TECHNICAL REPORT

ON

S.I.W.E.S PROGRAM

AT

LAGOS STATE MINISTRY OF ECONOMIC

PLANNING AND BUDGET

TOPIC:

COMPUTATIONAL MATHEMATICS AND DEEP LEARNING

Olanrewaju Daniel Ayomide

205571

Mathematics Department

University of Ibadan

Ibadan, Oyo State

Duration: Dec 2021-April 2022

The Director,

Industrial Training Coordinating Center,

University Of Ibadan.

Dear Sir ,

SUBMISSION LETTER

I, OLANREWAJU DANIEL with matriculation number 205571 of the department of Mathematics write this to you upon completion of the 8 weeks S.I.W.E.S which commenced between **Dec 2021-April 2022** at the Lagos State Ministry of Economic planning and Budget, Ikeja Secretariat, Lagos.

Submitted with this report is my industrial training logbook in partial fulfillment of the requirement of the S.I.W.E.S program. This report contains detailed report of the application of computational Mathematics in deep learning

I will be glad if my report is evaluated and given a satisfactory remark.

Yours Faithfully

Olanrewaju Daniel

0.1 ACKNOWLEDGMENT

I give glory to God for a successful internship. His faithfulness is unparalleled, His mercies are without bias, His love is overwhelming.

I thank my father for his endurance and support of my internship. I thank the supportive staffs at the administrative department of the Ministry of planning and budget in Lagos for their help, assistance and supervision throughout the training.

I thank the Director and staffs at Lagos Bureau of Statistics (L.B.S) for their support. And also, I thank the staffs and Director of Industrial Training Coordinating center, University of Ibadan, for their perseverance in ensuring that we interns get the best of training.

Without the endless sacrifices paid by the Lagos State government through various initiatives to ensure that students get the best out of industrial training, then placement, registration, training and completion of the training wouldn't have been really possible. Thanks to the Lagos state government.

0.2 ABSTRACT

At the dawn of the 21st century, computation and algorithm translated into a more sophisticated tool for making work easier, lives better, and education an endless adventure.

Mathematics as a base and foundation for cryptography and enigmas as in Algebra, has given the world an opportunity to see into the adventure and take advantage of opportunities embedded in the subject matter.

Aristotle once said, 'one day Automata (AI) will change the world and bring total abolition to slavery.' And this foretelling by the great philosopher is what we see in our world today. AI and robotics is finally taking over the labour market to replace 'labourers.' And also give innovators an opportunity to make the world a better place.

Mathematics as a critical part of building AI algorithms is inevitable. Most fields of mathematics are applied in Artificial Intelligence, namely; Linear and abstract algebra, functional analysis and normed linear spaces, operation research and mathematical modelling, stochastic analysis and many more.

Since my industrial training was supported by the Lagos Bureau of Statistics, data science and deep learning was inevitable for me. They also gave access to data scientists network laboratory in the ministry building, showing Lagos state government's commitment to AI innovation.

Deep learning in modern governmental establishments is a vital prognostication tool for effective administration, response, and service delivery among others.

In this report, I considered various application of statistics, and these areas can also be transformed through the use of deep learning and computational mathematics.

Table of Contents

CHAPTER ONE
INTRODUCTION

1.1 BACKGROUND OF SIWES

The Student Industrial Work Experience Scheme (SIWES), also known as Industrial Training is a compulsory skills Training Program Experience design to expose and prepare student of Nigerian University, polytechnics, colleges of Technology and colleges of Agriculture, for the industrial work situation they are likely to meet after graduation.

The scheme also affords students the opportunity of familiarizing and exposing themselves to the needed experience in handling equipment and machinery that are usually not available in their institution.

SIWES is aimed at bridging the gap between theoretical and practical studies, skill acquisition of professional educational program in Nigerian institutions. The scheme enables students to have knowledge of difficulties in Nigerian professions and identify how some problems could be tackled. In summary, students would have knowledge and idea of how they can contribute to the society via their courses of study while in school.

The student industrial work experience scheme program was introduced by the Nigerian government and funded via the Industrial Training Funds (ITF).

1.2 AIMS AND OBJECTIVES OF SIWES

The aim and objective of the SIWES program are as follows:

- SIWES provides the avenue for student in institutions of higher learning to acquire industrial skills and experiences in their course of study.
- Prepare the students for the industrial work situations they are likely to meet after graduation.
- Expose students to work method and techniques in handing equipment and machinery that may not be available in their institutions.

- Make the transition from school to the world of work easier and chance students contact for later job placement.
- SIWES provides student with an opportunity to apply their knowledge in real work situation there by bridging the gap between theory and practice.
- Enlist and strengthens Employers involvement in the entire educational process and prepare student for employment after graduation.
- As an academic exercise for improving the intellect of the studies
- Make the transition from the universities to the world of work easier, and thus, enhance students contact for later job placement

1.3 SCOPE AND COVERAGE AND IMPORTANT OF SIWES

The student industrial work experience scheme is a program which enables student have practical experience of their course of study. In practical, my practical experience at the LAGOS BUREAU OF STATISTICS (LBS), Ikeja Lagos, Enhanced my exposure to field work as it effects data collection and Socio-Economic activities and different programming languages.

The program lasted for 3 months from December 2021 to April 2022. It gave me a clear view and detailed understanding of the activities carried out at the Lagos Bureau of Statistics, Ikeja Lagos.

I was introduced into the various applications of mathematics in machine learning and deep learning. Using the MNIST dataset, I built a number detecting deep learning algorithm which will be considered later in this report.

CHAPTER TWO

THE HISTORY OF LAGOS BUREAU OF STATISTICS (LBS)

Statistics is a mathematical science pertaining to the collection, analysis, interpretation or explanation and presentation of data. It provides tools for predicting and forecasting the economic activities of a state. It is useful for an academician, government, business body etc. On the basis of various definitions provided by economists, statistics has been broadly defined in two senses: first is in plural sense and the second is in singular sense.

In plural sense, statistics refers to numerical facts and figures collected in a systematic manner with a specific purpose in any field of study. In this sense, statistics is also aggregates of facts expressed in numerical form.

The characteristics of statistical facts are:

- Numerically expressed
- Data affected by multiplicity of causes
- Enumerated according to reasonable standard of accuracy
- Collected in systematic accuracy
- Collected for pre-determined purpose and
- Placed in relation to other
- Aggregate of facts 7

In singular sense, statistics refers to a science which comprises methods that are used in the collection, analysis, interpretation and presentation of numerical data. These methods are used to draw conclusion about the population parameters.

The stages of statistical analysis are:

- Collection of data
- Organization of data

- Presentation of data
- Analysis of data and
- Interpretation of data

Statistics helps in business forecasting, decision making, quality control, search of new ventures, study of market, study of business cycles, useful for planning, useful for finding averages, useful for bankers, brokers, insurance, etc.

2.1 STATISTICS AND OTHER FIELDS

Statistics are numerical statement of facts capable of analysis and interpretation as well as study of the methods used in collection, organization, presentation, analysis and interpretation of numerical data.

2.1.1 STATISTICS AND ECONOMETRICS

Econometrics is one of the most recent fields of study concerning economics. It combines the methods and techniques of Statistics and Mathematics to build models for the analysis of economic problems and then provides solution to all those problems. EXAMPLE: In Econometrics Linear Regression models are formulated to analyze the effect of various determinants of demand on demand for a product.

2.1.2 STATISTICS AND NATURAL SCIENCES

Natural sciences are biology, Zoology, medicine, meteorology etc. the statistics are helpful in the field of physical sciences as the experiments conducted in this field are based upon the data collected with the help of descriptive statistics. Statistics is used both for analyzing data and drawing conclusions.

2.1.3 STATISTICS AND PHYSICAL SCIENCES

Physical sciences involve Geography, geology, astronomy, Physics etc. Descriptive statistics is used to present the complex phenomenon concerning these branches of study in numerical form in a very simple way. The methods of statistics are applied to analyze the data and to find conclusions. EXAMPLE: In geography data concerning temperature level over the years can be collected and analyzed with the help of the statistics.

2.1.4 STATISTICS AND SOCIAL SCIENCES

The statistical methods are also useful in the field of History, Sociology, Education, Psychology etc. Various researches in this field are done with the help of statistics. EXAMPLE: In the field of politics, Statistics is used to evaluate the effects of the policies of the government. In History, the record of all the past events is maintained with the help of the descriptive statistics.

2.2 NATIONAL BUREAU OF STATISTICS

The National Bureau of Statistics (NBS) came into being with the merger of the Federal Office of Statistics (FOS) and the National Data Bank (NDB). The creation was part of the implementation of the Statistical Master Plan (SMP), a program document of the Federal Government of Nigeria (FGN).

The merger was to give the agency a national outlook as the apex statistical agency for all the three tiers of Government.

NBS is expected to coordinate Statistical Operations of the National Statistical System in the production of Official Statistics in all the Federal Ministries, Departments and Agencies (MDAs), State Statistical Agencies (SSAs) and Local Government Councils (LGCs).

Nigeria operates Federal System of government with 36 States and Federal Capital Territory and 774 Local Government Areas (LGAs). At the federal level, each Ministry, Department and Agency has Director of Statistics. Each state has Director of Statistics and Head of statistics Unit at Local Government Areas. All these including Statistical Institutes constitute the Nigeria National Statistical System (NSS).

The Internal organization of the bureau is built on Statistics Act of 2007 which is the Legal Instrument established by the Acts 9 of Parliament. The National Bureau of Statistics oversees and publishes statistics for Nigeria.

2.2.1 Contributing bureaus

The National bureau of statistics is an inter-ministry bureau which cannot depend on, just its own findings. Hence, there is a great need for other bureaus, ministries and agencies to contribute to the researches by the National bureau of statistics.

The contributing bureaus are where the National Bureau of Statistics get their information. They include:

• National Planning Commission

• Economic and Financial Crimes Commission

• Federal Ministry of Health

• National Population Commission

• Nigerian Stock Exchange

• Nigerian Embassies and High Commissions

• Federal Ministry of Finance

• Central Bank of Nigeria

• Nigerian National Petroleum Corporation

• Nigerian Electricity Regulatory Commission

2.3 LAGOS BUREAU OF STATISTICS

The Lagos Bureau of Statistics is a department in the Lagos State Ministry of Economic Planning and Budget concerned with the coordination of statistical activities in the state. The department focus on the collections of statistical data on populations, Housing, Finance, Education, Health, Agriculture and Social Welfare Services among others. The department also collaborates with international bodies, Federal, States and Local Governments and other statistical agencies on statistics related matters.

2.3.1 Publications

Digest of statistics is one of the annual publications of the department. It contains the statistical data on the Socioeconomic activities of the State. It features data on the State population, Traffic

Management, Waste Management and Environment. It also provides information on Motor Vehicle Registration, Road Accidents, Traffic Management, Price Index Housing, among other sectors. Other publications include Abstract of Local Government Statistics, Basic Statistical Hotline, Price Statistics Bulletin, Statistical year book and Motor Vehicles Statistics.

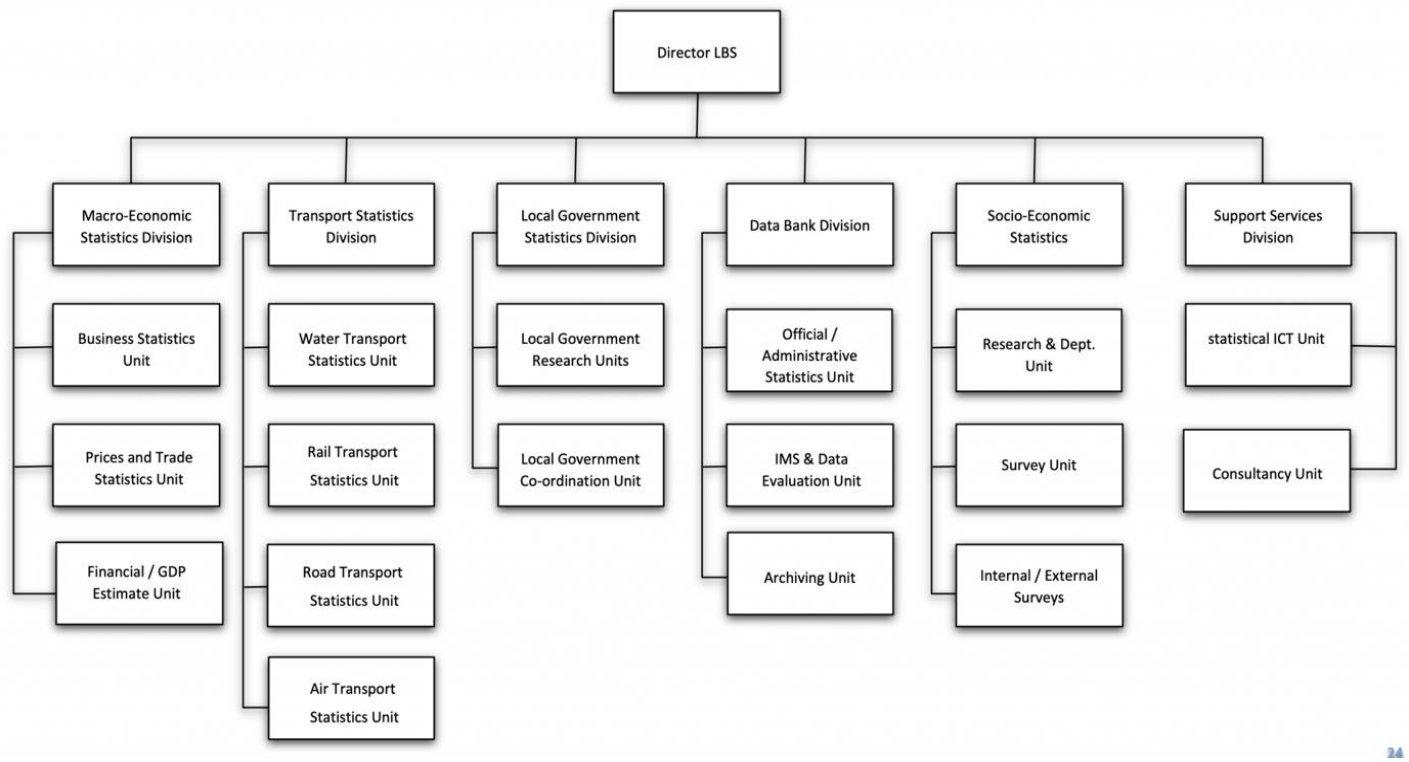
Also, the Lagos Bureau of statistics has publications for all major surveys and researches conducted by the bureau

2.3.2 Vision and mission of the bureau

To continually be the one-stop shop for qualitative, reliable and robust data for the development of the state. Ensuring a digitalized, efficient and timely statistical system for planning, policy formulation and decision making.

2.4 ORGANIZATIONAL CHART OF THE ORGANIZATION

The organization is basically divided into six broad department;



CHAPTER THREE

WORK EXPERIENCE

EXPLORING VARIOUS PYTHON FOR STATISTICAL ANALYSIS

PYTHON is a high-level, interpreted, interactive and object-oriented scripting language. Python can provide the speed needed for even compute intensive tasks. Python was designed to be highly readable with English keywords, where as other languages use punctuation. Python has fewer syntactical constructions than other languages.

3.1 IDLE Development Environment

- IDLE is an Integrated Development Environment for python, typically used on widows.

  Main features of IDLEs include –

- Multi-window text editor with syntax highlighting, auto-completion, smart indent,
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility.

3.2 Python Features

- **Python is interpreted**: This means that it is processed at runtime by the interpreter and you do not need to compile your program before executing it. This is similar to PERL and PHP

- **Python is Interactive**: This means that you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented**: This means that Python supports Object-Oriented style or technique of programming that encapsulates codes within objects.

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

- **Easy-to-learn**: Python has relatively few keywords simple structure, and a clearly defined syntax. This allows the student to pick up the language in a relatively short period of time.

- **Easy-to-read**: Python code is much more clearly defined and visible to the eyes.

- **Easy to maintain**: Python's success is that its source code is fairly easy-to-maintain.

- **A broad standard library**: One of python's greatest strengths is the bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

3.3 Some Python packages used in data science and mathematical problems

3.3.1 PANDAS

Pandas is an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labeled data in Python Programming language. Pandas stands for Python Data Analysis Library.

Pandas is a perfect tool for data wrangling or munging. It is designed for quick and easy data manipulation, reading, aggregation, and visualization.

Pandas take data in a CSV or TSV file or a SQL database and create a Python object with rows and columns called a data frame. The data frame is very similar to a table in statistical software, say Excel or SPSS.

FUNCTIONS OF PANDAS

- Indexing, manipulating, renaming, sorting, merging data frame
- Update, Add, Delete columns from a data frame
- Inputs missing files, handle missing data or NANs
- Plot data with histogram or box plot

3.3.2 NUMPY

One of the most fundamental packages in Python, NumPy is a general-purpose array-processing package. It provides high-performances multidimensional array objects and tools to work with the arrays.

NumPy is an efficient container of generic multi-dimensional data. NumPy's main object is the homogeneous multidimensional array. It is a table of elements or numbers of the same data type, indexed by a tuple of positive integers. In NumPy, dimensions are called axes and the number of axes is called rank. NumPy is used to process arrays that stores values of the same data type. NumPy facilitates math operations on arrays and their vectorization. This significantly enhances performance and speeds up the execution time correspondingly.

FUNCTIONS OF NUMPY

- Basic array operations: add, multiply, slice, flatten, reshape, index arrays
- Advanced array operations: stack arrays, split into sections, broadcast arrays
- Work with date time or linear algebra
- Basic slicing and advanced indexing in NumPy Python

3.3.3 MATPLOTLIB

Matplotlib is a library used in creating stories with data visualization. Matplotlib is the plotting library for Python that provides an object-oriented API for embedding plots into applications. It is a close resemblance to MATLAB embedded in Python programming language.

FUNCTIONS OF MATPLOTLIB

Histogram, bar plots, scatter plots, area plot to pie plot, matplotlib can depict a wide range of visualizations. With a bit of effort and tint of visualization capabilities, with Matplotlib, you can create any visualization:

- Line plots
- Scatter plots
- Area plots
- Bar charts and Histograms
- Pie charts
- Stem plots
- Contour plots
- Quiver plot
- Spectrogram, etc.

3.3.4 SEABORN

Seaborn is a data visualization library that provides a High-level interface for drawing attractive and informative statistical graphics.

FUNCTION OF SEABORN

- It is used to determine relationships between multiple variables (correlation)
- Observe categorical variables for aggregate statistics
- Analyze univariate or bivariate distribution of different data subsets
- Plot linear regression models for dependent variables
- Provides high-level abstractions, multiplot grids, etc.

  Seaborn is a great second-hand for R visualization libraries like corrplot and ggplot.

3.3.5 SCIKIT LEARN

Scikit learn is a robust machine learning library for Python. It features ML algorithms like SVMs, random forests, k-means clustering, spectral clustering, mean shift, cross-validation and more even

NumPy, SciPy and related scientific operations are supported by Scikit learn with Scikit Learn being a part of the SciPy Stack.

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. Supervised learning models like Naïve Bayes to grouping unlabeled data such as K-means.

FUNCTIONS OF SCIKIT

- Classification: Spam detection, image recognition
- Clustering: stock price
- Regression: Customer segmentation, Grouping experiment outcomes
- Dimensionality reduction: visualization, increased efficiency
- Model selection: improved accuracy via parameter tuning
- Pre-processing: preparing input data as a text for processing with machine learning algorithms

  Scikit Learn focuses on modeling data not manipulating data. We have Numpy and Pandas for summarizing and manipulating.

3.3.6 TENSORFLOW

TensorFlow is an Al library that helps developers to create large-scale neutral networks with many layer's using data flow graphs. TensorFlow also facilitates the building of Deep learning models, push the state-of-the-art in ML/AL and allow easy deploy of ML-powered applications.

One of the most developed websites amongst all libraries is of TensorFlow. Giants like Google, Coca-Cola, Airbnb, Twitter, Intel, DeepMind use TensorFlow. TensorFlow is quite efficient when it comes to classification, perception, understanding, discovering, predicting, and creating data.

FUNCTIONS OF TENSORFLOW

- Voice/sound Recognition: IoT, Automotive, Security, UX/UI, Telecom
- Sentiment Analysis: Mostly for CRM or CX

- Text-Based Apps: Threat Detection, Google Translate, Gmail smart reply
- Face Recognition: Facebook's Deep Face, Photo tagging, Smart Unlock**.**

3.3.7 KERAS

Keras is a TensorFlow's High-level API for building and training Deep Neural Network code. It is an open-source neural network library in Python. With keras, statistical modeling, working with images and text is a lot easier with simplified coding for deep learning.

FUNCTIONS OF KERAS

- Determine percentage accuracy
- Compute loss function
- Create custom function layers
- Built-in data and image processing
- Write functions with repeating code blocks: 20, 50, 100 layers

CHAPTER THREE

PERSONAL PROJECT

A DEEP LEARNING PROJECT USING TENSORFLOW AND SOME OTHER DEEP LEARNING PACKAGES

Finally, in this chapter I'll build a machine learning model using the MNIST dataset.

MNIST dataset, a classic in the machine-learning community, which has been around almost as long as the field itself and has been intensively studied. It's a set of 60,000 training images, plus 10,000 test images, assembled by the National Institute of Standards and Technology (the NIST in MNIST) in the 1980s.

You can think of "solving" MNIST as the "Hello World" of deep learning—it's what you do to verify that your algorithms are working as expected. For anyone that will do machine-learning as a profession, you'll see MNIST come up over and over again, in scientific papers, blog posts, and so on.

DATA IMPORTATION

```
from keras.datasets import mnist

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

train_images and train_labels form the training set, the data that the model will learn from.

The model will then be tested on the test set, test_images and test_labels.

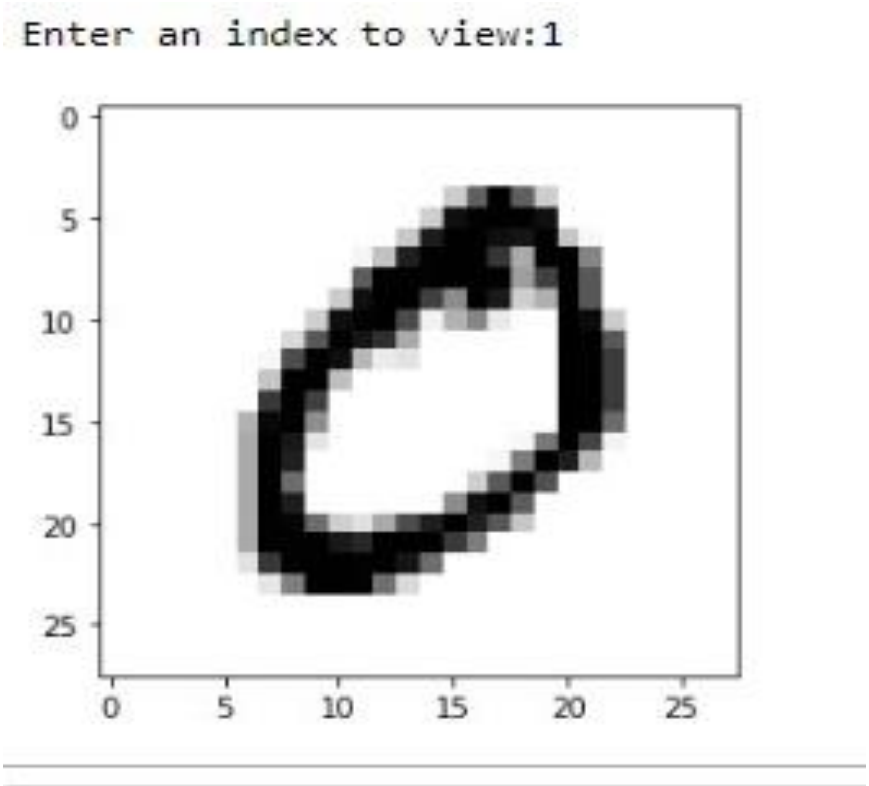
The images are encoded as NumPy arrays, and the labels are an array of digits, ranging from 0 to 9. The images and labels have a one-to-one correspondence.

AN IMAGE FROM THE DATASET

Code*: import matplotlib.pyplot as plt*

```
try :
    index = eval(input('Enter an index to view:'))
    plt.imshow(train_images[index],cmap = 'gray_r')
    plt.show()
except:
    print('Invalid response!')
```

Result:



Matplotlib was applied here to show the image in the MNIST dataset with index 1 which zero.

RESHAPING THE IMAGES AND CONVERTION TO CATEGORICAL

Before training, we'll preprocess the data by reshaping it into the shape the network expects and scaling it so that all values are in the [0, 1] interval. Previously, our training images, for instance, were stored in an array of shape (60000, 28, 28) of type uint8 with values in the [0, 255] interval. We transform it into a float32 array of shape (60000, 28 * 28) with values between 0 and 1.

Also, we categorically encode the labels.

Code : *train_images.shape*

```
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype('float32') / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype('float32') / 255


from keras.utils import to_categorical
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
val_data = train_images[:10000]
val_label = train_labels[:10000]
```

CREATING A VALIDATION SET AND BUILDING THE DEEP LEARNING MODEL

In order to monitor the accuracy of the model on data it has never seen before, I created a validation set by setting apart 10,000 samples from the original training data.

Code: *from keras import optimizers*

```
from keras import models

from keras import layers, regularizers

network = models.Sequential()

network.add(layers.Dense(450, activation='relu', input_shape=(28 * 28,)))

network.add(layers.Dropout(0.5))

network.add(layers.Dense(450, activation='relu'))

network.add(layers.Dropout(0.5))

network.add(layers.Dense(10, activation='softmax'))
```

The core building block of neural networks is the layer, a data-processing module that you can think of as a filter for data. Some data goes in, and it comes out in a more useful form. Specifically, layers extract representations out of the data fed into them—hopefully, representations that are more meaningful for the problem at hand.

Most of deep learning consists of chaining together simple layers that will implement a form of progressive data distillation.

A deep-learning model is like a sieve for data processing, made of a succession of increasingly refined data filters—the layers. Here, our network consists of a sequence of three Dense layers, which are densely connected (also called fully connected) neural layers.

The third (and last) layer is a 10-way SoftMax layer, which means it will return an array of 10 probability scores (summing to 1). Each score will be the probability that the current digit image belongs to one of our 10-digit classes.

BUILDING A CLASS FOR CALLBACKS

Every deep learning model may not be will not always have a 100% accuracy rate, but, an 80% accuracy rate and above is acceptable. For my project, I set the maximum at 99% and minimum at industry's standard of 80%.

I defined a class StopCallback to stop the epochs from continuing when a 99% accuracy is achieved.

Code:

```
import keras

    class StopCallback(keras.callbacks.Callback):

        def on_epoch_end(self, epoch, logs={}):

            if(logs.get('accuracy')>0.999):

                print("\nReached 99% accuracy so cancelling training!")

                self.model.stop_training = True


    callbacks = StopCallback()
```

MODEL COMPILATION

To make the network ready for training, we need to pick three more things, as part of the compilation step:

- A loss function—How the network will be able to measure its performance on the training data, and thus how it will be able to steer itself in the right direction.
- Metrics to monitor during training and testing—Here, we'll only care about accuracy (the fraction of the images that were correctly classified).
- An optimizer—The mechanism through which the network will update itself based on the data it sees and its loss function.

Categorical_crossentropy is the loss function that's used as a feedback signal for learning the weight tensors, and which the training phase will attempt to minimize. This reduction of the loss happens via minibatch stochastic gradient descent. Categorical_crossentropy formula is given below:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\frac{1}{N}\sum_{i}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

The exact rules governing a specific use of gradient descent are defined by the Adam optimizer passed as the first argument in the code below (In[10]). Adam optimizer combines the best properties of the **Adaptive Gradient Algorithm** (AdaGrad) and **Root Mean Square Propagation** (RMSProp) algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. Adam is the best among the adaptive optimizers in most of the cases.

Also, you will see later, the training loss will decrease with every epoch, and the training accuracy will increase with every epoch. After the compilation of the model, the training accuracy was 97.00%, the accuracy on the validation data was 98.77%, and the test accuracy was 98%.

Code: *network.compile(optimizer= 'Adam',*

loss='categorical_crossentropy',

metrics=['accuracy'])

history = network.fit(train_images, train_labels, epochs=5, batch_size=200, validation_data = [val_data,val_label])

Result:
```
Train on 60000 samples, validate on 10000 samples
Epoch 1/5
60000/60000 [==============================] - 21s 344us/step - loss: 0.40
36 - accuracy: 0.8755 - val_loss: 0.1327 - val_accuracy: 0.9614
Epoch 2/5
```

```
60000/60000 [==============================] - 18s 300us/step - loss: 0.17
35 - accuracy: 0.9482 - val_loss: 0.0857 - val_accuracy: 0.9746
Epoch 3/5
60000/60000 [==============================] - 18s 300us/step - loss: 0.13
55 - accuracy: 0.9589 - val_loss: 0.0648 - val_accuracy: 0.9794
Epoch 4/5
60000/60000 [==============================] - 19s 310us/step - loss: 0.11
34 - accuracy: 0.9653 - val_loss: 0.0479 - val_accuracy: 0.9858
Epoch 5/5
60000/60000 [==============================] - 19s 317us/step - loss: 0.09
55 - accuracy: 0.9699 - val_loss: 0.0407 - val_accuracy: 0.9884
```

VISUALIZING THE MODEL ACCURACY USING MATPLOTLIB

After training, the training loss decreases with every epoch, and the training accuracy increases with every epoch. That's what you would expect when running gradient-descent optimization—the quantity you're trying to minimize should be less with every iteration.

That is also the case for the validation loss and accuracy.

In some cases, a model that performs better on the training data isn't necessarily a model that will do better on data it has never seen before. In precise terms, some may experience overfitting after some epochs, and you end up learning representations that are specific to the training data and don't generalize to data outside of the training set. In this case, to prevent overfitting, you could stop training after the epochs you see divergence in a similar curve as shown above. This is one advantage of using matplotlib in deep learning

In general, you can use a range of techniques to mitigate overfitting. Some include;

- Adding dropout.
- Trying different architectures: add or remove layers.
- Trying different hyperparameters (such as the number of units per layer or the learning rate of the optimizer) to find the optimal configuration.
- Add L1 and/or L2 regularization.
- Optionally, iterate on feature engineering: add new features, or remove features that don't seem to be informative.

Code: *test_loss, test_acc = network.evaluate(test_images, test_labels)*

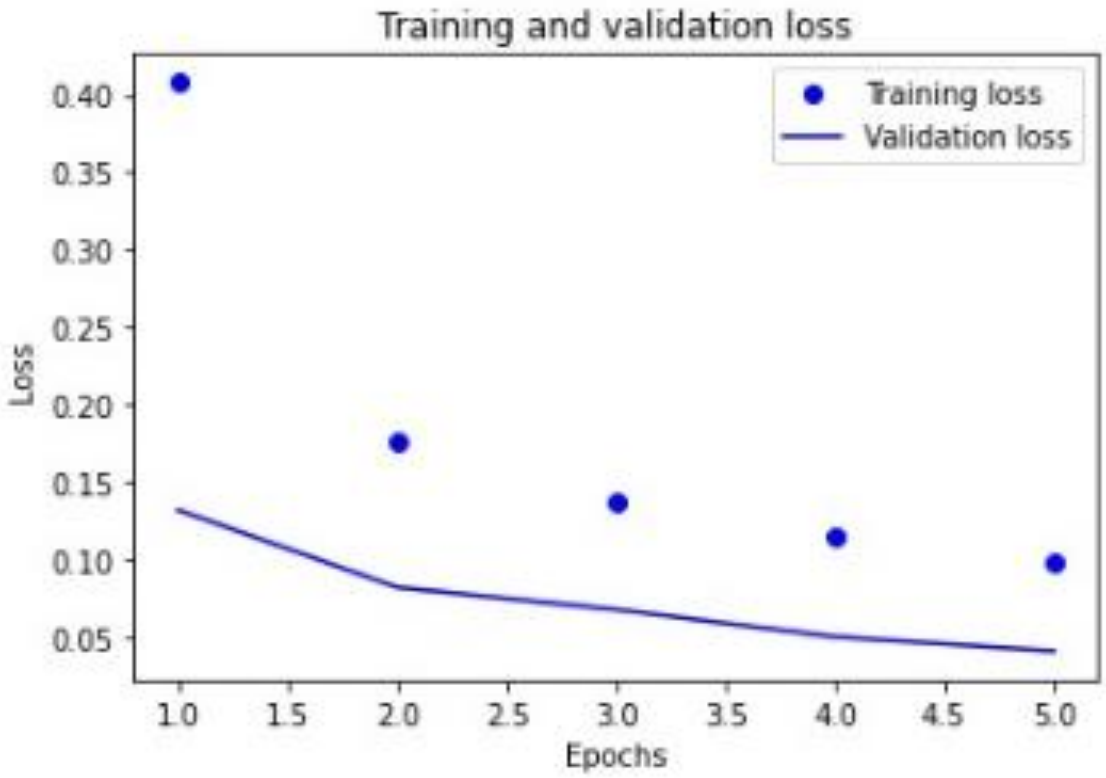
Result: `10000/10000 [==============================] - 2s 210us/step`

Code: *print(test_loss, test_acc)*

Result: `0.06844121536314487 0.9794999957084656`

Code:
```
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict['loss']
acc = history_dict['accuracy']
val_loss_values = history_dict['val_loss']
epochs = range(1, len(acc) + 1)
plt.plot(epochs, loss_values, 'bo', label='Training loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

Result:



Code : *plt.clf()*

acc_values = history_dict['accuracy']

val_acc = history_dict['val_accuracy']

plt.plot(epochs, acc, 'bo', label='Training acc')

plt.plot(epochs, val_acc, 'b', label='Validation acc')

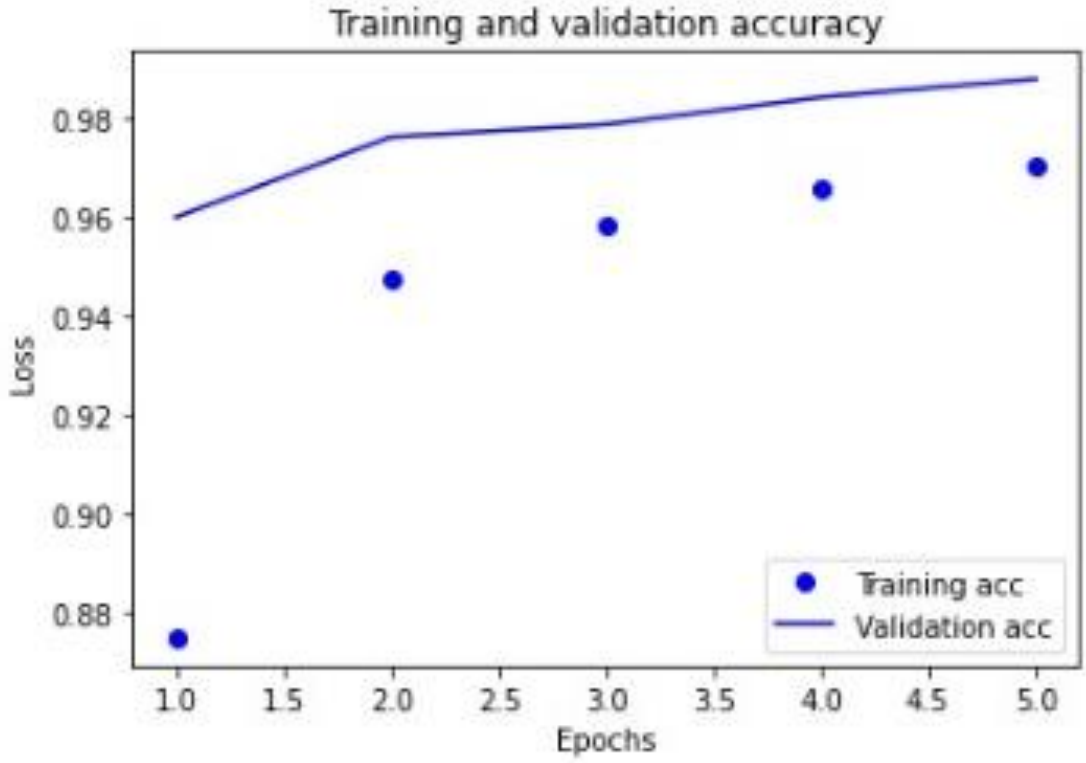
plt.title('Training and validation accuracy')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()



TESTING THE RELIABILITY OF MY MODEL

Finally, I tested the reliability of my model by testing a random picture named ‘test.jpg’ shown below for prediction.

Read the image as test.jpg with cv2 python package

Code: *import cv2 as cv*

image = cv.imread('test.jpg')

cv.imshow('test.jpg', image)

cv.waitKey(0)

Result:

Code: *image = cv.imread('test.jpg',cv.IMREAD_GRAYSCALE)*

 # image = test_images[1]

image = cv.resize(image, (28,28))

plt.imshow(image, cmap = 'Greys')

plt.show()

*image = image.reshape(1,28*28)*

image = image.astype('float32')/255

 # image = 255-image

pred = network.predict(image)

print('The writing is',pred.argmax())

jupyter IT PROJECT 205571 Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

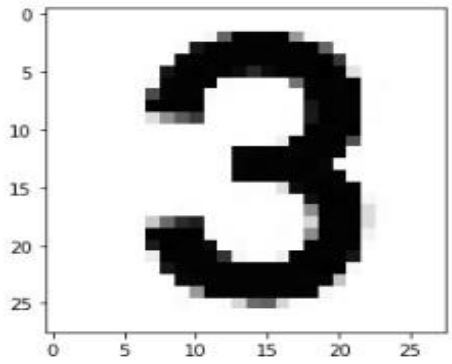
In [52]:
```
image = cv.imread('test.jpg',cv.IMREAD_GRAYSCALE)
# image = test_images[1]
image = cv.resize(image, (28,28))

plt.imshow(image, cmap = 'Greys')
plt.show()

image = image.reshape(1,28*28)
image = image.astype('float32')/255
# image = 255-image



pred = network.predict(image)
print('The writing is',pred.argmax())
```




```
The writing is 3
```

An accurate prediction of my image was delivered. Perfect model indeed!

CHAPTER FOUR

CONCLUSION AND RECOMMENDATION

Mathematics was fundamental in my understanding of the various components of the deep learning model discussed in the previous chapter.

Matrix theory was simplified through the use of NumPy python package.

Optimization theory as taught in operation research and mathematical modelling was applied to the model in Chapter 3 – the minimization of loss and optimization of accuracy while the epoch was carried out.

Computational mathematics was applied in the graph plotting and visualization.

To give students an opportunity in deep learning or AI career line, I recommend the following;

1. Mathematics for AI: A course in mathematics department should be added to the existing syllabus to concentrate the attention of students interested in the field on required math topics.
2. AI labs in the university should be made open to all interested students.
3. Equipment and tools should also be made available for practical.
4. AI library for readers to enjoy works, research, and publications in the field.
5. Sensitization of 100 level students on AI as a career path in mathematics.

References

lagosstate.gov.ng

deeplearning.ai

deep learning course by Andrew Ng

Wikipedia

Digest of statistics 2012-2019 by the Lagos Bureau of statistics.

Poverty report 2020 by Lagos bureau of statistics Ministry of Economic planning and budget 2020 budget

Lagos state price bulletin by the Lagos Bureau of statistics.