Management Analytics and Social Insights

www.masi.reapress.com

Manag. Anal. Soc. Insights. Vol. 2, No. 3 (2025) 172–181.

Paper Type: Original Article

Neuro-Fuzzy Synergy: Intuitionistic Fuzzy-Based Learning for Smarter Decisions

John Robinson P*

Department of Mathematics, PG & Research, Bishop Heber College, Affiliated to Bharathidasan University, Tiruchirappalli, 620017, Tamil Nadu, India; johnrobinson.ma@bhc.edu.in.

Citation:

Received: 12 February 2025 *Revised: 11 April 2025* *Accepted: 14 July 2025*	Robinson P, G. (2025). Neuro-Fuzzy Synergy: Intuitionistic fuzzy-based learning for smarter decisions. *Management analytics and social insights, 2*(3), 172-181.

Abstract

This paper introduces a novel Artificial Neural Network (ANN)-based Decision Support System (DSS) model that leverages intuitionistic fuzzy matrix information, aggregated using a newly proposed Hamming distance-influenced operator. The study explores the application of these innovative approaches and operators in solving Multiple Attribute Group Decision Making (MAGDM) problems under intuitionistic fuzzy environments. Key aspects of aggregation operators and ANNs, particularly the Back Propagation method, are rigorously examined to demonstrate their utility in this context. A groundbreaking aggregation operator is presented to effectively handle intuitionistic fuzzy data, providing a robust foundation for decision-making scenarios. To validate the proposed framework, a numerical example is analyzed and resolved using the ANN Back Propagation approach, showcasing improved accuracy and reduced Bias. The findings highlight the superiority of the proposed method over conventional ANN approaches in tackling MAGDM problems, paving the way for more effective and intelligent decision-making solutions.

Keywords: Multiple attribute group decision making, Artificial neural network, Aggregation operators, Back Propagation, Intuitionistic fuzzy sets, Artificial neural network.

1 | Introduction

The purpose of Artificial Neural Network (ANN) organizations is to operate as networks of parallel distributed computing. Neural networks have undisturbed fruition characteristics that are similar to biological neural networks, and they are mathematical models of information processing by nature. Artificial neurons, or simply neurons, are neural networks' fundamental processing hieroglyphs. In an ANN, signals are verified between neurons via a coupler. Every intermediary coupler has a corresponding weight that doubles the accumulated signal in a conventional neural net. To control its output signal, each neuron applies an activation function to its net input. A neural network is characterized by its architecture, which is the layout of the microcircuitry between the neurons; its activation function; and its approach to assigning weight to the connections, which can be either supervised or unsupervised. The artificial neuron found in recent years is

Corresponding Author: johnrobinson.ma@bhc.edu.in

https://doi.org/10.22105/masi.v2i3.73

similar to the biological neuron that functions in the human brain; the output from a neuron is either on or off, and the output depends only on the inputs. The modern computer is an accelerated succedent machine and is used in contrast to the remarkably parallel essence of the brain. The brain supervises some special enterprises that ensemble its design, such as arithmetic operations, which are sequential tasks to be done one after the other. In the case of vision or speech recognition problems, which are considered to be highly parallel to the brain, the brain is able to operate the complexities in parallel and easily.

The authors of [1] and [7] have worked extensively on ANN. Aggregation operators were created and used extensively for Multiple Attribute Group Decision Making (MAGDM) circumstances by the authors in [6], [8], [12]. The authors of [3], [6], [13], [14] applied state-of-the-art techniques to tackle real-world problems with fuzzy centers using machine learning applications. Although there have been a lot of ANN-related works done in the past, this chapter will focus on the innovative field of ANN with intuitionistic fuzzy sets and the use of the back propagation approach, which hasn't been covered by many authors before.

The Intuitionistic Fuzzy Power Generalized Weighted Averaging (IFPGWA) operator, introduced with the option of utilizing varieties of distance and similarity measures, is a novel and improvised aggregation operator that can be used to derive inputs in this paper's adapted ANN model of [6]. This ANN appears to be more effective and reasonable, particularly when the decision maker provides incomplete information about the problem statement. After processing the input using the proposed novel back-propagated intuitionistic fuzzy ANN, including the Bias, the outcomes are evaluated for efficacy by contrasting them with those of other ANN techniques and other ranking strategies.

2 | The Backpropagation Method in Artificial Neural Networks

Since its inception, applications of ANNs, or recursive nonlinear functions, have been transforming machine learning. In this instance, properly training a neural network is the most important step in producing a reliable model. ANN's backpropagation algorithm uses the chain rule to calculate the gradient of the loss function for a single weight. In contrast to an indigenous direct computational model, which computes the gradient but does not disclose how it is transmitted, the backpropagation skillfully computes one layer at a time. It generalizes the calculation in the delta rule.

In theory, there are the following key components for the Backpropagation method:

I. Forward pass: In the forward pass (Red arrow), data is passed through the network layer by layer. Each neuron receives inputs, computes a weighted sum, applies an activation function (e.g., sigmoid, ReLU), and passes the result to the next layer. The final output of the network is compared to the desired output to compute the error.

II. Backward pass (Backpropagation): The backward pass adjusts the weights of the network to minimize the error using Gradient Descent. The error is propagated backward from the output layer to the input layer. The gradient of the error with respect to the weights is calculated layer by layer. The weights are updated to reduce the error in subsequent iterations, ensuring the model learns effectively.

III. In a nutshell, the ANN has an input layer: Accepts features of the input data.

IV. Hidden layers: Extract patterns and relationships through nonlinear transformations.

V. Output layer: Produces predictions based on learned patterns.

VI. Loss function: Measures the difference between the predicted and actual outputs.

VII. Learning rate: Controls the step size during weight updates.

VIII. Advantages of the Backpropagation method are universal approximation: Feedforward networks can approximate any function with enough hidden layers and neurons.

IX. Adaptability: They can learn complex, nonlinear relationships in data.

X. Error minimization: Backpropagation ensures systematic error reduction through weight adjustments.

XI. Wide applications: Used in various fields such as image recognition, natural language processing, and decision-making systems.

XII. Efficiency: Optimizes performance using algorithms like gradient descent and momentum.

XIII. Scalability: Supports multiple layers, making it suitable for deep learning tasks.

Let us consider a simple numerical example to demonstrate the back propagation method. Consider the Learning rate, $\alpha = 0.25$. Consider the Inputs as: $I_1 = 0.2$, $I_2 = 0.2$, $I_3 = 0.3$.

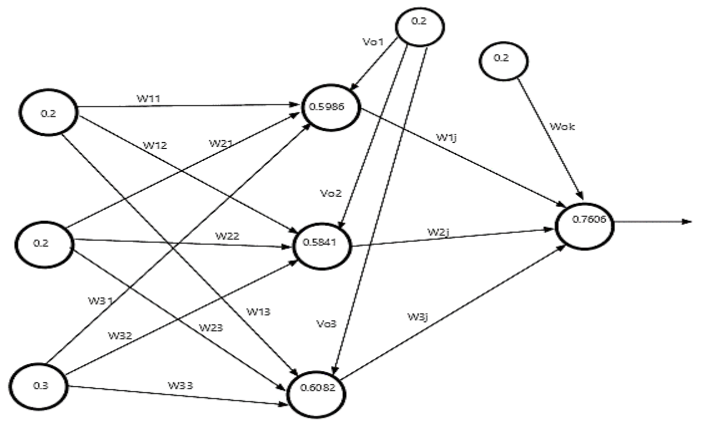

Fig. 1. 1st Iteration for simple back propagation method.

3 | The Intuitionistic Fuzzy Power Generalised Weighted Averaging Operator (With Varieties of Distance or Similarity Measures)

The Power Averaging (PA) operator was developed by Yager [11], which is a non-linear weighted average aggregation tool and is presented as follows:

$$\mathrm{PA}(a_1, a_2, \dots, a_n) = \frac{\sum_{i=1}^{n}(1+T(a_i))a_i}{\sum_{i=1}^{n}(1+T(a_i))},$$

where

$$T(a_i) = \sum_{\substack{j=1 \\ j\neq 1}}^{n} \mathrm{Sup}(a_i, a_j).$$

The $\mathrm{Sup}(i,j)$ is the support for a from b, and possesses the following three properties. 1) $\mathrm{Sup}(i,j) \in [0,1]$, 2) $\mathrm{Sup}(i,j) = \mathrm{Sup}(j,i)$, 3) $\mathrm{Sup}(i,j) \geq \mathrm{Sup}(u,v)$, if $|i-j| < |u-v|$. The similarity is higher between two values, or in other words, the more they support each other,the closer the two values are to each other. If $V_i = 1 + T(a_i)$, and $w_i = V_i / \sum_{i=1}^{n} V_i$, $w_i \in [0,1]$, $\sum_{i=1}^{n} w_i = 1$, then the PA operator is represented as: $\mathrm{PA}(a_1, a_2, \dots, a_n) = \sum_{i=1}^{n} w_i a_i$.

Definition 1 ([11]). The Power Generalized Average (PGA) operator is a map of dimension n, and it takes the form, $\mathrm{PGA}: R^n \to R$, such that,

$$\mathrm{PGA}(a_1, a_2, \dots, a_n) = \left(\sum_{j=1}^{n} w_j a_j^{\lambda}\right)^{\frac{1}{\lambda}},$$

Where

$$w_i = V_i / \sum_{i=1}^{n} V_i, V_i = 1 + T(a_i).$$

And

$$T(a_i) = \sum_{\substack{j=1 \\ j\neq i}}^{n} \mathrm{Sup}(a_i, a_j).$$

Perhaps, λ is a parameter and $\lambda \in (-\infty, 0) \cup (0, +\infty)$.

In the following, this paper introduces a novel definition for the IFPGWA operator, which functions mainly based on the distance metric or similarity measure or any other measure that provides a linear relationship between datasets for IFSs.

Definition 2. Let $\tilde{a}_j = (\mu_j, \nu_j)$ be a cluster of intuitionistic fuzzy numbers. Let $\omega = (\omega_1, \omega_2, \ldots, \omega_n)^T$ be the weighting vector of $\tilde{a}_j (j = 1,2,\ldots,n)$, where the weight vector has the property $\omega_j \geq 0, j = 1,2,\ldots,n$ and $\sum_{j=1}^{n} \omega_j = 1$. Now the novel and improvised operator called the IFPGWA operator is proposed as follows:

$$\mathrm{IFPGWA}(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n) = \left(\frac{\sum_{j=1}^{n} \omega_j \left(1+T(\tilde{a}_{ij})\right) \tilde{a}_j^{\lambda}}{\sum_{j=1}^{n} \omega_j \left(1+T(\tilde{a}_{ij})\right)} \right)^{\frac{1}{\lambda}},$$

where

$\lambda \in (0, +\infty)$, and $T(a_i) = \sum_{\substack{j=1 \\ j \neq 1}}^{n} \mathrm{Sup}(\tilde{a}_i, \tilde{a}_j)$, and $\mathrm{Sup}(\tilde{a}_i, \tilde{a}_j)$.

It's nothing but the support for $\tilde{a}_i$ from $\tilde{a}_j$, and possesses the following three properties:

1) $\mathrm{Sup}(\tilde{a}_i, \tilde{a}_j) \in [0,1]$, 2) $\mathrm{Sup}(\tilde{a}_i, \tilde{a}_j) = \mathrm{Sup}(\tilde{a}_j, \tilde{a}_i)$, 3) $\mathrm{Sup}(\tilde{a}_i, \tilde{a}_j) \geq \mathrm{Sup}(\tilde{a}_k, \tilde{a}_t)$, if $d(\tilde{a}_i, \tilde{a}_j) < d(\tilde{a}_k, \tilde{a}_t)$, Sup $(\tilde{r}_{ij}^{k}, \tilde{r}_{ij}^{l}) = 1 - d\ (\tilde{r}_{ij}^{k}, \tilde{r}_{ij}^{l})$, where d is the distance measure calculated using any available distance measure or similarity measure in the literature. The IFPGWA operator is commutative, idempotent, and bounded.

4 | Algorithm for the Proposed Backpropagation with Bias

The following is a proposed algorithm for backpropagation with:

Step 1. Input - decision matrix from the decision maker's information, where the decision maker's weights, $\sum w_{ij} = 1$. w_{ij} can be asked as an input (Or) a single weight can be made as a combination by itself on its own.

Step 2. Using aggregation operators (IFPGWA, with varieties of distance or similarity measures), these decision matrices can be made into a decision collective column matrix.

Step 3. Now that the input has been transformed to a column matrix, weights will be given as combinations to suit the hidden layers, and the learning rate will be fixed.

Step 4. Forward pass: Calculate net input to each hidden layer: $Z_{inj} = V_{oj} + \sum_{i=1}^{n} w_{ij} x_i$, (i=1,2,…,n).

Step 5. Apply activation function to calculate output: $Z_j = f(z_{inj}) = \frac{1}{1+e^{-Zinj}}$.

Step 6. To get the network output: $Y_{in\text{-}k} = w_{ok} + \sum_{i=1}^{n} z_i w_{ij}$.

Step 7. Apply activation function: $f(y_{in\text{-}k}) = \frac{1}{1+e^{-Yin-k}}$.

Step 8 ([1]). To calculate error: $_{1xj} - f(y_{in\text{-}k})$.

Step 9. Weight optimization: To update the change in weights and bias

δ_k = (Desired output–network output)^T f'($Y_{in\text{-}k}$), where

$$f'(Yin - k) = f(Yin - k)(ci - f(Yin - k)).$$
$$\Delta wjk = \alpha\, \delta k\, Yk, \quad \Delta wok = \alpha\, \delta k,$$

Where δ_k is the error correction, α is the learning rate, and Y_k is the output.

$$\delta\, in - j = \sum_{j} \delta_k \ w_{ij}, \delta j = \delta in - j\ T\ f(Zin - j)\ (1 - f(Zin - j)).$$

Step 10. Compute new weights:

$\text{wij(new)} = \text{wij(old)} + \Delta w_{ij}, \Delta w_{ij} = \alpha\delta_j x_i.$

Compute new Bias:

$v_{0j}(\text{new}) = v_{0j}(\text{old}) + \Delta v_{0j}\,, \Delta v^0{}_j = \alpha\delta_j$. (Bias)

Step 11. Now, restart the process again using new weights and new , but with the same input.

Step 12. Continue this process until we obtain the target or desired output 1.

Algorithm 1. Pseudo-code for backpropagated-ANN with bias.

```
Pseudo-code for Backpropagated-ANN with Bias:
$C_n$: n Matrix itemset of size k x m
Input {Intuitionistic Fuzzy Decision Matrices}
$A_n$ = {Collection of n Matrices of size k };
//* Aggregation Phase*//
Compute {IFPGWA aggregator (Any Distance or Similarity measure) & the Initial Weight Vector}
For (n=1; $A_n \neq \emptyset$; n++) do begin
Generate {Individual Preference Intuitionistic Fuzzy Decision Matrices, $X_n$}
//* $X_N$ is the collection of Individual Preference IF-Decision Matrices *//
Generate {Intuitionistic Fuzzy Attribute Weight Vector}
While $i \leq m$ do {Defuzzify the IF column matrix into Fuzzy Column matrix}
Generate {Collective Overall Preference Intuitionistic Fuzzy Decision Matrices}
//*Improvise the input vector by different Defuzzification function $(1 - \mu - \gamma)$, *//
Input vector //* Back Propagation: Start*//
Forward Pass: Calculate net input to each hidden layer: $a_j=\sum_j w_{ij} X_i,\ i \in N$
Error Calculation: Error = Target Output – Network Output
Backward Pass: (weights updating) $w_{ij}(\text{new}) = \Delta w_{ij} + w_{ij}(\text{old});\ \Delta w_{ij} = \alpha\delta_j X_i$
Continue the Forward Pass with updated weights and Bias from the Backward pass:
Calculate net input to each hidden layer & new Bias: $a_j=\sum_j w_{ij} X_i,\ i \in N; \Delta v^0{}_j = \alpha\delta_j$.
Continue the Weight & Bias updation until the error is minimized to a desired level
//*Choosing the MAGDM best alternative*//
Find The Weighted Arithmetic Averaging (WAA) values between the alternatives of the three input matrices and the 2nd iteration output.
Find The distance between WAA values & Net output}
While Distance values $\leq$ Threshold do generate {The best alternative}
Output: Best Alternative(s) to be chosen. End.
```

5 | Numerical Illustration for Backpropagation- Artificial Neural Network with Bias

In recent days, investing money in a proper firm or plan is essential for every common person. Consider an investment company possessing five alternatives to invest its money: A_1 is a vehicle company; A_2 is a beverages company; A_3 is a software company; A_4 is a heavy alloy company; A_5 is an oil and gas company. There are four attributes: G_1 is the risk analysis; G_2 is the cash flow analysis; G_3 is the liquidity analysis; G_4 is the debt and equity analysis. This investment strategic problem is intensively studied by three experts, providing their weighting vector for the problem, and the data provided by them are presented as decision matrices and are listed in the following:

$$\tilde{I}_1 = \begin{pmatrix} (0.4,0.3) & (0.5,0.2) & (0.2,0.5) & (0.1,0.6) \\ (0.6,0.2) & (0.6,0.1) & (0.6,0.1) & (0.3,0.4) \\ (0.5,0.3) & (0.4,0.3) & (0.4,0.2) & (0.5,0.2) \\ (0.7,0.1) & (0.5,0.2) & (0.2,0.3) & (0.1,0.5) \\ (0.5,0.1) & (0.3,0.2) & (0.6,0.2) & (0.4,0.2) \end{pmatrix}.$$

$$\tilde{I}_2 = \begin{pmatrix} (0.5,0.4) & (0.6,0.3) & (0.3,0.6) & (0.2,0.7) \\ (0.7,0.3) & (0.7,0.2) & (0.7,0.2) & (0.4,0.5) \\ (0.6,0.4) & (0.5,0.4) & (0.5,0.3) & (0.6,0.3) \\ (0.8,0.1) & (0.6,0.3) & (0.3,0.4) & (0.2,0.6) \\ (0.6,0.2) & (0.4,0.3) & (0.7,0.1) & (0.5,0.3) \end{pmatrix}.$$

$$\tilde{I}_3 = \begin{pmatrix} (0.4,0.5) & (0.5,0.4) & (0.2,0.7) & (0.1,0.8) \\ (0.6,0.4) & (0.6,0.3) & (0.6,0.3) & (0.3,0.6) \\ (0.5,0.5) & (0.4,0.5) & (0.4,0.4) & (0.5,0.4) \\ (0.7,0.2) & (0.5,0.4) & (0.2,0.5) & (0.1,0.7) \\ (0.5,0.3) & (0.3,0.4) & (0.6,0.2) & (0.4,0.4) \end{pmatrix}.$$

Calculate the input vector for ANN using the IFPGWA operator. Calculate the IFPGWA operator: IFPGWA

$(\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_n) = ((1 - (\prod_{j=1}^{n}(1-\mu_j^{\lambda})^{\frac{\omega_j\,(1+T(\tilde{r}_j))}{\sum_{j=1}^{n}\omega_j\,(1+T(\tilde{r}_j))}})^{1/\lambda}$,

1- $(1 - (\prod_{j=1}^{n}(1-\Upsilon_j^{\lambda}))^{\frac{\omega_j\,(1+T(\tilde{r}_j))}{\sum_{j=1}^{n}\omega_j\,(1+T(\tilde{r}_j))}})^{1/\lambda})$,

Suppose $\lambda = 1$, $\tilde{z}_{11} =$ (1- $((1-\mu_{11})^{\widetilde{\omega}_{11}} \times (1-\mu_{12})^{\widetilde{\omega}_{12}} \times (1-\mu_{13})^{\widetilde{\omega}_{13}} \times (1-\mu_{14})^{\widetilde{\omega}_{14}}$),

1-(1- $((1-(1-\Upsilon_{11}))^{\widetilde{\omega}_{11}} \times (1-(1-\Upsilon_{12}))^{\widetilde{\omega}_{12}} \times (1-(1-\Upsilon_{13}))^{\widetilde{\omega}_{13}} \times (1-(1-\Upsilon_{14}))^{\widetilde{\omega}_{14}}$))) =

(1-($(1-0.4)^{0.3092} \times (1-0.5)^{0.1936} \times (1-0.2)^{0.2061} \times (1-0.1)^{0.2908}$), 1-(1-((1- $(1-0.3))^{0.3092} \times$
$(1-(1-0.2))^{0.1936} \times (1-(1-0.5))^{0.2061} \times (1-(1-0.6))^{0.2908}$))) = (0.3084, 0.3770).

Similarly, all the other entries can be computed, and the results are as follows:

$$\widetilde{R}_1 = \begin{bmatrix} (0.3084, 0.3770) \\ (0.5343, 0.1802) \\ (0.4618, 0.2656) \\ (0.4264, 0.2376) \\ (0.4578, 0.1627) \end{bmatrix}.$$

$$\widetilde{R}_2 = \begin{bmatrix} (0.4114, 0.4841) \\ (0.6378, 0.2905) \\ (0.5622, 0.3464) \\ (0.5405, 0.2915) \\ (0.5601, 0.2154) \end{bmatrix}.$$

$$\widetilde{R}_3 = \begin{bmatrix} (0.3084, 0.5885) \\ (0.5343, 0.3955) \\ (0.4618, 0.4472) \\ (0.4264, 0.4110) \\ (0.4578, 0.3215) \end{bmatrix}.$$

Defuzzify the above three matrices to obtain:

$$R_1 = \begin{bmatrix} 0.3146 \\ 0.2855 \\ 0.2726 \\ 0.3360 \\ 0.3795 \end{bmatrix}.$$

$$R_2=\begin{bmatrix}0.1045\\0.0717\\0.0914\\0.1680\\0.2245\end{bmatrix}.$$

$$R_3=\begin{bmatrix}0.1031\\0.0702\\0.0910\\0.1626\\0.2207\end{bmatrix}.$$

What are the input training vectors for the ANN? And assume the weight vector is derived using the Gaussian distribution method.

$w_{11} = [\,0.15 \quad 0.22 \quad 0.25 \quad 0.20 \quad 0.18\,]$, $w_{12} = [\,0.22 \quad 0.25 \quad 0.20 \quad 0.18 \quad 0.15\,]$,
$w_{13} = [\,0.25 \quad 0.20 \quad 0.18 \quad 0.15 \quad 0.22\,]$, $w_{21} = [\,0.20 \quad 0.18 \quad 0.15 \quad 0.22 \quad 0.25\,]$,
$w_{22} = [\,0.18 \quad 0.15 \quad 0.22 \quad 0.25 \quad 0.20\,]$, $w_{23} = [\,0.15 \quad 0.25 \quad 0.18 \quad 0.22 \quad 0.20\,]$,
$w_{31} = [\,0.22 \quad 0.15 \quad 0.25 \quad 0.20 \quad 0.18\,]$, $w_{32} = [\,0.25 \quad 0.22 \quad 0.15 \quad 0.20 \quad 0.18\,]$,
$w_{33} = [\,0.20 \quad 0.22 \quad 0.25 \quad 0.15 \quad 0.18\,]$, $w_{j1} = [\,0.18 \quad 0.22 \quad 0.25 \quad 0.20 \quad 0.15\,]$,
$w_{j2} = [\,0.15 \quad 0.25 \quad 0.20 \quad 0.18 \quad 0.22\,]$, $w_{j3} = [\,0.22 \quad 0.18 \quad 0.20 \quad 0.25 \quad 0.15\,]$.

Assume the Bias weights as follows:

$w_{ok} = [\,0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2\,]$, $v_{01} = [\,0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2\,]$,
$v_{02} = [\,0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2\,]$, $v_{03} = [\,0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2\,]$.

And assume the weight vector, (And a permutation of w_{11} for all the other stages): w_{11} = $(0.15 \quad 0.22 \quad 0.25 \quad 0.20 \quad 0.18)$. Let the target output = 1. Computing the forward pass, backward pass, and updating weights, we get the output as:

$$f(\mathrm{Yin}-\mathrm{k}) = \frac{1}{1+\mathrm{e}-\mathrm{Yin}-\mathrm{k}} = \frac{1}{1.0601} = 0.9433.$$

Hence,

$$f(\mathrm{Yin}-\mathrm{k}) = [0.9433 \;\; 0.9433 \;\; 0.9433 \;\; 0.9433 \;\; 0.9433].$$

operator1 = 0.3333 x [0.3146 +0.1045+0.1031] = 0.1740.

operator2 =0.3333 x [0.2855 + 0.0717 + 0.0702] = 0.1424.

operator3 = 0.3333 x [0.2726 + 0.0914 + 0.0910] = 0.1516.

operator4 = 0.3333 x [0.3360 + 0.1680 + 0.1626] = 0.2221.

operator5 = 0.3333 x [0.3795 + 0.2245 +0.2207] = 0.2748.

To find the best alternatives, take the difference between the 2nd iteration output (0.9433) and the operator values [0.1740, 0.1424, 0.1516, 0.2221, 0.2748].

Output= [0.7693, 0.8009, 0.7917, 0.7212, 0.6685]= [A_1, A_2, A_3, A_4, A_5]

The ranking of the best alternative is: $A_5<A_4<A_1<A_3<A_2$. Therefore, A_5 is the best alternative among the five alternatives. (Comparison of 1st iteration $f(Y_{in-k})$ (Network output) = $\frac{1}{1+\exp(-2.2553)}$ = 0.9051.

With 2nd iteration $f(Y_{in-k})$ (Network output) $=\frac{1}{1+\exp(-2.8121)}$ = 0.9433).

Error calculation: Error = target output – obtained output = 1 – 0.9433.

Error = 0.0567.

Continue this process until we obtain the target or desired output ,as illustrated in the following figures:

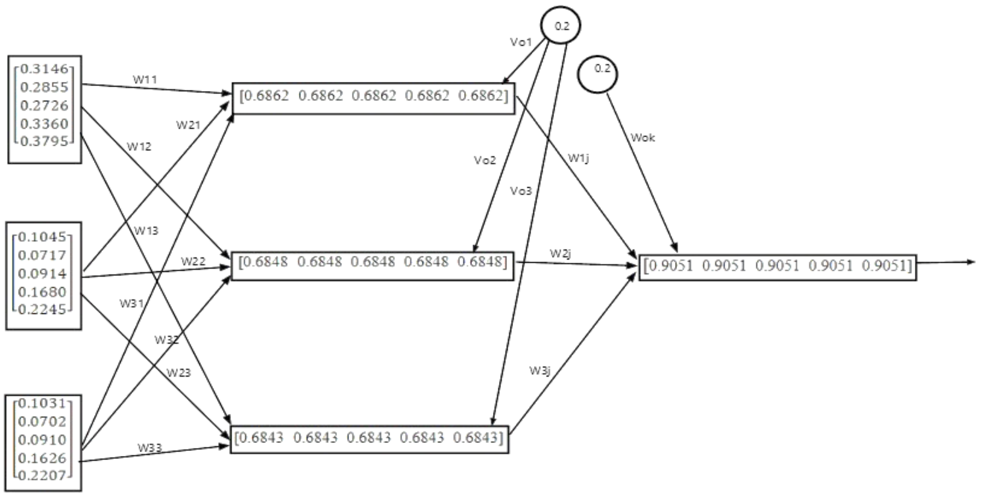

Fig. 2. 1st iteration network output.

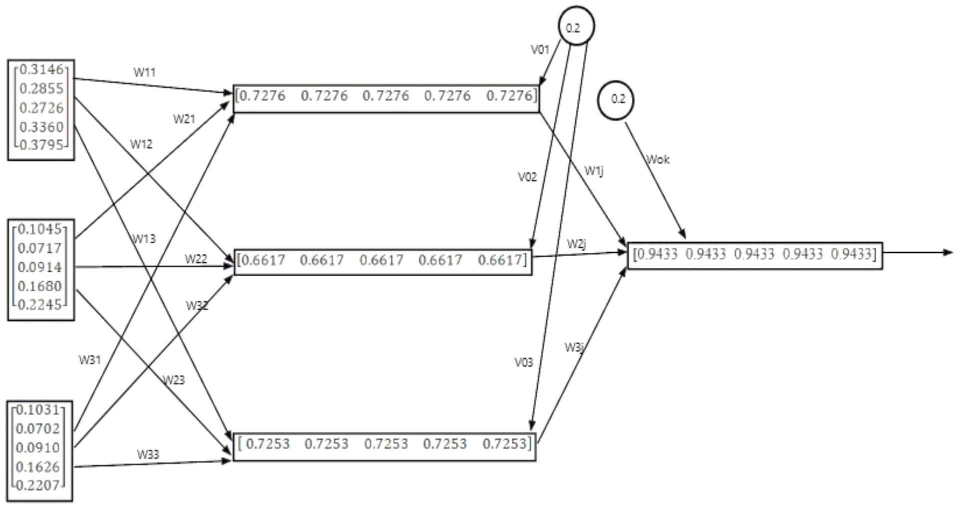

Fig. 3. 2nd iteration network output.

Table 1. Ranking of the alternatives with VF $= 1 - \mu - \gamma$ for defuzzyfication.

Iteration-1	Iteration-250	Iteration-750	Iteration-1000
Network output:	Network output:	Network output:	Network output:
0.90511195	0.99543226	0.99739419	0.99774884
Iteration: 1	Iteration: 250	Iteration: 750	Iteration: 1000
Error: 0.09488805	Error: 0.00456774	Error: 0.00260581	Error: 0.00225116
Final output	Final output	Final output	Final output
A5 < A4 < A1 < A3 < A2	A5 < A4 < A1 < A3 < A2	A5 < A4 < A1 < A3 < A2	A5 <A4 <A1 <A3 <A2

6 | Discussion

The ANN model based on the back propagation algorithm is performed as shown in *Table 1*, where different distance or similarity functions can be included in the aggregation process and utilized for transforming intuitionistic fuzzy input into a fuzzy input vector. The proposed IFPGWA operator with the opportunity of using a variety of distance and similarity measures provided the collective matrix for which the ANN was applied, and the results are tabulated. The network output in *Table 1* is done after the weight updates are performed. The number of iterations is restricted to n=1000 as observed in *Table 1*, which can be extended to any large value of n depending on the choice of the decision maker.

Moreover, from *Table 1* it can be observed that the closest value of the network output to 1 is 0.99774884, and the least error is also calculated to be 0.00225116, which is attained through the defuzzification function $1 - \mu - \gamma$, and at the 1000th iteration. The best alternative is A_5 for the corresponding choice and other iterations also yields a closest value to the target output. The same numerical example was also illustrated by the P-IFWG aggregation operator in [6], and the results were compared with existing ranking techniques and analyzed when delta and perceptron learning rules were employed. Hence, employing Back Propagated ANN also proves to be a consistent method in line with the earlier proposed methods.

7 | Conclusion

In this paper, a novel back-propagated model of ANN based on the back propagation algorithm is proposed with an improved aggregation operator for computation purposes. The new aggregation operator defined and employed in this work takes the decision matrices to the next level of processing through ANN, and finally, based on the updated weights of the ANN after successful forward and backward passes, the ranking of the decision alternatives is performed for solving the MAGDM problem.

The model proposed in this chapter was compared to the conventional MAGDM methods and techniques for handling the same investment choice problem. The new method of ANN, compared to the previous method, is very effective in a way that the complexities faced by the decision maker in the weight updation process in the MAGDM situation are handled creatively [6]. This MAGDM model, together with the proposed back-propagated Intuitionistic Fuzzy ANN with bias vector, can also be tried out with complex layers of hidden neurons, which will be reserved for future work.

Conflict of Interest Disclosure

The authors declare they have no competing interests as defined by the journal or other interests that might be perceived to influence the results reported in this paper.

Data Access

Anonymized data can be requested from the corresponding author following the journal's data sharing policies.

Financial Support

This work was conducted without external funding support.

References

[1] Atanassov, K., Sotirov, S., & Pencheva, T. (2023). Intuitionistic fuzzy deep neural network. *Mathematics, 11*(3), 716. https://doi.org/10.3390/math11030716

[2] Hájek, P., & Olej, V. (2015). Intuitionistic fuzzy neural network: The case of credit scoring using text information. In *Engineering applications of neural networks: 16th international conference, EANN 2015, Rhodes, Greece.* (pp. 337–346). Springer. https://doi.org/10.1007/978-3-319-23983-5_31

[3] Jekova, I., Christov, I., & Krasteva, V. (2022). Atrioventricular synchronization for detection of atrial fibrillation and flutter in one to twelve ECG leads using a dense neural network classifier. *Sensors, 22*(16), 6071. https://doi.org/10.3390/s22166071

[4] Krasteva, V., Christov, I., Naydenov, S., Stoyanov, T., & Jekova, I. (2021). Application of dense neural networks for detection of atrial fibrillation and ranking of augmented ECG feature set. *Sensors, 21*(20), 6848. https://doi.org/10.3390/s21206848

[5] Petkov, T., Bureva, V., & Popov, S. (2021). Intuitionistic fuzzy evaluation of an artificial neural network model. *Notes on intuitionistic fuzzy sets, 27*(4), 71–77. https://B2n.ir/xb2017f

[6] Robinson, P. J., & Leonishiya, A. (2023). Application of varieties of learning rules in an intuitionistic fuzzy artificial neural network. *International conference on machine intelligence for research & innovations* (pp. 35–45). Springer. https://doi.org/10.1007/978-981-99-8129-8_4

[7] Sotirov, S., & Atanassov, K. (2009). Intuitionistic fuzzy feed-forward neural network. *Cybernetics and information technologies*, *9*(2), 71–76. https://cit.iict.bas.bg/CIT_09/v9-2/62-68.pdf

[8] Leonishiya, A., & Robinson, J. (2023). Varieties of linguistic intuitionistic fuzzy distance measures for linguistic intuitionistic fuzzy TOPSIS method. *Indian journal of science and technology*, *16*(33), 2653–2662. https://doi.org/10.17485/IJST/v16i33.640-icrsms

[9] Leonishiya, A., & Robinson, P. J. (2023). A fully linguistic intuitionistic fuzzy artificial neural network model for decision support systems. *Indian journal of science and technology*, *16*, 29–36. https://doi.org/10.17485/IJST/v16iSP4.ICAMS136

[10] Xu, Z., & Yager, R. R. (2006). Some geometric aggregation operators based on intuitionistic fuzzy sets. *International journal of general systems*, *35*(4), 417–433. https://doi.org/10.1080/03081070600574353

[11] Yager, R. R., & Filev, D. P. (1999). Induced ordered weighted averaging operators. *IEEE transactions on systems, man, and cybernetics, part b (Cybernetics)*, *29*(2), 141–150. https://doi.org/10.1109/3477.752789

[12] Yager, R. R. (2001). The power average operator. *IEEE transactions on systems, man, and cybernetics-part a: Systems and humans*, *31*(6), 724–731. https://doi.org/10.1109/3468.983429

[13] Kumar, A., Sharma, T. K., & Verma, O. P. (2023). Detection of heart failure by using machine learning. *International conference on machine intelligence for research & innovations* (pp. 195–203). Springer. https://doi.org/10.1007/978-981-99-8135-9_17

[14] Sharma, R., Verma, O. P., & Kumari, P. (2023). Application of dragonfly algorithm-based interval type-2 fuzzy logic closed-loop control system to regulate the mean arterial blood pressure. *International conference on machine intelligence for research & innovations* (pp. 183–193). Springer.