ADTA 5550.701: Deep Learning with Big Data

Thuan L Nguyen, PhD

Assignment 2

1. Overview

1.1 Linear Algebra for Deep Learning

In mathematics, linear Algebra is a branch that aims to describe coordinates and interactions of planes in higher dimensions and perform operations on them.

Linear algebra can be considered as an extension of algebra (dealing with unknowns) into an arbitrary number of dimensions, which provides methods for solving linear systems of equations. Rather than working with scalars, the focus is on vectors and matrices (vectors are just a special type of matrix).

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++.

2. PART I: Biological Neural Network & Artificial Neural Network (30 Points)

Question 1.1:
Describe (including **images** for illustration) the human biological neural network and how it works

Question 1.2:
Describe (including **images** for illustration) the McCulloch-Pitt neuron model, a.k.a. Threshold Logic Unit, that is considered as the simplest neural network and how it works.

Question 1.3:
Discuss (including **images** for illustration) how the pioneers in the AI field did imitate the human biological brain system to conceive the first artificial neural networks.

SUBMISSION REQUIREMENT #1

--) The answers to the above questions

3. PART II: Linear Algebra for Deep Learning: Matrices (20 Points)

TO-DO

Given the following matrices:

$$A = \begin{bmatrix} 5 & 3 & 8 \\ 2 & -1 & 7 \end{bmatrix} \qquad B = ?$$

--) Provide a matrix with all the scalar elements for B and perform the matrix multiplication A x B

SUBMISSION REQUIREMENT #2

--) Matrix B with all its scalar elements
--) The result of C = A x B
--) Specify the dimensions of C
--) Explain how to get the dot product C = A x B

4. PART III: Linear Algebra for Deep Learning: Matrices (30 Points)

TO-DO

Given the following matrix as a 2D array:

$$\begin{bmatrix} 2 & 1 & 3 & 4 & 5 \\ 0 & 0 & 1 & 4 & 2 \\ 4 & 2 & 6 & 8 & 10 \\ 6 & 3 & 14 & 35 & 33 \end{bmatrix}$$

--) Question 3.1:
Let's consider this matrix as a vector of vectors.
How many vector elements does this matrix have?
Show each vector element, one by one.

--) Question 3.2:
Let's consider this matrix as a vector of vectors.
Add 3 to the vector element (of the matrix) of the index = 1. The addition is performed **element-wise** along **Axis 1**.
Display the matrix with all its scalar elements after the operation has been done in the format of a 2D-matrix.

--) Question 3.3:
Continuing from Question 3.2, i.e., after the above addition of 3 has been done:
Flatten the matrix and display the result.

SUBMISSION REQUIREMENT #3

Provide solutions to all the above questions.

5. PART IV: TensorFlow Code in Jupyter Notebook (20 Points)

TO-DO
--) Write TensorFlow code to provide the solutions to the following simple problems:

1. Declare two constant tensors that have the values of 15 and 45. Add these two tensors and print out the results.

2. Declare two variable tensors, a and b, that are initialized with scalar values of 2.75 and 8.5. Find their product and print out the result.

3. Create two placeholders: x and y - that are both scalars of 32-bit floats. Assign 5.25 to x and 12.6 to y, multiply them together, and print out the results.

4. Create one placeholder: z - that is an N-Dimensional array (N can be >= 1) that can have any shape (shape = None). Feed this vector [1, 3, 5, 7, 9] into z and multiply it by 3. Display the results.

5. Create a **constant** tensor that is a matrix of the shape (8, 8). The matrix is initialized with all ones (1). Create a **variable** tensor that is also a matrix of the shape (8, 8) and initialized with random integer values between 0 and 99. Add these two tensors and display the results.

IMPORTANT NOTES:
--) Write the code of each problem in a separate Jupyter Notebook cell in ONLY ONE Jupyter Notebook document, **not** in separate documents.

SUBMISSION REQUIREMENT #4:
--) Run the code of each problem
--) Submit the Jupyter Notebook document

6. HOWTO Submit

The student is required to submit the Microsoft Words document and the Jupyter Notebook document by sending them to the instructor (Thuan.Nguyen@unt.edu) as attachments to a UNT email.

The subject of the email must be: "ADTA 5550: Assignment 2 – Submission."

Due date & time: 11:00 PM – Wednesday 06/17/2020