TOWARDS ROBUST VISUAL TRANSFORMER NETWORKS VIA K-SPARSE ATTENTION

Sajjad Amini, Shahrokh Ghaemmaghami

Electronics Research Institute, Sharif University of Technology

ABSTRACT

Transformer networks, originally developed in the community of machine translation to eliminate sequential nature of recurrent neural networks, have shown impressive results in other natural language processing and machine vision tasks. Self-attention is the core module behind visual transformers which globally mixes the image information. This module drastically reduces the intrinsic inductive bias imposed by CNNs, such as locality, while encountering insufficient robustness against some adversarial attacks. In this paper we introduce K-sparse attention to preserve low inductive bias, while robustifying transformers against adversarial attacks. We show that standard transformers attend values with dense set of weights, while the sparse attention, automatically selected by an optimization algorithm, can preserve generalization performance of the transformer and, at the same time, improve its robustness.

Index Terms— Visual transformer, self-attention, sparse, adversarial robustness

1. INTRODUCTION

Convolutional Neural networks (CNNs), due to relative matching with human perception system, have appealed a vast amount of research to improve image perception. Recently, a promising schemes have been presented using CNNs with several layers of convolution operator, such as VGG, Inception and ResNet [1].

Convolution layers, the core unit in CNNs, imposes heavy locality inductive bias over the architecture. This bias, though useful prior for the case of limited training data, can limit the performance when a huge amount of training data is accessible [2]. Thus, due to growing resources of training data, it is needed to build architectures without a strong inductive bias.

In 2017, Vaswani *et al.* showed that a purely attention-based architecture can replace recurrent neural networks (RNNs) in an encoder-decoder scenario for machine translation [3]. Attention, the core idea in transformers, determines the meaning for each word in the sentence based on the other available words. Accordingly, an attention unit defines one value for each word based on all other words in the sentence, which leads to a global interpretation, beyond the local meaning shaped with the neighboring words. A parallelized set of attention units leads to Multi-head attention layer where several values for each word are calculated. In contrast to RNNs, transformers can easily handle very long range dependencies, while supporting parallelized implementations [3].

Since 2017, Transformers have been used in various natural language and speech processing applications, such as language modeling, sentiment analysis, text classification, machine translation and text summarization [4].

One straightforward path to apply transformers in image perception tasks is to replace words in sentences with pixels in images, which leads to quadratic computation cost with respect to the number of pixels and is prohibitive in real world applications [2]. In [5] the authors argue that image generation task can be formulated as an auto-regressive sequence, so a transformer can be utilized in this context. To handle the computational cost in applications for the real world tasks, the attention is limited to neighborhood aggregation [5].

The concept of sparse transformers, where sparse factorizations of the attention matrix is introduced, is presented in [6]. These factorizations reduce the quadratic time and memory requirement with respect to sequence length in the transformers to $O(L\sqrt{L})$ (L represents the sequence length), making it applicable to real world tasks. Reformer is another trend toward applicable transformers [7]. It reduces the prohibitive complexity of transformers by replacing the dot-product attention by a new variant utilizing locality-sensitive hashing. This replacement reduces the aforementioned complexity from quadratic to $O(L \log L)$. Using reversible residual layers also results in lower memory consumption.

While the starting efforts to utilize transformers for image processing have been based on pure application of self-attention mechanism, combining this mechanism with convolution operator is also considered in the following publications. Attention augmented convolutional networks are designed to help the locality of CNNs and globality of transformers by concatenating their feature maps leading to improved results, as compared to those of pure CNNs [8]. In contrast to augmenting, applying attention mechanism on top of CNNs feature maps is explored in [9], where raw pixels are replaced by semantic tokens.

The question of whether pure CNNs, fully-attentional model, or combined architectures could lead to better performance is answered in [10]. The authors show that with enough number of attention heads, self-attention mechanism

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

can express any convolutional layer and thus fully-attentional model can present both local and global behaviors. So, with typically fewer number of parameters, fully-attentional models can be generalized better, in contrast to combined models [2]. In [10], also, a fully-attentional model is demonstrated, where the tokens are 2×2 image patches.

In [2] another fully-attentional model is presented, where the input tokens are larger image patches, in comparison to those in [10]. In a visual Transformer (ViT), input image is split into flattened patches with no overlaps. Then, each patch is linearly embedded, and positional embedding is added. The resulting sequence of vectors is fed into a standard transformer encoder, consisting of L_x core modules. The core modules include normalization, multi-head attention, multi-layer perceptron, and residual connections. Innovative strategies to train the transformers based on self-supervised learning is also provided in [2] and the authors claim outperforming CNNs, when the pre-trained ViTs are fine-tuned in different vision tasks.

After successful application of the transformers in machine vision tasks, their robustness is being investigated recently. The authors in [11] have shown that visual transformers are at least as robust as ResNet in different ranges of perturbation, when pre-trained with a sufficient amount of data. They have also showed that the input perturbation patterns vary significantly across ResNet and transformer, due to their serious model differences.

In [12] the authors signify low transferability of adversarial attacks between CNNs and transformers, which again emphasizes on serious differences between the models employed. Various interesting results, considering the robustness of transformers, in comparison to that of CNNs, are reported in [13]. The authors show that the transformers output features contain less low-level information and thus provide superior robustness. Another interesting finding is that robustness of a hybrid model is improved by increasing the proportion of transformers.

In this paper, we are focused on the attention mechanism and its effect on the robustness of ViT proposed in [2]. Our contribution can be summarized as follows:

- We show that in scaled dot product attention, softmax output vector is dense. Our simulation results represent that to calculate the output for each input vector, different values are linearly combined using a dense set of weights and the module attend almost all input values.
- It is shown that increasing the sparsity of the weight vector (called $\mathbf{w}$ in this paper) will improve the robustness of ViT in image classification tasks against both targeted and untargeted adversarial attacks.
- We introduce a novel learning procedure to train K-sparse attention ViTs (KSA-ViT), where weight vectors are sparse and show that it results in higher robustness while preserving the generalization performance.

2. METHOD

The ViT, introduced in [2], relies on the standard transformer model, where attention is fashioned as:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent query, key, and value matrices, respectively, and d_k is used for normalization. We introduce weight matrix $\mathbf{W} = softmax(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k})$. The softmax operates over each row which leads to normalized row weight vectors $\mathbf{w}$. Each weight vector is then used to weigh matrix columns to generate the attention output. Our simulations have shown these row vectors, $\mathbf{w}$, to be dense in a trained ViT.

There are unnecessary features (features that do not contribute to the current task) in each architecture that may give room to attacks. Removing these features, through sparsification of the network (via stochastic pruning [14], for instance), reduces the data volume and the chance for an attack, improves accuracy, and hence robustifies the architecture. We propose to select unnecessary attention weight matrix elements in a ViT via ℓ_0 sparsity promoting regularization. This approach can lead to a simple and fast implementation that is desirable in deep learning architectures in real world applications. Our new formulation, called K-Sparse Attention, can train ViT (KSA-ViT) for classification, where the weight vectors are constrained to be K-sparse (having exactly K nonzero elements) in the selected attention layers. The KSA-ViT formulation is shown as:

$$\mathcal{P} : \min_{\mathbf{p}} \sum_i D(\mathbf{y}_j, \widehat{\mathbf{y}}_j) \; w.r.t \; \|\mathbf{w}^j_{i,l}\|_0 \leq K^j_{i,l}, \begin{cases} 0 \leq i \leq I_l \\ l \in \mathcal{S} \end{cases} \tag{2}$$

where $\mathbf{p}$ is the vector containing all the model parameters, $\mathbf{y}_j$ and $\widehat{\mathbf{y}}_j$ are target and network output vectors for j-th training sample, respectively, $D(\cdot, \cdot)$ is a distance metric, i is the sequence position, l is the layer number, I_l is the sequence length in layer l, $\mathcal{S}$ is the set of layers to be constrained, and $K^j_{i,l}$ is the maximum allowable number of nonzero elements for i-th weight matrix of the j-th training sample in the l-th layer. The constrained formulation (2) can be converted into regularized one, as:

$$\min_{\mathbf{p}} D(\mathbf{y}_j, \widehat{\mathbf{y}}_j) + \sum_{l \in \mathcal{S}} \sum_{0 \leq i \leq I_l} \mathcal{I}\left\{\|\mathbf{w}^j_{i,l}\|_0 \leq K^j_{i,l}\right\} \tag{3}$$

where the indicator function is defined as:

$$\mathcal{I}\{\|\mathbf{x}\|_0 \leq \delta\} = \begin{cases} 0 & if \;\; \|\mathbf{x}\|_0 \leq \delta \\ \infty & if \;\; \|\mathbf{x}\|_0 > \delta \end{cases}$$

Similar to [15], we relax (3) by introducing a penalty as:

$$\mathcal{P}_\mu : \min_{\mathbf{p},\{\mathbf{s}^j_{i,l}\}} D(\mathbf{y}_i,\widehat{\mathbf{y}}_i) + \sum_{l\in\mathcal{S}}\sum_{0\le i\le I_l}\left(\mathcal{I}\left\{\|\mathbf{s}^j_{i,l}\|_0 \le K^j_{i,l}\right\} + \frac{1}{2\mu^j_{i,l}}\|\mathbf{s}^j_{i,l}-\mathbf{w}^j_{i,l}\|_2^2\right). \qquad (4)$$

where $\{\mathbf{s}^j_{i,l}\}$ is the set of penalty parameter vectors and $\mu^j_{i,l}$ is penalty constant. The solution to $\mathcal{P}_\mu$ defined in (4) coincide with main KSA-ViT in $\mathcal{P}$, when $\mu \to 0$. In [15] a warm start is used to approximate the solution to $\mathcal{P}$, where a sequence of $\mathcal{P}_\mu$ problems for decreasing values of μ are solved and each problem is initialized using the previous problem results. We follow the same procedure in this paper. To solve problem $\mathcal{P}_\mu$, an alternative minimization is used which split the problem into two sub-problems.

The first sub-problem is optimization over the penalty parameter vectors $\{\mathbf{s}^j_{i,l}\}$. Due to the independency of problems for different values of i, l, and j, we have the following problem for each set of i, l, and j values at iteration k.

$$\mathbf{s}(k+1) = \arg\min_{\mathbf{s}} \ \mathcal{I}\left\{\|\mathbf{s}\|_0 \le K\right\} + \frac{1}{2\mu}\|\mathbf{s}-\mathbf{w}(k)\|_2^2 \qquad (5)$$

Problem (5) has a closed form solution based on the definition of proximal operator [16], as:

$$\mathbf{s}(k+1) = Prox_{\mathcal{I}}(\mathbf{w}(k)) = [\mathbf{w}(k)]_K \qquad (6)$$

where $[\mathbf{w}(k)]_K$ is K-sparse counterpart of $\mathbf{w}(k)$ and is calculated by preserving the K elements in $\mathbf{w}(k)$ with largest absolute values and setting the other elements to zero.

The second sub-problem is optimization over the vector of network parameters $\mathbf{p}$, as:

$$\mathbf{p}(k+1) = \arg\min_{\mathbf{p}} \ D(\mathbf{y}_i,\widehat{\mathbf{y}}_i) + \sum_{l\in\mathcal{S}}\sum_{0\le i\le I_l}\frac{1}{2\mu^j_{i,l}}\|\mathbf{s}^j_{i,l}(k+1)-\mathbf{w}^j_{i,l}\|_2^2. \qquad (7)$$

The objective function in problem (7) is smooth, hence any smooth optimization algorithm can be used to update $\mathbf{p}$.

Algorithm 1 summarizes the procedure to train the KSA-ViT. Two important points must be considered when using this training algorithm. First, based on $\mathbf{W}$ definition, all of its elements are positive in the open interval $(0, 1)$ while, in (7), the optimization step tries to move several elements of $\mathbf{w}^j_{i,l}$ vector toward zero (due to the fact that several elements of $\mathbf{s}^j_{i,l}$ are zero). This may lead to contradicting gradient directions. To avoid this issue, we replace softmax operator with a linear function that maps all the elements of $\mathbf{W}$ to closed interval $[0, 1]$, while each row sums to one. Our simulation results show no performance degradation using this replacement. Second, based on (7), the optimization step tries to approach $\mathbf{w}^j_{i,l}$ to $\mathbf{s}^j_{i,l}(k+1)$. The model forces the summation of elements of vector $\mathbf{w}^j_{i,l}$ to be unity, while this value is not necessarily unity for $\mathbf{s}^j_{i,l}(k+1)$, which again leads to contradicting directions of gradients. To cope with this problem, we divide $\mathbf{s}(k+1)$ by the sum of its elements to make unity the summation of its elements. In the next section, we explore the robustness of the proposed KSA-ViT, in comparison to that of standard ViT training, against adversarial attacks.

Algorithm 1 KSA-ViT training

```
procedure
    Input: Training patterns ($\{\mathbf{X}_i, \mathbf{y}_i\}$), $N_1$, $N_2$, $c$, $\mu$.
    Initialization: $\mathbf{p}_0$, $k = 0$ , $m = 0$
    Output: Network parameters vector $\mathbf{p}_{final}$
    while $m \le N_1$ do
        while $k \le N_2$ do
            $\mathbf{s}^j_{i,l}(k+1) = [w^j_{i,l}(k)]_{K^j_{i,l}}$ , for $i$, $l$ and $j$
            Update $\mathbf{p}$ using (7)
            $k \leftarrow k + 1$
        $\mu \leftarrow c \cdot \mu$, $m \leftarrow m + 1$
        $\mathbf{p}_0 \leftarrow \mathbf{p}$
        $k = 0$
    $\mathbf{p}_{final} \leftarrow \mathbf{p}$
```

3. SIMULATION RESULTS

In this section, we present several simulation results which support the robustification properties of the proposed KSA-ViT over three benchmark datasets, namely, MNIST [19], Fashion-MNIST [20] and CIFAR10 [21]. We fixed the architecture during our simulation. It is similar to one considered in [2], where the patch size is 7 (for MNIST) and 8 (For Fashion-MNIST and CIFAR10), and patch embedding dimension, depth, number of heads, and multi-layer perceptron dimension are 64, 6, 8 and 128, receptively. For simplicity, $\mu^j_{i,l}$ and $K^j_{i,l}$ are kept fixed for different values of i, j, l. All the hyper-parameters are selected using the grid search and we evaluate the KSA-ViT, while every single layer is constrained or all the layers are constrained to be sparse simultaneously. All the attacks have been implemented using Adversarial Robustness Toolbox (ART) [22].

Figure 1 compares the KSA-ViT and the standard ViT against untargeted adversarial attacks, Fast Gradient Sign Method (FGSM) [17] and Projected Gradient Descent (PGD) [18]. As shown, constraining the attention weights to be sparse leads to robustification against untargeted adversarial attacks in all benchmark datasets used. With more complex datasets, higher improvements are achieved (can be seen by comparing CIFAR10 and MNIST) and constraining a single layer or all layers makes no perceivable change. As expected, PGD degrades the performance further, as it uses an iterative strategy to design an adversarial perturbation.

Table 1 compares the robustness of the KSA-ViT and ViT against targeted attacks, namely CW with ℓ_2 and ℓ_∞ norm

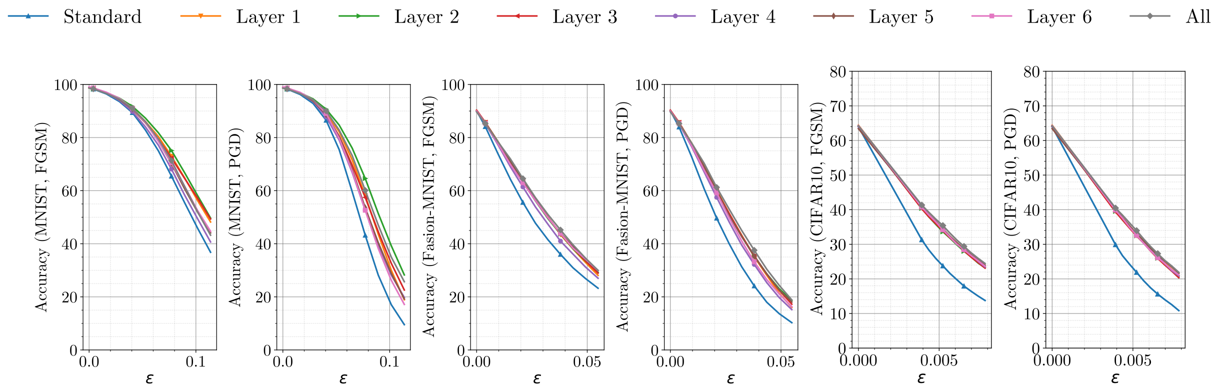
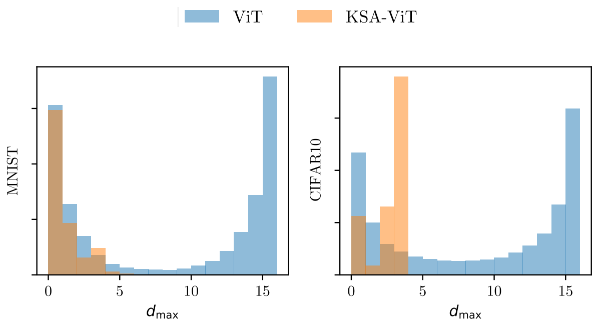


Fig. 1: KSA-ViT vs ViT in tolerating untargeted adversarial attacks (FGSM [17] and PGD [18]) over test set of different benchmark datasets.

Table 1: Performance comparison against CW targeted adversarial attacks over CIFAR10 dataset

Type	CW-L2				CW-Linf			
	ASR	L_1	L_2	L_∞	ASR	L_1	L_2	L_∞
Satndard	0.62	27.29	0.74	0.08	,0.81	,37.70	,0.81	,0.033
Layer 1	0.60	32.71	0.88	0.09	,0.78	,46.08	,0.97	,0.036
Layer 2	0.57	32.38	0.87	0.09	,0.76	,45.82	,0.96	,0.036
Layer 3	0.61	30.91	0.83	0.09	,0.76	,46.13	,0.97	,0.036
Layer 4	0.58	33.46	0.90	**0.09**	,0.79	,46.14	,0.97	,0.036
Layer 5	0.56	33.46	0.90	0.09	,0.77	,48.06	,1.00	,0.036
Layer 6	0.58	32.95	0.88	0.09	,0.77	,46.98	,0.98	,0.036
All	**0.56**	**34.83**	**0.93**	0.09	,**0.75**	,**49.51**	,**1.03**	,**0.036**

Fig. 2: Weighted histogram for the position of maximum difference between sorted attention of first layer in ViT architecture



metrics [23] over CIFAR10 dataset. The results show that the KSA-ViT not only reduces the attack success rate (ASR), but it also needs more perturbation energy to be successfully attacked, as compared to standard ViT. In contrast to untargeted attacks, the best performance is attained in almost all the metrics, when all the layers are constrained for targeted attacks. In untargeted attacks, the perturbation that changes the label is desired, while the perturbation is generated for each adversarial target in targeted attacks. Thus, constraining all the layers, as presented by the results of targeted attacks, can be a better option.

The simulation results confirm that the KSA-ViT can improve the robustness of standard ViT against both targeted and untargeted adversarial attack. Figure 1 shows how the new formulation affect the attention vectors in the first layer of a transformer, when all the layers are constrained to be sparse. In this experiment, attention vectors are constrained to be 4-sparse. To generate Figure 1, we sort attention vectors in the first layer, calculate their difference vector and find the position of largest value in the resulting vector. Then we calculate the weighted histogram of these positions, while the weights are unity minus the value of the corresponding attention in that position. We expect the peak in the histogram to be below 4 (the target sparsity of the attention vectors) and, as we can see, this is the case for both MNIST and CIFAR10 datasets.

4. CONCLUSIONS

Visual transformers have recently attracted a vast interest in the area. We have shown that visual transformers attend a dense set of (Key, Value) pair to generate attention modules. Conceptually, a more robust output for each vector in the input sequence could be found when attending a small subset of input sequence vectors. Previous research on conventional deep learning architectures has also proved that representation of sparsity leads to architecture robustness. Based on these two findings, we have proposed the new learning procedure of KSA-ViT, which formulates the attention for each input sequence to be limited to subsets of K (Key,Value) pairs, where K is a hyper-parameter. We have proposed an efficient optimization algorithm to solve the resulting challenging problem and have shown that our new learning algorithm can successfully sparsify the attention, while no degradation in the performance is observed for the case of clean test images. Our simulations on both targeted and untargeted adversarial attacks have also presented a significant improvement to the robustness of visual transformers.

5. REFERENCES

[1] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[4] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al., "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[5] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran, "Image transformer," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4055–4064.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019.

[7] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.

[8] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3286–3295.

[9] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda, "Visual transformers: Token-based image representation and processing for computer vision," *arXiv preprint arXiv:2006.03677*, 2020.

[10] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi, "On the relationship between self-attention and convolutional layers," *arXiv preprint arXiv:1911.03584*, 2019.

[11] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit, "Understanding robustness of transformers for image classification," *arXiv preprint arXiv:2103.14586*, 2021.

[12] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk, "On the robustness of vision transformers to adversarial examples," *arXiv preprint arXiv:2104.02610*, 2021.

[13] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh, "On the adversarial robustness of visual transformers," *arXiv preprint arXiv:2103.15670*, 2021.

[14] Ji Gao, Beilun Wang, Zeming Lin, Weilin Xu, and Yanjun Qi, "Deepcloak: Masking deep neural network models for robustness against adversarial samples," *arXiv preprint arXiv:1702.06763*, 2017.

[15] Sajjad Amini and Shahrokh Ghaemmaghami, "A new framework to train autoencoders through non-smooth regularization," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1860–1874, 2019.

[16] Neal Parikh and Stephen Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2015.

[18] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2014.

[19] Yann LeCun and Corinna Cortes, "MNIST handwritten digit database," 2010.

[20] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[21] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Citeseer, 2009.

[22] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al., "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.

[23] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.