Deep Learning Project: Attacking and Defending Neural Networks

Group 12

Team Members: Arda Güney, student number 3391280
Diego Dalsasso, student number 3549305
January 31, 2025

Abstract

This report presents a study done on a pre-trained deep learning model to explore how an attack and defense to a deep learning model can lead to increase the robustness of the network. The pre-trained model used to experiment on, is a T5, an NLP model for Portuguese-English and English-Portuguese translation tasks. To perform the attack we exploited embedding-level perturbations and gradient analysis (famous as FGSM attack). To defend the network, a gradient-based regularization with input noise addition have been used to reduce the effects of the attack. The success of the models translations and the effects of attack and defense is assessed using BLEU score, which is a common metric to evaluate machine translation performance.

Index Terms

Adversarial Attacks, T5 architecture, BLEU Score, Deep Learning, FGSM attack, Gradient-based regularization, NLP, Machine Translation.

1 Introduction

Deep learning models are neural network models that are composition of many non-linear functions, to model the complex dependency between input features and labels. This method had show significant success on various fields such as computer vision and natural language processing[2]. In scope of this study a deep learning model that is based on an architecture called T5 (Text-to-Text Transfer Transformer) is examined. This is a generative neural network Transformer-based, so it presents the encoder-decoder architecture, introduced by Google in 2019[1]. Thus it allows simple but powerful natural language processing models to be built. However the success and efficiency of deep learning models are undoubted in many domains such as natural language processing this does not mean that they don't comprise any challenge. Despite the impressive performances DNN's have, they are still vulnerable to attacks such as adversarial samples [4]. Very often is possible to see attacks performed on Image Classification Networks adding noise to the inputs and acting on the gradients but is not so common to perform this kind of attacks on Text to Text Networks. So in this work we want to apply what we have studied in the theory, performing an FGSM attack to the given pretrained network to see how much the degradation of the traslation is effective, and explore if (and how) a gradient-based regularization is sufficient for defending from such kind of an attack.

2 Methodology

2.1 Model Setup and Dataset

2.1.1 Model

In this study, we used the T5 (Text-to-Text Transfer Transformer) model for the languages Portuguese and English that can perform translation in both directions. The transformed T5 is a state-of-the-art transformer-based model that frames every NLP task as a text generation problem, where input text is mapped to output text. As propsed from the builders of the model, this approach is well suited for translation tasks, especially when the development of high-quality translation models is still expensive. We have used directly a T5 model fined by the usage of a different English tokenizer useful for some special characters[5]. All what is regarding the model is shared by the authors of the paper in appendix, such as tokenizer, weights and test set.

2.1.2 Model Setup

In order to implement and use the translation model T5 we used the pre-trained model Portuguese-to-English. Model has fined tuned specifically for translation tasks and as the authors have provided all the necessary files, we can load the model locally and access weights and test set to perform our experiments. Remaining of this section will explain the set-up process step by step:

Firstly, we acquire model and tokenizer from the current directory so we are sure we have access to all what we need to perform a white box attack.

After the set-up initialization the process could by handled through having a text prompt as input. Then, the tokenized text is passed trough the model and translated back into the natural language form that humans can understand.

2.1.3 Dataset

The model originally is trained by using six different datasets, and is evaluated on two datasets: the WMT19 Biomedical Translation Task dataset and a subset of 99K sentence pairs of the ParaCrawl dataset. This second dataset is the one we are going to use to perform the attack and to evaluate the performances. ParaCrawl99k is a dataset that allows you to comparing your result to Google Translate (GT). It is fully available to allow you to compare your translations with Google Translate. The author removed data from testing that had a jaccard similarity with any sentence in training larger than 0.7. Therefore they shared the final 99k as a test set. [5].

3 Attack and Defense Algorithms

3.1 Adversarial Attack

The attack we performed is an adversarial attack based on Fast Gradient Signed Method (FGSM), a type of non targeted, white box attack, as we had access to the model's structure and weights. FGSM is a straightforwards approach that generates adversarial samples. The algorithm uses the gradient generating in the back-propagation neural network to perturb the input data so that although the perturbation is unrecognizable, input data cannot be significantly different [3]. Thus, we generate adversarial samples by perturbing the input in the direction of the gradient of the loss function, with respect to the input. FGSM method modifies the input embeddings rather than the input text directly, targeting the model's sensitivity to small changes in its internal representation.

3.1.1 Attack Setup

The test set was derived from a Pickle file that contained pairs of input sentences

and their corresponding target translations. However, due to the limited computational resource and time a subset of data that had 100 samples was used for the analysis.

3.1.2 Gradient Calculation

The source and input sentences were tokenized for each input. The model's encoder enabled us to reach the input embeddings, and by setting some parameters for embeddings it was possible to track the gradients.

Afterwards, we successfully computed the loss based on the output and the target translation by running the model forward thanks to these embeddings we have reached. So we derived the gradients of the loss with respect to the input embeddings.

3.1.3 Perturbation Application

Taking the direction of the gradient into account, a small perturbation was added to the input embeddings. By this way our FGSM attack occurred. The "`epsilon`" parameter was used for determining the change of the perturbation done to the embeddings and controlled the attack. The mathematical formula behind this perturbation applied to the embeddings can be explained as below:

$$\mathbf{a} = \mathbf{e} + \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{e}} \mathcal{L}\right)$$

- $\mathbf{a}$ represents the adversarial embeddings,
- $\mathbf{e}$ represents the original embeddings,
- ϵ is the perturbation size,
- sign is the sign function,
- $\nabla_{\mathbf{e}} \mathcal{L}$ is the gradient of the loss with respect to the embeddings.

3.1.4 Evaluation

To observe the effects of the adversarial attack, translations were generated for both untouched and the perturbed embeddings of inputs. Afterwards by using the token ID's of the model, each translation turned into natural language form that can be understood by humans.

In continuation, after deriving the translations from both kind of embeddings, a very well known evaluation method called BLEU Score has used for evaluating the performance of untouched and perturbed embeddings. Bilingual Evaluation Understudy (BLEU) is an evaluation method that is used for machine-generated translations and works by comparing them to reference translations.

First of all the candidate and reference translations are prepared where candidate translations are the output from the machine translation model and reference translations are the candidate translation that are used as the ground truth, provided by one or more humans.

Secondly, n-gram precision is utilized to evaluate the quality of candidate translations. BLEU measures the precision of n-grams (word sequences) in the candidate translation that match those in the reference translation. This approach ensures that only n-grams present in both the candidate and reference translations are considered, effectively preventing inflation in the evaluation of the candidate translation.

Below is the formula for modified precision of n grams of size n:

$$P_n = \frac{\sum_{C \in \text{candidate}} \min(\text{count}(C, n), \text{max\_count}(R, n))}{\sum_{C \in \text{candidate}} \text{count}(C, n)} \quad (1)$$

- **count**(C, n)**:** number of an n-gram in the candidate translation.
- **max_count**(R, n)**:** Maximum count of the same n-gram in any reference translation.

Afterwards, BLEU combines the precision of different sized n-grams like unigrams, bigrams etc. The result from this process is

called the Geometric Mean of N-Gram Precision which the formula of it placed below:

$$\text{Geometric Mean} = \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log P_n\right)$$

- N represents the number of highest n-gram size.

Trough this step penalization of any n gram's score occurs if it is lower than the calculated geometric mean across all n grams.

Afterwards Brevity Penalty (BP) is applied to prevent any short translations to achieve high scores. This process done by the comparison of the length of the candidate translation ($|C|$) with the length of the closest reference translation ($|R|$).

Below the formula of (BP) can be seen:

$$BP = \begin{cases} 1 & \text{if } |C| > |R| \\ \exp(1 - \frac{|R|}{|C|}) & \text{if } |C| \leq |R| \end{cases} \quad (2)$$

Lastly, the final BLEU score is calculated by multiplying the Brevity Penalty (BP) with the Geometric Mean of N-Gram Precision as in the formula below:

$$\text{BLEU} = BP \cdot \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log P_n\right)$$

By using the BLEU score, that is calculated as explained above, the adversarial and original embeddings are successfully compared [7].

An example of the effectiveness of our attack it's shown in Table 1. We noticed that, as the magnitude of the gradients of such a big network is quite small (order of 10^{-5}) the big enough "`epsilon`" to make show up a distortion in the traslation is 5.0.

Table 1: Example of adversarial attack on embeddings performed on this model.

	Text
Input	Eu gostaria de me casar com você e viver minha vida com você.
Target	I would like to marry you and live my life with you.
No attack	I would like to marry you and live my life with you.
Attack ($\epsilon = 5.0$)	I would like to get married with you and live my.

3.2 Defense Algorithm

For the explained adversarial attack based on FGSM in the previous section there may several applicable defense method to protect the network and lower the effects of the adversarial attack. Some of the applicable defenses are adversarial training, data augmentation and gradient based approaches such as gradient regularization.

Let's explain these methods briefly. Adversarial training is a method that works by training a schema that utilizes an alternative objective function to support the model for adversarial attacks by letting the model generalize over adversarial data as well as unperturbed data [9]. Another method is data augmentation, this defends the system by preprocessed input samples that are generated specifically for training the model before inference to adversarial perturbations in order to overcome them. By obfuscating the gradients of DNN models based on the data generated for train it against the perturbations, these approaches can defeat a considerable number of conventional attacks[8]. Lastly, a gradient based method such as **Gradient regularization** works through adjusting the gradients during training. By effectively penalizing, limiting the gradients through loss function with respect to inputs. This method changes the models behavior by preventing gradients becoming large gradients to take over and offers a unified view to deal with adversarial examples[6].

In this study, gradient regularization plus input noise addition were chosen as the defense mechanism against adversarial attacks due to

their efficiency and ease of implementation. The first approach works by directly manipulating the gradients through straightforward modifications to the network's code, making it significantly simpler to implement and experiment with compared to alternative methods such as data augmentation or adversarial training. Unlike these approaches, which often require substantial modifications to the dataset or model architecture and involve considerable computational overhead during the training process, gradient regularization provides an effective defense with minimal additional resources, facilitating faster experimentation and implementation. Input noise addition is a technique used to improve the robustness of machine learning models. This involves introducing random noise (e.g., Gaussian or uniform noise) to the inputs or intermediate representations (like embeddings) during training or inference. This added noise simulates variations in the input data and encourages the model to learn features that are less sensitive to small perturbations.

3.2.1 Gradient Regularization Method

By gradient regularization we tried to overcome the over-fitting caused by the adversarial example during the forward pass. It adds a penalty for large gradients to the loss, encouraging the model to produce smoother gradients. This helps mitigate the effect of adversarial perturbations. To implement this, firstly the loss between the models predicted translation and target translation is calculated by the use of original embeddings. Afterwards, the gradients of the loss is calculated by using back-propagation. Back-propagation reveals the highly interconnected nature of the network and its sensitivity to perturbations applied to the embeddings. In continuation the L2 norm of the gradients is computed in order to add a regularization term on the total loss. The regularization term can be mathematically expressed as $\lambda \cdot \|\nabla_{\mathbf{e}}\mathcal{L}\|_2$, where λ is the regularization coefficient and $\nabla_{\mathbf{e}}\mathcal{L}$ is the gradient of the loss with respect to the embeddings. Due to this term the large gradients are penalized and decreases the effects of perturbations over the actual embeddings occurred during the adversarial attack. In the end the new, total loss after gradient regularization would be:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{original}} + \lambda \cdot \|\nabla_{\mathbf{e}}\mathcal{L}\|^2$$

Where:

- $\mathcal{L}_{\text{original}}$ is the standard loss,
- $\mathcal{L}_{\text{total}}$ is the modified loss,
- $\lambda \cdot \|\nabla_{\mathbf{e}}\mathcal{L}\|^2$ is the regularization term.

This eventually lead the deep neural network to not take the noisy gradients in consideration as much as compared to the version where there was no gradient regularization. Thus would expected to mitigate the effects of the FGSM attack.

3.2.2 Input noise addition

As already mentioned, input noise addition introduces random noise (e.g., Gaussian or uniform noise) to the inputs or intermediate representations (like embeddings) during training or inference to improve the robustness of the performance. In this specific case we have injected a random Gaussian noise with 0-mean and a standard deviation of 0.1. The important parameter σ (std deviation) is the one that allows us to control the magnitude of noise and should be properly tuned. It is important that noise should be applied to row features, so usually, in image classification networks, noise is applied to the pixels. In this case noise is applied to the embeddings. Noise addition prevents the model from relying on specific input details, making it less vulnerable to adversarial attacks or small input changes. Noise injection is a regularization technique very useful against FGSM, combined with gradient regularization.

3.2.3 Final Defended Embedding

In the end the final mathematical formulation of the defence is the following. We introduced a regularization term to penalize large gradient magnitudes in the embedding space:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{original}} + \lambda \cdot \|\nabla_{\mathbf{e}} \mathcal{L}\|^2$$

where λ is a hyperparameter controlling the regularization strength.

We add Gaussian noise to the embeddings to enhance robustness:

$$\mathbf{e}^{noise} = \mathbf{e} + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2 I), \qquad (3)$$

where σ is the standard deviation of the noise.

Combining both defenses, we obtain the final embedding used for translation:

$$\mathbf{e}^{def} = \mathbf{e}^{noise} - \lambda \cdot \nabla_{\mathbf{e}} \mathcal{L}. \qquad (4)$$

This formulation ensures that the embeddings are both regularized and resilient to small adversarial perturbations, improving the model's robustness against FGSM attacks.

4 Results and Discussion

The results of our experiments on adversarial attacks and defenses applied to the T5 text-to-text translation model are summarized in Table 3. We measured the impact of the FGSM attack at varying levels of perturbation intensity (ε) and evaluated the effectiveness of the gradient regularization defense strategy. We must specify that the model we downloaded and used for this experiments is reported performing with a BLEU score around 45 under some conditions, regarding Portuguese-English traslation on the ParaCrawl dataset. What we got, after running many times the model locally, is an average score of 53.48. We performed the translation on a subset of the 99K sentences present in the dataset due to computational power we had at disposal. In particular we used 1000 sentences each time randomly taken from the big dataset. However, the adversarial attack have been successfully implemented and also the defence, as reported in the data below.

Table 2: For observing the effects of parameter ϵ. BLEU scores under FGSM attack and after applying gradient regularization defense while ϵ is changing and λ stable at 5.0

ε	**Original**	**Post-Attack**	**Post-Defense**
1.0	53.48	44.17	58.77
3.0	53.48	42.72	46.71
5.0	53.48	42.72	46.71
10.0	53.48	13.88	46.71
15.0	53.48	10.88	21.10

Table 3: For observing the effects of parameter λ. BLEU scores under FGSM attack and after applying gradient regularization defense while λ is changing and ϵ stable at 3.0

λ	**Original**	**Post-Attack**	**Post-Defense**
1.0	53.48	42.72	44.17
3.0	53.48	42.72	49.39
5.0	53.48	42.72	49.39
10.0	53.48	42.72	13.48
15.0	53.48	42.72	13.13

Table 2 highlights the significant drop in BLEU scores as the perturbation intensity increases, demonstrating the model's vulnerability to adversarial attacks. For $\varepsilon = 3$, $\varepsilon = 5$ and$\varepsilon = 10$ the defense strategy successfully restored a large portion of the BLEU score, achieving a near-original performance. Particularly for $\varepsilon = 1$ and $\lambda = 5$ the strength of regularization also improves the performance of original model and increases the BLEU score to 58 from 53. However, for higher ε values such as 15, the performance of defense is significantly low as 21 compared to original score 53 even it is higher than the attacked score 10 which emphasizing the need for further robust defense mechanisms to get a closer result to original score. $\varepsilon = 5$ is a good threshold to see the effectiveness of the attack as the magnitude of the embeddings is

of the order 10^1, so we need a large enough perturbation.

Secondly, Table 3 presents the effect of λ which is the parameter responsible from the strength of the gradient regularization. As expected to a certain limit while the size of λ increases the BLEU score of defended model from the attack also increases. However after very large values such as $\lambda = 10$ the defended models score drops even more than the attacked model. This situation points out that power of the regularization could degrade performance after a limit because of over suppression of the gradients. Thus as the table indicates a value of $\lambda = 3$ or $\lambda = 5$ performs just as expected and performs a successful defense that leads to a close BLEU score to the original model after attack.In the end the highest BLEU score of the defended model after attack is 49.39 while $\lambda = 3$ or $\lambda = 5$ and $\varepsilon = 3$.

5 Conclusion

The improvements in the deep learning field attracts attention from different actors with different intentions and therefore it is becoming a vital need to defend the deep neural network from the increasing and improving threats. In this paper we tried to demonstrate attack and defense operation to a deep neural network. In this demonstration, we implemented an attack algorithm that changes target's specific features in such a way that the system can misunderstand it and therefore causing a wrong translation. Then we implemented a defense algorithm to make the target embeddings more robusts. The experiment results were sufficient with a 4% loss related to the the BLEU score. The use of gradient-based methods allows for efficient generation of adversarial examples without requiring extensive computational resources. This experiment and demonstration shows that a deep neural network can be defended by taking preventive measures in the network and therefore contributes to the further researches in this field of deep neural network security.

6 Use of AI

During the preparation of this work, the author(s) used no artificial intelligence tools.

References

[1] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[2] J. Fan, C. Ma, and Y. Zhong. "A selective overview of deep learning". In: *Statistical Science: A Review Journal of the Institute of Mathematical Statistics* 36.2 (2021), pp. 264–290. DOI: 10.1214/20-sts783.

[3] Zihua Fang. *Analysis of adversarial attack based on FGSM*. Ed. by Kannimuthu Subramanian. International Society for Optics and Photonics, 2023. DOI: 10.1117/12.2656064. URL: https://doi.org/10.1117/12.2656064.

[4] Linyang Li et al. *BERT-ATTACK: Adversarial Attack Against BERT Using BERT*. 2020. arXiv: 2004.09984 [cs.CL]. URL: https://arxiv.org/abs/2004.09984.

[5] Alexandre Lopes et al. *Lite Training Strategies for Portuguese-English and English-Portuguese Translation*. Online, Nov. 2020. URL: https://www.aclweb.org/anthology/2020.wmt-1.90.

[6] Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. *A Unified Gradient Regularization Family for Adversarial Examples*. 2015. DOI: 10.1109/ICDM.2015.84.

[7] Kishore Papineni et al. *BLEU: a method for automatic evaluation of machine translation*. Philadelphia, Pennsylvania, 2002. DOI: 10.3115/1073083.1073135. URL: https://doi.org/10.3115/1073083.1073135.

[8] Han Qiu et al. "FenceBox: A Platform for Defeating Adversarial Examples with Data Augmentation Techniques". In: *ArXiv* abs/2012.01701 (2020). URL: https://api.semanticscholar.org/CorpusID:227254219.

[9] Weimin Zhao, Sanaa Alwidian, and Qusay H. Mahmoud. "Adversarial Training Methods for Deep Learning: A Systematic Review". In: *Algorithms* 15.8 (2022). ISSN: 1999-4893. DOI: 10.3390/a15080283. URL: https://www.mdpi.com/1999-4893/15/8/283.