ADTA 5550: Deep Learning with Big Data

Assignment 4

1. Overview

TensorFlow was created by the Google Brain team. TensorFlow is an open-source library for numerical computation and large-scale artificial intelligence (AI) machine learning and deep learning projects. TensorFlow bundles together a broad spectrum of machine learning and deep learning models. It uses Python to provide a convenient front-end API for building applications with the framework while executing those applications in high-performance C++.

IMPORTANT NOTES:
--) If MS Words is the document format required for submission, the student is required to submit the contents as MS Words, ***not*** *submit PDF documents.*

IMPORTANT NOTES:
--) If Jupyter Notebook documents are required in a Submission Requirement, the student is required to submit the Jupyter Notebook documents, i.e., ****.ipynb documents***, ***not*** *python programs, i.e.,* ***not*** **.py programs.*

IMPORTANT NOTES:
As discussed in the syllabus, the course curriculum covers two AI frameworks: TensorFlow and Keras.
--) The ***first half*** *of the course (until midterm) focuses on* ***Keras****.*
--) The ***second half*** *of the course (including the final project) focuses on* ***TensorFlow****.*

--) While working on ***HW 4****, it is* ***required*** *that the student should use the* ***TensorFlow (Version 1.xx) directly*** *as the AI framework to build the neural network (as shown in the lecture videos of WEEK 5 and WEEK 6).*

--) In other words, the student should ***not*** *use Keras APIs, e.g., Keras sequential API, to build the neural network.*

IMPORTANT NOTES:
*--) In HW 4, the student is required to use "****STEP****" (in the code shown in the videos of WEEK 5 and WEEK 6),* ***not*** *"EPOCH," in training the neural network.*

- ***STEP****: Each time an input batch is fed into the model during the training*
- ***EPOCH****: Each time all the input batches (split from the training data set) have been fed into the model during the training*

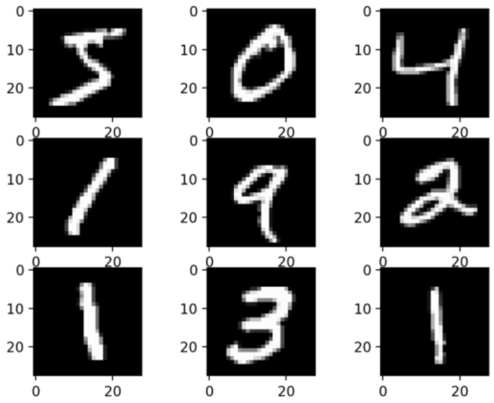
2. Data Sets: MNIST

(*Source: The MNIST DATABASE: http://yann.lecun.com/exdb/mnist/*)

The MNIST dataset of handwritten digits, available from this page, has a training set of 60,000 examples and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

The original black and white (bi-level) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels and translating the image so as to position this point at the center of the 28x28 field.

The MNIST database was constructed from NIST's Special Database 3 and Special Database 1, which contain binary images of handwritten digits. NIST originally designated SD-3 as their training set and SD-1 as their test set. However, SD-3 is much cleaner and easier to recognize than SD-1. The reason for this can be found in the fact that SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. Drawing sensible conclusions from learning experiments requires that the result be independent of the choice of the training set and test among the complete set of samples. Therefore it was necessary to build a new database by mixing NIST's datasets.



IMPORTANT NOTES:

*--) As shown in the lecture videos, the data set **MNIST is obtained directly from TensorFlow code**. The student does not need to download the dataset from any external source.*

3. PART I: Build, Train, and Test CNN on MNIST Dataset (60 Points)

TO-DO
--) Rework on the project of building, training, and testing the **convolutional neural network** on the **MNIST** dataset using the TensorFlow AI framework and Python, as discussed **in the video lectures**.

--) After building, training, and testing the model, copy the results of the tests into a section of an MS Docx file named "**Project Report – CNN on MNIST with TensorFlow**."

--) Write a report on the results of testing the model (in the same project report file)

SUBMISSION REQUIREMENTS #1:
The student is required to submit the following work items:
--) The design of the model including all the layers (in the above project report file)
--) The results of testing the model (in the above project report file).
--) The report on the results of the test (in the above project report file)
--) The Jupyter Notebook document that contains all the code of building, training, and testing the model.

4. PART II: Redesign Convolutional Neural Network (30 Points)

TO-DO
--) **Update the design** of the convolutional neural network by using **only one** **convolution layer**.

--) Redesign the model with all the layers

--) **Update the code accordingly** to build, train, and test the model in a **separate** Jupyter Notebook document.

--) After building, training, and testing the model, copy the results of the tests into **another section** of the project report file that has been created in PART I.

--) Write a report on the results of testing the model (in the above project report file)

SUBMISSION REQUIREMENTS #2:
The student is required to submit the following work items:
--) The design of the **updated model** including all the layers (in the above project report file)
--) The results of testing the model (in the above project report file)
--) The report on the results of the test (in the above project report file)
--) The Jupyter Notebook document that contains all the code of building, training, and testing the model.

5. PART III: Update Number of Steps of Training CNN Model (10 Points)

TO-DO
--) **Update the code of training** the **updated model** (in PART II) by changing the number of steps of training to 3000 (Three Thousand) times.

--) **Update the code accordingly** to build, train, and test the model in a **separate** Jupyter Notebook document.

--) After building, training, and testing the model, copy the results into **another section** of the project report file that has been created in PART III.

--) Write a report on the results of testing the model (in the above project report file)

SUBMISSION REQUIREMENTS #3:

The student is required to submit the following work items:
--) The results of testing the model (in the above project report file)
--) The report on the results of the test (in the above project report file)
--) The Jupyter Notebook document that contains all the code of building, training, and testing the model.

6. HOWTO Submit

The student is required to submit all the required documents – Microsoft Word **and** Jupyter Notebook documents – to Canvas

Due date & time: 11:59 PM – Thursday 07/11/2024

The student is required

- To submit an MS Words document for the report (**not** PDF)
- To submit a **native Jupyter Notebook** document for the code in PART I
- To submit a **native Jupyter Notebook** document for the code in PART II
- To submit a **native Jupyter Notebook** document for the code in PART III