A.I. Seminar Project
Defending Against Adversarial Examples

Fadji OHOUKOH

Côte d'Azur University

Rémi BERNHARD

In the context of adversarial attacks on machine learning models, a defense scheme is a strategy or method used to protect the model from these attacks. The goal of a defense scheme is to make the model robust against adversarial examples, which are input data designed to mislead the model into making incorrect predictions. Here we consider The White-Box setting and the Black-box setting with SPSA attack and adversarial examples transferred from another model.

Simplifying the model to adress complexity leading to overfitting.

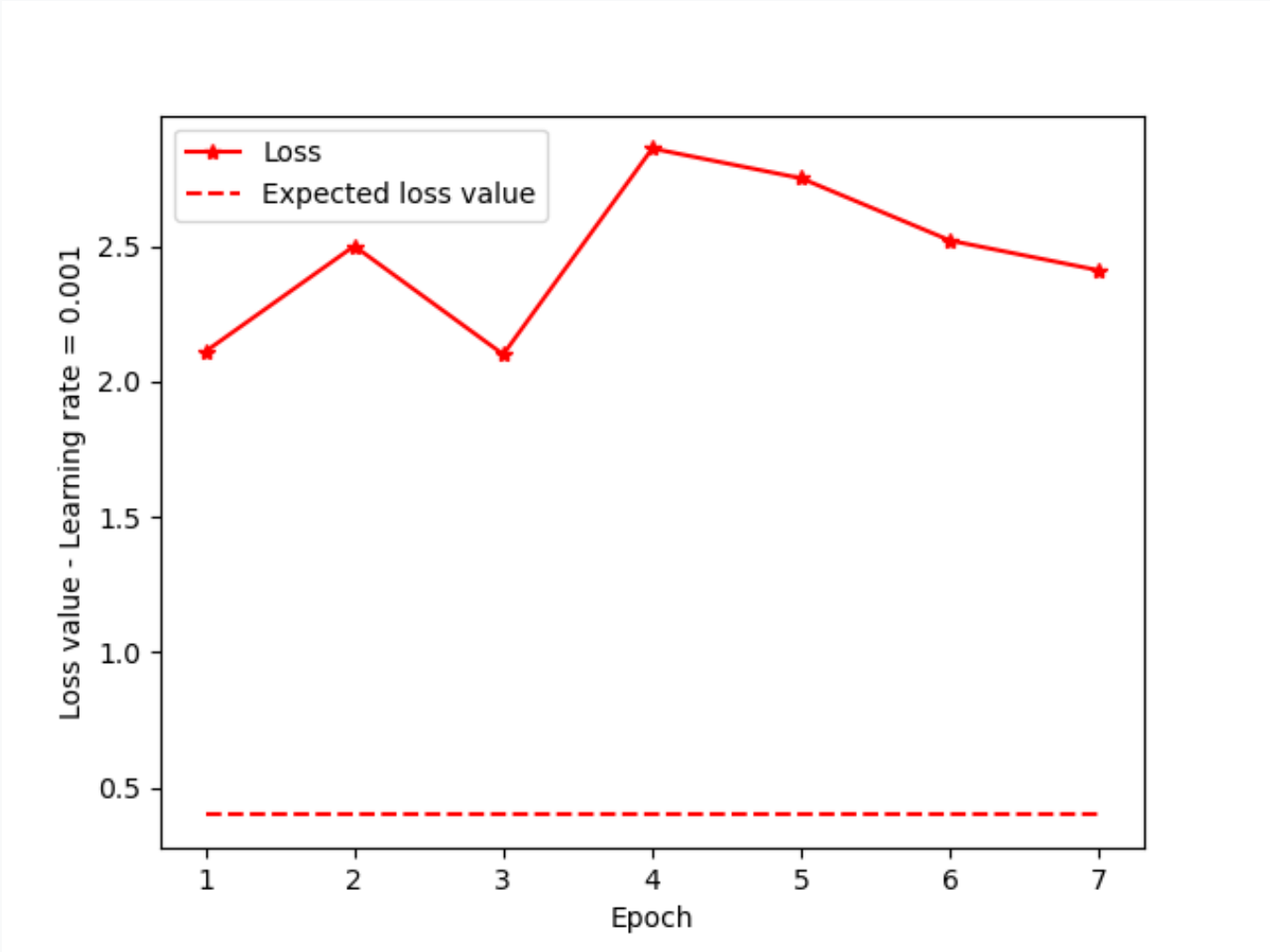
Epoch 5/10, Batch 300/400, Loss: 0.30899232625961304
Epoch 6/10, Batch 400/400, Loss: 0.2977037727832794
Epoch 7/10, Batch 100/400, Loss: 0.20920304954051971
Epoch 10/10, Batch 300/400, Loss: 0.18715685606002808
Total training accuracy: 88.41449999999999%
Total test accuracy: 10.01 %
Accuracy of the model on clean test images: 9 %

White-box setting

In the context of adversarial attacks on machine learning models, a "**white-box**" setting refers to a scenario where the attacker has complete knowledge of the model. This includes the architecture of the model, the parameters (weights and biases), the training method, and even the specific data points used for training.

In our code, the white-box setting is represented by the fact that the adversarial attack **('LinfPGD')** is being applied directly to the model using the **attack(fmodel, data, target, epsilons=[0.3])** line. The **fmodel** object contains all the information about the model, including its architecture and parameters, and this information is accessible to the 'attack' method.

We first start implementing the adversarial attack with l_∞ PGD attack Learning rate 0.001 with ADAM optimizer. r0.3

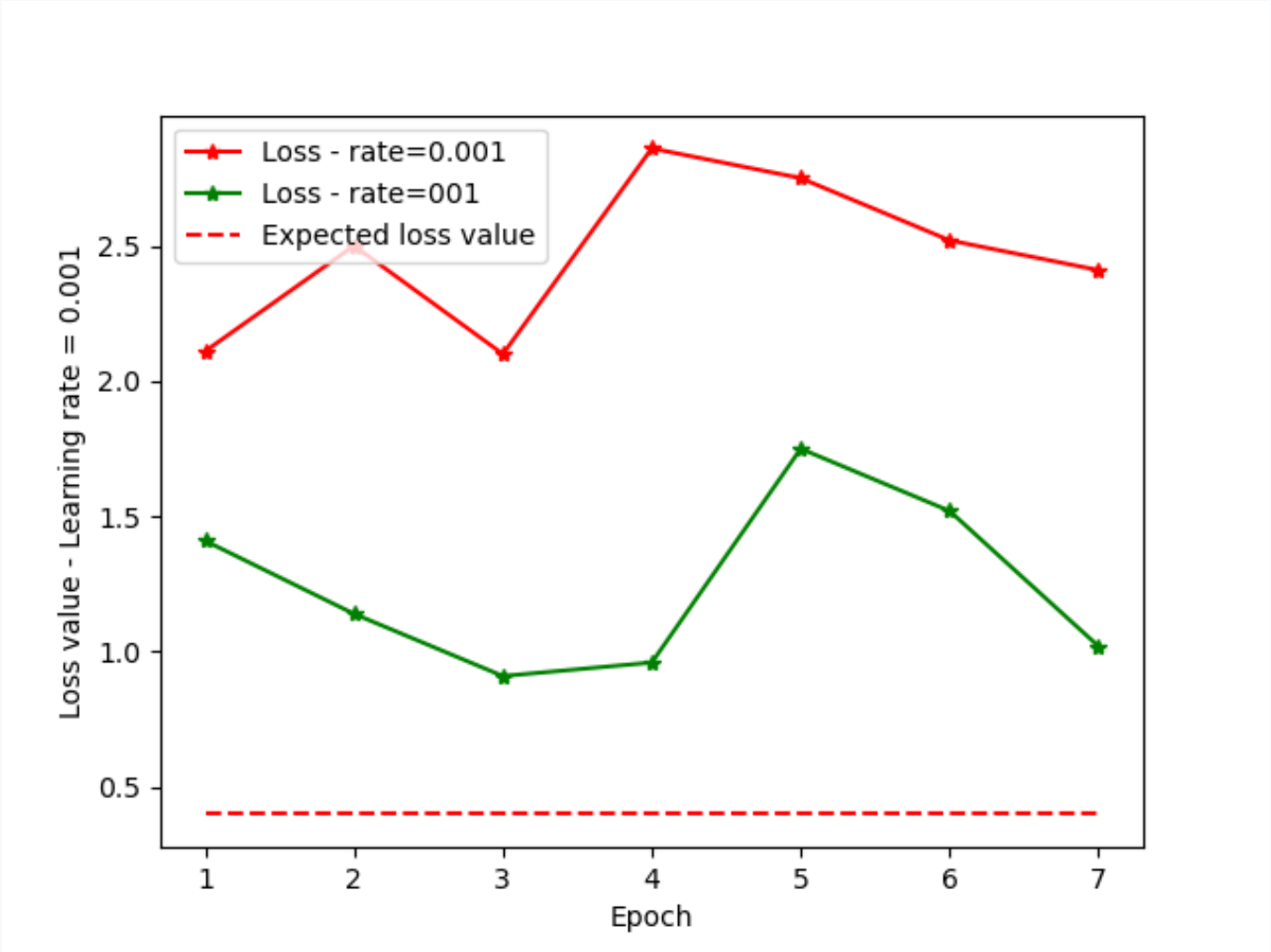


Epoch 1/30, Batch 1/2/3/4, Loss: 2.1193376779556274
Epoch 2/30, Batch 1/2/3/4, Loss: 2.5441186904907227
Epoch 3/30, Batch 1/2/3/4, Loss: 2.111435174942017
Epoch 4/30, Batch 1/2/3/4, Loss: 2.8636491894721985
Epoch 5/30, Batch 1/2/3/4, Loss: 2.7519114017486572
Epoch 6/30, Batch 1/2/3/4, Loss: 2.5245552659034729
Epoch 7/30, Batch 1/2/3/4, Loss: 2.4121231555938721
Training Accuracy between 9 and 10 percent. Since the model is not improving in learning we start reflecting on changing the learning rate

A change in the ***LEARNING RATE*** can help improve in learning.

$0.001 \rightarrow 0.01$

When We increased the learning rate from 0.001 to 0.01, you allowed the model to learn faster, i.e., make larger updates to its weights at each step. This can help the model to converge faster and escape shallow local minima, which might have resulted in the noticeable improvement in the loss.


Loss - rate=0.001
Loss - rate=001
Expected loss value
Loss value - Learning rate = 0.001
2.5
2.0
1.5
1.0
0.5
1
2
3
4
5
6
7
Epoch

Epoch 1/30, Batch 100/200/300/400/469, Loss: 1.4193376779556274
Epoch 2/30, Batch 100/200/300/400/469, Loss: 1.1441186904907227
Epoch 3/30, Batch 100/200/300/400/469, Loss: 0.9111435174942017
Epoch 4/30, Batch 100/200/300/400/469, Loss: 0.9636491894721985
Epoch 5/30, Batch 100/200/300/400/469, Loss: 1.7519114017486572
Epoch 6/30, Batch 100/200/300/400/469, Loss: 1.5245552659034729
Epoch 7/30, Batch 100/200/300/400/469, Loss: 1.020231555938721

The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Choosing the right learning rate is crucial as it heavily influences model performance.

Regularization l_2 and learning rate scheduler
can help improve in learning.
Learning-rate $\rightarrow$ 0.01 $\times$ *Learning — rate each 3 Epoch*

Regularization is a technique used in machine learning to prevent overfitting, which occurs when a model learns the training data too well and performs poorly on unseen data. Regularization l_2 adds a penalty term to the loss function to discourage the model from learning overly complex patterns in the training data.

We combined the regularization and the weight decay. Here is the results:

Epoch 1/10, Batch 100/400, Loss: 0.47082167863845825
Epoch 2/10, Batch 100/400, Loss: 0.457643985748291
Epoch 3/10, Batch 100/400, Loss: 0.3652268588542938
Epoch 4/10, Batch 100/400, Loss: 0.35956454277038574
Epoch 5/10, Batch 100/400, Loss: 0.29040566086769104
Epoch 6/10, Batch 200/400, Loss: 0.28241854906082153
Epoch 7/10, Batch 100/400, Loss: 0.32950145006179810
Epoch 10/10, Batch 400/400, Loss: 0.30253246426582336

Finished Adversarial Training - Total training accuracy = 87.68%
The model is up to generalize well on PGD-attacked unseen data with a high accuracy. Which means that the defense scheme of the model against this type of attack is strong. Finished Adversarial Testing - Total testing accuracy= 95.88

An important fact is how the model is able to recognize the normal unseen data. In this context the accuracy is still high.Testing Total accuracy: 95

Black-box setting

Black-box setting means that the attacker only has access to the inputs to the model and the corresponding outputs, without knowing the details of the model's architecture or parameters.

SPSA Attack

If your model is robust against the SPSA attack, it's a good sign that the model's robustness is not due to gradient masking. However, you should still test the model against other types of adversarial attacks to get a more complete understanding of its

Another model

In conclusion, if your model is robust against adversarial examples transferred from another model, it provides evidence that its robustness is not solely due to gradient masking. This type of robustness suggests that the model has learned to handle adversarial examples from a variety of sources, making it more reliable in real-world scenarios where attacks may come from different models or adversarial strategies.