Computational Analysis of Robustness in Neural Network Classifiers

Iveta Bečková, Štefan Pócoš, and Igor Farkaš

Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
Mlynská dolina 84248 Bratislava, Slovakia

Abstract. Neural networks, especially deep architectures, have proven excellent tools in solving various tasks, including classification. However, they are susceptible to adversarial inputs, which are similar to original ones, but yield incorrect classifications, often with high confidence. This reveals the lack of robustness in these models. In this paper, we try to shed light on this problem by analyzing the behavior of two types of trained neural networks: fully connected and convolutional, using MNIST, Fashion MNIST, SVHN and CIFAR10 datasets. All networks use a logistic activation function whose steepness we manipulate to study its effect on network robustness. We also generated adversarial examples with FGSM method and by perturbing those pixels that fool the network most effectively. Our experiments reveal a trade-off between accuracy and robustness of the networks, where models with a logistic function approaching a threshold function (very steep slope) appear to be more robust against adversarial inputs.

Keywords: robustness · logistic sigmoid · pixel attack

1 Introduction

Neural networks (especially with deep architectures) have demonstrated excellent performance in various complex machine learning tasks such as image classification, speech recognition, or natural language processing [1]. At the same time, neural networks in general are due to their multilayer nonlinear structure not transparent and so it is hard to understand their behavior. Therefore, a lot of effort has been dedicated to uncovering their function [6]. In addition, it was discovered that solutions they provide are surprisingly not robust, and that they can be fooled with carefully crafted images called adversarials, which are similar to original ones. The notion of adversarial images was introduced in [14], where it was shown that even small perturbations of input can lead to misclassification with a high confidence. It is even possible to fool the classifier with single pixel attacks as demonstrated in [12]. There exist multiple ways to generate these adversarial inputs and it is applicable not only for images but many other kinds of neural network as for example voice recognition [16]. Thus the ability of a network to resist adversarial attacks (robustness) became a major

concern. Several papers already proposed methods how to alleviate the problem. For instance, enforcing compactness in the learned features of a convolutional neural network by modifying the loss function was observed to enhance robustness [9]. Alternatively, a modified version of conditional GAN network was used to generate boundary samples with true labels near the decision boundary of a pre-trained classifier, and these were added to the pre-trained classifier as data augmentation to make the decision boundary more robust [13]. Other papers have focused on the relationship between robustness and model accuracy. In [8] this opposing relationship is more formally explored and shown how the evoked tension between the two properties impacts the accuracy of models.

Related line of research has focused on looking at internal representations and their relation to model performance. In our recent work, we analyzed trained deep neural network classifiers in terms of learned representations (and their properties) and confirmed that there exist qualitatively different, but quantitatively similar (with similar testing errors) solutions to the complex classification problems, depending on activation functions used in the hidden layers [4]. One of these standard activations is the logistic function whose effect is investigated here, but in the context of robustness.

In the following, we provide our pilot results on the analysis of the robustness and its relation to accuracy of the trained relatively simple feedforward networks. We introduce four image datasets and the models used (Section 2), methods of analysis of trained models (Section 3), their results (Section 4), and conclusion (Section 5).

2   Data and models

We used four well-known datasets, with increasing levels of complexity, namely MNIST, Fashion-MNIST, SVHN and CIFAR10, to be able to better track the analysis.

The MNIST database is a set of 28×28 pixel grayscale images corresponding to ten classes. It is made of centered, upright hand-written digits on black background, with 60000 samples in the training set and 10000 in the testing set. Since MNIST is one of the most basic datasets for image classification, one does not require a deep convolutional network to classify the images with satisfying accuracy [5].

Fashion MNIST (referred to as F-MNIST) is a dataset created due to overuse of MNIST. It consists of 28×28 pixel grayscale images, where each of them belongs to one of ten classes describing clothing or accessory. Training set (50000 samples) and testing set (10000 samples) make up all 60000 samples of the dataset [15].

The SVHN (Street View House Numbers) database consists of 32×32 pixel RGB images of house numbers, each image centered on one digit, thus belonging to one of ten classes. We used ∼73000 images for training and ∼26000 for testing [7].

CIFAR10 provides variety of 32×32 pixel RGB images, belonging to one of ten classes representing an animal or a vehicle. The number of images included in training and testing sets are 50000 and 10000 respectively [3].

For easier datasets (MNIST and F-MNIST) we use a multilayer perceptron with a single hidden layer containing 256 neurons with a logistic sigmoid activation function and softmax outputs. An architecture for SVHN and CIFAR10 classification is depicted in Fig. 1. To optimize the learning process we use Adam optimizer and cross entropy loss. The training length of models is set to 100 epochs for MNIST and F-MNIST and 50 epochs for SVHN and CIFAR10. All models are trained with a batch size 64 and their properties are evaluated after each epoch.
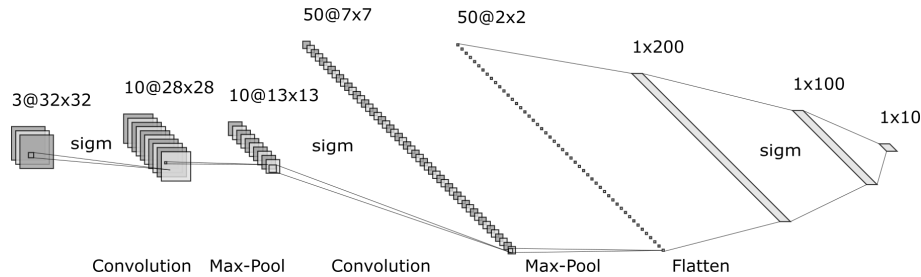

Fig. 1: SVHN and CIFAR10 classifier architecture (plotted using NN-SVG tool).

3   Methods of analysis

In our computational analysis we monitored the distribution of neuron's activation values during training and computed the gradient of the loss function with respect to the input. To make the monitoring of the neuron activation levels more understandable, we quantized these into three regimes. We expected to find a relation between these regimes and the network robustness, evaluated by generating adversarial examples using two different methods.

3.1   Quantized activation levels

We investigated the effect of temperature T in the logistic sigmoid defined as

$$f(net) = \frac{1}{1 + \exp(-net/T)} \qquad (1)$$

By modifying the value of T one can adjust the slope at $net = 0$, thus affecting the distribution of function values. We considered $T = \{1/64, 1/32, \ldots, 4, 16\}$ for MNIST and F-MNIST datasets and $T = \{1/8, 1/4, \ldots, 4, 8\}$ for SVHN and CIFAR10, since these temperatures allowed to train the models with higher

accuracy. Using values beyond these ranges in many cases led to significant drop of network performance.

To test how the choice of temperature affects the activations of hidden neurons, we divide *net* values of neurons on the last hidden layer, which is a set of real numbers $\mathcal{R}$ into three disjoint regions – linear, nonlinear and saturated, dependent on T. We define the linear part as follows: first we calculate the tangent line of the logistic sigmoid at $net = 0$. Since the first derivative is symmetrical and monotonous for $net \in [0, \infty)$, with an increasing (or decreasing) *net*, the value $f(net)$ moves away from (closer to) this line. We choose a threshold δ_1 and numerically approximate the smallest $net > 0$, for which the Euclidean distance of $(net, f(net))$ from this tangent line is larger than δ_1. Let's denote it r_1. Then the linear part is $L = [-r_1, r_1]$.

The saturated part is defined using derivatives. The first derivative of logistic sigmoid is zero for $net = \pm\infty$, so these should be included in the saturated part. The same as with the linear part, we pick a threshold value δ_2 and numerically approximate the greatest positive *net* with the absolute value of the first derivative greater than threshold. We denote it r_2. Then the saturated part is defined as $S = (-\infty, r_2] \cup [r_2, \infty)$.

The remaining, nonlinear part $N = \mathcal{R} \setminus L \setminus S = (-r_2, -r_1) \cup (r_1, r_2)$. It's important to note, that depending on the temperature and choice of δ_1 and δ_2, $r_2 > r_1$ does not always hold. In that case, $N = \emptyset$, so to avoid this, we choose $\delta_1 = 0.05$ and $\delta_2 = 0.0005$. Corresponding values of r_1 and r_2 for tested temperatures are shown in Fig. 2.

Then, we define the operation of a tested network in the respective regime as a fraction of *net* values occurred in that region.
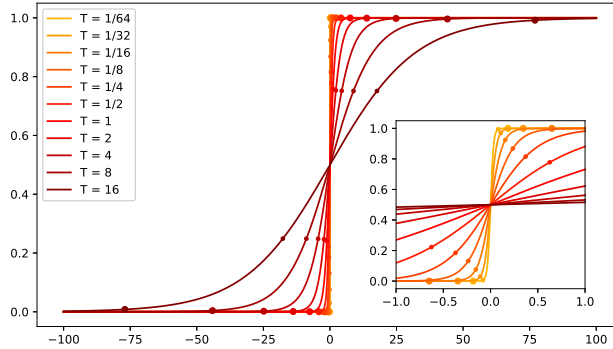

Fig. 2: Logistic sigmoid $f(net)$ with various temperatures. The larger dots denote $\pm r_2$, and smaller ones $\pm r_1$. These dots delineate the three different activity regions used in analysis.

3.2   Robustness

For testing the robustness we use fast gradient sign method (FGSM) described in [2], which works as follows. First, the gradient of loss function with respect to input image is calculated. This gradient shows such a local change of each pixel of input image that the loss of network increases the most. Second, the attack itself is executed by adding the sign of the corresponding gradient multiplied by a small factor ϵ to each input pixel. For creating adversarial examples we mainly use $\epsilon = 0.1$ for each sample from the testing sets. Then the class robustness is defined as $1-$ fraction of adversarial inputs created from samples belonging to the given class that are classified incorrectly. Finally, network robustness is evaluated as an average robustness over all classes. This is the most frequent way to analyze the network robustness and is used in many applications such as [11]. On the other hand, there exist more complex definitions of robustness, that mainly involve measuring the strength of perturbation. Some examples can be found in [9,13]. Since we generate the adversarial examples with a given magnitude of perturbation, we stick to the most basic but effective evaluation of robustness.

In the FGSM attack, the original input image is changed by adding the sign of a gradient of the loss function with respect to the input. Thus the magnitude of change is the same for each pixel (except for pixels whose values in adversarials would exceed $[0, 1]$; these are trimmed). This does not take into account the magnitude of gradients, or differences among pixels. Some sources [10] suggest the use of fast gradient value method (FGVM) that does not incorporate the sign. We decided to use FGSM but we also analyzed the values of gradients by computing $G(x_i) = \sum_{n=1,m=1}^{N,M} |\frac{\partial L}{\partial x_i(n,m)}|/(N.M)$, i.e. the arithmetic mean of absolute values of partial derivatives of the loss function L with respect to individual pixels (in case of SVHN and CIFAR10 also for individual channels).

3.3   Pixel attack

Another way to generate adversarial examples besides FGSM (which adds minimal perturbation to all pixels) is to seek such (greater) perturbations of only a few pixels so that the modified input would not be classified correctly. An important step here is the pixel selection. In order to select those, which cause the biggest change in the output probabilities, we look at the gradients mentioned above. We sort the pixels by absolute values of the corresponding gradients. If we choose top n pixels, there is a chance for misclassification. Naturally, the higher n implies the bigger chance to misclassify the input, but as we show, in many cases even one pixel can do the trick. Therefore, using the same definition of class robustness as above, we get another measure of model's robustness just by changing the way of generating the adversarials.

4   Experiments

We managed to train all our models with up to 98%, 90%, 92% and 65% testing accuracy on MNIST, F-MNIST, SVHN and CIFAR10 respectively. In some cases (mainly for CIFAR10) the accuracy could have been significantly improved by using a more sophisticated architecture and training methods, but for subsequent analysis of robustness the given accuracy was satisfactory. In all four datasets, the models with extreme temperatures (too high or too low) were the most difficult to train, as shown in Fig. 3. Large values of T yield too small gradients, and too small T leads to zero gradients except $net \approx 0$ where they are enormous. Under these extreme circumstances the hardest task for a given network is to start the learning procedure.
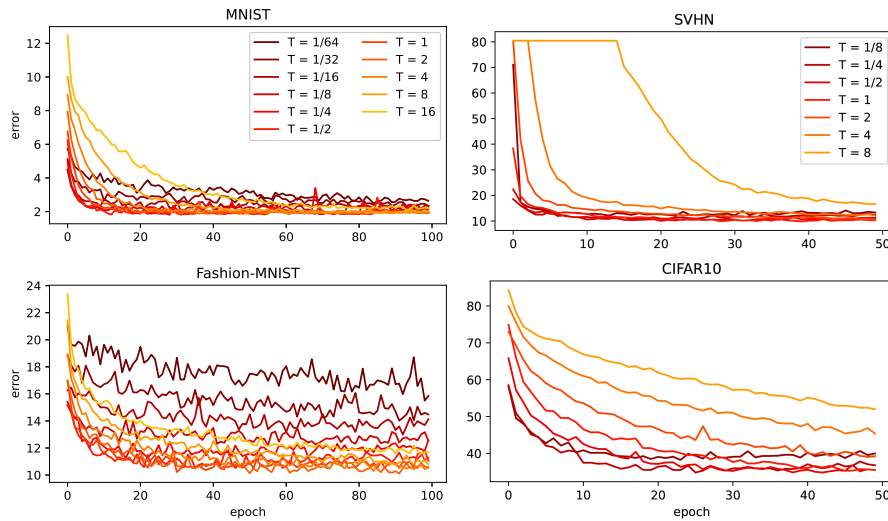

Fig. 3: Learning curves for all four datasets using selected temperatures. Figures on either side use the same temperatures, shown in upper figure.

4.1   Quantized activation levels

After each epoch of training we analyzed our model's properties. All of the trained networks show similar behavior. Linearity drastically drops with decreasing temperature. In these areas high level of saturation is detected. Detailed caption of these properties is shown in Fig. 4.

All the figures, quite consistent for all four datasets, reveal the dynamics of gradual transitions between respective operation modes during learning. For most values of T, the network gradually spends less time in the linear and nonlinear modes (albeit the baseline for the latter is significantly higher), and more
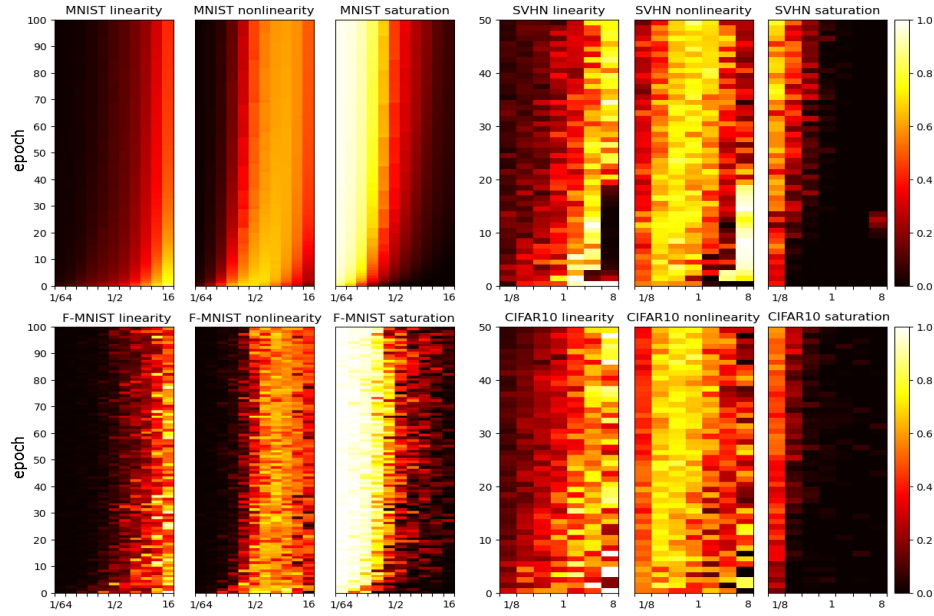
Fig. 4: Operation in the three modes during training for all datasets and various values of T (on x-axis). Consistent gradual shift of the dominant regime is observed in all cases, in terms of gradual transition from linear and nonlinear modes towards the saturation mode.
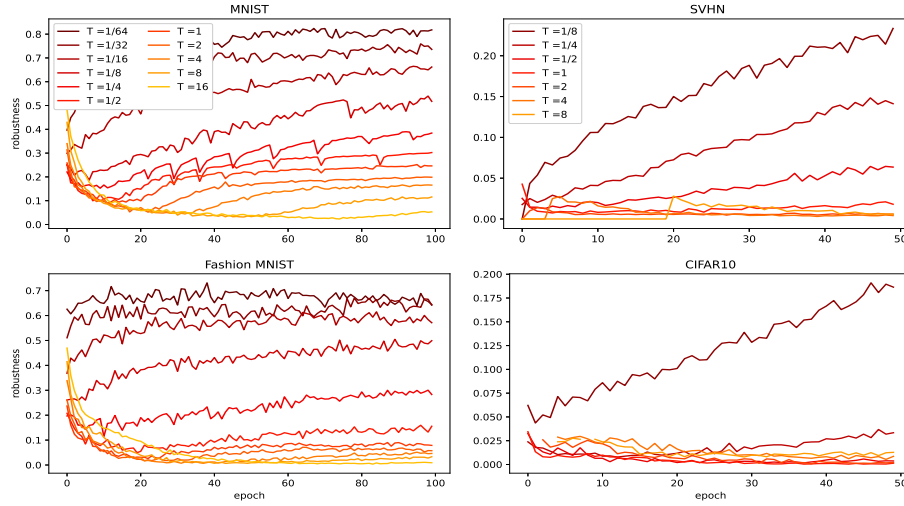

Fig. 5: Development of network robustness during training on all datasets. The same pattern is visible in all cases.
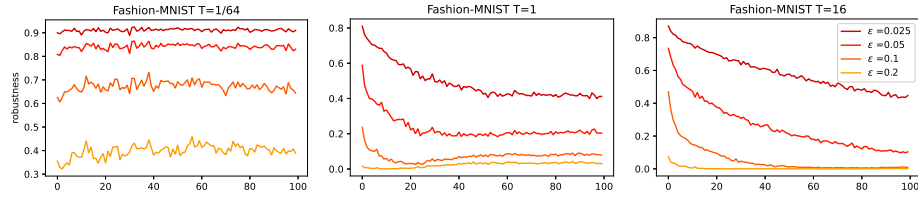
Fig. 6: Development of network robustness during training on Fashion-MNIST dataset evaluated for different values of ϵ.

time in the saturation mode. All transitions highly depend on T. In case of MNIST, these transitions are quite smooth, probably due to lower task complexity.

Fig. 5 shows how FGSM robustness evolves during training, evaluated for fixed $\epsilon = 0.1$. One can notice that all of the trained classifiers on four datasets are more robust when using lower sigmoid temperatures and less robust for high temperatures. Also, evolution of robustness profiles (for any T) tends to correlate with that of the saturation modes in all datasets, implying that increasing robustness is associated with growing contribution of the saturation mode. We can also observe vast difference in the level of robustness between feedforward and convolutional networks. CIFAR10 and SVHN are much less robust in comparison to MNIST and F-MNIST, regardless the temperature and network accuracy. We also generated adversarial examples using different values of ϵ and then evaluated the robustness of individual networks. According to our expectations we consistently see that the higher ϵ we use, the more successful the attack is. Also it is still clear that networks trained with lower temperatures are more robust. This phenomenon is demonstrated on F-MNIST dataset in Fig. 6.

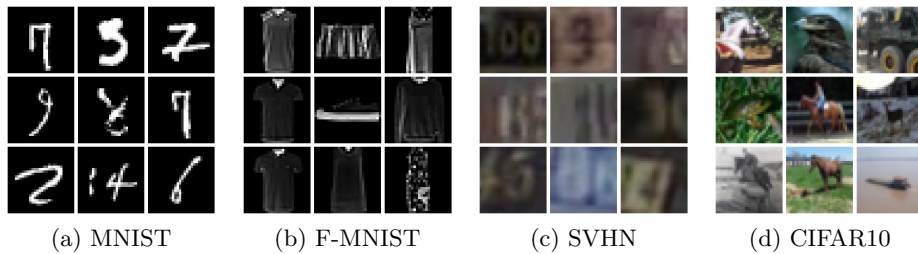(a) MNIST   (b) F-MNIST   (c) SVHN   (d) CIFAR10

Fig. 7: Examples of images from the four datasets with large $G(x_i)$, all obtained using models with $T = 1$. These images are more susceptible to misclassification.

Next, we looked at gradients. First we tried to average $G(x_i)$ (described in Sect. 3.2) across all samples from the testing sets, hoping to find some relation
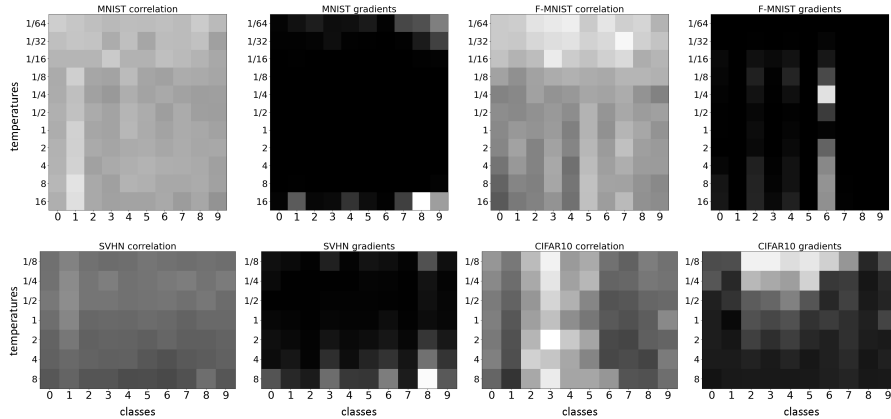
Fig. 8: Class-specific medians of input gradients $G(x_i)$ and the correlation coefficients between $G(x_i)$ and the target class confidence. All pictures showing correlations are in the range $[-1, 0]$. Figures showing gradient magnitudes are in range $[0, 0.0002]$ for MNIST, $[0, 0.2]$ for Fashion-MNIST, $[0, 0.04]$ for SVHN and $[0, 0.15]$ for SVHN. All ranges are scaled from black to white.

between T and $G(x_i)$. However, we found that some inputs had much higher gradient magnitudes than others, thus affecting the average value significantly. So, we eventually used the median instead. Also, after plotting a few examples of these high gradient inputs, we found (as shown in Fig. 7) that they were not "pretty". Some of them are cropped incorrectly (MNIST dataset), blurred (SVHN dataset), or too dark (F-MNIST dataset). Some are ambiguous even to humans. Therefore, we also looked at the correlation coefficient between the output confidence of the correct class (given by the activation value of the representing output neuron) and the input gradient $G(x_i)$ to find out if these inputs are indeed for our models hard to classify correctly.

Fig. 8 depicts how the median values of $G(x_i)$ depend on T and an input class. It seems that the network has generally higher gradients when it is trained with too big or too small temperatures, exception being networks trained on Fashion-MNIST, where gradients depend more on input class than T. The correlation between median of $G(x_i)$ and the correct class confidence is similar in all datasets (strong and negative), significant at $p < .001$ in almost all cases (exception being the class 3 in CIFAR10 dataset). This means that inputs with greater gradients are more likely to be classified incorrectly.

4.2   Pixel attack

We conducted the search for perturbations changing up to 3 pixels in case of MNIST and F-MNIST images, and up to 2 pixels in SVHN and CIFAR10 images. After selecting the pixels for perturbation, we sought all combinations until the adversarial input was found (or not). For simplicity we tried to change each
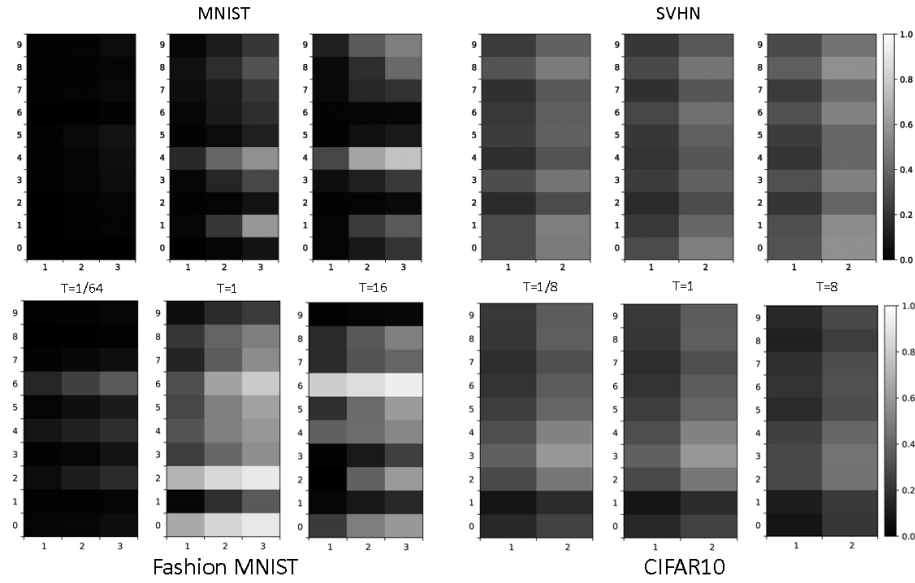
Fig. 9: Success of pixel attack for different classes and all four datasets for different T. Columns correspond to the number of perturbed pixels and rows to class index. The lower the temperature, the less likely it is to generate successful perturbation resulting in misclassification. Relative robustness remains the same for all temperatures. We see that in MNIST and F-MNIST, different classes yield different robustness, while in SVHN and CIFAR10 dataset, they are more equal.

pixel to the lowest and highest value, i.e. 0 and 1, respectively. After running through the test set, we evaluated the percentage of images for which we found at least one perturbation that caused misclassification. It holds that the lower the percentage, the more robust is the network.

Pixel attack on SVHN and CIFAR10 images is slightly different because of 3 color channels. In this case, the arithmetic mean of absolute values of gradients was computed for each pixel, as we get 3 gradients per pixel, each for one of the color channels. Pixel attack then consists of perturbation of three times more values. Results depicted in Fig. 9 reveal high similarity between pixel attack and FGSM robustness, since the higher the temperature, the less robust is the network, except for F-MNIST dataset. Fig. 10 shows a few examples on two selected datasets of original and perturbed images, along with network's output confidences. One can notice that perturbation of even two pixels may cause misclassification with a high confidence score.

1: 1.000	1: 0.001	2: 1.000	2: 0.445	0: 0.999	0: 0.227	1: 0.998	1: 0.228
7: 0.000	7: 0.999	8: 0.000	8: 0.555	6: 0.000	6: 0.757	9: 0.001	9: 0.766

4: 1.000	4: 0.000	6: 1.000	6: 0.119	2: 0.988	2: 0.002	6: 0.947	6: 0.398
9: 0.000	9: 1.000	5: 0.000	5: 0.881	3: 0.006	3: 0.475	5: 0.053	5: 0.602

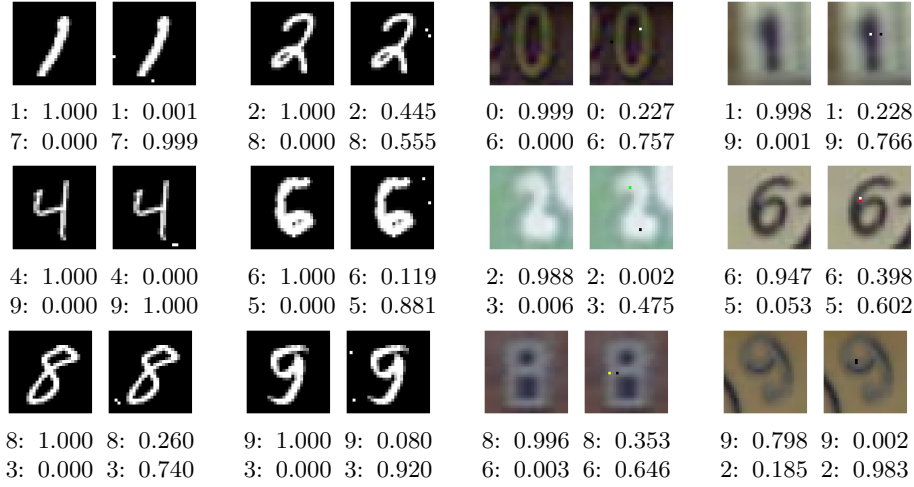|							
8: 1.000	8: 0.260	9: 1.000	9: 0.080	8: 0.996	8: 0.353	9: 0.798	9: 0.002
3: 0.000	3: 0.740	3: 0.000	3: 0.920	6: 0.003	6: 0.646	2: 0.185	2: 0.983

Fig. 10: Visualization of input images (from MNIST and SVHN datasets), classified correctly and adversarials created by altering only two pixels (for $T = 1$). Below the individual images is shown the network confidence (after softmax) for correct classification as well as for the winner class of the corresponding adversarial input. Interestingly, even the perturbed pixels at the boundaries (outside the digits) can evoke successful adversarials as seen in case of MNIST digits.

5   Conclusion

We looked at potential factors affecting the robustness which has been revealed as a general problem of end-to-end trained neural network classifiers. This gradient-based approach apparently does not induce any stability of categories during learning, despite the finding that hidden layers tend to learn features with a growing abstraction that eventually enable correct classification on novel data. In our computational experiments, we found many significant differences among classifiers for various temperatures, most crucial being the ability to train and solve the task satisfactorily, which is lost for too extreme temperatures. The speed of training is also altered, with greater temperatures converging more slowly. That also implies the difference in the network's linearity/saturation.

   Probably the most interesting result is that two different methods of evaluating robustness both showed that lower temperatures lead to higher robustness of the models. On the other hand, shallow and simple feed-forward network yielded much higher level of robustness than a convolutional network. Another important finding are the inter-class differences in robustness, but the similarity of robustness for different temperatures. Our analysis of input gradients discovered some similarities even between MNIST and SVHN dataset classes. There was a strong negative correlation between the magnitude of input gradients and the network output confidence of the correct class, suggesting that by evaluating

input gradients one can select inputs that are likely to be classified incorrectly. These findings might be informative for the training methods that could lead to increased robustness of the networks against adversarial inputs.

References

1. Deng, L., Yu, D.: Deep learning: Methods and applications. Foundations and Trends in Signal Processing **7**(3–4), 197–387 (2014). https://doi.org/10.1561/2000000039
2. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014), http://arxiv.org/abs/1412.6572
3. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. Rep. TR-2009, University of Toronto (2009)
4. Kuzma, T., Farkaš, I.: Computational analysis of learned representations in deep neural network classifiers. In: IEEE International Joint Conference on Neural Networks (IJCNN) (2018)
5. LeCun, Y., Cortes, C., Burges, C.: The MNIST database of handwritten digits (2010), http://yann.lecun.com/exdb/mnist/
6. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (2018). https://doi.org/10.1016/j.dsp.2017.10.011
7. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
8. Papernot, N., McDaniel, P.D., Sinha, A., Wellman, M.P.: Towards the science of security and privacy in machine learning (2016), http://arxiv.org/abs/1611.03814
9. Ranjan, R., Sankaranarayanan, S., Castillo, C.D., Chellappa, R.: Improving network robustness against adversarial attacks with compact convolution. CoRR (2017), http://arxiv.org/abs/1712.00699
10. Rozsa, A., Rudd, E., Boult, T.: Adversarial diversity and hard positive generation. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 410–417 (2016). https://doi.org/10.1109/CVPRW.2016.58
11. Schott, L., Rauber, J., Brendel, W., Bethge, M.: Robust perception through analysis by synthesis. CoRR (2018), http://arxiv.org/abs/1805.09190
12. Su, J., Vargas, D.V., Kouichi, S.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation **23**(5) (2019). https://doi.org/10.1109/TEVC.2019.2890858
13. Sun, K., Zhu, Z., Lin, Z.: Enhancing the robustness of deep neural networks by boundary conditional GAN (2019), http://arxiv.org/abs/1902.11029
14. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6199
15. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (2017), https://arxiv.org/abs/1708.07747
16. Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., Xu, W.: DolphinAttack: Inaudible voice commands. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. pp. 103–117 (2017)