ADTA 5550: Deep Learning with Big Data

Thuan L Nguyen, PhD

Assignment 3

1. Overview

1.1 Keras: Another Popular AI Framework for Deep Learning

Keras is an open-source neural network library written in Python.

- It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, and other AI framework.
- It is designed to enable fast experimentation with deep neural networks; it focuses on being user-friendly, modular, and extensible.
- It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).
- Its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library.
Chollet explained that Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used.

1.2 TensorFlow

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++.

1.3 Multilayer Perceptrons (MLPs): Fully Connected Neural Networks

A multilayer perceptron (MLP) is a deep, artificial neural network.

An MLP is composed of:

- An input layer to receive the signal
- An output layer that makes a decision or prediction about the input
- And an arbitrary number of hidden layers (between the input layer and output layer) that are considered as the computational engine of the MLP.

Multilayer perceptrons are often used for supervised learning problems

- They train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs
- Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error.

2. Data Set

Title: Iris Plants Database
Sources:
(a) Creator: R.A. Fisher
(b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
(c) Date: July 1988

Relevant Information:

- The data set contains three classes of 50 instances each; each class refers to a type of iris plant.
- Predicted attribute: class of iris plant.
- Number of Instances: 150 (50 in each of three classes)
- Number of predictors: 4 numeric, predictive attributes and the class

Attribute Information:

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class: Iris Setosa, Iris Versicolour, Iris Virginica

IMPORTANT NOTES:
--) The student needs to download the data set Iris.csv from the Canvas module: .../DATA_SETS, and then upload it to the remote deep learning server

--) The steps of how to upload the data file to the remote server are discussed in the document:
***HOWTO_upload_files_to_remote_server_using_GCP_SSH**.pdf (Canvas module: .../SW_DOCS)*

--) In one of the first steps of coding, the student needs to load the data set (Iris.csv) into a Pandas data frame. The details of how to do this are discussed in the document:
***HOWTO_load_dataset_into_dataframe_in_remote_server.pdf** (Canvas module: .../SW_DOCS)*

*--) For Python codes, the student is required to **run the code** and **submit the Jupyter Notebook document** of its native format, i.e., *ipynb file, that contains the results. The student should **not** copy the results of Python code into the MS Word document.*

3. PART I: One-Hot Encoding (20 Points)

TO-DO

It is assumed that a dataset that has a class attribute (used for a classification problem) has **10 classes** that are recorded as numeric values: **0, 1, 2, .. 9**. However, these numeric values are assigned to the classes arbitrarily. There is **no** ordinal ranking meaning among them. It is similar to the case of the values 0 and 1 that are used to represent NO (or NEGATIVE) and YES (or POSITIVE).

The dataset will be used in a deep learning project.

Question 1.1:
Using critical thinking and based on the lectures, is it necessary to perform any kind of coding (integer coding or one-hot coding) on these class values for better performance of the project?

Question 1.2:
If the answer to Question 1.1 is "YES," what type(s) of encoding needs to be done to process the class values before using the dataset for the deep learning project?

Question 1.3:
Based on the answer to Question 1.2, explain the steps of what needs to be done for each type of encoding.

Question 1.4:
Based on the answer to Question 1.2, perform the necessary encoding tasks to transform the class values before using the dataset for the deep learning project.

SUBMISSION REQUIREMENT #1

--) Provide the answers/solutions to each of the above questions.
--) Display the results of the encoding task.
--) All contents are saved in an MS DOCS document named "**Assignment 3 – MLPs on Iris with Keras.**"

4. PART II: MLPs (Fully Connected Neural Networks) with Keras (50 Points)

TO-DO

Based on the lectures:
--) First, **design** an MLP neural network (the same as discussed in the lectures)
--) Using MS PowerPoint or Draw Tool in MS Words to draw a diagram of the neural networks with all the layers, the neurons, and the feed-forwarding connections.
--) **Redo all the steps** of the project "MLP on Iris with Keras" in a Jupyter Notebook document.

SUBMISSION REQUIREMENT #2

--) Add one section to discuss the design of the MLP into the aforementioned MS Words document: "**Assignment 3 – MLPs on Iris with Keras,"** including the **diagram** of the neural network.
--) Run all the steps of the project in the Jupyter Notebook document to get the results of each step.
--) Add another section to the MS DOCS document: "**Assignment 3 – MLPs on Iris with Keras**" to discuss the results obtained from the project.

5. PART III: Redesign the MLP (30 Points)

TO-DO
Based on the lectures:
--) First, **redesign** the above MLP neural network by **adding one hidden dense layer**, i.e., the MLP now has three layers: Input layer (NOT counted), 2 hidden Layers, output layer.
--) Using MS PowerPoint or Draw Tool in MS Words to draw a diagram of the redesigned neural networks with all the layers, the neurons, and the feed-forwarding connections.
--) **Redo all the steps** of the project "MLP on Iris with Keras" in **another** Jupyter Notebook document.

SUBMISSION REQUIREMENT #3

--) Add one section to discuss the new design of the MLP into the aforementioned MS DOCS document: "**Assignment 3 – MLPs on Iris with Keras,"** including the **diagram** of the neural network.
--) Run all the steps of the project in the Jupyter Notebook document to get the results of each step.
--) Add another section to the MS DOCS document: "**Assignment 3 – MLPs on Iris with Keras**" to discuss the results obtained from the updated project with the new design, especially comparing them with those from PART II.

6. HOWTO Submit

Due date & time: 11:59 PM – Thursday 06/27/2024

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