Lie Detection System

Mithali Shashidhar
PES University
CSE Department
Bangalore, India
mithalishashidhar8@gmail.com

Prajna Girish
PES University
CSE Department
Bangalore, India
prajna2310@gmail.com

Aniket Kaulavkar
PES University
CSE Department
Bangalore, India
aniket.kaulavkar@gmail.com

***Abstract*—The goal of the project was to substitute back-propagation used in a Neural Network by other search strategies. Our solution uses Eagle's strategy which combines a global and local search to reach the result, which works with a deep neural network at its ground level.**

I. Introduction

The base source code which we were given had implemented a deep neural network model to build a lie detection system. The lie detection system has 3 components - a camera capture on facial features, an audio for capturing voice and a set of IOT devices for capturing body movements. Each sequence of communication between the interviewer and interviewee is considered to be an instance. The truth value of each instance is captured at the end at the end of the dialogue. Based on these truth values, we have trained our deep neural network to classify the given data as "lie" or "truth". We have used the eagle strategy for this purpose. Similar to how an eagle accelerates its flying towards its goal as soon as it identifies its target, we initially randomise the data and then narrow down based on some heuristics later based on how close the predicted value is to the observed values.

II. Implementation

The system architecture involves three layers in the neural network as mentioned in the diagram shown below. The first two layers are using the ReLu activation function and the last one uses the sigmoid activation function. In order to classify whether the instance 1 (for truth) or 0 (for a lie), we first split the given dataset into a training and testing dataset. In order to train the data, we run the training dataset through the neural network so that it gets 'smarter'. Following this, we pass the testing dataset and obtain a confusion matrix which tells us about the accuracy of the given program.

III. Technologies Used

We have used python3 to build and set up the neural network. The dataset was read by the program using the pandas module, and it created a pandas data frame to store the data. Following this, we used sklearn to split the date into a test and train dataset, and then normalised the data to ensure uniform values throughout. The keras module in python had inbuilt activation functions- in our case we needed only relu and sigmoid, and this helped us set up our neural network after specifying the number of neurons in every level. We used the adam optimiser function and the method of mean squared error to calculate least loss, which was done using the keras function-model.compile. The copy module in python was used for its deepcopy function to make working with arrays easier.

IV. Basic Algorithm used

Pseudo code for the Eagle Strategy

Objective functions f(x)
Initialization and random initial guess $x^{t=0}$
while (stop criterion)
1. Global exploration by randomization
2. Evaluate the objectives and find a promising solution
3. If p_E *<rand, switch to a local search*
Intensive local search around a promising solution via an efficient optimizer
if (a better solution is found)
Update the current best
end
end
Update t=t+1 end
Post-process the results and visualization.

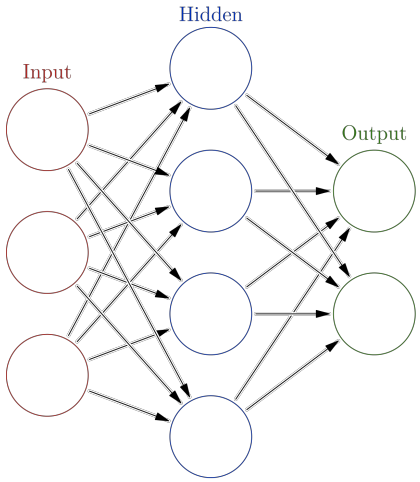

Fig:The neural network with one hidden layer. 24 inputs, and one output.

V. Modules Developed

- getRandomWeights: Used for global search, randomly updates all the weights in all the layers of the neural network. Chooses a random value between 0 and 1.

- getRandomWeightsLocalSearch:This is used for the local search. Randomly picks one of the layers and increases/ decreases its weights by 20%.
- test_model:Does forward propagation and finds the accuracy and confusion matrix of the classification model created.

VI. WORKING OF THE MODEL

Global search algorithms do not require an initial solution, and their goal is to find the global optima of the cost function. On the other hand, local search algorithms start exploring the state space in a certain point of the state space ie. it helps to find a local minima. The Eagle Strategy (ES), one of the famous AI methods to find global minima or maxima. This model implements Eagle's strategy with the help of a global search algorithm followed by a local search algorithm. The architecture of the neural network can be described as having one input, one output and one hidden layer. The two aforementioned algorithms manipulate the weights of the neural network to perform optimise the search

The global search method tries to find the best set of weights by randomising all the weights in all the layers. This process keeps repeating till we obtain an accuracy greater than 70%. Once this accuracy threshold has been reached, we start a local search. In this process, we change the weights of a particular layer, picked randomly, by a value, in our case this value being 20%. This process is repeated until we get an accuracy greater than approximately 80%. If the value of accuracy doesn't change for 100 epochs, we randomise the values again and repeat the process. This is continued till anything greater than the 80% threshold is achieved.

VII. CONCLUSION

This aim of this project was to achieve a high accuracy in classifying some of the instances as true and the others as false. In order to achieve this task, we implemented Eagle's strategy and were able to obtain an accuracy of 84%.

Although we faced obstacles while implementing these algorithms, this project has greatly helped us in learning the practical applications of Eagle's strategy as well as the various search algorithms. The most challenging part was being able to replace back propagation with an algorithm just as efficient.

In conclusion, this entire project was a great learning experience where we were able to practically implement some of the algorithms we have only seen on paper.

VIII. ACKNOWLEDGMENT

We would like to thank our mentor, K S Srinivas for guiding us through the entire project and helping us learn and understand the required concepts. We would also like to thank our peers for constant support and encouragement.

IX. REFERENCES

1. Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization- *AH Gandomi, XS Yang, S Talatahari, S Deb*

2. Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization-*XS Yang, S Deb - Nature Inspired Cooperative Strategies for Optimization*

3.https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6

4.https://skymind.ai/wiki/neural-network

5.http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

6.https://medium.com/technology-invention-and-more/everything-you-need-to-know-about-artificial-neural-networks-57fac18245a1

7. Two-stage eagle strategy with differential evolution *XS Yang, S Deb - arXiv preprint arXiv:1203.6586, 2012*