Review of Different Types of Neural Network Architectures

T.L. Makosso[1*], A. Almaktoof[1], K. Abo-Al-Ez[2]
[1]Cape Peninsula University of Technology (CPUT), Cape Town, South Africa
[2]University of Johannesburg, Johannesburg, South Africa
[*]corresponding author's email: 218283695@mycput.ac.za

Abstract – *Innovative technologies come with such a huge amount of data that can only computerize with fast and more complex software. As time went by, more complicated problems arose such as pattern recognition, machine learning and prediction and unfortunately the conventional computer system was unable to carry out such tasks. Which leads to intelligent computational systems such as artificial neural networks. It is developed so that artificial neurons combined together would behave like a human brain. Different layers of mathematical processing are used to provide an accurate response regarding the input. Based on their architecture, training or learning methodology, and activation function, these artificial neurons are classified. The arrangement of neurons to create layers and the connections between and within the layers make up the neural network architecture. This paper aims to provide a clear and concise understanding of several types of architecture and its applications. Five mains' architectures and their applications and gaps are presented in this paper. The different architectures are: feed-forward, Convolutional and, recurrent neural networks, Auto encoder and generational encoders and Deep reinforcement learning architecture.*

Keywords: *Deep learning, Neural network architectures*

Article History
Received 15 June 2024
Received in revised form 12 August 2024
Accepted August 18 August 2024

I. Introduction

The brain's extremely high computational efficiency, or its ability to produce a vast amount of computing output for a ridiculously small amount of power consumption. According to studies, the brain is more efficient at processing information than electronic computers [1].

The fundamental units of the nervous system are neurons. They send and receive signals to various physical regions. Both physical and electrical methods are used to accomplish this. Information is transmitted through a variety of distinct types of neurons. Information from the body's sensory receptor cells is transmitted to the brain by the sensory neurons. On the other hand, motor neurons carry messages from the brain to the muscles. Information is transferred between the body's many neurons by interneurons [2].

When a neuron gets a lot of input from other neurons, the signals build up until they surpass a certain level. The neuron is prompted to send an impulse up its axon, known as an action potential, after this threshold is crossed. Electrostatically charged atoms (ions) flow across the axon membrane to produce an action potential. The membrane potential of neurons is the difference in the negative charge between the neurons and the surrounding fluid while the neurons are at rest. Typically, it is -70 millivolts (mV) [3].

With the advancement of technology, the need for fast computing systems like neural networks became inevitable to run an infinite amount of data [4]. This research aims to contribute to the existing body of knowledge by classifying five main types of neural network architecture which are: feed-forward, Convolutional and recurrent neural networks, Auto encoder and generational encoders and Deep reinforcement learning architecture. This paper is organized as follow: Section one feed-forward networks, Section two Convolutional neural networks, Section three recurrent neural networks, Section four Auto encoder and generational encoders and Section Deep reinforcement learning architecture five and conclusion [5].

II. Feedforward Neural Network

Feed-forward networks, a type of artificial neural network, lack looping nodes. It is also known as a multi-layer neural network since all input is simply sent forward.[6].

Data flow involves the receipt of data at input nodes, transmission across cover layers, and output at nodes. There are no linkages in the network that could be manipulated to convey data back from the output node. Functions are approximated functions in the following way; $y = f(x)$ is used the algorithm for the classifier calculations

A. Working principle

The feed forward neural network might resemble a single layer perceptron when it is simplified as presented in Fig. 1.

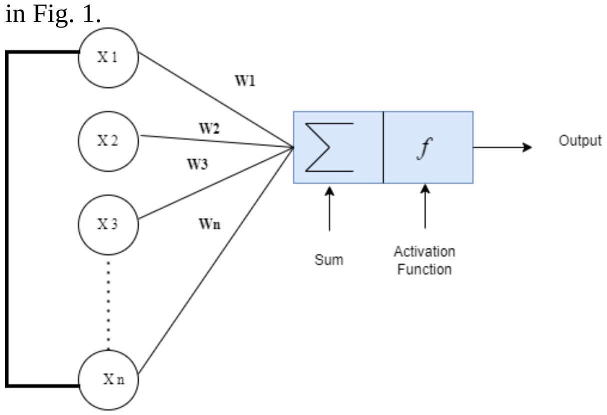

Fig.1. Model of a neural network

In this model, weights are multiplied by inputs as they get into the layer. Next, using the weighted input values, the sum is computed. The weights are used the outputs with the expected values [7]. Training and learning result in gradient decline. Additionally, multi-layered perceptrons modify their weights. It is known as back-propagation. In such a scenario, the network's hidden layers will adjust based on the output values produced by the top layer.

B. Layers of the feedforward neural network

It is made of different elements such as**:** Input, Output and hidden layer, Rectified linear unit, Neurons, Sigmoid and activation function [8].

C. Input Layer

This layer's neurons receive data and send it to the network's other layers. The number of neurons in the input layer must match the feature or attribute counts in the dataset.

D. Output Layer

This layer represents the forecasted feature based on the type of model being developed.

E. Hidden Layer

They are employed to separate the layers of input and output. Several hidden layers could exist, depending on the type of model.

Numerous neurons at buried levels alter the input before it is transmitted to the following layer. In this network, weights are constantly changed to facilitate prediction.

F. Neurons

Neurons are connected by a weight that measures their quantity or strength. Weight typically has a value between 0 and 1, with a range of 0 to 1. In feed-forward networks, which were later modified from real neurons, artificial neurons are utilized. Artificial neurons make up a neural network [9]. The two methods that neurons work is by first creating weighted input sums and then activating those sums to make them normal. Linear and nonlinear activation functions are the two varieties. Neurons have weights determined by their inputs. Throughout the learning phase, the network examines these weights.

G. Activation Function

Neurons are in charge of making decisions in this domain. They choose whether to make a linear or non-judgment based on the activation function. It avoids the cascade effect by moving through so many levels, which keeps neuron outputs from rising. Three primary categories of activation functions exist: sigmoid, Tanh, and Rectified Linear Unit (ReLu).

H. Sigmoid

Input values are set 1 between.

I. Tanh

A value between -1 and 1 mapped to the input values.

J. Mutilayer Perceptrons

Multilayer Perceptrons (MLPs), a particular family of layered feedforward networks, have come to be designed using the back-propagation algorithm. A multilayer perceptron, as depicted in Fig.. 2 contains two layers that connect the network: an input and output layer respectively made up of source nodes and neurons (i.e., computation nodes). The multilayer perceptron typically comprises one or more hidden layers in addition to these two layers since these neurons are not easily accessible [17]. Important features present in the input data are extracted by the hidden neuron.

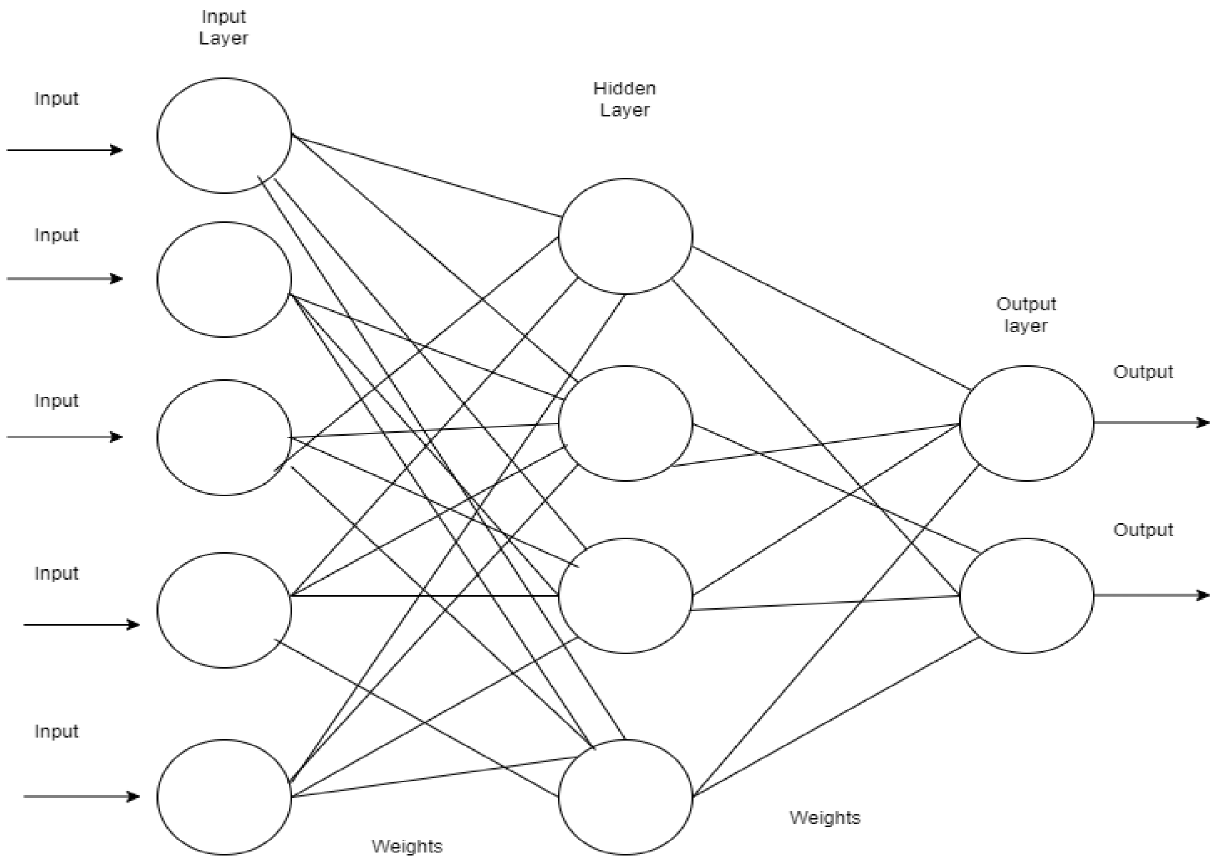

Fig.. 2 Model Multiple layer perceptrons

K. Layers of the feedforward neural network

Only positive values can flow through this function. Negative values get mapped t. In [10], a research was carried oud to enhance gender voice recognition accuracy and address data imbalance. Compared to a binary single-layer perceptron (BSLP), a binary multi-layer perceptron (BMLP) is a type of feedforward artificial neural network (ANN) with extra layers. A spectrogram is a graphic representation of the frequency content of an audio signal that varies over time. This paper [11], explains how a BMLP can be applied to the male and female voice spectrograms using various methods, such as copying, adding noise, and SMOTE, to solve the imbalanced data problem and detect the gender. The approaches are tested using Keras implementation. The classification performance of the neural network is enhanced by 5% using this method.

In this study [12], two new acoustic models were introduced, DFSMN and CTC, using the CNN model as a foundation. This acoustic model applied speech recognition techniques to detect mispronunciations of Mandarin among Tibetan students. It employed extended initial final (XIF) as bias primitives and designed 64 bias types to further enhance the detection performance. The findings of the experiment demonstrate that the strategy presented in this paper, with a DA of 88.02%, FRR of 7.95%, and FAR of 25.74%, can accurately identify mispronunciation and offer appropriate feedback**.** There is currently no theoretical option for network scale guidance when artificial neural networks are widely used to solve real-world issues. To determine the proper network scale, users frequently employed empirical or trial-and-error techniques. When the scale is too large, overfitting and resource and time waste result from the neural network having calculating many parameters during forward and back propagation. An undersized scale will result in under fitting. This paper [13], proposed a straightforward, easy, and efficient method to realize network growth in order to address this issue. A growth function considers several variables and is intended to continuously modify the network architecture's the expansion.

This research [14], proposed a convolutional deep feedforward network (C-DFN). The four fully-connected layers of C-DFN are arranged after a trainable feature extractor called the Gabor-convolutional layer. Tests are carried out to assess the suggested network's effectiveness in comparison to three alternative structures: the deep feedforward network, the convolutional deep belief network, according to the experiment results, C-DFN had the lowest average misclassification rate (9.41%) for classifying images.

In [15], the use of a targeted piecewise feedforward neural network (PW-FNN) to reduce the distortions that are both nonlinear and linear in the direct modulated laser

(DML) based intensity modulation direct detection (IM-DD) system. With this equalizer, a 16.8GHz DML with a bit error rate (BER) below the $3.8 \times 10 -3$ hard-decision forward error correction (HD-FEC) threshold is experimentally shown More than 2dB improvement in receiver sensitivity is demonstrated by the proposed PW-FNN equalizer when compared to Volttera and the conventional FNN equalizer. The outcomes of the experiment confirm that the PW-FNN suggested can be a viable option for a DML-based IM-DD system with a short reach.

In this research [16], an innovative classification system was proposed, as illustrated in Fig.. 1. In hearing devices, where a compact, reliable, and low-power system is required, this technology may be employed. By suggesting a voice activated detector (VAD) that uses adaptive wavelet technology and energy analysis of the sub-bands rather than Hurst exponents and the $\ell 2$-norm for wavelet sub-band selection, this study expands by using a more computationally efficient approach. Mel-frequency cepstrum coefficients (MFCCs), which are retrieved from the selected sub-bands for feature extraction, are merged with mean absolute values (MAVs) from all sub-bands to further account for the global variation across the sub-bands for training and testing.

III. Convolutional Neural Network

Compared to conventional artificial intelligence, they perform better when given inputs like voice, audio, or images. They consist of three main types of layers; Convolutional, Pooling and Fully connected. With every layer, the CNN gets more intricate and can identify a wider region of the image. As the visual data passes through the CNN layers, the larger features or shapes of the item are identified initially, and eventually the target object is identified.

A. Convolutional Layer

It is the most important component of at the system, which contains most of the computation. It needs input data, a filter, and a feature map, among other things. Its task is to search the image's receptive fields and identify any features that are present or absent. This procedure is called convolution [18].

The filter advances and retracts one step each time, continuing until the kernel traverses the complete image. Throughout training, several parameters—like the weight values—change as a result of the backpropagation and gradient descent processes. However, because they affect the output volume size, three hyper parameters need to be changed before the neural network is trained.

It includes:

i. The number of filters affects the depth of the emission. For example, three different feature maps with three different filters would have three depths.
ii. Stride refers to how far the kernel travels across the input matrix, or how many pixels it moves. A longer stride produces less.
iii. Zero padding when the filters do not suit the input image is typically employed. This sets all elements that are not part of the input matrix to zero, resulting in an output that is larger or equally sized.

B. Pooling Layer

Down sampling reduces the amount of parameters in the input and does dimensionality reduction; it is also frequently referred to as pooling layers. The entire input is filtered by the pooling operation, just like the convolutional layer does, except this filter is weightless [19]. The two main types of pooling that are:

i. Max pooling selects the maximum value inside each short window or region, reducing the spatial dimensions of characteristics.
ii. Average pooling finds the average value of all the values included in it, then substitutes this average value for the original data.

Fully connected layer as its name suggests, the full-connected layer is exactly that. Partially connected layers, as previously mentioned, the pixel values of the input image and the output layer are not directly connected. And node is connected one another. Fig. 3 shows the architecture of a sequential CNN. The trainable classifier has three fully connected layers, while the feature learning module has two convolutional layers and two pooling layers.

C. Fully Connected Layer

Fully connected layer as its name suggests, the full-connected layer is exactly that. Partially connected layers, as previously mentioned, the pixel values of the input image and the output layer are not directly connected. And node is connected one another. Fig. 3 shows the architecture of a sequential CNN. The trainable classifier has three fully connected layers, while the feature learning module has two convolutional layers and two pooling layers.

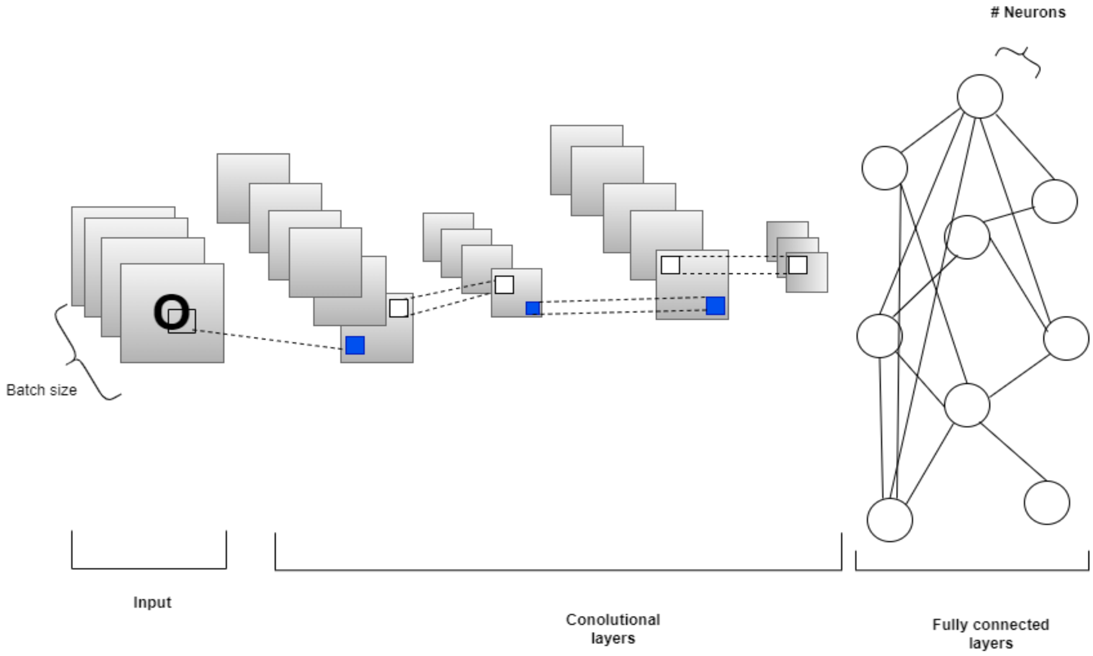

Fig. 3. Architecture of a sequential CNN

TABLE I
APPLICATION IN THE COMPUTER VISION SPACE

Apps	Year	References	Description
Computer vision	2022	[20]	In-depth discussions on whether MLP can represent a new paradigm for computer vision was provided in this review paper. It conducted a detailed comparison of the inherent similarities and differences among token-mixing MLP, convolution, and self-attention mechanism. The benefits and drawbacks of token-mixing MLP are discussed, and then recent MLP-like variations—from network architecture to module design—as well as their uses are carefully examined.
	2022	[21]	The suggested methodology combined randomizing pre-processing techniques and unsupervised instance-based resampling with MLP. Three experiments were carried out to verify the efficacy of MLP-progressive. These included verifying the methodology's validity over multi-and binary-class datasets, examining the influence of suggested pre-processing filters on the model's performance, and contrasting the MLP-progressive outcomes with those from innovative research.
Image analysis	2022	[22]	In this work, it was proposed and tested a malware classifier that can assign each piece of malware that is input into the appropriate family. It was accomplished by combining a malware visualization technique with the multi-layer perceptron algorithm. It means creating grayscale images out of a malware binary.
	2023	[23]	The multi-layer perceptron (MLP) neural network (NN) was employed in this work to classify stars. There are eighteen features that use MLP. The correlation between these features was demonstrated in this paper. The MLP model's accuracy in this paper was 97%. To determine which optimizer is the best, this study compared them. With the highest validation accuracy, the Adagrad optimizer is the best optimizer.

Object detection	2022	[24]	This work describes a novel application of CSA for training MLP architectures to address real-world issues like wheat classification, flower classification, active sonar target classification, and breast cancer diagnosis. The best weights and biases are determined using the CSA in order to greatly improve the MLP's classification accuracy. A performance comparison between the proposed method and popular training strategies, including genetic algorithms (GA), ant colony(ACO) and particle swarm optimization (PSO) was introduced.
	2023	[25]	This study presents a combination method in which the feature extraction stage is entirely independent of the classification stage. However, deep networks are used during the feature extraction phase. This used a deep feature-fed multi-layer perceptron (MLP) neural network.
	2023	[26]	In this work, the accuracy was improved without adding extra FLOPs by integrating the LIF neuron mechanism into the MLP models. To communicate between patches, a full-precision LIF operation that includes both vertical and horizontal LIF in various directions was suggested. Additionally, it suggested using group LIF to extract more accurate local features. Then the SNN-MLP model with LIF modules achieves state-of-the-art results.

IV. Recurrent Neural Network

A recurrent neural network consists of linked units (neurons) with non-linear interactions and at least one cycle in the network's structure.

Inputs are sent through the nodes at time t to get input activation then goes to the hidden neurons. The output y (t) at time t can be influenced by the input x (t_1) at time t_1 through the use of these recurrent connections. Then there is two matrix made of normal and recurrent weights between the input and hidden layers. Each node can learn an offset through to the biases represented by the vectors [27].

In this case [28], the model is not cyclic; instead, it can be viewed as a deep network with one layer for each time step and shared weights between time steps. It becomes evident that the unfolded network can be trained over a large number of time steps. This approach is called backpropagation through time (BPTT), which was first introduced in 1990.

The 1980s saw the beginning of the foundational research on recurrent networks. Hopfield unveiled a group of recurrent neural networks in 1982. It will not go into detail about these networks because they are not currently being used for sequence learning. Hopfield's nets can recognize some patterns, but they do not clearly provide a supervised training approach.

Recurrent neural networks have long been thought to be challenging for learning. Neural networks in general have NP-Hard optimization. However, the learning can be particularly challenging because of the challenges associated with learning long-range dependencies, as explained by some researcher in 1994 and further elaborated in [29]. The well-known issues of vanishing and exploding gradients arise when errors are propagated over numerous time steps. The vanishing gradient problem is more difficult when the activation function is a constant, but it is simpler to visualize the result with a rectified linear unit max (0, x). explosion of gradient, even in this simple case. A comprehensive mathematical analysis of the exploding and vanishing gradient issues, describing the precise circumstances in which these issues could arise. Considering these circumstances where the gradient could disappear or blow up. In [30], they recommend using a regularization term as a training tool, which forces the weights to values at which the gradient doesn't explode or disappear. One way to address this is with truncated backpropagation through time (TBPTT). issue for networks that are always in operation [30].This kind of illustration can be found in [31].

Examples of GPU-based forward and backward propagation implementations are Theano, which have simplified the process of implementing quick training algorithms. Before the LSTM was developed in 1996, experiments were conducted to train recurrent nets to bridge long time gaps, but the results showed that these methods performed no better than chance.

But nowadays, RNNs that have been trained successfully are common. Hessian-Free, or truncated Newton, neural network training was successfully reported by Sutskever and Martens and applied to a network that gains the ability to produce one character at a time through text [32] . This variation is specifically made to avoid critical points, such as saddle points, in contrast to Newton's method, which is drawn to them. A demonstration of enhanced performance on recurrent networks is one of the experimental findings. Newton's approach necessitates computing the Hessian.

In the first, LSTM [33], authors present a computational unit that takes the place of conventional artificial neurons in a network's hidden layer. These memory cells allow networks to get around some of the training problems that came with older recurrent nets. Bidirectional Recurrent Neural Networks, Schuster and Paliwal presented the BRNN architecture, which uses data from the past and future to predict the output at any given time t. This is different from earlier systems, which were effective only for sequence labelling tasks in natural language processing and only for past input to influence the output. In [34], the author described in this paper the use of Deep Recurrent Neural Networks (DRNNs) for user behavior prediction in Tor networks. In the lab, we built a Tor server and a Tor client, which is a Deep Web browser. The client then uses the Tor network to send the browsing data to the Tor server. After gathering the data with Wireshark Network Analyzer, we used DRNNs to predict the outcome. The simulation's output demonstrates how well the model predicts user behavior in Tor networks.

In [35], the use of recurrent neural networks to improve the precision of sound event detection in intelligent video surveillance systems was covered. Applying recurrent neural networks with controlled elements opens new avenues for sound recognition. In a fog computing environment, a novel approach is put forth for using neural networks to identify sound events.

Applications that have been developed for smartphones and PCs are displayed. The results of the experiment verify the suggested method's effectiveness.

The input sequence will be transformed into a single-dimensional vector (hidden vector) by the encoder. The output sequence will be created by the decoder from the hidden vector. With the input sequence, to maximize the conditional probability of the target sequence, encoder-decoder models are jointly trained.

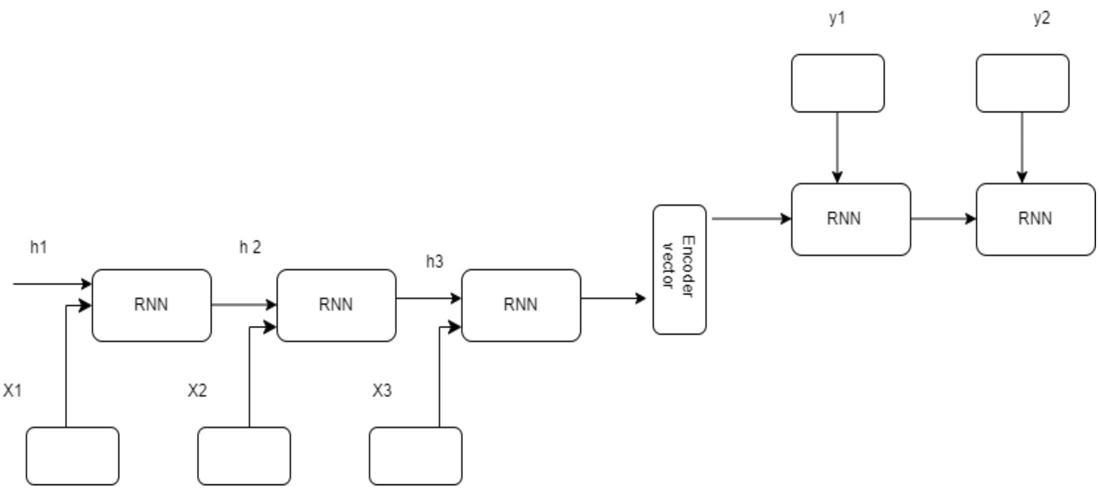

Fig.. 4. Recurrent Neural Network structure

The encoder can be formed by stacking multiple RNN cells together. RNN reads each input in turn. The hidden state or vector, h, is updated for each time step (each input) t based on the input received at that time step X[i] [36].

The final hidden state of the model represents the context once the encoder model has read all of the inputs and synopsis of the complete input sequence. For instance: Think of the input sequence that needs to be encoded: "I am a Student." For the Encoder model, there will be a total of 4 time steps (4 tokens). At each time step, the hidden state h will be updated using the input from the current time step and the previous hidden state [37].

The previous hidden state, h_0, will be chosen at random or treated as zero at the first time step, t_1. Thus, using the first input and h_0, the first RNN cell will update the hidden state. The output for each stage and the updated hidden state are the two outputs that each layer produces. It will just spread the concealed states to the following layer after the outputs at each stage are rejected.

The hidden state h_i and the second input X will be provided with the formula:

$$h_t = f\left(W^{(hh)} h_{t-1} + W^{(hx)} xt\right) \qquad (1)$$

As inputs at the second time step t_2, and the hidden state h_2 will be updated in accordance with both inputs. Subsequently, the new input will be incorporated into the hidden state h_1, which will generate the hidden state h_2. Regarding the given example, this occurs for all four stages.

A stack of multiple recurrent units, each of which takes one element from the input sequence, gathers data for that element, and propagates it forward (LSTM or GRU cells for better performance). All of the words from the question make up the input sequence in the question-answering problem. Every word is represented by the symbol x, i where i is the word's order.

V. Decoder

By forecasting the subsequent output Y_t given the hidden state h_t , the decoder creates the output sequence. The last hidden vector found at the end of the encoder model serves as the decoder's input [38]. Each layer will contain three inputs: the original hidden vector h, the hidden vector from the previous layer (h_{t-1}), and the output (y_{t-1}). y_1 and the updated hidden state h_1 (the information of the output will be subtracted from the hidden vector) are the outputs obtained at the first layer, which also receives the encoder's output vector and empty hidden state h_{t-1} as input.

The second layer generates the hidden vector h_2 and output $y_2$using the updated hidden state h_1, the prior output y_1, and the original hidden vector h_i as current inputs. The actual output is what happens at each time step of the decoder. Up until the END symbol appears, the model will predict the output [39].

A stack of multiple recurrent units, each of which forecasts an output at time step t, denoted as y_t. The output sequence in the question-answering problem is an assemblage of every word from the solution. Every word is represented by the symbol y_i, where i is the word's order. It is applied using:

$$h_t = f\left(W^{(hh)} h_{t-1}\right) \qquad (2)$$

The output layer is used to produce the probability distribution from a vector of values with the target class of high probability [40]. It uses the weight W(S) and the hidden state at the current time step to compute the outputs. Softmax is used to construct a probability vector that will help us determine the ultimate output—the word in the problem that requires answering questions, for example.:

$$y_t = softmax\left(W^{(S)} h_t\right) \qquad (3)$$

This type of architecture makes it possible to solve a whole new class of issues.

VI. Deep Reinforcement Learning Architecture

In DQN, Q value is estimated using DNN. The Bellman equation is satisfied using this network [41]. It takes its cues straight from the Q-learning methodology and offers several benefits over the Q network. For instance, DQN uses screen images to determine the best way to play a video game and obtain the highest score. By gathering data. Using this analogy, an algorithm is created. You will obtain reward (r) and the subsequent state $(s1)$ for any pair (s, a) that has the action (a) and the current state (s).

A. Double Deep Network

The originator of DDQN was "Hado Van Hasselt." By breaking down the max operation in the target to action selection, it helps to reduce the overestimation issue. In [42] , It is the result of combining DNN and DQN. This technique was developed to address the issue of Q values in previously discussed models being overestimated. Since the action with a higher Q value is known to have the best chance of leading to the next state, the accuracy of the Q value depends on the actions taken, the results obtained, and the next state of this trial. There isn't enough Q values at the start of the experiment to determine the optimal possibility.

Since there are currently fewer Q values to choose from, selecting the greatest Q value among the available options could mislead you in the direction of the goal. DDQN is utilized to get around this issue. In this case, we employ two DQNs: one selects the Q value, while the other uses the target network to compute the target Q value for selecting that particular action. By minimizing the overestimation of the Q values, this DDQN aids in cutting down on training time [43]

B. Dueling Q Network

Using two networks—the current network and the target network—the Dueling Q network is used to solve the DQN model's problems. Approximating the Q value is the current network. Conversely, the target network chooses the next optimal action and executes the target's selected action. Approximating the value of each action is not necessary in all situations [44]. A dueling Q network is used for this. When there's a collision in certain game environments, it can move left (or right), but in other situations, and must know which way to go. It has created a single Q network architecture known as a "dueling network. It used two sequences after the convolution layer as opposed to just one. Real-time DRL two sequence applications are used to divide the advantage function and estimation values, then combine them to create a single Q worth. Consequently, the Q function, which is trained using a variety of current algorithms, is the dueling network's output. SARSA and DDQN[45].

This paper [46], focuses on a deep learning based auto detection mechanism for flexible nanowires. To segment all moveable nanowires in AFM pictures, a fully convolutional network (FCN) and an instance segmentation network based on You Only Look Once version 3 (YOLOv3) are used. When coupled with subsequent image morphology and fitting algorithms, this permits high abstraction level nanowire posture and position detection.

Thanks to these techniques, the program has over 90% reliability in the testing dataset and can automatically detect nanowires with various morphologies at manometer precision. The detection results show the strong robustness of this approach and are less impacted by image complexity than the results of previous methods [47]. To maintain the battery state of charge (SoC) within the operable limits and minimize the long-term operational costs, it introduces three distinct Deep

Reinforcement Learning (DRL) algorithms in [48. The purpose of using three distinct DRLs is to highlight the advantages and disadvantages of the DQN, DDPG, and TD3 algorithms for handling management problems over longer time horizons, even when dealing with continuous states and action spaces. The suggested RL algorithms can be used to address this and other comparable management issues, according to experimental findings. These demonstrate how DRL algorithms have the potential to resolve even more complicated issues with uncertainty, but it is challenging to ensure that they will arrive at the best solution.

This paper [49] considers the flexible ramping capacity offered by wind power and suggests an intelligent dispatching method based on deep reinforcement learning for wind-thermal hybrid systems. First, a dispatching model is built that considers the variable ramping capacity offered by wind power. Subsequently, the task of dispatching is converted into one of reinforcement learning. An intelligent wind-thermal hybrid system dispatching method is proposed, based on the deep deterministic policy gradient (DDPG) approach, and takes into account the flexible ramping capacity offered by wind power.

To reduce operating costs for the micro grid, an intelligent deep reinforcement learning-based energy management strategy is examined in this paper. The micro grid considers thermostatically controlled loads to guarantee operational flexibility. The energy management problem is first formulated as a Markov decision process. The decision-making problem is then solved using the most advanced deep reinforcement learning technique, proximal policy optimization. In the meantime, system uncertainties are taken into account, such as wind power generation, electricity prices, and electricity loads. To demonstrate the efficacy of the suggested method, a comparison study with alternative approaches is conducted. Based on the findings of this review study, the following comparative tables were developed for summary purposes of the overall architecture strengths and weaknesses. Table II presents the Comparison analysis of various architectures

TABLE III
COMPARISON ANALYSIS OF VARIOUS ARCHITECTURES

Parameters	Feedforward neural network	Convolutional neural network	Recurrent neural network	Deep reinforcement learning architecture(DRL)	Auto encoder & generational encoder
Input type	Structured data	Images, video	Structure and Unstructured data	Environmental states and actions	Sequential data
Use	Classification and regression tasks	object detection.	time-series forecasting	decision-making under uncertainty	Anomaly detection, noise reduction
Advantages	Interesting in power systems especially for load forecasting. It usually demands less processing power and training time because of its simpler architecture.	Attractive when it comes to fault analysis especially if image or spatial data are provided	They are perfect for time-series forecasting, demand prediction, and dynamic system monitoring in power systems since they are specifically made to handle sequential data.	DRL can learn to operate in complex, uncertain, and dynamic environments, which are common in power systems, by interacting with the environment and receiving feedback.	Auto encoders can be used to identify abnormalities or unexpected patterns in power system operations, such as equipment failures or grid disruptions, by learning to rebuild inputs.
Limitations	For tasks like time-series analysis in power systems, where the sequence or order of data points matters, FNNs are not appropriate.	It is limited in its application in time-series prediction applications even if it is good at handling spatial data since they are not naturally made to analyse temporal sequences.	Because of problems like vanishing and bursting gradients, RNNs can be difficult to train, particularly for lengthy sequences.	To learn efficient policies, DRL usually needs a lot of data or simulated scenarios, which isn't always possible in real-world power systems.	May over fit to train data if they are not adequately regularized, particularly if the model is very sophisticated in comparison to the quantity of data that is available.

VII. Conclusion

This paper illustrates how wide the neural network architecture is, with the introduction of the different types of architectures which are feed-forward networks, Convolutional neural networks, recurrent neural networks, Auto encoder and generational encoders and Deep reinforcement learning architecture.

The study then presents in depth the different applications and characteristics of each architectures. The review will help the fellow researcher to have a distinct understanding of each architectures and based on the literature, understand how any architecture can be implement in future projects.

Conflict of Interest

The authors declare no conflict of interest in the publication process of the research article.

Author Contributions

T.L. Makosso conceptualized and designed the research. A. Almaktoof and K. Aboalez reviewed and edited the work.

References

[1] P. Yunfei, “Gender Voice Imbalance Classification Comparisons and Analysis,” in 2nd International Conference on Machine Learning and Computer application, Shenyang, China, 2020.

[2] Yu.Xinke, Z.Gan and T. Zhao, “Deep Feedforward Sequential Memory Networks Based Mispronunciation Detection for Tibetan Students' Mandarin,” in 2021 2nd International Conference on Information Science and Technology, Chongqing,China, 2021.

[3] W. L, L. G, M. Z and z. X, “Artificial neural network approaches on composition–property relationships of jet fuels based on GC–MS,” Sciendirect, vol. 86, no. 16, p. 20, November 2007.

[4] J.Then, H. Lim. and T. Phang, “Convolutional Deep Feedforward Network for Image Classification,” in 2019 7th International Conferenceon Smart Computing and Communications, Sarawak, Malaysia, 2019.

[5] Q. Wu, Q. Chen and M. Tan, “V2C: Visual Voice Cloning,” in 2022/CVF IEEE International Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 2022.

[6] M.Zamani, S. Abdullah and A. Demosthenous, “A Discrete Wavelet Transform-Based Voice Activity Detection and Noise Classification with Sub-Band Selection,” in 2021 IEEE International Symposium on Circuits and Systems, Daegu, Korea, 2021.

[7] D. Shi, R. Liu and L. Tao , “Heuristic Dropout: An Efficient Regularization Method for Medical Image Segmentation Models,” in 2022 International Conference on Acoustics, Speech and Signal Processing, Singapore, Singapore, 2022.

[8] J. Hussain, M. Uppal and W. Lee, “Enhancing accuracy in brain stroke detection: Multi-layer perceptron with Adadelta, RMSProp and AdaMax optimizers,” 2022.

[9] B. I, O. A and B. M, “Image-Based Malware Classification Using Multi-layer Perceptron,” 2021.

[10] E. Yusesoy, Gender Recognition Based on the Stacking of Different Acoustic Features, Applied Sciences MDPI, 2024

[11] A. Abdel and M. Soliman, “A Multi-Layer Perceptron (MLP) Neural Networks for Stellar Classification: A Review of Methods and Results,” International Journal of Advances in Applied Computational Intelligence (IJAACI), vol. 3, no. 2, October 2023.

[12] A. Bataineh, D. Kaur and S. Mohamed, “Multi-Layer Perceptron Training Optimization Using Nature Inspired Computing,” in IEEE Access, 2022.

[13] F. M and F. S, “Developing a Tuned Three-Layer Perceptron Fed with Trained Deep Convolutional Neural Networks for Cervical Cancer Diagnosis,” in National Center for Biotechnology, 2023.

[14] S. Mihalache, D. Burileanu, Using Voice Activity Detection and Deep Neural Networks with Hybrid Speech Feature Extraction for Deceptive Speech Detection, Computation MDPI, 2022

[15] W. Li, G. Liu and Z. Zhang, “Brain inspired multiple layer perceptron with spiking neurons,” 2023.

[16] S. J., “Artificial intelligence based power fault detection and power restoration,” in 6th International Conference on Electronics,Communication and Aerospace technology, 2022.

[17] F. Xu, “An intermittent high impedance fault identification method based on transient method power direction detection and intermittency detection,” in IEEE Power and Energy Society general meeting, 2018.

[18] K.A. Saleh, A. Hooshyar and F. E. and Sadaany, “Ultra high speed travelling wave based protection scheme for medium voltage DC microgrids,” in IEEE Transaction on Smart Grids, 2019.

[19] M. Tomar and A Souratishtriya, “A comparative study for fault location detection in bipolar HVDC transmissions system using wavelet transform and artificial neural network,” in 2021 3rd International on Signal Processing and Communication ICPSC, 2021.

[20] Y. Wang, X. Liu and S. Gao, “Fault travelling wave detection method based on VMD -MPE,” in 2021 International Conference on Power System Technology: Carbon neutrality and new type of power system, 2021.

[21] L. Baroli, O. Tetsuya and R. Obukata , “Application of Deep Recurrent Neural Networks for Prediction of User Behavior in Tor Networks,” in 2017 31st International Conference on Advanced Information Networking and Applications Workshops, Taipei,Taiwan, 2017.

[22] B. A. Bansal, “Robust technique for environmental sound classification using convolutional recurrent neural network,” December 2023.

[23] A. Ui Haq, A. F. Sindi and S. Gul “Modeling and fault categorization in thin-film and crystalline PV arrays through multilayer neural,” IEEE Access, vol. 8, pp. 102235-102255, 2020.

[24] H. Homenih, S. N. Sadoodi and F. Garibeh, “Fault diagnosis in photovoltaic arrays using GBSSL method and proposing a fault correction,” in IEEE Transactions on Energy, 2020.

[25] M. Shafulillah, M. Abido and M. Abdel, “Distribution Grids Fault Location employing ST based optimized machine learning approach,” Energies, 11 September 2018.

[26] B. M. Karmakar and A.K. Pradhan, “Detection and classification of faults in solar PV arrays using Thevenin equivalent resistance,” IEEE Journal, vol. 3, 20 March 2020.

[27] M. Haji, “Multivariate feature extraction based on supervised machine learning for fault detection and diagnosis,” Europe, vol. 4, pp. 313-321, 2021.

[28] M. Shafulah, A. M. Abido and Al, Hamouz “Wavelet based extreme learning machine for distribution grid fault location.,” IET Generation,Transmission and Distribution, vol. 11, 2017.

[29] M. Tovar, F. Rashid and M. Robles, “PV power production using CNN-LSTM hybrid neural network model Case of study : Temxico Morelos,” vol. 13, no. 24, 2020.

[30] S. Wu and H. Bai, “Deep-learning-based nanowire detection in AFM images for automated nanomanipulation,” Journal of Applied Sciences, vol. 4, no. 1, p. 15, 2021.

[31] N. Albalawi, G. Shahata and K. Aljahatoni, “Augmented Reality-Assisted Deep Reinforcement Learning-Based Model towards Industrial Training and Maintenance for NanoDrop Spectrophotometer,” MDPI, vol. 23, no. 13, p. 21, 2023.

[32] Y. Gyuangua and A. Karhaman, “Home Energy Management System based on Deep Reinforcement Learning Algorithms,” in 2022 IEEE IEPS Innovative Smart Grid Technologies Conference Europe, Novi Sad, Serbia, 2022.

[33] J. Duan, M. Liu and Q. Zhang, “Intelligent Dispatch of Wind-Thermal Hybrid System Based on Deep Reinforcement Learning Considering Flexible Ramping Capacity Provided by Wind Power,” in 5th International Conference on Power and Energy Applications, Guangzhou, China, 2022.

[34] S.M. Chopdar and K. A. Koshdi, “Fault Detection and Classification in Power System Using Artificial Neural Network,” in 2nd International Conference on Intelligent Technologies, 2022.

[35] A. F., “A novel convutional neural network based approach for fault classification in photovoltaic arrays,” in IEEE Access, 2020.

[36] K. Dubey and J. P. Jena, “Impedance Angle-Based Differential Protection Scheme for Microgrid Feeders,” IEEE System Journal, pp. 3291-3300, 2020.

[37] R. Chen, “Computational Fault time difference based fault location methods for branched Power Distribution Networks,” in IEEE, IEEE Access, 2019. [

[38] A. Minic, L. Jovanovic, C. Stoaen, M. Zivkovic, P. Spalevic, A. Petrovik and R. Stoaen, Applying Recurrent Neural Networks for Anomaly Detection in Electrocardiogram Sensor Data, Sensors MDPI, 2023
.

[39] M.M, Islam; M.Z, Islam; A. Asraf.; M.S, Al-Rakhami; W. Ding; A.H, Sodhro. Diagnosis of COVID-19 from X-rays using combined CNN-RNN architecture with transfer learning. MDPI 2022, 2, 100088

[40] Hassaballah, M.; Wazery, Y.M.; Ibrahim, I.E.; Farag, A. Ecg heartbeat classification using machine learning and metaheuristic optimization for smart healthcare systems. MDPI2023, 10, 429.

[41] Y. Jin; Z. Li; C. Qin; J. Liu; Y Liu; L. Zhao; Liu, novel attentional deep neural network-based assessment method for ECG quality. Biomed. Signal Process. Control 2023, 79, 104064

[42] H. Hamze, H. Lesani and A. Fereidunian, “Fault detection and Location for Reinforcement of Smart distribution system restoration, using discrete orthogonal stockwell transform and regression ANN,” in 2022 12th Smart grid Conference (SGC), 2022.

[43] H. Liang, H. Li and G. wang, “A single phase to ground fault detection method based on the ratio fluctuation coefficicent of the zero sequence current and voltage differential in a differential network,” in IEEE Access, 2023.

[44] K. Havriliuk, “Faults detecting of turnout using DWT and ANN,” in Proceedings of the 20th IEEE International Conference on Modern Electrical and Energy Systems (MEES), 2021.

[45] G. Song, “Pilot protection of hybrid MMC DC grid based on active detection,” Protetction and control of modern power systems, vol. 5, no. 1, 2020.

[46] Z. Chen, J. J. Chavez and Z. Yang, “A Control Method for Converter-interfaced Sources to Improve Operation of Directional Protection Elements,” in 2023 IEEE Power and Energy Society General Meeting, Orlando, FL, USA, 2023.

[47] M.M. Taye, Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions, Computation MDPI, 2023

[48] S. Boda; M. Mahadevappa.; P.K, Dutta,. An automated patient-specific ECG beat classification using LSTM-based recurrent neural networks. MDPI, 2023.

[49] Y. Zhang; Z. Zhao; Y, Deng; X, Zhang; Zhang, Y. ECGID: A human identification method based on adaptive particle swarm optimization and the bidirectional LSTM model. MDPI. 2021, 22, 1641–1654.