FGSM attack and toggle activity

Marco Prattico'

March 2022

Contents

Abstract

Deep Learning is a breakthrough in the current rise of artificial intelligence. Its application can range from self-driving to surveillance and security. Nowadays, research has started to be focused on how to increase the strength of these networks for obvious security reasons.
In this study, we tried to take advantage of the recent discovery of adversarial attacks, to analyze if it's possible to have a dataset, starting from a few images, that activates as many paths as possible inside the network.

1 Introduction

The main goal of these experiments is to analyze the effects of a particular adversarial attack - the *FGSM* (Fast Gradient Sign method) Attack - on a given Deep Neural Network. The network used in these experiments is a pre-trained *Res-net 18* network [1].

During the experiment, we are considering mainly the effects of the attack on the *Toggle Activity* and on the *Neural Ranking* to understand if it is possible to create a dataset that solicits many different paths of the network, using a few images.

2 Background

2.1 Adversarial attack

Adversarial attacks are made to Deep Neural networks in order to fool their prediction. Szgedy *et al.* [2] discovered that - despite their high accuracies - modern deep neural networks are really susceptible to small perturbations to images and, unfortunately, the attacked model shows high confidence in a wrong result. So, from now on, we're going to use the concept of *Adversarial example/image* that indicates a modified version of a clean image, intentionally perturbed, to fool/confuse a machine learning technology, such as deep neural networks [3]. During these experiments, we will be using the FGSM Attack. Goodfellow *et al.*[4] developed a method to compute the perturbation to add to the clean image:

$$\rho = \epsilon \, sign(\nabla(\mathcal{J}(\theta, I_c, \ell)) \tag{1}$$

where $\nabla(.,.,.)$ computes the gradient of the cost function given the parameters θ, I_c, ℓ and the ϵ value denotes a small scalar value and sign(.) the sign function. The FGSM perturbation aims to increase the loss function of the classifier on the resulting image.

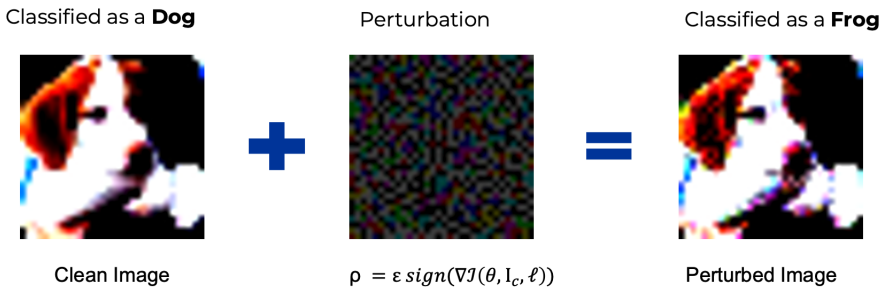

Figure 1: example of an FGSM perturbation

2.2 Toggle activity

Another important topic is *Toggle Activity*, which is a tool for measuring how many neurons change their value from a value below a threshold to another above. In practise, we can see as a percentage how many neurons are activated per layer or in the total network. To do these measurements, we have to measure the average output value of a neuron and we obtain a probability distribution of a neuron's range of values. Next, we can choose to work with binary or ternary thresholding.

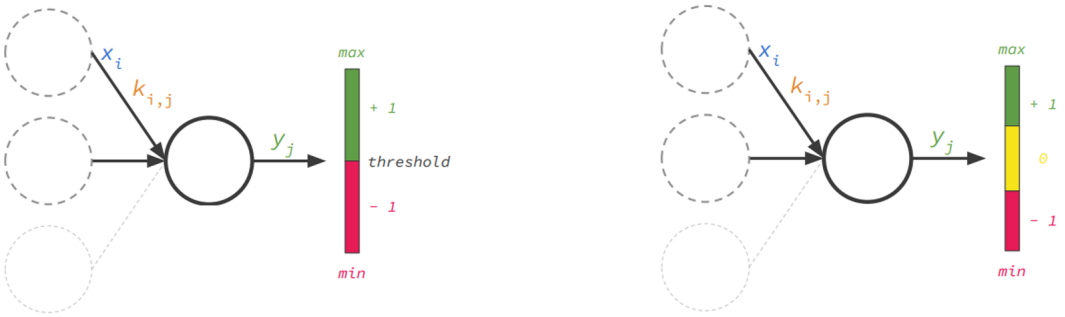

Figure 2: On the left, it is represented the binary thresholding and on the right side, it's represented the ternary.

In the first case, we have to choose just a value as a threshold and then assign the value +1 if the neuron output is above the threshold or -1 if not. In order to choose the best value for the threshold, we need to use techniques such as *kernel density estimation* (or KDE) that estimates the probability distribution of a neuron values. When we obtain the kernel, we select the two most prominent peaks

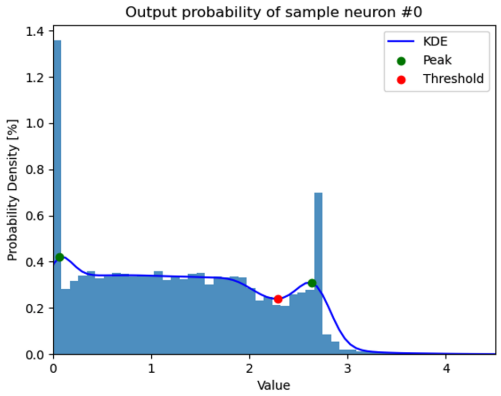

Figure 3: KDE distribution over a neuron output probability.

(green in fig 3) representing the most probable values for which the neuron is on or off. In this case, the threshold is the minimum KDE value (red) between the two peeks.

Related to the ternary threshold, we have to choose two threshold values to assign to each neuron a value in [-1, 0, +1]. In this case, we choose the threshold values as a percentage of the historical maximum, as in fig 4.

The main difference between the two ways of choosing the threshold values is that, for the binary

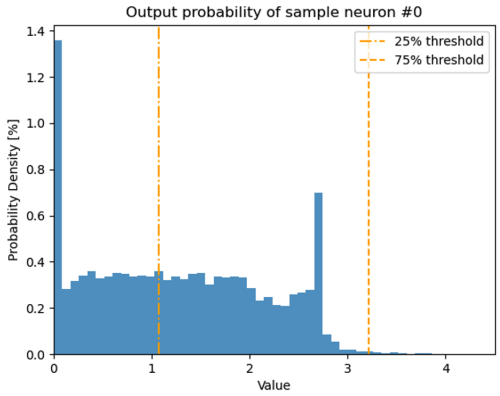

Figure 4: Ternary thresholding using the neuron output probability

case, there is a lower sensibility to the noise, because we are taking into account a probability distribution and not the maximum value. So, the binary case estimates better what actually means have a neuron turned on or off. To the opposite, the ternary thresholding does not capture the nature of the network, since the maximum value of a neuron is unbounded. The ternary case shows an higher toggle activity, because the neuron value can assume three different values (-1,0,+1), instead of two. During the experiments, we used both ternary and binary thresholds.

2.3 Neural ranking

Finally, we have to describe the meaning of *Neural Ranking*. This is a ranking based on the scores assigned to each neuron. The first neurons are those that contribute more to the final prediction [5] and are considered critical neurons. The other neurons at the bottom of the classification are considered redundant.

3 Proposal Approach

3.1 Neural Ranking

The first experiment we have conducted is focused on the effects of the attack on the Neural Ranking of the deep neural network. At the beginning, we have computed the Neural Ranking of the network using a dataset only composed of clean images. We did that starting from the *CIFAR-10* dataset and considering randomly just a subgroup of the whole dataset. Next, starting from the previously created dataset, we created another one composed of both the clean and perturbated versions of the images to see how the Neural Ranking changes.

3.2 Toggle Activity

In this experiment, we analyzed the *Toggle Activity* of the network after the input of different size datasets containing adversarial and clean images. We analyzed the toggle activity of the neural network in three different cases:

1. Using different sizes of datasets, containing clean and adversarial images.

2. Using different datasets of a fixed size and then we averaged the results to avoid particular cases.

3. First, using different values of ϵ and then using different size datasets containing high perturbated images.

3.2.1 Datasets of different sizes

In this case, we used different sizes of clean and adversarial datasets. So, from the *CIFAR-10* dataset, we have created a clean dataset composed of 5, 10, 50, 100, 500 and 1000 images. Then, we have created the adversarial datasets starting from the clean ones. The new datasets are composed of clean and the adversarial version of the images.

3.2.2 Mean of the toggle activity using different datasets

In this particular experiment, we used different datasets - to avoid particular cases - and then we computed the mean of the obtained toggle activity. Here, we created the dataset in a slightly different way from the previous case. In fact, we tried to see what happens if the datasets used are of the same size. So, we created the adversarial dataset starting from the clean one, choosing randomly $n/2$ images and then adding their perturbated version to the dataset. The dataset obtained has the same size n as the clean dataset.

3.2.3 High perturbation

In this case, we added a very high perturbation to the clean images. So, we computed the perturbation using the maximum value of ϵ in eq. 1 - that is 1 because of the normalization of the image. The graphical result is that of the figure below. Then, we created the adversarial datasets in the

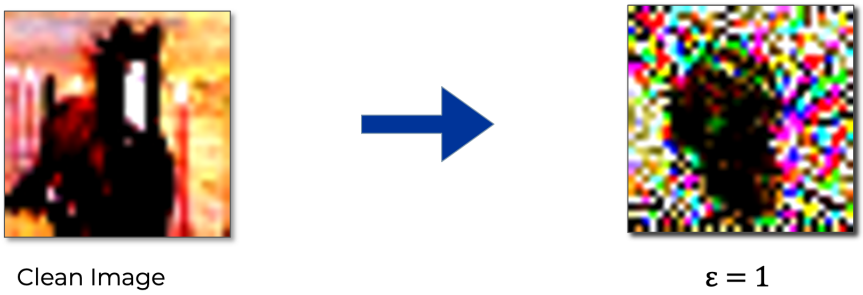

Figure 5: Example of a high perturbation added to a clean image

same way as before, but using high-perturbated images. So, the two datasets, the clean and the perturbated one, are made of the same number of images.

4 Results

4.1 Neural Ranking

In this experiment, we saw from the ranking of the clean dataset that the most critical neurons are positioned more or less in the middle of the network. Whereas, the ranking results using an

adversarial dataset lead us to point out that the most critical neurons are not anymore in the middle of the network, but are now positioned in the final layers of the network. It is interesting to note that the new critical neurons had more or less the same score as in the case with the clean dataset. Therefore, we deduce that the scores of neurons in the middle of the network have been inhibited when the adversarial dataset was used.

We tried to analyze even the case with a dataset composed only of images belonging to the same class, for example, only dog images. We did again the steps we have already described and the obtained results are the same as the previous case.

Rank	LayerName	Dimension_1	Dimension_2	Dimension_3	Value	Class
0	7	9	15	9	9.427.706.718.444.820	1.0
1	7	9	15	8	9.409.834.861.755.370	1.0
2	7	9	15	10	9.398.271.560.668.940	1.0
3	7	9	15	7	9.355.398.178.100.580	1.0
4	7	9	15	11	9.269.206.047.058.100	9.0
5	7	9	15	6	9.260.739.501.953.125	1.0
6	7	9	15	5	9.201.274.871.826.170	1.0
7	7	9	15	12	9.139.973.640.441.890	9.0
8	7	9	15	4	9.116.368.293.762.200	9.0
9	7	9	15	3	9.027.382.850.646.970	9.0
10	7	9	15	13	8.998.499.870.300.290	9.0

Table 1: Neural Ranking using a dataset without perturbed images

Rank	LayerName	Dimension_1	Dimension_2	Dimension_3	Value	Class
0	15	4	2	2	7.724.188.804.626.460	5.0
1	17	21	4	4	7.649.984.836.578.360	2.0
2	15	4	2	3	7.530.786.514.282.220	7.0
3	17	21	4	5	7.508.491.516.113.280	3.0
4	17	21	3	5	7.496.907.711.029.050	7.0
5	17	21	4	3	7.494.227.409.362.790	2.0
6	17	21	3	4	7.472.943.305.969.230	2.0
7	17	21	4	6	7.430.801.391.601.560	9.0
8	15	4	2	4	7.348.335.266.113.280	7.0
9	15	4	1	2	7.315.776.348.114.010	5.0
10	17	21	3	6	7.312.129.497.528.070	9.0

Table 2: Neural Ranking using an adversarial dataset

4.2 Toggle activity

4.2.1 Dataset of different sizes

Here, we computed the toggle activity after the input of both types of datasets, with clean and adversarial images. To make the results clear, we plot the obtained data, as in fig 6

Regarding the left graph, it is evident that there is a difference between the toggle activity in the two cases. This happens because we used a small dataset, and in the *FGSM* case, doubling the dataset size has a non-negligible difference.

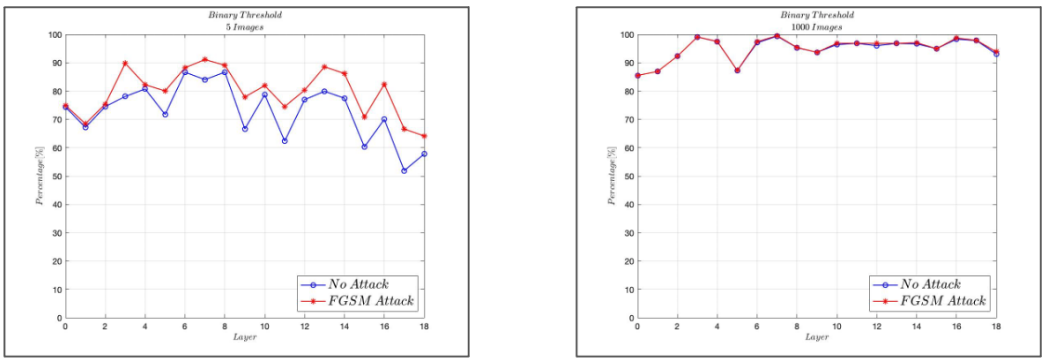

Figure 6: On the left graph it's showed the toggle activity per layer when we used a dataset composed of 5 images for the case called *no attacks*, while - for the *FGSM case* - we used a dataset with 5 clean images and their perturbed version. On the right graphs, we did the same but using 1000 images.

In the second case, the toggle activity is very high due to the large dataset size. We have to underline that doubling the dataset size, adding the adversarial examples, has not an evident effect on the toggle activity.

According to what we said before, it's interesting to see for which dataset sizes the use of the perturbed images gives an evident gain to the network toggle activity. The graph in the figure 7

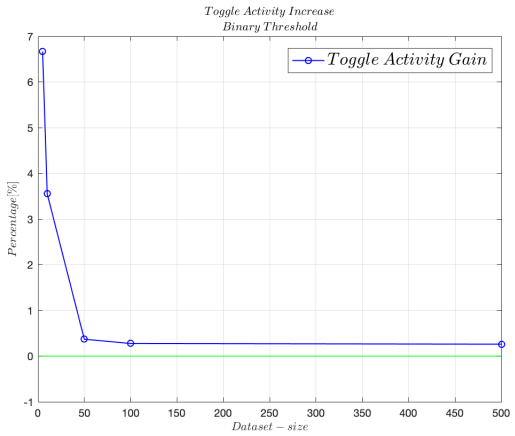

Figure 7: The graph shows what is the gain of the toggle activity using also perturbed images as input

shows that the toggle activity gain decreases rapidly with the increase of the dataset size and tents to 0 asymptotically.

If we consider a ternary threshold, instead of a binary one, it seems the toggle activity gain is more evident, but it's not enough to say that the usage of the perturbed image gives an important improvement to the toggle activity.

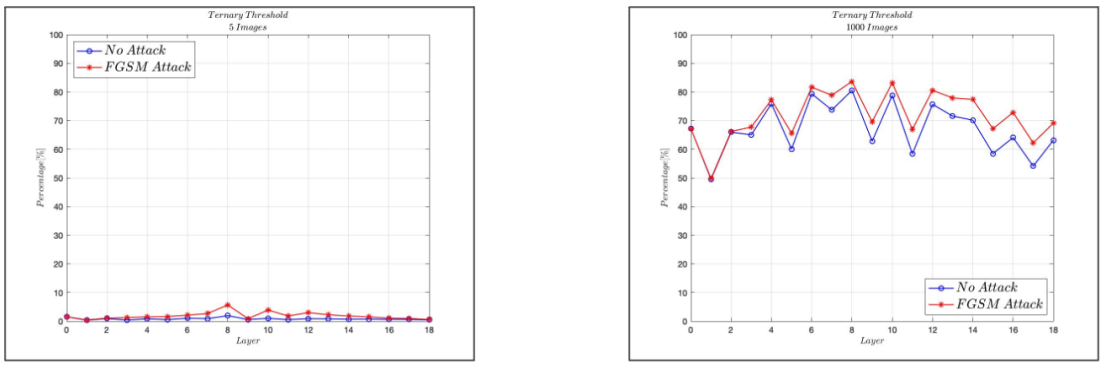

Figure 8: On the left graph it's showed the toggle activity per layer when we used a dataset composed of 5 images for the case called *no attacks*, while - for the *FGSM case* - we used a dataset with 5 clean images and their perturbed version. On the right graphs, we did the same but using 1000 images.

4.2.2 Mean of the toggle activity using different datasets

In this experiment, as mentioned above, we tried to analyze what happens if the clean and the adversarial dataset are of the same size.

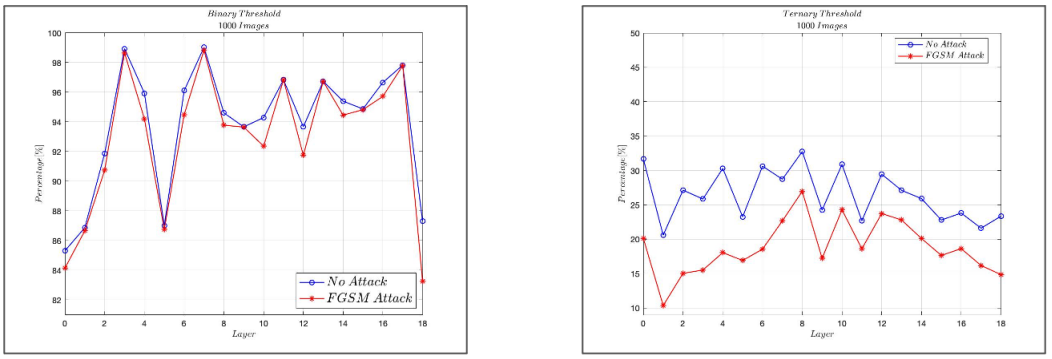

Figure 9: The graphs show a zoom on the toggle activity after the average of the results obtained using different datasets made of 1000 images

From the graph in the figure, it is shown that the toggle activity of each layer using the adversarial dataset is even lower than when we used the clean dataset. From this results, it is clear that an adversarial example solicits more or less the same path of his clean version. So, if we want to increase the toggle activity of a neural network, it is better to insert in the dataset a completely new image instead of inserting an adversarial example of an already existing image inside the dataset.

4.2.3 High perturbation

In this case, after the creation of a dataset composed of high perturbated images, we computed the toggle activity per layer, but the graphs in fig.10 show that there is not much difference compared to the previous case. In fact, the adversarial toggle activity increases slightly but it is still lower than the case with no perturbations.

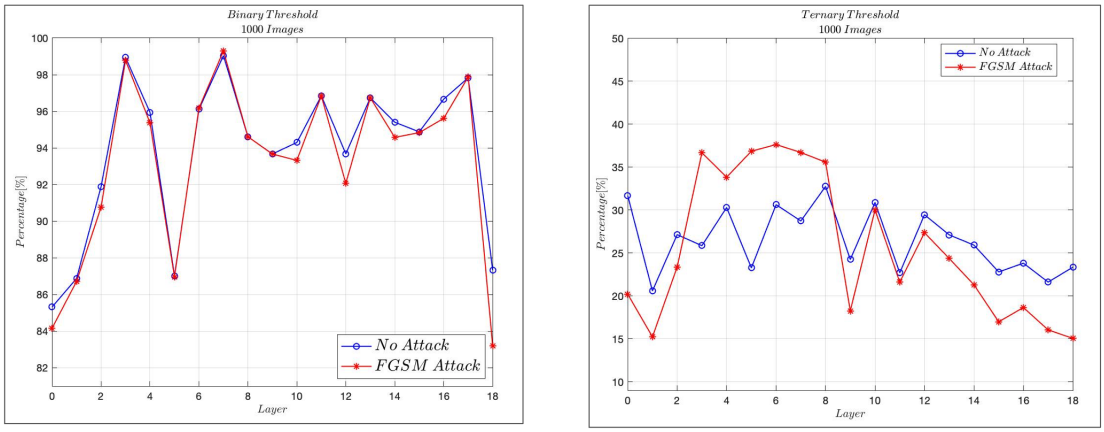

Figure 10: Toggle activity per layer using a dataset made of only clean images and using a dataset with clean and high perturbated images.

For this reason, we tried to see how the total toggle activity changes compared to the change in the value ϵ. From the figure 11, it is evident that the maximum toggle activity gain is around 5% for the binary case and less than 1% for the ternary threshold case.

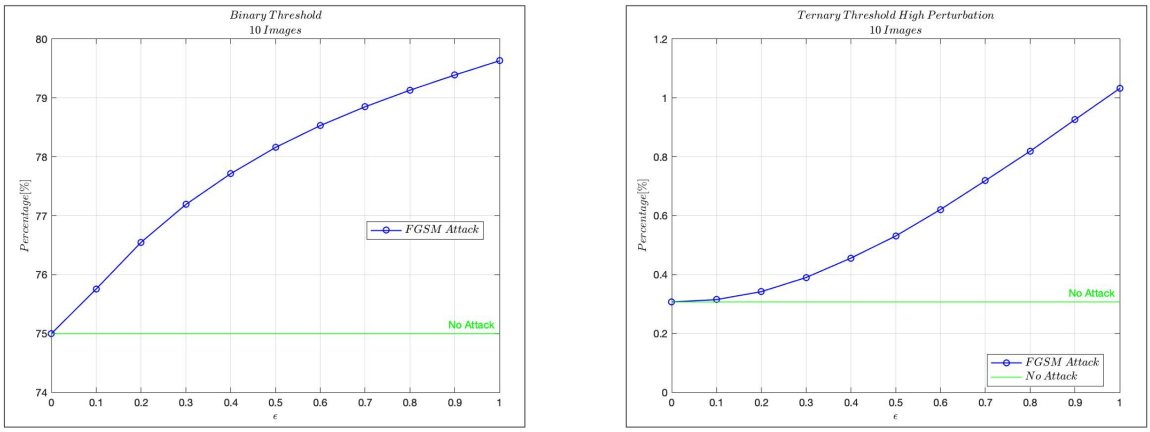

Figure 11: Total toggle activity changing the ϵ value.

5 Conclusions

In conclusion, the use of the FGSM attack does not significantly increase the toggle activity of the neural network. Therefore, we cannot use the FGSM attack to create a small dataset that solicits more paths than an equivalent-sized dataset without the adversarial images. In fact, the goal of adversarial attacks is to add a small perturbation that results imperceptibly to the human being, but it is able to fool the neural ranking prediction. Then, it makes sense that the adversarial and the clean image activate more or less the same paths without clearly changing the toggle activity.

This conclusion leads to other possible ways to see the problem. In fact, we can analyze the different shapes of the toggle activity curves of both datasets – with and without the adversarial images – to understand which neurons make the prediction change. From this, we can improve the strength and robustness of a neural network against adversarial attacks.

References

[1] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[2] Christian Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014. URL: http://arxiv.org/abs/1312.6199.

[3] Naveed Akhtar and Ajmal Mian. "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey". In: *CoRR* abs/1801.00553 (2018). arXiv: 1801.00553. URL: http://arxiv.org/abs/1801.00553.

[4] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. 2015. URL: http://arxiv.org/abs/1412.6572.

[5] Annachiara Ruospo and Ernesto Sanchez. "On the Reliability Assessment of Artificial Neural Networks Running on AI-Oriented MPSoCs". In: *Applied Sciences* 11.14 (2021). ISSN: 2076-3417. DOI: 10.3390/app11146455. URL: https://www.mdpi.com/2076-3417/11/14/6455.