

Efficacy of Two Hidden Layers Artificial Neural Network Synapticity for Deep Learning: A Case of Pattern Recognition

Osigbemeh, M.S.^{1,*}, Azubogu, A.C.O.², Ayomoh, M.K.O.³, Okahu, A.C.⁴

¹Dept of Electrical and Electronics Engineering, Alex Ekwueme Federal University Ndufu-Alike Ikwo

²Dept of Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria

³Dept of Industrial and Systems Engineering, University of Pretoria, South Africa

⁴Dept of Chemical Engineering, Alex Ekwueme Federal University Ndufu-Alike Ikwo

Received: 15.11.2024 • Accepted: 03.03.2025 • Published: 15.04.2025 • Final Version: 30.04.2025

Abstract: Most research works in Artificial Neural Network (ANN) commonly use single hidden layer (SHL) topology without considering the problem type, its complexity and desired depth of supervised or unsupervised learning. This could be partly due to the inherent complexities associated with the use of more than one hidden layer which in turn affects solution efficiency. However, there is often a trade-off between efficiency and effectiveness of result. When effectiveness is prioritized especially for sensitive or mission-critical systems, then multiple hidden layers can become advantageous. This research has investigated the ability of an Artificial Neural Network (ANN) with two hidden layer topology to demonstrate improved learning performance in comparison with a single hidden layer architecture ANN system. A two hidden layer (THL) Neural Network was developed and implemented using Microsoft Visual Studio programming suite and applied to a pattern recognition problem. The gradient descent optimization of the back propagation algorithm in a feed forward scheme was used in the development of the supervised ANN which consisted of thirty inputs at the input layer, two hidden layers with five nodes and a single output layer with one node for a Boolean response. Normalized images mapped into a pattern extraction template using principal component analysis (PCA) of the original images served as pre-processed inputs to the two hidden layer architecture with an initial learning rate of $\eta = 0.1$ and maximum tolerable rate of $\eta = 0.4$ for fast convergence. Iterations for validation of the feed forward back propagation algorithm using three image patterns showed that over 96% recognition of presented data was recorded. Graphical comparison of the results obtained from separate iterative sessions of the One Hidden Layer (OHL) and (THL) architectures under same input-output dataset revealed more visible features of attained deep learning by the two hidden layer architecture due to enhanced synapticity of additional nodes.

Keywords: Two Hidden Layers, Synapticity, Pattern Recognition, Deep Learning, Principal Component Analysis, Gradient Descent Optimization, Artificial Neural Network

1. Introduction

Application of Artificial Neural Networks (ANN) to pattern recognition problems where a significantly high level of data mining accuracy is required for effective decision making has recently gained substantial attention in some knowledge domains comprising machine learning, artificial intelligence, software engineering, business process forecasting, and recent advances in artificial

* Corresponding Author: eosigbemeh@yahoo.com

intelligence amongst others. ANN has over the years been used to show significant relationship between input and output parameters in non-linear statistical modelling of huge databases that appears meaningless and unlinked to the human reasoning process. Advancement in the use of ANN for connection of patterns in large databases have continued to record a heightened attention in the literature as it keeps up with the provision of unique and verifiable solutions to recent problems.

An assessment of the use of ANN in solving pattern recognition tasks has been embarked upon by [5] for which they stated that the use of ANN over other methods such as statistical approaches is “the best possible way of utilizing available sensors, processors and domain knowledge to make decisions automatically”.

Supervised learning provided by gradient descent methods such as the standard back propagation algorithm (BPA) and its variants such as the Levenberg-Marquardt Algorithm (LMA) have proved to produce a guaranteed solution to presented input/output pairs when used to teach or learn a target function. This has created the flexibility of being deployed in numerous optimization tasks in engineering. The synapticity provided by the randomly generated connection weights and biases coupling of input/output pairs further makes the BPA and its variants to imitate the natural learning process of the biological synaptic neuron network that involves mostly reflex responses to excitation signals. Though the speed and accuracy provided by these machine learning efforts including the fastest LMA, cannot match the imitated biological system which is capable of providing solutions to tasks such as pattern recognition and classification in fractions of seconds effortlessly. The interpretability of results from numerous iterations in a session of implementing a BPA or other ANN based techniques is also of considerable importance for utilization in machine learning. Critics of the BPA have pointed out that the very slow implementation of the steepest descent method and the need for a huge dataset required for training, validation and testing purposes makes ANN clearly not applicable to certain engineering problems. However, the merits of ANN outweigh its demerits; for instance, variation of the learning rate and introduction of a momentum term to the steepest descent optimization equation can help get the BPA faster to convergence within appreciable error tolerance [37]. Various performance and convergence optimizations for ANN has also been proposed by [3, 20], etc. The dependence on machine learning tools for solving society challenges must be designed in easily verifiable manner as pointed out by [38]. Pattern recognition premised on ANN for processing of pre-normalized inputs can be used for detection of changes in previously identified static images or scenes for security purposes and for analyzing spatial patterns of features in maps. It can also be used to identify defective or inconsistency in parts, goods, fracture bound products such as spectral analysis of engineering materials. Satellite imagery, weather forecasting and seismic processing can depend on ANN for faster and improved analytical deductions.

The function of data pre-processing for better data interpretation and pattern recognition by ANN have been elucidated by [7-8] in which it was stated that one of the advantages of the feature extraction (also known as dimensionality reduction) will lead to improved performance by the ANN. Another benefit according to the sources includes cleaning up deficiencies usually present in real data in the form of missing or disjointed data. The role of prior knowledge about the pre-processed data such as information which is relevant to the solution of a problem and present in the data can be harnessed to help get better results of interpretation from artificial neural networks based on improved computational intelligence [45, 48].

Pattern recognition premised on connectionist based ANNs was proposed in [28, 34], while [33] and [1] adapted the use of ANN to face recognition by extracting the features in a face image using the popular principal component analysis (PCA) and Karhunen-Loeve transformation (KLT) as used

in [11] with the feed forward neural network deployed to learn and recognize the patterns from the PCA generated features. [26] combined a hybrid of local image sampling, a self-organizing map neural network, and a convolutional neural network in a method for face recognition while [42] and [25] analyzed eigenfaces by using PCA for recognition of faces. PCA involves a mathematical procedure that transforms a number of possibly correlated variables in the image under analysis into a smaller number of uncorrelated variables called principal components. PCA for dimensionality reduction for face recognition according to the authors involved a face image being projected to several face templates called eigenfaces which is then considered as a set of features that characterizes the variation between images. Once a set of eigenfaces is computed, a face image can be approximately reconstructed using a weighted combination of the eigenfaces. The authors combined PCA and BPA to achieve over 90% recognition rate for validation of faces. Furthermore, in the analysis of textures, [12, 8] carried out pre-processing of images using Gabor filters for feature extraction from texture mosaics that are classifiable as comprising grass, sand, raffia, herring, wood and wool (center square). They achieved a high percentage classification rate on both artificial and natural textures using the combined supervised and unsupervised neural network approach to pattern recognition. A special type of pattern detection using kernel density estimation (KDE) was applied in [15]. It used spatial patterns in smooth and continuous interpolated surface maps of occurred accidents in a metropolis to estimate traffic accidents. The authors developed sixteen models using KDE and other parameters to analyze relative accident-risk cities with room for extrapolating risks in cities with unavailable data by calculating the density of events around each point and scaling the distance from the point to each event on the map. An analysis of the view finder for mobile robot navigation as presented in [19] used two separate neural network schemes to analyze the resultant path of mobile robots on encountering obstacles with the use of ultrasound range finder data. A PCA layer which condenses the input data down to a few principal components was placed between the inputs and hidden layer of the ANN. The proposed method was tested in a static obstacle environment with the aim of aiding the robot to operate within the optimal time while in search for the shortest possible and safe path to the target position.

Most usage of ANN to proffer solution to engineering problems has been largely based on supervised learning. This is due to the need for accuracy and speed in solving non-linear problems or fitting such problems as close as possible to linearity. Unsupervised learning behavior of neural networks as seen in the Hopfield network learns unique features on its own [17], thus making them closer to the complex natural neural processing ability of biological systems of man and animals which is highly stochastic and fuzzy. A Hardware implementation for solving such optimization problems using analogue processors can be found in [41]. The synapticity of neural networks as captured by Hebb in his original work on synaptic plasticity of neural networks states as follows: “When an axon of cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic changes takes place in either one or both cells such that A’s efficiency, as one of the cells firing B, is increased” [16]. Analysis of synaptic plasticity of neural networks based on spiking neurons integrating and firing was investigated by [10] in the field of neuromorphic approach to natural computation. The task involves a very large-scale integration (VLSI) implementation of a small Hebbian spike-driven plasticity network for learning stochastically actual changes in the synaptic efficacies of inputs based on multiple microcontrollers and workstations handling of inputs, control parameters, spike recording, visualizing and analyzing data. The resulting chip from the work holds huge prospects for real-time enabled synaptic plasticity of neuronal activity with provision for long term activity storage or synaptic stability. [21-23, 29 and 46] gave guidelines on the analysis of prediction errors over the output nodes and over all the records in the training set

of an implementation of the gradient descent algorithm in BPA. They also analyzed the use of the results to adjust the randomly generated weights on initialization of the ANN.

As application of results from ANN processing continue to find relevance in solving real world challenges such as wild fire propagation over several hectares of land, instantaneous building up of hurricanes, disappearance of airplanes during clashes, migration of large animals such as whales, birds, etc. in a particle swarm optimization model, there is urgent need for better understanding of the plausibleness offered by the analysis of multiple hidden units of ANN. Also, with current deployment of IoTs as sensorial units for machine learning inputs and its attendant security detection systems as noted by [18, 24, 27 and 43], the knowledge band-gap for synaptic learning for connected patterns in real data continue to increase with computational and processing capacity of computer systems.

2. Materials and Methods

The methodology adopted for this research involves transposing the salient features available on an image or picture that represents a pattern or trend unto a y by x condensed pixel format that allows for 30 inputs or cells for multiple hidden layer ANN processing. Each input cell corresponding to the salient features of pattern of interest is associated with a sub pixel comprising any of nine mono-colored shapes that have been assigned values ranging from 0.1 to 0.9 a priori. This simple template matching implementation of PCA resulted in a dimensionality reduction, enlargement or same size of the original picture or trend. The normalized-extracted features of the analyzed picture are then sent for ANN processing to identify the resulting image, pattern or hidden detail of interest using a two hidden layer ANN topology to maintain the deep learning intended and a compromise on the speed of execution of all iterations. Since the area spanned by the image was intentionally been made small as (y by x) sub pixels yet can represent vast expanse of land such as available on a map, ocean bed analysis or even small patterns that can be enlarged for greater details and clarity. In application of these condensation or enlargement, a priori knowledge or feedback on the pattern to be determined served as a guide in the image reconstruction to ensure that severe distortions to extracted image does not cause inability to fully reconstruct the pattern or trend conveyed by the extracted features which will create a lacuna in the interpretation of the solution. This is also bearing in mind that real inputs are always mostly sparse, inconsistent and stochastic. The adopted a priori feedback control and the dependence on the ability of multiple hidden layer neural networks for deep learning of the resulting extraction would lead to a guarantee interpretation and pattern recognition. The deep computational learning scheme provided by a two hidden layer neural network was used in the resulting analysis. In the two hidden layers' topology, the first hidden layer (FHL) contained three nodes while the second hidden layer (SHL) consisted of two nodes. The coupling of three to two nodes in the hidden layers and not vice versa is based on the fact that the synaptic learning of the back propagation's gradient descent optimization is better propagated from more to fewer nodes. The three-by-two hidden layer topology choice had been informed from the investigation of the optimal number of hidden nodes for most optimization problems in ANN by [14, 39]. The network has a full connectionist network of all nodes from the input layer to each node of the FHL which in turn has all the nodes of the FHL fully connected to the SHL and finally all nodes from the SHL connected to the output node. The ANN was implemented with the feed forward error back propagation algorithm and its topological schematic is depicted in Figure 1.

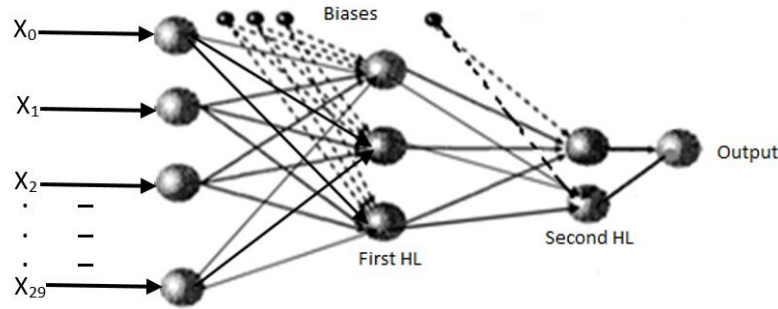


Figure 1. Architecture of the two hidden layer neural network

2.1. Gradient Descent Optimization

The gradient or steepest descent algorithm for the BPA is premised on a prediction error obtained from the sum of square of the difference between the current output and the expected output of the first forward sweep of the network. The squared prediction errors are summed over the output nodes and over all the records in the training set from the database under analysis and then used to adjust the randomly generated weights in the next forward sweep.

2.1.1 First Network Sweep

To obtain the value of the weights and direction of adjustments of the SHL nodal values, the resulting solution from the FHL nodes are deployed as inputs to the SHL as in (1)

$$net_{j(output)} = net_{j(firstHL)} = \sum w_{ij} X_{Ni} \quad (1)$$

$$y_{in} = net_{j(output)} \quad (2)$$

For $i = 1, 2$ and 3 , the obtained sum after sigmoidation by the squashing function is then used to compute SHL nodal values [4]. From the SHL, a summation of the values with corresponding generated nodal weights are used to feed the output layer after squashing again with the sigmoid function as illustrated below. The randomly generated weights w_{in} and the inputs from the FHL's output, net_j are related by the summation below to obtain nets.

$$net_{s(secondHL)} = \sum_{downstream} w_{in} y_{in} \quad (3)$$

$$= w_{cn} y_{cn} + w_{0n} y_{0n} + w_{1n} y_{1n} + w_{2n} y_{2n} \quad (4)$$

and since $y_{cn} = 1$; this becomes

$$= w_{cn} + w_{0n} y_{0n} + w_{1n} y_{1n} + w_{2n} y_{2n} \quad (5)$$

where

w_{cn} = constant value required to keep the ANN from damping to zero

The net_s computed from each node of the second hidden layer is very small due to the effect of the sigmoid squash and its randomly generated weights. Hence, to improve its synaptic strength in the current iteration and for flexibility, the resulting nets is expressed as a reciprocal function to yield

$$net_s = \frac{1}{net_{s(secondHL)}} \quad (6)$$

The reciprocal function minimized the effect of the squashing operation by the sigmoid function that was used in the upstream hidden layer and also to improve the peculiar slow convergence for steepest or gradient descent optimization of multiple hidden layers. The resulting values for nodes one and two of the SHL obtained from (6) resulted in the sigmoidation as in (7)

$$Y_{in} = \frac{1}{(1 + e^{-netNs})} \quad (7)$$

The resultant or actual output for the iteration was obtained using

$$net_{s(SecLayerOutput)} = \sum_i w_{in} y_{in} \quad (8)$$

2.1.2 Second Network Sweep

The error responsibility for adjusting the initially generated (or last upstream) weights of the second forward sweep's output was obtained from

$$\delta_{hi} = output_{hi}(1 - output_{hi})(actual_{hi} - output_{hi}) \quad (9)$$

While the weight change was computed using

$$\Delta W_{ohi} = \eta \delta_{hi}(i) \quad (10)$$

where η = learning rate. The new weight for populating each output node is given as

$$W_{ohi,new} = W_{ohi,current} + \Delta W_{ohi} \quad (11)$$

Furthermore, the first hidden layer's error responsibility was obtained from

$$\delta_{hiddenLayer1} = output_{hiddenLayer1}(1 - output_{hiddenLayer1}) \sum_{downstream} w_{jk} \delta_{hi} \quad (12)$$

The updated rule for the hidden layer's new weight is given as

$$\Delta W_{hiddenLayer1} = \eta \delta_{hi} . output_{hiddenLayer1} \quad (13)$$

While the new weight for populating each hidden node in the FHL is

$$W_{ohi,new} = W_{ohi,current} + \Delta W_{ohi} \quad (14)$$

The second hidden layer's error responsibility is computed from

$$\delta_{hiddenLayer2} = output_{hiddenLayer2}(1 - output_{hiddenLayer2}) \sum_{downstream} w_{jk} \delta_{hi} \quad (15)$$

The new weights are then updated and the weights for populating the second hidden layer's nodes were obtained from equations (13) and (14). For the third forward sweep, the iterative process in the second forward sweep is repeated and so on, until an optimal convergence epoch is sought or reached.

2.2. Feature Extraction Stage

With the analysis completed, the next stage was to generate the image of presented data pattern. The image pattern was obtained from the feature extractive mapping of all contributory cells corresponding to the object, area of interest or migration pattern and are then represented by an array of pixels with each pixel carrying an associated value ranging from 0.1 to 0.9 to indicate the level or degree of participation or influence. The resultant values of each selected pixel of the image or pattern under investigation then acts as an input into the cells of each of the thirty ANN input nodes.

A value of {0} is associated with a completely white pixel while {0.1} is assigned to a left vertically separated pixel. Also, a value of {0.2} is assigned to the right orientation pixel while a left

diagonal oriented pixel is assigned {0.3} and {0.4} assigned to a right diagonal-oriented pixel. This continues until all available shapes or cell/pixel combinations are assigned their corresponding shape values. A permutation or arrangement of the pixels on the template can then be used to approximately map a picture, pattern or image for processing onto the feature extraction template. For example, a distorted or compromised black-box recording of geographical data corresponding to flight pattern for an airplane during a crash can be used to create a feature extraction map and used to feed the ANN as possible flight pattern through which extrapolations and other deductions can be proposed to obtain an approximate direction or search angle for searching for the crashed airplane. Another example, is the fact that the build-up of several hurricanes towards the west has been attributed with migration pattern of breezes and increasing magnitude of winds moving across Sahara Africa and originating from far east of African [9, 13]. The various local readings or collected data that include wind speeds and pressure in isobars for this occurrence can be obtained and fed to the ANN. However, due to the absence of these real data sets, hypothetical data sets that leads to the development and generation of data samples that corresponds to the suppositions from public literature to enable the testing of this postulation that a two hidden neural network processing is more efficacious than a single hidden ANN. The permutation herein was achieved via careful selection of the proper pixel constituting the image or pattern of interest unto the mapping template. If x_i denotes the value of any contributing pixel in the image or pattern extraction, then the vector set $\mathbf{x}$ containing all contributing pixels for that pattern detection can be obtained as

$$\mathbf{x} = [x_1, x_2, \dots, x_k] \quad (16)$$

so that each image or pattern is represented by the summation of all the contributing pixels

$$I = \sum_{i=0}^{k-1} \mathbf{x}_i \quad (17)$$

Where, k is the total number of contributing pixels to the template of the concerned pattern. Any permutation of all the members of $\mathbf{x}$ can then be used to approximately translate images or pattern onto the extraction template as shown in Figures 2 through 7 where the matching template had been used to map the examples or illustrations (1 through 3) of winds build-up and its resultant propagation trajectory; nomadic movements of herdsmen; and analyzing MRI scans. It must be noted that the pixel values were chosen with uniformity and uniqueness as priorities in such a way that the existing pattern is utilized by all contributing pixel arrays of the corresponding images for generation of the ANN input for each illustrative example.

2.2.1 Illustration 1: Build-up of winds and its propagation trajectory

Figure 2 shows the world map and Figure 3 shows the hypothetical picture of the pattern created by the build-up of hurricane as it propagates and increase in magnitude towards the West based on available literature [9, 13]. With the pattern extraction template superimpose on the world map, a visible trajectory can be detected. The detected trajectory image can then be processed into pixel values that are then inputted into the ANN for execution and inferencing. The resulting inference can then be analyzed with extrapolation to determine the exact and eventual destination the hurricane and it's accompany disaster would be localized. Thus, the hurricane and its accompanying magnitude can be determined earlier enough thereby allowing for emergency response, disaster preparedness, outright evacuation, etc. to be initiated. This result can be shared with other detection systems and programmes in place to mitigate occurrence of disasters.



Figure 2. shows the world mapped onto a 5 by 6 pixels' template [31]

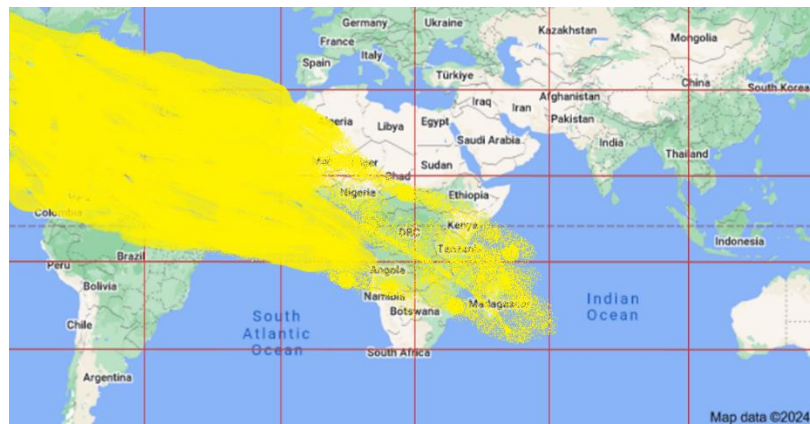


Figure 3. shows the migration pattern generated from inputs per pixel location

2.2.2 Illustration 2: Nomadic movements in Africa

Another illustration to validate the ability of the deep learning capacity of multiple hidden layers over single hidden layer for ANN processing was to use available local intel or other data acquisition models or information system to generate data for further analysis and processing to reveal trends or patterns in nomadic herdsmen's movements of cattle and other livestock in sub Saharan Africa to probably check insurgents' behavioral operations and general migration pattern.

A superimposition of a cluster of herdsmen's activities with propensity for malicious attacks on the immediate local economy or community with impunity can be deeply investigated using a pattern recognition and extractive template to graphically analyze, detect and anticipate activities that could lead to eventual economic sabotage or security compromise even with sparse and seemingly unconnected data. By leveraging on available intel from locals and law enforcement agencies a dataset corresponding to the migration pattern of a particular sect can be generated and processed. Figure 4 shows the African continent and Figure 5 shows a hypothetical pattern of the nomadic movement of the herdsmen as the investigative interest.

Another illustration could be that the coastal areas of Africa can be studied more adequately and deeply through the mapping of the ocean bed unto a feature extractive template to investigate aquatic life such as whales' migration or communication with each continent as they migrate, etc; these

acquired data can then be used to enrich training, testing and revalidating of ANN processing thereby predicting future occurrence with better precision and sensitivity.



Figure 4. shows the continent of Africa mapped onto a 6 by 5 pixels' template [32]

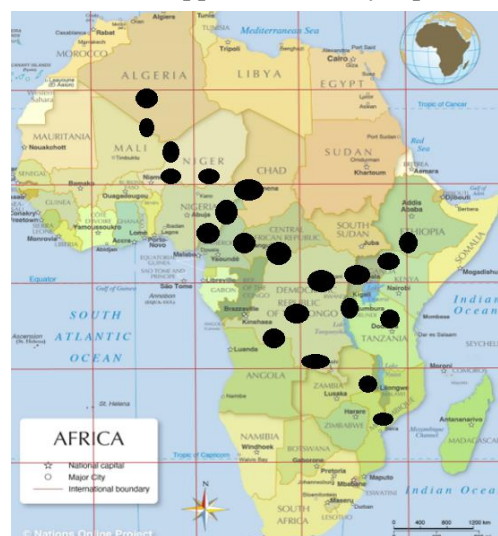


Figure 5. shows the pattern of migration of investigated interest.

2.2.3 Illustration 3: Analyzing MRI scans

A further illustration on deep learning capability of two hidden layer ANN topology is the fact that some countries in sub Saharan African have continued to witness unhealthy migration of skilled healthcare givers to Europe and elsewhere in the world thereby creating a dearth in accessibility of these professionals for the poor and vulnerable Africans. With deep learning, images or inputs of presented symptoms of patients such as MRI scans for diagnosing brain tumors was used to point a current diagnosis of a patient by training the ANN with an a priori database of known symptoms, presentations and inferences in line with views from [2, 40, 44 and 47]. This preliminary result can then be used with caution to deduce new cases, train and make reverifiable suggestions to available semi-skilled professionals during emergency and no alternative situations. Figure 6 shows MRI scans of a patient under diagnosis at different views and Figure 7 shows a hypothetical pattern that can be

used for further analysis of the generated patient's presentation. In general, broader implication of these findings is the plausibleness of otherwise missing details or features in depending on SHL only to be more prominent when revalidation or comparison is made using the THL ANN. This is because these illustrations show that the THL network topology would always outperform the SHL topology for problems with large surface area as shown in illustration 1 and 2 or more participatory pixel elements as shown in 3 provided that the computational edge from processing is attained.

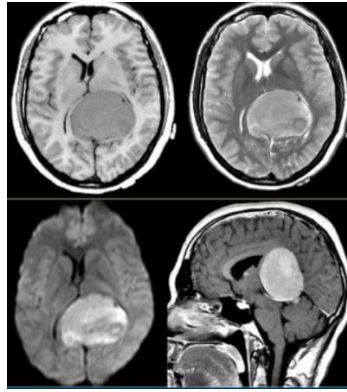


Figure 6. shows MRI scans of a patient under diagnosis [30]

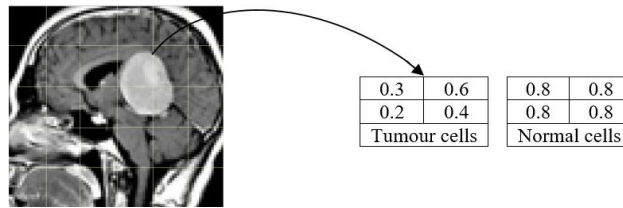


Figure 7. shows a pixel generated image of the investigated MRI scan.

2.3. Template Matching Stage

The extracted image pixels for the buildup of hurricane as it migrates from Africa to the Americas based on hypothetical data shown in Figure 3 was used to generate vector values for input into the ANN to attain the winning input-output pairs of $X_{hurricane} = [0.9, 0.7, 0.1, 0.0, 0.0, 0.0, 0.9, 0.9, 0.7, 0.1, 0.0, 0.0, 0.5, 0.7, 0.9, 0.7, 0.1, 0.0, 0.0, 0.0, 0.3, 0.4, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.0]$.

This constitutes a unique identifier for all normalized input images from local collection system from left of template to right of template for each template row thus constituting the inputs into the ANN pattern recognition or identification system. Each vector x_i represents the i th input on the ANN input node that has been designed to accept thirty different pixel inputs. The dimensionality reduction provided by the pixel rearrangement or reduction for such a huge and large landmass must be analyzed with known or a priori data to avoid significant image distortion which will lead to error in the ANN pattern recognition system of not been able to converge to some meaningful convergence. The accuracy and quality of local input for ANN processing must be determined and preprocessed from uncertainty before it can be used as part of ANN input and analysis. However, for complex pictures extraction, cascading of multiple matching templates can be used as inputs into a massive training ANN such as proposed by [39]. It is also necessary for large pixel dependent images recognition such as available on maps be processed for high consistent resolution before further analysis and verification of analysis.

Furthermore, the pixel values generated by the hypothetical migration pattern is represented by the following

	pixels	values:	$X_{\text{Migration}} =$
			$[0.0, 0.1, 0.0, 0.0, 0.0, 0.0, 0.8, 0.3, 0.0, 0.0, 0.0, 0.0, 0.9, 0.6, 0.2, 0.0, 0.0, 0.5, 0.9, 0.0, 0.0, 0.0, 0.6, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$

The pixel values of generated brain tumor data was of the form represented by the following: $X_{\text{Tumor}} = [0.8, 0.6, 0.6, 0.6, 0.6, 0.8, 0.8, 0.7, 0.9, 0.3, 0.6, 0.7, 0.9, 0.6, 0.8, 0.2, 0.4, 0.7, 0.5, 0.5, 0.6, 0.6, 0.8, 0.8, 0.8, 0.7, 0.8, 0.8, 0.5, 0.7]$ which represented the extracted pixel values from the hypothetical patient's MRI scan for onward processing by the ANN. The feature extractive process was also been hinged on the propositions from [35].

3. Results and Discussion

Training and validation of the neural network was premised on a developed ANN solution engine with the nomenclature SOSIC Artificial Neural Network Graphic User Interface (GUI). This was based on a data set that comprised various combinations of normalized values for identification of contributing images and generated pixel values after due considerations of inconsistencies in data samples.

A 2.0 GHz Intel(R) Pentium(R) M Processor with 2GB memory size was used to host the developed software that was designed in the Microsoft Visual Studio platform using VB 6.0. The pattern recognition of 60 input samples segmented into 20 fed samples per pattern was found to be high for trained and validated data set resulting in 97% identification of input samples used in the training and validation process. Extreme values within the interval $\{0 < w < 1\}$ were avoided in the randomized generation of weights (w) for each processed iteration and randomly generated weights were ensured to be with adequate spread within the weight generation interval. Results obtained from training and validation of data sets using the template match for pre-normalized identifications of sample images inputted into ANN and used in this research for illustration 1, 2 and 3 are shown in Tables 1 and 2 respectively. The averages of the number of iterations or epochs required by the network to converge in order to generate validated weights for all three object image samples are as presented in Tables 1 and 2.

Table 1. Convergence Table for Two Hidden Layers ANN.

Two HL	Pixel Template Pattern	Optimal Convergence Epoch at $\eta = 0.1$	Early Convergence Epoch at $\eta = 0.1$	Optimal Convergence Epoch at $\eta = 0.4$
	Illustration 1	697	313	138
	Illustration 2	668	270	144
	Illustration 3	695	271	138
Average =		687	285	140

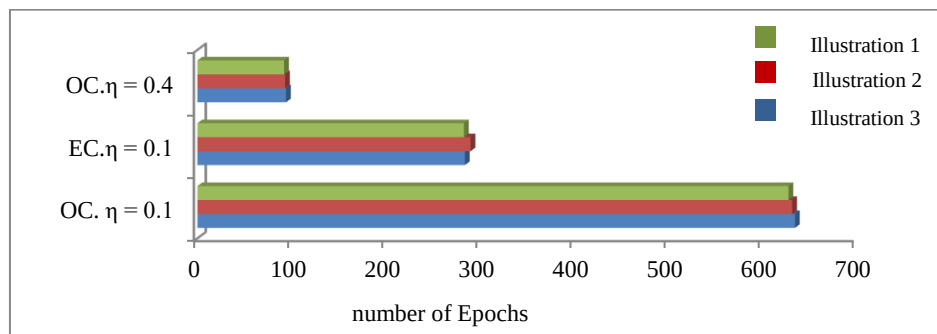
On comparing the outputs obtained from the one hidden layer (OHL) and two hidden layer (THL) ANN architectures under same input conditions for both simulations, the OHL architecture which consisted of a single layer of three nodes was seen to reach a convergence point earlier than the THL architecture in all the optimal iterations. However, the OHL appears to be of a better choice with respect to deep learning for early convergence simulation trials for Illustration 2 and 3 template patterns. However, with further iteration to a point of optimality, the THL exhibited a deep learning behavior over the OHL architecture. Deep learning behavior is the ability of an ANN to show more aspects or features of a learned pattern due to an enhancement in the ANN's topology or architectural framework allowing for improved validation and testing after training. These outputs as shown in Figure 8 for the OHL and Figure 9 for the THL is a depiction of the degree of sensitivity of the two hidden layer networks to reaching a peak level in their respective learning abilities and attributes. For epochs at $\eta = 0.1$ considering early convergence, the THL recorded a maximum value of 313 for a pattern representing Illustration 1 while the minimum was 270 for Illustration 2 patterned template. Similarly, for early convergence, the OHL obtained its peak value of 290 for Illustration 2 template and lowest value of 283 for Illustration 3 pattern. In the case of optimal convergence for epoch at $\eta = 0.1$, the THL

had a peak value of 697 and lowest value of 668 for Illustration 2 template pattern. All the optimal convergence values obtained by the THL are greater than the corresponding values obtained for optimal convergence for OHL. The OHL had a peak value of 635 for Illustration 1 pattern and least value of 628 for Illustration 3 patterned template. Furthermore, for epoch at $\eta = 0.4$ which has been investigated to speed up iteration and convergence by [36], THL had a peak convergence of 144 for Illustration 1 and 138 for both Illustration 1 and 3 extracted templates. From the OHL, at epoch $\eta = 0.4$, a peak value of 94 was attained for Illustration 1 patterned template while the lowest convergence of 92 was recorded from Illustration 3 patterned pixel template. The peak values for convergence varied with changing epoch values for the image patterns adopted in both cases of hidden layer network.

Table 2. Convergence Table for One Hidden Layer ANN.

One HL	Pixel Template Pattern	Optimal Convergence Epoch at $\eta = 0.1$	Early Convergence Epoch at $\eta = 0.1$	Optimal Convergence Epoch at $\eta = 0.4$
	Illustration 1	635	284	94
	Illustration 2	632	290	93
	Illustration 3	628	283	92
Average =		632	286	93

Furthermore, from Table 1 considering $\eta = 0.1$, an average of 687 epochs was reached by the THL network for the optimal convergence while 285 epochs were attained in the early convergence. Also, for $\eta = 0.4$, the THL network recorded an average of 140 epochs. The averages for the OHL for $\eta = 0.1$ included 632 and 286 for optimal and early convergence respectively while 93 epochs were attained for optimal convergence when $\eta = 0.4$. These experiments are not efficiency driven rather they are premised on effectiveness because of the connection with the number of iterations attained at the point of optimal convergence. This is obvious from the SOSIC ANN GUI software in which iterations relates linearly with time. In both networks, it was noticed that the convergence speed increased with increasing learning rate η and peaking at $\eta = 0.4$ for maximum error tolerance in accordance with [36]. Figures 8 and 9 respectively depicts the graphical representations of the THL and OHL convergences for both early and optimal iterations as shown in ables 1 and 2 respectively.



OC. = Optimal Convergence at, EC. = Early Convergence at

Figure 8. Iterative result for one hidden-layer ANN

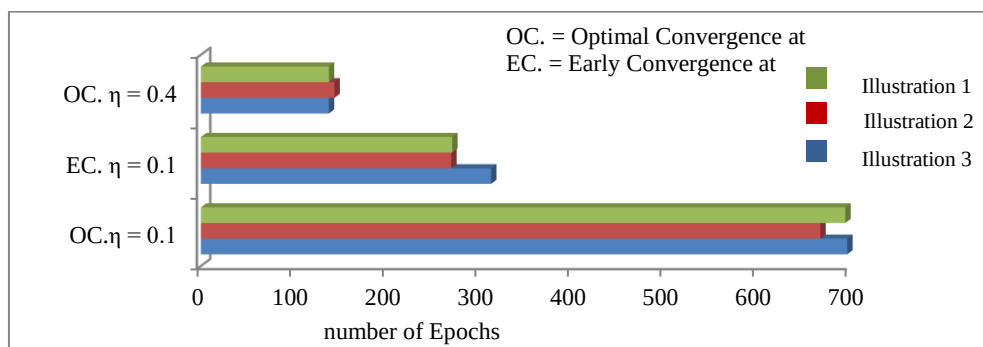


Figure 9. Iterative result for two hidden-layers ANN

4. Conclusion

Following a unique assignment of values to represent the principal components of fed inputs to a two hidden layer artificial neural network (ANN) topology, different patterns in three identification tasks were recognized in this research. The inputs representing picture images that required pattern recognition were pre-normalized using principal component analysis and mapped unto a feature extraction template a 6 by 5 area dimensionality for further processing by ANN. The condensed (6 x 5) pixel template and extracted input from the image acquisition process was used as pre-normalized training inputs into the designed artificial neural network. Preliminary results of testing and validation using real world scenarios presented as normalized images of illustrations 1 – 3, showed over 96% recognition rate when optimal iteration weights were used to train, test and validate the ANN in a single hidden layer topology (SHLT) and a two hidden layer topology (THLT). The enhanced synapticity resulting from the two hidden layers' synchronization for the THLT over the SHLT showed a significant deep learning in their performance metrics during comparison. With the THLT and SHLT analysed under the same simulation conditions of same input-output pairs at varied learning rates, η of 0.1 and 0.4 for the hidden layer neural network processing. The resulting analysis and inferences or extrapolation can be quickly used with caution to fine-tune decision making or adaptation to real problem solving when presented with complex seemingly unconnected data from real sources thus increasing the possibility of using the back propagation algorithm that implements gradient descent optimization for solving real problems in real-time.

Authors Declaration

The authors declare that there are no competing interests.

References

- [1] M. Agarwal, N. Jain, M. Kumar and H. Agrawal, "Face Recognition Using Eigen Faces and Artificial Neural Network". *International Journal of Computer Theory and Engineering*, vol.2, no. 4, pp. 624–629, 2010.
- [2] F. Aires, C. Prigent, W. Rossow, "Neural Network Uncertainty Assessment Using Bayesian Statistics: A Remote Sensing Application," *Neural Computation*, vol. 16, no. 11, pp. 2415-2458, 2004.
- [3] T.F. Awolusi, O.L. Oke, O.O. Akinkulere, A.O. Sojobi and O.G. Aluko, "Performance Comparison of Neural Network Training Algorithms in the Modeling Properties of Steel Fiber Reinforced Concrete," *Heliyon*, Vol. 5, no. 1, 2019. Available: doi: 10.1016/j.heliyon.2018.e01115,
- [4] AR. Barron. "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Transaction on Information Theory*, vol. 39, no. 1, pp. 930-945, 1993. doi: 10.1109/18.256500.
- [5] J.K. Basu, D. Bhattacharyya, & T. Kim. "Use of Artificial Neural Network in Pattern Recognition," *International Journal of Software Engineering and Its Applications*, vol. 4, no.2, pp. 23–33, 2010.
- [6] C.M. Bishop, "Neural Networks for Pattern Recognition", Clarendon Press, Oxford, 1995.
- [7] C.M. Bishop, "Neural Networks: A Pattern Recognition Perspective. In E. Friesler and R. Beale, (eds)" *Handbook of Neural Computation*, Oxford University Press, 1996.
- [8] A.C. Bovik, M. Clark and W. S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp.55–73, 1990.
- [9] T.N. Carlson, "Synoptic histories of three African disturbances that developed into Atlantic hurricanes," *Monthly weather review*, vol. 7, no. 3, pp. 256–276, 1967.
- [10] E. Chicca, D. Badoni, V. Dante, M. D'Andreagiovanni, G. Salina, et al. "A VLSI Recurrent Network of Integrate-and-Fire Neurons Connected by Plastic Synapses with Long-Term Memory," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1297–1307, 2003.
- [11] M.R. Gallagher, "Multi-layer Perceptron Error Surfaces: Visualization, Structure and Modelling," Ph.D. Thesis from the Department of Computer Science and Electrical Engineering, University of Queensland, Australia, 2000.
- [12] H. Greenspan, R. Goodman and R. Chellappa, "Texture Analysis via Unsupervised and Supervised Learning," *Proceedings of the 1991 International Joint Conference on Neural Networks*, vol. 1, pp. 639–644, 1991.

- [13] E. Friedl, "Hurricanes: Types, Formation, Causes & Effects," 2024. Available: <https://study.com/academy/lesson/hurricanes-types-formation-causes-effects.html>
- [14] M.T. Hagan, H.B. Demuth, M.H. Beale and O. DeJesus, "Neural Network Design, 2nd Edition," 2013.
- [15] S. Hashimoto, S. Yoshiki, R. Saeki, Y. Mimura, R. Ando, et al. "Development and Application of Traffic Accident Density Estimation Models using Kernel Density Estimation," *Journal of Traffic and Transportation Engineering (English Ed.)*, vol. 3, no. 3, pp. 262–270, 2016.
- [16] D.O. Hebb, "The Organization of Behaviour," Wiley, New York, 1949.
- [17] J.J. Hopfield, "Neurons with Graded Response have Collective Computational Properties like Those of Two-state Neurons," *Proceedings of National Academy of Science*, vol. 81, pp. 3088–3092, 1984.
- [18] W. Hu, L. Cao, Q. Ruan and Q. Wu, "Research on Anomaly Network Detection Based on Self-Attention Mechanism," *Sensors*, vol. 23 no. 5059, 2023. Available: <https://doi.org/10.3390/s23115059>
- [19] D. Janglová, "Neural Networks in Mobile Robot Motion," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 15–22, 2004.
- [20] O. Kisi and E. Uncuoglu, "Comparison of Three Back-propagation Training Algorithms for Two Case Studies," *Indian Journal of Engineering and Materials Science*, vol. 12, pp. 434–442, 2005.
- [21] D.T. Larose, "Discovering Knowledge in Data: An Introduction to Data Mining". John Wiley & Sons, Inc. New Jersey, U.S.A, pp. 60–214, 2005.
- [22] D.T. Larose, "Data Mining Methods and Models" John Wiley & Sons, Inc., New Jersey, U.S.A, pp. 204–239, 2006.
- [23] M. Kubat, "An Introduction to Machine Learning (3rd ed.)," Springer, 2021.
- [24] J.A.N. Lansky, S. Ali, M., Mohammadi, M. K. Majeed and A.M. Rahmani, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021. <https://doi.org/10.1109/ACCESS.2021.3097247>
- [25] P. Latha, L. Ganesan and S. Annadurai, "Face Recognition using Neural Networks" *Signal Processing: An International Journal (SPIJ)*, vol. 3, no. 5, pp. 153–160, 2009.
- [26] S. Lawrence, C.L. Giles, A.C. Tsoi, and A.D. Back, "Face Recognition: A Convolutional Neural Network Approach," *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition*, vol. 8, no. 1, pp.98–113, 1993.
- [27] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, pp. 4396, 2019.
- [28] H. Lu, R. Setiono and H. Liu, "Effective Data Mining using Neural Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 6, pp. 957–961, 1996.
- [29] M.T. Mitchell, "Machine Learning," McGraw – Hill Science/Engineering/Math Publishing, USA, pp. 81–127, 154–200, 1997.
- [30] Radiology Assistant, 2024. Available: <https://radiologyassistant.nl/neuroradiology/brain-tumor/systematic-approach>
- [31] Map Data, "World Map," 2024. Available: https://www.google.com/maps/@3.5382859,-11.9163454,3z?entry=ttu&g_ep=EgoyMDI1MDEyMi4wIklXMDSoASAFQAw%3D%3
- [32] Nations Online Project, "Political Map of Africa," 2024. Available: <https://images.app.goo.gl/BeDX3Bg2LAW39Jx5>
- [33] S.A. Nazeer, N. Omar, K.F. Jumari and M. Khalid, "Face Detecting using Artificial Neural Networks Approach," *First Asia International Conference on Modeling & Simulation*, 2007.
- [34] R. Nisbet, J. Elder and G. Miner, "Handbook of Statistical Analysis and Data Mining Applications," Burlington, Massachusetts: Academic Press-Elsevier Inc. Burlington, 2009.
- [35] M.S. Osigbemeh, C.O. Ohaneme and H.C. Inyiama, "An Algorithm for Characterizing Pre-fuzzified Linguistic Nuance using Neural Network". *International Journal of Speech Technology*, vol. 20, no. 2, pp.355–362, 2017. DOI.10.1007/s10772-017-9413-5.
- [36] M.S. Osigbemeh, C.C. Okezie and H.C. Inyiama, "Performance Metrics of various Topologies of a Feed Forward Error-Back Propagation Neural Network," *Journal of Engineering and Applied Science (JEAS)*, vol. 12, no. 10, pp. 94–102, 2017.
- [37] M.S. Osigbemeh, C.C. Osuji, M.O. Onyesolu and U.P. Onochie, "Comparison of the Artificial Neural Network's Approximations Based on the Levenberg-Marquardt Algorithm and the Gradient Descent Optimization Method on Datasets." *Artificial Intelligence Evolution*, vol. 5, no. 1, pp.24–38, 2024. DOI: <https://doi.org/10.37256/aie.5120243781>.

- [38] C. Rudin and K. L. Wagstaff, "Machine Learning, Special Issue on Machine Learning for Science and Society," Springer, 2013. DOI 10.1007/s10994-013-5425-9
- [39] K. Suzuki, J. Shiraishi, H. Abe, H. MacMahon and K. Doi, "False-positive Reduction in Computer-aided Diagnostic Scheme for Detecting Nodules in Chest Radiographs by Means of Massive Training Artificial Neural Network," *Academic Radiology*, vol. 12, no. 2, pp. 191–201. doi:10.1016/j.acra.2004.11.017.
- [40] J. Swets, "Measuring the Accuracy of Diagnostic Systems," *Science*, vol. 240, no. 4857, pp.1285-1293, 1988.
- [41] D.W. Tank and J.J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit and a Linear Programming Circuit," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 5, pp. 533–541, 1986.
- [42] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 7–86, 1991.
- [43] M. Wazid, A. Das, S. Shetty, P. Gope and J. Rodrigues, "Security in 5G-Enabled Internet of Things Communication: Issues, Challenges and Future Research Roadmap," *IEEE Access*, vol. 9, pp. 4466–4489, 2021. Available: <https://doi.org/10.1109/ACCESS.2020.3047895>
- [44] I.H. Witten, E. Frank and M.A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. Burlington, Massachusetts: Morgan Kaufmann Publishers-Elsevier Inc., 2011.
- [45] Y. Wu, X. Zhang and P. Jia, "Recent Advances in Machine Learning and Computational Intelligence," *Applied Sciences*, 2023. Available: <https://www.mdpi.com/journal/applsci/special issues/2IEDK599AN>
- [46] B. Yegnanarayana, "Artificial Neural Networks for Pattern Recognition," *Sadhana*, vol. 19, no. 2, pp. 189–238, 1994.
- [47] Z. Zhou, N.V. Chawla, Y., Jin and G.J. Williams, "Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–73, 2015. Doi: 10.1109/ MCI.2014.2350953
- [48] D.K Sharma, H. S. Hota and A.R. Rababaah, "Machine Learning for Real World Applications," Springer, 2024. Available: <https://doi.org/10.1007/978-981-97-1900-6>