Review

Critical Review of Neural Network Generations and Models Design

Jabar H. Yousif * and Mohammed J. Yousif

Posted Date: 9 November 2023

doi: 10.20944/preprints202309.1149.v2

Keywords: neural networks; machine learning; convolutional neural networks; computational complexity; ANN performance

Review

Critical Review of Neural Network Generations and Models Design

Jabar H. Yousif [1,*] and Mohammed J. Yousif [2]

[1] Faculty of Computing and Information Technology, Sohar University, PO Box 44, Sohar, PCI 311, Oman
[2] Faculty of Science, Computer Science Department, Memorial University of Newfoundland, Canada
* Correspondence: jyousif@su.edu.om

Abstract: In recent years, Neural networks are increasingly deployed in various fields to learn complex patterns and make accurate predictions. However, designing an effective neural network model is a challenging task that requires careful consideration of various factors, including architecture, optimization method, and regularization technique. This paper aims to comprehensively overview the state-of-the-art artificial neural network (ANN) generation and highlight key challenges and opportunities in machine learning applications. It provides a critical analysis of current neural network model design methodologies, focusing on the strengths and weaknesses of different approaches. Also, it explores the use of different learning approaches, including convolutional neural networks (CNN), deep neural networks (DNN), and recurrent neural networks (RNN) in image recognition, natural language processing, and time series analysis. Besides, it discusses the benefits of choosing the ideal values for the different components of ANN, such as the number of Input/output layers, hidden layers number, activation function type, epochs number, and model type selection, which help improve the model performance and generalization. Furthermore, it identifies some common pitfalls and limitations of existing design methodologies, such as overfitting, lack of interpretability, and computational complexity. Finally, it proposes some directions for future research, such as developing more efficient and interpretable neural network architectures, improving the scalability of training algorithms, and exploring the potential of new paradigms, such as Spiking Neural Networks, quantum neural networks, and neuromorphic computing.

Keywords: neural networks; machine learning; convolutional neural networks; computational complexity; ANN performance

1. Introduction

Machine learning (ML) is a branch of artificial intelligence that explores computer algorithms capable of automatically improving with experience [1]. It involves building models using training data to make predictions or decisions without explicit programming. ML is utilized in various fields, such as computer vision, speech recognition, biotech, risk management, and cyber security, where traditional algorithms could be more complex and practical to develop. While machine learning is similar to computational statistics, it is not exclusively statistical learning, as mathematical optimization provides methods, theory, and application areas for ML. Data mining emphasizes exploratory data analysis through unsupervised learning [2, 3]. Artificial neural network (ANN) is one of the most influential and popular machine learning techniques that simulate the human brain's behavior and structure. Neural networks comprise single/multi-layers of interconnected nodes to process data and transmit generalized information for predicting future conditions or making decisions [4, 5]. ANN has shown remarkable success in various applications, including image enhancement, speech recognition, natural language processing (NLP), robotics, and finance. ANNs learn from raw data by adjusting the weights of node connections of complex datasets, resulting in more appropriate figures, which allows them to extract complex patterns and relationships. The field of ANNs has seen several generations of development, each characterized by a distinct set of architectures and algorithms that enable them to perform increasingly complex tasks [6, 7]. The first-generation ANNs, which emerged in the 1950s and 1960s, were relatively simple models with only a few layers of neurons, mainly used for pattern classification and regression tasks [8, 9]. However, their limited capacity to learn nonlinear relationships and extract meaningful features from complex data made them unsuitable for many real-world problems. In the 1980s and 1990s, second-generation ANNs, such as the multilayer perceptron (MLP) and the backpropagation algorithm, emerged as a more robust approach to machine learning [10, 11]. These models used deep architectures with multilayer neurons and the ability to learn linear/nonlinear functions, enabling them to extract more complex patterns and relationships from data. This made them more suitable for many applications [12]. The third generation of ANNs, also known as deep learning, emerged in the 2000s. It is characterized by multilayer deep neural networks, which can learn complex patterns from large datasets [13, 14]. This was made possible by the availability of large amounts of labeled data and advances in computing power, which made it feasible to train models with hundreds of layers and billions of parameters. Deep learning applies to many applications, like computer vision, text generation, natural language processing, and speech recognition [15,16,17,18].

Fourth-generation neural networks are a new development focusing on learning from fewer data and being more interpretable. Research areas in this field include neuro-symbolic AI, few-shot learning, and explainable AI [19, 20, 21, 22]. It combines the concepts of ANN with symbolic reasoning techniques to achieve more interpretable results. Few-shot learning aims to develop neural network models that can learn from a small amount of data. Explainable AI focuses on developing neural network models that can explain their predictions. The Structured Probabilistic Model (SPM) and TreeNet are two models used for data processing and modeling. SPM can process structured and unstructured data and is suitable for applications, such as computer vision [23], NLP, and analysis of time series [24]. On the other hand, TreeNet is a robust model used for structured data in regression [25], classification, anomaly detection, and time series analysis applications.

The fifth generation of artificial neural networks (ANNs) can learn from structured and unstructured data. It can incorporate domain knowledge and prior beliefs, resulting in high-quality outputs. However, these models require even more computational resources than fourth-generation models and can be even more challenging to interpret. Spiking Neural Networks (SNNs) are typically considered fifth-generation neural networks, as they can model temporal dynamics and spiking behavior, allowing them to capture the complexity of biological neural networks more accurately [26, 27]. Despite the impressive performance of ANNs in many applications, many challenges remain to be addressed, such as their interpretability, robustness, and scalability. Deep neural networks are complex and challenging to be understood by users to make the right decisions. Even slight changes to the input data can result in misclassification or incorrect outputs. This makes it limited to be implemented in some domains [28]. They require extensive data and computing resources, limiting their scalability and applicability in specific settings [29]. There are still various challenges and limitations that these technologies currently need to overcome. These challenges include:

- Bias and Fairness: A significant challenge with artificial intelligence and machine learning is ensuring that the algorithms are fair and unbiased. Since machine learning algorithms rely on data, they can inherit biases and prejudices from the data, resulting in unfair or discriminatory outcomes. Ensuring fairness and avoiding bias in machine learning is a complex and ongoing research area.
- Interpretability: Another issue with machine learning is the need for more interpretability or transparency of the algorithms. Many machine learning models are complex and nonlinear, making understanding how they arrive at their predictions or decisions challenging. This lack of interpretability can be problematic in domains such as healthcare and finance, where interpretability and explainability are crucial.
- Data Quality and Quantity: Machine learning algorithms depend heavily on the quality and quantity of data used to train them. Only complete, accurate, or biased data can positively affect the performance and reliability of the algorithms. Furthermore, in some domains, such as

healthcare, acquiring large amounts of high-quality data can be a time-consuming and challenging task.

- Privacy and Security: Machine learning often involves using significant amounts of personal and sensitive data, raising concerns about privacy and security. Ensuring the confidentiality and integrity of data remains an ongoing challenge for machine learning practitioners.
- Computational Resources: Some machine learning algorithms can be computationally expensive and require significant computational resources, such as powerful GPUs or specialized hardware. This can make it difficult to apply machine learning in resource-constrained environments.

Addressing these challenges and limitations of artificial intelligence and machine learning is a future research and development directions. Researchers are exploring new directions in ANN research to address these challenges, such as incorporating biological inspiration, developing more efficient and energy-saving architectures, and combining ANNs with other AI techniques like evolutionary computation [30, 31]. They are also exploring ways to improve the interpretability and robustness of ANNs, such as developing methods for explaining their decisions and designing models less susceptible to adversarial attacks.

Overall, ANN research is an exciting and rapidly-evolving field that generates new insights and solutions to some of society's most pressing challenges. This article offers an overview of the progression of ANN generations and their impact on computing data and improving the simulation of historical data. It also highlights the potential for these advancements to paving the way for developing the next generation of neural networks.

2. Neural network Components

Artificial neural network (ANN) is one of the most influential and popular machine learning techniques that simulate the human brain's behavior and structure. Neural networks comprise three layers of interconnected nodes: Input, Hidden, and Output. Each neuron has a weight, a numerical value multiplied by the inputs as shown in Figure 1 [32, 33].

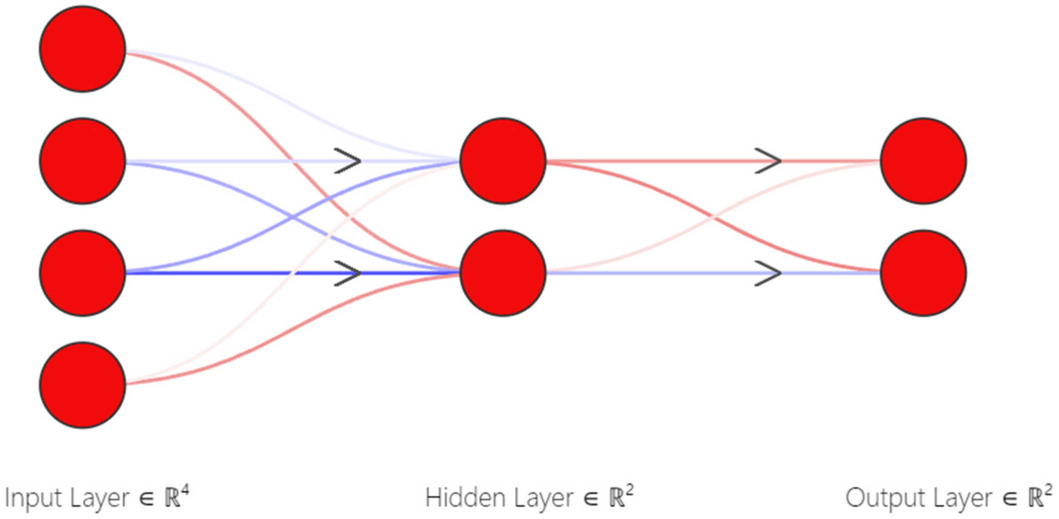


Figure 1. neural network architecture.

The neural network consists of the following components:

- The input layers represent the input space vector, which processes data and transmits generalized information.
- The hidden layer is between the input and output nodes with flexible boundaries.
- The output layers are the final nodes of the neural network that consist of resulting information for predicting future conditions or making decisions.

doi:10.20944/preprints202309.1149.v2

- Weights are computed values based on a machine learning algorithm that adjusts their values depending on the difference between predicted results and the training datasets.
- An activation function that allows a neuron to fire (turn on/off) using a mathematical formula. The node with a value greater or equal to the value of the activation function will fire (give output). Otherwise, it will not fire.

3. Machine-Learning Approaches

There are three broad categories of machine learning approaches based on the type of "signal" or "feedback" available to the system [34, 35, 36]. Here are some of the main categories:

1. Supervised Learning: It trains a model on labeled data that each is associated with known targets. The aim is to understand a function that maps unseen dataset inputs to a generalized outputs and predicting new outputs. Common supervised learning algorithms include linear regression (LR), logistic regression, decision trees (DTs), random forests (RF), and neural networks (NN).
2. Unsupervised Learning: It finds patterns/clusters in unlabeled data without known targets using algorithms like k-means clustering, hierarchical clustering, PCA, and autoencoders.
3. Semi-Supervised Learning: It is a hybrid approach that combines labeled and unlabeled data to improve accuracy, such as self-training, co-training, and multi-view learning. It uses labeled data to guide learning and unlabeled data to extract generalized features.
4. Reinforcement Learning: It involves training a model through the interaction with its environment and rewarding or penalizing a model based on feedback. The objective is to develop a policy that leads to positive outcomes over time. Q-learning, policy gradient methods, and deep reinforcement learning are examples of algorithms used.
5. Transfer Learning:
   It is used to leverage knowledge or experience from one task to improve another. It involves transferring relevant features from the source to the target domain and refining the model. Pre-trained neural networks and other algorithms can be used for transfer learning.

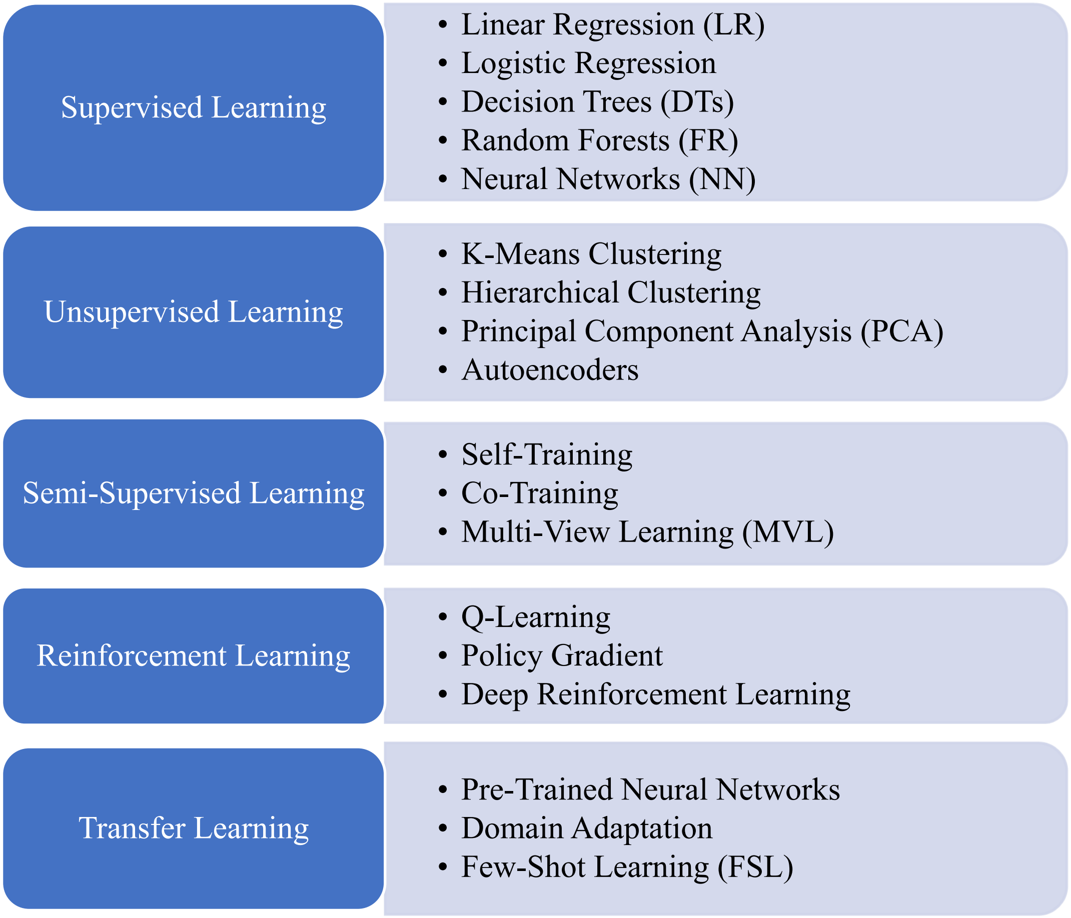
Figure 2 shows the different machine learning approaches, their subtypes, and some typical applications.

These machine learning approaches can be further categorized into other subtypes, and the choice of approach depends on different factors, including problem nature, amount of data, and actual outcomes [37, 38].

However, neural networks can enhance transfer learning and leverage experience gained in several ways:

1. Pre-trained Models: Neural networks can be pre-trained on large, diverse datasets to learn generic features or representations that can be transferred to other tasks or domains. For example, a neural network trained on millions of images can learn to recognize basic visual features such as edges, corners, and textures, which can then be used as the basis for other image recognition tasks. By leveraging the pre-trained features, the model can learn from smaller or more specialized datasets and achieve better performance with fewer data.
2. Fine-Tuning: Neural networks can be fine-tuned on new or specific data to adapt to the target task or domain. This involves starting with a pre-trained model and updating the weights or parameters using the target data while keeping the learned features fixed or frozen. Fine-tuning can improve the model's performance on the target task and help it learn more quickly and effectively from limited data.
3. Multi-Task Learning: Neural networks can simultaneously be trained on multiple related tasks to share knowledge and improve generalization. This involves designing a single model with multiple outputs or loss functions, each corresponding to a different task or objective. By jointly optimizing the model for all tasks, the network can learn more robust and transferable features and benefit from the experience gained across tasks.

4. Domain Adaptation: Neural networks can be adapted to new domains or environments by adjusting their parameters or structure based on the differences between the source and target data. This involves identifying the relevant domain-specific features or biases and modifying the model to account for them. Domain adaptation can improve the model's performance in scenarios where the training and testing data may differ in some way, such as in medical imaging or natural language processing.



Supervised Learning	• Linear Regression (LR) • Logistic Regression • Decision Trees (DTs) • Random Forests (FR) • Neural Networks (NN)
Unsupervised Learning	• K-Means Clustering • Hierarchical Clustering • Principal Component Analysis (PCA) • Autoencoders
Semi-Supervised Learning	• Self-Training • Co-Training • Multi-View Learning (MVL)
Reinforcement Learning	• Q-Learning • Policy Gradient • Deep Reinforcement Learning
Transfer Learning	• Pre-Trained Neural Networks • Domain Adaptation • Few-Shot Learning (FSL)

Figure 2. Machine learning approaches.

Neural networks can enhance transfer learning and leverage experience gained by learning generic features, fine-tuning new data, multi-task learning, and domain adaptation [39]. These techniques can improve neural network performance, efficiency, and generalization in various domains and applications.

4. Neural Network Design

Neural Network design refers to the different approaches and strategies to create effective neural network models for solving specific problems. In each of these cases, the neural network model design methodology involves selecting appropriate architectures, hyperparameters, and optimization algorithms to achieve optimal performance [40, 41]. Critical analysis points could be considered when evaluating current neural network model design. A comprehensive analysis would need to consider the specific context and objectives of the research. Some possible critical analysis points that could be considered when analyzing current neural network model design methodologies:

1. Architecture selection: Different neural network architectures have been developed for different applications. A critical analysis should assess the suitability of each architecture for a given

problem, considering factors such as dataset size, task complexity, and the availability of computational resources.

2. Hyperparameter tuning: Neural networks have many hyperparameters that must be tuned to achieve optimal performance. A critical analysis should examine the model's performance based on using different hyperparameters, such as the learning rate value, size of batching, activation function, and the number of layers.
3. Regularization techniques: Overfit processing is a mutual problem in neural network training. Regularization techniques such as dropout and L1/L2 regularization can help to reduce overfitting. A critical analysis should evaluate the efficiency of such techniques to improve model performance and generalization ability.
4. Optimization methods: Gradient descent and its variants commonly optimize neural network models. A critical analysis should assess the limitations of these methods, such as convergence issues and sensitivity to initialization, and explore alternative optimization techniques.
5. Interpretability: One major hurdle for neural network models is their need for interpretation. A critical analysis should examine the current state-of-the-art interpretability techniques, such as feature visualization and attribution methods, and assess their effectiveness in providing insights into the model's decision-making process.
6. Computational complexity: Training large neural networks can be computationally expensive and time-consuming. A critical analysis should consider the scalability of different neural network architectures and optimization methods and explore ways to reduce the computational complexity of training.
7. Robustness: Neural network models are susceptible to adversarial attacks, where small input perturbations can cause significant output changes. A critical analysis should examine the robustness of different neural network architectures and explore ways to improve their resilience to such attacks.

5. Deep Learning Architecture

Deep learning is a class of algorithms and topologies that applies to a wide range of problems. Deep learning involves training neural networks utilizing data samples and rewarding those who perform well [42]. Several architectures and algorithms used in deep learning are wide and varied. We will review the most interesting architectures, which include convolutional neural networks (CNN) [43, 44], recurrent neural networks (RNN) [45, 46], Autoencoders [47, 48], Generative Adversarial Networks (GAN) [49, 50], long short-term memory (LSTM) [51, 52], and A self-organizing map (SOM) [53, 54].

1. Convolutional Neural Networks (CNN): These are neural networks that use convolutional layers to extract relevant features from images or other types of data.
2. Recurrent Neural Networks (RNN): These are neural networks that use recurrent connections to process sequences of data, such as time series or natural language [55, 56].
3. Long short-term memory (LSTM): Long short-term memory (LSTM) networks can learn long-term dependencies with sequence prediction problems. It uses memory cells to control information flow and solve the issue of vanishing gradients in RNNs.
4. Autoencoders: These are neural networks that use unsupervised learning to learn a compressed representation of input data.
5. Generative Adversarial Networks (GANs): These neural networks use a generator and discriminator to generate realistic data from random noise.
6. Self-organizing map (SOM): A self-organizing map (SOM) is unsupervised learning aiming to reduce a high dimensionality of input space to construct a low-dimensional output space. It applies competitive learning between adjacent neurons using neighborhood function to maintain the input space topological properties.

Figure 3 shows the main deep learning architectures.

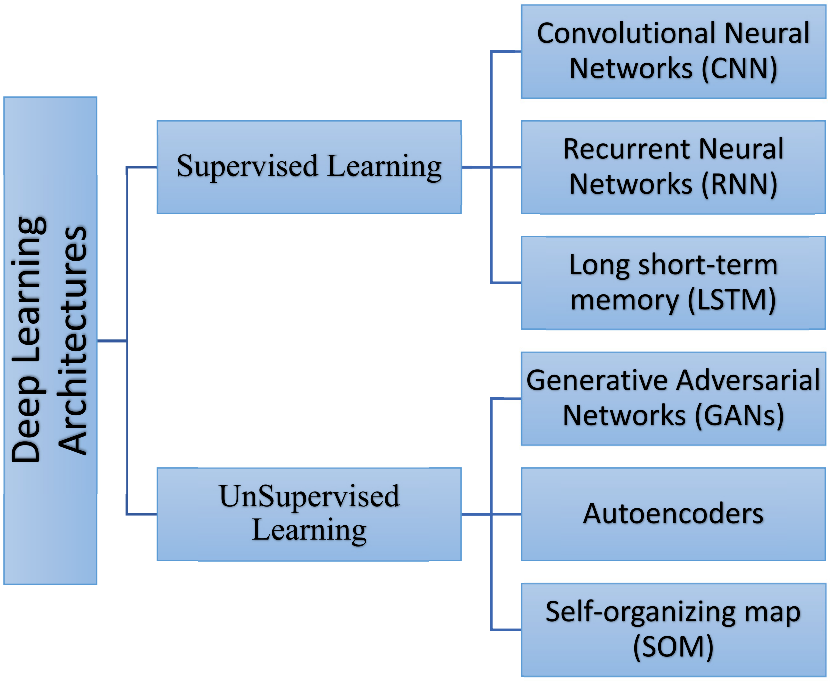

Deep Learning Architectures
Supervised Learning
UnSupervised Learning
Convolutional Neural Networks (CNN)
Recurrent Neural Networks (RNN)
Long short-term memory (LSTM)
Generative Adversarial Networks (GANs)
Autoencoders
Self-organizing map (SOM)

Figure 3. the main deep learning architectures.

These methodologies can be combined and adapted to create customized hybrid neural network models for specific applications as presented in Table 1.

Table 1. Recurrent Neural Network Design Methodologies summary.

Methodology	**Applications**	**Advantages**	**Disadvantages**
Convolutional Neural Networks (CNN)	Image and video recognition, Image Classification, Image Segmentation, object detection	High accuracy, can handle large inputs, Feature Learning, Spatial Hierarchies	Can require large amounts of data, CNNs can be computationally intensive, requiring powerful hardware.
Recurrent Neural Networks (RNN)	Speech recognition, language modeling, time-series analysis and prediction, Image Captioning	Can handle sequential inputs, flexible architecture	Can suffer from vanishing gradients, Short-term Memory, Training Difficulty
Long short-term memory (LSTM)	Natural Language Processing (NLP) tasks such as language modeling, text generation, and sentiment analysis.	Long-term Dependencies. Avoiding, Vanishing/Exploding Gradients, Feature Extraction	Computational Complexity, Overfitting, Interpretability
Autoencoders	Data Compression, Image Generation, Dimensionality reduction, anomaly detection, Denoising	Unsupervised learning, can handle noisy inputs, Dimensionality Reduction, Data Representation	Can be sensitive to initialization, Lack of Interpretability, Limited to Data Distribution, Need Hyperparameter Tuning
Generative Adversarial Networks (GANs)	Image and video synthesis, Image Generation, data augmentation, Super-Resolution	Can generate new data, Realistic Generation, unsupervised learning, Flexibility,	Can suffer from mode collapse, Training Instability, Evaluation Challenges, Computationally Intensive

Self-organizing map (SOM)	Image Processing, Data Clustering, Speech and Audio Processing	Unsupervised Learning, Flexibility, Non-linearity, Dimension Reduction, Topological Ordering	Training Sensitivity, Limited to Low Dimensions, Interpretability, finding good initializations can be challenging

6. Training Data

It refers to a set of data extracted to train the ANN machine learning model. It is essential to train the ANN model on the input patterns and get the relationships and associations that help make predictions or decisions with new, unseen data. A training dataset has no minimum size but depends on the task's complexity and desired output [57, 58]. However, we usually start with a reasonable dataset size and monitor model performance. This process can be repeated with different data sizes until getting high accuracy.

- **Impacts of Training Data**

Training data plays a critical role in determining the efficiency of a neural network. High-quality training data is crucial in examining the accuracy and effectiveness of neural network models. A large and diverse training dataset can lead to more accurate neural network models as it helps cover a broader range of input possibilities. Conversely, sufficient or low-quality training data can result in better performance and accurate predictions [59, 60]. It is also essential that the training data is representative of the problem domain and the inputs that the neural network is expected to handle. The training data should reflect the real-world scenarios the neural network encounters. Otherwise, the neural network will not be efficient in simulating and generalizing unseen data.

- **The quality and quantity of training data**

The training data should be preprocessed to eliminate inconsistencies or biases. This includes techniques such as data cleaning, feature scaling, and normalization. Preprocessing the data this way can lead to more efficient neural networks by ensuring that the input data is consistent and that the neural network can better recognize relevant patterns and relationships. In summary, it is important to have a diverse, representative, and preprocessed dataset to ensure optimal performance [60].

Table 2 shows that the quality and quantity of training data can be determined through various methods. In order to evaluate the quality of training data, we need to use normalization and feature scaling for the training data. Data augmentation techniques, including rotation and translation of images. Domain experts can assess the relevance and accuracy of the data and domain knowledge about the problem to be solved. On the other hand, the quantity of training data can be assessed through cross-validation techniques to evaluate the model's performance on new data and learning curves to determine the rate of improvement with additional data. In addition, evaluate model complexity analysis to determine the optimal model size given a fixed amount of data and resource constraints such as available memory and computational power [61, 62, 63].

Table 2. The quality and quantity of training data methods.

Factor	Impact on Efficiency	Impact on Accuracy	Methods to Determine
Quality of Training Data	Higher quality data can lead to faster convergence during training process.	Higher quality data can lead to better generalization to unseen data.	Data preprocessing, data augmentation, expert evaluation, domain knowledge
Quantity of Training Data	More data can improve the accuracy of the model to generalize to unseen data.	More data can increase the computational resources required for training.	Cross-validation, learning curves, model complexity analysis, resource constraints

7. Neural Network Generations Development

Several generations of neural network developed that include the following.

doi:10.20944/preprints202309.1149.v2

First-Generation Neural Networks developed in the period between the 1950s and 1960s. The network architecture is simple, comprises only a few layers, and is primarily used in pattern and character recognition applications. One example is Perceptron, which consists of a single-layer feedforward neural network that is mainly used for binary classification tasks. Another example is the Adaline neural network, which consists of a single layer of linear neurons and a continuous activation function [64, 65].

Second-generation neural networks were developed in the period between 1980s and 1990s. The neural network architecture is more complicated than first-generation networks and consists of multiple layers of neurons. They are utilized in several applications, such as speech recognition and image processing. One example is Multi-Layer Perceptron (MLP), which is a typical neural network model consisting of multiple perceptron layers and nonlinear activation functions. Another example is the radial Basis Function Network (RBFN), which uses radial basis functions to model complex patterns [66, 67, 68].

Third-generation neural networks appeared in the 2000s, known as deep learning. Using various activation functions, deep neural network architecture with multilayers can analyze complex patterns from large datasets. Deep learning has been applied in different applications, such as natural language processing, image processing, computer vision, and speech recognition [69, 70]. The typical type is a Recurrent Neural Network (RNN) that models sequential features, such as natural language text classification or speech recognition [71, 72]. Another type is Convolutional Neural Network (CNN), which is well-suited for image and video processing tasks [73, 74].

Fourth-generation neural networks have recently developed that are characterized by their ability to learn faster because they use less data in the training process and are easier to understand. The current research directions of fourth-generation neural networks include neuro-symbolic AI, few-shot learning, and explainable AI. Neuro-symbolic AI combines neural networks and symbolic reasoning approaches to produce the results and make them easy to interpret. Furthermore, Few-Shot neural network models can learn using a small amount of data that mimics humans' abilities to learn new concepts from experience. Also, Explainable AI neural network models help develop more interpretable prototypes, making them more understandable to human users [75, 76].

The fifth generation of artificial neural networks (ANNs) can learn from structured and unstructured data. It can incorporate domain knowledge and prior beliefs, resulting in high-quality outputs. However, these models require even more computational resources than fourth-generation models and can be even more challenging to interpret. Spiking Neural Networks (SNNs) are typically considered fifth-generation neural networks, as they can model temporal dynamics and spiking behavior, allowing them to capture the complexity of biological neural networks more accurately [77, 78]. Table 3 presents several generations of neural networks.

Table 3. several generations of neural network.

Generation	Advantages	Disadvantages
First generation	- Ability to learn from labeled data	- Limited ability to generalize to new data. - Require manual feature engineering
Second generation	- Ability to learn from unlabeled data. - Can automatically learn useful representations	- Limited scalability to large datasets. - Can be sensitive to hyperparameters
Third generation	- Ability to learn from sequential and temporal data. - Can model complex relationships	- Require large amounts of training data. - Can be computationally expensive
Fourth generation	- Can model high-dimensional, unstructured data (e.g., images, audio). - Can learn from limited data	- Require large amounts of computational resources. - Can be difficult to interpret
Fifth generation	- Can learn from a combination of structured and unstructured data. - Can incorporate domain knowledge and prior beliefs. - Can generate high-quality outputs	- Require even more computational resources than fourth-generation models. - Can be even more difficult to interpret

Neural network generations have no strict definition, and it is possible for them to overlap. Research currently being conducted could lead to new developments and generations of neural network models. Figure 4 illustrates different neural networks architecture

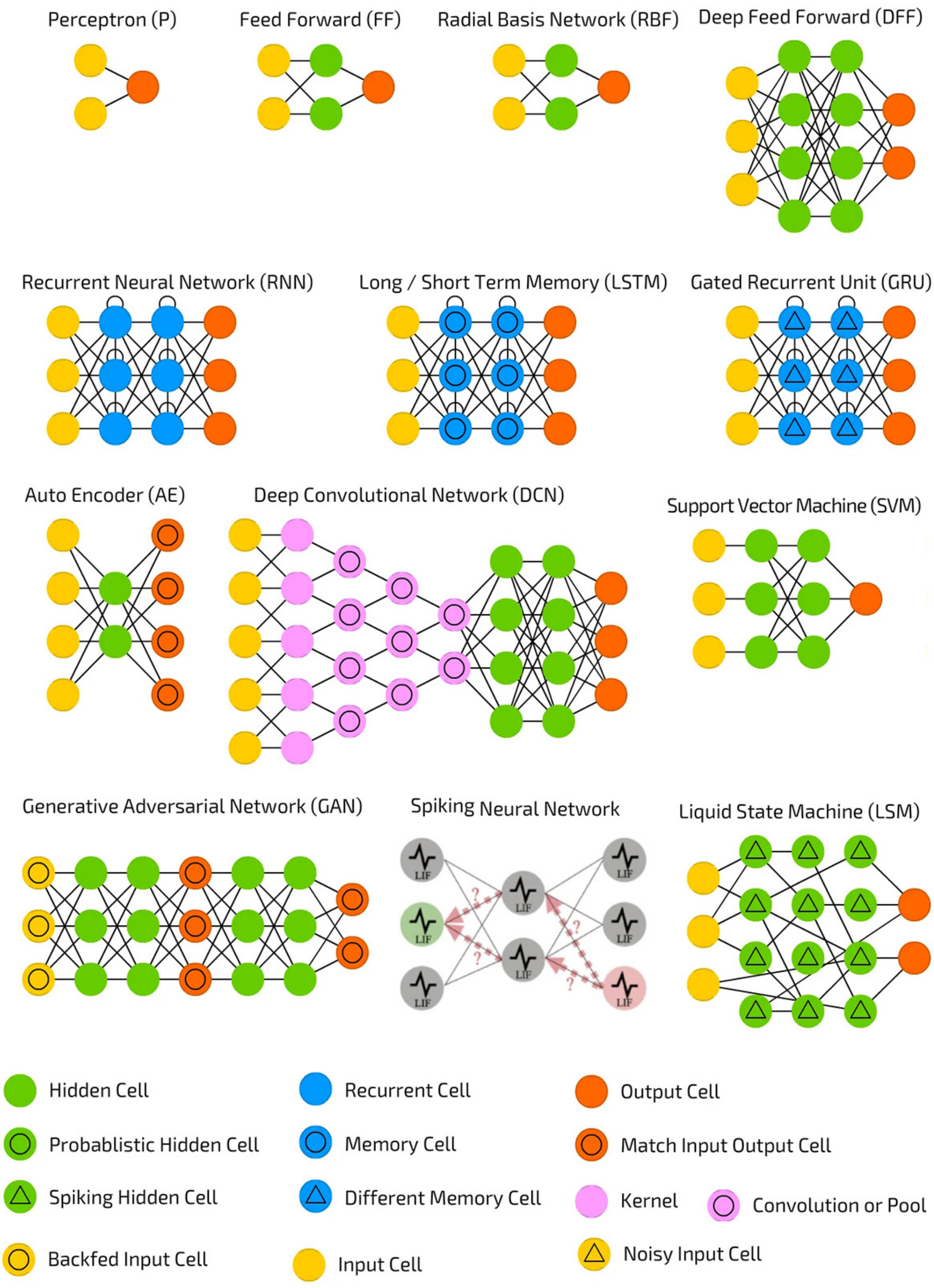


Figure 4. Different neural networks architecture (https://www.asimovinstitute.org/author/fjodorvanveen/).

Table 4 presents some examples of neural network models.

Table 4. some examples of software for implementing neural network models.

Neural Network	Generation	Software	Used Tools
Perceptron	First	TensorFlow, Keras, PyTorch, scikit-learn, MATLAB	Classification, pattern recognition, image processing, signal processing
Adaline	First	TensorFlow, Keras, PyTorch, scikit-learn, MATLAB	Regression, pattern recognition, image processing, signal processing
Multilayer Perceptron (MLP)	Second	TensorFlow, Keras, PyTorch	Regression, classification, pattern recognition
Convolutional Neural Network (CNN)	Second	TensorFlow, Keras, PyTorch, scikit-learn	Computer vision, image processing, speech recognition, natural language processing
Recurrent Neural Network (RNN)	Second	TensorFlow, Keras, PyTorch, scikit-learn	Speech recognition, text analysis, language modeling, time series prediction, video analysis
Generative Adversarial Networks (GANs)	Third	TensorFlow, PyTorch	Image and video synthesis, data augmentation, privacy-preserving data sharing, unsupervised learning
Capsule Networks	Third	TensorFlow, Keras, PyTorch	Object recognition, image classification, medical imaging, natural language processing
Structured Probabilistic Model (SPM)	Fourth	TensorFlow, PyTorch, scikit-learn	Recommendation systems, natural language processing, computer vision, time series analysis
TreeNet	Fourth	XGBoost, LightGBM, scikit-learn	Regression, classification, anomaly detection, time series analysis
Spiking Neural Network (SNN)	Fifth	Brian, NEST, PyNN	Neuroscience, robotics, pattern recognition, time series analysis
Liquid State Machine (LSM)	Fifth	NEST, Brian, PyNN	Time series prediction, speech recognition, anomaly detection, robotics, brain-computer interfaces

8. Transfer Function Impacts

The transfer function is a mathematical function used in neural networks to introduce non-linearity to the output of each neuron in the network. The choice of the transfer function can significantly impact the efficiency and accuracy of the model that is translating each input to a specific output. Sigmoid function is one of the common transfer functions used in neural networks, which maps the input to a value between 0 and 1 [79, 80]. The sigmoid function is commonly used in classification tasks, producing outputs that can be interpreted as probabilities. The sigmoid function can cause neural networks to converge slower during training or not at all due to the gradients of the loss function (vanishing gradient) problem. To address the vanishing gradient problem, alternative transfer functions have been developed. For example, the rectified linear unit (ReLU) transfer function has become increasingly popular recently. ReLU maps the input to a value between 0 and infinity and has a much faster convergence rate than sigmoid, making it ideal for deep neural networks. However, the "dying ReLU" problem can affect the functionality of ReLU, where some node in the network can become "dead" and produce zero output. Other transfer functions commonly used include the hyperbolic tangent (Tanh) function, which maps the input to a value between -1 and 1, and the softmax function, used in multi-class classification problems [81, 82].

The transfer function's choice can significantly impact a neural network's efficiency and accuracy. Choosing the best transfer function enhances neural network efficiency, which depends on the problem specifications and model characteristics. Therefore, we should involve various transfer functions and evaluate their performance based on the network and problem specifications. Table 5 presents the most used transfer function and their specifications.

Table 5. transfer function summary.

Name	Range of Positive Values	Pros	Cons
Step	1 (Binary)	Simple, easy to understand	Not differentiable or continuous, can lead to "vanishing gradients" during training
Sigmoid	(0,1) - Continuous	Smooth, interpretable	Can suffer from the "vanishing gradients" problem
Softmax	[0, 1] - Continuous	Used for multi-class classification problems	Only appropriate for multi-class classification problems, not suitable for regression or other tasks
Tanh	(-1,1) - Continuous	Similar to sigmoid, but with a range of (-1,1)	Can suffer from the "vanishing gradients" problem, slower than ReLU
ArcTan	(-1, 1) - Continuous	Smooth curve, suitable for regression tasks	Less widely used than some other activation functions
ReLU (Rectified Linear Unit)	[0, inf) - Continuous	Simple, fast, widely used, avoids "vanishing gradients" problem	Can lead to "dead neurons" during training
Leaky ReLU	[0, inf) - Continuous	Similar to ReLU, but avoids "dead neurons"	Not as fast or widely used as ReLU
ELU (Exponential Linear Unit)	(-1, inf) - Continuous	Similar to ReLU, but smoother and can reduce risk of "dead neurons"	More computationally expensive than ReLU and Leaky ReLU
Swish	[0, inf) - Continuous	Smooth curve can improve performance on some types of data	Not as widely used as some other activation functions
GELU (Gaussian Error Linear Unit)	(-inf, inf) - Continuous	Smooth, differentiable, can improve performance on certain types of data	More computationally expensive than some other activation functions

9. Hidden layers Impacts

The number of hidden layers in a neural network implementation has the potential to impact its effectiveness and output in various ways. Therefore, more hidden layers in the neural network implementation can improve the accuracy and performance of generalizing the results. It allows us to solve more complex data relationships and attributes. However, On the other hand, adding too many hidden layers can lead to overfitting, and it becomes too specialized for the training data and may perform poorly with unseen data. Additionally, adding more hidden layers can increase the network's computational complexity and training time, which can be a disadvantage in some applications where speed is important. The best number of hidden layers depends on the task specification and dataset and may require some experimentation and tuning [83, 84].

However, determining the best hidden layers number is challenging process as it depends on many features, such as the complexity of the task, the dataset's size, and the network's architecture.

It is generally recommended to use more advanced techniques, such as cross-validation or model selection, to empirically determine the optimal number of neurons in a given layer. However, the N / 2 + 3N rule may provide a useful starting point for determining the size of a neural network. It should be relied upon only partially and supplemented with additional analysis and experimentation to ensure that the resulting network is accurate and efficient [85].

However, there are some general guidelines and techniques that can help in this process:

1. Start with a small number of hidden layers: It is recommended to start with a simple network architecture that includes one or two hidden layers and then gradually increase the number of hidden layers until the desired performance is achieved.
2. Perform model selection: Use techniques such as cross-validation to evaluate the network's performance with different numbers of hidden layers. This can help select the best architecture for the specific task and dataset.
3. Consider the dataset size: A simpler network architecture with fewer hidden layers is recommended to avoid overfitting if the dataset is small.

4. Consider the complexity of the task: More complex tasks may require a larger number of hidden layers to capture the underlying relationships in the data.
5. Regularization techniques: Techniques such as dropout and L1/L2 regularization can help prevent overfitting and reduce the required hidden layers.

10. Neural Network Design

Critical analysis points could be considered when evaluating current neural network model design. A comprehensive analysis would need to consider the specific context and objectives of the research. Some possible critical analysis points that could be considered when analyzing current neural network model design methodologies:

1. Architecture selection: Different neural network architectures have been developed for different applications. A critical analysis should assess the suitability of each architecture for a given problem, considering factors such as the size of the dataset, the complexity of the task, and the availability of computational resources.
2. Hyperparameter tuning: Neural networks have many hyperparameters that must be tuned to achieve optimal performance. A critical analysis should examine the impact of different hyperparameters on the model's performance, such as learning rate, batch size, activation function, and the number of layers.
3. Regularization techniques: Overfitting is a common problem in neural network training. Regularization techniques such as dropout and L1/L2 regularization can reduce overfitting. A critical analysis should evaluate the effectiveness of these techniques in improving model performance and generalization ability.
4. Optimization methods: Gradient descent and its variants commonly optimize neural network models. A critical analysis should assess the limitations of these methods, such as convergence issues and sensitivity to initialization, and explore alternative optimization techniques.
5. Interpretability: One of the main challenges of neural network models is their need for interpretability. A critical analysis should examine the current state-of-the-art interpretability techniques, such as feature visualization and attribution methods, and assess their effectiveness in providing insights into the model's decision-making process.
6. Computational complexity: Training large neural networks can be computationally expensive and time-consuming. A critical analysis should consider the scalability of different neural network architectures and optimization methods and explore ways to reduce the computational complexity of training.
7. Robustness: Neural network models are susceptible to adversarial attacks, where small input perturbations can cause significant output changes. A critical analysis should examine the robustness of different neural network architectures and explore ways to improve their resilience to such attacks.

11. Spiking Neurons

Studies of the cortical pyramidal neurons have shown that the timing of individual spikes as a mode of encoding information is crucial in biological neural networks. A presynaptic neuron communicates with a postsynaptic neuron via trains of spikes or action potentials. Biological spikes have a fixed morphology and amplitude. The transmitted information is usually encoded in the frequency of spiking (rate encoding) and/or in the timing of the spikes (pulse encoding) [86]. Pulse encoding is more powerful than rate encoding regarding the wide range of information the same number of neurons may encode. Rate encoding is a special case of pulse encoding. The average firing rate can be computed if the spike timings are known. The early first-generation neurons developed in the 1940s and 1950s did not involve any encoding of the temporal aspect of information processing. These neurons acted as simple integrate-and-fire units that fired if the internal state (defined as the weighted sum of inputs to each neuron) reached a threshold. It did not matter when the threshold was exceeded. Translating this assumption to a biological perspective implied that all inputs to the neuron were synchronous, i.e., contributed to the internal state at precisely the same time and, therefore, could be directly summed. However, unlike biological neurons, the magnitude of the input was allowed to contribute to the internal state. This may have represented a primitive form of rate encoding because a more significant input (representing a higher firing rate of the presynaptic neuron) may cause the postsynaptic neuron to reach the threshold [87].

12. Performance Metrics

Performance metrics are used to evaluate the accuracy and effectiveness of Artificial Neural Networks (ANNs) in solving a given task [88, 89]. Here are some commonly used performance metrics for ANNs:

- ➢ Accuracy measures the proportion of correct predictions the network makes on a test dataset. It is commonly used for classification tasks.
- ➢ Precision and Recall: They are used for classification tasks with multiple classes. Precision measures the proportion of true positives (correctly classified instances of a given class) to the total number of instances classified as that class. Recall measures the proportion of true positives to the total number of instances of that class in the test dataset.
- ➢ F1 Score combines precision and recall into a single value. It helps compare the performance of different models.
- ➢ The correlation coefficient (R) has a value between 0 and 1 determines the relationship between the two variables.
- ➢ Mean Squared Error (MSE) measures the average of the squared differences between the predicted and actual values.
- ➢ Root Mean Squared Error (RMSE) is the square root of the MSE and is also commonly used for regression tasks.
- ➢ Mean Absolute Error (MAE) measures the average absolute differences between the predicted and actual values. It is another commonly used metric for regression tasks.
- ➢ ROC Curve and Area Under the Curve (AUC) are used for binary classification tasks and measure the tradeoff between true and false positive rates at different thresholds. The ROC curve plots the true positive rate against the false positive rate at different thresholds, while the AUC measures the area under the ROC curve.
- ➢ Confusion Matrix shows the number of true positives, true negatives, false positives, and false negatives for each class in a classification task.

These are some commonly used performance metrics in ANNs, but other metrics may be more appropriate for specific tasks. Therefore, metrics in ANNs should be selected based on the task and goals of the model [90- 94]. Here are some commonly used performance metrics for Artificial Neural Networks (ANNs) along with their formulas:

1. Accuracy:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives.

2. Precision:

$$\text{Precision} = TP / (TP + FP)$$

3. Recall:

$$\text{Recall} = TP / (TP + FN)$$

4. F1 Score:

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

5. Mean Squared Error (MSE):

$$MSE = (1 / n) * \sum(i=1 \text{ to } n) (yi - \hat{y}i)^2$$

where n = number of instances, yi = actual value, and ŷi = predicted value.

6. Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{}(MSE)$$

7. Mean Absolute Error (MAE):

   MAE = (1 / n) * ∑(i=1 to n) |yi - ŷi|

8. ROC Curve and Area Under the Curve (AUC):

   The ROC curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds. The AUC is the area under the ROC curve.

   TPR = TP / (TP + FN)

   FPR = FP / (FP + TN)

9. Confusion Matrix is an indicator of the evaluation performance that provides a summary of the classification results. It is mainly, a table that shows the number of true positives, true negatives, false positives, and false negatives for each class in a classification task.

These are some commonly used performance metrics for ANNs along with their formulas. It's important to select appropriate metrics based on the task and goals of the model.

13. Conclusion

Neural networks are increasingly used to learn complex patterns and make accurate predictions. Designing an effective neural network model is a challenging task that requires careful consideration of architecture, optimization methods, and regularization techniques. This paper provides a comprehensive overview of the current artificial neural network generation state-of-the-art. It explores various learning approaches, identifies common pitfalls and limitations of existing design methodologies, and proposes future research directions. Despite impressive accomplishments, new developments in the field of neural network computation and architectures. However, Neural networks still suffer from different limitations that need to be addressed, including robustness, understandability, and scalability.

Some pitfalls and limitations of existing neural network design methodologies:

- Overfitting is one of the most common problems in neural network design is overfitting, which occurs when the model becomes too complex and learns the training data too well, resulting in poor performance on new, unseen data. This can be mitigated by regularization, early stopping, and data augmentation.
- Lack of Interpretability: Understanding how the model arrived at a particular decision or prediction can be challenging. This can be particularly problematic in domains where transparency and accountability are important, such as healthcare or finance. Several techniques have been proposed to address this issue, such as model distillation, attention mechanisms, and gradient-based attribution methods.
- Computational Complexity: Neural networks can be computationally expensive to train and evaluate, especially for large datasets and complex architectures. This can limit their practical use in specific domains or applications. Many researchers have explored techniques such as model compression, quantization, and knowledge distillation to address this limitation.
- Lack of Generalization: Neural networks can sometimes need help to generalize to new, unseen data that may differ from the training data in some pattern. This is particularly true when training datasets are limited or biased, or the model is too complex and prone to overfitting. Researchers have explored techniques such as transfer learning, domain adaptation, and adversarial training to address this.

Some potential research gaps related to the development of neural network generations:

- First Generation Neural Networks:
  While the perceptron and Adaline were important early models in neural network history, their limitations in handling non-linearly separable data were quickly realized. However, it is still an open research question how these models can be improved to work on more complex data.
- Second Generation Neural Networks:

  - MLPs are widely used in many applications. However, there is still a need for research on how to effectively choose the number of hidden layers and neurons and optimize the learning rate and other hyperparameters.
  - RBFNs have been shown to work well in specific applications, but their effectiveness can be limited by the choice of radial basis functions and the number of hidden units.
- Third Generation Neural Networks:
  - CNNs have achieved remarkable success in image and video processing tasks, but there are still challenges in extending their use to other domains, such as natural language processing.
  - RNNs have shown promise in modeling sequential data. However, there is still a need for research on overcoming the vanishing gradient problem and handling long-term dependencies effectively.
- Fourth Generation Neural Networks:
  - Neuro-symbolic AI is an emerging field that combines neural networks with symbolic reasoning techniques, but there are still challenges in effectively integrating these two approaches.
  - Few-shot learning is an active area of research, but research is still needed to develop more robust and scalable models that can generalize to new domains.
  - Explainable AI is an important area of research. However, there are still challenges in developing accurate and interpretable models and understanding how to communicate the explanations to end users effectively.

Finally, while significant progress has been made in developing neural network models, there are still many open research questions and opportunities for improvement and innovation, such as developing techniques to make ANNs more interpretable and explainable for applications where decisions must be justified or understood. Also, seek to improve transfer Learning to make it more accessible and adaptable to different domains and languages. In addition, researchers need to propose more robust models and increase security-critical defenses against adversarial attacks.

References

1. Kreuzberger, D., Kühl, N. and Hirschl, S., 2023. Machine learning operations (mlops): Overview, definition, and architecture. IEEE Access.
2. Shu, X. and Ye, Y., 2023. Knowledge Discovery: Methods from data mining and machine learning. Social Science Research, 110, p.102817.
3. Yousif, J.H. and Saini, D.K., 2020. Big Data Analysis on Smart Tools and Techniques. Cyber Defense Mechanisms, pp.111-130.
4. Hu, Weiming, Xintong Li, Chen Li, Rui Li, Tao Jiang, Hongzan Sun, Xinyu Huang, Marcin Grzegorzek, and Xiaoyan Li. "A state-of-the-art survey of artificial neural networks for whole-slide image analysis: from popular convolutional neural networks to potential visual transformers." Computers in Biology and Medicine 161 (2023): 107034.
5. Yousif, J.H., Kazem, H.A., Al-Balushi, H., Abuhmaidan, K. and Al-Badi, R., 2022. Artificial Neural network modelling and experimental evaluation of dust and thermal energy impact on monocrystalline and polycrystalline photovoltaic modules. Energies, 15(11), p.4138.
6. Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R. and Shahzad, M., 2023. A survey of uncertainty in deep neural networks. Artificial Intelligence Review, pp.1-77.
7. Yousif, J.H. and AlRababaa, M.S., 2013. Neural technique for predicting traffic accidents in Jordan. Journal of American Science, 9(11), pp.528-525.
8. Yilmaz, I. and Kaynar, O., 2011. Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. Expert systems with applications, 38(5), pp.5958-5966.
9. Yousif, J.H. and Saini, D.K., 2011. Hindi Part-Of-Speech Tagger Based Neural Networks. Journal of Computing, 3(2), pp.59-66.
10. Wu, X., Niu, R., Ren, F. and Peng, L., 2013. Landslide susceptibility mapping using rough sets and back-propagation neural networks in the Three Gorges, China. Environmental earth sciences, 70, pp.1307-1318.

11. Saini, D.K. and Yousif, J.H., 2013. Environmental Scrutinizing System based on Soft Computing Technique. International Journal of Computer Applications, 62(13).
12. Fekihal, M.A. and Yousif, J.H., 2012. Self-organizing map approach for identifying mental disorders. International Journal of Computer Applications, 45(7), pp.25-30.
13. Song, H., Kim, M., Park, D., Shin, Y. and Lee, J.G., 2022. Learning from noisy labels with deep neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems.
14. Yousif, J., 2018. Neural computing based part of speech tagger for Arabic language: a review study. International Journal of Computation and Applied Sciences IJOCAAS, 5(1).
15. Sahoo, S., Kumar, S., Abedin, M.Z., Lim, W.M. and Jakhar, S.K., 2023. Deep learning applications in manufacturing operations: a review of trends and ways forward. Journal of Enterprise Information Management, 36(1), pp.221-251.
16. Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R. and Poria, S., 2023. A review of deep learning techniques for speech processing. Information Fusion, p.101869.
17. Gheisari, M., Ebrahimzadeh, F., Rahimi, M., Moazzamigodarzi, M., Liu, Y., Dutta Pramanik, P.K., Heravi, M.A., Mehbodniya, A., Ghaderzadeh, M., Feylizadeh, M.R. and Kosari, S., 2023. Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey. CAAI Transactions on Intelligence Technology.
18. Yousif, J.H. and Sembok, T., 1812. Arabic part-of-speech tagger based neural networks. In proceedings of International Arab Conference on Information Technology ACIT2005, ISSN (Vol. 857).
19. Garcez, A.D.A. and Lamb, L.C., 2023. Neurosymbolic AI: The 3 rd wave. Artificial Intelligence Review, pp.1-20.
20. Sheth, A., Roy, K. and Gaur, M., 2023. Neurosymbolic ai-why, what, and how. arXiv preprint arXiv:2305.00813.
21. Krenzer, A., Heil, S., Fitting, D., Matti, S., Zoller, W.G., Hann, A. and Puppe, F., 2023. Automated classification of polyps using deep learning architectures and few-shot learning. BMC Medical Imaging, 23(1), p.59.
22. Javed, A.R., Ahmed, W., Pandya, S., Maddikunta, P.K.R., Alazab, M. and Gadekallu, T.R., 2023. A survey of explainable artificial intelligence for smart cities. Electronics, 12(4), p.1020.
23. Tang, N., Gong, S., Zhou, J., Shen, M. and Gao, T., 2023. Generative visual common sense: Testing analysis-by-synthesis on Mondrian-style image. Journal of Experimental Psychology: General.
24. Pany, S.S., Singh, S.G., Kar, S. and Dikshit, B., 2023. Stochastic modelling of diffused and specular reflector efficiencies for scintillation detectors. Journal of Optics, pp.1-12.24.
25. Xi, L., Tang, W. and Wan, T., 2023. TreeNet: Structure preserving multi-class 3D point cloud completion. Pattern Recognition, 139, p.109476.
26. Ahmed, F.Y., Masli, A.A., Khassawneh, B., Yousif, J.H. and Zebari, D.A., 2023. Optimized Downlink Scheduling over LTE Network Based on Artificial Neural Network. Computers, 12(9), p.179.
27. Chunduri, R.K. and Perera, D.G., 2023. Neuromorphic Sentiment Analysis Using Spiking Neural Networks. Sensors, 23(18), p.7701.
28. Yi, Z., Lian, J., Liu, Q., Zhu, H., Liang, D. and Liu, J., 2023. Learning rules in spiking neural networks: A survey. Neurocomputing, 531, pp.163-179.
29. Ali, Y.H., Ahmed, F.Y., Abdelrhman, A.M., Ali, S.M., Borhana, A.A. and Hamzah, R.I.R., 2022, October. Novel Spiking Neural Network Model for Gear Fault Diagnosis. In 2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA) (pp. 1-6). IEEE.
30. Li, N., Ma, L., Xing, T., Yu, G., Wang, C., Wen, Y., Cheng, S. and Gao, S., 2023. Automatic design of machine learning via evolutionary computation: A survey. Applied Soft Computing, p.110412.
31. Liu, S., Lin, Q., Li, J. and Tan, K.C., 2023. A survey on learnable evolutionary algorithms for scalable multiobjective optimization. IEEE Transactions on Evolutionary Computation.
32. Alkishri, w. and al-Bahri, Mahmood, Deepfake Image detection methods using discrete Fourier transform analysis and convolutional neural network.
33. KHAMIS, Y. and Yousif, J.H., 2022. Deep learning Feedforward Neural Network in predicting model of Environmental risk factors in the Sohar region. Artificial Intelligence & Robotics Development Journal, pp.201-2013.
34. Kamalov, F., Cherukuri, A.K., Sulieman, H., Thabtah, F. and Hossain, A., 2023. Machine learning applications for COVID-19: a state-of-the-art review. Data Science for Genomics, pp.277-289.

35. Ghazal, T.M., Hasan, M.K., Ahmad, M., Alzoubi, H.M. and Alshurideh, M., 2023. Machine Learning Approaches for Sustainable Cities Using Internet of Things. In The Effect of Information Technology on Business and Marketing Intelligence Systems (pp. 1969-1986). Cham: Springer International Publishing.
36. Yousif, J.H., Zia, K. and Srivastava, D., 2022. Solutions Using Machine Learning for Diabetes. Healthcare Solutions Using Machine Learning and Informatics, pp.39-59.
37. Ray, A., Kolekar, M.H., Balasubramanian, R. and Hafiane, A., 2023. Transfer learning enhanced vision-based human activity recognition: a decade-long analysis. International Journal of Information Management Data Insights, 3(1), p.100142.
38. Figueiredo, E., Omori Yano, M., Da Silva, S., Moldovan, I. and Adrian Bud, M., 2023. Transfer learning to enhance the damage detection performance in bridges when using numerical models. Journal of Bridge Engineering, 28(1), p.04022134.
39. Aimen, A., Ladrecha, B., Sidheekh, S. and Krishnan, N.C., 2023. Leveraging Task Variability in Meta-learning. SN Computer Science, 4(5), p.539.
40. Maier, H.R., Galelli, S., Razavi, S., Castelletti, A., Rizzoli, A., Athanasiadis, I.N., Sànchez-Marrè, M., Acutis, M., Wu, W. and Humphrey, G.B., 2023. Exploding the myths: An introduction to artificial neural networks for prediction and forecasting. Environmental Modelling & Software, p.105776.
41. Yousif, J., 2021. Implementation of Big Data Analytics for Simulating, Predicting & Optimizing the Solar Energy Production. Appl. Comput. J, 1, pp.133-140.
42. Khan, N.S. and Ghani, M.S., 2021. A survey of deep learning based models for human activity recognition. Wireless Personal Communications, 120(2), pp.1593-1635.
43. Pomazan, V., Tvoroshenko, I. and Gorokhovatskyi, V., 2023. Development of an application for recognizing emotions using convolutional neural networks.
44. Srivastava, D., Sharma, N., Sinwar, D., Yousif, J.H. and Gupta, H.P. eds., 2023. Intelligent Internet of Things for Smart Healthcare Systems. CRC Press.
45. Alam, M.S., Mohamed, F.B., Selamat, A. and Hossain, A.B., 2023. A review of recurrent neural network based camera localization for indoor environments. IEEE Access.
46. Kazem, H.A., Yousif, J.H., Chaichan, M.T., Al-Waeli, A.H. and Sopian, K., 2022. Long-term power forecasting using FRNN and PCA models for calculating output parameters in solar photovoltaic generation. Heliyon, 8(1).
47. Li, P., Pei, Y. and Li, J., 2023. A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing, p.110176.
48. Seghiour, A., Abbas, H.A., Chouder, A. and Rabhi, A., 2023. Deep learning method based on autoencoder neural network applied to faults detection and diagnosis of photovoltaic system. Simulation Modelling Practice and Theory, 123, p.102704.
49. Wang, R., Bashyam, V., Yang, Z., Yu, F., Tassopoulou, V., Chintapalli, S.S., Skampardoni, I., Sreepada, L.P., Sahoo, D., Nikita, K. and Abdulkadir, A., 2023. Applications of generative adversarial networks in neuroimaging and clinical neuroscience. Neuroimage, p.119898.
50. Gui, J., Sun, Z., Wen, Y., Tao, D. and Ye, J., 2021. A review on generative adversarial networks: Algorithms, theory, and applications. IEEE transactions on knowledge and data engineering, 35(4), pp.3313-3332.
51. Xie, P., Zhou, A. and Chai, B., 2019. The application of long short-term memory (LSTM) method on displacement prediction of multifactor-induced landslides. IEEE Access, 7, pp.54305-54311.
52. Fang, L. and Shao, D., 2022. Application of long short-term memory (LSTM) on the prediction of rainfall-runoff in karst area. Frontiers in Physics, 9, p.685.
53. Alattar, N., Yousif, J.A.B.A.R., Jaffer, M. and Aljunid, S.A., 2019. Neural and Mathematical Predicting Models for Particulate Matter Impact on Human Health in Oman. WSEAS Trans Env & Dev, 15, pp.578-585.
54. Yousif, J.H., Saini, D.K. and Uraibi, H.S., 2011, July. Artificial intelligence in e-leaning-pedagogical and cognitive aspects. In Proceedings of the World Congress on Engineering (Vol. 2, pp. 6-8).
55. Wang, J., Li, X., Li, J., Sun, Q. and Wang, H., 2022. NGCU: A new RNN model for time-series data prediction. Big Data Research, 27, p.100296.
56. Meyes, R., Donauer, J., Schmeing, A. and Meisen, T., 2019. A recurrent neural network architecture for failure prediction in deep drawing sensory time series data. Procedia Manufacturing, 34, pp.789-797.
57. Kim, J., Jung, W., An, J., Oh, H.J. and Park, J., 2023. Self-optimization of training dataset improves forecasting of cyanobacterial bloom by machine learning. Science of The Total Environment, 866, p.161398.

58. Yousif, J.H., 2011. Information Technology Development. LAP LAMBERT Academic Publishing, Germany ISBN 9783844316704.
59. Esfe, M.H., Eftekhari, S.A., Alizadeh, A.A., Emami, N. and Toghraie, D., 2023. Investigation of best artificial neural network topology to model the dynamic viscosity of MWCNT-ZnO/SAE 5W30 nano-lubricant. Materials Today Communications, 35, p.106074.
60. Haritha, K., Shailesh, S., Judy, M.V., Ravichandran, K.S., Krishankumar, R. and Gandomi, A.H., 2023. A novel neural network model with distributed evolutionary approach for big data classification. Scientific Reports, 13(1), p.11052.
61. Chen, H., Tao, R., Fan, Y., Wang, Y., Wang, J., Schiele, B., Xie, X., Raj, B. and Savvides, M., 2023. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. arXiv preprint arXiv:2301.10921.
62. Abdalgader, K. and Yousif, J.H., 2022. Agricultural Irrigation Control using Sensor-enabled Architecture. KSII Transactions on Internet & Information Systems, 16(10).
63. Besbes, O., Ma, W. and Mouchtaki, O., 2023. Quality vs. quantity of data in contextual decision-making: Exact analysis under newsvendor loss. arXiv preprint arXiv:2302.08424.
64. Behler, J., 2021. Four generations of high-dimensional neural network potentials. Chemical Reviews, 121(16), pp.10037-10072.
65. Lakra, S., Prasad, T.V. and Ramakrishna, G., 2012. The future of neural networks. arXiv preprint arXiv:1209.4855.
66. Ghosh, J. and Nag, A., 2001. An overview of radial basis function networks. Radial basis function networks 2: new advances in design, pp.1-36.
67. Howlett, R.J., 2001. Radial basis function networks 1: recent developments in theory and applications.
68. Lim, E.A., Tan, W.H. and Junoh, A.K., 2019. An improved radial basis function networks based on quantum evolutionary algorithm for training nonlinear datasets. IAES International Journal of Artificial Intelligence, 8(2), p.120.
69. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F.E., 2017. A survey of deep neural network architectures and their applications. Neurocomputing, 234, pp.11-26.
70. Deng, L., Hinton, G. and Kingsbury, B., 2013, May. New types of deep neural network learning for speech recognition and related applications: An overview. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 8599-8603). IEEE.
71. Yousif, J.H. and Sembok, T., 2006. Recurrent neural approach based Arabic part-of-speech tagging. In proceedings of International Conference on Computer and Communication Engineering (ICCCE'06) (Vol. 2, pp. 9-11).
72. Bashar, D.A., 2019. Survey on evolving deep learning neural network architectures. Journal of Artificial Intelligence and Capsule Networks, 1(2), pp.73-82.
73. Dhillon, A. and Verma, G.K., 2020. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, 9(2), pp.85-112.
74. Sutskever, I., Martens, J. and Hinton, G.E., 2011. Generating text with recurrent neural networks. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 1017-1024).
75. Samek, W. and Müller, K.R., 2019. Towards explainable artificial intelligence. Explainable AI: interpreting, explaining and visualizing deep learning, pp.5-22.
76. Tjoa, E. and Guan, C., 2020. A survey on explainable artificial intelligence (xai): Toward medical xai. IEEE transactions on neural networks and learning systems, 32(11), pp.4793-4813.
77. 77.Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y. and Shi, L., 2019, July. Direct training for spiking neural networks: Faster, larger, better. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 1311-1318).
78. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T. and Maida, A., 2019. Deep learning in spiking neural networks. Neural networks, 111, pp.47-63.
79. Yonaba, H., Anctil, F. and Fortin, V., 2010. Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. Journal of hydrologic engineering, 15(4), pp.275-283.
80. Le, T.H., Jang, H. and Shin, S., 2021. Determination of the optimal neural network transfer function for response surface methodology and robust design. Applied Sciences, 11(15), p.6768.
81. Yousif, J.H. and Sembok, T.M.T., 2008, August. Arabic part-of-speech tagger based Support Vectors Machines. In 2008 International Symposium on Information Technology (Vol. 3, pp. 1-7). IEEE.

82. Yousif, J.H., 2013. Natural language processing based soft computing techniques. International Journal of Computer Applications, 77(8), pp.43-49.
83. Shafi, I., Ahmad, J., Shah, S.I. and Kashif, F.M., 2006, December. Impact of varying neurons and hidden layers in neural network architecture for a time frequency application. In 2006 IEEE International Multitopic Conference (pp. 188-193). IEEE.
84. Ramchoun, H., Ghanou, Y., Ettaouil, M. and Janati Idrissi, M.A., 2016. Multilayer perceptron: Architecture optimization and training.
85. Heaton, J., 2008. Introduction to neural networks with Java. Heaton Research, Inc.
86. Paugam-Moisy, H. and Bohte, S.M., 2012. Computing with spiking neuron networks. Handbook of natural computing, 1, pp.1-47.
87. Maguire, L.P., McGinnity, T.M., Glackin, B., Ghani, A., Belatreche, A. and Harkin, J., 2007. Challenges for large-scale implementations of spiking neural networks on FPGAs. Neurocomputing, 71(1-3), pp.13-29.
88. Yousif, J., 2019. Hidden Markov Model tagger for applications based Arabic text: A review. Journal of Computation and Applied Sciences IJOCAAS, 7(1).
89. Kazem, H.A., Yousif, J.H. and Chaichan, M.T., 2016. Modeling of daily solar energy system prediction using support vector machine for Oman. International Journal of Applied Engineering Research, 11(20), pp.10166-10172.
90. Alsumaiei, A.A., 2020. Utility of artificial neural networks in modeling pan evaporation in hyper-arid climates. Water, 12(5), p.1508.
91. Yousif, J.H. and Kazem, H.A., 2016. Research Article Modeling of Daily Solar Energy System Prediction using Soft Computing Methods for Oman. Research Journal of Applied Sciences, Engineering and Technology, 13(3), pp.237-244.
92. Yousif, J.H. and Abdalgader, K., 2022. Experimental and mathematical models for real-time monitoring and auto watering using IoT architecture. Computers, 11(1), p.7.
93. Prediction and evaluation of photovoltaic-thermal energy systems production using artificial neural network and experimental dataset.
94. Yousif, J.H. and Sembok, T., 2006. Design and implement an automatic neural tagger based arabic language for NLP applications. Asian Journal of Information Technology, 5(7), pp.784-789.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.