# CSE 598 Machine Learning Security and Fairness.

## Instructions

- This homework is due at **11:59:59 p.m. on Monday Sep 26th, 2022.**
- The submission includes two parts:
    a. A single pdf file as your write-up, including your plots and answers to all the questions and key choices you made. The write-up must be an electronic version. No handwriting, including plotting questions. LATEX is recommended but not mandatory.
    b. a zip file including all your code, and files specified in questions with Submit, all under the same directory. You can submit Python code in either .py or .ipynb format.

## Python Environment

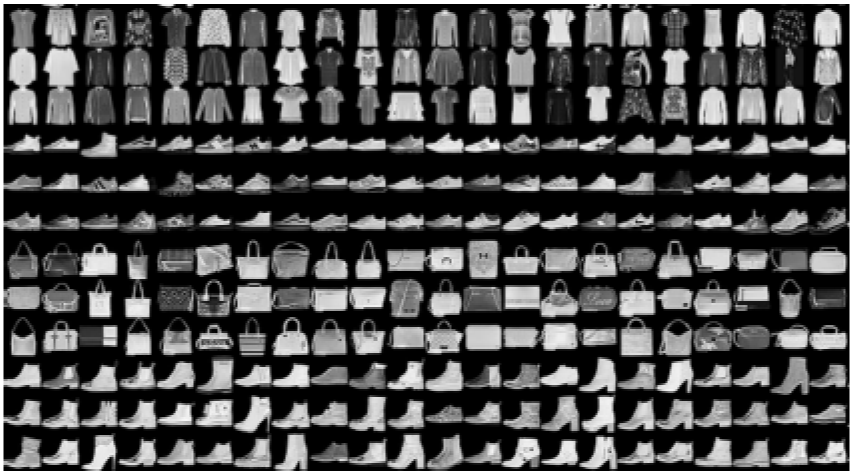
We are using Python 3.7 for this course. You can find references for the Python standard library here: https://docs.python.org/3.7/library/index.html. To make your life easier, we recommend you to install Anaconda for Python 3.7.x (https://www.anaconda.com/download/). This is a Python package manager that includes most of the modules you need for this course.

In this homework, you are required to use PyTorch for building and training neural networks. Install PyTorch as torch and torchvision for datasets. You will also need matplotlib.pyplot to visualize results. As a deep learning library, PyTorch performs backpropagation automatically for you and trains your network faster.

For the details about how to use pytorch, please refere to https://pytorch.org/ and https://pytorch.org/tutorials/beginner/basics/intro.html.

# Task1: Train a Fashion-MNIST Classification

In this part, you will implement and train Convolutional Neural Networks (ConvNets) in PyTorch to classify images.



The dataset we use is Fashion-MNIST dataset, which is available at https://github.com/zalandoresearch/fashion-mnist and in torchvision.datasets. Fashion-MNIST has 10 classes, 60000 training+validation images (we have splitted it to have 50000 training images and 10000 validation images, but you can change the numbers), and 10000 test images.

We have provided some starter code in part.py where you need to modify and experiment with the following:

- **The architecture of the network. Implement the LeNet.**
- **Define the optimizer (SGD, RMSProp, Adam, etc.) and its parameters.**
- **Training parameters (batch size and number of epochs)**

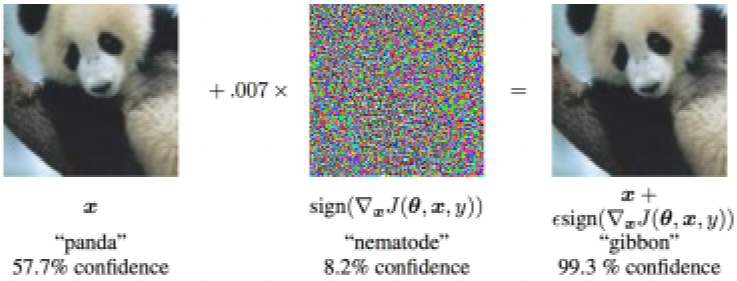
You should train your network on training set and change those listed above based on evaluation on the validation set. You should run evalution on the test set only once at the end.

**You could download the sample code from https://drive.google.com/file/d/1SFtPVy78i2eqn42Qt3oS_KuV-nEDLUvv/view?usp=sharing.**
**Complete the following:**
**1. Submit a program which trains with your best combination of optimizer and training parameters, and evaluates on the test set to report an accuracy at the end.**
**2. Report the detailed architecture of your best model. Include information on hyperparameters chosen for training and a plot showing both training and validation loss across iterations.**
**3. Report the accuracy of your best model on the test set. We expect you to achieve over 90%.**

**You need to save the models using torch.save for the following tasks**

# Task 2: Implement attacks for the Fashion-MNIST Classification

1. Implement a non-targeted white-box FGSM evasion attack (https://arxiv.org/abs/1412.6572) against the deep learning model from Task 1 (please use torch.load to load the checkpoint of the previous saved model). Applying the perturbation magnitude as 25/255 ($\epsilon$ = 25/255).



2. Implement a non-targeted white-box PGD evasion attack (https://arxiv.org/pdf/1706.06083.pdf) against the deep learning model from Task 1. Applying the perturbation magnitude as 25/255 ($\epsilon$ = 25/255). Please try to apply different attack steps including {1,2,5,10}

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\Big\{\boldsymbol{X}_N^{adv} + \alpha\,\text{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\big)\Big\}$$

**Complete the following:**

**Note that**: **You only need to attack the test data.**

**1. Submit a program including the above two attacks**

**2. Report the attack success rate of the FGSM attacks.**

**3. Report the attack success rate of the PGD attacks among different attack steps.**

**4. Randomly select 10 adversarial images of 10-step PGD attack and visualize them.**

Hints:

Useful resources :

https://medium.com/swlh/gradient-based-adversarial-attacks-an-introduction-526238660dc9#:~:text=The%20Projected%20Gradient%20Descent%20(PGD,initializes%20to%20the%20original%20point.