CDS 1020-1: Introduction to Computational Data Science
Homework 4

François Chollet, founder of the deep learning library Keras, defines intelligence as "the ability to face the future given things you've seen in the past."[1] In a manner that reflects biology, neural networks mimic intelligence using "hidden" layers of functionality (incorporating activation functions) to generate an approximation of the data it receives. A type of neural network known as a convolutional neural network is known for its success in learning the class of an image and being able to scale to large images. Here we will conduct a comparison between two models, a logistic regression model and a convolutional neural network, in performing the task of image classification.

Assigned: Wednesday, April 19th, 2023, 11:59pm
Deadline: Wednesday, April 26th, 2023, 11:59pm

40 points

Part I: Exercises with PyTorch (10 pts)
Use the Torch library for the following exercises.

1. Create two random, 0-dimension tensors x and y that are Gaussian distributed with a zero mean unit variance.
2. Using the previous tensors, create a PyTorch object z that returns x + y if x > y, and x – y if otherwise.
3. To prepare for classification using PyTorch, specific columns are extracted from a dataset. Create a tensor called X, containing these hypothetical values (i.e., they can be random values or actual values from a dataset). The tensor should have 5 rows and 4 columns. Confirm the dimensions with a print statement.

Part II: Programming Exercise—Developing and Deploying Two Neural Networks (30 pts)
Here you will create and train models on the MNIST dataset for multiclass classification. For the input and output layers, the dimensions will be 784 and 10, respectively. The first network is a simple neural network or a logistic regression, consisting of two layers: a flattening layer and a linear layer. As these layers do not have learnable parameters, they are not considered hidden layers. The second network is a convolutional neural network, consisting of several hidden layers including convolutional layers. For both

[1] Kilcher, Yannic, et al. "#51 Francois Chollet - Intelligence and Generalization ." *Machine Learning Street Talk*, YouTube, 16 Apr. 2021.

models, the portions marked “Placeholder” require completion. Once complete, each model can be consecutively implemented. Compare performance in terms of accuracy and loss.

- **File:** one Jupyter notebook script or Python file. The “HW4_part_II” file contains the code for the models.
- **Data:** The MNIST dataset is built-in to PyTorch and accessible through the PyTorch module datasets. The data consists of 70,000 images of handwritten digits. The images are labeled with a number between 0 and 9, representing one of 10 classes. Each greyscale image has dimensions of 28 x 28 pixels; thus, it is a matrix M^{28x28}, consisting of 784 pixels.

A. Logistic regression for Classification of the MNIST Dataset:
   1. Substitute your code for every area marked Placeholder and complete the following questions. The names of some objects are used before assignment; therefore, when defining them in the script, maintain consistency with the names already used downstream. Several areas marked Placeholder, but not all, are described in detail below.
   2. Load the split MNIST dataset separately into training and testing data objects. PyTorch has built-in datasets such as the MNIST data that is pre-loaded in the class “datasets” from the torch package. Hint: there will be 4 arguments when loading the training and testing data. Show the type and contents of the data objects for training and testing.
   3. The PyTorch module Utils has a class called Dataloader for wrapping data, enabling setting batches, shuffling and multi-processing. Load each split dataset as a dataloaders object, specifying a batch size of 64 images and enable shuffling. Confirm it is a dataloaders object with a print statement.
   4. For the loss function, we will use Cross Entropy Loss from the PyTorch torch.nn package.
   5. For the optimization algorithm, we will use the algorithm Adam from the torch.optim package. Set the learning rate to 0.001. Hint: the first argument requires calling a method related to parameters for your model.
   6. Define the dimensions of the input and output layers. For the input, we are treating each image pixel as a feature.
   7. Generate visualizations for accuracy and loss over 10 epochs, then 25 epochs. Interpret the performance according to these metrics. Compare the two performances (i.e., their similarities and differences). Hint: a list data structure can be used.

B. Convolutional Neural Network for Classification of the MNIST Dataset
   1. Substitute your code for every area marked Placeholder and complete the following questions. The names of some objects are used before assignment; thus when defining them in the script, maintain the names

already given downstream. Several areas marked Placeholder, but not all, are described in detail below.

2. In the same way as with the first model, download the data, define the optimization algorithm, and define the loss function.
3. Complete the code for the remaining areas marked Placeholder.
4. As with the first model, generate visualizations for accuracy and loss over 10 epochs, then 25 epochs. Interpret the performance according to these metrics. Compare the two performances (i.e., their similarities and differences).
5. How does the performance between the two models, the logistic regression and convolutional neural network, compare? Briefly explain any similarities and differences.

Submission:

Submit two completed standard Python files (.py): one for Part I and the other for Part II. For Part I, the file name will be “HW4_part_I.” For Part II, use the existing file name. A script is only accepted in a python file. Code will not be graded if it cannot execute. For visualization and text answers, please use a document (.pdf) with appropriate numbering and the file name “HW4_solutions.” The entire submission must be compressed in a folder called “[last name]_HW4.” The submission to Canvas will consist of this folder.