Adversarial Robustness Across Representation Spaces

Sruthy Annie Santhosh *, Diego Coello de Portugal *, Heliya Hasani*, Aditya Nair *,
Supervisor: Mofassir ul Islam Arif *

Stiftung Universität Hildesheim, Institut für Informatik

santhosh/coello/hasani/nair @uni-hildesheim.de
mofassir@ismll.de

* equal collaboration

Introduction

When training a deep neural network model to understand an image data-set, a common occurrence is that the trained model output changes significantly between an original image and that same image with imperceptible perturbations. These perturbations in the original image that interfere with the model performance are called Adversarial Attacks. One of the fundamental implications of adversarial attacks is that in some instances where the task corresponds to classification, the model evaluates the image from the adversarial attack as belonging to a class completely different from the correct class even though the original image was classified correctly. It is worth mentioning, that the changes done to the original image are so small that the perturbed image should also belong to the same class, being in most cases imperceptible to the human eye. Thus the adversarial attacks prove that these type of models are not foolproof and can be inconsistent. The capability of a model to resist being fooled is called Adversarial Robustness.

The aim of this project is to improve adversarial robustness in various data-sets, using adversarial training in neural networks across different representation spaces. Pixels don't give precise information regarding images. For more information waves signals should be considered. For instance, DCT is a type of a signal which is used in image compression because it is important that the pictures do not take up space on the computer. The primary reason we use Fourier series is that we can better analyze a signal in another domain rather in the original domain. Adversarial robustness does not generalize well between different attacks or representation spaces, but adversarial robustness is crucial in real life scenarios because there are different types of parameters which cannot be trained before but should be detected by the machine learning algorithms when deployed.

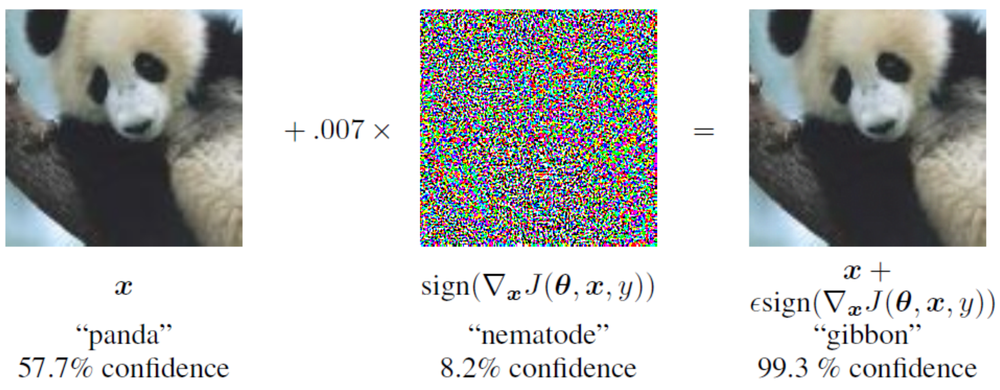


Figure 1: Perturbation attack with *FGSM* [3]

Project Approach

The baseline paper for this project is *Adversarial Robustness across Representation Spaces* [2]. In this paper, the authors mention the importance of training against different adversarial attacks, since training against specific representation spaces only improves the robustness against the same type of representation spaces (Table 1). The most commonly used methods to create adversarial examples are Fast Gradient Sign Method (FGSM [3]) and Project Gradient Descent Method (PGD), the latter being used in the baseline paper. Both of them aim to do small perturbations on the image to get an adversarial example. These perturbations are bounded depending on the representation chosen: pixel based, *Discrete Cosine Transform* (DCT) among others. Lastly, the model used will be *ResNet-50* since it is widely used for image related tasks and is also the one used in the baseline.

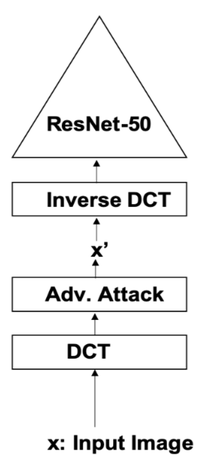
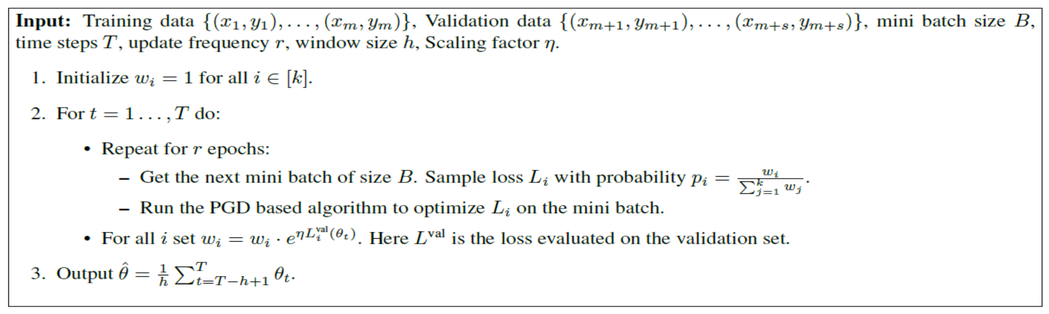


Figure 2: Network architecture example for *Discrete Cosine Transformation* (DCT)

- At the beginning, we used *MNIST* and *FashionMNIST* data-sets to prove the validity of the statements given by the baseline. The first hypothesis of robustness not being transferable between different representation spaces was found to be true. The second hypothesis of *Multiplicative Weights* training method being better than other training procedures such as *Round Robin* or *Greedy* was also found to be true in these cases. In this part of the project we used *FGSM*, *FFGSM* and *PGD* attacks in the representation spaces: pixel, (DCT) and *Discrete Fourier Transform* (DFT). We have further implemented an AutoEncoder to confirm that the transformation loss of the images are negligible.



Input: Training data $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, Validation data $\{(x_{m+1}, y_{m+1}), \ldots, (x_{m+s}, y_{m+s})\}$, mini batch size B, time steps T, update frequency r, window size h, Scaling factor η.

1. Initialize $w_i = 1$ for all $i \in [k]$.
2. For $t = 1 \ldots, T$ do:
   - Repeat for r epochs:
     - Get the next mini batch of size B. Sample loss L_i with probability $p_i = \frac{w_i}{\sum_{j=1}^{k} w_j}$.
     - Run the PGD based algorithm to optimize L_i on the mini batch.
   - For all i set $w_i = w_i \cdot e^{\eta L_i^{\text{val}}(\theta_t)}$. Here L^{val} is the loss evaluated on the validation set.
3. Output $\hat{\theta} = \frac{1}{h} \sum_{t=T-h+1}^{T} \theta_t$.

Figure 3: *Multiplicative Weights* training procedure proposed by Georgeyu [2].

	Test w/ PGD_{Pixel}	Test w/ PGD_{DCT}	Test w/ PGD_{DFT}	Clean images
Train w/ PGD_{Pixel}	**52.79±0.45**	38.64±1.0	68.83±0.35	73.45±0.83
Train w/ PGD_{DCT}	50.53±1.02	**49.86±0.86**	65.92±1.64	69.23±1.73
Train w/ PGD_{DFT}	14.48±1.39	5.83±0.51	**77.88±0.43**	**86.06±0.46**

Table 1: Accuracy in *FashionMNIST* for training and testing in different representation spaces. The results are the average of 5 experiments.

- Next, we used more complex data-sets (*CIFAR10/CIFAR100*), added PGDL2 attack (PGD with norm 2) and other representation spaces: *JPEG* transform and *LOG* transform. The first hypothesis of the model not being robust when changing the representation spaces used for the attacks was still true. However, we found that in the more complex data-sets with a huge number of attacks (more than 20), *Multiplicative Weights* was no longer outperforming *Round Robin*.

	Clean images	FGSM	PGD	PGDL2	Average
STD	**86.78±0.26**	6.95±1.09	0.0±0.0	39.94±0.95	33.68±34.21
Round Robin	79.14±1.84	**19.98±1.27**	1.4±0.17	**67.32±1.01**	**47.8±27.0**
Greedy	28.69±2.59	18.88±1.52	**16.39±2.13**	27.16±2.08	24.83±3.71
Multiplicative Weights	77.64±2.22	19.56±2.42	1.33±0.42	64.82±1.08	46.09±26.5

Table 2: Accuracy in *CIFAR10*. The training procedure used FGSM, FFGSM, PGD and PGDL2 attacks with Pixel, DCT, JPEG, DFT and LOG representations. The Average column is the average accuracy of all the attacks, including the ones that didn't fit in the table. FGSM, PGD and PGDL2 columns are in Pixel representation. The results are the average of 5 experiments.

- Additionally, we performed Holdout Testing on CIFAR10 dataset. Here we trained on a fixed set of attacks (any 6) for all epochs and the testing was done on all the attacks. It was observed that the robustness was not translated well across the attacks. Lastly we performed a dynamic variation of the baseline approach. Here the training was done on random sets of attacks in each epoch. Round Robin outperforms the other two training methods.

Conclusions

- From our experiments we have found that, training in a specific representation space to increase robustness does not **translate** to other representation spaces.
- *Multiplicative Weights* method is better than *Round Robin* for small/medium amount of attacks/representation spaces (<17) in less complex data-sets (*MNIST, FashionMNIST*).
- However, the claim of *Multiplicative Weights* being **scalable** does not hold true for a higher amount of attacks (>20) and for more complex data-sets (*CIFAR10, CIFAR100*). *Multiplicative Weights* method tends to collapse, as more probability is given for attacks with low robustness. Therefore, there is no longer a significant difference between *Multiplicative Weights* and *Round Robin* training in this case.
- Not all representation spaces have the same impact for adversarial training. For example, *Log space* and *DFT* tend to squish down the perturbation after doing the inverse transformation, reducing the impact of the attack.

The code for our research implementation can be found at:
https://github.com/adnair11/adverserial-robustness-project

References

[1] Shuhao Cao. Replicate the fourier transform time-frequency domains correspondence illustration using tikz. *stackexchange*, 2013.

[2] Yu. George et al. Adversarial robustness across representation spaces. *arXiv:2012.00802*, 2020.

[3] Ian J. Goodfellow et al. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014.

[4] ihritik. Matlab — rgb image representation. *GeeksforGeeks*, 2018.

[5] Syed Ali Khayam. The discrete cosine transform (dct): Theory and application. *Michigan State University*, 2003.

[6] Dave Marshall. The discrete cosine transform (dct). 2001.

[7] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.