ECE1513H - Winter 2020 **Assignment 5**
University of Toronto - LEC0101 (Prof. Papernot) **Due Mar 30**

This assignment contains 4 questions worth a total of 14 points. Because assignments will help you learn things that are complementary to what we covered in class, they should be completed on your own. Otherwise, you will not learn from taking this course and you are harming yourself.

It is **mandatory** to hand in a Python script (i.e., file with extension `.py`) with this assignment, **otherwise it will be considered late**. You can generate a Python script automatically through the `File > Download .py` menu in Colab. Submitting a link to a Colab notebook does **not** constitute a valid submission.

Recall that you can attach a free GPU to your Colab notebook and accelerate your computations. Go to `Runtime > Change runtime type`, and choose GPU under "hardware accelerator".

Problem 1 In the previous assignment, you trained a CNN on MNIST with JAX and stax. In this problem, you can reuse this code as boilerplate. The goal of this assignment is to explore concepts we introduced in Lecture 8.

1. (4 points) In this question, you will search for an adversarial example. Let x be an image from the test set which is correctly classified by the model. To obtain an adversarial example, you should modify x into $x^* = x + \varepsilon \cdot \texttt{sign}(\frac{\partial L}{\partial x})$ where L is the cross-entropy loss and ε is a hyperparameter. You can set the hyperparameter to a value around $\varepsilon = 0.3$ for the purpose of this question. Using JAX, find a perturbed image x^* which is misclassified by the model while the original image x was originally correctly classified. For the original image, take an image of the class 7 from the test set. You are expected to hand in (1) the code used to find the perturbation, (2) a visualization of the perturbed image, and (3) the prediction vector output by the model on the original and perturbed image.

2. (2 points) Now repeat the process for $1,000$ images from the test set. Plot the average accuracy of the model on the $1,000$ adversarial examples x^* you produce as a function of ε. That is, you should produce a graph with the model's accuracy on the vertical axis and ε on the horizontal axis. Also hand in the python code used to generate this graph (you should use the matplotlib library to generate the plot).

3. (6 points) We will now refine the way the perturbation is found by adding several *smaller* perturbations to the image rather than modifying the image in one large perturbation. Modify the code you wrote in the first question to instead iteratively perturb the input as follows. For k iterations, take the input x, compute $x^* = x + \varepsilon \cdot \texttt{sign}(\frac{\partial L}{\partial x})$, replace x by x^* and repeat. Take the same test image than in the first question and show that you can find a misclassified perturbed image for $k = 5$ and a smaller value of the hyperparameter $\approx \frac{\varepsilon}{k}$. You are expected to hand in (1) the code used to find the perturbation, (2) a visualization of the perturbed image, and (3) the prediction vector output by the model on the original and perturbed image.

4. (2 points) For $k = 5$, plot the same graph than in the second question.

*
* *