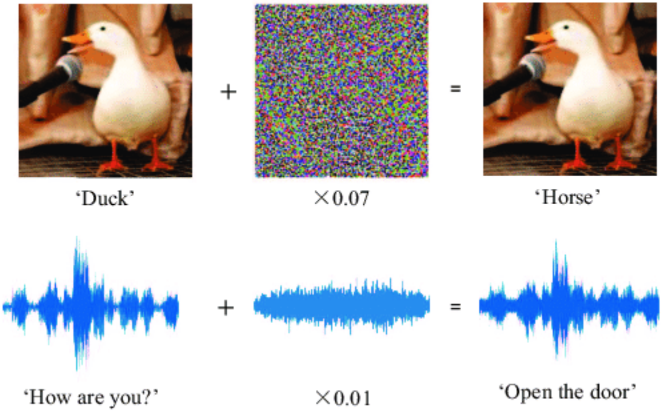
$$\text{Similar inputs} \implies \text{Similar outputs}$$

Tools

Adversarial Training



Adversarial Training



How does adversarial training help our models ?

Adversarial examples creation techniques

- Visual techniques: (Morris u. a., 2020).

Original Input	This film has a special place in my heart	Positive
Adversarial example	This film has a special plcae in my herat	Negative

Table 1: Example extracted from (Gao u. a., 2018)

- Semantic:(Jin u. a., 2019).

Original
<u>**Perfect**</u> performance by the actor → **Positive (99%)**

Adversarial
<u>**Spotless**</u> performance by the actor → **Negative (100%)**

Figure 1: Example extracted from (Jin u. a., 2019)

Our adversarial examples

Given a sequence of tokens $s = \{x_1, \ldots, x_T\}$

1. We map each discrete token to an embedded representation in the continuous space

$$\mathbf{E}x_i = e_i.$$

2. We add a little perturbation to the embedding

$$e_i' = e_i - \epsilon \frac{\nabla_{e_i} J(s, \theta)}{\|\nabla_{e_i} J(s, \theta)\|_2}.$$

Current loss function:

$$\mathcal{J}(\theta) = \sum_{s} \mathcal{L}(s, \theta) + \alpha \sum_{s'} \mathcal{L}_{adv}(s', \theta), \quad \alpha \in [0, 1].$$

Example

Original: Elephant. Positive: Tiger. Negative: Pizza.

Definition (Contrastive loss)

Let a_i be the original inputs, p_{a_i} positive examples and n_a negative examples. The contrastive loss is defined as follows:

$$\mathcal{L}_{cont} = -\sum_{a_i \in A} \log \frac{\exp(a_i \cdot p_{a_i}/\tau)}{\sum_{n_a \in A-\{a_i\}} \exp(a_i \cdot n_a/\tau)}$$

Framework

Adversarial Training with Contrastive Learning (ATCL)

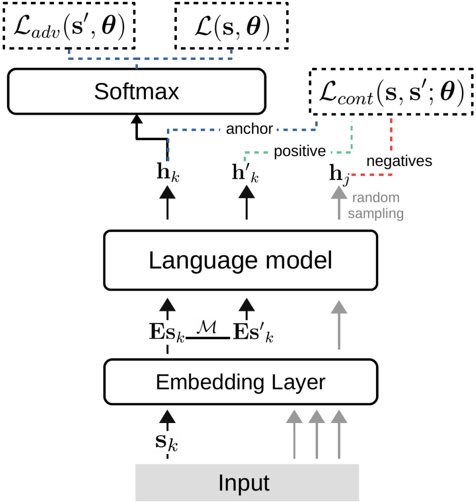


Figure 2: ACTL Framework. Image obtained from the original paper (Rim u. a., 2021).

Considerations about ATCL

Definition (ATCL's loss function)

The used loss function to train ATCL is:

$$\mathcal{J}(\theta)_{ATCL} = (\mathcal{L} + \alpha\mathcal{L}_{adv} + \beta\mathcal{L}_{cont})$$

Experiments and results

Language modeling

Model and Training	Wikitext 103		Penn Tree Bank	
	Validation	Test	Validation	Test
T XL base	23.10	24.00	56.72	54.52
T XL +TT	23.61	25.70	57.90	55.40
Baseline TT	22.70	22.42	41.91	36.13
Baseline +Adv. only	21.75	21.67	42.68	36.46
Baseline +ATCL (n=5)	22.79	22.59	37.93	32.89
Baseline +ATCL (n=10)	21.75	21.67	**35.29**	**29.08**
Baseline +ATCL (n=20)	**20.73**	**20.61**	42.52	36.85

Table 2: Perplexity achieved.

Language modeling

Word	Baseline	+Adv. only	+ATCL
friend	‘brother’, ‘daughter’, ‘understanding’, **‘director’**	‘cousin’, ‘colleague’, **‘knowledge’**, ‘mentor’	‘cousin’, ‘fellow’, ‘partner’, ‘colleague’
hate	‘admit’, ‘prejudice’, ‘troubled’, ‘regret’	**‘loving’**, ‘hurt’, **‘embrace’**, ‘committing’	‘dirty’, ‘poison’, ‘regret’, ‘shame’

Table 3: Four closest neighbors of some words in the vocabulary of the WikiText-103. Table from (Rim u. a., 2021).

Model name	En-De	De-En	En-Fr	Fr-En
Baseline (Transformer S)	24.61	30.34	35.23	34.51
Baseline +Adv	25.04	30.36	35.02	**35.97**
Baseline +ATCL	24.63	**31.34**	**36.40**	35.35
	25.26	30.36	35.60	35.48
	24.74	30.13	35.38	35.46

Table 4: BLEU test scores for the IWSLT dataset. Table from (Rim u. a., 2021).

Conclusions

- Usage of two general machine learning techniques applied to natural language processing.
- Applying this method has a similar effect to regularization.
- Results are promising in Language modeling, but not so much in Neural Machine Translation.

Thank you for your attention.

References

[Gao u. a. 2018] Gao, Ji ; Lanchantin, Jack ; Soffa, Mary L. ; Qi, Yanjun: Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. (2018). – URL http://arxiv.org/abs/1801.04354

[Jin u. a. 2019] Jin, Di ; Jin, Zhijing ; Zhou, Joey T. ; Szolovits, Peter: Is BERT Really Robust? Natural Language Attack on Text Classification and Entailment. (2019). – URL http://arxiv.org/abs/1907.11932

[Morris u. a. 2020] Morris, John X. ; Lifland, Eli ; Yoo, Jin Y. ; Qi, Yanjun: TextAttack: A Framework for Adversarial Attacks in Natural Language Processing. (2020). – URL https://arxiv.org/abs/2005.05909

[Rim u. a. 2021] Rim, Daniela N. ; Heo, DongNyeong ; Choi, Heeyoul: Adversarial Training with Contrastive Learning in NLP. (2021). – URL https://arxiv.org/abs/2109.09075

Elements

We consider:

- $\mathbf{S} = \{s_1, \ldots, s_B\}$ a set of sentences, where each sentence $s_k = \{x_{k1}, \ldots, x_{kN}\}$ contains N tokens.
- $\mathbf{E}s_k = \{\mathbf{E}x_{k1}, \ldots, \mathbf{E}x_{kN}\} = \{e_{k1}, \ldots e_{kN}\}$ the continuous space embedding of each sentence.
- The vocabulary $\mathcal{V}$ and a **restricted** subset from it in which we exclude incomplete words (single characters or symbols) $\mathcal{V}_R$
- The restriction function from $\mathcal{V}$ to $\mathcal{V}_R$:

$$\mathcal{M}(\mathbf{E}x_{ki}) = \begin{cases} 1 & \text{if } x_{ki} \in \mathcal{V}_R \\ 0 & \text{otherwise} \end{cases}$$

  (this function avoids taking senseless adversarial candidates).
- h_{kj} the representation of each sentence s_k.