FreeLB: Enhanced Adversarial Training for Language Understanding

Clment Royer, Christophe Roux, Pierre Guetschel

Sorbonne University

Adversarial Training

The goal of **adversarial training** is to enhance the robustness of a model by training it on adversarial examples. Adversarial examples are created by adding small perturbations to the data which cause misclassification. In the context of **Natural Language Processing** tasks, the inputs are discrete (sequences of word ids). Hence they can't be directly perturbed. Instead the adversarial perturbations are added to the word embeddings.

Mathematical formulation

Adversarial training consist of solving the following optimization problem:

$$\min_{\theta} \mathbb{E}_{(X,y)\sim\mathcal{D}} \left[\max_{||\delta||\leq\epsilon} L(f_\theta(X+\delta), y) \right]$$

With the adversarial perturbation δ, the perturbation bound ϵ, the loss function L , the label y,the input data X, the parameters of the classifier θ and the data $\mathcal{D}$.

Free adversarial training

Canonical **K-PGD** finds the inner-max by computing K projected gradient ascent steps. **FreeAT**[1] and **FreeLB**[2] attempt to reduce the complexity of K-PGD[3]. At each iteration, they use the previous backward step of the descent step to compute the ascent step. Their complexity is comparable to natural training.

Training the model

We use a pre-trained transformer-based neural network named **Distilbert**[4] from the 'Huggingface'[5] repository, which is a lightweight alternative to BERT. We then fine-tune the model using the different adversarial learning methods. The classifier is trained on the **Stanford Sentiment Treebank**[6] dataset. The task is to predict the sentiment of a given sentence. We use a binary (positive/negative) class split, and use only sentence-level labels.

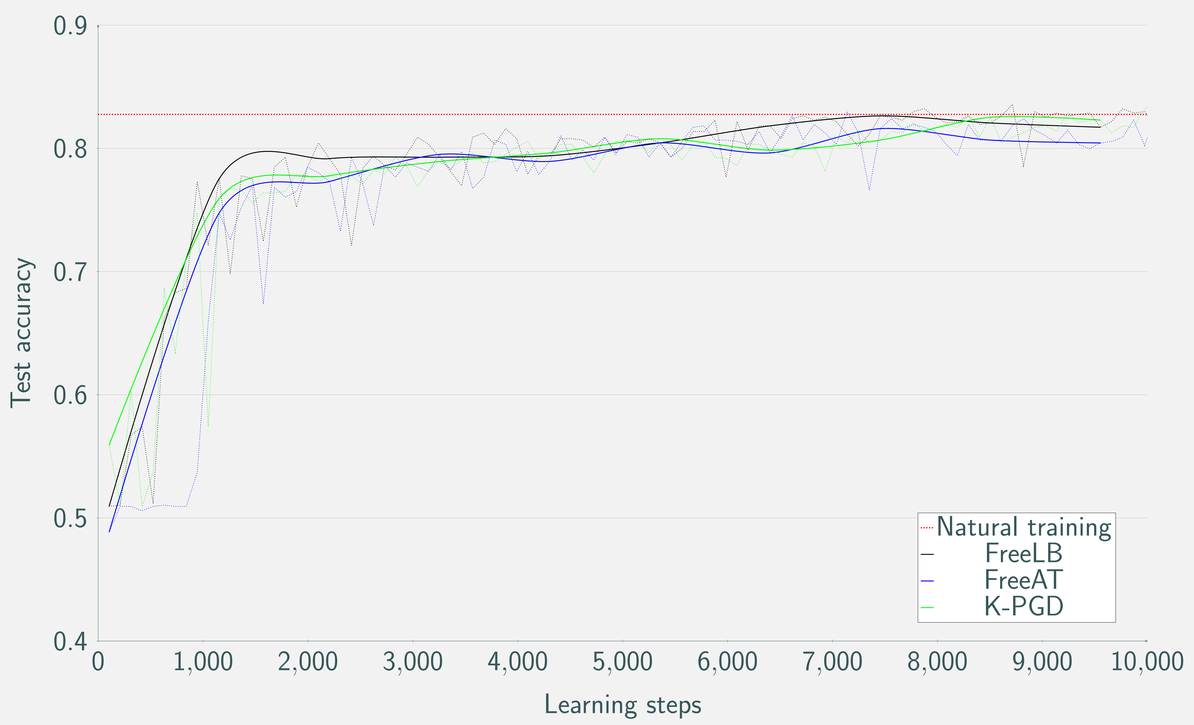
K-PGD pseudo-code

Input: $\mathcal{D}, \epsilon, \tau, K, \alpha$
Initialize θ
for *N epochs* **do**
  for *batch* $X \in \mathcal{D}$ **do**
    $\delta_0 = \mathcal{U}(-\epsilon, \epsilon)$
    for $t = 1, ..., K$ **do**
      $g_{adv} = \nabla_\delta L(f_\theta(X+\delta_{t-1}), y)$
      $\delta_t = \textbf{clip}\,(\delta_{t-1} + \alpha \cdot \textbf{sgn}(g_{adv}), -\epsilon, \epsilon)$
    end
    "Create one adversary with K ascent steps, train the network on this adversary"
    $\theta = \theta - \tau \nabla_\theta L((f_\theta(X+\delta_K), y)$
  end
end

FreeLB pseudo-code

Input: $\mathcal{D}, \epsilon, \tau, K, \alpha$
Initialize θ
for *N epochs* **do**
  for *batch* $X \in \mathcal{D}$ **do**
    $\delta_0 = \frac{1}{\sqrt{N_\theta}} \cdot \mathcal{U}(-\epsilon, \epsilon), \quad g_0 = 0$
    for $t = 1, ..., K$ **do**
      $g_t = g_{t-1} + \frac{1}{K} \cdot \nabla_\theta L(f_\theta(X+\delta_{t-1}), y)$
      "Compute ∇_δ by reusing ∇_θ"
      $g_{adv} = \nabla_\delta L(f_\theta(X+\delta_{t-1}), y)$
      $\delta_t = \Pi_{||\delta||\leq\epsilon}(\delta_{t-1} + \alpha \cdot \frac{g_{adv}}{||g_{adv}||})$
    end
    "Train the network on the accumulated gradients w.r.t. to the adversaries at each step"
    $\theta = \theta - \tau g_K$
  end
end

Numerical results



Hyperparmeters

Parameters	K-PGD	FreeAT	FreeLB
α	16e-5	-	1e-1
ϵ	5e-4	5e-4	6e-1
τ	2e-5	2e-5	2e-5
K	3	3	2

Conclusion

- On NLP tasks, adversarial training might be viewed as a form of regularisation
- The advantage over other "free" training algorithms is small
- We were not able to find a significant performance difference in our experiments

References

[1] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein.
Adversarial training for free!

[2] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu.
Freelb: Enhanced adversarial training for language understanding.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu.
Towards deep learning models resistant to adversarial attacks.

[4] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf.
Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

[5] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew.
Huggingface's transformers: State-of-the-art natural language processing.

[6] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts.
Recursive deep models for semantic compositionality over a sentiment treebank.
EMNLP, 1631:1631–1642, 01 2013.