nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-019-0104-6

Adversarial explanations for understanding image classification decisions and improved neural network robustness

Walt Woods*, Jack Chen* and Christof Teuscher*

For sensitive problems, such as medical imaging or fraud detection, neural network (NN) adoption has been slow due to concerns about their reliability, leading to a number of algorithms for explaining their decisions. NNs have also been found to be vulnerable to a class of imperceptible attacks, called adversarial examples, which arbitrarily alter the output of the network. Here we demonstrate both that these attacks can invalidate previous attempts to explain the decisions of NNs, and that with very robust networks, the attacks themselves may be leveraged as explanations with greater fidelity to the model. We also show that the introduction of a novel regularization technique inspired by the Lipschitz constraint, alongside other proposed improvements including a half-Huber activation function, greatly improves the resistance of NNs to adversarial examples. On the ImageNet classification task, we demonstrate a network with an accuracy-robustness area (ARA) of 0.0053, an ARA 2.4 times greater than the previous state-of-the-art value. Improving the mechanisms by which NN decisions are understood is an important direction for both establishing trust in sensitive domains and learning more about the stimuli to which NNs respond.

Fields of industry wishing to harness the explosion of machine learning techniques are concerned about the lack of accountability and explainability within the field[1,2]. Biomedical papers report systems that surpass human experts, but have difficulty proving the added insight of their techniques beyond statistical correlations[1,3]. The responses generated from Machine learning systems are also very sensitive to changes in the input[4]. These concerns about explainability and stability apply to a variety of machine learning algorithms[5], but we focus on the subfield of neural networks (NNs).

State-of-the-art methods attempting to explain the reasoning behind NN decisions focus on the generation of heatmaps that indicate regions of input salient to the NN's output[6–20]; for example, gradient-weighted class activation mapping (Grad-CAM)[6] in Fig. 1. However, these heatmaps do not communicate information beyond a rough silhouette, making it difficult to infer specific qualities within the general region being considered. These methods also rely on the linearization of a highly non-linear network, and capture relevant details only for the exact, corresponding input[21]. Some minor perturbations, called adversarial attacks or adversarial examples, can abitrarily change the NN's output[4,22–28]. Worse, the input perturbations from adversarial attacks do not align with the heatmaps generated by state-of-the-art explanation techniques (demonstrated in Supplementary Section 1). That is, without robustness to adversarial attacks, attempts to explain an NN's decision have limited validity.

We contribute a set of novel techniques that allow for adversarial explanations (AEs) to illustrate key features salient for classification, a much more reliable method of explaining an NN's decision. The AE process is illustrated in Fig. 2. Unlike previous state-of-the-art techniques, AEs work with network nonlinearities to represent the NN's decision surface with greater fidelity than heatmaps can provide. The techniques that afford high-quality AEs centre around minimizing the Lipschitz constraint of the end-to-end model, a variation of Lipschitz regularization that has not been previously explored[29–31]. The new regularization paired well with existing robustness techniques, most notably adversarial training[27,32]. In addition to producing visually rich explanations, our techniques create robust NNs that surpass the state of the art in terms of classification performance in the presence of adversarial examples: we demonstrate classification networks for the ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC 2012) challenge[33] with improved accuracy-robustness area (ARA) 2.4 times greater than the state of the art[25,27,32,34–41], as shown in Fig. 3. Together, the methodology outlined in this work demonstrates the viability of producing cogent explanations via adversarial attacks on robust networks.

Interpreting AEs

A comparison of the Grad-CAM method of explaining an NN and our AEs is shown in Fig. 1. This figure was produced using our CIFAR-10[42] network with the highest attack ARA, a metric that captures a machine learning method's robustness to adversarial attacks, as defined in the Methods. The left half of Fig. 1a demonstrates four different input images, and the corresponding NN predictions for the most confident class and either the second-most-confident class, or the true class (if it was not the most confident prediction). Next to the input image are Grad-CAM1 and Grad-CAM2, containing the Grad-CAM explanations for the two displayed class predictions. We note that even for very disparate classes, such as cat and truck in the first row, the Grad-CAM explanations are mostly the same, and do little to indicate the textures or shapes that influenced the decision. Following the Grad-CAM explanations, in the right half of Fig. 1a, are two AEs, representing positive explanations for each of the two class predictions displayed by the original input image. Each AE shows the new top-two network predictions, an image of the differences between the original input and the adversarial image, and the adversarial image itself. Below are Fig. 1b, which detail each of the AEs.

Department of Electrical and Computer Engineering, Portland State University, Portland, OR, USA. *e-mail: wwoods@pdx.edu; chenjac@pdx.edu; teuscher@pdx.edu

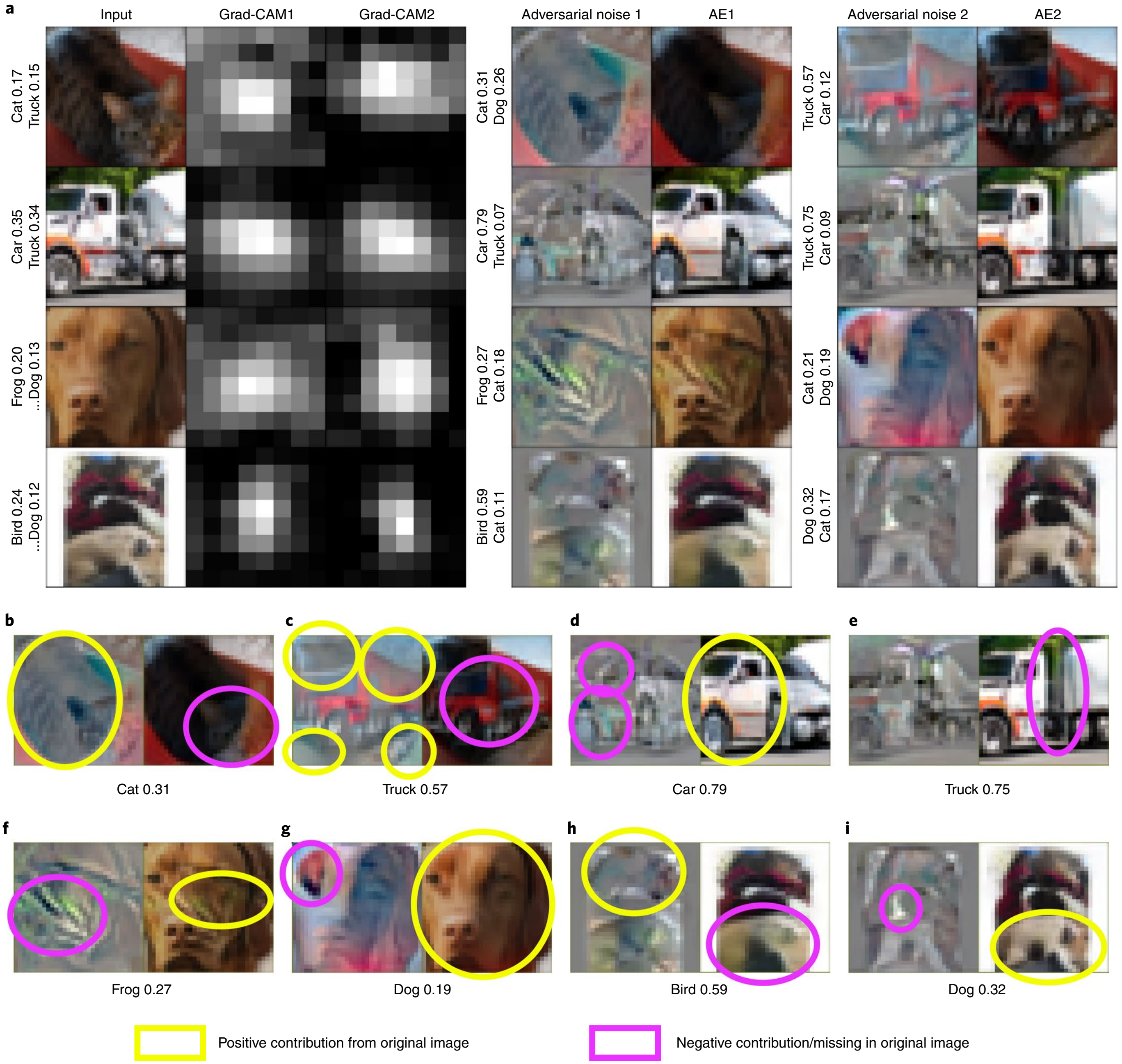

Fig. 1 | Comparing explanatory power between Grad-CAM and AEs when applied to a robust NN trained on CIFAR-10. a, Explanations compared across different inputs. For each row, the columns show: the original 'Input' image, labelled with the correct class and the most confidently predicted incorrect class, sorted by the NN's confidence in each, where ellipses denote that the correct class was not in the top two predictions; two Grad-CAM explanations, one for each predicted class shown by the input; two AEs, divided into the adversarial noise used to produce the AE, and the AE itself. **b–i**, Annotated versions of the AEs for **a**, indicating regions that contributed to, or detracted from, each predicted class. All photographs in this article were taken by the authors or those mentioned in the Acknowledgements.

Figure 1b,c demonstrates relevant conclusions that may be drawn from the AEs in row 1 of Fig. 1a. The network correctly classified this image as cat, but from the difference image in Fig. 1b, it can be seen that the cat class confidence would have been even higher with a blacker body and without the cat's face. The body was annotated as a positive contribution as, while the adversarial image changed the body, it kept the overall structure of that region, and increased its contrast. On the other hand, the cat's face is almost entirely removed from the adversarial image, indicating that it contributed against the cat classification. This indicates that the NN did not possess the logic needed to recognize a face in that configuration as belonging to a cat, perhaps because the cat's face is too small a feature in the image. In Fig. 1c, the explanation for the truck prediction illustrates that the framing of the central cat mimics the framing of many truck photos in the training data. That is, the shapes of the corners of the image were well preserved, with the high-contrast, upper-right corner being similar to the division between a trailer and the sky. The truck, which was added as part of the explanation, was missing in the original image, and was thus annotated as a counter-indicator. Note, however, that the magnitude of the perturbation between

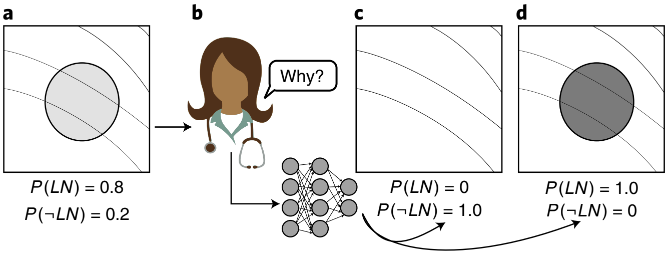

Fig. 2 | Illustration showing how AEs might improve trust in a medical NN's decision. While adversarial examples are considered a nuisance by most, they have the potential to provide reliable explanations with the same richness of information as the original input. **a**, For example, consider an NN trained in finding lung nodules in radiographs, which outputs $P(LN)$, the probability of the image containing a lung nodule. **b**–**d**, When this NN needs investigation (**b**), an attack may be targeted at a desired new network output—such as changing a nodule classification to a non-nodule classification (**c**) or emphasizing the nodule (**d**)—to produce a new image that is minimally changed but produces the desired output. By comparing these inputs and looking at the differences, a human operator can identify relevant features in the input with greater fidelity than previous methods of explanation.

the original image and either of Fig. 1b,c is the same—while the truck is more readily noticed due to its detail, the cat's body was preferentially much darker for a more confident cat classification. Note also that the final class confidences for these AEs are 0.31 and 0.57, indicating that in L_2 space, the input image is much closer to a truck manifold than a cat manifold. With AEs, we gain information about the network's function not only through the input features that would be need to be modified, but also through the resulting class confidences.

Figure 1d,e annotates the AEs from row 2 of Fig. 1a. In Fig. 1d the AE for car relaxes the slope of the pillar against the windshield, and removes much of the colouring around the wheel well. Neither of those features would often be found in cars, although they were present in the original input. With these modifications in place, the shape of the vehicle's front matches that of a car, and it becomes unclear whether or not the trailer is in the foreground. The truck AE (Fig. 1e) indicates that the main reason this input was not identified as a truck was the missing gap between the tractor and the trailer. With that feature in place, the NN would confidently classify the image as truck.

Figure 1f,g shows the AEs corresponding to row 3 of Fig. 1a. The reasoning behind the network's final guess of frog is hard to see at first, but two major factors clearly contributed. First, in Fig. 1f the frog-skin shading already existed on the right side of the face. While the AE exaggerated this shading, it was clearly already present. Second, in Fig. 1g almost the entire image was turned redder and higher contrast to inspire a dog prediction. Looking at the final confidences, with a maximum of 0.27 even with large perturbations, this image was probably distant from the original training data's manifold, and possessed just enough of the frog-skin shading on the face to convince the network that the frog class was most applicable.

The AEs corresponding to row 4 of Fig. 1a are shown in Fig. 1h,i. The top half of the image in Fig. 1h was similar to a bird face when rounded out a bit. The actual dog pixels in the bottom half of the image were smoothed in this AE, indicating that they were counter-indicators of the bird class. In Fig. 1i, one key feature prevented a dog classification. If the white piece of clothing in the middle left of

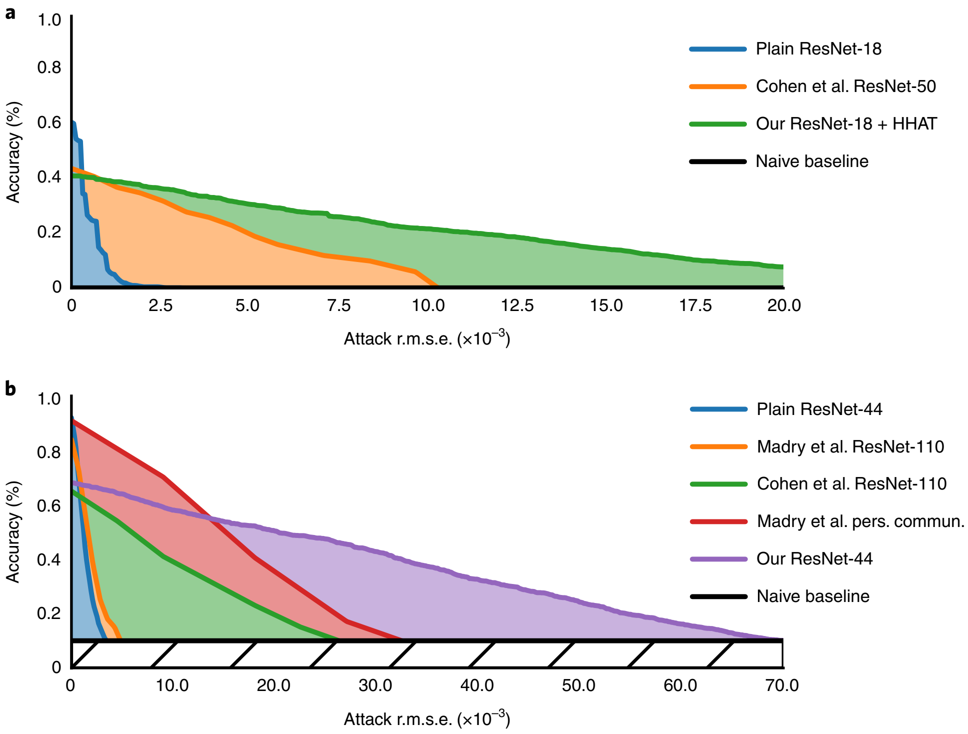

Fig. 3 | Comparison of different state-of-the-art, robust NNs. a,**b**, Accuracy falls as permissible noise magnitude is increased on ILSVRC 2012 (**a**) and CIFAR-10 (**b**) datasets. Compared with the state of the art, our method achieved NNs that were tolerant of greater adversarial perturbations on both ImageNet and CIFAR-10. By measuring the ARA, a model's resistance to adversarial attacks is taken into consideration alongside its ability to make predictions better than the naive baseline (the hatched region). See Methods for additional details. Madry et al. used a network that was ten times wider than a traditional ResNet-110, and also trained against an $L_{\inf}$ adversary rather than an L_2 adversary[32]. Cohen et al. used a standard ResNet-110[36]. The red curve came from personal communication with A. Madry regarding a standard ResNet-50.

the original image swept further down, then the centre of the image would have looked more dog-like, with the resulting black bubble forming a nose. It is also clear that a slightly greater contrast within the dog's pixels would also have helped.

Altogether, AEs show more information about the NN's operation than previous state-of-the-art techniques like Grad-CAM. An ILSVRC 2012 version of Fig. 1 is included as Supplementary Fig. 5. Further comparisons with previous state-of-the-art explanation methods may be found in Supplementary Methods 1.

AE generation and evaluation. Adversarial attacks were conducted with two separate goals: adversarial attacks aimed at reducing the classification accuracy of a network, and adversarial attacks aimed at producing classification explanations.

Untargeted adversarial attack generation for the evaluation of models' ARA metrics followed Algorithm 1; this was a variant of Carlini et al.[34] and also leveraged normalizing gradient steps by their magnitude, as first proposed by Rony et al.[43] in the context of adversarial attacks. We found that alternating between target loss maximization and L_2 error minimization, rather than pursuing both simultaneously, allowed for better automatic balancing between the two errors, resulting in smaller perturbation magnitudes. In contrast to the attacks presented by Carlini et al.[34], the algorithm presented will not begin a magnitude refinement before the target classification error is reached. The threshold at which Algorithm 1 switches between minimizing the correct class's post-softmax prediction s_t and minimizing the attack magnitude is defined by $g(s,t)$.

Algorithm 1 Process used to generate adversarial examples.

Input: N, the number of attack-optimizing steps; $f(\cdot)$, the NN; x, the network input; t, the true class of the input; $O(\cdot)$, an optimizing method such as SGD with momentum; $g(s, t)$, a goal function returning true if the network outputs from the attack are suitably different from the true class t; η, a balancing term between categorical loss and MSE loss.

Output: δ_{best}, the adversarial noise which satisfies the goal $g(\cdot)$ and has minimal vector length.

```
begin
    $\vec{0}$
    $M_{best} \leftarrow$ inf
    for $n \in [0,\ldots,N-1]$ do
        $\hat{x} \leftarrow c(x+\delta)$ // $c(\cdot)$ clips elements of its argument to a valid input range, e.g. [0,1]
        $y \leftarrow f(\hat{x})$
        s $\leftarrow softmax(y)$
        if $g(s, t)$ then
            $\Delta\delta \leftarrow 2\delta$ // $L_2$ loss for magnitude
            if $||\delta||_2 < M_{best}$ then
                $M_{best} \leftarrow ||\delta||_2$
                $\delta_{best} \leftarrow \delta$
        else
            $\Delta\delta \leftarrow \partial s_t/\partial(x+\delta)$
            $\Delta\delta \leftarrow \eta\frac{\Delta\delta}{||\Delta\delta||_2}$ // Fixed gradient magnitude
        $\delta \leftarrow o(\delta, \Delta\delta)$ // Apply optimizer step
```

We present two choices of $g(s,t)$ for the current work. The first, $g_{\text{adv}}(s,t)$, was the well-known adversarial attack metric used by all previous work in this field[27,36], and denotes the boundary at which top-1 accuracy would decrease:

$$g_{\text{adv}}(s,t) = \begin{cases} 1 & \text{if } s_t - \max_j(s_j; j{\neq}t) < 0, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

This was the $g(s,t)$ used to produce Fig. 3. When ARA values are reported for a model, we evaluated random validation or testing images until we had 1,000 that were correctly classified. We then made a list of the root mean squared error (r.m.s.e.) below which each image would retain the correct classification, minimized as per Algorithm 1. This list was extended with zeroes for each image evaluated that was initially incorrectly classified: if a model scored a 70% classification accuracy on unmodified images, we would have a final r.m.s.e. list of about 1,429 in length, 1,000 of which were nonzero. This list was then evaluated for accuracy at different levels of the r.m.s.e., as seen in Fig. 3, and the area above the naive baseline was taken to produce the attack ARA metric. Through this process, the ARA measures a combination of the classifier's predictive power and its ability to overcome an adversary; a larger value is desirable. When two NNs have identical clean accuracies, a network with an ARA 3× larger than another network's ARA retains predictive power against adversaries that produce 3× more noise than when second network would fail.

In the context of Algorithm 1, we used $N = 450$, $o(\cdot)$ was a stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and momentum 0.9, and $\eta = 0.55$. Examples of this attack can be seen in Fig. 4.

In the context of explanations, however, we found the $g_{\text{adv}}(s,t)$ metric to be lacking. The decision boundary was not always sufficiently distant from the data point to reveal salient features. Instead, we targeted an amount of perceptual difference between the explanation and the original input, optimizing the shape of the perturbation for that which would maximally impact the network's output in a desired manner. Comparing these explanations with the original input then demonstrates precisely which features would lead to a desired output. This was accomplished by following $\partial s_t/\partial(x+\delta)$ up to a boundary r.m.s.e., at which point the r.m.s.e. would be

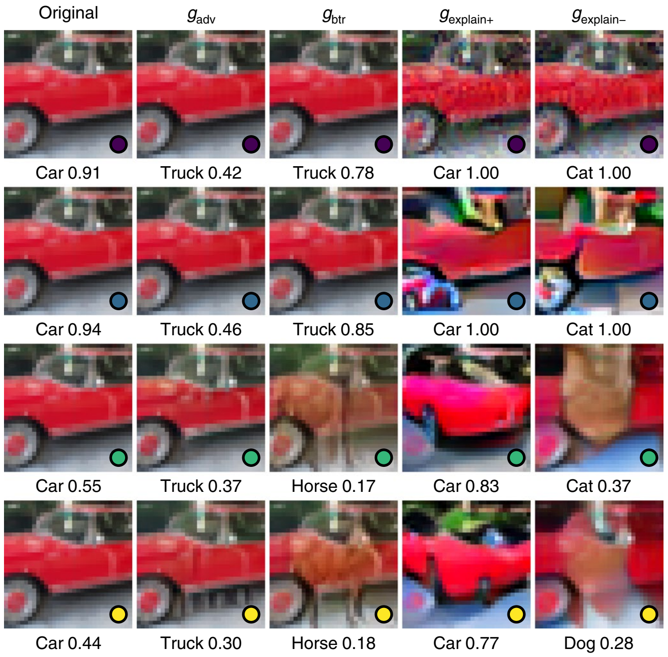

Fig. 4 | Demonstration of different AEs for a car classification problem, as computed on four different NNs trained on the CIFAR-10 dataset. Columns show the input (left) and AEs using different $g(\cdot)$ functions: g_{adv} is a traditional adversarial example, g_{BTR} is an adversarial example which reduced the confidence in the true class to 10%, $g_{\text{explain+}}$ is an AE which encouraged the car classification, and $g_{\text{explain-}}$ is an AE which encouraged the cat classification instead. The rows correspond to a traditional, non-robust NN (purple dots), an NN trained with adversarial training (blue dots) an NN trained with both adversarial training and equation (1) (green dots) and an NN trained with only equation (1) (yellow dots) for robustness. The corresponding experiments are listed in Table 1.

Table 1 | Selected experiment results on CIFAR-10

Description	Accuracy	Attack ARA	BTR ARA
Traditional ResNet-44[a]	92.2	0.0013	0.0014
Regularization strength			
$\psi = 0.55$, $K = 1$	92.5	0.0022	0.0026
$\psi = 4.0$	92.4	0.0039	0.0048
$\psi = 30$	90.2	0.0059	0.0080
$\psi = 220$	84.5	0.0083	0.0135
Regularization stochasticity			
$L_{2,\text{adv}}$, $\psi = 220$, $K = 1$	84.5	0.0083	0.0135
$K = 2$	85.0	0.0084	0.0134
$K = 4$	85.0	0.0084	0.0135
$K = 8$	84.8	0.0082	0.0133
HHReLU			
$\psi = 12{,}000$, normal ReLU	56.5	0.0110	0.0347
$\psi = 12{,}000$, HHReLU	78.0	0.0125	0.0261
No HHReLU, but $\psi = 220$ for similar accuracy	77.1	0.0102	0.0194
Combination with adversarial training			
Method from Madry et al.[32], best parameters[b]	87.4	0.0107	0.0153
Equation (4) and adversarial training[c]	68.4	0.0197	0.0450
Equation (4) only, best parameters[d]	68.7	0.0151	0.0423

The effects of different robustness procedures on the overall classification accuracy and adversarial resistance of an NN classifying CIFAR-10 images are presented here. Accuracy refers to the accuracy on CIFAR-10 test data after the final epoch. Attack ARA and BTR ARA both represent the NN's performance against adversaries of varying strength. [a]First row of Fig. 4. [b]Second row of Fig. 4. [c]Third row of Fig. 4. [d]Fourth row of Fig. 4.

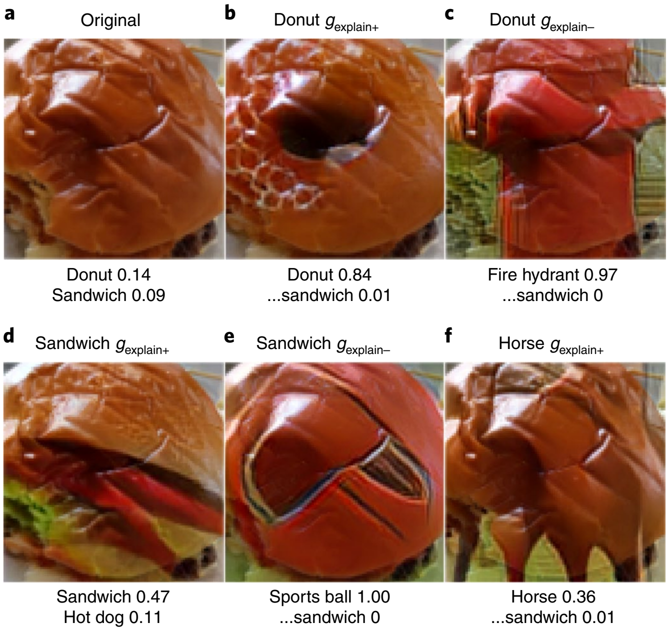

Fig. 5 | Different explanation techniques using $\rho = 0.075$ with an NN trained on the COCO dataset. a, The original image. **b**, A positive explanation for the donut class; we note alignment of the added 'hole' with a wrinkle in the original sandwich bun. **c**, A negative explanation for the donut class resulted in the removal of the round shape of the sandwich. **d**, A positive explanation for the true class, sandwich, results in exposed contents (peppers or tomatoes) and the beginnings of lettuce. **e**, A negative explanation for the sandwich class reveals homogenization of the bun's texture, and further rounding out of the overall shape. **f**, Positive explanation for a completely unrelated class, horse: legs were clearly added, and the textured area in the upper-left of the image is appropriated as a saddle.

minimized, a tick-tock method similar to Algorithm 1, but substituting a slightly different boundary criterion:

$$g_{\text{explain}+}(\delta;\rho) = \begin{cases} 1 & \text{if } ||\delta|| > \rho, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We note that $g_{\text{explain}-}$ is also possible, by modifying Algorithm 1 to maximize the selected class loss rather than minimizing it. These techniques are demonstrated in Fig. 5.

The attack ARA was not found to be quantitatively indicative of the quality of AEs. For example, consider two closely related classes from CIFAR-10: automobile and truck. These classes are often mistaken for one another, leading to a decrease in the magnitude of untargeted attacks for members of either class. With respect to the network's ability to tell these two apart, g_{adv} remains a good metric. However, as a classifier learns to distinguish these related classes from the other unrelated classes, the s_j value corresponding to these related classes may rise in tandem. The described phenomenon is illustrated in Fig. 6. This situation would indicate that the network possesses a greater capacity for deciding what is automobile or truck compared with the remaining classes, but the attack magnitude would not decrease as these two classes would still be easily confused. Since the confusion between these two classes is built into the problem, g_{adv} hits a ceiling beyond which an attack magnitude based on the g_{adv} metric cannot be improved. As such, we also considered better than random (BTR) as a measure of the classifier's knowledge of class-specific features. The BTR magnitudes were defined on the basis of the distance between the classifier's prediction and a prediction at which the true label's valuation matches that of a random classifier. As shown in Fig. 6, the BTR quantity continues to increase even as related classes both are more confidently predicted. Thus, g_{BTR} (where V is the number of classes in the prediction) is defined as:

$$g_{\text{BTR}}(s,t) = \begin{cases} 1 & \text{if } s_t < \frac{1}{V}, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We note that the existence of NN output meeting the BTR constraint is very likely, as resetting all pre-softmax outputs to 0 achieves the required condition. The BTR ARA gracefully degrades into that attack ARA on binary classification problems. Note also that we deliberately chose a truly random classifier, and not a naive classifier, for unbalanced datasets (such as the COCO dataset). When calculating BTR ARA metrics from a population of adversarial examples created using g_{BTR}, a naive classifier was still used as the baseline for the area calculation.

For a traditional ImageNet ResNet-18, we measured an attack ARA of 0.0004 and a BTR ARA of 0.0013. For a CIFAR-10 ResNet-44, we measured an attack ARA of 0.0013 and a BTR ARA of 0.0014. An intuitive sense of attack ARAs may be gathered by perusing the Supplementary Figures.

Lipschitz minimization for adversarial robustness

The quality of an AE was found to be related to an NN's robustness to adversarial attacks. This is illustrated in Fig. 4: the first row demonstrates our AE algorithm applied to a non-robust NN, and

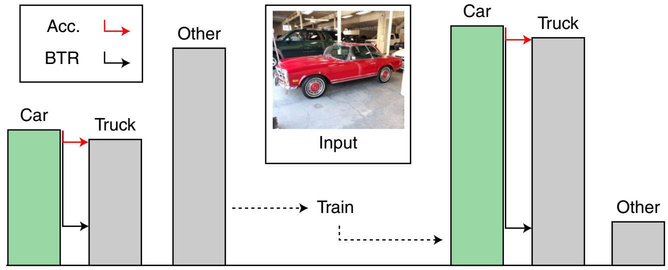

Fig. 6 | Demonstration of the limited utility of attack ARA when considering if an NN has learned salient features. The previously mentioned attack ARA metric depends on the relative confidence between the correct class and the second-most-confident class. For related classes such as car and truck, the distance between these two classes may not increase through training. However, by measuring the distance from the correct class (green bar) back to a fixed baseline, as is done with the BTR ARA metric, improvements in feature recognition may be measured regardless of the presence of related classes.

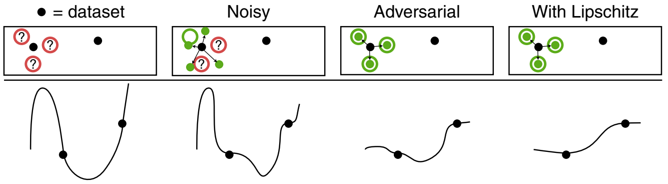

Fig. 7 | Illustration of the benefits of noisy training, even with a Lipschitz regularization. Black dots demonstrate values in the training dataset, other areas are valid inputs not represented in the training dataset. Top panels: a two-dimensional representation showing that, without noisy or adversarial training, valid input values not in the training dataset might perform poorly without any corrective force. Bottom panels: a one-dimensional example demonstrating that the over-parametrization of NNs leads to undefined behavior between training data points. Left: training an NN on samples from a dataset (left, black dots) specifies desired behaviour at the data points, but does not describe behaviour between those data points, allowing the decision surface to take an arbitrary shape. Middle left: adding some noise can improve the behaviour between dataset samples, but is statistically unlikely to improve the worst behaviours. Middle right: adversarial training deliberately attempts to improve the worst performing points, leading to a smoother decision surface. Right: adding a stochastic Lipschitz loss, as equation (4), further smooths behaviour between data points.

each AE panel reveals no intelligible features when compared with the original input.

Briefly, Lipschitz continuity is the bounding of a function's value, such that the function's value is not allowed to change between two points more than a constant value times the distance between those points. In the machine learning literature, this has previously been approximated as a bounding of derivatives within each individual layer[29–31], rather than minimizing the derivatives across the whole network.

Instead, we developed a new Lipschitz regularization term that aggressively minimizes the end-to-end rate of change between a classification NN's pre-softmax outputs and its inputs. For efficiency, this regularization term was made stochastic. For an NN with an input vector $\mathbf{x}$ of dimensionality N and an output vector $\mathbf{y}$ of dimensionality V, we chose K random outputs to regularize each training step through the loss function $L_{\mathrm{adv},z}$ with strength ψ:

$$\begin{aligned} v_k &\sim [1, 2, \ldots, V] \\ L_{\mathrm{adv,z}} &= \frac{\psi}{KN}\sum_{k=1}^{K}\sum_{j=1}^{N}\left|\frac{\partial y_{v_k}}{\partial x_j}\right|^z \end{aligned} \tag{4}$$

Experiments were performed to determine the effects of the parameters in equation (4) and other factors that affect robustness using the CIFAR-10 dataset. A selection of these experiments is detailed below. The full listing of experiments with the CIFAR-10, ILSVRC 2012, Japanese Society of Radiological Technology (JSRT) lung nodule[44], and common objects in context (COCO)[45] datasets, as well as NN architectures used and other experiment details, can be found in Supplementary Methods 1–3.

Table 1 shows the results from several experiment groups. When analysing the efficacy of the equation (4) regularization across the different experiments shown in Table 1, several quantities were considered. The accuracy shown is the NN's accuracy on clean data, in the absence of an adversary. The attack ARA shown is the area between a naive classifier and the NN's accuracy at different levels of perturbation, illustrated in Fig. 3. The BTR ARA is calculated similarly to the attack ARA, but with the network considered 'accurate' when the NN's confidence in the true class was greater than a random chance confidence.

Regularization strength

We sought to verify that increasing the strength, ψ, of our proposed regularization would lead to an increase in adversarial resistance. Table 1 demonstrates that this was the case, with the classifier's attack and BTR ARAs increasing monotonically with the strength of the regularization. It is also notable that, up to a certain level ($\psi = 4.0$), we were able to maintain the classifier's accuracy on clean data while gaining adversarial resistance. After that, clean accuracy decreased as adversarial resistance increased. Therefore, ψ can be varied in accordance with which is more desirable: accuracy or adversarial resistance.

Regularization stochasticity

The stochastic formulation of equation (4) was expected to yield the same results as a non-stochastic formulation. To check the validity of this assumption, we tried different values of K from 1 to 8. These experiments demonstrated that varying K had little effect, validating the resource-efficient choice of $K = 1$. As these experiments were identical, they demonstrated that our method had an accuracy standard deviation of ±0.2%, and ARA metric standard deviation of ±0.0001.

Half-Huber rectified linear unit

The experiments before this point used a traditional rectified linear unit (ReLU) activation function. A ReLU is continuous in value, but its derivative is discontinuous. Regularizing as per equation (4) requires the the derivative of the activation functions used by the network to be optimized, and as such we desired the first derivative to be continuous. We devised a new activation function, the half-Huber ReLU (HHReLU), which was a rectified version of the Huber function and met our needs. Table 1 shows that, for the same level of accuracy, an NN using the HHReLU had an attack ARA about 25% greater and a BTR ARA 35% greater than an NN using a ReLU.

Combination with adversarial training

We evaluated adversarial training on its own using the ARA metrics, as shown in Table 1. The parameters chosen for adversarial training matched personal communication with A. Madry[32] regarding their best results. We note that the attack and BTR ARAs were both lower for an NN with adversarial training than for our HHReLU NN, although accuracy on clean data was higher for adversarial training.

Adversarial training was then combined with equation (4), yielding our best results on CIFAR-10. Both the attack ARA and BTR ARA were greatly improved when compared with adversarial training alone, and the attack ARA was improved when compared with equation (4) alone. Regarding the accuracy difference between adversarial training and a combined approach, we note that the accuracy of the combined NN is higher than that of the adversarially trained network at an r.m.s.e. of 0.013, an imperceptible perturbation; see Supplementary Figs. 2, 3 and Supplementary Methods 1.1 for more information.

A comparison experiment with only equation (4) is also shown in Table 1. We note that while the BTR ARA is similar, the attack ARA is lower without adversarial training. We posit that adversarial training helps to stabilize the direction of steepest ascent for the loss function, while our proposed regularization stabilizes the entire loss surface, as illustrated in Fig. 7. The definitions of the two techniques provide this distinction, and the empirical evidence seems to support it.

Figure 4 demonstrates AEs applied to the NNs from this experiment group.

Discussion

We demonstrated a regularization technique based on the Lipschitz constraint that enhanced the ability of networks to resist adversarial examples. This was paired with other innovations, including an HHReLU and an improved adversarial training methodology. On ILSVRC 2012, the methods in this work increased the ARA by 2.4× over the previous state of the art, while retaining the same level of accuracy on clean data and using a network one-third of the size of the previous state of the art. More central to the tenets of this work, we demonstrated that the stability added by these techniques allows for adversarial examples to be generated with very discernible features. These adversarial examples could then be used as nonlinear explanation mechanisms, termed AEs, working with the network and its nonlinearities to produce more reliable explanations than previous work. In Supplementary Methods 2 and 3, we also demonstrate that AEs might be annotated as part of an active learning pipeline to yield improved adversarial resistance. We hope that this work provides a basis for future efforts with both adversarial resistance and explainable machine learning, making algorithms more reliable for industry fields where accountability matters, such as biomedical applications or autonomous vehicles.

Methods

Defense via Lipschitz continuity. An integral part of many white-box attacks, including Algorithm 1, involves following the gradient of some loss. The rate at which the output of the network might be changed is likewise dependent on that gradient. To see how this might affect classification networks, consider the softmax operation, here denoted as $s(\cdot)$, applied to the output of an NN, y:

$$s(y) = \frac{e^{y}}{\sum_{i=1}^{V} e^{y_i}} \tag{5}$$

There are 1,000 classes in the ILSVRC 2012 challenge[33]. Assuming 999 of those classes have an NN output of $y_i = 0$, then a value for the remaining class of $y_j = 10$ corresponds to a confidence in class j of 95.7%. For a confidence of 4.3%, that value need only fall to $y_j = 3.8$. In reality, an adversarial attack also has the option of increasing the confidence of classes $i \neq j$ to reduce confidence of class j. If $y_j = 3.8$, and another $y_k = 6.2$, $k \neq j$, then the confidence of class j falls to 2.9% and class k rises to 32.1%. In other words, instability on the output values will quickly overwhelm the softmax operation. If we assume locally linear behaviour of the network, this instability may be modelled by looking at the expected change in the network's output given some gradient information. Using $E_i[\cdot]$ to denote an expectation conditioned on i, N as the number of input elements, Δ to signify an actual value change and ∂ to signify a variable's partial:

$$E_i[|\Delta y_i|] \approx E_i\left[\left|\sum_{j=1}^{N} \Delta x_j \frac{\partial y_i}{\partial x_j}\right|\right] \tag{6}$$

$$\lesssim \sum_{j=1}^{N} E_{i,j}\left[\left|\Delta x_j \frac{\partial y_i}{\partial x_j}\right|\right] \tag{7}$$

These quantities are neither independent nor equivalent, but we will simplify them as such:

$$E[|\Delta y_i|] \lesssim NE\left[|\Delta x_j|\right]E\left[\left|\frac{\partial y_i}{\partial x_j}\right|\right] \tag{8}$$

Equation (8) provides a loose guideline for targeting different values of $|\partial y_i/\partial_x|$. In fact, as a network becomes more non-linear, equation (8) becomes less accurate.

To see how effective the guideline given by equation (8) was in practice, we built a ResNet-18 and trained it on ILSVRC 2012 training data, detailed in Supplementary Methods 2. Leveraging PyTorch's automated differential engine, we collected gradients for one of the NN's outputs, before the softmax, with respect to each of the 15,0528 input elements (224·224·3). The mean absolute value of the computed derivatives then resulted in an aggregate number that summarized the network's volatility in the original input space. For our ResNet-18, this value worked out to 0.051/input. The mean of the maximum absolute derivative per image was a much larger 3.9/input, indicating a large spread in these values. Attacks were generated against this network with a 50% confidence margin in favour of an adversarial class. Again, based on a local-linearity assumption, the magnitudes of these attacks were measured as the mean absolute difference per pixel between the original and attacked images. The harmonic mean of the mean absolute distances of all such attacks against this network was found to be 0.0033/input; according to equation (8), the sum of Δy_i between the true and adversarially targeted classes should then be less than 50.7. The actual measured sum of Δy_i across the true and target classes averaged 26.1.

The change in network output was shown in equation (8) to be bounded proportionally to the gradient of the output with respect to each input element, as long as local network behaviour was linear. Since this assumption seemed to hold for real networks, we theorized that limiting this gradient would therefore provide some adversarial resistance in these linear regions of the network by forcing additional nonlinearities to compensate for the limitation. This is a form of Lipschitz continuity. From another point of view, limiting $|\partial y_i/\partial x_j|$ makes each training element a stable point for the network, enforcing a neighbourhood of validity for each decision. The classification loss then enforces necessary nonlinearities between these stable regions. As such, one of this work's contributions is to explore the relation between limiting $E[|\partial y_i/\partial x_j|]$ and adversarial attacks. We note that, particularly with the ReLU activation function, even a gradient of **0** does not guarantee a neighbourhood of validity, hinting at the utility of combining adversarial training with the proposed regularization equation (4).

For networks with 1,000 outputs, minimizing $|\partial y_i/\partial x_j|$ directly for all i is computationally prohibitive—each training image processed would require 1,000 additional gradient propagations. Instead, we use a regularizing loss that is stochastically defined with a scaling parameter ψ and a power factor z, in equation (4).

Equation (4) therefore draws K random indices (without replacement) from the available output nodes and minimizes the derivative of each selected output with respect to all inputs. Backpropagation makes this an efficient computation regardless of the number of input elements. When $K = V$, $L_{\mathrm{adv},z}$ ceases to be stochastic. K and N are both included in the denominator such that the expected force per image relative to the classification loss is maintained regardless of the number of inputs or outputs.

It is also possible for n_k to be drawn from a non-uniform distribution. To test the merits of this, we considered distributions that yielded the correct label ζ% of the time and were pulled from a random distribution (including the correct label) the rest of the time. Another variant of non-uniform distribution involved substituting the minimization of the true class's gradient ζ% of the time for minimizing the gradient $(\partial y_t - \max_{i \neq t} \partial y_i)/\partial x_j$, the difference between the true class and the maximum non-true class prediction. This regularization, which we label $L_{\mathrm{adv,tandem}}$ because it aligns the gradients of two different classes in tandem, was chosen on the basis of the automobile versus truck discussion. While regularizing only one class at a time guarantees that the gradient for that class will approach zero, this provides an opportunity for a related class to dominate. As the softmax operation assigns probabilities on the basis of the difference between elements of its input, it was determined that it might be more effective to regularize the difference between those inputs (the NN's output). This $L_{\mathrm{adv,tandem}}$ technique was used for our NNs that yielded the best attack and BTR ARAs.

Further discussion of our testing methodology for this regularization, including our adherence to best practices for new defences in accordance to recent recommendations[46], may be found in Supplementary Methods 1. Investigations of different values of z in equation (4), non-uniform sampling, $L_{\mathrm{adv,tandem}}$, and additional techniques to enhance the regularization effect may be found in Supplementary Methods 3. These experiments were conducted to show that the proposed regularization technique is in fact a rich family of techniques based on approximations of which quantities are relevant for adversarial defence.

Adversarial and noisy training. Even with gradients of **0** at all points in the training data, an activation function such as ReLU guarantees no neighbourhood for which the gradient will remain **0**. That is, a loss 'cliff' might be arbitrarily close to any training point. This is partly due to the overparameterization of NNs, illustrated in Fig. 7. As such, we investigated combining our method with either random Gaussian noise or adversarial training.

Adversarial training was implemented two ways. In the first, denoted L_2, an L_2 distance ε was chosen and the adversary attempted to find the highest loss value within that ε ball. The gradient of the classification cross-entropy loss was followed for seven steps, each time being normalized to $\varepsilon/7$ magnitude. This was very similar to the original adversarial training approach proposed by Madry et al.[32]. In the second, minimal adversarial training denoted $L_{2,\min}$, an L_2 distance ε was also chosen, but before taking each step, the network's classification was evaluated. If the network correctly classified the example, then the gradient of classification loss was normalized to length $\psi_{7,n}\varepsilon$ and followed, where $\psi_{k,n} = 2(k-n)/[k(k+1)]$ and $n \in [0, 1, ..., k-1]$ is the index of the step being taken. In this formulation, the step sizes at subsequent steps yield progressively finer movements. If the network incorrectly classified the example, then the gradient was replaced with the negation of the current perturbation, normalized to size $\psi_{7,n}\varepsilon$ and followed. That is, the $L_{2,\min}$ method of adversarial training sought to train on adversarial examples near the boundary at which the network would misclassify those examples.

Training configurations where batches were composed of half adversarial examples and half original examples from the dataset were considered. In Tsipras et al.[27], this technique was called half-half training, and we keep that nomenclature. We found half-half training to be more effective when using $L_{2,\min}$ adversarial training.

NN architecture. All of the CIFAR-10 results in this work were based on a ResNet-44[47], modified to be in pre-activation form[48] with each residual block's output convolution weights initialized to zero as per ref. [49]. ILSVRC 2012 networks were similar, but based on ResNet-18[47] with the same modifications as the CIFAR-10 network. All NNs were implemented in PyTorch[50]. Further training information can be found in Supplementary Methods 2.

Other NN regularizations, such as Shake-Drop[51] and Stochastic Depth[52], were tried and found to not enhance the regularization provided by equation (4).

Additional methods and results. The Supplementary Methods contain detailed additional materials regarding related work, experiments on other datasets and analysis of the concepts discussed throughout this work. The Supplementary Figures comprise many visual comparisons of the concepts discussed in this work.

Data availability

All data used in this work, including the CIFAR-10[42], ILSVRC 2012[33], JSRT[44], and the COCO[45] datasets are freely available.

Code availability

A reference implementation of the techniques presented throughout this work, applied to the CIFAR-10 dataset, can be found at https://github.com/wwoods/adversarial-explanations-cifar.

Received: 11 May 2019; Accepted: 23 September 2019;
Published online: 04 November 2019

References

1. Finlayson, S. G. et al. Adversarial attacks on medical machine learning. *Science* **363**, 1287–1289 (2019).
2. Stilgoe, J. Machine learning, social learning and the governance of self-driving cars. *Soc. Stud. Sci.* **48**, 25–56 (2018).
3. Tsao, H.-Y., Chan, P.-Y. & Su, E. C.-Y. Predicting diabetic retinopathy and identifying interpretable biomedical features using machine learning algorithms. *BMC Bioinform.* **19**, 283 (2018).
4. Szegedy, C. et al. Intriguing properties of neural networks. Preprint at https://arxiv.org/abs/1312.6199 (2013).
5. Papernot, N., McDaniel, P. & Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. Preprint at https://arxiv.org/abs/1605.07277 (2016).
6. Selvaraju, R. R. et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* 618–626 (IEEE, 2017).
7. Ribeiro, M. T., Singh, S. & Guestrin, C. Why should I trust you? Explaining the predictions of any classifier. In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1135–1144 (ACM, 2016).
8. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: visualising image classification models and saliency maps. Preprint at https://arxiv.org/abs/1312.6034 (2013).
9. Landecker, W. *Interpretable Machine Learning and Sparse Coding for Computer Vision*. PhD thesis, Portland State Univ. (2014).
10. Bau, D., Zhou, B., Khosla, A., Oliva, A. & Torralba, A. Network dissection: quantifying interpretability of deep visual representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 6541–6549 (IEEE, 2017).
11. Bau, D. et al. Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations* (ICLR, 2019).
12. Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R. & Yu, B. Interpretable machine learning: definitions, methods, and applications. Preprint at https://arxiv.org/abs/1901.04592 (2019).
13. Hong, S., You, T., Kwak, S. & Han, B. Online tracking by learning discriminative saliency map with convolutional neural network. In *International Conference on Machine Learning* 597–606 (ICML, 2015).
14. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision* 818–833 (Springer, 2014).
15. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
16. Luo, W., Li, Y., Urtasun, R. & Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems* 4898–4906 (NIPS, 2016).
17. Jetley, S., Lord, N. A., Lee, N. & Torr, P. Learn to pay attention. In *International Conference on Learning Representations* (ICLR, 2018).
18. Li, K., Wu, Z., Peng, K.-C., Ernst, J. & Fu, Y. Tell me where to look: guided attention inference network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 9215–9223 (IEEE, 2018).
19. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* 5998–6008 (NIPS, 2017).
20. Cui, Y. et al. Attention-over-attention neural networks for reading comprehension. In *Proc. 55th Annual Meeting of the Association for Computational Linguistics* Vol. 1, 593–602 (ACL, 2017).
21. Ghorbani, A., Abid, A. & Zou, J. Interpretation of neural networks is fragile. *Proc. 33rd AAAI Conference on Artificial Intelligence* 3681–3688 (AAAI, 2019).
22. Athalye, A., Engstrom, L., Ilyas, A. & Kwok, K. Synthesizing robust adversarial examples. In *Proc. 35th International Conference on Machine Learning* (eds Dy, J. & Krause, A.) 284–293 (PMLR, 2018).
23. Goodfellow, I., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations* (ICLR, 2015).
24. Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. & Mukhopadhyay, D. Adversarial attacks and defences: a survey. Preprint at https://arxiv.org/abs/1810.00069 (2018).
25. Athalye, A., Carlini, N. & Wagner, D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In *Proc. 35th International Conference on Machine Learning* (eds Dy, J. & Krause, A.) 274–283 (PMLR, 2018).
26. Khoury, M. & Hadfield-Menell, D. On the geometry of adversarial examples. Preprint at https://arxiv.org/abs/1811.00525 (2018).
27. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A. & Madry, A. Robustness may be at odds with accuracy. In *International Conference on Learning Representations* (ICLR, 2019).
28. Stutz, D., Hein, M. & Schiele, B. Disentangling adversarial robustness and generalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 6976–6987 (IEEE, 2019).
29. Weng, T.-W. et al. Evaluating the robustness of neural networks: an extreme value theory approach. In *International Conference on Learning Representations* (ICLR, 2018).
30. Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y. & Usunier, N. Parseval networks: improving robustness to adversarial examples. In *Proc. 34th International Conference on Machine Learning* 854–863 (JMLR, 2017).
31. Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D. & Jacobsen, J.-H. Invertible residual networks. In *Proc. 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) 573–582 (PMLR, 2019).
32. Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations* (ICLR, 2018).
33. Russakovsky, O. et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision (IJCV)* **115**, 211–252 (2015).
34. Carlini, N. & Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy* 39–57 (IEEE, 2017).
35. Pei, K., Cao, Y., Yang, J. & Jana, S. Deepxplore: Automated whitebox testing of deep learning systems. In *Proc. 26th Symposium on Operating Systems Principles* 1–18 (ACM, 2017).
36. Cohen, J., Rosenfeld, E. & Kolter, Z. Certified adversarial robustness via randomized smoothing. In *Proc. 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) 1310–1320 (PMLR, 2019).
37. Liao, F. et al. Defense against adversarial attacks using high-level representation guided denoiser. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 1778–1787 (IEEE, 2018).

38. Kurakin, A. et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems* 195–231 (Springer, 2018).
39. Tramèr, F. et al. Ensemble adversarial training: attacks and defenses. In *International Conference on Learning Representations* (ICLR, 2018).
40. Wong, E., Schmidt, F., Metzen, J. H. & Kolter, J. Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems* 8400–8409 (NIPS, 2018).
41. Su, D. et al. Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models. In *European Conference on Computer Vision* 631–648 (Springer, 2018).
42. Krizhevsky, A. & Hinton, G. *Learning Multiple Layers of Features from Tiny Images*. Technical report, Univ. Toronto (2009).
43. Rony, J. et al. Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *The IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2019).
44. Shiraishi, J. et al. Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *Am. J. Roentgenol.* **174**, 71–74 (2000).
45. Lin, T.-Y. et al. Microsoft COCO: common objects in context. In *European Conference on Computer Vision* 740–755 (Springer, 2014).
46. Carlini, N. et al. On evaluating adversarial robustness. Preprint at https://arxiv.org/abs/1902.06705 (2019).
47. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
48. He, K., Zhang, X., Ren, S. & Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision* 630–645 (Springer, 2016).
49. He, T. et al. Bag of tricks for image classification with convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 558–567 (IEEE, 2019).
50. Paszke, A. et al. Automatic differentiation in PyTorch. In *Proc. 31st Conference on Neural Information Processing Systems* (NIPS, 2017).
51. Yamada, Y., Iwamura, M., Akiba, T. & Kise, K. Shakedrop regularization for deep residual learning. Preprint at https://arxiv.org/abs/1802.02375 (2018).
52. Huang, G., Sun, Y., Liu, Z., Sedra, D. & Weinberger, K. Q. Deep networks with stochastic depth. In *European Conference on Computer Vision* 646–661 (Springer, 2016).

Acknowledgements

This work was supported in part by the Center for Brain-Inspired Computing (C-BRIC), one of six centres in the Joint University Microelectronics Program (JUMP), a Semiconductor Research Corporation (SRC) programme sponsored by the Defense Advanced Research Projects Agency (DARPA). W.W. acknowledges additional funding from Defense Threat Reduction Agency (DTRA) (award no. HDTRA1-18-1-0009). J.C. acknowledges funding from the Maseeh College of Engineering & Computer Science's Undergraduate Research and Mentoring Program and the SRC Education Alliance (award no. 2009-UR-2032G). W.W. and J.C. received funding from F. Maseeh. We thank A. Madry[27,32] and J. Cohen[36] for helpful discussions and clarifications about their work. We thank FuR and A. Parise for assisting with the collection of photos for the examples throughout this work.

Author contributions

W.W. contributed the original idea, algorithms, experimental design, ablation studies, some active learning annotations and wrote the majority of the paper. J.C. conducted LIME and Grad-CAM integrations, annotated the majority of the active learning annotations, provided text for the active learning sections of the paper and contributed editing support. C.T. provided scope advisement, editing support and funding for the work.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0104-6.

Correspondence and requests for materials should be addressed to W.W., J.C. or C.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.