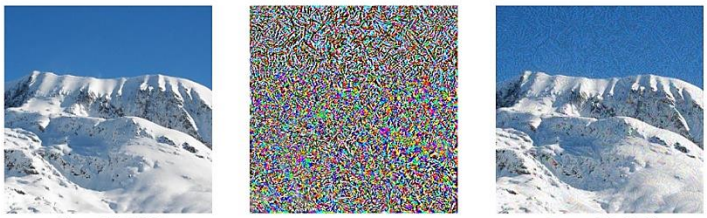
Term Project Proposal – Yarin Bar 0845029

Introduction

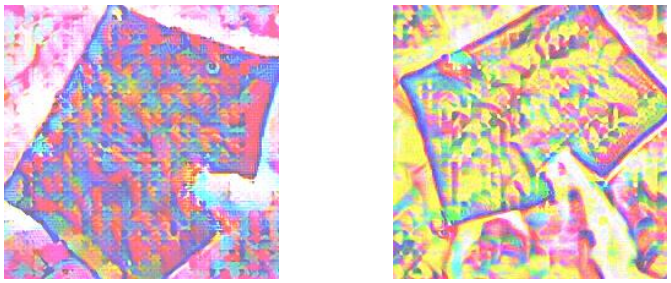
Deep learning (DL) or Deep Neural Network (DNNs) is paving its way into every technology and is rapidly gaining traction in every aspect of our lives; whether it is in smart cars, security systems, IOT (internet of things), cloud services and much more. But as recently discovered, DNNs are highly vulnerable to "adversarial attacks" (AAs) or "adversarial examples" (AEs). These correspond to a legitimate input, that has been slightly modified in order to "confuse" the Neural Net.



Due to the sensitivity to adversarial attacks, the future of DNNs as an underlying technology in our daily life, is in jeopardy. For instance, a face recognition security system is built on the premise that the DNN can discern between different faces.

Proposal

This field is extremely interesting in my opinion. In particular, I am interested in finding ways to tell if two images contain the same content after they are being modified. For example:



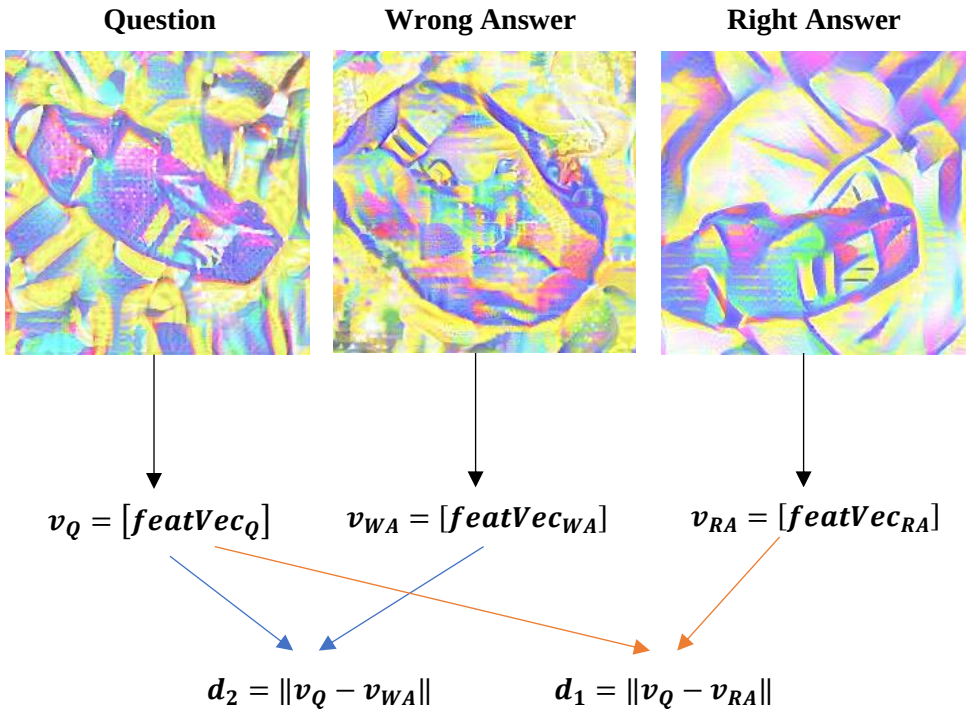
In the example above the two pictures contain the same content but they have been modified to trick NN to classify them as different pictures.

I propose to use a CNN to extract a feature vector from each picture and algebraically manipulating those vectors to try and decide if these two images contain the same content.

On way to do this is measuring the Euclidean distance between the two vectors and if the distance is lower than a certain threshold decide that the two images contain the same content. One problem with this method is that it might be too naïve and finding the threshold which gives us the best results can be challenging too.

Other method which seems more viable, but also a lot more complex, is training a Convolutional Neural Net to extract the most relevant feature vector based on training. The network will be presented with 3 images at a time; one question, one right answer and one wrong answer. The network will then tell us which of the images is closer to the question image.

For example:



Our goal is to ensure $d_1 = \|v_Q - v_{RA}\| < \|v_Q - v_{WA}\| = d_2$.

This is a well-known problem which is also called "KNN – K nearest neighbors". This method will be much more challenging as I need to construct my own NN and train it with custom made Loss function that I will need to invent and test.

I am somewhat skeptic it would be successful, but I am very interested to see how this idea works and even maybe get better idea on how to make it work better.

Application

We can use this algorithm to sort an array of attacked pictures starting from the closest match until the farthest picture. This algorithm can be used to crack a simple adversarial Captcha.

When presented with a question and 9 possible answers we can sort the candidate answers by their distance that has been calculated by our algorithm to try and pick the correct pictures and "break" the captcha (when the correct answers are the same picture as the question with different attacking methods).

Implementation

The neural network will exist as a separate, independent unit written in python. We will use sorting algorithm written in c++ for speed. The user will interact with our application using a python wrapper for maximal convenience.

Testing will be held in a python using unittest which will run all the tests and will also print the accuracy of the testing.

Please consider this illustration: