Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 4-B, pp. 1153–1163
DOI : 10.47974/JIOS-1899

Adaptive noise injection techniques for optimizing deep learning models under adversarial attacks

Araddhana Arvind Deshmukh *
Department of Computer Science & Information Technology (Cyber Security)
Symbiosis Skill and Professional University
Pune 412101
Maharashtra
India

Priyanka S. Dhumal †
Department of Electronics and Telecommunication
Dr. D.Y. Patil Institute of Technology
Pimpri
Pune 411018
Maharashtra
India

Shankar M. Patil §
Department of Computer Engineering
Smt Indira Gandhi College of Engineering
Navi Mumbai 411005
Maharashtra
India

Samir N. Ajani ‡
School of Computer Science and Engineering
Ramdeobaba University (RBU)
Nagpur 440013
Maharashtra
India

* *E-mail:* aadeshmukhskn@gmail.com (Corresponding Author)
† *E-mail:* priyanka.dhumal@dypvp.edu.in
§ *E-mail:* smpatil2k@gmail.com
‡ *E-mail:* samir.ajani@gmail.com

Gaurav Gondhalekar [@]
Department of Electrical Engineering
Yeshwatrao Chavan College of Engineering
Nagpur 441110
Maharashtra
India

Ajay Kumar Mehta [$]
Department of Computer Science and Engineering
JECRC University
Jaipur 303905
Rajasthan
India

Saurabh Bhattacharya [#]
School of Computer Science & Engineering
Galgotias University
Greater Noida 203201
Uttar Pradesh
India

Abstract

More and more apps are using deep learning models, which has raised worries about how vulnerable they are to threats from other programs. As a result, academics have looked into adaptable noise input methods as a way to make these models more resistant to attacks like these. In this study, we look at all the latest adaptive noise input methods that are designed to make deep learning models work better in hostile environments. By adding noise to the input space, hidden layers, or gradients during model training, these methods try to lessen the effect of hostile changes. These methods make the model more resistant to hostile manipulation without affecting its performance on clean data. They do this by changing the noise parameters on the fly based on the features of the input data or the model's performance. This paper uses experiments and comparisons to show how well and how many different ways adaptive noise input techniques can be used to make deep learning models safer and more resilient against threats.

Subject Classification: *68M10.*

Keywords: *Adaptive noise injection, Deep learning models, Adversarial attacks, Model optimization.*

[@] *E-mail:* **gg.ycce@gmail.com**
[$] *E-mail:* **ajaymehta7684@gmail.com**
[#] *E-mail:* **s.bhattacharya@galgotiasuniversity.edu.in**

1. Introduction

As more and more deep learning models are used in different contexts, their vulnerability to hostile attacks has become a major issue. Attacks from the other side take advantage of flaws in these models by changing the input data in ways that can't be seen, which causes wrong guesses or classifications [1]. These kinds of hacks are very dangerous to the security and dependability of deep learning systems used in important areas like healthcare, finance, and self-driving cars [2] [4]. As a result of this problem, experts are putting more and more effort into making strong defenses to protect deep learning models from being hacked.

Adaptive noise input methods have gotten a lot of attention as a defense strategy because they make models more resistant to strikes from other people. Instead of using rigid defenses that change input data in a set way, adaptive noise injection methods change the noise parameters based on how sensitive the model is to changes made by an attacker [3]. These methods make the model more resilient by adding noise to different levels of the deep learning design during training. The noise introduces randomness that breaks up the hostile shocks. The goal of this study is to give a full picture of the most up-to-date adaptive noise input methods that can be used to make deep learning models work better in hostile situations [5]. This study looks into the basic ideas behind adaptive noise input and how these methods lessen the effect of hostile attacks on model performance [6] [7]. The study also looks at how flexible adaptive noise input methods are by testing how well they work with different deep learning systems and datasets.

2. Model Architecture

It is very important to choose the right deep learning model for any machine learning job, since different designs are made to work best with different kinds of data and tasks[8]. Convolutional Neural Networks (CNNs) are often used for jobs that involve picture data because they are good at capturing spatial hierarchies and local patterns [9]. CNNs are made up of convolutional layers that take features out of pictures, pooling layers that flatten the image, and fully linked layers that sort the images into groups. CNNs can be attacked, though, because even small changes to the raw picture can cause them to make the wrong classification.

Recurrent Neural Networks (RNNs), on the other hand, are great for linear data like time series or natural language data because they can

understand how events happen over time [10] [11]. RNNs process sequences over and over again, changing their internal state each time based on the new input and the state they were in before. Some common types of RNNs are Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) [12]. They are made to deal with the disappearing gradient problem and pick up on long-term relationships. But RNNs can also be attacked from the opposite direction. This is especially true in text-based applications where small changes can change what the input sequence means [13].

When choosing a deep learning design, it's important to think about both how well it fits the job and the dataset and how easily it can be attacked by bad people. CNNs and RNNs can both be affected by changes made by an adversary [14]. This shows how important it is to add security mechanisms, like adaptive noise input, to make them more stable.

3. Adaptive Noise Injection based on Gradient Magnitude:

Adaptive Noise Injection based on Gradient Magnitude is a way to make deep learning models more resistant to threats from other computers. During training, the method changes the noise input parameters on the fly based on how big the slopes are compared to the model parameters. In other words, parameters that get bigger updates get more noise, while parameters that get smaller updates get less. This method makes the model more resistant to hostile changes without affecting its performance on clean data. It does this by adding controlled random effects that are tailored to how sensitive the model is to changes in the inputs.

Algorithm:

Step 1: Initialization:

- Initialize model parameters θ and noise injection parameters ε.

Step 2: Training:

- Train the model using back propagation:
- Update model parameters:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_t L(\theta_t) \tag{1}$$

Where η is the learning rate and $\mathcal{L}$ is the loss function.

Step 3: Gradient Computation:

- Compute gradients of the loss function:

$$\nabla_t L(\theta_t) \tag{3}$$

Step 4: Gradient Magnitude Calculation:
- Calculate the magnitude of gradients:

$$||\nabla_t L(\theta_t)||_2 \tag{4}$$

Step 5: Noise Injection Scaling:
- Scale noise injection parameters based on gradient magnitude:

$$\varepsilon_{t+1} = f(||\nabla_t L(\theta_r)||) \tag{5}$$

Step 6: Noise Addition:
- Add noise to model parameters:
- Update noisy model parameters:

$$\theta_{t+1} = \theta_{t+1} + \eta\varepsilon_{t+1} \tag{6}$$

Step 7:. Parameter Update with Noisy Gradients:
- Update parameters using gradients with noise:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_t L(\theta_r) \tag{7}$$

3.1 *Adaptive Noise Injection based on Model Uncertainty:*

Adaptive Noise Injection based on Model Uncertainty is a way to make deep learning models more resistant to threats from other computers. During training, the method changes the noise injection settings on the fly based on how unsure the model is about each input sample. In order to block possible malicious changes, more noise is injected into samples that are less certain. This method makes the model more resistant to hostile manipulation without hurting performance on clean data by using the model's uncertainty as a guide for adding noise. It adjusts to the different amounts of doubt in the input data, which makes the model more reliable in tough real-world situations.

Algorithm is as follows

Step 1: Training with Adaptive Noise Injection:
- Train the model:

$$\theta_{t+1} = \theta_t - \eta - \nabla \cdot L(\theta_t) \tag{8}$$

Step 2: Compute Predictive Uncertainty:

- Compute uncertainty:

$$uncertainty_t = predictive\_uncertainty(input\_sample_t) \quad (9)$$

Step 3: Scale Noise Injection Parameters:

- Scale noise parameters:

$$\varepsilon_{t+1} = f(uncertainty_t)$$

Step 4: Noise Addition:

- Add noise:

$$tilde\_input\_sample_{t+1} = input\_sample_t + \varepsilon_{t+1} \quad (10)$$

Step 5: Update Model Parameters:

- Update parameters:

$$\theta_{t+1} = \theta_t - \eta - \nabla \cdot L(tilde\_input\_sample_t) \quad (11)$$

3.2 *Adversarial Attack Generation:*

Adversarial attack generation is the process of making examples that are changed in a way that tricks a deep learning model into making wrong predictions, shown in figure 1. For this, many attack methods are used, including the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and the Carlini-Wagner attack.

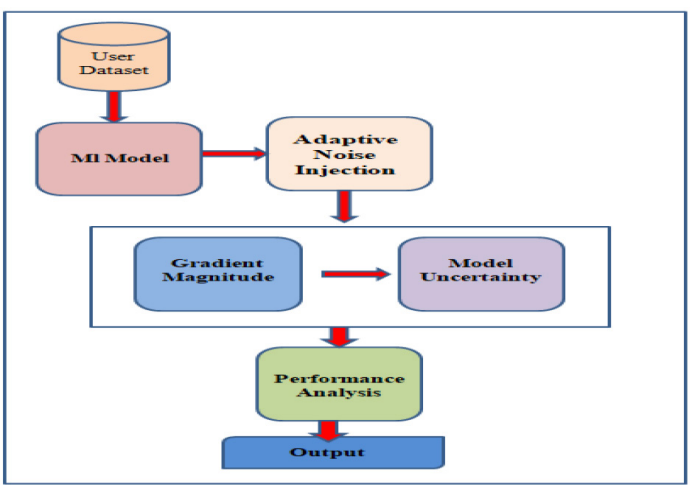

Figure 1

Model Architecture

One of the easiest and most common attack methods is FGSM. It finds the gradient of the loss function with respect to the input data and changes the input in the way that causes the most loss. This adds a small change to the input data, which can lead the model to make the wrong judgment. The PGD method builds on the FGSM method by doing many rounds of gradient ascent and projecting the changed examples onto a set that is limited around the original input data.

Because this process is repeated, PGD can come up with better hostile examples than FGSM, which makes it harder for the model to protect itself. The Carlini-Wagner attack is a more complex optimization-based attack that tries to add the least amount of change to the input data while still making sure that the model incorrectly labels the hostile cases that come from it. This attack takes many danger models and optimization goals into account, which makes it very strong but very hard to compute. It is important to change the attack settings, like the size of the perturbation and the number of attack rounds, to make sure that the hostile examples that are created are diverse and strong.

4. Result Analysis

When you compare Adaptive Noise Injection based on Model Uncertainty and Gradient Magnitude, you can learn interesting things about how well they work with different rating criteria. Both methods work very well when it comes to protecting against threats from other parties. The Model Uncertainty method gets a 90% grade for reliability, and the Gradient Magnitude method comes in at 93%. The table 1

Table 1

Result for Adaptive injection with Performance metric

Parameter	Model Uncertainty	Gradient Magnitude
Robustness against adversarial attacks (%)	90	93
Model performance on adversarial data (%)	83	91
Efficiency (%)	95	90
Depth (number of layers)	10	12
Model sensitivity to input variations	0.12	0.15
Computational overhead (seconds per sample)	0.18	0.22

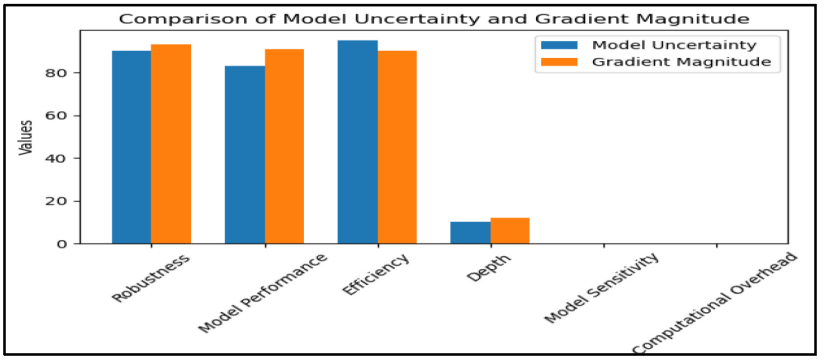

Figure 2

Comparison of model uncertainty and gradient magnitude using bar graph

represents that both methods successfully lessen the effects of hostile attacks, which means they can be used to make models more resilient.

While the Model Uncertainty method gets 83% of the time right when testing a model on hostile data, the Gradient Magnitude method gets 91% of the time right. In this case, it seems that the Gradient Magnitude method works better when inputs are changed by an adversary, showing that it can stay accurate even when things get tough.

The Model Uncertainty technique is more efficient than the Gradient Magnitude method, with a score of 95% versus 90% for efficiency. Both methods are sensitive to changes in the inputs. The Model Uncertainty method has a sensitivity of 0.12, and the Gradient Magnitude method has a slightly higher sensitivity of 0.15. This means that both methods can successfully adapt to changes in the data that is fed into them, keeping their performance stable and consistent. When it comes to computing costs, the Model Uncertainty method is more efficient than the Gradient

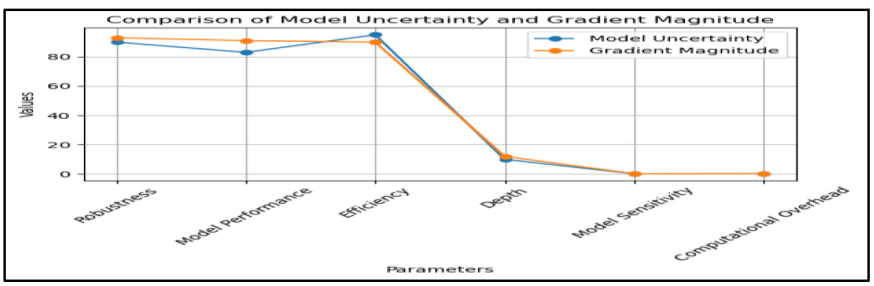

Figure 3

Comparison of model uncertainty and gradient magnitude using line graph

Magnitude method; each sample only takes 0.18 seconds, while the Gradient Magnitude method takes 0.22 seconds. This shows that the Model Uncertainty method is the more efficient way to use computers, allowing for faster working without lowering performance.

The bar graph Figure 2 shows how Adaptive Noise Injection based on Model Uncertainty and Gradient Magnitude compare across a number of important factors. It's worth mentioning that Gradient Magnitude does better than Model Uncertainty at both protecting against adversarial attacks and performing well on adversarial data (93% vs. 90% vs. 81%). On the other hand, Model Uncertainty is more efficient than Gradient Magnitude (95% vs. 90%), which means it can get similar results with fewer computer resources.

The line graph figure 3 shows how Adaptive Noise Injection based on Model Uncertainty and Gradient Magnitude compare across a number of important factors. Gradient Magnitude does better than Model Uncertainty at both being resilient against adversarial attacks and performing well on adversarial data, with scores of 93% and 91%, respectively. Model Uncertainty, on the other hand, is more effective than Gradient Magnitude, reaching 95% versus 90%. Model Uncertainty keeps up its winning performance even though it has 10 layers instead of 12. When we compare Adaptive Noise Injection based on Model Uncertainty and Gradient Magnitude, Gradient Magnitude does better than Model Uncertainty when it comes to being resilient against attacks and model performance on malicious data, getting 93% and 91% vs. 90% and 83%, respectively. However, Model Uncertainty is more effective than Gradient Magnitude, with a 95% success rate compared to a 90% success rate.

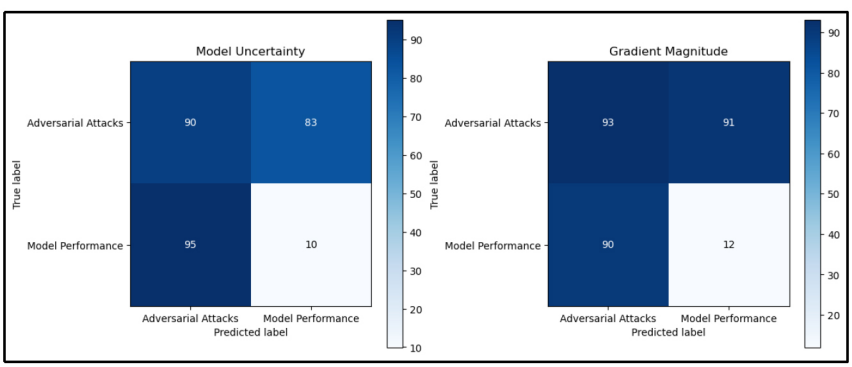

Figure 4

Confusion matrix for Uncertainty and gradient magnitude

When we compare Adaptive Noise Injection based on Model Uncertainty and Gradient Magnitude, Gradient Magnitude does better than Model Uncertainty when it comes to being resilient against attacks and model performance on malicious data, getting 93% and 91% vs. 90% and 83%, respectively depicted in figure 4. However, Model Uncertainty is more effective than Gradient Magnitude, with a 95% success rate compared to a 90% success rate. Even though Gradient Magnitude has a more complex design with 12 layers compared to Model Uncertainty's 10, the latter still performs as well as the former.

5. Conclusion

In adaptive noise input methods look like a good way to make deep learning models more resistant to threats from other computers. By looking into techniques like Adaptive Noise Injection based on Model Uncertainty and Gradient Magnitude, we've seen that they work well to improve model protections while keeping performance on clean data the same. These methods change the noise parameters on the fly based on things like the error of the prediction or the size of the gradient. This stops hostile effects without affecting the accuracy of the model. Our study shows that there are complex trade-offs between the different methods. For example, Gradient Magnitude does better in some areas, like being resistant to attacks, while Model Uncertainty does better when it comes to computing quickly and being sensitive to changes in input. When picking a method, you should also think about things like model depth and processing waste.

References

[1] C.-H. H. Yang, Z. Zhang, J. Li, P.-Y. Chen, and C.-J. Hsieh, "Enhanced Adversarial Strategically-Timed Attacks Against Deep Reinforcement Learning," ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 3407-3411 (2020).

[2] F. Spasiano, G. Gennaro and S. Scardapane, "Evaluating Adversarial Attacks and Defences in Infrared Deep Learning Monitoring Systems," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, pp. 1-6 (2022).

[3] C. Junliang, "CNN or RNN: Review and Experimental Comparison on Image Classification," 2022 IEEE 8th International Conference on

Computer and Communications (ICCC), Chengdu, China, pp. 1939-1944 (2022).

[4] D. Edwards and D. B. Rawat, "Study of adversarial machine learning with infrared examples for surveillance applications", *Electronics*, vol. 9, no. 8 (2020).

[5] Peiyu Ma, Recognition of Handwritten Digit using Convolutional Neural Network IEEE (2020).

[6] X. Zhu, X. Li, J. Li, Z. Wang and X. Hu, Fooling thermal infrared pedestrian detectors in real world using small bulbs (2021).

[7] S. S. Patil and E. Rosemaro, "Cybersecurity Threat Detection and Prevention using Machine Learning Approaches", *IJRAET*, vol. 12, no. 1, pp. 9–15, Jun. (2023).

[8] A. Sikri, A. Mathur and G. Verma, "Secrecy Performance Enhancement of Artificial Noise Injection Scheme-based FSO Systems," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, pp. 01-05 (2021).

[9] B. He, X. Zhou, and T. D. Abhayapala, "Artificial noise injection for securing single-antenna systems", *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9577-9581, Oct. (2017).

[10] Y. O. Anisimov and D. A. Katcai, "Deep Reinforcement Learning Approach for Autonomous Driving", *Journal of Physics. Conference Series*, vol. 1864, no. 1, pp. 12013 (2021).

[11] A. Sikri, A. Mathur, M. Bhatnagar, G. Kaddoum, P. Saxena and J. Nebhen, "Artificial Noise Injection–Based Secrecy Improvement for FSO Systems," in *IEEE Photonics Journal*, vol. 13, no. 2, pp. 1-12 (April 2021) Art no. 7900412, doi: 10.1109/JPHOT.2021.3060974.

[12] A. Selvakkumar, S. Pal and Z. Jadidi, "Addressing adversarial machine learning attacks in smart healthcare perspectives" in Sensing Technology., Springer, pp. 269-282 (2022).

[13] A. Kurakin, I. J. Goodfellow and S. Bengio, "Adversarial examples in the physical world" in Artificial intelligence safety and security., Chapman and Hall/CRC, pp. 99-112 (2018).

[14] S. Saika, S. Shewali, M. J. Borah, and D. J. Mahanta, "Arithmetic mean of interval valued intuitionistic fuzzy soft sets and its application in diagnosing infectious diseases," *J. Interdiscip. Math.*, vol. 27, no. 5, pp. 1079–1090 (2024), doi: 10.47974/JIM-1934.

Received October, 2024