Efficient Computation of Robustness of Convolutional Neural Networks

Paolo Arcaini  
National Institute of Informatics  
Tokyo, Japan  
arcaini@nii.ac.jp

Andrea Bombarda  
University of Bergamo  
Bergamo, Italy  
andrea.bombarda@unibg.it

Silvia Bonfanti  
University of Bergamo  
Bergamo, Italy  
silvia.bonfanti@unibg.it

Angelo Gargantini  
University of Bergamo  
Bergamo, Italy  
angelo.gargantini@unibg.it

Abstract—Validation of CNNs is extremely important, especially when they are used in safety-critical domains. In particular, in the latest years, the focus of validation has been put on assessing the robustness of CNNs, i.e., their ability to correctly classify perturbed input data. A way to measure robustness is to check the network accuracy over many datasets obtained by altering the input data in different ways, but this is time and resource-consuming. In this paper, we present ASAP, a method to efficiently compute the robustness of a CNN, exploiting a parabola-based approximation which allows to adaptively select only relevant alteration levels. The method is tested on two different benchmarks (MNIST and breast cancer classification). Moreover, we compare ASAP with other techniques based on uniform sampling, numerical integration, and random sampling.

Index Terms—Convolutional Neural Networks, robustness, efficient robustness computation, image classification, alteration

I. INTRODUCTION

Convolutional Neural Networks (CNNs) are increasingly used to perform different activities [3], among which many of them are safety-critical [14] as, e.g., in autonomous driving [10], or in the medical practice [8]. Thus, especially in these cases, validation activities must be performed [20]. The most recent developments in the validation of CNNs are based on the evaluation of the robustness, i.e., the ability of the network to correctly classify slightly altered inputs. However, the majority of research works are focused on adversarial examples, which are often created by exploiting the internal structure of the network [16], and may not reflect plausible inputs that could occur during real network usage [12], [13], [18]. Instead, the robustness should be defined by considering real alterations that may occur to input data as well. In [4], we proposed a robustness definition for CNNs that considers the alterations that are typical of the domain in which the CNN is used. Moreover, in [5], we presented ROBY, a Python tool for automatic robustness analysis, implementing the robustness definition and supporting different types of input data.

Computing robustness can be really expensive in terms of time and resource consumption, since every input of the test dataset has to be repeatedly altered with different alteration levels. One way to obtain a correct estimation of the network robustness is by increasing the number of alteration levels analyzed. However, this has the drawback of increasing the time required for the analysis. Thus, in this paper, we propose ASAP (Adaptive Sampling by Parabolic estimation), a novel method to estimate the robustness of a CNN, aiming at computing robustness more efficiently. It is based on a parabolic approximation of the accuracy curve and allows, by means of specific parameters, to trade off the precision of the computed robustness against the time required to compute it.

We assess ASAP on two different benchmarks, namely the MNIST dataset recognition and the breast cancer classification. Furthermore, we compare the approach with other possible techniques for robustness computation based on uniform sampling, numerical integration, and random sampling.

Paper structure. Sect. II introduces the definition of robustness and highlights the limits of existing approaches for its calculation. Sect. III presents ASAP. Sect. IV introduces the experiments design, with the two benchmarks used for the evaluation. Sect. V evaluates ASAP w.r.t. other available techniques. Finally, Sect. VI reviews some related work, and Sect. VII concludes the paper.

II. BACKGROUND AND BASIC DEFINITIONS

A convolutional neural network (CNN) is a type of deep neural network, mainly used to analyze images, which uses the linear mathematical operation convolution (instead of the regular matrix multiplication) in at least one of its layers. Digital images taken as input of the CNN can be susceptible to modifications during their acquisition. In [4], we proposed a robustness definition that assesses how much input modifications influence the classification accuracy; the definition is built on the concept of alteration defined as follows.

Definition 1 (Alteration). An alteration of type $A$ of an input $t$ is a transformation of $t$ that mimics the possible effect on $t$ when a problem during its acquisition, or in its elaboration, occurs in reality. In the following, we identify with $P^A$ the set of data obtained by altering all the input data in $P$ with an alteration of type $A$ of level $l \in [L_A, U_A]$, where $[L_A, U_A]$ is the range of plausible alteration levels of type $A$.

The robustness of a CNN is then defined as its ability to correctly classify altered data. Intuitively, given an alteration
interval \([L_A, U_A]\), the robustness is the portion of that interval in which, if we classify the altered data with the same CNN, the accuracy is still acceptable, i.e., it is above a given threshold \(\Theta\). In Fig. 1, the accuracy is the black curve, and the robustness is the total length of the intervals in which the accuracy is above the threshold (gray line), divided by the total length of the alteration interval. The robustness is formally defined as follows.

**Definition 2 (Robustness).** Let \(\Theta\) be a threshold representing the minimum accepted accuracy. The robustness of a CNN \(C\) w.r.t. alteration of type \(A\) in the range \([L_A, U_A]\) (using a set of inputs \(P\)) is defined as the percentage of alteration values for which the accuracy \(acc(C, P^k)\) is above \(\Theta\). Formally:

\[
\text{rob}_A(C, P) = \frac{\int_{L_A}^{U_A} H(acc(C, P^l) - \Theta)dl}{U_A - L_A}
\]

where \(H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}\)

Computing precisely the robustness using this formula (where \(H\) is the Heaviside function) is very difficult. Indeed, the accuracy function is not known a priori and, therefore, it should be computed for all the alteration levels \(l\) in \([L_A, U_A]\) which could be many, if not infinite. Moreover, computing the accuracy of the network \(C\) when a single alteration level \(A_l\) is applied to the set \(P\), requires a considerable effort, so a method to select suitable alterations is necessary in order to reduce the computation time.

A naive solution is to uniformly sample in \([L_A, U_A]\) and compute the accuracy for the sampled points. We followed this approach in [4], where we proposed a robustness definition exploiting equidistributed discrete points. In that case, we count for how many points (\(n_{\text{acc}}\)) the accuracy is acceptable, relatively to the total number of sampled points \(n\). Formally:

**Definition 3 (Uniform robustness).** Given \(n\) equidistributed points \(SP = \{l_1, \ldots, l_n\}\) sampled in the interval \([L_A, U_A]\), the uniform robustness is defined as:

\[
\text{rob}_A(C, P) = \frac{n_{\text{acc}}}{n} = \frac{|\{l \in SP \mid acc(C, P^l) \geq \Theta\}|}{|SP|}
\]

Note that also numerical integration methods for integral calculation rely on sampling of points and, so, they could be used to compute robustness. However, in their application, the user has little or no control over the points to be used to compute the accuracy function, and this may result in oversampling not relevant areas and/or undersampling relevant ones. We will also compare with this approach in our experiments.

### A. The limits of the uniform sampling approach

The robustness, as defined in Def. 3, may still require a lot of computational power and time to be evaluated, especially when the alteration levels are many. In fact, given \(n\) the number of alteration levels to be applied for an alteration \(A\), \(k\) the number of inputs, and \(t_A\) the time required for applying the alteration \(A\) to a single image, the total time required to perform robustness analysis is \(t_{\text{tot}} = n \cdot k \cdot t_A\). For instance, if \(n = 1000\), \(k = 1000\) and \(t_A = 0.1\) sec, the total time required to compute the robustness is approximately 28 hours.

One way to reduce the effort is to reduce \(n\), i.e., the number of sampled levels for the alteration \(A\). This is a viable solution, but has some drawbacks, especially when analyzing networks whose accuracy varies a lot. For example, Fig. 2 shows the evaluated points when uniform sampling of an accuracy function is performed with \(n = 10\). Using Def. 3, we would compute a robustness of \(\text{rob}_A(C, P) = 100\%\). Nonetheless, the real value of the robustness for the analyzed network is significantly lower (\(\approx 50\%)\).

Thus, a way to solve these problems is to adaptively select the points to be sampled (possibly not uniformly), as usually done for input values in software testing. In fact, choosing the correct input parameters and the correct values to be tested in a program is challenging because different inputs may lead to different bug discoveries; however, sampling some inputs is required since exhaustive testing can not be performed.

### III. Adaptive Sampling by Parabolic Estimation

In order to tackle the limitations of the uniform sampling approach, we propose the ASAP algorithm to automatically select the points where to evaluate the accuracy. The best points to select would be those in which the accuracy curve intersects the threshold \(\Theta\). However, since we do not know the analytical form of the curve and we can not compute these intersections, we try to select points as close as possible to \(\Theta\).

ASAP is based on the assumption that, once we have computed the accuracy for two alteration levels \(A\) and \(B\), the real accuracy curve between \(A\) and \(B\) will be included in the
area between two parabolas passing through the points A and B, and having concavity depth respectively +\( \hat{a} \) and −\( \hat{a} \) (see Fig. 3), i.e., with equation \( y = ax^2 + bx + c \) with \( a = \pm \hat{a} \). If there is an intersection between that area and the threshold \( \Theta \), and the distance between A and B is sufficiently large, then we compute the accuracy in the middle point \( M \) between A and B, we add it to the sample set, and we recursively apply the same procedure to the two intervals \([A, M]\) and \([M, B]\). In this way, the number of evaluated points is adaptively determined and depends on the value of the parameter \( \hat{a} \). Intuitively, the higher is the value of \( \hat{a} \), the higher is the number of alteration levels evaluated by the algorithm. Thus, users must choose the \( \hat{a} \) value based on the robustness estimation accuracy needed, and on the time available for robustness analysis.

Algorithm 1 describes how the approximation method works. It recursively considers two alteration levels \( x_A \) and \( x_B \) in \([L_A, U_A]\), and evaluates the accuracy of the model in the two points, using the \( \text{getAccuracy} \) function which applies the selected level of alteration (lines 2-3) to the test set \( TS \).

Then, the algorithm checks, using the function \( \text{parabIntsect} \) (line 4), whether at least one of the two parabolas passing for the two points intersects the threshold. A sufficient condition is that the two accuracy values are opposite w.r.t. the threshold \( \Theta \) (see an example in Fig. 4(a)): this is checked at line 11. If this is not the case, i.e., both accuracy values are above or under the threshold \( \Theta \) (see examples in Figs. 4(b)-(c)), the algorithm computes the parabolas passing for A and B and having concavity depth \( \pm \hat{a} \), using the \( \text{parab} \) function (line 14). Parabola coefficients \( b \) and \( c \) are obtained with this system of equations (where \( a \) is passed as argument):

\[
\begin{align*}
\frac{a \cdot x_A^2 + b \cdot x_A + c}{a} &= acc_A \\
\frac{a \cdot x_B^2 + b \cdot x_B + c}{a} &= acc_B
\end{align*}
\]

Then, the parabola vertex \( V(x_v, y_v) \) is computed (line 15). The method verifies that the parabola is in the area of interest (line 16), by checking that: (i) \( x_v \in [x_A, x_B] \) (first operand of the conjunction), and (ii) the parabola intersects the threshold \( \Theta \), i.e., \( y_v \) is opposite to \( acc_A \) w.r.t. \( \Theta \) (second operand).

Finally, if one of the parabolas intersects the threshold and the sampled points are not too close (line 5), the computation is recursively repeated in the intervals \([x_A, x_M]\) and \([x_M, x_B]\) (lines 6-7), where \( x_M \) is the average alteration level between \( x_A \) and \( x_B \).

Starting from the set of computed points with their accuracy values \( RES = \{(l_1, acc_1), \ldots, (l_n, acc_n)\} \), the robustness is computed by generalizing the formula in Def. 3 as follows:

\[
\text{rob}_A(C, P) = \sum_{j=2}^{n} H(acc_j - \Theta) \cdot (l_j - l_{j-1}) / U_A - L_A
\]
Example 1. Fig. 5 shows the effect of ASAP on two different synthetic functions. In Fig. 5(a), the robustness is computed by using uniform sampling with 50 equidistributed alteration levels, obtaining a robustness of 38.8%, while in Fig. 5(b) only 15 levels are used by ASAP (while $\hat{\alpha}$ equal 256) and a robustness of 39.06% is obtained. Note that the two results are very close and that the real robustness associated with the accuracy plot shown in Fig. 5(a) and Fig. 5(b) is 40.00%. The same behavior can be observed between Fig. 5(c), where robustness 77.9% is obtained by uniform sampling with $n = 50$, and Fig. 5(d), where ASAP uses only 34 alteration levels (focused in the area near the threshold value instead of uniformly distributed ones), obtaining a robustness of 81.2%. Note that, also in this case, the two results are close, and that the real robustness associated with the accuracy plot shown in Fig. 5(c) and Fig. 5(d) is 80.4%. This shows that ASAP uses fewer alteration levels, so it saves time, but it still provides an accurate approximation of the robustness.

A. Maximum error estimation of the computed robustness

Our method provides theoretical guarantees regarding the maximum error that it can do in computing the robustness. To define this, we first need to select from $RES$ (which contains every $i$-th sampled point $p_i = (l_i, acc_i)$) the pairs of two consecutive points $p_j$ and $p_{j+1}$ such that the parabolas passing from them with concavity depth $\pm \hat{\alpha}$ intersect the threshold $\Theta$, i.e.,

\[
IP = \left\{ (l_j, l_{j+1}) \mid (l_j, acc_j, l_{j+1}, acc_{j+1}) \in RES \wedge \right.
\]

\[
\left. \left( parabIntsect(l_j, acc_j, l_{j+1}, acc_{j+1}, \hat{\alpha}, \Theta) \lor \right) \right. \]

\[
\left. \left( parabIntsect(l_j, acc_j, l_{j+1}, acc_{j+1}, -\hat{\alpha}, \Theta) \right) \right. \}
\]

Intuitively, a pair of points $p_j$ and $p_{j+1}$ in $IP$ identifies the points between which at least one of the two parabolas intersects the threshold, and, therefore, also the real curve may intersect, but ASAP has quit sampling because the two points have alteration levels sufficiently close ($l_{j+1} - l_j < minStep$).

Assuming that we used an appropriate value $\hat{\alpha}$ for ASAP, the error we can do in computing the robustness only comes from the intervals identified by the points in $IP$. This intuition is formalized in the following theorem.

**Theorem 1.** Let $C$ be a CNN and $A$ a given alteration type defined in the range $[L_A, U_A]$. Let $rob_A$ be the robustness computed for $C$ and $A$ by ASAP using a given $\hat{\alpha}$. Let $\hat{rob}_A$ be the real robustness value. Under the assumption that $\hat{\alpha}$ is a suitable parameter, i.e., the real accuracy curve is included in the areas of two parabolas with concavity depth $\hat{\alpha}$ (see Fig. 3), the maximum error of the computed robustness has a guaranteed upper bound defined as follows:

\[
|rob_A - \hat{rob}_A| \leq \varepsilon_A \quad \text{with} \quad \varepsilon_A = \frac{\sum (l_{j+1} - l_j)_{IP}}{|U_A - L_A|} \quad \text{(2)}
\]

**Proof.** The error in robustness computation is due to the cases in which the real curve crosses the threshold line but ASAP fails to find the exact intersection point. Let’s consider where this can happen by considering all the sub-intervals $[l_j, l_{j+1}]$ of the points in $RES$:

- if $(l_j, l_{j+1}) \notin IP$, then the parabolas with concavity depth $\pm \hat{\alpha}$ do not intersect the threshold. Since, by theorem assumption, $\hat{\alpha}$ is a suitable parameter, the real curve is included in the computed parabolas, and so it also does not intersect the threshold. So, no contribution of error in robustness computation comes from these points.
- if $(l_j, l_{j+1}) \in IP$, then we can distinguish two cases:
  - the two points are opposite w.r.t. the threshold line. So, the real curve does intersect the threshold line, but in an unknown point which does not belong to $RES$.
  - the two points are both below or above the threshold: ASAP ignores the possible intersection of the real curve with the threshold line, since for ASAP the sampled points are close enough.

In both cases, the maximum absolute error is $l_{j+1} - l_j$. So, the total error in robustness is given by the sum of the errors for all the pairs of points in $IP$. Hence, the upper bound of the error is as defined in Eq. 2.

**IV. Experiments Design**

**Benchmarks**

We have chosen the following two different benchmarks to evaluate the efficiency of ASAP:

**MNIST Dataset:** The MNIST (Modified National Institute of Standards and Technology database) dataset is a well-known dataset containing a lot of images of hand-written number digits [2]. It is shipped with Keras, and we have tested the approaches presented in this paper using the first 1000 images in the test set over a publicly available model [1].
Breast cancer classification: Breast cancer diagnoses, in particular for Invasive Ductal Carcinoma (IDC), are based on the analysis of images of histological features of tissue or cells removed with surgery or biopsy. These images are captured by a microscope and examined by pathologists to make a decision about the benignity or the malignancy of the suspected cancer. We have performed a robustness analysis using input images coming from a publicly available dataset curated by [11], and the same model we presented in [4].

Oracle definition by uniform sampling

In order to assess the performance of ASAP, we need to check how close the computed robustness values are to the "correct" values. So, in order to define a suitable oracle to perform the necessary comparisons, we have computed the robustness for the case studies using the uniform sampling technique with a very large number of points \( n = 1024 \) (using the tool ROBY [5]). Tab. I reports the alterations applied to both benchmarks, together with the robustness results and the required time of the oracle, when using accuracy threshold \( \Theta = 0.8 \). Note that the results obtained in this way can be reasonably considered as oracle: indeed, we have observed that, with \( n = 1024 \), the accuracy function is very stable and the alteration step becomes between \( 10^{-3} \) and \( 2 \times 10^{-3} \), which is small enough for the defined alterations.

V. Experimental Evaluation

To evaluate ASAP, we have performed several experiments on the benchmarks presented in Sect. IV. We have applied it with \( n = 1024 \) and \( \tilde{a} = 128 \) using the proposed alterations. Moreover, we have also applied, as comparison approaches, numerical integration, and uniform sampling with the same number of points actually used by ASAP.

For each experiment, we have recorded the assessed robustness, the number of points required for the estimation, and the required time (see Tab. I). \( \Delta T \) is the relative change in percentage of the time taken to compute the robustness by using each analyzed technique w.r.t. the oracle, while \( \Delta R \) is the difference between the robustness values. Negative values of \( \Delta T \) indicate a saving of time while a negative value of \( \Delta R \) means an underestimation of the robustness.

In particular, using the gathered values, we are interested in answering the following research questions:

**RQ1** Is ASAP efficient and effective?

**RQ2** Is ASAP more effective than:

- **RQ2.1** numerical integration techniques?
- **RQ2.2** uniform sampling with the same number of points?
- **RQ2.3** randomly selecting the alteration levels?

**RQ3** Is the number of alteration levels actually used by ASAP correlated with the maximum number of points allowed (\( n \))? 

**RQ4** How do the values of parameters \( n \) and \( \tilde{a} \) influence the accuracy of the robustness estimation?

**RQ5** Is the estimation error provided by Theorem 1 reliable?

**RQ1: Is ASAP efficient and effective?**

Each alteration entails different accuracy curve shapes for which ASAP may lead to different results in terms of time saving and error in the robustness estimation w.r.t. the oracle. From Tab. I, it can be seen that a significant reduction of the number of points is obtained with both benchmarks and all the alterations. Moreover, the flatter the accuracy alteration curve is, the higher is the decrease in time using ASAP, since it uses fewer points than the oracle and it is more probable that the computed parabolas with the chosen \( \tilde{a} \) do not intersect the threshold. In terms of accuracy of the robustness estimation, compared to the oracle, the error is always under 2\%. In conclusion, ASAP always allows estimating the robustness with
RQ2.1: Is ASAP more effective than numerical integration techniques?

In Def. 2, we have presented the general formula for computing robustness using an integral computation of the Heaviside function. One could apply numerical integration techniques that are able to automatically sample the points where to compute the accuracy for computing the integral. However, these techniques may not find the right alteration levels and so correctly compute the integral, since the Heaviside function is discontinuous and it hides the accuracy curve as its argument. For this reason, we wanted to assess the feasibility and efficiency of numerical integration techniques to compute the robustness, and compare their performance with ASAP.

We have applied the `scipy.integrate.quad` function which uses adaptive quadrature and allows to set the maximum number of points in which function to be integrated can be evaluated. We have set this number to 1024, in order to allow a fair comparison with ASAP and the oracle.

Tab. I reports the results obtained with the experiments. The cases in which the numerical algorithm cannot compute the integral with the given maximum number of points are considered timeout. For the breast cancer benchmark, two alterations gave timeout as result. When comparing the results of numerical integration with the ones obtained by ASAP, we can see that, for the breast cancer classification, ASAP always gives better results, i.e., |ΔRob_A| < |ΔRob_U|, except for the Zoom alteration in the MNIST benchmark, and for the Gaussian Noise in the Breast Cancer classification.

RQ2.2: Is ASAP more effective than randomly selecting the alteration levels?

Evaluating the robustness using as few alteration levels as possible is of key importance to speed up the estimation process. A possible solution to reduce the number of evaluated alteration levels could be using random sampling and fixing the number of points to be evaluated. We have applied it to the breast cancer benchmark, using different values of n, and repeating each experiment for 100 times. Fig. 7 reports the results for vertical translation and brightness alterations over the analyzed benchmark. The robustness reached applying the vertical translation (see Fig. 7(a)) is 100%. In fact, the variance of the estimation is 0, even when using fewer points. Instead, for the brightness alteration shown in Fig. 7(b), the use of fewer random sampled points leads to a great variance. A similar trend can be observed for all the other alterations. We can conclude that the performance of the random sampling approach depends directly on the robustness w.r.t. the considered alteration. If the robustness is ≃ 100% or ≃ 0%, randomly choosing the points in which to compute the accuracy leads to accurate results with low variance. In all other cases, random sampling can imply a high variance and less accurate results.
RQ3: Is the number of alteration levels actually used by ASAP correlated with the maximum number of points allowed (n)?

Using ASAP, the user can set the value of n, i.e., the maximum number of points to be used. Increasing its value means that a lower interval minStep (see Alg. 1 for reference) allowed between two adjacent points is used. However, depending on the value of n, and on the $\hat{a}$ fixed, the algorithm may use fewer points than the maximum allowed, since it stops adding points if no intersection between the parabolas and the threshold is found. Thus, we are interested in defining how ASAP reacts to a variation of n in terms of used points. We have analyzed the presented benchmarks and all the alterations using different values of n. We have found that the number of actually used points depends on the alteration and on how the network accuracy curve changes when a defined alteration level is applied: the flatter the curve of accuracy is, the lower the number of points used by ASAP is (Fig. 8). As a rule of thumb, the higher $\hat{a}$ and n are, the higher the number of considered alteration levels is, as shown by Fig. 9, since minStep is lower. Moreover, Fig. 9 shows that the lower $\hat{a}$ is, the lower the advantage of increasing the value of n is since the number of used points for analysis does not increase significantly.

RQ4: How do the values of n and $\hat{a}$ influence the accuracy of the robustness estimation?

The ASAP method is based on two parameters that are configurable by the user in order to adapt to different networks: n and $\hat{a}$. The former is used to evaluate the distance between two points on the x-axis, i.e., to compute the minStep value. The latter allows to include the evaluation of the distance between points and the threshold $\Theta$. Thus, choosing different values for the two parameters may lead to a different accuracy in the robustness estimation. We have performed several tests on the two benchmarks varying the values of $\hat{a}$ and n, to evaluate their effect on the robustness estimation. The results are shown in Fig. 10. The analysis indicates that the robustness estimation is more accurate with high n and $\hat{a}$. Both the parameters, in fact, contribute to increasing the number of alteration levels evaluated for robustness estimation and, consequently, the accuracy of ASAP.

RQ5: Is the estimation error provided by Theorem 1 reliable?

Theorem 1 gives an upper bound of the possible error done using ASAP, under the assumption that the value of $\hat{a}$ has been chosen correctly. Since for the two benchmarks presented in this paper, we already have the oracle values, we can assess if the error estimations given by the theorem are near to the actual errors. Tab. II shows the error done by ASAP ($\Delta \text{Rob}_A$) and the upper bound of the error computed by the theorem ($\varepsilon_A$). For almost all of these results, $\varepsilon_A$ is very close to $|\Delta \text{Rob}_A|$ and $|\Delta \text{Rob}_A| < \varepsilon_A$, except for the Zoom alteration on the MNIST benchmark. Since Theorem 1 assures that $\varepsilon_A$ is an upper bound for the error under the

Table II

<table>
<thead>
<tr>
<th>Alteration</th>
<th>$\Delta \text{Rob}_A$ [%]</th>
<th>$\varepsilon_A$</th>
<th>$\Delta \text{Rob}_A$ [%]</th>
<th>$\varepsilon_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Noise</td>
<td>0.00</td>
<td>0.00</td>
<td>-0.54</td>
<td>0.78</td>
</tr>
<tr>
<td>Compression</td>
<td>-0.10</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Vertical Trans.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Horizontal Trans.</td>
<td>-0.10</td>
<td>0.10</td>
<td>-0.61</td>
<td>7.71</td>
</tr>
<tr>
<td>Blur</td>
<td>-0.10</td>
<td>0.10</td>
<td>-0.04</td>
<td>0.10</td>
</tr>
<tr>
<td>Brightness</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>Zoom</td>
<td>1.72</td>
<td>0.00</td>
<td>0.29</td>
<td>0.00</td>
</tr>
</tbody>
</table>

37
assumption of the correct choice of $\hat{\alpha}$, having $|\Delta \text{Rob}_A| > \varepsilon_A$ means that $\hat{\alpha}$ has not been chosen correctly for that particular case. In fact, increasing $\hat{\alpha}$ and using $\hat{\alpha} = 256$, we have obtained $\Delta \text{Rob}_A = 0.03$ which is below the new upper bound $\varepsilon_A = 0.30$.

VI. RELATED WORK

Efficient analysis is an important topic when it comes to neural networks. In fact, training, validation, and testing activities require a lot of time, especially when the models need to be trained and tested over a great number of inputs. A well-known solution adopted for decreasing the test time is the one presented in [7], where the authors have proposed a method to reduce data to train, test, and validate neural networks, which is based on a stratified sampling of input data. Other approaches are based on data pre-processing, in order to reduce the dimensionality of input data, such as in [9], where RBF networks are analyzed and a separability-correlation measure is introduced to define which inputs are irrelevant for the classification. These approaches are different from the one proposed in this paper, since with ASAP we reduce the sampled points but not the input. Indeed, this is one of the advantages of ASAP, since by not reducing the input data, the user can be sure about testing every possible input feature. Efficient computation of robustness, even if referred to adversarial examples, is performed by several tools such as CNN-Cert [6], using state-of-the-art algorithms (Fast-Lin and CROWN [19]), or PROVEN [17] which exploits CNN-Cert and a probabilistic approach to reduce the time of computing adversarial robustness.

CNNs are often part of more complex Machine Learning based Systems (MLSs) [15] (e.g., autonomous driving); hence, when testing an MLS, also its embedded CNN must be tested. Since MLSs are complex systems, they must be thoroughly tested; however, literature [15] reports that also the cost of testing MLSs is usually high. As future work, we plan to investigate whether ASAP can improve the efficiency of MLS testing, without degrading its effectiveness.

VII. CONCLUSIONS

In this paper, we have introduced ASAP, an efficient way to compute the robustness of a CNN w.r.t. unforeseen (but plausible) input modifications. It exploits the reduction of the number of alteration levels, based on the assumption that the accuracy of the model between two points will likely be included in a parabola-delimited region. Users can adapt the approximation method to their networks using two different parameters: the maximum points $n$ to be computed, and the concavity depth $\hat{\alpha}$ of the evaluated parabolas. We have shown that the proposed solution allows estimating the robustness of a CNN in a more efficient way than the standard (based on uniform sampling or numerical integration) and the random sampling approaches, guaranteeing a small error. As future work, we are investigating solutions for the automatic computation of a suitable value for the parameter $\hat{\alpha}$, in order to avoid errors due to a wrong parameterization of ASAP.