Evaluating the Performance of the Adam Optimizer

Rayyan Al-Haj
Department of Artificial Intelligence
Bahcesehir University
Istanbul, Turkey
rayyanalhaj@yahoo.com

Omnia Elmenshawy
Department of Artificial Intelligence
Bahcesehir University
Istanbul, Turkey
Omnia.elmenshawy22@gmail.com

Abstract

This report investigates the application of the Adam optimizer in various machine learning model contexts, emphasizing its versatile performance in model optimization and robustness in adversarial situations. This report presents a comparative evaluation of different neural network architectures – Convolutional Neural Networks (ConvNets), Multi-Layer Networks, and Logistic Regression – utilizing the Adam optimizer, an innovative stochastic optimization method proposed by Kingma and Ba in 2014. The second task in this report used Adam to implement a Deep Feature Space Trojan (DFST) Attack on two Convolutional Neural Network (CNN) models trained on the CIFAR10 dataset. The results underline the importance of the Adam optimizer in deep learning tasks and highlight the necessity of model security against adversarial attacks. The complete code for this project, including all data processing, model training, and analysis scripts, is available at the following GitHub repository: https://github.com/omniaelmenshawy/StochasticOptimizationUsingAdam.

1 Introdcution

The first part explores the efficacy of Adam optimizer across three different neural network architectures— Convolutional Neural Networks (ConvNets), Multi-Layer Networks, and Logistic Regression. The models are trained and tested on the MNIST dataset, which consists of handwritten digits. Several hyperparameters for the Adam optimizer are tweaked to gauge their impact on model performance. The second task of our study was motivated by this pressing concern. It entailed conducting an adversarial attack on two pre-existing Convolutional Neural Network (CNN) models trained on the CIFAR10 dataset, an extensively used benchmark dataset in the machine learning community.

2 Methodologies

2.0.1 Task 1: Evaluating Adam Optimizer and Conducting Sensitivity Analysis

The initial phase of this study was devoted to evaluating the performance of the Adam optimizer within three distinct machine learning models, namely Logistic Regression, Multi-layer Neural Network, and Convolutional Neural Network.

The Adam optimizer demonstrated an effective capability for binary classification within the Logistic Regression model. Concurrently, its prowess in managing intricate neural network structures within the Multi-layer Neural Network model was likewise commendable. The Convolutional Neural Network model also benefitted significantly from the optimizer in terms of efficient extraction of high-level features.

After model training, a sensitivity analysis was executed on the hyperparameters of Adam: weight decay, Beta1, Beta2, and Epsilon. The results of this analysis highlighted the substantial influence

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

these hyperparameters exert on the optimizer's performance, thus emphasizing the criticality of parameter tuning for optimal results.

Implementation was achieved through the PyTorch library, which served as the framework for the neural network models and the Adam optimizer. Data was procured from the MNIST dataset, which was duly loaded and preprocessed for compatibility with the models.

The neural network models were categorized into: 1. Logistic Regression: A single-layer neural network characterized by a lone linear layer. 2. Multi-Layer Network: This model incorporated two hidden layers of 256 units each, ReLU (Rectified Linear Units) activation functions, and a 0.5 dropout rate for regularization. 3. Convolutional Neural Network (ConvNet): This model was composed of two convolutional layers, each succeeded by a batch normalization layer and a max-pooling layer, and two fully connected layers.

For each model, the Adam optimizer was deployed with varying hyperparameters (weight decay, Beta1, Beta2, and Epsilon) within a predefined range to assess their influence on the model's performance. A cross-entropy loss function was utilized as the criterion for the optimizer.

Training for all models was carried out for ten epochs, with the optimizer and criterion defined previously. Post training, models were tested and their training and testing loss, as well as accuracy, were documented after each epoch.

A sensitivity analysis was performed to assess the model's performance, encompassing the confidence level of the model's correct and incorrect predictions. An error analysis was also conducted to identify instances of incorrect predictions. This process was repeated for each combination of hyperparameters for each model, with the results stored for future reference

2.0.2 Task 2: Deploying Adam Optimizer in Deep Feature Space Trojan Attack Execution

Adversarial attacks pose significant challenges within the sphere of machine learning and artificial intelligence. The second task of our study is devoted to investigating these attacks, specifically, the Deep Feature Space Trojan (DFST) attack, on two Convolutional Neural Network (CNN) models trained on the CIFAR10 dataset.

Two CNN models served as targets for this task: a custom model developed internally with an initial training accuracy of 73.96%, and a pre-trained ResNet30 model boasting a testing accuracy of 84.62%. Both models were trained on the CIFAR10 dataset, which comprises 60,000 32x32 color images categorized into 10 classes.

The DFST attack was executed employing the Adam Optimizer to manage the CycleGAN model's learning rate. This generative adversarial network was tasked with creating the trigger to distort the targeted CNN models. In executing the attack, a poisoning ratio of 10% was applied, representing a realistic scenario wherein an attacker lacks total control over the training data.

3 Results

3.0.1 Task 1: Performance Evaluation and Sensitivity Analysis of Adam Optimizer

The Convolutional Neural Network (ConvNet) emerged as the most accurate model, achieving an average test accuracy of 99.11% across all tested hyperparameters. The Multi-Layer Network and Logistic Regression models trailed with maximum average test accuracies of 98.31% and 92.64%, respectively.

The superior accuracy of the ConvNet model does come at the cost of increased computational complexity, and thus, resource considerations may justify the use of the simpler models even at lower accuracy levels. It was also noted that the performance of all models was sensitive to the Adam optimizer parameters, implying potential performance improvements through further fine-tuning.This intricate interplay of factors reemphasizes the importance of hyperparameter tuning and the necessity for task-specific and model-specific considerations when deploying the Adam optimizer.

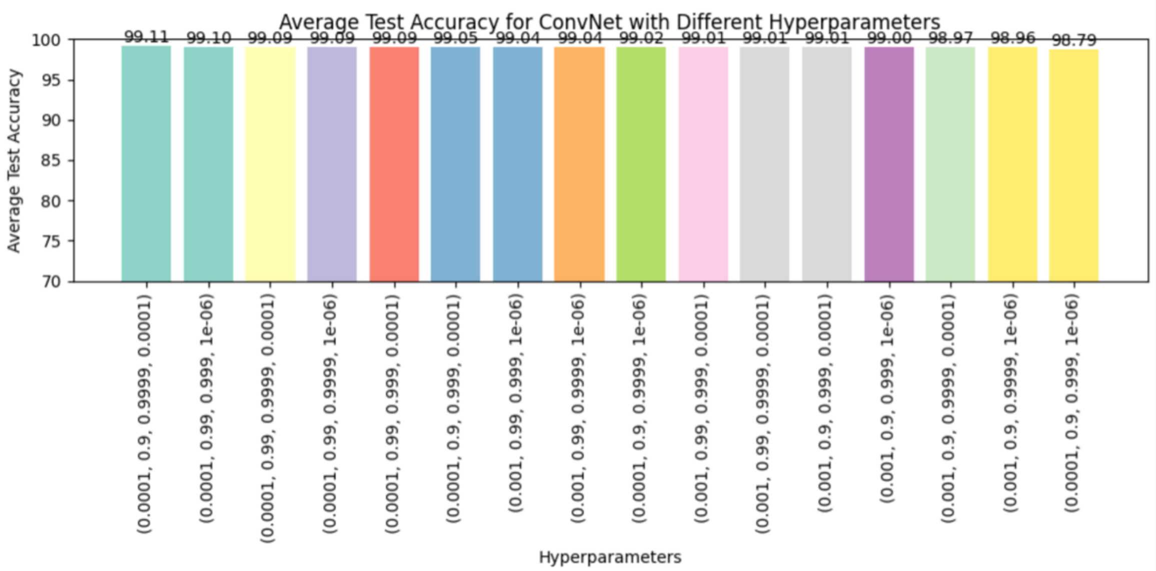
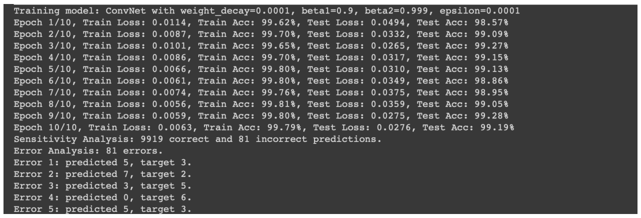


Figure 1: results visualizing

3.0.2 Task 2: Execution of Deep Feature Space Trojan Attack using Adam Optimizer

Post-execution of the DFST attack, a marked performance degradation was observed in the two targeted models. The internally developed custom model saw its accuracy drop from a pre-attack level of 73.96% to 62.34% post-attack. Similarly, the ResNet30 model's accuracy fell from 84.62% to 69.53% post-attack.

These outcomes highlight the vulnerabilities inherent in machine learning models vis-à-vis adversarial attacks, and simultaneously, underscore the efficacy of the Adam optimizer in facilitating such attacks. This knowledge provides a crucial steppingstone for future research aimed at developing more robust models and countermeasures against adversarial attacks.



```
Training model: ConvNet with weight_decay=0.0001, beta1=0.9, beta2=0.999, epsilon=0.0001
Epoch 1/10, Train Loss: 0.0114, Train Acc: 99.62%, Test Loss: 0.0494, Test Acc: 98.57%
Epoch 2/10, Train Loss: 0.0087, Train Acc: 99.70%, Test Loss: 0.0332, Test Acc: 99.09%
Epoch 3/10, Train Loss: 0.0101, Train Acc: 99.65%, Test Loss: 0.0265, Test Acc: 99.27%
Epoch 4/10, Train Loss: 0.0086, Train Acc: 99.70%, Test Loss: 0.0317, Test Acc: 99.15%
Epoch 5/10, Train Loss: 0.0066, Train Acc: 99.80%, Test Loss: 0.0310, Test Acc: 99.13%
Epoch 6/10, Train Loss: 0.0061, Train Acc: 99.80%, Test Loss: 0.0349, Test Acc: 98.86%
Epoch 7/10, Train Loss: 0.0074, Train Acc: 99.76%, Test Loss: 0.0375, Test Acc: 98.95%
Epoch 8/10, Train Loss: 0.0056, Train Acc: 99.81%, Test Loss: 0.0359, Test Acc: 99.05%
Epoch 9/10, Train Loss: 0.0059, Train Acc: 99.80%, Test Loss: 0.0275, Test Acc: 99.28%
Epoch 10/10, Train Loss: 0.0063, Train Acc: 99.79%, Test Loss: 0.0276, Test Acc: 99.19%
Sensitivity Analysis: 9919 correct and 81 incorrect predictions.
Error Analysis: 81 errors.
Error 1: predicted 5, target 3.
Error 2: predicted 7, target 2.
Error 3: predicted 3, target 5.
Error 4: predicted 0, target 6.
Error 5: predicted 5, target 3.
```

Figure 2: results and predictions

4 Conclusion

A significant component of our first task was a sensitivity analysis on Adam's hyperparameters – weight decay, beta values, and epsilon. The examination demonstrated the sensitive interplay between these hyperparameters and their influence on the learning process of the models. We observed that minor alterations to these parameters led to measurable performance differences, highlighting their critical role in the optimization process. The results from this analysis will be instrumental in refining optimization techniques in the future.

In the second task, we ventured into a more adversarial domain, employing the Adam optimizer in a Deep Feature Space Trojan (DFST) attack against two CNN models. Here, Adam was used in the CycleGAN architecture to generate an adversarial trigger. The significant decline in model accuracy following the attack demonstrated Adam's effectiveness even in adversarial settings, and underlined the potential vulnerabilities of deep learning models to such malicious attacks.

To sum up, our research has underscored the diverse applicability and robustness of the Adam optimizer in both benign and adversarial contexts. The study has revealed Adam's ability to enhance the performance of multiple models through optimized learning rates, while also demonstrating its effectiveness in adversarial machine learning. The insights from this study highlight the importance of understanding the role of hyperparameters in optimization and emphasize the urgent need for developing defenses against adversarial attacks. These findings open doors to more extensive investigations into fine-tuning model performance and securing models against potential threats.

5 References

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Cheng, S., Liu, Y., Ma, S., & Zhang, X. (2020). Deep Feature Space Trojan Attack of Neural Networks by Controlled Detoxification. ArXiv.

Elmenshawy O. & Alhaj R.(2023). Stochastic Optimization Using Adam. GitHub `https://github.com/omniaelmenshawy/StochasticOptimizationUsingAdam`