Adversarial Attacks and Defenses for Image-Based Recommendation Systems using Deep Neural Networks

Master Thesis

Philipp Normann

Department of Computer Science
University of Applied Sciences Wedel

November 5, 2020

Overview

Motivation

Recommendation systems have reached widespread adoption

Numerous companies, ranging from e-commerce marketplaces, to streaming services, as well as social networks and news aggregators, successfully deploy such systems.

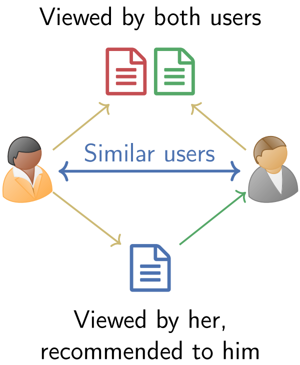
Malicious actors try to exploit these systems to their advantage

Depending on the application area of the system, a successful compromise can have far-reaching consequences.

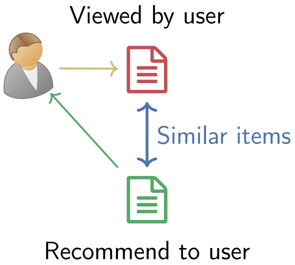
A better understanding of attacks and defenses is needed

This thesis closes this research gap by developing targeted attacks and defenses using standard techniques from the field of adversarial examples for a visual recommendation system.

Recommendation Systems



(a) Collaborative Filtering



(b) Content-based Filtering

Figure: Typical categorization for recommendation systems.

Visual Recommendation Systems

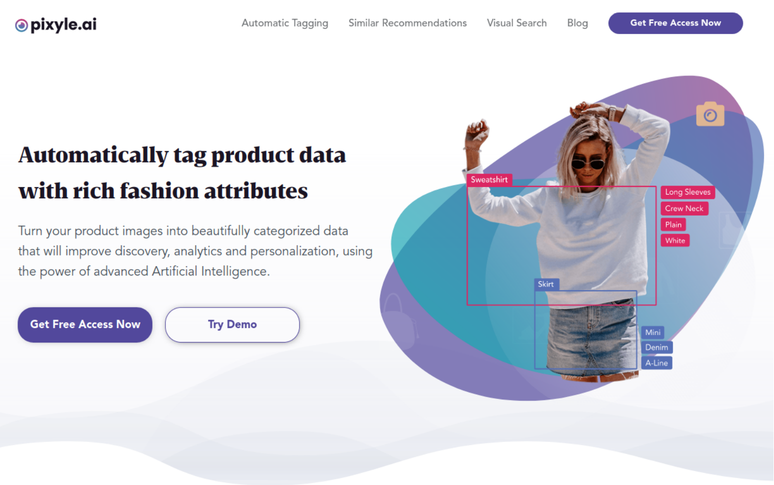


Figure: Pixyle.ai: Visual AI in fashion e-commerce

Adversarial Examples

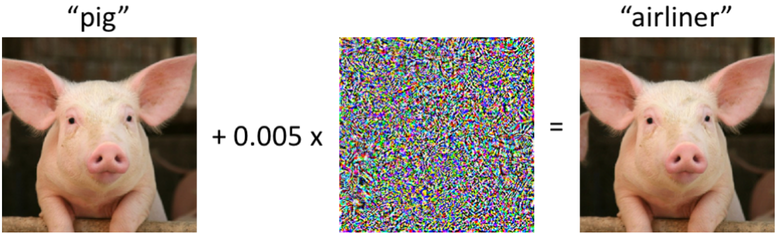
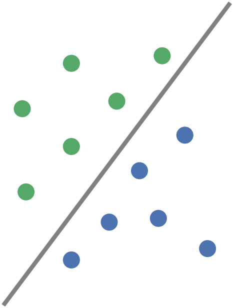
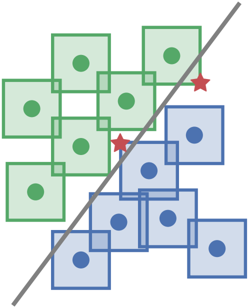


Figure: On the left, we have an image of a pig that is correctly classified as such by a state-of-the-art CNN. After perturbing the image slightly, the network now returns class "airliner" with high confidence (Mądry & Schmidt, 2018).

Adversarial Examples



(a) A set of points that can be easily separated with a simple decision boundary.



(b) The simple decision boundary does not separate the l_∞-balls around the data points. Hence there are adversarial examples that will be misclassified.

Figure: Adapted from Madry et al., 2017

Adversarial Training

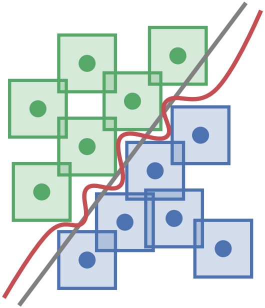


Figure: Separating the l_∞-balls requires a significantly more complicated decision boundary. The resulting classifier is robust to adversarial examples with bounded l_∞-norm perturbations.

Related Work

Adversarial Training Towards Robust Multimedia Recommender System

Jinhui Tang, *Senior Member, IEEE,* Xiaoyu Du, Xiangnan He, Fajie Yuan, Qi Tian, *Fellow, IEEE,* and Tat-Seng Chua

TABLE 3: Performance drop (relatively decreasing ratio in NDCG@10) of VBPR and AMR in the presence of adversarial perturbations during the testing phase.

	$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$	
Dataset	**VBPR**	**AMR**	**VBPR**	**AMR**	**VBPR**	**AMR**
Pinterest	-4.2%	-2.6%	-11.9%	-6.2%	-31.8%	-18.4%
Amazon	-8.7%	-1.4%	-30.4%	-5.3%	-67.7%	-20.2%

Figure: Tang et al., 2019 explored the general vulnerability of content-based recommenders using CNNs to untargeted attacks.

Related Work

TAaMR: Targeted Adversarial Attack against Multimedia Recommender Systems

Tommaso Di Noia
Politecnico di Bari
tommaso.dinoia@poliba.it

Daniele Malitesta
Politecnico di Bari
daniele.malitesta@poliba.it

Felice Antonio Merra
Politecnico di Bari
felice.merra@poliba.it

(a) original (sock)
probability: 60%
rec. position: 180th

(b) attacked (running shoe)
probability: 100%
rec. position: 14th

Fig. 2: Example of a product image before (a) and after (b) a PGD attack ($\epsilon = 8$) against VBPR on `Amazon Men`.

Figure: Di Noia et al., 2020 explored the vulnerability of content-based recommenders using CNNs to targeted misclassification attacks.

Dataset

- We use the DeepFashion Attribute Prediction [1] dataset published by Liu et al., 2016

	Classification		Samples
Dataset	Type	No.	Total
DeepFashion Category	Multinomial	46	279,057
DeepFashion Texture	Multinomial	156	106,649

Table: Summary of the preprocessed DeepFashion dataset.

[1] http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion/AttributePrediction.html

Dataset

(a) Cardigan striped (b) Tank print (c) Tee striped (d) Dress stripe (e) Sweater striped

(f) Blouse floral (g) Shorts houndstooth (h) Dress chevron (i) Skirt dotted (j) Poncho tribal

Figure: Randomly sampled images from the DeepFashion dataset.

Model

- Reproduced model, published by Tuinhof et al., 2018.
- Two-stage model using a CNN classifier and a k-NN search
- CNN classifier is trained to predict category and texture
- Latent embeddings of the trained CNN classifier are used for similarity based k-NN recommendations
- As a similarity measure, cosine distance is used

Category	Ours	Tuinhof et al., 2018
Accuracy	68.25	63.00
Top-5 Accuracy	93.14	84.00

Table: Our category classifier results in comparison to the results reported in the original paper by Tuinhof et al., 2018.

Model

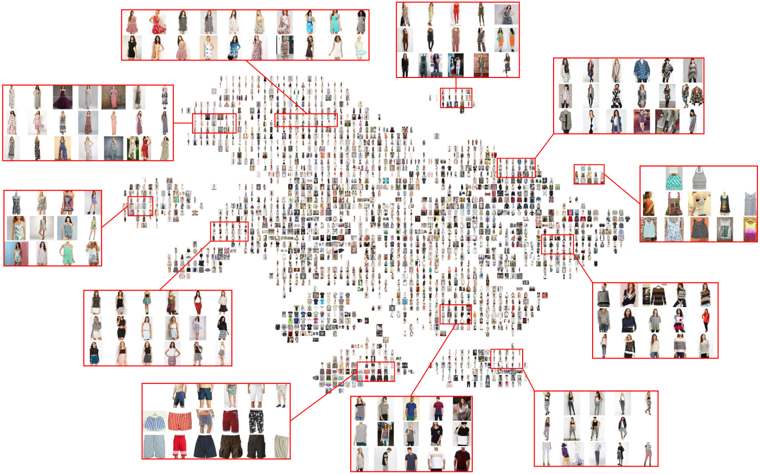


Figure: t-SNE visualization of articles from the DeepFashion dataset, using their feature vectors from the penultimate layer of the classifier.

Model

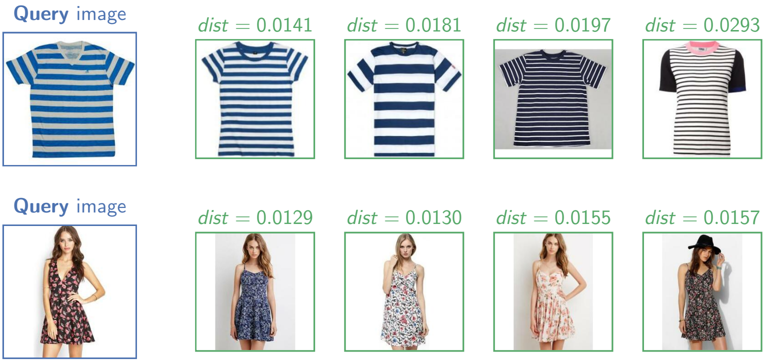


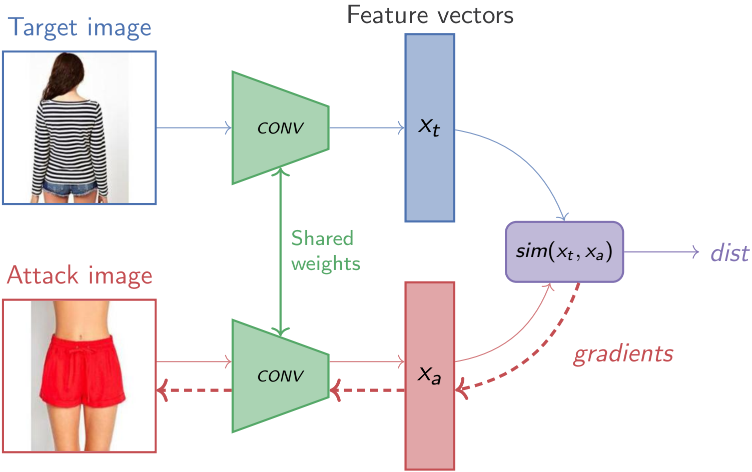
Figure: Ranked k-NN results for two randomly selected items

Attacks

Threat model based on guidelines by Carlini and Wagner, 2017:

- **adversary goal:** The adversary is interested in minimizing the cosine distance between the latent-space embeddings of an attack article image to a pre-existing target article image. By minimizing this distance, the chosen attack article decreases its rank in the list of nearest neighbors of the target article, thereby promoting the attack article.
- **adversary knowledge:** We assume a white-box knowledge setting, in which the adversary holds full knowledge of the feature extraction model parameters.
- **adversary capability:** We restrict the adversary capability to make l_∞-norm constrained perturbations to the image.

Attacks


Target image
Feature vectors
CONV
x_t
Shared weights
$sim(x_t, x_a)$
dist
Attack image
CONV
x_a
gradients

Fast Gradient Sign Method

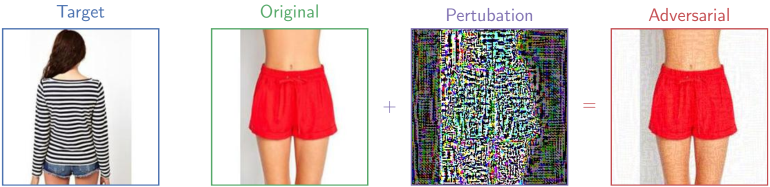


Figure: Adversarial example, created using the FGSM with $\epsilon = 0.03$. The perturbation is normalized for visualization purposes.

Cosine distances before and after FGSM attack for this example:

$$dist(\mathcal{F}(A), \mathcal{F}(T)) = 0.6247 \quad (1)$$

$$dist(\mathcal{F}(A + \delta), \mathcal{F}(T)) = 0.5267 \quad (2)$$

Projected Gradient Descent

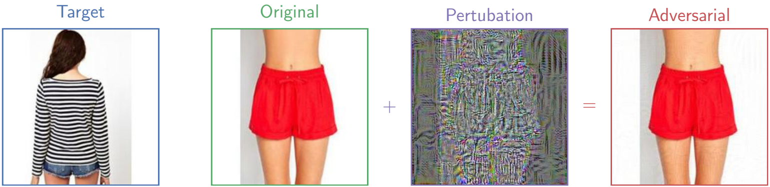


Figure: Adversarial example, created using PGD with $\epsilon = 0.03$ and 32 iterations. The perturbation is normalized for visualization purposes.

Cosine distances before and after PGD attack for this example:

$$dist(\mathcal{F}(A), \mathcal{F}(T)) = 0.6247 \tag{3}$$

$$dist(\mathcal{F}(A+\delta), \mathcal{F}(T))) = 0.0500 \tag{4}$$

Projected Gradient Descent

Figure: Recommendation results for original k-NN index (top) and manipulated index with injected PGD adversarial example (bottom)

Carlini & Wagner Method

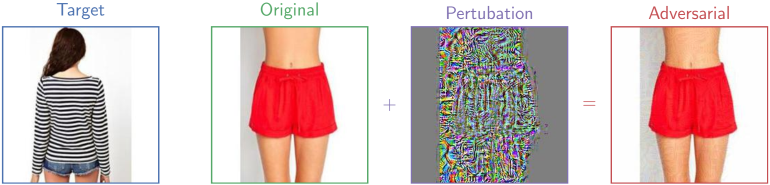


Figure: Adversarial example, created using the CW method with $\epsilon = 0.03$ and 1,000 iterations. The perturbation is normalized for visualization purposes.

Cosine distances before and after CW attack for this example:

$$dist(\mathcal{F}(A), \mathcal{F}(T)) = 0.6247 \tag{5}$$

$$dist(\mathcal{F}(A+\delta), \mathcal{F}(T)) = 0.0049 \tag{6}$$

Carlini & Wagner Method

Figure: Recommendation results for original k-NN index (top) and manipulated index with injected CW adversarial example (bottom)

Comparison

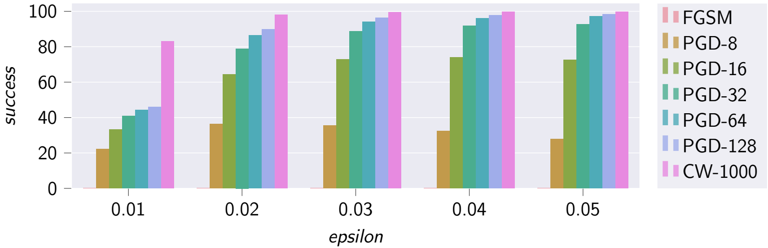


Figure: Success rates (%) for $rank_{min} = 3$, calculated over 10,000 random tuples (1,000 in the case of CW) for all attacks and ϵ values.

Defenses

How can we defend our recommendation system against adversarial inputs?

Adversarial Training

- Train on adversarial examples using correct labels
- Adversary objective, is to increase the likelihood of misclassification for the category, and texture attributes
- Adversarial examples during training are generated using PGD-8 and restricting l_∞ perturbations to $\epsilon = 0.03$

Category	Adversarial	Regular	Δ
Clean Accuracy	56.06	68.25	$-$ 12.19
Adversarial Accuracy	48.71	0.02	$+$ 48.69

Table: Category classification results on a clean and adversarial test set for a adversarially trained and regular classifier. The adversarial test set was generated using the PGD-8 attack and $\epsilon = 0.03$.

Adversarial Training

Figure: A recommendation result of our adversarially trained model after a targeted attack. The adversarial example generated using the CW-1000 method for $\epsilon = 0.3$, which we injected into the product catalog ranks on place 39 and is therefore not visible in the nearest neighbors displayed above.

Adversarial Training

(a) $\epsilon = 0.01$ (b) $\epsilon = 0.05$ (c) $\epsilon = 0.1$ (d) $\epsilon = 0.2$ (e) $\epsilon = 0.3$

Figure: Adversarial examples generated using CW-1000 for our adversarially trained recommendation system with increasing ϵ values ranging from 0.01 to 0.3. The target item for the attack is the same striped sweater. Interestingly the adversarial images with high epsilon values start to show relevant features of the target image.

Adversarial Training

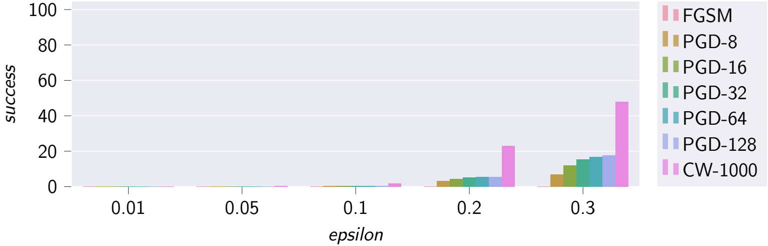


Figure: Success rates (%) for $rank_{min} = 3$, targeting an adversarially trained model, calculated over 10,000 random tuples (1,000 in the case of CW) for all attacks and ϵ values.

Curriculum Adversarial Training

- Trade robustness for clean performance by increasing attack strength during training, starting with $k = 0$
- Adversarial examples are generated using PGD attacks with up to $k = 8$, restricting l_∞ perturbations to $\epsilon = 0.03$

Category	Curriculum	Regular	Δ
Clean Accuracy	62.29	68.25	$-$ 5.96
Adversarial Accuracy	27.45	0.02	$+$ 27.43

Table: Category classification results on a clean and adversarial test set for a classifier trained using curriculum adversarial training and a regular classifier. The adversarial test set was generated using the PGD-8 with $\epsilon = 0.03$

Curriculum Adversarial Training

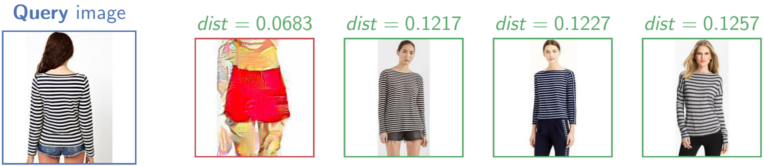


Figure: A recommendation result of our CAT model after a targeted attack. The adversarial example generated using the CW method for $\epsilon = 0.2$, which we injected into the product catalog, ranks first among the target's neighbors.

Curriculum Adversarial Training

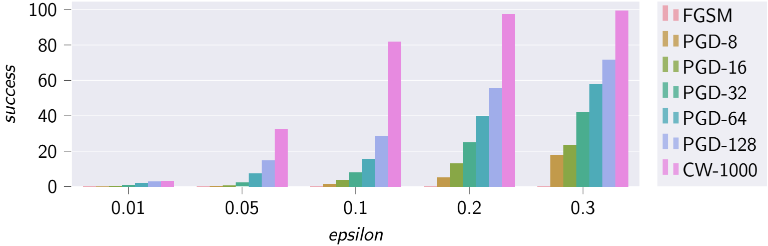


Figure: Attack success rates for $rank_{min} = 3$ calculated over 10,000 random article tuples (1,000 in the case of CW) for all evaluated attacks and various ϵ values.

Comparison

	Attack		
Defense	FGSM	PGD-128	CW-1000
Unsecured	0.07	98.32	99.70
AT	0.03	0.07	0.30
CAT	0.00	14.89	32.80

Table: Attack success rates for $rank_{min} = 3$ calculated over 10,000 random article tuples (1,000 in the case of CW) for all evaluated models and $\epsilon = 0.05$.

Conclusion

- We developed a new type of targeted item-to-item attack using state-of-the-art white-box methods and observed their effectiveness in compromising the integrity of the attacked visual recommendation system.
- We tested two defense mechanisms utilizing adversarial training (AT) and were able to show that AT had a significant positive impact on the robustness of our system.
- Although our experiments demonstrated a strong robustness against our evaluated white-box attacks, it is unclear if and how far these results generalize for black-box or future unknown attacks.

Conclusion

- Also, the effect of similar attacks and defenses on hybrid RS using DNN remains to be explored.
- Additionally, the trade-off in recommendation quality and robustness caused by AT remains to be quantified, possibly by conducting user-surveys or A/B testing.
- Overall, our findings have once again demonstrated the inherent vulnerability of DNN, but have also given us hope that adversarially robust recommendation system models using DNN might be within current reach.

Technical Details

Source code and results are published on GitHub [2]

Implemented in *Python* using the following libraries:

- Deep Learning Framework: ***PyTorch*** [3]
- Experiment Monitoring: ***TensorBoard*** [4]
- Approximate K-NN search: ***NMSLIB*** [5]
- Image Deduplication: ***imagededup*** [6]
- Data Preprocessing: ***pandas*** [7]
- Visualizations: ***seaborn*** [8]

[2] https://github.com/philippnormann/master-thesis
[3] https://github.com/pytorch/pytorch
[4] https://github.com/tensorflow/tensorboard
[5] https://github.com/nmslib/nmslib
[6] https://github.com/idealo/imagededup
[7] https://github.com/pandas-dev/pandas
[8] https://github.com/mwaskom/seaborn

Fast Gradient Sign Method

	Maximal Perturbation				
$rank_{min}$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.05$
1	0.12	0.07	0.06	0.02	0.01
3	0.27	0.16	0.14	0.09	0.07
10	0.64	0.44	0.32	0.18	0.13
100	2.87	2.45	1.83	1.36	0.99

Table: Success rates (%) using FGSM for 10,000 random tuples.

Fast Gradient Sign Method

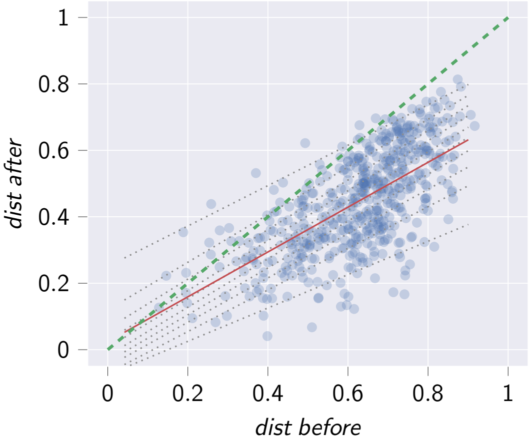


Figure: Quantile regression plot of cosine distances between target and attack article, before and after FGSM attacks, using $\epsilon = 0.05$

Projected Gradient Descent

	Maximal Perturbation				
$rank_{min}$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.05$
1	36.44	77.81	86.81	89.65	91.02
3	44.33	86.40	94.06	96.13	97.09
10	50.21	89.61	95.90	97.56	98.22
100	62.55	94.13	97.95	98.74	99.13

Table: Success rates (%) using PGD-64 for 10,000 random tuples.

Projected Gradient Descent

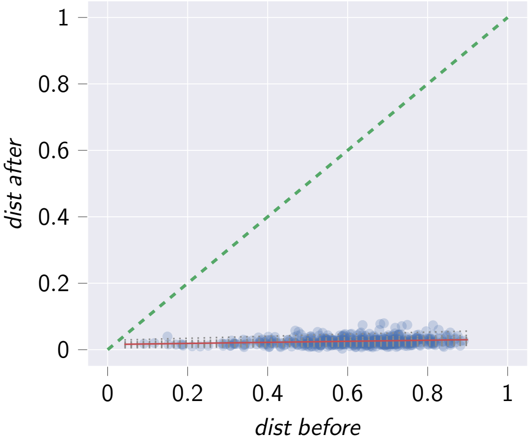


Figure: Quantile regression plot of cosine distances between target and attack article, before and after PGD-32 attacks, using $\epsilon = 0.05$

Carlini & Wagner Method

	Maximal Perturbation				
$rank_{min}$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.05$
1	74.60	94.10	96.40	97.60	97.80
3	83.10	98.10	99.40	99.70	99.70
10	86.60	98.40	99.50	99.90	99.90
100	91.30	99.40	99.90	100.00	100.00

Table: Success rates (%) using CW-1000 for 1,000 random tuples.

Carlini & Wagner Method

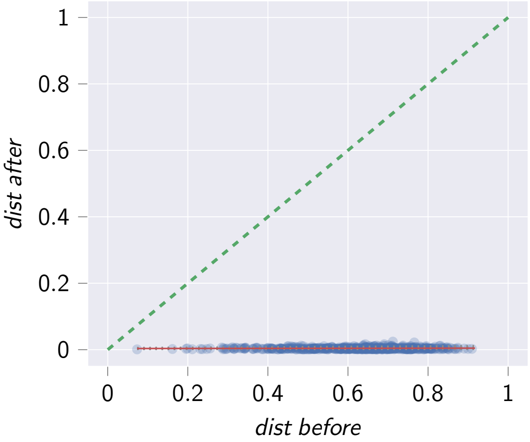


Figure: Quantile regression plot of cosine distances between target and attack article, before and after CW-1000 attacks, using $\epsilon = 0.05$

Adversarial Training

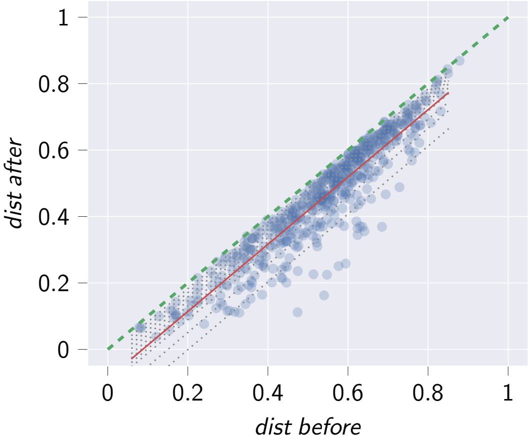


Figure: Quantile regression plot of cosine distances between target and attack article, before and after performing CW-1000 attacks targeting an adversarially trained model, using $\epsilon = 0.05$.

Curriculum Adversarial Training

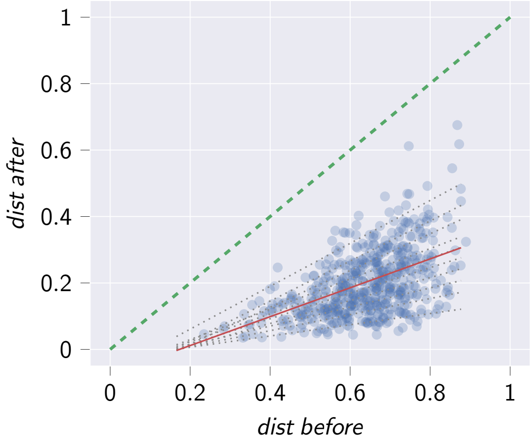


Figure: Quantile regression plot of cosine distances between target and attack article, before and after performing CW-1000 attacks targeting a model trained using curriculum AT for $\epsilon = 0.05$.

References I

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. IEEE. Retrieved August 24, 2020, from https://arxiv.org/pdf/1608.04644.pdf. (Cit. on p. 16)

Di Noia, T., Malitesta, D., & Merra, F. A. (2020). Taamr: Targeted adversarial attack against multimedia recommender systems. Retrieved August 24, 2020, from https://arxiv.org/pdf/1706.01084.pdf. (Cit. on p. 10)

References II

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of ieee conference on computer vision and pattern recognition (cvpr)*. Retrieved August 24, 2020, from https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Liu_DeepFashion_Powering_Robust_CVPR_2016_paper.pdf. (Cit. on p. 11)

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*. Retrieved August 24, 2020, from https://arxiv.org/pdf/1706.06083) (cit. on p. 7)

References III

Mądry, A., & Schmidt, L. (2018). A brief introduction to adversarial examples. Retrieved August 24, 2020, from https://gradientscience.org/intro_adversarial/ (cit. on p. 6)

Tang, J., Du, X., He, X., Yuan, F., Tian, Q., & Chua, T.-S. (2019). Adversarial training towards robust multimedia recommender system. *IEEE Transactions on Knowledge and Data Engineering*. Retrieved August 24, 2020, from https://arxiv.org/pdf/1809.07062.pdf (cit. on p. 9)

Tuinhof, H., Pirker, C., & Haltmeier, M. (2018). Image-based fashion product recommendation with deep learning. In *International conference on machine learning, optimization, and data science*. Springer. Retrieved August 24, 2020, from https://arxiv.org/pdf/1805.08694.pdf. (Cit. on p. 13)