

Generating images to fool a trained neural network using simple search algorithms

Abstract:

The paper addresses the problem of generating images that can "fool" a neural network (NN) trained for image classification. The NN is considered to be fooled if it assigns the generated image with high probability to one of its classes. The developed solution can be used for dataset augmentation with the goal of achieving better generalizations of the NN model. Also, the solution may be used for understanding complex NN models. The solution was verified by using NNs of different architectures and trained for different classification tasks, and it performed well in all cases.

Keywords:

Artificial neural network, convolutional neural network, machine learning, search algorithm, image generation

1. Motivation

An algorithm that can generate images that fool a specific neural network (that was trained for image classification) can potentially be used for several purposes. First of all, it can be used when evaluating a neural network's performance when it is presented with adversarial examples. The generated images can also be used for additional training of a neural network, and thus achieving better generalization. Also, the algorithm can prove helpful for better understanding the way a neural network learned to classify images.

2. Research questions

The goal of this paper is to present a simple and easy to implement algorithm that generates images that can fool a specific neural network into classifying the image as one of its classes. Several solutions for this problem already exist, one example would be the solution presented in *Explaining and harnessing adversarial examples* [1]. Another example is generative adversarial networks [2] that use image generation to train neural networks more effectively. The solution presented in this paper is less complex than the previous examples and shows that fooling a neural network does not require a complicated approach. Also, the described algorithm does not require detailed knowledge of the internal structure of the neural network in question (weights, number of layers, activation functions, etc.), while some of the other approaches rely on this information.

3. Methodology

The approach described in this paper is based on searching for a valid image that can fool the Neural Network through the space of all possible images of a specific dimension (specified by the Neural Network's input size). This means that there are a total of Bpp^n candidate images, where n is the total number of pixels multiplied by the number of channels, and Bpp represents the number of possible values that a pixel can have (this is usually 256 for standard 24-bit images). When looking at the internal structure of Neural Networks, it is possible to conclude that, when searching for a maximum output value for a particular class, only the pixel values of 0 or 255

need to be considered. This lowers the total number of candidate images to 2^n . In addition, the Neural Network itself can be used as a sort of a heuristic function while performing the search algorithm. Neural network models can be represented by simple mathematic models, which is one of the main principles this methodology relies upon.

4. Solution/Discussion

The solution presented in this paper is a very simple brute-force like algorithm, although any search algorithm could be used. The algorithm starts from an initial image (usually a zero image) and alternates pixel values between 0 and 255 one by one. If changing a value of a specific pixel increases the probability of the neural network assigning the image to a desired class then the change is kept, otherwise it is reverted. This simple algorithm was able to easily generate images that can fool fully connected neural networks in a single iteration (considering each pixel of the starting image only once). On the other hand, it needed several iterations over all pixels of the starting image in order to fool the Convolutional Neural Network. However, the algorithm always succeeds in finding the image that fools the network.

The algorithm was evaluated using three Neural Networks. The first one is a fully connect network, trained to recognized digits from the MNIST dataset. The second one is a Convolutional Neural Network also trained on the MNIST dataset. And the third one is fully connected and trained to detect human faces using the *Labeled faces in the Wild* dataset. Tables 1 and 2 contain examples of generated images that fooled the networks trained on the MNIST dataset, while Image 1 is an image that was generated to fool the network trained for face detection.





















Generated Image		Example from the same class in the MNIST Dataset	
			
			
			
			
			

Table 1 Some of the images that fooled the fully connected network trained on the MNIST dataset. Image size: 28x28





















Generated image			Example from the MNIST dataset			Generated image			Example from the MNIST dataset		
											
											
											
											
											

Table 2 Images that fooled the Convolutional Neural Network trained on the MNIST dataset. Image size: 28x28



Image 1 Generated image that fooled the NN for face detection. Image size: 32x32

5. References

- [1] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. *Explaining and harnessing adversarial examples*. 20 Mar 2015.
- [2] Ian J. Goodfellow , Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair , Aaron Courville, Yoshua Bengio. *Generative Adversarial Nets*. 11 Dec 2014