CPSC 585 - Artificial Neural Networks

Introductory Project, Spring 2023 - due March 4

This introductory project focuses on building your skills with Python and some of the libraries and tools you will need to work with neural network models. As such, this project is to be completed individually. Future projects will be completed in teams.

# Dataset

The [Optical recognition of handwritten digits dataset](https://scikit-learn.org/stable/datasets/toy_dataset.html#digits-dataset), included with the scikit-learn library, consists of 8✕8 grayscale images of hand-written digits sorted into classes 0-9. Once the dataset has been [loaded](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_digits), the feature vectors and labels are accessible as the data and targets attributes.

# Tasks

1. Split the dataset into training and test sets, then train a fully-connected neural network to recognize the digits. You will need to determine the size and number of layers, activation functions, and other hyperparameters. You can see a similar example using Keras as Problem 21 in the [Programming Exercises and Sample Code](https://www.cambridge.org/highereducation/books/the-science-of-deep-learning/23B3CE5B09590BD9E30474C850FA5358/resources/student-resources/41031AEA59090827210CEE5B3B2D7121/programming-exercises-and-sample-code/3B215D6121E694E5C614CA68213F8F1E).

Note that while it is possible to achieve 100% accuracy on the training set, this may not generalize to the same accuracy on the test set. Try varying the architecture and parameters to see how they influence the test score.

1. When you are satisfied with your model's performance, dump the trained weights and biases from the model as NumPy arrays for each layer. If you are using Keras, this is the [get\_weights](https://keras.io/api/layers/base_layer/#getweights-method) method. For PyTorch, you will need the model’s [state\_dict](https://pytorch.org/tutorials/beginner/saving_loading_models.html#what-is-a-state-dict) and the [numpy](https://pytorch.org/docs/stable/generated/torch.Tensor.numpy.html) method.

Once you have the weights, implement forward propagation using NumPy. You can see relevant examples in Problems 1 through 6 and 8 in the [Programming Exercises and Sample Code](https://www.cambridge.org/highereducation/books/the-science-of-deep-learning/23B3CE5B09590BD9E30474C850FA5358/resources/student-resources/41031AEA59090827210CEE5B3B2D7121/programming-exercises-and-sample-code/3B215D6121E694E5C614CA68213F8F1E).

1. Compute the training and test accuracies produced by your code in step (2) and compare it to those reported for the model in step (1). Verify your code’s predictions by plotting images for several results in each class.

## Platform

Perform the tasks above and document their results using a notebook on [Google Colab](https://colab.research.google.com/) with your @csu.fullerton.edu account. If you are not familiar with Google Colab or Jupyter Notebooks, the [Welcome To Colaboratory](https://colab.research.google.com/) notebook should help you get started. Note, in particular, the section [Using Accelerated Hardware](https://colab.research.google.com/#using-accelerated-hardware).

While you may choose to work locally, especially if you have access to a physical machine with a GPU, your project submission must be uploaded to Google Drive and run successfully in Colab.

## Libraries

You will need the following Python libraries:

* [scikit-learn](https://scikit-learn.org/) to obtain the dataset and split the data into training and test sets
* [NumPy](https://numpy.org/) to perform matrix and vector operations
* Either [Keras](https://keras.io/) or [PyTorch](https://pytorch.org/) to construct and train a neural network
* [Matplotlib](https://matplotlib.org/) to view images of the digits

# Documenting your results

Notebooks allow you to create documents mixing text, equations, code, and visualizations. Your project should make good use of these features. For example:

* Identify each task to be performed, documenting any decisions made.
* Include both the code to perform each task and its output. Where appropriate, tasks should be broken up into separate blocks, with the results shown for each.
* Include written analysis of results along with code output and visualizations.

In short, a reader unfamiliar with the project should be able to read your notebook and understand what you did and what results you obtained.

# Submission

From inside your Google Colab notebook, use the **Share** button at the top-right of the toolbar to [share your notebook](https://colab.research.google.com/notebooks/basic_features_overview.ipynb#scrollTo=aro-UJgUQSH1) with the professor:

1. Make certain that you are logged into Colaboratory with your @csu.fullerton.edu email address.
2. Add the professor’s @fullerton.edu email address as a **Viewer**.
3. Leave **General access** set to ***Restricted***, rather than setting it to ***Cal State Fullerton*** or ***Anyone with the link***.
4. Use the **Copy Link** button to copy the link to the clipboard.
5. [Submit the link](https://community.canvaslms.com/t5/Student-Guide/How-do-I-enter-a-URL-as-an-assignment-submission/ta-p/286) you copied via Canvas by the deadline.

## Grading

The project will be evaluated on the following five-point scale, inspired by the [general rubric](https://cs533.ekstrandom.net/f21/assignments/#general-rubric) used by Professor Michael Ekstrand at Boise State University:

**Exemplary (5 points)**

The project is a success. All requirements met. The quality of the work is high.

**Basically Correct (4 points)**

The project is an overall success, but some requirements are not met completely, or the quality of the work is inconsistent.

**Solid Start (3 points)**

The project is mostly finished, but some requirements are missing, or the quality of the work does not yet meet professional standards.

**Serious Issues (2 points)**

The project has fundamental issues in its implementation or quality.

**Did Something (1 point)**

The project was started but has not been completed enough to assess its quality fairly or is on the wrong track.

**Did Nothing (0 points)**

The project was not submitted, contained work belonging to someone else, or was of such low quality that there is nothing to assess.