Become a Machine Learning expert in 6 easy-to-understand tutorials.

By Peter A Noble PhD
Email: panoble2017@gmail.com Web: http://peteranoble.com

The five tutorials outline how to build a large Artificial Neural Network (ANN) model using basic Pytorch building blocks. The sixth tutorial shows how to assess model performance using R program. The tutorials were designed to flow from one tutorial to another, starting from a simple model and ending with a complex one. I have marked up the scripts (1 to 4) to show which lines were modified/ removed and which ones were added to. Script 5 shows how to build a classifier by modifying script 4. Script 6 deals with assessing model performance using Area Under Curve –Receiver Operating Characteristic curves in R.

The purpose is to help users not familiar with Pytorch build ANN models for themselves and assumes a basic understanding of ANN models. I have provided Powerpoint presentations at each of the 4 steps, along with the final Pytorch scripts and example data. In theory, the user should be able to run the scripts on their computers – however some assumptions were made (below).

Although Pytorch can handle very large data sets, it is currently not possible to train an ANN model using a data set consisting of 12,849 columns and millions of rows (too much memory/space). For demonstration purposes, I have provided a fraction of the data.

Assumptions

(i) The example data are normalized to a minimum value of zero and a maximum value of one. I have purposely *not* provided the Python code to normalize/de-normalize the data because the procedures takes up too much memory and normalization can be better handled outside the Pytorch program (in C++). Apparently, Pytorch keeps all records in memory; therefore, minimizing data handling yields more memory for training.

(ii) The inputs (x's) of the training/testing data are in all columns except the last one, which contains an output (y's). That is, each row of data is an individual record with multiple inputs and one output. Users will have to adjust the number of inputs to analyze their own data.

(iii) The tutorials assume the User has access to a Virtual Machine (presumably in the Azure Cloud) with Juypter notebooks. While the first three tutorials depend on CPUs (not GPUs), the last one depends on a Virtual Machine with GPUs.

(iv) Of note, the user might have to adjust the Pytorch programs to the size of their Virtual Machines. The tutorials were made using a Virtual Machine that had 4 GPUs with 224 GB of memory, and about a Terabyte disk space.