ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

PCA improves the adversarial robustness of neural networks

Ammar Al-Najjar and István Megyeri *

University of Szeged, Hungary

Abstract. Deep neural networks perform well in many visual recognition tasks, but they are sensitive to adversarial input perturbation. More robust models can be learned when attacks are applied to the training data or preprocessing is used. However, the effect of preprocessing is frequently underestimated and it has not received sufficient attention as it usually does not affect the network's clean accuracy. Here, we seek to demonstrate that preprocessing can play a role in improving adversarial robustness. Our empirical results show that principal component analysis, a simple yet effective preprocessing method, can significantly improve neural networks' robustness for both regular and adversarial training.

1 Introduction

Improving neural network robustness to input perturbations is still a challenging task. Although in recent years, researchers have made great progress for both empirical [1, 2] and certified robustness [3, 4], the gap between clean accuracy and robust accuracy is still quite large.

Here, we will simplify the robustness problem by eliminating certain input modifications. We expect that this elimination will simplify the robustness problem and help us to get a more robust representation. To reduce perturbations, we used principal component analysis(PCA), which maximizes the variance of the projected data and eliminates irrelevant input features.

Our empirical results confirm this when, we trained models jointly with PCA preprocessing and observed improved robustness for both normal and adversarial training. We used white box gradient attacks which assumed that the attacker is aware of the used preprocessing.

There is a line of work, where PCA is used to improve adversarial robustness. In [5], the authors improve certified robustness to l_2 norm bounded perturbations using a low-rank representation and randomized smoothing. The essence of their method is the randomized smoothing applied in a low-rank space that is estimated by sparsified PCA. They also generalized this certification to l_∞ norm bound perturbations.

In [6], the authors improved robustness to l_∞ norm bounded perturbations by introducing a prepossessing method. The method consists of two steps, namely randomly dropping some pixels from the input image and reconstructing the image with the matrix estimation method. The proposed method uses randomness to destroy the adversarial perturbation in the masking step. In the

*The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program (RRF-2.3.1-21-2022-00004).

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

reconstructing phase, their method requires estimating the matrix for each input image since it is assumed that the matrix to be reconstructed is a single image.

Our contributions can be summarized as follows. We designed experiments which demonstrate that PCA alone can improve robustness without randomization. In the related works, PCA was applied with an arbitary number of principal components. In contrast, our findings suggest that proper selection is essential. Also, PCA improves robustness to l_1, l_2, l_∞ norm bounded perturbation. So far l_1 robustness improvement was hidden.

2 Combining principal component analysis and neural networks

Let us introduce our notations. We assume there is a set of examples in the form of (x, y) pairs and $x \in \mathbb{R}^d$ where d is the number of input features or intensity values in an image. $y \in \mathcal{C}$ is the class label and $\mathcal{C}$ is the set of all labels. We define $X \in \mathbb{R}^{n \times d}$, which denotes n images in matrix form. Further, we will use linear projections on an x sample with a projection matrix $P \in \mathbb{R}^{d \times m}$. It projects the input image onto an m dimensional space. Our aim here is to combine the projection matrix with neural networks that will perform the classification to improve their robustness. We will denote the neural network by f, which is a function that maps its input to the label set $\mathcal{C}$ with weight values θ.

Let us define our baseline model $f_{baseline}$. It has learnable parameters P_{bl} its projection matrix and θ_{bl} are the neural network weights. Note that both the projection matrix and the neural networks are trained jointly according to the training procedure. The output of the network is given by $f_{baseline}(xP_{bl}, \theta_{bl})$.

As a stronger baseline, we consider a neural network with random projection f_{random}. It onnly has θ_r as learnable weights. The projection matrix P_r is sampled from a normal distribution with zero mean and $\frac{1}{m}$ variance and it is frozen during training. The output of the network is given by $f_{random}(xP_r, \theta_r)$.

We have two baseline models, and we will define models that employ PCA. We applied PCA in the following way. First, we got the correlation matrix $X^T X$, which is a $d \times d$ matrix. The d is usually much less than n the number of samples in the dataset, so we have a more efficient method than simply applying PCA on the original matrix. Next, we calculate the singular value decomposition(SVD) of $X^T X$, which will give us three matrices U, S and W. U and W are orthogonal matrices and S is a diagonal matrix. It can readily be seen that the W obtained this way is equivalent to the one which is given by the SVD applied on the X matrix and S is the square of the X's singular values.

f_{pca} is defined with $P_{pca} = W_m$ as a projection matrix. Here, the subscript means that we took only the first m rows of W. Note that the rows are sorted according to values in S and the earlier a row is, the more variance is attributed by it. There are also learnable weights θ_p but P_p is frozen again so it will not change during training. The output of the network is given by $f_{pca}(xP_p, \theta_p)$.

We have another variant with reconstruction to combine PCA and neural network training called f_{pca-r}. The projection matrix same $P_{pr} = W_m$ but before passing the projected data to network we reconstruct it, therefore the output of the network is $f(xP_{pr}P_{pr}^T, \theta_{pr})$. Note that since W is orthogonal, simply trans-

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

	Explained variance ratios						
Dataset	0.75	0.8	0.85	0.9	0.95	0.99	0.999
MNIST	16	23	34	53	103	281	458
F-MNIST	2	4	6	16	65	319	619

Table 1: The number of components/neurons in the linear layer. They were selected according to the explained variance ratios got by PCA. These are the percentages of variance that are attributed to the selected components.

posing it will give the inverse matrix that can transform the projected data back to the input space. However, the reconstruction lose certain information because we don't use the full matrix. We expect that this information loss don't harm the accuracy but rather improve its robustness to irrelevant modifications.

3 Model training and evaluating robustness

3.1 Training objectives

We tried out two training strategies, namely normal training(NT) and adversarial training(AT). NT means that the models are trained on the clean images using the categorical cross-entropy loss function. In the case of AT, l_∞ projected gradient descent(PGD) was applied on each train batch for 10 steps with a step size 0.025 and $\epsilon = 0.1$. The training objective was again the categorical cross-entropy loss.

3.2 Databases and hyper parameters

We conducted our experiments on two benchmark image datasets called MNIST [7] and Fashion-MNIST [8]. We normalized the 28×28 gray scale images so that the intensity values have a range of $[0, 1]$. For model selection purposes, we excluded 1000 samples from the train set and used them as a validation set.

We trained all the models with the Adam [9] optimizer using of 10^{-4} learning rate and batch size of 50 for 100 epochs. The best checkpoint was selected based on validation accuracy or robust accuracy according to the training objective.

We trained fully connected neural networks with the following number of hidden layers: $0, 1, 2$ and 4. Here 0 means that there is no hidden layer so it is equivalent to a multi-class logistic regression classifier. In each hidden layer, we used Relu activation and added 256 neurons. The number of components/neurons in the first linear layer for all the models was evaluated for the range of values given in Table 1. The values were selected according to the explained variance ratio from 0.75 up to 0.999. These are the percentages of variance that are attributed to the selected components.

3.3 Robustness measures

For robustness measures, we applied PGD variants with l_1, l_2, l_∞ norm bounds using the FoolBox [10] library. As we implemented PCA as part of the neural network architecture, both the preprocessing and the neural networks were used

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

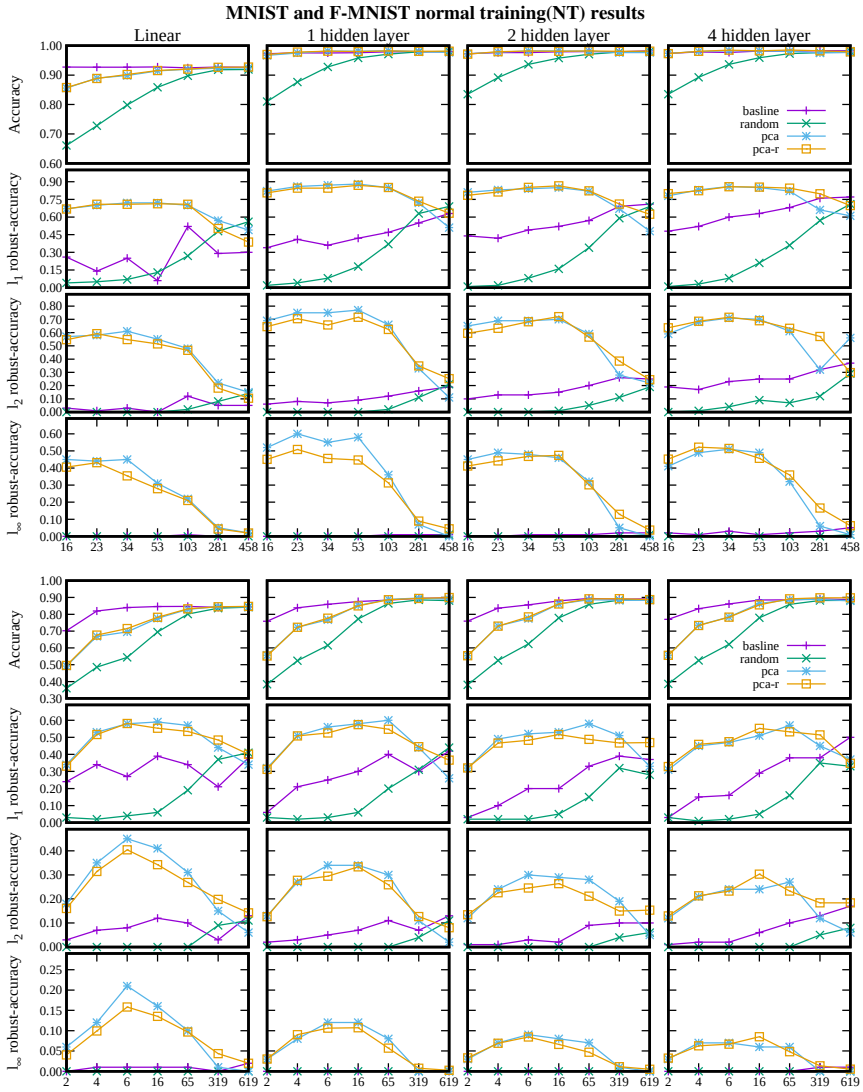

Fig. 1: Accuracy and robust accuracy are shown as a function of m, the dimension of the projected space. MNIST model results are at the top and F-MNIST are at the bottom. All the models were trained using clean images. Models combined with PCA have a similar accuracy while displaying improved robustness in all the norms.

for gradient propagation. This is a white-box scenario when the attacker is fully aware of the model pipeline. For each norm, we used 100 steps and 10 restarts with random initialization and the step size was set to $\epsilon/100 * 2.5$. The ϵ values were $10, 1, 0.1$ for norms l_1, l_2 and l_∞, respectively.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

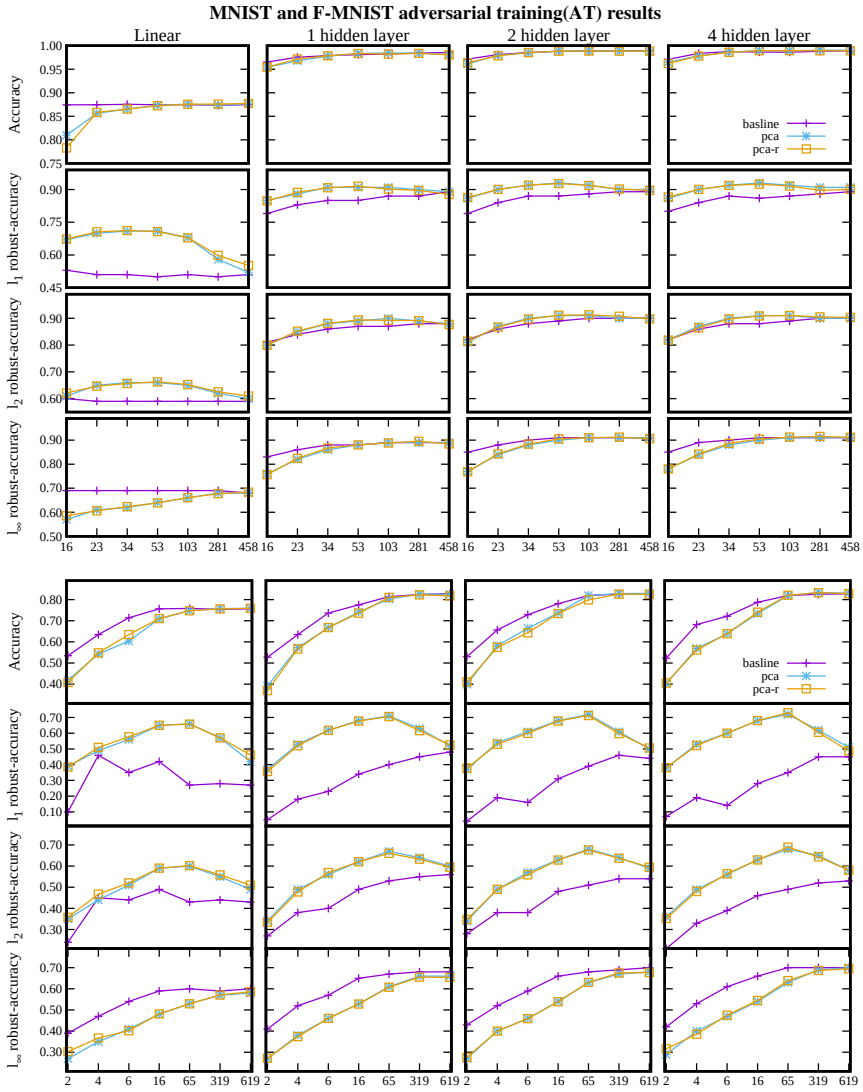

Fig. 2: Accuracy and robust accuracy are shown as a function of m, the dimension of the projected space. MNIST model results are at the top and F-MNIST on at the bottom. All the models were trained using l_∞ PGD. Models combined with PCA show improved robustness for l_1 and l_2 norm attacks, while accuracy and l_∞ robust accuracy are comparable to the baseline model.

3.4 Results

Figure 1 and 2 show accuracy and robust accuracy for NT and AT results, respectively. In the case of a small projection size, baseline models provide

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

better accuracy on clean images than the other methods. However, this difference diminishes when we add hidden layers and the accuracy values are at the same level for baseline and PCA variants. As for robust accuracy, the methods differ greatly. PCA-based methods have better robustness in all norms than those with the baseline and random projection. Interestingly, robustness curves fot the PCA variants have a sweet spot. This is best seen on the Fashion-MNIST dataset. Figure 2 shows that robust accuracy improved in l_2 and l_1 norms. Note that l_∞ PGD was applied on the training images, while l_2 and l_1 were not.

4 Conclusion

A combination of PCA and neural networks was presented with and without reconstruction. The reconstructed version is more flexible and suitable for CNNs. We demonstrated that the combination methods indeed improve robustness against l_1, l_2 and l_∞ norms for both regular and adversarial training without any randomness. The results also show that there is a sweet spot for projected space size and ad-hoc selection might give unsatisfying robustness.

References

[1] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Int. Conf. on Learning Representations*, 2018.

[2] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Int. Conf. on Machine Learning*, 2019.

[3] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th Int. Conf. on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019.

[4] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sébastien Bubeck. *Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[5] Pranjal Awasthi, Himanshu Jain, Ankit Singh Rawat, and Aravindan Vijayaraghavan. Adversarial robustness via robust low rank representations. In *Proceedings of the 34th Int. Conf. on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[6] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. ME-net: Towards effective adversarial robustness with matrix estimation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th Int. Conf. on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7025–7034. PMLR, 09–15 Jun 2019.

[7] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, November 1998.

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. (cs.LG/1708.07747), 2017.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a Conf. paper at the 3rd Int. Conf. for Learning Representations, San Diego, 2015.

[10] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th Int. Conf. on Machine Learning*, 2017.