



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 6, Issue 2, February 2018

A Neroph Approach for Classifying *E. coli* Data

Christopher Ejiofor*, Okon and Emmanuel Uko

Department of Computer Science, University of Port-Harcourt, Port-Harcourt, Nigeria

E-mail: christopher.ejiofor@uniport.edu.ng

Abstract: Classifications are used in handling classification problems, which usually existing when entities or objects need to assigned predefined groups or classes, perhaps based on attributes, parameters and values. This research paper provide a simplify description of neuroph classification using *E. coli* data. The data were structured into training and testing data. The neuroph neural network architecture caters for 34 neurons: twelve input neurons (12), seventeen (17) hidden neurons and five (5) output neurons. The training accommodated approximately 185 iterations with a cumulative error of 6.8544 and an average error of 0.0367. The Total Mean Square Error (TMSE) obtained from testing the trained data gave an approximate value of 2.7372. This minima error showed the optimality in training and classification using *E. coli* data.

Keywords: Neuroph; Classification; *E.coli*

I. INTRODUCTION

Classification; an ordered set of related entities is applied in grouping data based on similarities or attributes [1-10]. This concept is an integral functionality of human existence facilitating appropriate decision making from the perspective of shared attributes (descriptors). Classification identifies exhibited parameters displayed or possessed by these entities (objects) [6]. Classifications are designed with classification problems in perspective which usually existing when entities or objects need to assigned predefined groups or classes, perhaps based on attributes, parameters and values [2]. The success of classifying any entities, depends, largely and efficiently on the associated attributes internally and externally exhibited [9]. This attributes identify the existences and functionalities of the object been classified. Classification is a sub domain of clustering problems which may classify both inanimate and animate entities [1,2]. The medical field has explored entity classification in addressing medical problems due to the abundance of medical data and its similarities [4]. Medical Classification or Medical is an offshoot of clustering or classification. MC is a transformation process in which medical descriptors or attributes of medical diagnoses and procedures are categorized into entities with unique attributes [4]. The classification of various disease agents (viral, bacterial or pathogenic) has gradually evolved within the medical fields. Notable among this agent is *Escherichia coli* (*E.coli*) which has contributed to the propagation of diarrhea and other infectious disease caused by these bacteria. *E. coli* are bacteria found in the environment, foods, and intestines of people and animals. *E. coli* are a large and diverse group of bacteria [8]. Although some strain are harmless enhancing body functionalities, some strain of *E.coli* causes sick such as diarrhea, kidney failure, urinary tract infections, respiratory illness, pneumonia, and even death. These problems are most likely to occur in children and in adults with weak immune systems [5]. Symptoms of infection include: nausea or vomiting, severe abdominal cramps, watery or very bloody diarrhea, fatigue and fever [8].

Notably, the classification of medical entities are handled using simulator: Matric Laboratory (MATLAB), Neuroph, Simulik, Mapslim, Mathematica, Netslim, ProLB, SimEvent, TRNSYS, PSF Lab, VisSim, Vortex, VisualSim Architect, NI Muttisim [7,10].

Neuroph has been used repeatedly applied in classifying neural network based entities with varied parameters and associated classes. Neuroph is a lightweight Java neural network framework used in developing neural network architectures. It consists of well designed, open source Java Library with small number of basic classes' corresponding to neural network concept. Neuroph simplify the development of neural networks by providing Java neural network library and GUI tool that supports creating, training and saving neural networks.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 6, Issue 2, February 2018

Consequently, it is the intent of this research to employ Neuroph as a computer simulator in classifying *E. coli* data.

II. NEUROPH ARCHITECTURE

Neural networks are artificial intelligence (machine learning technology) suitable for ill-defined problems, such as recognition, prediction, classification, and control [11]. Neural are perhaps suitable for implementation using Neuroph. Neuroph is a Java Neural Network development environment built with close integration with NETBean Platform and Neuroph Framework [11]. It is customized as an Integrated Development Environment (IDE) for neural network simulation (<https://www.predictiveanalyticstoday.com/neuroph/>). It is an interactive Java framework that provides classes for building neural network. In addition to providing easy-to-use neural network wizards and tools, Neuroph also integrates basic Java development modules from the NetBeans IDE, enabling developers to create, test, and deploy various Java components based on neural networks in the same environment.

Neuroph contains well designed, open source Java Library with small number of classes which corresponds to basic NN concept. It also provides Java neural Library and GUI tools that support creating, training and saving Neural networks.

III. NEUROPH SIMULATION

The simulation of *E. coli* experimental data using Neuroph simulator provide necessary simulation steps, creating a neuroph architecture with experimental data transformed through normalization, training and testing. The simulation steps includes: preparation of training data, creating a neuroph project, creating a training set, creating a neural network, training the network and testing the network. The Preparation of the training set is the first step which involves a preprocessing paradigm where experimental data are kept with a constrained boundary. The preparation/normalization/preprocessor formula is found on equation 1. The value M is the standard normalized value for variable V in the Ith row is calculated as:

$$M' = \frac{M - \text{Min}(V)}{\text{Max}(V) - \text{Min}(V)} \quad (1)$$

Where:

M' = Normalised variable V in the ith case

M = Non-Normalised variable V in the ith case.

min (V) = Minimum Non-Normalised variable V in dataset.

max (V) = Maximum Minimum Non-Normalised variable V in dataset

The second step involves creating a new Neuroph project succeeding data standardization, which involves placing the data in a .CSV file ready for training set. It also involves setting up the number of input, hidden neuron and expected output. It also involves setting up the transfer function and leaning rule (Figure 1) portrays the neuroph project with a total of twelve (12) inputs, seventeen (17) hidden neuron and five (output). It also portrays a transfer function of sigmoid membership and a back propagation as a learning paradigm.

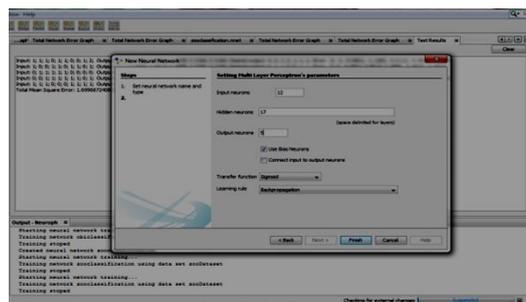


Figure 1: Neuroph initiation project.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 6, Issue 2, February 2018

The third step involves creating the simulation training set, which involves preferring a suitable training data name with associated input parameters identified. The step also involves choosing the learning and classification paradigm. For this research a supervised paradigm with ZOODATASET picked as the data name. Figure 2 shows the simulation interface for creating training data.

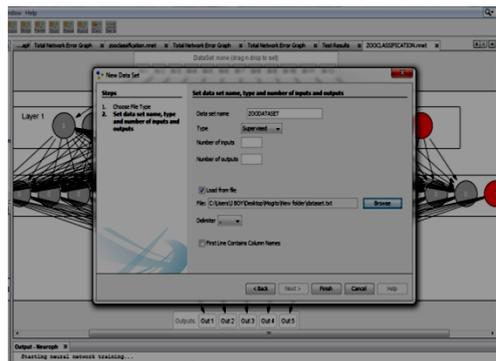


Figure 2: Creating training data.

Figure 3, portrays these normalized, scaled data between a value range of 0 -1. It also shows clearly a total of twelve (12) inputs accommodated across five outputs. It also shows a total network error graph.

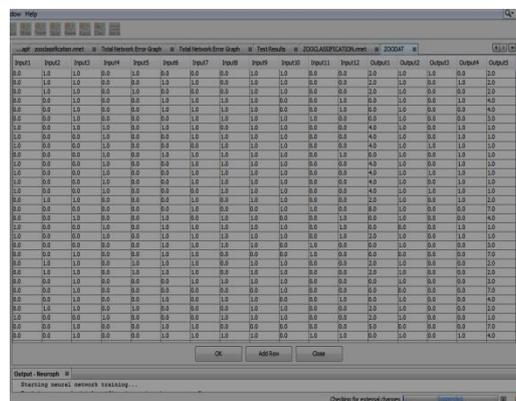


Figure 3: Normalized training data.

The fourth Step involves creating Neural Network Architecture using an appropriate number of neurons inclusive of hidden neurons. Since this phase is not universal across board, identifying appropriate number of neurons input, hidden and output will prevent the training structure for converging toward a noisy approach or structure. The associated architecture for this research portray on Figure 4, identify a total of 34 neurons, 12 neurons for input, 17 hidden neuron and 5 output neurons. The architecture also shows clearing layer 1 (input neurons), layer 2 (hidden neurons) and layer 3 (output neurons).

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 6, Issue 2, February 2018

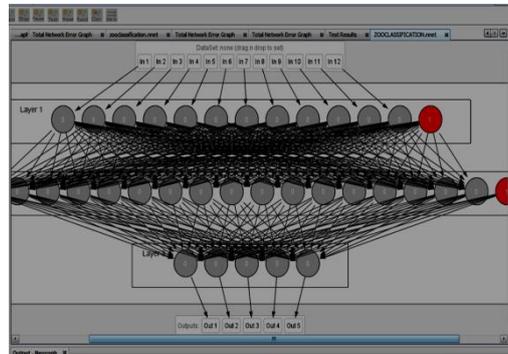


Figure 4: Neuroph neural network architecture.

The fifth Step involves training the network using the created training and normalizes data. This point involves setting the learning parameter, the maximum error, learning rate and the momentum. This research identify a maximum error is 0.01, learning rate 0.2 and momentum 0.7, the iteration count is 4 and the percentage distribution giving as 60%, 20%, 20%. Figure 5 identify the network training parameters. Figure 6 identifying the training, with associated training error capturing 185 iteration with an average summation error of 6.8544 and an average error of 0.0367.

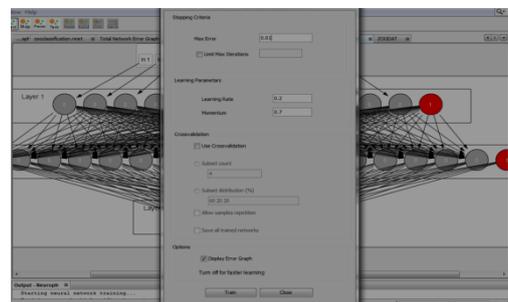


Figure 5: Training parameter.

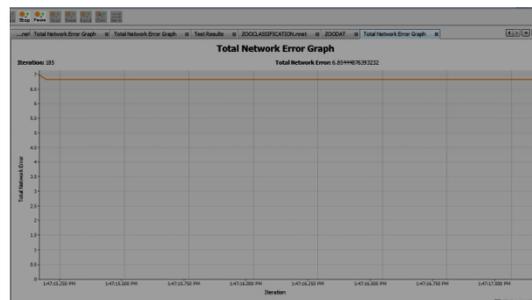


Figure 6: Neuroph training error.

IV. NEUROPH TESTING

The neuroph training identifies training input and compared actual output with desired output. Succeeding network testing, training determines the total number error and all individual errors. These associated values are clearly exemplified on Figure 7.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 6, Issue 2, February 2018

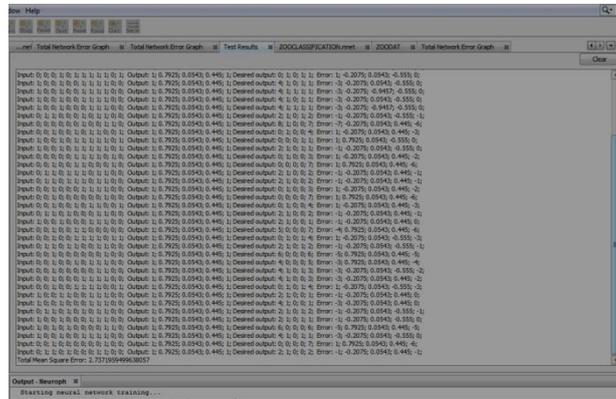


Figure 7: Neuroph testing data.

From the training data above comprise approximately about 50 training cases with 12 input and five expected output.....Total Mean Square Error: 2.7371959499638057~2.74.

V. RESULTS AND DISCUSSION

The application of Neuroph for the classification of *E. coli* data has been implemented. This approach purely focuses on machine learning using a Java based integrated platform Neuroph. It imbed 34 neurons with 17 neurons used internally and 17 used externally (input and output). The iteration combine produced a summation 6.8544 error and an average error of 0.0367. Testing the training data, a testing error of 2.73719 was obtained.

VI. CONCLUSION

This research paper has applied neuroph to the classification of *E. coli* data. The neural network based architecture provides five (05) outputs from twelve (12) inputs using seventeen (17). The training was optimal with an iteration of one hundred and eight five (185) and summation error of about 6.0. This training data shows the usefulness and versatility of Neuroph in providing optimal classification.

VII. REFERENCES

1. A Amir, D Lipika, et al. A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. Pattern Recognit. Lett 2007; 28: 110-118.
2. A Robert, Race and biological diversity in humans, Race and Human Diversity: A Biocultural Approach. Upper Saddle River, NJ: Prentice Hall 2011; 1-10.
3. A Mihael, BM Markus, et al. Optics: ordering points to identify the clustering structure. International Conference on Management of Data 1999.
4. P Bekir, Uk Alper, Classification of medical documents according to diseases, Signal Processing and Communications Applications Conference 2015.
5. Central Statistical Office CSO, what is Classification 2014.
6. JM Kaplan, (2011) Race: What Biology Can Tell Us about a Social Construct. Encyclopedia of Life Sciences (ELS) John Wiley and Sons 2011.
7. JA Karl, P Albertos, et al. Control of Complex Systems 2001.
8. MedlinePlus E.Coli Infection 2017.
9. NF Paul, F Morteza, A classification of success factors for decision support systems, The Journal of Strategic Information Systems 2011; 7: 53-70.
10. S Rajarishi, L Vei-Chung, et al. Modeling and Simulation Methods for Design of Engineering Systems. Journal of Computing and Information Science in Engineering 2001; 1: 84-91.
11. S Zoran, Neural Networks on the NetBeans. Technology network 2011.