IMPROVED RESISTANCE OF NEURAL NETWORKS TO ADVERSARIAL IMAGES THROUGH GENERATIVE PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

We train a feed forward neural network with increased robustness against adversarial attacks compared to conventional training approaches. This is achieved using a novel pre-trained building block based on a mean field description of a Boltzmann machine. On the MNIST dataset the method achieves strong adversarial resistance without data augmentation or adversarial training. We show that the increased adversarial resistance is correlated with the generative performance of the underlying Boltzmann machine.

1 INTRODUCTION

In an adversarial attack on a classification system one seeks inputs that lie very close (according to some norm or measure) to a regular input, but give a completely different classification result. Adversarial attacks on machine learning systems have been studied for over a decade, starting from the study of attacks on spam filtering systems (for a historical review see Biggio and Roli (2018)). Adversarial attacks on deep learning image recognition models have first been discussed by Szegedy et al. (2013). In addition to highlighting security implications adversarial images have acquired a new meaning in this context, presenting us with a fundamental conundrum: How can deep learning systems succesfully generalise and at the same time be extremely vulnerable to minute changes in the input? The topic of adversarial images in neural nets has been studied intensely with attacks and defences leapfrogging each other (for an overview see Kurakin et al. (2018)). All common neural network models suffer from susceptibility to adversarial attacks. It has even been shown by Moosavi-Dezfooli et al. (2017) that adversarial attack patterns can be transferred between different models, suggesting some common underlying mechanism. Despite the intense study of the subject a solution remains elusive, with Jetley et al. (2018) suggesting that classification performance and vulnerability to adversarial attacks are correlated, while Gilmer et al. (2018) present evidence for the opposite in a simple model. The difficulty in finding neural networks robust against adversarial attacks suggest that the standard discriminative neural network training procedure and the resulting networks' vulnerability are linked. Making use of generative methods (e.g variational autoencoders, introduced in this context by Gu and Rigazio (2014)) has resulted in creating the most adversarially robust classifications but with added computational cost at inference time (Schott et al. (2018)).

In this article we concentrate on one common generative model, Boltzmann machines, and the impact of its use on adversarial resistance. We examine the adversarial resistance on the standard MNIST dataset. Boltzmann machines are a stochastic model with binary units (see e.g. Hinton (2007)). A number of "hidden" units form a model for the data distribution over the "visible" units. In a classification scenario one would fix the visible units and use the probability distribution over the hidden units for further processing. In a restricted Boltzmann machine the hidden units are independent and therefore the distribution over the hidden units in a classification scenario becomes a product distribution fully characterised by the average values of the individual hidden units. Pre-training using restricted Boltzmann machines has been used in early neural networks (Hinton et al. (2006)) but was abandoned as it became clear that good training can be achieved without it. Here we follow a slightly different path. We will use a Boltzmann machine to model image patches only. We do the modelling in a hierarchical fashion: train one Boltzmann machine for small image patches, then add more hidden units to model the joint distribution over larger patches. This forces us to depart from a simple restricted Boltzmann machine model and therefore we cannot obtain the hidden

probability distribution as easily as for a restricted Boltzmann machine. Applying mean field methods developed for the description of spin glasses in physics (Thouless et al. (1977); Georges and Yedidia (1991); Kühn and Helias (2017)) and applied to a restricted Boltzmann machine by Gabrié et al. (2015) and Budzianowski (2016), we can arrive at a description of the hidden units in terms of their average values, where now we replace the single neural network layer resulting from the restricted Boltzmann machine with a many-layer iteration derived from the mean field description. This pre-trained building block can be used as a fixed input layer in a neural network trained in the usual discriminatory fashion.

In summary the contributions of this article are as follows:

- We construct a new neural network building block, based on the mean field description of a Boltzmann machine. The result is a deep feed forward neural network with shared weights and a structure derived from the energy function of the underlying Boltzmann machine.
- We use this building block in a neural network to achieve state of the art resistance to adversarial attacks on MNIST and show a correlation between the loss function of the Boltzmann machine (i.e. it's generative capability) and the adversarial resistance of the resulting neural network.
- We show that the function mapping visible unit inputs to hidden unit outputs implemented by the new building block is strongly dependent on the input. A ball of random distortions around a training example gets mapped to a smaller output volume compared to the same mapping starting from a random input.

2 Boltzmann Machines as Feature Extractors

Let us review the the mean field description of a Boltzmann machine (Budzianowski (2016); Gabrié et al. (2015)). Boltzmann machines are stochastic, energy based models derived from the statistical physics of interacting spins. Boltzmann machines are a generative model, the generated probability distribution being the thermal equilibrium probability distribution over all possible spin combinations $\boldsymbol{s} = (s_1, s_2, \ldots, s_N)$ with $s_n = \pm 1$, as determined by the energy function through its parameters $\boldsymbol{\theta} = (\boldsymbol{b},\ \boldsymbol{J})$

$$E(\boldsymbol{\theta}, \boldsymbol{s}) = \boldsymbol{s} \cdot \boldsymbol{J} \cdot \boldsymbol{s} + \boldsymbol{b} \cdot \boldsymbol{s},$$

so that the probability of realising the configuration $\boldsymbol{s}$ is proportional to the Boltzmann factor

$$p(\boldsymbol{\theta}, \boldsymbol{s}) \propto e^{-E(\boldsymbol{\theta}, \boldsymbol{s})}.$$

The spins are divided into two sets, the visible units $\boldsymbol{v}$ and the hidden units $\boldsymbol{h}$. When training a Boltzmann machine we want to generate a target probability distribution $p_0(\boldsymbol{v})$ over the visible units, as defined by example configurations $\boldsymbol{v}_j^{(0)}$, $j = 1, \ldots, n_J$. This can be achieved by adjusting the parameters of the energy function to minimise the relative entropy between the marginal distribution over the visible units

$$p(\boldsymbol{v}, \boldsymbol{\theta}) = \mathrm{Tr}_{\boldsymbol{h}}\, p(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{\theta}).$$

and the target distribution:

$$D(p_0||p) = \sum_{j=1}^{n_J} p_0(\boldsymbol{v}_j^{(0)}) \log p_0(\boldsymbol{v}_j^{(0)}) - p_0(\boldsymbol{v}_j^{(0)}) \log p(\theta, \boldsymbol{v}_j^{(0)}). \tag{1}$$

The gradient of the relative entropy with respect to the parameters of the energy function θ can be expressed in terms of expectation values of spins

$$\frac{dD(p_0||p)}{db_k} = \sum_{j=1}^{n_J} \frac{p_0(\boldsymbol{v}_j^{(0)})}{p(\theta, \boldsymbol{v}_j^{(0)})} \left\langle s_k \right\rangle_{C(j)} - \left\langle s_k \right\rangle_F, \tag{2}$$

and spin-spin correlators

$$\frac{dD(p_0||p)}{dJ_{kl}} = \sum_{j=1}^{n_J} \frac{p_0(\boldsymbol{v}_j^{(0)})}{p(\theta, \boldsymbol{v}_j^{(0)})} \left\langle s_k s_l \right\rangle_{C(j)} - \left\langle s_k s_l \right\rangle_F, \tag{3}$$

where $\langle \ldots \rangle_F$ denotes the expectation value of the model without constraints (free) and $\langle \ldots \rangle_{C(j)}$ the expectation value if the visible units are fixed to example $\boldsymbol{v}_j$ (clamped). This gradient can be used in a first order optimisation method to arrive at an optimal parameter configuration θ_0. While the expression for the gradient looks simple, evaluating the probabilities and expectation values is not, and some approximation scheme needs to be used. Two schemes are usually used, one based on sampling (Hinton and E. (2002)), the other one based on a mean field approximation. Here we will concentrate on the mean field approximation.

The central quantity in mean field theory is the free energy. We can obtain the free energy of our model as an expansion in the order of the coupling J as derived by a number of authors (Georges and Yedidia (1991); Kühn and Helias (2017)).

$$\mathcal{F} = \sum_n F_n$$

To lowest order the free energy is given by the entropy of independent spins.

$$F_0 = \sum_{i=1}^{N} \left(\frac{1-m_i}{2}\right) \log\left(\frac{1-m_i}{2}\right) + \left(\frac{1+m_i}{2}\right) \log\left(\frac{1+m_i}{2}\right),$$

where m_i is the magnetisation of spin i. Up to second order we obtain

$$F_1 = \sum_{\langle ij \rangle} J_{ij} m_i m_j + \sum_i b_i m_i,$$

$$F_2 = \sum_{\langle ij \rangle} J_{ij}^2 \left(1 - m_i^2\right)\left(1 - m_j^2\right),$$

where the notation $\langle ij \rangle$ denotes summation over unequal indices. For orders up to four see appendix A. From the free energy we can derive an equation to obtain the mean field values m_i in the approximation up to order N

$$m_i = \tanh\left(\sum_{n=1}^{N} R_i^{(n)}\right) \tag{4}$$

with

$$R_i^{(n)} = \frac{dF_n}{dm_i}.$$

In particular, the first two orders are

$$R_i^{(1)} = b_i + 2\sum_j J_{ij} m_j$$

and

$$R_i^{(2)} = -4m_i \sum_j J_{ij}^2 (1 - m_j^2).$$

Higher orders can be found in the appendix A. In the free case, the transcendental equation, Eq. 4 can be solved by iteration, starting from zero or random magnetisations, until the magnetisations converge. In the clamped case we keep the visible magnetisations fixed, while iterating over the hidden units. This results in two sets of distinct magnetisations: the free magnetisations m_F and the clamped magnetisations $m_{C(j)}$, where example j is fixed on the visible units. We also get a set of corresponding free energies $\mathcal{F}_F$ and $\mathcal{F}_{C(j)}$. The correlation functions can be obtained from the free energies

$$\langle s_k s_l \rangle_{F/C(j)} = \frac{1}{2} \frac{d\mathcal{F}_{F/C(j)}}{dJ_{kl}},$$

and the expectation values are given directly by

$$\langle s_k \rangle_{F/C(j)} = m_{F/C(j),k}.$$

These expectation values can be inserted in the expressions for the gradient, Eqs. 2 and 3, and used to minimise the relative entropy Eq. 1. Probabilities can be estimated from the free energies

$$p(\theta, \boldsymbol{v}_j^{(0)}) = e^{\mathcal{F}_{C(j)} - \mathcal{F}_F}.$$

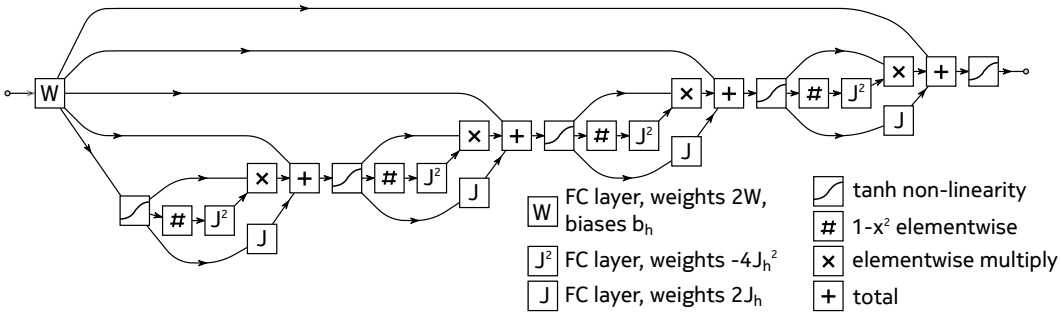

Figure 1: An example of an unrolling of the second order mean field iteration into a 5 layer neural network. The weights in all layers except the first one are shared, resulting in a deep network with a small number of parameters. The structure of the network is fully determined by the mean field equations derived from the energy function of Boltzmann machine.

The training of a Boltzmann machine emphasises the main task of the machine as generating examples from the target distribution, but one can look at the machine from a different point of view: using the hidden units for constructing a model of the example data. The probability distribution over the hidden units if an example is fixed on the visible units then tells us something about the model required to describe the example. In a restricted Boltzmann machine (RBM) the probability distribution is a product distribution over the hidden units and features, but one can imagine introducing couplings between hidden units to account for correlations between individual features to arrive at a better/simpler description of the data. In the remaining article we will discuss machines that have bipartite connectivity between visible and hidden units and full connectivity between hidden units.

During inference we want to obtain the hidden magnetisations given the magnetisations of the visible units. We split the magnetisations into visible and hidden parts $\boldsymbol{m} = (\boldsymbol{m}_v, \boldsymbol{m}_h)$. Similarly we split the magnetic fields $\boldsymbol{b} = (0, \boldsymbol{b}_h)$ and the couplings

$$\boldsymbol{J} = \begin{pmatrix} 0 & \boldsymbol{W} \\ \boldsymbol{W} & \boldsymbol{J}_h \end{pmatrix},$$

where $\boldsymbol{W}$ now describes the bipartite connectivity between visible and hidden units and $\boldsymbol{J}_h$ the coupling between hidden units. Keeping terms to second order the mean field equations become

$$\boldsymbol{m}_h = \tanh\left[2\boldsymbol{W} \cdot \boldsymbol{m}_v + \boldsymbol{b}_h + 2\boldsymbol{J}_h \cdot \boldsymbol{m}_h - 4\boldsymbol{m}_h \boldsymbol{J}_h^2 \cdot \left(1 - \boldsymbol{m}_h^2\right)\right]. \tag{5}$$

Solving the mean field equation Eq. 5 requires iteration, e.g. starting from the initial magnetisations $\boldsymbol{m}_h = \boldsymbol{0}$. The iteration can be unrolled into an r layer neural network, where r is greater than the number of iterations necessary for convergence (see Fig. 1). We will refer to this neural network as the mean field Boltzmann network.

3 Training the mean field Boltzmann network

The Boltzmann machine underlying the mean field Boltzmann network is trained on data derived from the standard MNIST training set. We binarize the MNIST images with a threshold (50/255) and create two datasets. The first dataset contains 4 x 4 patches cut from the binarised images, the second contains 8 x 8 patches. We start by training a Boltzmann machine for the 4 x 4 patches. Of all 2^{16} possible patches, 679 patches contain over 98% of all occurences in the dataset. We use these 679 to train a mean field Boltzmann machine with 32 hidden units (see Fig. 2). Four of these trained small machines are then combined with 128 hidden units to model the 8 x 8 patches. The dataset of 8 x 8 patches contains approximately 600000 examples. For the calculation of the gradient we use batches of 10000 examples. To calculate the free expectation values we use the mean field expansion up to fourth order, for the clamped expectation values it suffices to go to second order. The magnetisations in the free case are calculated using 100 mean-field iterations, for the clamped magnetisation we use 20 iterations.

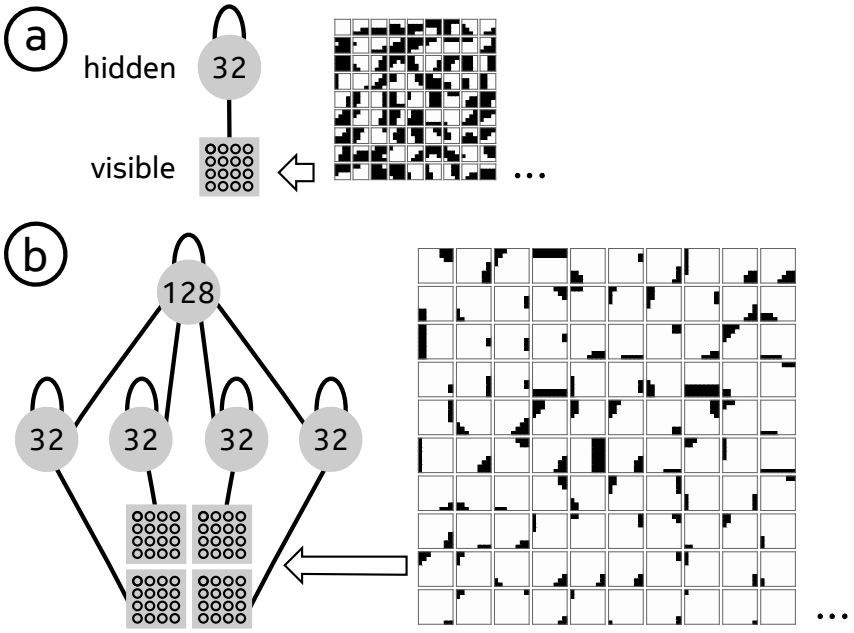

Figure 2: Illustration of the structure of the Boltzmann machine and the training procedure. a) shows the Boltzmann machine used in the first training step consisting of 16 visible units and 32 hidden units. The line between the visible and hidden units represents bipartite connectivity between the units in the two sets. The loop on top of the hidden units indicates full connectivity between hidden units. 4 x 4 patches from the example data are used to generatively train the machine. b) shows the structure of the Boltzmann machine used in the second training step, consisting of 64 visible units and 256 hidden units. The trained small machines are assigned a 4 x 4 sub-patch in the 8 x 8 patches. Each of the 32 hidden units in the small machines gains bipartite connectivity to a further fully connected 128 hidden units and the full machine is trained according to the standard Boltzmann machine training procedure.

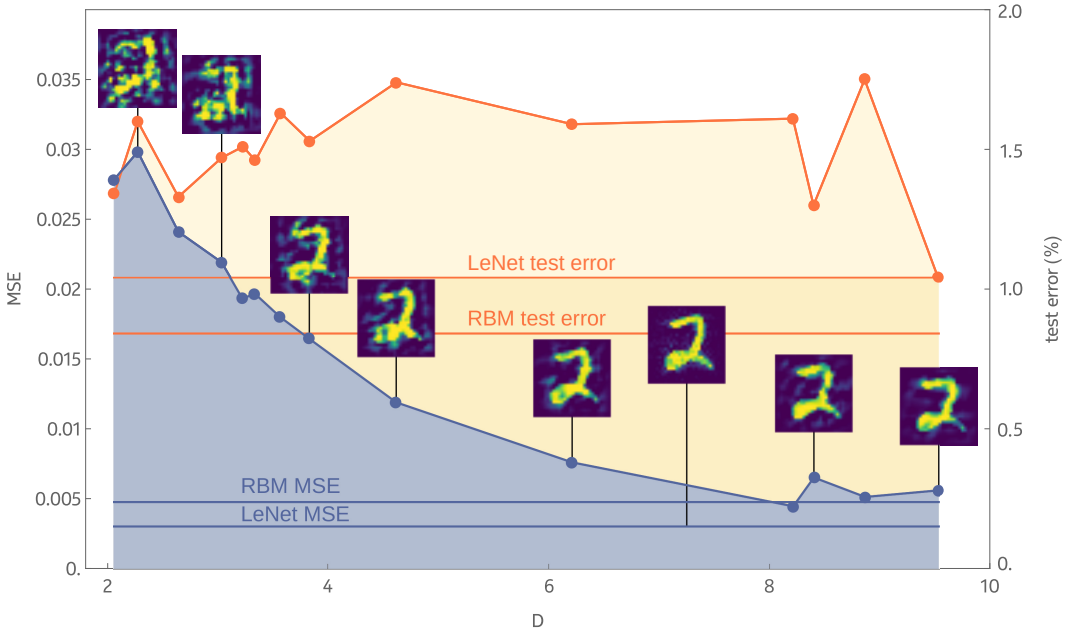

Figure 3: Dependence of adversarial resistance and test error on Boltzmann machine pre-training for a neural network with a mean field Boltzmann network input layer. The blue plot shows the mean square error (for image pixels in the range $[0, 1]$) required to create an adversarial image averaged over the first 500 members of the MNIST test set as a function of the relative entropy D achieved in Boltzmann machine pre-training. Lower relative entropy at the pre-training stage results in higher resistance to adversarial attacks. The blue lines indicate the adversarial resistance for LeNet and a pre-trained first RBM layer. Attached to the datapoints are examples of adversarial images at a given relative entropy. The orange line shows the test error (in %) of a mean field Boltzmann network over the whole clean MNIST test set, with the straight orange lines showing the reference test error of LeNet and a network with RBM pre training.

4 Adversarially Robust Neural Networks

The trained mean field Boltzmann network constitutes a new neural network building block that takes in 8 x 8 patches and outputs a vector of 128 hidden magnetisations. This building block has identical input/output dimensions to a standard convolutional layer with 8 x 8 filter size and 128 filters. We can use this building block as an input layer in an identical fashion to a standard convolutional layer, sweeping it over the two input image dimensions. Starting from the 28 x 28 MNIST images, this results in an intermediate representation of 128 x 21 x 21 dimensions. On top of this representation we build a standard three layer convolutional neural network with ReLU activations. We add 3 convolution layers (kernel size 3 x 3, stride 2) with 128 filters, 64 filters and 32 filters, respectively, followed by a fully connected layer and a softmax function). The input layer parameters are kept fixed during training and the remaining layers are trained in a standard discriminative fashion.

Let us establish the connection between better generative pre-training and increased adversarial resistance. We initialize the mean field Boltzmann network with parameters from various stages of the generative pre-training. As a measure for the generative training process we can use the relative entropy of the generated and target probability distribution, Eq. 1. To assess adversarial resistance we use an iterative gradient based method with L_2 constraints and record the mean squared error for the first image iteration that is misclassified. We average the mean squared error for the first 500 images in the MNIST test set. We also calculate the test error for the whole clean MNIST test set for a given neural network. The results are presented in Fig. 3. We see that a smaller relative entropy in the pre-training phase correlates with a higher adversarial resistance. Due to the stochasticity of the various training processes involved the relation is not monotonous. For comparison we apply the same procedure to a standard LeNet network as well as a neural network

Table 1: Comparison of our method with the ABS model. The entries in the table show the average L_2 norm and the robustness at a threshold of $\epsilon = 1.5$. (*) Our method is tested over the first 500 test images in MNIST.

	gradient based iterative method	Boundary Method
mean field Boltzmann network model	2.27 / 85 % (*)	8.2 / 93% (*)
Analysis by Synthesis model	3.1 / 87 %	2.6 / 83 %

where the mean field Boltzmann network is based on an RBM with 64 visible and 128 hidden units trained in the same mean field approximation. Both LeNet and RBM pre-trained network do not show the strong adversarial resistance observed in the neural network using the mean field Boltzmann network as input layer. On the other hand we see that the classification error of the robust neural network on the clean test set, while above the LeNet and RBM error, is independent of the pre-training. We find no evidence for a strong correlation of classification error and adversarial resistance in our model.

One reason for the apparent robustness of neural networks are obfuscated gradients (Athalye et al. (2018)). Therefore we also evaluate the robust network using a gradient free method: the Boundary Method (Brendel et al. (2017)), implemented in Foolbox. For evaluation we use the network that showed the highest adversarial resistance to the iterated gradient method. The Boundary Method, using 10000 iterations and the blended uniform noise attack as initial attack, fails to find adversarial images in 19 out of 500 examples from the MNIST test set. We calculate the mean square error over the remaining 481 examples and obtain an average MSE of 0.1. This is much higher than the gradient based method is able to obtain. For comparison we can use the same method on the LeNet architecture. Here we obtain a very small MSE of 0.0026, comparable to the result from the gradient based method. One could argue that the initial random search used in the boundary method is not efficient. To explore this possibility we consider a blended uniform noise attack (implemented in Foolbox) and increase the number of random directions from the standard setting of 1000 by factors of 10 up to 10^7. Over the sample of 500 test images we observe that the average MSE has no strong dependence on the number of random directions tried.

Another reason for the apparent robustness of our model could be the very deep (20 layers) pre-processing stage leading to vanishing gradients. To investigate this further we remove the last, fully connected layer in our neural network and train a second, shallower network with the 32 x 6 x 6 dimensional activation data. The distilled network consists of 4 convolution layers: (8 x 8, stride 1, 256 filters), (3 x 3, stride 2, 128 filters), (3 x 3, stride 2, 128 filters), (3 x 3, stride 1, 32 filters), with ReLU activations. Putting back the fully connected layer from the full network we can measure the adversarial resistance in the same way as earlier. We obtain an average MSE of 0.0142 and an error rate over the clean dataset of 0.66%, compared to an average MSE of 0.0298 for the original network. Although the distilled network is showing a considerable loss of robustness, the remaining robustness is still more than four times larger than the average LeNet robustness. This shows that some robustness can be transferred from the network structure used for learning to a relatively simple network and suggests that the robustness is not an artefact of a deep network structure.

A comparison of our results with currently the most adversarially robust method, the Analysis By Synthesis (ABS) model of Schott et al. (2018), is shown in Tab. 1. For results from other methods see table 1 in Schott et al. (2018). Our model achieves comparable robustness results.

5 Properties of Mean Field Boltzmann Networks

To understand the emerging adversarial robustness as a function of improved generative capability we examine the response of the mean field Boltzmann network around different input vectors. For one set of input vectors we will consider sample patterns from the training set, for the other set we select random binary patterns. We then add uniform random noise of amplitude ϵ to the input vectors and evaluate the different outputs of the network. The measure $\Delta_{r/t}$, presented in Fig. 4, is the L_2 norm of the average difference of clean and noisy output of a set of 1000 random and training set patterns respectively with 10000 random noise patterns for each input pattern. Training set patterns

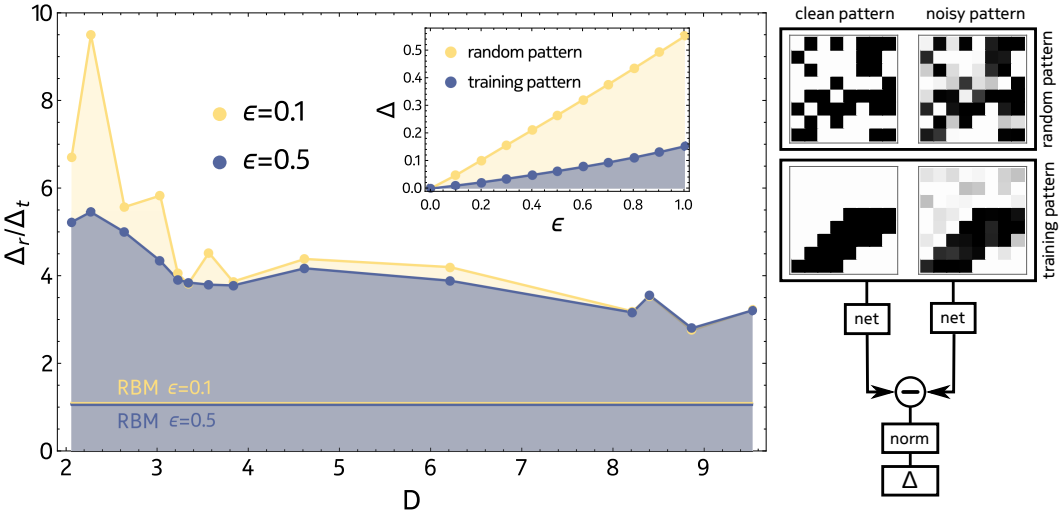

Figure 4: Response of a mean field Boltzmann network to noise at the input for different sets of input patterns. The inset shows how much adding random noise (uniformly distributed) of maximum amplitude ϵ to the input pattern changes the output pattern. The main plot shows the ratio of the differences for the random and training set patterns as a function of training progress of the Boltzmann machine as measured by the relative entropy.

are much more resistant to noise than random patterns and the resistance increases as the Boltzmann machine training progresses, particularly for small amplitude noise.

6 Discussion

We have shown that we can construct a feed forward neural network that shows strong adversarial resistance. This is achieved by incorporating a generatively pre-trained building block derived from the mean field description of a Boltzmann machine, which we called the mean field Boltzmann network. The resulting adversarial resistance strongly correlates with the effectiveness of the generative pre-training. We believe that the increased adversarial resistance can be traced to how the mean field Boltzmann network rejects noise around training example patches. The noise rejection does not happen for randomly selected examples. The rejection behaviour is an indication of the local non-linear behaviour of the mean field Boltzmann network. Compare this with the properties of a random matrix as feature extractor, where the restricted isometry property tells us that distances are preserved in the mapping with high probability. Too weak non-linearity of neural networks has been suggested as one reason for the existence of adversarial images and the increased non-linearity in our model correlates with increased adversarial resistance. On the level of individual image patches we can still find noise patterns that result in a large response when added to a training example, but they are much less likely. Take this together with the convolutional way that the mean field Boltzmann network is deployed. To get a large adversarial response from a given image region all overlapping image patches have to have a large response to the particular adversarial pattern. With the probability of large response supressed for one pattern, it becomes even smaller for all the overlapping pattern, resulting in strong adversarial resistance for the full neural network.

Our results, while encouraging, leave a number of questions open. First we have to ask ourselves if the success is due to the simplicity of the MNIST dataset. Will we be able to find typical image elements and model them with a Boltzmann machine for real image data? Second, while the adversarial images we find show a large deviation from the original image they do not resemble other image classes. The algorithm still confidently places them in a particular class. We would prefer a behaviour where images unrecognisable by a human would result in a low confidence of the prediction.

REFERENCES

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 274–283, 2018.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. doi: 10.1016/j.patcog.2018.07.023.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *CoRR*, abs/1712.04248, 2017.

Paweł Budzianowski. Training Restricted Boltzmann Machines Using High-Temperature Expansions. Master's thesis, University of Cambridge, 2016.

Marylou Gabrié, Eric W. Tramel, and Florent Krzakala. Training restricted boltzmann machines via the thouless-anderson-palmer free energy. *CoRR*, abs/1506.02914, 2015.

A Georges and J S Yedidia. How to expand around mean-field theory using high-temperature expansions. *J. Phys. A. Math. Gen.*, 24(9):2173–2192, 1991. ISSN 0305-4470. doi: 10.1088/0305-4470/24/9/024.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian J. Goodfellow. Adversarial spheres. *CoRR*, abs/1801.02774, 2018.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *CoRR*, abs/1412.5068, 2014.

Geoffrey Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007. ISSN 1941-6016. doi: 10.4249/scholarpedia.1668.

Geoffrey E. Hinton and Geoffrey E. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Comput.*, 14(8):1771–1800, 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.*, 18(7):1527–1554, 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527.

Saumya Jetley, Nicholas A. Lord, and Philip H. S. Torr. With friends like these, who needs adversaries? *CoRR*, abs/1807.04200, 2018.

Tobias Kühn and Moritz Helias. Expansion of the effective action around non-Gaussian theories. 2017. ISSN 1751-8113. doi: 10.1088/1751-8121/aad52e.

Alexey Kurakin, Ian J. Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan L. Yuille, Sangxia Huang, Yao Zhao, Yuzhe Zhao, Zhonglin Han, Junjiajia Long, Yerkebulan Berdibekov, Takuya Akiba, Seiya Tokui, and Motoki Abe. Adversarial attacks and defences competition. *CoRR*, abs/1804.00097, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 86–94, 2017. doi: 10.1109/CVPR.2017.17.

L. Schott, J. Rauber, W. Brendel, and M. Bethge. Towards the first adversarially robust neural network model on mnist. 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

D. J. Thouless, P. W. Anderson, and R. G. Palmer. Solution of 'Solvable model of a spin glass'. *Philos. Mag.*, 35(3):593–601, 1977. ISSN 0031-8086. doi: 10.1080/14786437708235992.

A HIGHER ORDER TERMS IN THE MEAN FIELD EXPANSION

The third and fourth order term in the free energy becomes

$$F_3 = \sum_{\langle ijk \rangle} J_{ij} J_{jk} J_{ki} \left(1 - m_i^2\right) \left(1 - m_j^2\right) \left(1 - m_k^2\right) + \frac{8}{3} \sum_{\langle ij \rangle} J_{ij}^3 m_i \left(1 - m_i^2\right) m_j \left(1 - m_j^2\right),$$

$$\begin{aligned} F_4 = 2 \sum_{\langle ijkl \rangle} & J_{ij} J_{jk} J_{kl} J_{li} \left(1 - m_i^2\right) \left(1 - m_j^2\right) \left(1 - m_k^2\right) \left(1 - m_l^2\right) + \\ & + 16 \sum_{\langle ijk \rangle} J_{ij}^2 J_{ik} J_{kj} m_i \left(1 - m_i^2\right) m_j \left(1 - m_j^2\right) \left(1 - m_k^2\right) \\ & - \frac{2}{3} \sum_{\langle ij \rangle} J_{ij}^4 \left(1 - m_i^2\right) \left(1 - m_j^2\right) \left(1 + 3m_i^2 + 3m_j^2 - 15 m_i^2 m_j^2\right) \end{aligned}$$

The third and fourth order terms for the mean field iteration are

$$R_i^{(3)} = -8 \sum_{\langle jk \rangle} J_{ij} J_{jk} J_{ki} m_i \left(1 - m_j^2\right) \left(1 - m_k^2\right) + \frac{2}{3} \sum_{j} J_{ij}^3 \left(1 - 3m_i^2\right) m_j \left(1 - m_j^2\right),$$

$$\begin{aligned} R_i^{(4)} = -16 \sum_{\langle jkl \rangle} & J_{ij} J_{jk} J_{kl} J_{li} m_i \left(1 - m_j^2\right) \left(1 - m_k^2\right) \left(1 - m_l^2\right) + \\ & - 32 \sum_{\langle jk \rangle} J_{kj}^2 J_{ij} J_{ki} m_i m_j \left(1 - m_j^2\right) m_k \left(1 - m_k^2\right) \\ & + 32 \sum_{\langle jk \rangle} J_{ij}^2 J_{kj} J_{ki} \left(1 - 3m_l^2\right) m_j \left(1 - m_j^2\right) \left(1 - m_k^2\right) \\ & - \frac{16}{3} \sum_{j} J_{ij}^4 m_i \left(1 - m_j^2\right) \left[1 - 3m_i^2 + 3m_j^2 \left(5m_i^2 - 3\right)\right] \end{aligned}$$