EXPLOITING CNN LAYER ACTIVATIONS TO IMPROVE ADVERSARIAL IMAGE CLASSIFICATION

R. Caldelli,[*] *R. Becarelli*

MICC, University of Florence
Florence, Italy

F. Carrara[†], *F. Falchi, G. Amato*[‡]

ISTI CNR
Pisa, Italy

ABSTRACT

Neural networks are now used in many sectors of our daily life thanks to efficient solutions such instruments provide for diverse tasks. Leaving to artificial intelligence the chance to make choices on behalf of humans inevitably exposes these tools to be fraudulently attacked. In fact, adversarial examples, intentionally crafted to fool a neural network, can dangerously induce a misclassification though appearing innocuous for a human observer. On such a basis, this paper focuses on the problem of image classification and proposes an analysis to better insight what happens inside a convolutional neural network (CNN) when it evaluates an adversarial example. In particular, the activations of the internal network layers have been analyzed and exploited to design possible countermeasures to reduce CNN vulnerability. Experimental results confirm that layer activations can be adopted to detect adversarial inputs.

***Index Terms*—** Adversarial images, neural networks, layer activations, adversarial detection.

1. INTRODUCTION

Deep neural networks are more and more pervading many sectors of our daily life due to the fact that such instruments provide highly efficient solutions for different tasks like automotive, computer vision, information management, and so on. Notwithstanding that, leaving to artificial intelligence (AI) the chance to solve problems and/or to make choices on behalf of humans inevitably exposes such tools to the risk to be maliciously attacked in order to mislead their final decisions. In fact, it has been shown in literature [1] that adversarial examples, intentionally crafted to fool a neural network, can drastically induce a misclassification though appearing perceptually similar to the original version for a human eye.

According to this, many techniques have recently been designed to increase the robustness of the attacked models to these adversarial inputs [2]. One of the main approaches is based on the so-called *adversarial training* which consist of generating on the fly and including in the training phase a group of adversarial examples starting from the training set itself [3]. So doing, the model is also trained to learn these misleading samples, and consequently, the perturbation needed to fool the neural network is stronger and easily detectable. However, this kind of strategy is not exhaustive because it cannot take into account of all the possible types of attacks, and in any case, a new training phase would be necessary to add new-born attack procedures. Other solutions have been envisaged. In [4], the authors proposed to improve the trained model by resorting at a smoothing operation (named *distillation*) along the gradient directions around training points an attacker would exploit. Different kinds of defense are based on image processing [5]: they try to remove the adversarial perturbation imposed on the image by means of color depth reduction or median filtering. Anyway, they seem to be effective against specific attacks and are not always applicable.

Another diverse approach consists in detecting adversarial inputs and consequently providing a reliability score or validating each decision taken by the neural network. Many strategies have been proposed based on detector subnetworks [6, 7], statistical tests [8, 9], or perturbation removal [10], but results achieved so far are not so satisfactory in terms of robustness [11]. Supported by the relevance of intermediate representations proved by many works [12, 13, 14], the use of internal representations learned by the network to solve the problem of adversarial detection has been explored in various papers [7, 15, 16, 17, 18].

On such a basis, the present paper focuses on the problem of image classification in an open-set scenario and proposes an analysis to better understand what really happens inside a convolutional neural network (CNN) when is asked to decide on adversarial examples. In particular, the activations of the internal layers composing the network have been analyzed by comparing their behavior and, above all, their evo-

[*] R. Caldelli is also with National Inter-University Consortium for Telecommunications - CNIT, Parma, Italy.

[†] This work was partially supported by ADA, Automatic Data and documents Analysis to enhance human-based processes co-founded by the Tuscany region under the POR FESR 2014-2020 program, CUP CIPE D55F17000290009.

[‡] All of the authors would like to gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp and Telsa K40 GPUs used for this research.

lution throughout the layers in presence of adversarial inputs with respect to genuine ones. After that, differences emerged between the two cases have been encoded and exploited to design possible countermeasures to reliably detect adversarial examples, thus reducing CNN vulnerability. Experimental tests have been carried out on diverse kinds of adversarial crafting algorithms with the assumption that the technique used to create a fake sample is not known to the classifier as it usually happens in practice. Achieved results confirm that layer activations can explain the behavior of the CNN in presence of an adversarial example and that such knowledge can be used for detection of these fake inputs.

The rest of the paper is organized as it follows: Section 2 presents the rationale and introduces the activations space, while Section 3 is dedicated to the experimental verification; in particular, Section 3.1 describes the results which experimentally confirm the theoretical hypotheses, and Section 3.2 proposes some possible countermeasures to adversarial examples. Finally Section 4 draws conclusions.

2. EXPLOITING THE ACTIVATIONS SPACE

2.1. The basic idea

Let us try to understand what it happens in the internal layers of a CNN when an adversarial image I_A is passed as input (see Figure 1). What determines that such an image is in the end wrongly classified as belonging to the class C_A? Being perceptively indistinguishable with respect to the same original image I_O, why is it not identified within the class C_O as expected? The first consideration to be made is that these two "similar" samples should presumably follow two diverging paths flowing through the network and, consequently, generate different layer activations yielding to diverse output decisions. The idea is to find if and where the paths diverge in the neural network. Specifically, there should be an evolution, throughout the layers, that induces a wrong final choice; moreover, we would try to exploit this knowledge in order to design a detection procedure to make an assessment on the reliability of the classification done by the CNN.

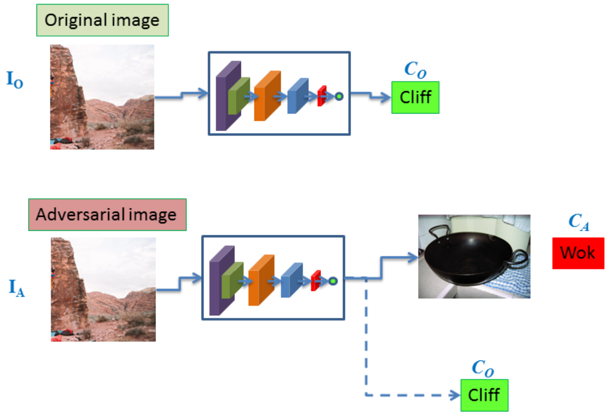

Fig. 1. CNN decisions: original and adversarial cases.

2.2. The activations space

According to the previous considerations, the layer-by-layer activations have been taken into account to try to highlight this diverse behavior of the neural network. Given a certain image I and being $\Delta : I \rightarrow \{1, \ldots, C\}$ a CNN image classifier made up of L layers where C is the number of the output classes, we indicate with $a_l \in R^{N_l}$ the N_l-dimensional activation vector of the l-th layer with $l = 1, \ldots, L$. Such activation vectors a_l can be extracted from every layer of the neural network for each input test image (adversarial or pristine) but they are useless if they are not compared with a sort of reference that permits to evidence the possible presence of an anomaly.

To do this, we assume to have at disposal the training set S_{Train}, used to train the network, and, in particular, all the activations of the images to construct class reference points for every layer l (see sub-section 2.3 for details). This is not a strong assumption because it can be performed during the training phase and, above all, it can be done once for all and stored. It is plausible to imagine that images belonging to the class $c \in C$ should manifest a certain level of homogeneity in their evolutions through the layers of the network and could well model the class itself by means of a representative.

So, basically, the idea is to get a comparison, in the feature distance space layer-by-layer, between the representation of the to-be-tested image which is classified by the CNN in a certain class $C_{out} \in C$ (wrongly or rightly) and the representative, according to the images of the training set, for that class C_{out}. Dissimilarities should reveal that the test image is an adversarial example.

2.3. Class representatives and dissimilarity measures

On the basis of the previous sub-section, we have defined the class representative $\mathbf{R_c}$ $(c = 1, \ldots, C)$ as indicated in Equation (1) where the *medoid* is computed:

$$R_c^l = \underset{y \in a_l^k}{\operatorname{argmin}} \sum_{k=1}^{K_c} d(y, a_l^k) \tag{1}$$

where $d(\cdot)$ indicates the L_2 metric, a_l^k is the activation of the k-th image at layer l and K_c states for the cardinality of the class c. Obviously, other representatives could be chosen but it is out of the topic of this work to investigate diverse solutions. According to this, $\mathbf{R_c} = [R_c^1, R_c^2, \ldots, R_c^L]$ will be the representative of class c being N_l the dimension of R_c^l at each layer (generally such a dimension is different layer by layer).

All the $\mathbf{R_c}$ can be seen as a sort of layer-level reference map in the activations space that can be used to make a comparison with the corresponding position assumed by a test image at that specific layer. Therefore, given an image I_{test} belonging to the test set S_{Test}, its representation at each layer l can be matched, in terms of distance (e.g. L_2 norm), with all the C representatives, leading to the construction of a new feature $F_{I_{test}}$ in the distance space whose dimension is $L \times C$. Such a feature $F_{I_{test}}$ will be used as dissimilarity evidence to determine if the image under analysis has been classified reliably by the CNN and, consequently, if it can be labeled as an adversarial example or not. It is expected that original images, correctly classified by the network, should follow a path much more similar to that of the representative of their output class than adversarial ones which malevolently fall in there.

3. EXPERIMENTAL VERIFICATION

This section presents some of the experimental tests that have been carried out in order both to validate the theoretical assumptions (see sub-section 3.1) and to demonstrate that the internal behavior of a neural network, in terms of layer activations, can be exploited to detect adversarial examples (see sub-section 3.2).

For the verification of our hypotheses, we have taken into account the configuration proposed in the context of the *NIPS 2017 Adversarial Defenses Kaggle Competition* [19] where the network, named *InceptionV3*, was used as baseline. Such a network was trained on a training set of over one million images that have been classified against the ILSVRC2014 [20] wordnet subset comprising 1000 synsets (more than 1000 images for each synset).

For what concerns adversarial images, we have used the *DEV image set* again provided within the *Kaggle* competition; the test set consists in fact of 5000 images (they are not part of ImageNet dataset) subdivided into 5 groups, each of 1000 images, and is mapped on the same ILSVRC2014 wordnet subset. One group is composed of the original images and the other four contain the attacked versions obtained by applying FGSM technique [1] with $\epsilon = 16$ choosing a random target class, by just adding a random gaussian noise ($[-16, +16]$) and the last two groups by using again FGSM, firstly with a target class and secondly by iterating the attack with $\epsilon = 1$ for 20 iterations [21]. For each image, we collect the 12 activations tensors (so $L = 12$ in this case) corresponding to the output of the logical blocks (Inception block) comprising the InceptionV3 network. Each activation consist of multiple feature maps from which we extract a compact feature vector by applying a global spatial average pooling operation.

3.1. Theoretical assumption verification

In this sub-section, some of the experimental results carried out to validate the theoretical assumptions made before are presented. To this purpose, in Figure 2, a visualization of the activations at different layers are depicted. In particular, being the dimension N_l different at each layer and in order to plot a bidimensional representation perceptively and intuitively significant, we have resorted at the *t-SNE* algorithm [22].

Figure 2 provides a straight-forward way to comprehend what happens in the activations space when an adversarial and genuine image are observed. In this case, we have taken for exemplification the test images I_{1342} and I_{1617} (original) which belong to the class 138 (*water hen*) and 973 (*cliff*) respectively, and the image I_{4617} (adversarial) which originally belongs to the class 973 (*cliff*) but being an adversarial example, it is instead classified by the CNN within the class 910 (*wok*). By looking at Figure 2 (top-left), it can be seen that the points representing the images of the training set at layer 1 (only 200 images per class are plotted for sake of clarity) are not yet well clustered and that the original and adversarial samples are visible within the cloud of points.

Going ahead through the layers, it can be observed (see Figure2 top-right and bottom-left for layer 7 and 8 respectively) that the image activations tend to group according to their membership class: class 138 (*water hen*, red cloud), class 973 (*cliff*, blue cloud) and class 910 (*wok*, green cloud). It is very interesting to notice that all the images still belong to their correct cluster: the adversarial example I_{4617} is still close to I_{1617} and to its blue cloud (*cliff*) but in Figure 2 bottom-right, at layer 12, it is definitely appreciable that the attack has fooled the CNN and the adversarial image I_{4617} now is near to the "wrong" class identified by the green cloud (class 910, *wok*) while the original ones remains within the red cloud of class 138 (*water hen*) and 973 (*cliff*) as expected.

Though with different evidence and at diverse layers of the neural network, such a behavior has been pointed out for all the 5000 image of the test set and this seems to fully verify what has previously been hypothesized.

3.2. Adversarial examples detection

In this sub-section, we have tried to exploit what has been observed in terms of layer activations in order to design some possible features and to demonstrate that they can be useful to detect adversarial examples. To do this, as explained in sub-section 2.3, we have computed the medoids as class representatives for each layer and measured the L_2 distance of every test image from each of them. This leads to a 1000-dimensional vector (being 1000 all the C classes) that evolves throughout the $L = 12$ layers (a 1000-dimensional sequence of length 12). According to this, we have then used an LSTM (Long Short-Term Memory) network, which usually well performs in sequence processing, to decide whether the input sequence originates from an authentic or adversarial image; the network has been tested by subdividing the image set in training set (80%), validation set (10%) and testing set (10%), and by resorting at a K-folding procedure ($K = 5$). LSTM has a hidden state size of 100 and the last one is fed to a fully connected layer with one output followed by a sigmoid activation. Training is done with Adam optimizer for 100 epochs with a

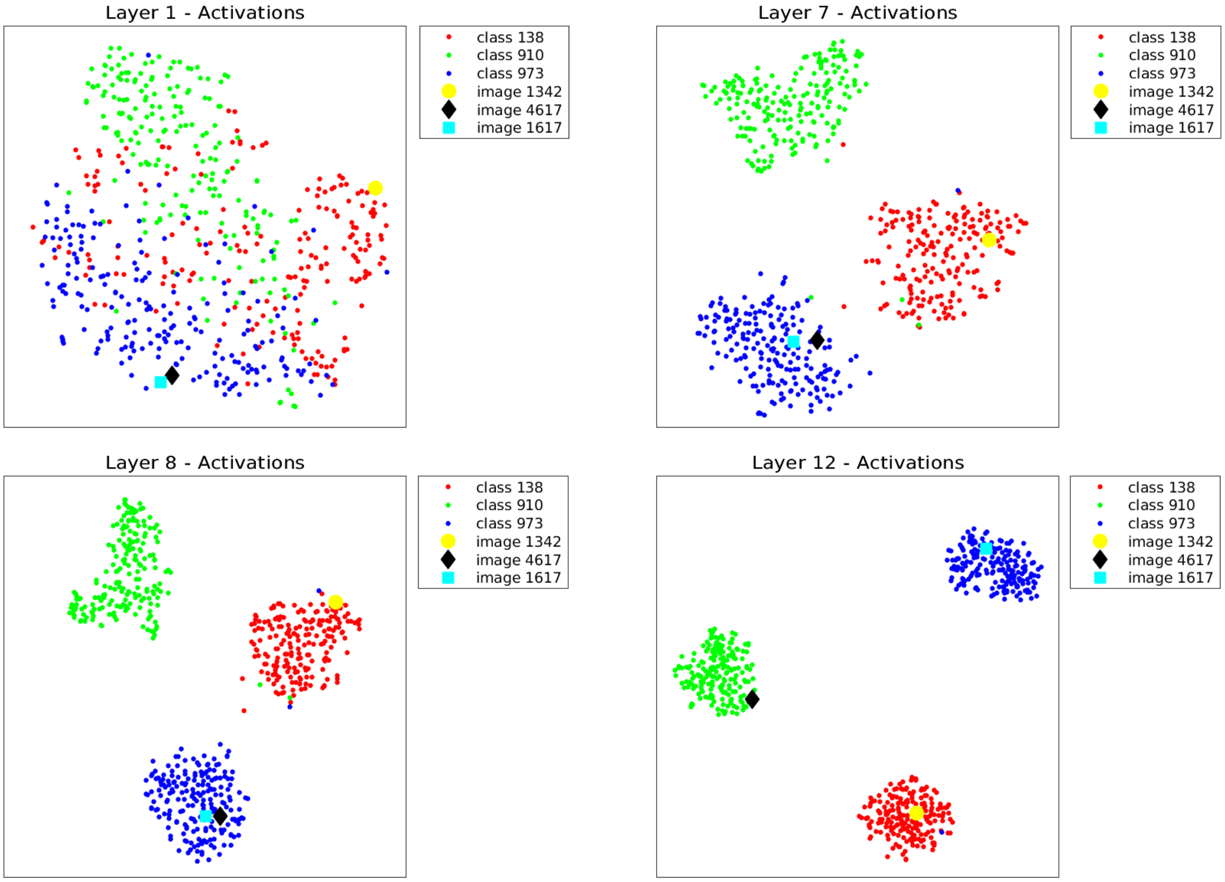

Fig. 2. t-SNE representation of CNN layer activations: I_{1342} (yellow circle) and I_{1617} (cyan square) are original images while I_{4617} (black rhombus) is the adversarial example.

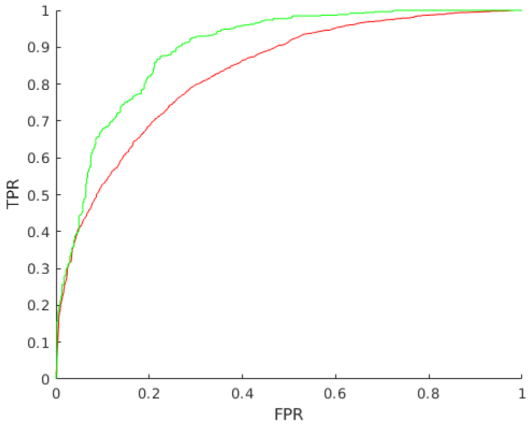

Fig. 3. ROC curves (positive = adversarial): LSTM-based detection (green line) and the method in [16] (red line).

batch size of 100. In Figure 3, the average ROC curve (positive means adversarial) obtained with this approach (green line) is just presented for comparison with another state-of-the-art approach [16] based on *k-NearestNeighbors* only applied on a single layer (red line): a value of 90.4% is achieved in terms of AUC with respect to 83.2%.

This permits to comprehend that the space of internal layer activations can be adopted to extract information regarding the reliability of CNN choices.

4. CONCLUSIONS

This work has presented an analysis to better insight what it happens when an adversarial input is provided to a network trained for image classification. In particular, some theoretical assumptions have been formulated and then experimentally verified by resorting at the activations of the internal layers of a CNN. Finally, it has been demonstrated that such activations can be used to construct some distinctive features to implement a detector for adversarial identification. Future works will be dedicated both to better exploit the potentiality of the activation space and to design more efficient detection solutions.

5. REFERENCES

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples (2014)," *arXiv preprint arXiv:1412.6572*.

[2] M. Barni, M. C. Stamm, and B. Tondi, "Adversarial multimedia forensics: Overview and challenges ahead," in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 962–966.

[3] Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvári, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[4] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," *arXiv preprint arXiv:1511.04508*, 2015.

[5] Weilin Xu, David Evans, and Yanjun Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[6] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku, "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960*, 2017.

[7] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[8] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[9] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[10] Xin Li and Fuxin Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *ICCV*, 2017, pp. 5775–5783.

[11] Nicholas Carlini and David Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, New York, NY, USA, 2017, AISec '17, pp. 3–14, ACM.

[12] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[13] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky, "Neural codes for image retrieval," in *Computer Vision–ECCV 2014*, pp. 584–599. Springer, 2014.

[14] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 512–519.

[15] Fabio Carrara, Fabrizio Falchi, Roberto Caldelli, Giuseppe Amato, Roberta Fumarola, and Rudy Becarelli, "Detecting adversarial example attacks to deep neural networks," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, Florence, Italy, 2017, CBMI '17, pp. 38:1–38:7, ACM.

[16] Fabio Carrara, Fabrizio Falchi, Roberto Caldelli, Giuseppe Amato, and Rudy Becarelli, "Adversarial image detection in deep neural networks," *Multimedia Tools and Applications*, pp. 1–21, 2018.

[17] Mohammadreza Amirian, Friedhelm Schwenker, and Thilo Stadelmann, "Trace and detect adversarial attacks on CNNs using feature response maps," in *8th IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR), Siena, Italy, September 19–21, 2018*. IAPR, 2018.

[18] Fabio Carrara, Rudy Becarelli, Roberto Caldelli, Fabrizio Falchi, and Giuseppe Amato, "Adversarial examples detection in features distance spaces," in *Proceedings of the International Workshop on Objectionable Content and Misinformation (WOCM18)*, Munich, Germany, 8^{th} September 2018, ECCV2018.

[19] Google Brain, "NIPS 2017: competition on adversarial attacks and defenses," https://www.kaggle.com/nips-2017-adversarial-learning-competition.

[20] "Imagenet Large Scale Visual Recognition Challenge 2014," http://image-net.org/challenges/LSVRC.

[21] Nicolas Papernot, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Fartash Faghri, Alexander Matyasko, Karen Hambardzumyan, Yi-Lin Juang, Alexey Kurakin, Ryan Sheatsley, Abhibhav Garg, and Yen-Chen Lin, "cleverhans v2.0.0: an adversarial machine learning library," *arXiv preprint arXiv:1610.00768*, 2017.

[22] Laurens van der Maaten and Geoffrey Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.