How Predictive Coding Rescues Feed-Forward Networks on Adversarial Attacks

Ehsan Ganjidoost (eganjidoost@uwaterloo.ca)
Jeff Orchard (jorchard@uwaterloo.ca)
Neurocognitive Computing Lab
Cheriton School of Computer Science
University of Waterloo, ON Canada

Abstract

This study introduces Predictive Coding Networks (PCnets) as a defence mechanism against adversarial attacks on neural network classifiers. By integrating PCnets into Feed-Forward Networks (FFnets), we enhance their resilience to adversarial perturbations. Using MNIST, we experimentally demonstrate the effectiveness of PCNets in identifying and mitigating adversarial examples generated to attack a fully-connected network, and a CNN. Leveraging the generative nature of PCnets, the defence mechanism effectively counters adversarial efforts, reverting perturbed images closer to their original forms. This innovative approach presents a promising solution for improving the security and reliability of neural network classifiers amidst the rising threat of adversarial attacks.

Keywords: Biological Plausibility, Classification, Gradient-Based Adversarial Examples, Predictive Coding

Introduction

Unlike humans, who robustly interpret visual stimuli, Neural Networks can be misled by Adversarial Attacks (ATs), specifically Perturbation attacks (Kumar, Brien, Albert, Viljöen, & Snover, 2019). These attacks subtly perturb an image x to fool a highly accurate Feed-Forward Network (FFnet), trained as a classifier (Biggio et al., 2013; Szegedy et al., 2013) (see figure 1). One common way to craft an Adversarial Example (AE) is to find a perturbation, δ, to minimize the loss function

$$\underset{\delta}{\operatorname{argmin}} \mathcal{L}(F(x+\delta), y_t), \tag{1}$$

where x represents an image, y_t is a 1-hot vector indicating an incorrect target class, $\mathcal{L}$ is the cross-entropy loss and F is mapping input image to the FFnet classifier's output. This optimization can be achieved iteratively, or in one step using the Fast Gradient Sign Method, FGSM (Goodfellow, Shlens, & Szegedy, 2014). Testing a trained FFnet MNIST classifier (accuracy $\approx 98\%$) against FGSM-generated AEs yields an adversarial success rate of about 41%. To defend against ATs, augmenting the training dataset with AEs can improve the classifier's resilience, achieving approximately 94% accuracy. Alternatively, a min-max approach to directly counteract AEs can enhance robustness within specific perturbation limits (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2017; Zhang et al., 2019).

Defensive strategies against ATs can also include generative mechanisms that can revert the perturbed images to their original form. Predictive coding provides a theoretical framework to support such a defence.

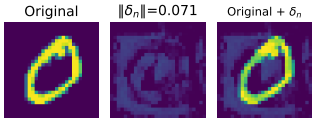

Figure 1: FFnet perceives the original $\Pr(y=0)=0.99$ while the perception changed to $\Pr(y=3)=0.87$ on perturbation.

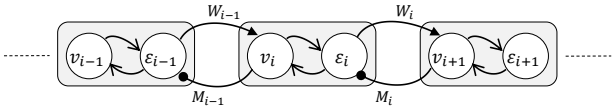

Figure 2: a typical PCnet arranged in a feed-forward manner. Each box represents a population of neurons containing value and error nodes.

Model Schema and The Learning Algorithm

Predictive Coding suggests the brain minimizes prediction error (Rao & Ballard, 1999). In a Predictive Coding Network (PCnet), unlike traditional neurons in Artificial Neural Networks (ANNs), each neuron, or *PC unit*, comprises a **value** (v) and **error** node (ε). PC units are collected into layers (similar to ANNs), forming PCnets that learn by adjusting predictions to minimize errors between layers. For instance, in a PCnet, layer i contains vectors v_i and ε_i, as shown in Figure 2. Vector v_i predicts the next layer's values v_{i-1} using prediction weights M_{i-1}. The resulting error, ε_{i-1}, is communicated back via correction weights W_{i-1}, allowing v_i to refine its predictions. The network dynamics are described by,

$$\tau\dot{\varepsilon}_i = v_i - M_i^T\sigma(v_{i+1}) - b_i - \xi\varepsilon_i \tag{2}$$

$$\tau\dot{v}_i = -\varepsilon_i + W_{i-1}^T\varepsilon_{i-1}\odot\sigma'(v_i) \tag{3}$$

$$\gamma\dot{M}_i = \varepsilon_i \otimes \sigma(v_{i+1}) \tag{4}$$

$$\gamma\dot{W}_i = \sigma(v_{i+1})\otimes\varepsilon_i \tag{5}$$

$$\gamma\dot{b}_i = \varepsilon_i \tag{6}$$

These include the activation function σ, Hadamard product $\odot$, outer product $\otimes$, decay coefficient ξ, and time constants τ and γ, where $\tau < \gamma$.

Training a PCnet involves clamping input-layer value nodes to target values and running the network to equilibrium. The network state (v_i, ε_i) reach equilibrium faster than the parameters (M_i, W_i, b_i), because $\tau < \gamma$. Post-training, parameters M, W, b are fixed, effectively setting γ to infinity.

A perfect prediction zeroes out the error signal (ε), stabilizing the value node without further corrections. This state

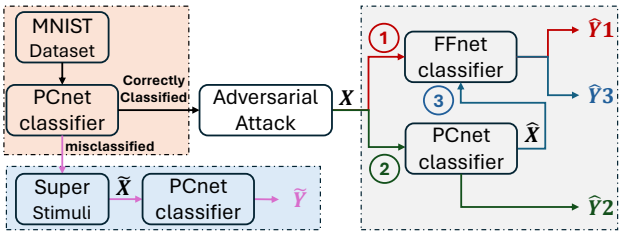

Figure 3: Work flow from pre-processing, creating AEs from X, and experiments on AEs and adjusted AEs $\hat{X}$, and creating superstimulus $\tilde{X}$ on misclassified by PCnet.

minimizes the Hopfield-like energy function (Bogacz, 2017),

$$E = \frac{\xi}{2} \sum_i \|\varepsilon_i\|^2. \tag{7}$$

Experiments

We conducted experiments to compare the performance of PCnet and FFnet classifiers on the MNIST dataset, employing both fully connected (FC) and CNN architectures for the FFnet. The experimental setup is illustrated in Figure 3.

The process begins with data partitioning into groups of correctly and incorrectly classified images, as indicated by the **brown box**. The AT module generates AEs from these classified images, targeting all possible labels (0 through 9).

These AEs, denoted X, are subsequently input into the FFnet and PCnet classifiers, producing outputs $\hat{Y}1$ and $\hat{Y}2$, respectively, depicted in the **gray box**. From that network state, the PCnet was run to equilibrium again, this time *unclamped*. At this new equilibrium, the adjusted image, $\hat{X}$, was re-evaluated by the FFnet to obtain final classification $\hat{Y}3$.

Lastly, as highlighted in the **blue box**, instead of AEs, where the target label matches the image's original label, the perturbed image transforms into a superstimulus, $\tilde{X}$, from which the PCnet outputs $\tilde{Y}$.

Results

We trained PCnet, FC and CNN models on the MNIST dataset with 76.82%, 92.8%, and 96.94% accuracy. We then partitioned the MNIST dataset based on the PCnet classifier. We generated AEs targeting nine incorrect labels per image using a gradient-based method. We collected successful AEs that fooled the FFnet, while PCnet correctly identified 76% of these cases (Fig. 4).

Notably, perturbing an image to create AEs using gradient ascent can lead to moving away from a local minimum. Conversely, the dynamical structure of PCnet guides the network's state towards lower energy, corresponding to an equilibrium. The network's energy decreases as its state moves toward equilibrium, as shown in Figure 5. The PCnet alters the AE and converges to an equilibrium consistent with its generative expectation.

Consequently, as shown in Figure 4, the FFnet recovered 83% of the cases on adjusted AEs $(\hat{X})$ compared to 0% initially. Moreover, in Figure 6, the red arrow shows how rapidly

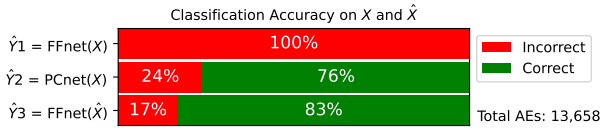

Figure 4: FC-based FFnet were fooled on 100% of AE cases, where PCnet recognized 76% of them. FFnet correctly recognized 83% of the cases adjusted by PCnet $\hat{X}$. The similar failure rates using CNN-based FFnet are 100%, 26%, and 5%.

the FFnet loses confidence on '0' and is fooled and misled to '3' as we perturb the image. Conversely, the green arrow shows how the changes in PCnet's state revert the AT process, and FFnet gains confidence for '0' after several steps.

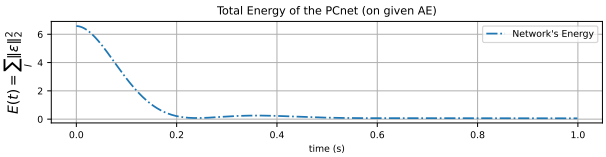

Figure 5: The network's energy drops as the AE changes through the network's dynamics in PCnet.

Summary

Integrating PCnets into FFnets presents a promising defence strategy against adversarial attacks on neural network classifiers. Guided by generative functionality, PCnets effectively revert adversarial perturbations, pushing images closer to their original forms. While demonstrating efficacy on the MNIST

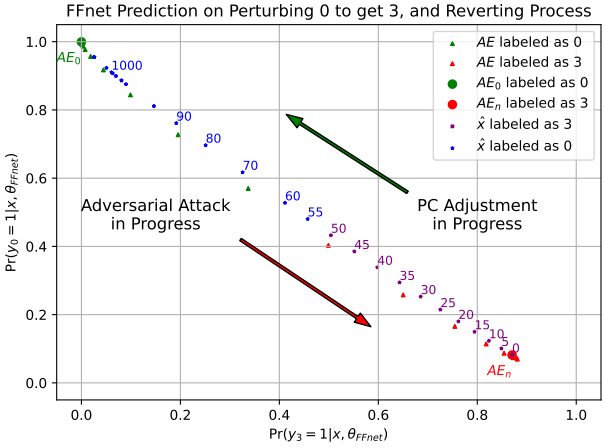

Figure 6: Red arrow illustrates FFnet misprediction as perturbed images transition from AE_0 to AE_n, mistakenly predicting '3' instead of '0'. Conversely, the green arrow depicts PCnet's iterative adjustments, enabling FFnet to classify AEs as '0' over milliseconds within a second correctly.

dataset with FC and CNN architectures, further research is needed to assess scalability, generalization, and robustness across diverse datasets and adversarial attack scenarios. Nonetheless, PCnets offer a significant step towards enhancing the security and reliability of neural network classifiers in the face of evolving adversarial threats.

References

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., ... Roli, F. (2013). Evasion attacks against machine learning at test time. In *Machine learning and knowledge discovery in databases: European conference, ecml pkdd 2013, prague, czech republic, september 23-27, 2013, proceedings, part iii 13* (pp. 387–402).

Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, *76*, 198–211.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Kumar, R. S. S., Brien, D. O., Albert, K., Viljöen, S., & Snover, J. (2019). Failure modes in machine learning systems. *arXiv preprint arXiv:1911.11034*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., & Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning* (pp. 7472–7482).