MNIST Image Classification, How to Write a Neural Network

@some1and2

Table of Contents

Purpose

The goal of this project is to be able use python to interpret the value of a hand written digit from the `MNIST` dataset. This is going to be done by the use of a neural network trained using the program. The completed Neural Network should be either saved using the python "`pickle`" library (a library for object serialization, gives the functionality of saving variables as files) or by simply saving the weights and biases of the neural network to external files (this decision will be largely based on file size and load performance).

Road Blocks

Some of the problems that must be faced in this project are getting the data and understanding the mechanics of neural networks enough to be able to actually get the program to function, as well as being able to save the model. Getting the data from the original source is slightly more difficult than expected as the official website for getting the data is password protected. For the mathematics, with a background in basic calculus some of the notation found from most sources can be understood, however getting a sufficient understanding to be able to replicate the functionality of the original sources will be a challenge. Saving the weights and biases of the AI will also have to be done, this should be relatively straight forward however because most of the values will be stored in a "`numpy`" array and they have a built in function to save values as files. This is also not a huge concern because python classes (which is the context the AI is going to run from) can be "`pickled`", meaning the class itself can be saved as a file a loaded whenever is necessary.

Getting the Dataset

Getting the Dataset is the first problem to be addressed. To get the data set, the dataset file first has to be downloaded. While there are multiple sources this can be acquired from, the program gets the dataset from the same source that the library Keras gets theirs from (which is this URL). The file is downloaded then the SHA256 checksum algorithm checks to verify that the data is correct. From there the file is unzipped into the "AI/Datasets" folder by using the "`zipfile.ZipFile()`" function. This however does required the file is saved so to prevent writing to disk then reading immediately, the "`io.BytesIO()`" function is used.

Citations

Keras-Team. (2022, August 5). Keras/mnist.py at *master – Keras-Team/Keras.* GitHub. Retrieved March 29, 2023, from

https://github.com/keras-team/keras/blob/master/keras/datasets/mnist.py