

Fooling Neural Networks using Adversarial Images

Akkapaka Saikiran

Indian Institute of Technology Bombay

Abstract

Deep neural networks have revolutionized many high-level pattern recognition tasks previously thought impossible. However, at times they show some intriguing properties. One of these is the existence of blind spots in the representation space, which can be exploited by optimization to yield adversarial images, i.e. images which are nearly indistinguishable from an original image yet lead the model to a false prediction. In this paper we demonstrate some such images on the FashionMNIST dataset. We also analyze certain properties of these adversarial images. All code can be found at [this link](#).

1 Introduction

Deep neural networks have brought about significant advances in several visual recognition tasks. However, they occasionally behave in unexpected ways, i.e. ways which demonstrate differences between computer and human vision. In this paper we explore one intriguing property of neural networks. Neural networks are not robust to small specific perturbations to their inputs. Suppose we have a network trained on image classification. We observe that it is possible to arbitrarily change its predictions by imperceptibly perturbing a test image. This is unexpected because neural networks show remarkable performance on unseen images, thus showing that they can generalize beyond the training set. These perturbations are found by optimizing the image to maximize prediction error on the desired misclassification label. These perturbed examples are called “adversarial images” [1].

Deep neural networks show non-local generalization. Very dissimilar test images can be classified under the same label, i.e. neural networks manage to map images of two completely different looking dogs (say) to the same label (that of a dog). These two images, which are far apart in the image space, get projected to embeddings that are close to each other in the representation space. However, local generalization is often taken for granted. If an image \mathbf{x} is labeled as y with probability p , then one expects that there exists a small neighbourhood of size ϵ in the vicinity of \mathbf{x} such that every image $\mathbf{x}' = \mathbf{x} + \mathbf{r}$ such that $\|\mathbf{r}\| < \epsilon$ is assigned a probability p' of being y and $|p - p'| < \delta$ for some small δ . This kind of smoothness prior is often valid for computer vision problems, where small perturbations are *imperceptible*.

We demonstrate in this paper that via a simple optimization procedure, it is possible to generate adversarial images. We also perform some analysis on the images generated thereof after laying out a formal framework to describe our process.

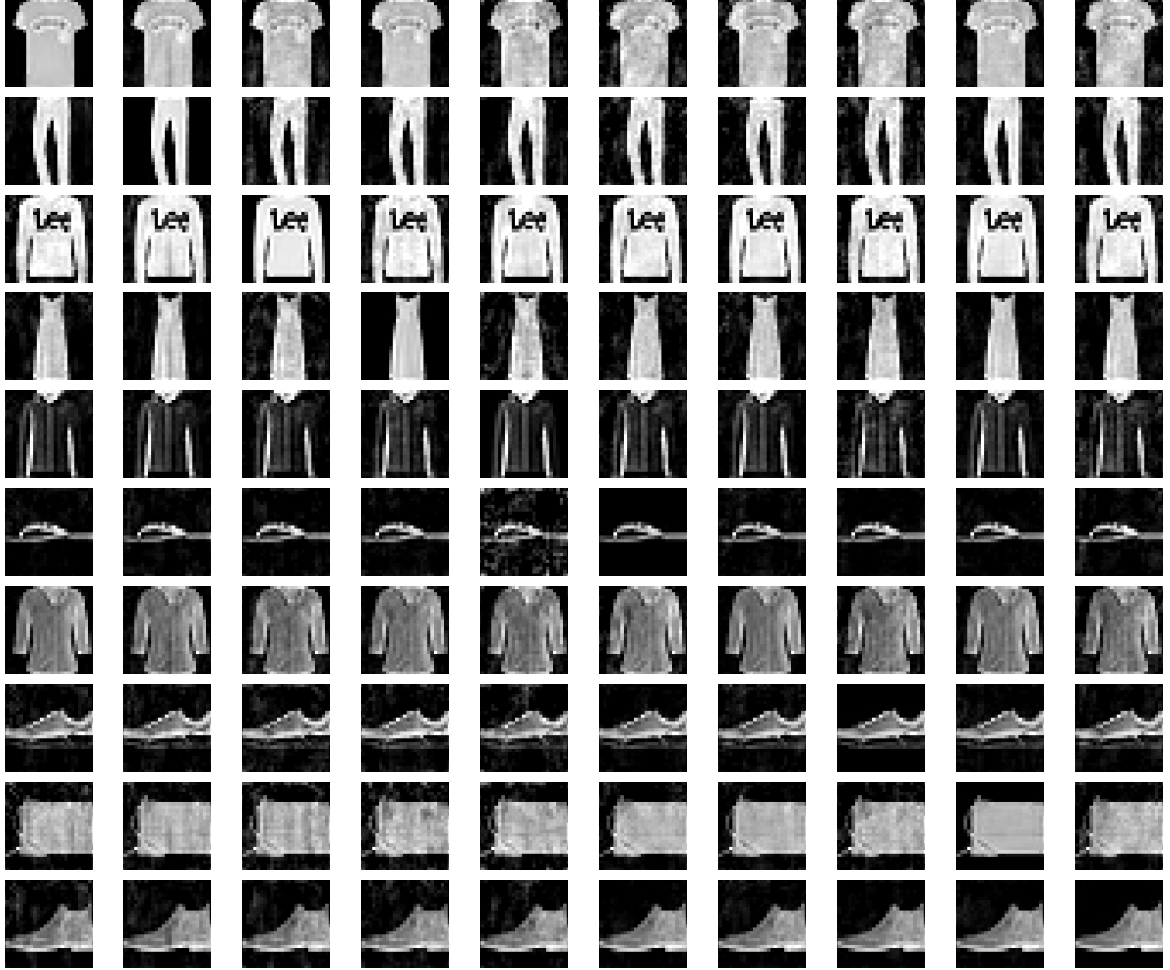


Table 1: **Adversarial images.** Each row corresponds to a true label, each column to a false prediction.

2 Mathematical Formulation

Let $f : \mathbb{R}^m \rightarrow \{0, \dots, k-1\}$ be a classifier that maps images to a discrete label set. Let $L_f : \mathbb{R}^m \times \{0, \dots, k-1\} \rightarrow \mathbb{R}_{\geq 0}$ be the associated loss function that assigns a non-negative penalty to the output scores of the classifier. For a given image $\mathbf{x} \in \mathbb{R}^m$ and a target label $l \in \{0, \dots, k-1\}$, we want to find an $\mathbf{r} \in \mathbb{R}^m$ which minimizes $\|\mathbf{r}\|$ and that causes a misclassification, i.e. $f(\mathbf{x} + \mathbf{r}) = l$ with the additional constraint that $\mathbf{x} + \mathbf{r} \in [0, 1]^m$, i.e.

that image should be valid. Note that if $f(\mathbf{x}) = l$, the problem is trivial and $\mathbf{r} = \mathbf{0}^m$ is a solution. However, this optimization problem is in general hard, so we approximate it in the following way.

We find \mathbf{r} which minimizes $L_f(\mathbf{x} + \mathbf{r}, l) + \lambda \|\mathbf{r}\|$, subject to $\mathbf{x} + \mathbf{r} \in [0, 1]^m$. We use stochastic gradient descent to iteratively optimize \mathbf{r} . On the other hand, Szegedy et al. [1] used a box-constrained L-BFGS instead. Also, they minimize $L_f(\mathbf{x} + \mathbf{r}, l) + \lambda \|\mathbf{r}\|$, i.e. they use L1 regularization on \mathbf{r} but we found that it induces sparsity and produces more visually discernible images, which is undesirable. At the same time, the model misclassifies with a lower probability when the image is optimized using L1. Figure 1 provides an illustration. Note that while by misclassification we strictly mean that the majority vote goes to a non-true label class, in most of our experiments, the model is more than 90% confident about its false prediction.

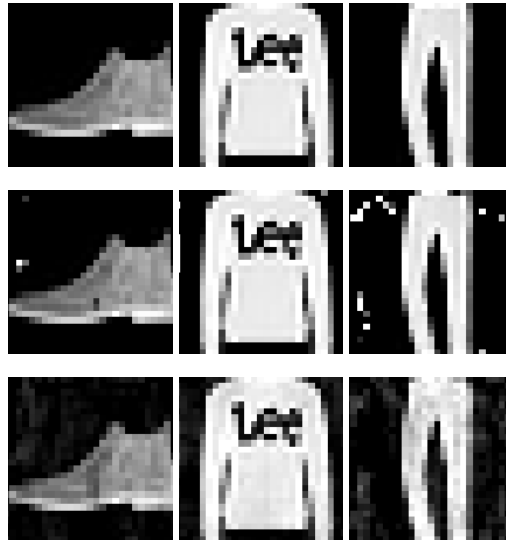

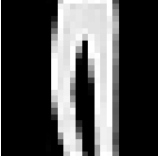

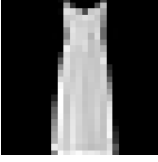
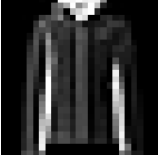


Figure 1: **L1 vs L2**. The top row shows original images, the middle row shows adversarial images generated by penalizing the perturbation using L1 norm while the bottom row shows L2. Notice how the sparsity induces visible artifacts in the middle row.

3 Experiments

3.1 Setting

We use the FashionMNIST dataset [2] for all our experiments. We use a simple neural network with two hidden layers having 64 and 128 neurons each. We train the model for 10 epochs using Adam optimizer [3] with learning rate $1e - 3$ and no weight decay. We also don't employ dropout [4] or batch normalization [5], to keep the models as similar to the original paper investigating adversarial images [1].

Label	Class	Image
0	T-shirt/top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	

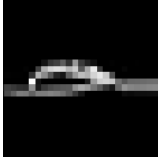

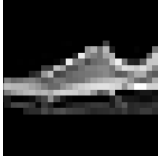
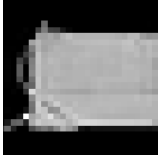
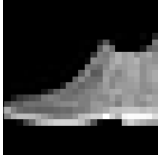
Label	Class	Image
5	Sandal	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boot	

Table 2: The FashionMNIST classes with a reference image each.

3.2 Ease of fooling across classes

A shirt can easily be mistaken for a coat, both are worn on the upper body and typically have sleeves. Similarly, an ankle boot can be mistaken as a sneaker, both are footwear. Such inter-class similarities may mean that it is easier to fool a model into predicting that an image that is very similar to a shirt is a coat. Formally, for similar classes y and y' , where y is the true label of \mathbf{x} , one might expect that small $\|r\|$ values are needed to classify $\mathbf{x} + \mathbf{r}$ as y' .

We thus perform an experiment to check this. The results can be found in figure 2, where we plot $\|r\|$ as a heatmap (darker corresponds to smaller values). Each row corresponds to a true label and each column to a false prediction. The diagonal entries are all black, since no deviation is needed for correct predictions. The reddish spots point to pairs (y, y') which are similar. While the results don't fully back our hypothesis, we can see that for the last row, which corresponds to a true label 9 (Ankle Boot), false predictions 5 (sandal) and 7 (sneaker) result in dark red squares. Similarly row 6 column 1 is dark, which corresponds to a shirt being misclassified as a t-shirt. You can find the full list of FashionMNIST classes in table 2. You can also see the actual values of $\|r\|$ in table 3 (these are averaged over three runs with

different random seeds).



Figure 2: **Ease of fooling across classes.** Darker means smaller values.

Truth	False prediction									
	0	1	2	3	4	5	6	7	8	9
0	0.0023	1.6139	2.1638	1.1949	2.7834	2.6848	2.4518	3.0366	1.6798	2.6239
1	1.1524	0.0009	2.8018	1.7477	1.9185	2.6613	2.7083	2.7918	1.7261	2.6451
2	2.3143	2.0963	0.0010	2.6581	1.6422	1.7365	1.2543	2.0705	1.4380	1.9254
3	1.6222	1.3426	2.6407	0.0008	3.0557	1.7109	2.0877	2.1065	1.1475	1.4403
4	1.1845	1.1348	1.3197	1.6567	0.0008	1.5766	1.0068	2.3123	1.0185	2.2364
5	8.9667	1.4028	1.2285	1.0370	3.1858	0.0007	1.2092	8.0931	8.7755	1.4825
6	5.0503	1.6537	1.7951	1.4856	1.8359	1.6728	0.3585	2.0930	1.3318	1.9036
7	1.5358	1.6330	2.2217	1.7052	2.4920	0.8168	1.5999	0.0007	9.7714	1.3531
8	2.4295	2.1820	2.9272	3.0535	3.1421	1.4479	1.3605	2.0637	0.0009	2.0840
9	1.9562	1.7071	1.7356	1.8056	1.5922	6.9085	1.1748	7.2961	0.8491	0.0615

Table 3: L2 norm of perturbations needed to misclassify images.

4 Conclusion

In this paper we demonstrated the existence of certain directions in neural networks’ representation space which lead them astray, i.e. adding perturbations in this direction leads the otherwise accurate model to confidently misclassify images. We analyzed the magnitude of perturbation needed to fool the model across classes. There are many interesting ideas which have yet to be explored. One is the density of these adversarial examples in the models’ representation space. This amounts to questioning the robustness of the adversarial images themselves, i.e. if we tweak the perturbation by a small amount in a random direction, do the images still fool the models? Another interesting direction is to see how prone other

models are to this sort of an adversarial attack - especially models trained in an unsupervised or self-supervised manner. We leave these ideas for future work.

References

- [1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.
- [2] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.
- [3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
- [5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.