**Adversarial Machine Learning - Home Assignment 1**

This home assignment provides a “first taste” of adversarial learning. Its goal is to demonstrate two of the key aspects of adversarial perturbations. First, you will use manual tools to identify adversarial examples against a MNIST classifier, and then you will test the ‘transferability’ of those examples to a second classification model.

1. Train a simple neural network classifier over the MNIST dataset.
2. Randomly pick an image from the test set and make sure it is correctly classified by the model you have trained.
3. Pick a random pixel coordinate, and manually tune that pixel’s intensity until the classifier fails to correctly classify the resulting image.
4. Repeat step #3 above 4 more times with different pixel coordinates. Each time, start with the original image and change only the value of the chosen pixel.  
   Eventually you should have 5 wrongly classified images (adversarial perturbations), where each image differs from the original sample by just one pixel.
5. Train a second model using a different learning algorithm for classifying the MNIST dataset. For instance, you might use a linear SVM, or a Random Forest.
6. Test each of the 5 adversarial perturbations you have created in step 3,4 against the second classifier.

Submission Instructions

* Submit a MS-Word report accompanied by source code as either a Jupyter notebook or a plain python file.
* Make sure to include in your report at least the following details –
  + Implementation details of the two classifiers – algorithm, architecture, hyper parameters, accuracy over the training and testing set
  + The index and print-out of the test-set image you have chosen in step 2
  + Printouts of the 5 adversarial perturbations you have created
  + The class labels assigned to your perturbations by the two classification models
  + The L2 (Euclidean) and L0 (Hamming) distance of the perturbations from the original test image
* Make sure to include the names of both partners

Have fun

Zvika & Ziv