Accuracy of TextFooler black box adversarial attacks on 01 loss sign activation neural network ensemble

1st Yunzhe Xue
Department of Data Science
New Jersey Institute of Technology
Newark, NJ, USA
yx277@njit.edu

2nd Usman Roshan
Department of Data Science
New Jersey Institute of Technology
Newark, NJ, USA
usman@njit.edu

***Abstract*—Recent work has shown the defense of 01 loss sign activation neural networks against image classification adversarial attacks. A public challenge to attack the models on CIFAR10 dataset remains undefeated. We ask the following question in this study: are 01 loss sign activation neural networks hard to deceive with a popular black box text adversarial attack program called TextFooler? We study this question on four popular text classification datasets: IMDB reviews, Yelp reviews, MR sentiment classification, and AG news classification. We find that our 01 loss sign activation network is much harder to attack with TextFooler compared to sigmoid activation cross entropy and binary neural networks. We also study a 01 loss sign activation convolutional neural network with a novel global pooling step specific to sign activation networks. With this new variation we see a significant gain in adversarial accuracy rendering TextFooler practically useless against it. We make our code freely available at https://github.com/zero-one-loss/wordcnn01 and https://github.com/xyzacademic/mlp01example. Our work here suggests that 01 loss sign activation networks could be further developed to create fool proof models against text adversarial attacks.**

***Index Terms*—text classification, 01 loss neural networks, black box adversarial attack**

I. Introduction

Adversarial attacks remain a security vulnerability in neural networks today since they were first discovered [1]. A recent paper evaluating the operational feasibility of adversarial attacks in military defense [2] found that patch attacks posed a minimal danger in practice but both white and black-box attacks can be significantly more effective - a white box attack reduces the target model's accuracy by 65% and a black-box attack reduces it by 55% to 63% depending upon knowledge and access to training data and target model architecture. Various defenses have been developed and broken [3]. Adversarial training [4], which is to train the model with clean and adversarial data, remains the most effective solution to date but it is computationally expensive and lowers clean test accuracy [5], [6].

Recent work has shown that 01 loss sign activation networks are hard to attack in image classification datasets [7]–[12]. In this paper we investigate its robustness against text black box adversrial attacks.

The TextFooler [13] method is designed to find syntactically and semantically similar adversarial documents by replacing important words with similar ones until the document is misclassified. We attack models with TextFooler on four document classification datasets: Internet Movie Database (25K train, 25K test, mean words per document=215) and Yelp (560K train, 38K test, mean words per document=152) positive and negative reviews (IMDB and Yelp), sentence classification of positive and negative sentiments (9K train, 1K test, mean words per document=20, denoted as MR), and sentence-level classification of news items in World and Sports categories (120K train, 7.6K test, mean words per document=43, denoted as AG) [13].

II. Methods

A. Gradient-free stochastic coordinate decent for sign activation 01 loss network

Suppose we are given binary class data $x_i \in R^d$ and $y_i \in \{-1, +1\}$ for $i = 0...n-1$. Consider the objective function of a single hidden layer neural network with sign activation and 01 loss given below. We employ a stochastic coordinate descent algorithm shown in Algorithm 1 (similar to recent work [7]–[9]) to minimize this objective.

$$\frac{1}{2n} \underset{W,W_0,w,w_0}{\arg\min} \sum_i (1 - sign(y_i(w^T(sign(W^T x_i + W_0)) + w_0))) \tag{1}$$

We can train sign activation networks with and without binary weights using our SCD training procedure above. In the case of binary weights we don't need a learning rate. We apply GPU parallelism to simultaneously update features and other heuristics to speed up runtimes.

B. Implementation, test accuracy, and runtime

We implement our training procedure in Python, numpy, and Pytorch [14]. and make our code freely available from https://github.com/zero-one-loss/wordcnn01 and https://github.com/

Algorithm 1 SCD01: Stochastic coordinate descent for sign activation 01 loss single hidden layer network

Procedure:

1. Initialize all network weights W, w to random values from the Normal distribution $N(0, 1)$.

2. Set network thresholds W_0 to the median projection value on their corresponding weight vectors and w_0 to the projection value that minimizes our network objective.

while $i < epochs$ **do**

1. Randomly sample 75% of data equally from each class.

2. Perform coordinate descent separately first on the final node w and then a randomly selected hidden node u (a random column from the hidden layer weight matrix W)

3. Suppose we are performing coordinate descent on node w. We select a random set of features (coordinates) from w called F. For each feature $w_i \in F$ we add/subtract a learning rate η and then determine the w_0 that optimizes the loss (done in parallel on a GPU). We consider all possible values of $w_0 = \frac{w^T x_i + w^T x_{i+1}}{2}$ for $i = 0...n-2$ and select the one that minimizes the loss (also performed in parallel on a GPU).

4. After making the update above we evaluate the loss on the full dataset (performed on a GPU for parallel speedups) and accept the change if it improves the loss.

end while

xyzacademic/mlp01example. Since sign activation is non-convex and our training starts from a different random initialization we run it 8 times and output the majority vote.

To illustrate our real runtimes and clean test accuracies we compare our model with a single hidden layer of 20 nodes to the equivalent network with sigmoid activation and logistic loss (denoted as MLP) and the binary neural network (denoted as BNN) [15]. We used the MLPClassifier in scikit-learn [16] to implement MLP and the Larq library [17] with the *approx* approximation to the sign activation. This has shown to achieve a higher test accuracy than the original straight through estimator (STE) of the sign activation [18].

We perform a 1000 iterations of SCD01. Our training runtimes are comparable to gradient descent in MLP and BNN and thus practically usable. We can trivially parallelize training an ensemble by doing multiple runs on CPU and GPU cores at the same time.

III. Results

We include in our experiments a WordCNN with sigmoid activation and cross-entropy loss (described below) denoted as CNN and a random forest classifier [19] denoted as RF. In Table I we see that ensembles of our models give the highest adversarial accuracy on all four datasets and require the greatest number of queries. If a smaller limit was placed on the allowed queries (for example imposed by the system being attacked) we can expect a higher adversarial accuracy for our models. Here we show ensembles of 8 votes for each model - in the random forest we also use 8 trees.

TABLE I

Accuracy of clean and TextFooler black-box adversarial examples denoted by Cl and Adv respectively. All models shown here are 8 votes. Also shown are the number of queries made by the attacker denoted as Que. We round accuracies and queries to the nearest integer. For each dataset highest adversarial accuracy shown in bold.

	IMDB			Yelp		
	Cl	**Adv**	**Que**	**Cl**	**Adv**	**Que**
SCD01	82	**51**	3279	85	**54**	1908
CNN	89.2	0	524	94	1.1	492
MLP	85	0	686	87.3	. 2	500
BNN	83.8	21.5	2301	85	32.3	1622
RF	76.7	11	1823	77.7	7.5	935
	MR			**AG**		
	Cl	**Adv**	**Que**	**Cl**	**Adv**	**Que**
SCD01	74	**14**	186	99	**93**	672
CNN	78	2.8	123	96.5	49.1	258
MLP	75	2.3	123	99	51.4	366
BNN	73.2	5.8	150	99.1	75.6	564
RF	68.1	2.1	115	96.7	72.2	532

WordCNN stacks word vectors [20] of each word in a document into a matrix to treat it as 2D image [21] (see Figure 1). In the other models that take feature vectors as inputs we consider the averaged word vector of all words in a document [22]. For all models we use 200 dimensional Glove word embeddings pre-trained on 6 billion tokens from Wikipedia and Gigawords [20]. This gives a lower clean test accuracy than WordCNN but still above an acceptable level in practice.

With sign activation and 01 loss the CNN becomes CNN01. We study a new variation of it where instead of global average pooling over +1 and -1 we only sum the 1's. We call this CNN01-FS. For example if we have two sentences of length 10 and 100 each with 4 keywords, by averaging the 4 words we have weights 40% and 4% respectively (lower in the larger sentence), but by summing it's the same in both. We find this to significantly improve the adversarial accuracy of CNN01. In Tableřeftable2 we see that CNN01-FS improves several orders of magnitude upon the original one. Even the CNN with sigmoid activation has a slight improvement with the variation but not nearly as large as CNN01.

IV. Discussion

One reason why our models are hard to deceive is that TextFooler relies upon output probabilities to craft its attack. Our models focus on the sign of the outputs in their loss. We output probabilities TextFooler needs by counting the number of 0 and 1 outputs in the ensemble. However, these probabilities are not useful enough for TextFooler to volley an effective attack.

V. Conclusion

Our work here shows that 01 loss sign activation network ensembles are hard to deceive with TextFooler text adversarial

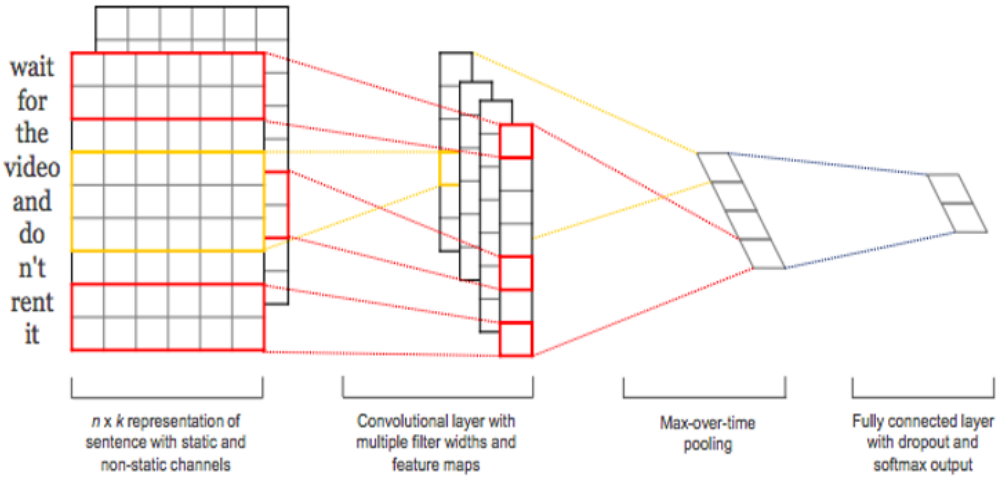

Fig. 1. CNN text classiication model - the model above shows two input channels for a given sentence

TABLE II
ACCURACY OF CLEAN AND TEXTFOOLER BLACK-BOX ADVERSARIAL EXAMPLES DENOTED BY CL AND ADV RESPECTIVELY. ALL MODELS SHOWN HERE ARE 8 VOTES. ALSO SHOWN ARE THE NUMBER OF QUERIES MADE BY THE ATTACKER DENOTED AS QUE. WE ROUND ACCURACIES AND QUERIES TO THE NEAREST INTEGER. FOR EACH DATASET HIGHEST ADVERSARIAL ACCURACY SHOWN IN BOLD.

	IMDB			**Yelp**		
	Cl	**Adv**	**Que**	**Cl**	**Adv**	**Que**
CNN01-FS	80.7	**67.9**	1288	89.6	**64.6**	566
CNN01	84.9	0.2	299	89.4	19.5	296
CNN-FS	88.4	8.5	419	92.9	3	220
CNN	86.9	0.3	301	92.2	10.7	252
	MR			**AG**		
	Cl	**Adv**	**Que**	**Cl**	**Adv**	**Que**
CNN01-FS	76.6	**40.9**	265	84.3	**66.7**	484
CNN01	77.6	13.1	157	85.7	5.7	207
CNN-FS	79.1	14.9	170	85.8	4.3	214
CNN	79.4	9.2	142	86.9	3.3	208

attacks. This is consistent with earlier work on image adversarial attacks on the same model. Going forward this may serve as a helpful baseline for robust secure AI models.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[2] L. A. Zhang, G. S. Hartnett, J. Aguirre, A. J. Lohn, I. Khan, M. Herron, and C. O'Connell, *Operational Feasibility of Adversarial Attacks Against Artificial Intelligence*. Santa Monica, CA: RAND Corporation, 2022.

[3] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018, pp. 274–283.

[4] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[5] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, "Adversarial training can hurt generalization," in *Identifying and Understanding Deep Learning Phenomena ICML Workshop*, 2019.

[6] J. Clarysse, J. Hörmann, and F. Yang, "Why adversarial training can hurt robust accuracy," *arXiv preprint arXiv:2203.02006*, 2022.

[7] M. Xie, Y. Xue, and U. Roshan, "Stochastic coordinate descent for 0/1 loss and its sensitivity to adversarial attacks," in *Proceedings of 18th IEEE International Conference on Machine Learning and Applications - ICMLA 2019*, 2019, pp. 299–304.

[8] Y. Xue, M. Xie, and U. Roshan, "On the transferability of adversarial examples between convex and 01 loss models," in *IEEE International Conference on Machine Learning and Applications*, 2020.

[9] ——, "Towards adversarial robustness with 01 loss neural networks," in *IEEE International Conference on Machine Learning and Applications*, 2020.

[10] Z. Yang, Y. Yang, Y. Xue, F. Y. Shih, J. Ady, and U. Roshan, "Accurate and adversarially robust classification of medical images and ECG time-series with gradient-free trained sign activation neural networks," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 2456–2460.

[11] Y. Yang, F. Y. Shih, and U. Roshan, "Defense against adversarial attacks based on stochastic descent sign activation networks on medical images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 03, p. 2254005, 2022.

[12] Y. Xue and U. Roshan, "Accuracy of white box and black box adversarial attacks on a sign activation 01 loss neural network ensemble," in *The First Tiny Papers Track at ICLR 2023, Tiny Papers @ ICLR 2023, Kigali, Rwanda, May 5, 2023*, K. Maughan, R. Liu, and T. F. Burns, Eds. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=QimsmhYvsf

[13] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? a strong baseline for natural language attack on text classification and entailment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8018–8025.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[15] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in neural information processing systems*, 2016, pp. 4107–4115.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] L. Geiger and P. Team, "Larq: An open-source library for training binarized neural networks," *Journal of Open Source Software*, vol. 5, no. 45, p. 1746, 2020.

[18] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 722–737.

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[21] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[22] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 2015, pp. 136–140.