

# CPSC 585 - Artificial Neural Networks - Spring 2020

## Project 1, due March 11

*Last updated Saturday, February 22, 7:15 pm PST*

### Platforms

For this project (and, in general, for most machine learning or data science projects) you will need a [Jupyter notebook](#) with Python 3. Jupyter allows you to create documents mixing text, equations, code, and visualizations.

The Jupyter project itself [recommends Anaconda](#) if you intend to run notebooks locally on a laptop or desktop computer, but there are several cloud services that offer free Jupyter notebooks, including [Microsoft Azure Notebooks](#) and [Google Colaboratory](#).

### Libraries

You may write your code using standard Python 3, but you are encouraged to use [NumPy](#) to implement vector operations such as dot product. Use the [Matplotlib](#) library to generate plots. You may **not** use machine learning libraries such as scikit-learn.

### Dataset

The file [dataset.py](#) contains the letters A through Z as 5-by-7 dot-matrix fonts, in a format similar to the [Hebb Net example](#) for character recognition. There are two different fonts, one for training and one for test.

### Training Perceptrons

Implement one perceptron for each letter, for a total of 26. Each perceptron should learn to output 1 for its assigned letter and 0 for all other letters. Begin with weights and biases initialized to random values (rather than zero) for each perceptron, and apply the [perceptron learning algorithm](#) until all items in the training set are classified correctly or until it becomes clear that the weights will not converge.

As each item in the training set is classified by each perceptron, record whether the output was correct or not. At the end of each pass through the training set by a perceptron, record the error rate (number of misclassified items) of that perceptron for that epoch.

If a letter is linearly separable from the others and the learning rate is small enough, you should see a downward trend in error rate until each point is classified correctly. You may wish to use Matplotlib to [plot the error rate as a function of the number of epochs](#) in order to visualize this trend.

Q. Are all letters in the training set linearly separable?

## Testing the Learned Weights

Once the perceptrons have been trained (in other words, once the perceptrons for the letters which are linearly separable from the other letters have converged), test each trained perceptron against the letters in the test set.

Q. Are all letters in the test set correctly classified?

If any items in the test set are misclassified, compare them to the corresponding items in the training set.

Q. How similar are the misclassified items to the items in the training set?

## Submission

Submit your project by uploading your .ipynb file and any other relevant artifacts to the project1/ subdirectory of the folder that will be shared with you on Dropbox. You may work alone, or make a single submission for a team of 2-3 students. If you work in a team, only one submission is required, but for safety consider uploading copies to each team member's submission folder. (Make certain, however, that the copies are the same in each case; the instructor will not attempt to ascertain which is the "real" submission.)

A printed submission sheet will be provided on the due date. To finalize your submission, fill out the sheet with the requested information and hand it in to the professor by the end of class. Failure to follow any of these submission instructions exactly will incur a **10%** penalty on the project grade for all team members.