ISSN 2395-1621

Logic Based Rule Generation from a Trained Artificial Neural Network for Input Data Reconstruction

Venkat Raman B[1], Dr. Nagaratna P Hegde[2]

[1] *Research Scholar, Osmania University, Hyderabad, India*
[2] *Professor, Vasavi College of Engineering, Hyderabad, India*

ABSTRACT

The reverse engineering process of reconstructing the input data from a given trained artificial neural network is an important step in recall operation from the brain, especially when we try to build an artificial brain from an associative memory perspective. We present a robust approach for generalized input data reconstruction given the label of a class and an artificial neural network trained as a classifier. We build logical rules for the input data which helps in rebuilding the trained data based on the given class label.

Keywords—Reconstruction, recall operation, associative memory, logic, artificial neural network.

ARTICLE INFO

Article History

Received: 8th March 2020

Received in revised form :

8th March 2020

Accepted: 10th March 2020

Published online :

11th March 2020

I. INTRODUCTION

There are two main approaches to build an artificial brain namely computing using the perceptron model[1] and developing an associative memory where we store and recall patterns content based. The latter is a distinct approach which views the brain as a quantum associative memory[2]. In this paper, we propose a robust approach which would reconstruct different types of trained data given the identity of the class. The method would be useful in vast applications like image reconstruction, recovering data records of a particular category, correlation analysis, etc.

II. LITERATURE REVIEW

Artificial Neural Networks and Artificial Neuron

An artificial neuron is the basic building block for constructing an Artificial Neural Network. It works on the principle of solving a set of linear equation or inequations. The input matrix 'X' of order 'm x n' consists of n features/attribute values of m entities. The weight matrix W of order 'n x 1' consists of weights associated with the corresponding attributes to determine the target matrix 'y'. 'y' is of order 'n x 1' determining the target or classification identity of the entities in 'X'. The goal of the system is to determine the weight matrix, 'W' which would satisfy the classification problem of assigning the records of 'X' to the target variables in 'y'.[3]

$$\begin{pmatrix} x11 & x12 & x13\ldots & x1n \\ x21 & x22 & x23\ldots. & x2n \\ \ldots\ldots\ldots\ldots & & & \\ xm1 & xm2\ldots\ldots & & xmn \end{pmatrix} \begin{pmatrix} w1 \\ w2 \\ \ldots \\ wn \end{pmatrix} = \begin{pmatrix} y1 \\ y2 \\ \ldots \\ yn \end{pmatrix}$$

or

$$\mathbf{X * W = y}$$

The problem can be represented by the model of an artificial neuron in the following way

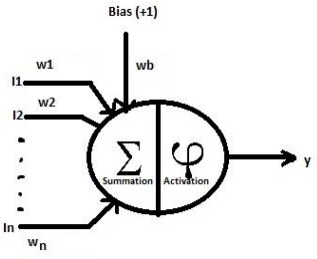

Figure 1: Model of an artificial neuron[4]

Figure 1 represents model of an artificial neuron which computes target value for a single entity/record. 'n' attribute values are taken as inputs with corresponding weight vectors. The summation function calculates the weighted sum of the inputs and then produces the result to

activation function which gives the final output of the neuron

Propositional Logic

Propositional logic is based on atomic logic sentences called propositions which can be either be 'True' or 'False' but not both[5]. The logic propositions have Negation (~), Union or OR (U) and Intersection or AND (.) operations. The propositions represent gained knowledge regarding the relationship between the variables. The propositions can also be used to represent rules or constraints that have to be satisfied by the variables. These variables may represent a real world state or action that can have binary values. The fundamental purpose behind using propositional logic is to make use of inference rules that can help us in concluding the acceptable values of the 'X' matrix to be re-generated.

III. PROPOSED METHOD

There would be two inputs to the system namely a trained classifier neural network[6] and the class label for which the input training data has to be reconstructed. The label would be taken based on the activation value of the trained network. We considered a binary step function as the activation used for the trained network.

The basic idea is to formulate propositional logic rules for the input values to the given activation output. These rules represent the constraints that input values have to satisfy in order to produce the given output. The rules would make us identify the range of acceptable inputs along with the allowed combination of inputs. Based on these rules we re-generate the expected inputs to the given output value. However, if encoding is done for text based inputs, decoding has to be applied for text input recovery.

IV. IMPLEMENTATION

We have initially tried it on reconstruction of logic gates input based on their output. This serves as a basic demonstration of how our proposed method would work. Let's consider the given neural network which works as OR gate.

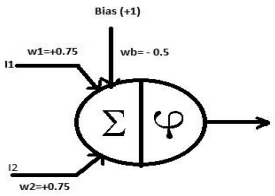

Figure 2: Neuron which computes OR function of 2 binary inputs

In Figure 2, I1 and I2 represent two binary inputs; W1 and W2 represent their corresponding weights. Wb represents the weight for the bias input. Bias input is fixed to +1 value. The activation used is binary step function which operates as follows

$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$$

We can represent the neuron as a linear equation

Y= (0.75) I1 + (0.75) I2 - 0.5……. (Eq. 1)

Now when we know that the output is 1, we generate the rules for the expected input values as follows

(0.75) I1 + (0.75) I2 – 0.5 >= 0

Therefore, we can derive that

I1 >= (0.5 – (0.75) I2) / 0.75; …… (Eq 2)

I2 >= (0.5 – (0.75) I1) / 0.75; …… (Eq 3)

As we know that the inputs would be binary, to satisfy Eq 2 and Eq 3, the following rule would be generated

I1 U I2 = 1 ….. (Eq 4)

This tells us that I1=0 and I2=0 is not the input for the given output value. Thus we would randomly select the range of acceptable inputs which satisfy the rules generated in the process.

Similarly, when inputs are not just binary it would be a continuous range of numeric values. Rules can be generated for such inputs and for generating inputs back, we can pick the average value in the acceptable range unlike the previous case where we took a random value in the acceptable range.

Continuous numeric values: Average of values in acceptable range

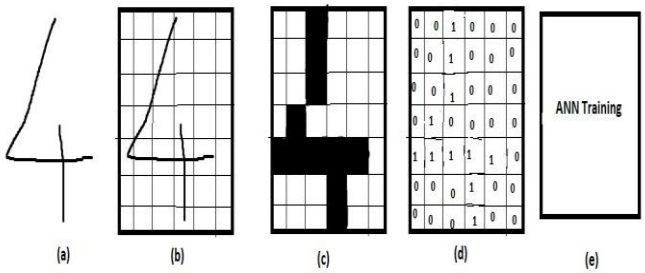

Figure 3: Process of Training Digit Recognition (Classification) Neural Network

Discrete numeric (categorical) values: random or most probable input (mode value)

The idea is tested for a self prepared image dataset for digit recognition. We prepared 10 binary images of each digit from 0 to 9 and trained those using Artificial Neural Networks to recognize the digits. It was trained by making grids of the images and assigning 0 or 1 to them. A '0' was assigned if it crosses a threshold of the number of black pixels in the grid; else the grid was assigned a '1'. The values of the grids were given as inputs to the neural network.

Figure 3 diagrammatically represents the steps in training an artificial neural network for digit recognition (classification). The trained neural network was later used to reconstruct the image given a digit (say '4') as input. Logical rules were inferred to get acceptable input values. Each grid in the image was calculated for its average grayscale value in the acceptable range. Depending upon the range of average values with respect to two threshold values (upper and lower) the grid was given color either black or white or grey. The color for each grid in the image was computed according to the following formula

Grid_value = black, if avg_value >= upper_threshold
= grey, if upper_threshold < avg_value > lower_threshold
= white, if avg_value <= lower threshold

The output image was the average of the range of the pixel values that can be acceptable satisfying the rules generated.

V. RESULTS

We were able to regenerate a noisy generalized digit image with binary grids by this reverse engineering process. We tested for all digits given as input and the following images were produced as output - generated training input images of the previously trained artificial neural network.

VI. CONCLUSION AND FUTURE SCOPE

The given procedure works fine in the reconstruction of the generalized or most probable input value for a given classification label. The inputs may be robust ranging from image values to database records. Further work may be taken up for exact data recovery in large scale and be made scalable. In particular the proposed method should be made applicable to deep learning models like Convolutional Neural Networks[7] and Recurrent Neural networks[8] to generate back the images and sequence streams trained using deep learning methods.

REFERENCES

1. Alaa Sagheer, Mohammed Zidan and Mohammed M. Abdelsamea, "**A Novel Autonomous Perceptron Model for Pattern Classification Applications**", Entropy 2019, 21, 763; doi:10.3390/e21080763
2. Venkat Raman B, K Chandrashekar, Gandhasiri Ranjith Kumar, Gurram Sudarshan, "**Memorization Approach to Quantum Associative Memory Inspired by a Natural Phenomenon of Brain**", 6th International Conference on Innovations in Computer Science & Engineering (ICICSE-2018), Springer
3. Mithun Roy, Anupam Mukherjee, Alok Basu, Pratik Kumer Halder, "**SOLVING LINEAR EQUATIONS FROM AN IMAGE USING ANN**", International Journal of Research in Engineering and Technology (IJRET), Volume: 04 Issue: 02 , Feb-2015
4. Venkat Raman B, Nagaratna P Hegde, Mallesh Dudimetla, Anjaneyulu Bairi, "**Quron: Basic Structure and Functionality**", 6th International Conference on Innovations in Computer Science & Engineering (ICICSE-2018), Springer
5. Thomason, Richmond, "**Logic and Artificial Intelligence**", *The Stanford Encyclopedia of Philosophy* (Winter 2018 Edition), Edward N. Zalta (ed)
6. Rajni Bala1, Dr. Dharmender Kumar, **"Classification Using ANN: A Review**", International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 7 (2017), pp. 1811-1820
7. Deepika Jaswal, Sowmya.V, K.P.Soman, "**Image Classification Using Convolutional Neural Networks**", International Journal of Advancements in Research & Technology, Volume 3, Issue 6, June-2014, ISSN 2278-7763
8. Alex Sherstinsky, "**Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network**", arXiv:1808.03314v4 [cs.LG], 4 Nov 2018
9. "Artificial Neural Networks" by B Yegnanarayana, Eastern Economy Ediution, PHI (Book)

