STUDENT PREDICTION USING NEURAL NETWORKS

A Project Report submitted in partial fulfilment for the award of

MBA (FinTech)

Submitted by

Sai Pratyusha Pusarla (121923901008)

Under the guidance of

Mr. DivakarAllavarapu

Assistant Professor

****

GITAM INSTITUTE OF MANAGEMENT

GITAM UNIVERSITY

(U/s 3 of UGC Act)

VISAKHAPATNAM

(2019-21)

DECLARATION

We, the undersigned, hereby declare that, the project titled “Student prediction using neural networks” submitted to GITAM Institute of Management, GITAM University for the award of the Masters of Business Administration (FinTech) is the original work done by us, under the guidance of Mr.Divakar Allavarapu, Assistant Professor, GITAM Institute of Management.

The empirical findings in this report are based on the data collected by us. It was not copied from any other project. The project has not been submitted to any Substitute/University for the award of any Diploma/Degree.

Date: -

Place: - Visakhapatnam Sai Pratyusha Pusarla (121923901008)

CERTIFICATE

This is to certify that the project entitled, “Student prediction using neural networks”, is a bonafide work done by SAI PRATYUSHA PUSARLA(121923901008) is submitted in partial fulfillment for the Masters of Business Administration (Fintech) of GITAM University. It has not been submitted for the award of any diploma/degree in any other Institution/University.

Date: Mr.DivakarAllavarapu

Place: Visakhapatnam Assistant Professor

1.Introduction

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence and stated that “it gives computers the ability to learn without being explicitly programmed”.

And in 1997, Tom Mitchell gave a “well-posed” mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

To understand machine learning in terms of layman let’s consider a example that u are trying to throw a paper into dustbin After first attempt, you realize that you have put too much force in it. After second attempt, you realize you are closer to target but you need to increase your throw angle. What is happening here is basically after every throw we are learning something and improving the end result. We are programmed to learn from our experience.

This implies that the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing’s proposal in his paper “Computing Machinery and Intelligence”, in which the question “Can machines think?” is replaced with the question “Can machines do what we (as thinking entities) can do?”

Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to “produce reliable, repeatable decisions and results” and uncover “hidden insights” through learning from historical relationships and trends in the data set(input).

Suppose that you decide to check out that offer for a vacation . You browse through the travel agency website and search for a hotel. When you look at a specific hotel, just below the hotel description there is a section titled “You might also like these hotels”. This is a common use case of Machine Learning called “Recommendation Engine”. Again, many data points were used to train a model in order to predict what will be the best hotels to show you under that section, based on a lot of information they already know about you.

So if you want your program to predict, for example, traffic patterns at a busy intersection (task T), you can run it through a machine learning algorithm with data about past traffic patterns (experience E) and, if it has successfully “learned”, it will then do better at predicting future traffic patterns (performance measure P).

The highly complex nature of many real-world problems, though, often means that inventing specialized algorithms that will solve them perfectly every time is impractical, if not impossible. Examples of machine learning problems include, “Is this cancer?”, “Which of these people are good friends with each other?”, “Will this person like this movie?” such problems are excellent targets for Machine Learning, and in fact machine learning has been applied such problems with great success.

**Classification of Machine Learning**

Machine learning implementations are classified into three major categories, depending on the nature of the learning “signal” or “response” available to a learning system which are as follows:-

**Supervised learning** : When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

**Unsupervised learning** :Whereas when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

**Reinforcement learning** : When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error.

Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others. An interesting example of reinforcement learning occurs when computers learn to play video games by themselves.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google DeepMind has created a reinforcement learning program that plays old Atari’s video games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.

Semi-supervised learning : where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

**Categorizing on the basis of required Output**

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

**Classification** : When inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are “spam” and “not spam”.

**Regression** : Which is also a supervised problem, A case when the outputs are continuous rather than discrete.

**Clustering**: When a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

Machine Learning comes into the picture when problems cannot be solved by means of typical approaches.

**2.PYTHON:**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

**Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects**.**

**Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**Python Features**

Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

**Python Modules**

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

Example

The Python code for a module named aname normally resides in a file named aname.py.

Here's an example of a simple module, support.py

def print\_func( par ):

print "Hello : ", par

return

The import Statement

You can use any Python source file as a module by executing an import

statement in some other Python source file. The import has the following

syntax −

import module1[, module2[,... moduleN]

When the interpreter encounters an import statement, it imports the module if the module is present in the search path. A search path is a list of directories that the interpreter searches before importing a module. For example, to import the module support.py, you need to put the following command at the top of the script −#!/usr/bin/python

# Import module support

import support

# Now you can call defined function that module as follows

support.print\_func("Zara")

When the above code is executed, it produces the following result −

Hello : Zara

A module is loaded only once, regardless of the number of times it is imported. This prevents the module execution from happening over and over again if multiple imports occur.

The from...import Statement

Python's from statement lets you import specific attributes from a module into the current namespace. The from...import has the following syntax −

from modname import name1[, name2[, ... nameN]]

For example, to import the function fibonacci from the module fib, use the following statement −

from fib import fibonacci

This statement does not import the entire module fib into the current namespace; it just introduces the item fibonacci from the module fib into the global symbol table of the importing module.

The from...import \* Statement

It is also possible to import all names from a module into the current namespace by using the following import statement −

from modname import \*

**2.1 PYTHON PANDAS**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Key Features of Pandas**

* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**Python Pandas - Environment Setup**

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, pip.

pip install pandas (Note:-If anaconda is installed in your computer pandas package is already been installed)

**python nympy:**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

**Operations using NumPy**

Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

**NumPy – A Replacement for MatLab**

NumPy is often used along with packages like SciPy (Scientific Python) and Mat−plotlib (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language.It is open source, which is an added advantage of NumPy.

**NumPy - Environment**

Standard Python distribution doesn't come bundled with NumPy module. A lightweight alternative is to install NumPy using popular Python package installer, pip.

pip install numpy

PYTHON KERAS:

KERAS is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer.

Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend. So Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano.

Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

**BACKEND:**

Backend is a term in Keras that performs all low-level computation such as tensor products, convolutions and many other things with the help of other libraries such as Tensorflow or Theano. So, the "backend engine" will perform the computation and development of the models. Tensorflow is the default "backend engine" but we can change it in the configuration**.**

**THEANO,TENSERFLOW AND CNTK BACKEND:**

Theano is an open source project that was developed by the MILA group at the University of Montreal, Quebec, Canada. It was the first widely used Framework. It is a Python library that helps in multi-dimensional arrays for mathematical operations using Numpy or Scipy. Theano can use GPUs for faster computation, it also can automatically build symbolic graphs for computing gradients. On its website, Theano claims that it can recognize numerically unstable expressions and compute them with more stable algorithms, this is very useful for our unstable expressions**.**

**TENSERFLOW:**

Tensorflow is the rising star in deep learning framework. Developed by Google's Brain team it is the most popular deep learning tool. With a lot of features, and researchers contribute to help develop this framework for deep learning purposes.

**CNTK BACKEND:**

Keras is The Microsoft Cognitive Toolkit or CNTK. It is an open-source deep learning framework that was developed by Microsoft Team. It can run on multi GPUs or multi-machine for training deep learning model on a massive scale. In some cases, CNTK was reported faster than other frameworks such as Tensorflow or Theano.

**Advantages of Keras**

* Fast Deployment and Easy to understand

Keras is very quick to make a network model. If you want to make a simple network model with a few lines

* Large Community Support

There are lots of AI communities that use Keras for their Deep Learning framework. Many of them publish their codes as well tutorial to the general public.

* Have multiple Backends

You can choose Tensorflow, CNTK, and Theano as your backend with Keras. You can choose a different backend for different projects depending on your needs. Each backend has its own unique advantage.

* Cross-Platform and Easy Model Deployment

With a variety of supported devices and platforms, you can deploy Keras on any device like

iOS with CoreML

Android with Tensorflow Android,

Web browser with .js support

Cloud engine

Raspberry Pi

* Multi GPUs Support

You can train Keras with on a single GPU or use multiple GPUs at once. Because Keras has a built-in support for data parallelism so it can process large volumes of data and speed up the time needed to train it.

**Disadvantages of Keras**

* Cannot handle low-level API

Keras only handles high-level API which runs on top other framework or backend engine such as Tensorflow, Theano, or CNTK. So it's not very useful if you want to make your own abstract layer for your research purposes because Keras already have pre-configured layers.

SKLEARN:

Scikit-learn is an open source Python library for machine learning. The library supports state-of-the-art algorithms such as KNN, XGBoost, random forest, SVM among others. It is built on top of Numpy. Scikit-learn is widely used in kaggle competition as well as prominent tech companies. Scikit-learn helps in preprocessing, dimensionality reduction(parameter selection), classification, regression, clustering, and model selection.

**Important features of scikit-learn:**

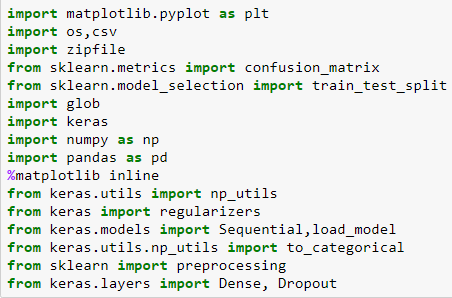
* Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
* Accessible to everybody and reusable in various contexts.
* Built on the top of NumPy, SciPy, and matplotlib.
* Open source, commercially usable – BSD license.

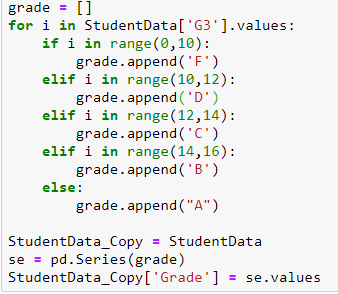
STEP BY STEP EXPLANATION OF PROJECT:

* Install keras using pip install in anaconda prompt using

Conda install –c conda-forge keras

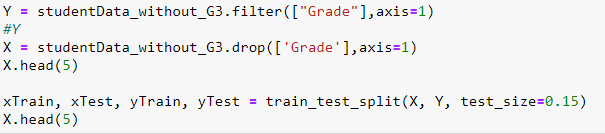
* Gather a student dataset from internet to train and test the model and for prediction
* Import all required package for prediction and import the dataset into python platform



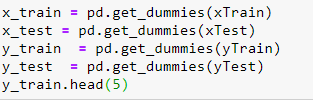
* In this project the dataset consists of potugal student dataset if that the grading system is from 1 to 20 points to simplify we convert those into 5grade point example –a students get 0 to 10 point he is awarded as F if he got 10 to 12 he will be awarded as D grade
* 
* After adding grades into dataset drop g3 values so that g3 values are not calculated twice



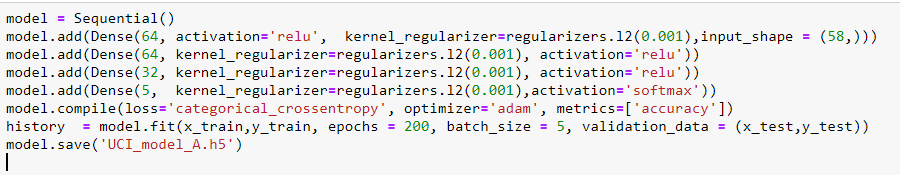
* After droping g3 split the data into train and test variables in the form of 20% and 80%



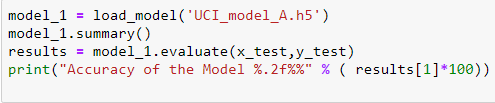
* After splitting the data perform one hot encoding



* Next create a nueral network model and save the predicted data into a file



* Calculate the accuracy of model



* Next to know the impat of prediction repeat the process and predict the
* values by removing grade 2(g2) and again by removing grade1(g1)