War Attack Project Report

Team name: Scientists

Team members: Roni Asatourian (rasatour), Iman Reihanian Mashhadi (ireihani), Ryan Nicholas Cruz (rncruz1)

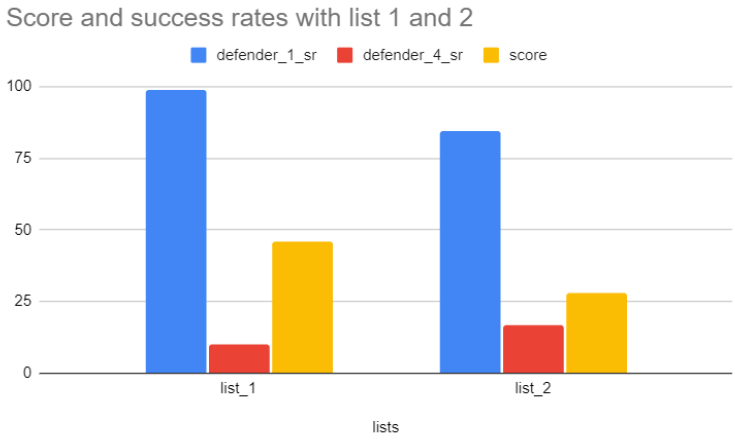
Description of attack methods

Our attack method is a white-box attack that is based on Projected Gradient Descent (PGD). The attack method we implemented iteratively add noise to the original image in order to misclassify our image to be the target image. The attack starts by iterating through a list of epsilon values in the range of ten through one hundred and stepping by ten. We then start with a random perturbation of the original image that is within the inner optimization problem. From here we perform our projected gradient descent on the image by calculating the distance of this perturbed image from the original image and adding the image to get our adversarial attacked imaged. We then check to see if this new image matches the target and if so return the image. If the image does not match the target we continue to perturb the image until we can no longer calculate the gradient. If we have not been able to successfully reach the target our attack will then iterate to the next eplison value to repeat the process and attempt to succesfully attack the image. One difference in our attack compared to the original PGD attack is that we do not perform random restarts on the starting perturbation image. Instead we only do a random start of the perturbed image and from here attempt to reach the target.

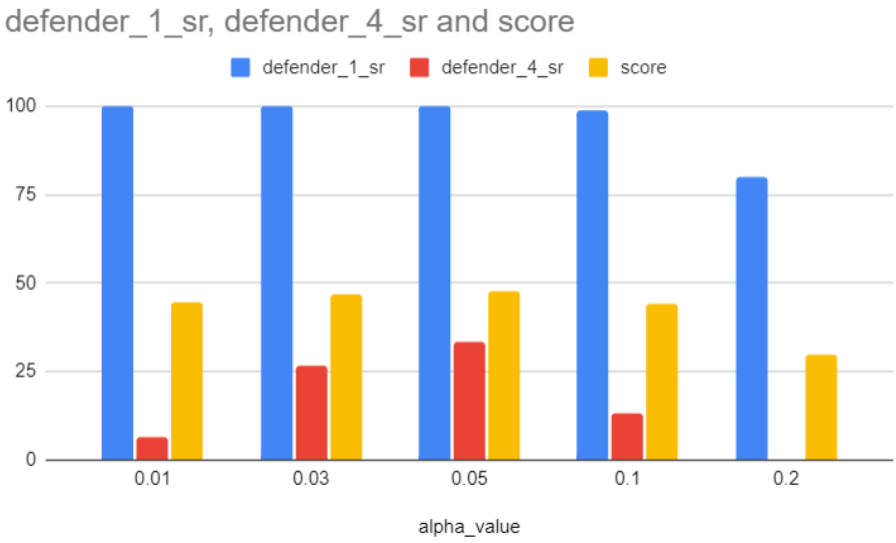
Analysis of Attack

When running our round run submission locally we received s scoreof 53.03% and were able to successful attack all 70 images for defender 1 but did not successfully attack a single image for defender 4. So we changed to our new PGD attack in which we tried to perfect our hyperparameters of our epsilon list and alpha value. When we first created our attack we did not

iterate through a list of epsilon values for perturbing the image. We initially left our epsilon value as 0.1 which results in use receiving a score of 22.73%. We were not satisfied with these results so we implemented the for loop to iterate through the list of epsilon values and perturb the image with these values. We tried two different lists one containing the values we used in our final version described in our attack method and the other containing values 0.00001, 0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 10, and 100. The graph to the right shows the attack success rate for each of the defenders with these two different lists and alpha value 0.1 with list 1 being our final version and list 2 being the second version. We found that our final epsilon list of values ten through one hundred incrementing by ten worked best.



We then tuned our alpha value to attempt to increase the attack success rate of defender four and improve our score. We did this by trying different alpha values of 0.1, 0.2, 0.05, 0.01, 0.03 with the final version of our epsilon list. By doing this we found that alpha value 0.05 worked the best with a score of 47.66%. The graph below displays the results we got for these different values.



Conclusion

Overall our submission for round 2 has a slightly lower score when ran locally compared to round one but does successfully attack more images with the second defender which is why we decided to submit this version and believe it would result in a higher score on the leaderboard since we would have a higher success rate for the hidden defenders. If we attempted this project again we would fine tune our attack method to use less queriers for in order to increase our score. The code was ran and this report was written based on the github version 98a7872 titled “added new defense for students”.

Works cited

Haldar, Siddhant. “Gradient-Based Adversarial Attacks : An Introduction.” Medium, The Startup, 9 Apr. 2020, https://medium.com/swlh/gradient-based-adversarial-attacks-an-introduction-526238660dc9.

Madry, Aleksander, et al. “Towards Deep Learning Models Resistant to Adversarial Attacks.” Arxiv.org, Arxiv, 4 Sept. 2019, https://arxiv.org/abs/1706.06083.