Review Article

Realizations of the Artificial Neural Network for Process Modeling. Overview of Current Implementations

Alytis Gruodis*
Department of IT, Vilnius Business College, Kalvarijų g. 129-401, LT-08221 Vilnius, Lithuania
* Corresponding author, e-mail: *alytis.gruodis@gmail.com*

Received: 01 December 2023
Accepted: 20 December 2023
Online: 30 December 2023

JEL: C45.

Abstract. This work is intended to review the most typical realizations of Artificial Neural Networks (ANNs), implemented in a Feedforward Neural Network (FNN) as well as a Recurrent Neural Network (RNN). Essential differences in ANN architecture and basic operating principles are discussed. The problems of learning processes are presented in several cuts. The advantages of prediction using ANNs have been demonstrated in several popular fields such as adaptive educology, classification of medicine and biology, industry, etc.

Keywords: Artificial Intelligence; Artificial Neural Network; Feedforward Neural Network; Recurrent Neural Network; Perceptron.

Citation: Alytis Gruodis (2023) Realizations of the Artificial Neural Network for Process Modeling. Overview of Current Implementations. – *Applied Business: Issues & Solutions* 2(2023)22–27 – ISSN 2783-6967.
https://doi.org/10.57005/ab.2023.2.3

Introduction

In variuos fiels of industries, Artificial Intelligence (AI) and Machine Learning (ML) have a wide range of applications. In manufacturing to reduce downtime, and to enhance overall operational efficiency, AI could be applied for quality control, supply chain optimization, and process automation [1-2]. For development of self-driving cars and drones several complex operations such as real-time decision-making, navigation, and object detection, image recognition could be realized using AI only [3-4]. In social plane, to create communication between humans and computers, the Natural Language Processing (NLP) as subfield of AI must be implemented for speech recognition, language translation, chatbots etc [4-5]. In economic plane, performance in forecasting stock indices and future predictions could be applied using different types of networks in relation to Multilayer Perceptron (MLP) model [6].

In education, personalized learning and recomendations of educational content could be realized due to AI and ML technologies [7-8]. Possibility to create an individual learning style adapted to human represents an effective realization of one of the greatest educational issue of XX century [9].

In AI and ML, Artificial Neural Networks (ANNs) represent main realization of predictions in decision making processes. According to neuron networks in biological systems, nonlinear reswitching function of neuron was implemented in perceptrons which represent an operation unit in ANNs [10]. Behaviours presented below determine the essence of ANNs: a) learning from data and continuous improvement; b) fault tolerance; c) non-linearity; d) adaptability.

Big amount of real data is too complex for direct analysis using formula representations. After tranditional modelling, some conclusions could be questioned due to locality of solutions and impossibility to join several fields of data. Usage of ANN allows to learn the complex patterns and relationships from large weak correlated data sets of different type [11]. In many fiels of human development, amount of data increases continuously. Due to that, previously created and trained ANN can be further trained and fine-tuned to improve their prediction again and again. Basheera et al [12] accented that attractiveness of ANNs comes from their remarkable information processing characteristics pertinent mainly to high parallelism, noise tolerance, and learning capabilities. This adaptability allows them to evolve and stay relevant over time [13].

Fault and noise tolerance represents critical problem in information science, which must be solved using individual approach. For ANNs, predictions using well trained networks are not sensitive to missing of portion of data or network damaging due to network facilities. In many cases, ANNs often exhibit fault tolerance [14].

Non-linear data are obtained from big number of processes related to the real-world problems. Traditional linear models in economics cannot be sufficient for representation of such complexities. In that case, non-linear data relationships could be succesfully analysed using ANNs [15-17]. Jaiswal et al [15] claims that stock price does not follow any deterministic regulatory function, factor or circumstances rather than many considerations such as economy and finance etc.

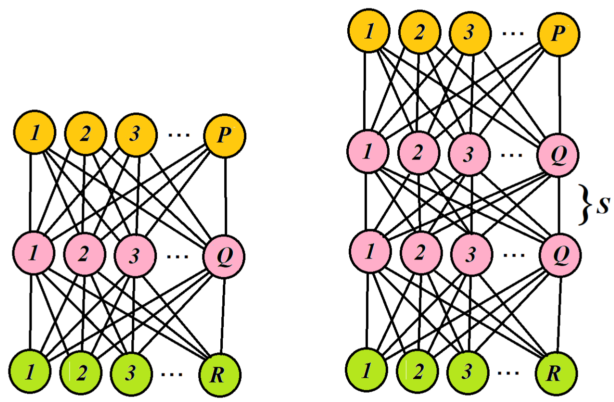
Adaptability of ANN to complex parameters is well known due to network possibility to generalize new, unknown data of different type [18]. Classifications of current changes in inputs and following predictions allow adjusting their internal parameters during the learning process. ANNs represent non-linear, data-drive, and self-adaptive methods that do not require specific assumptions about the underlying model [6]. Methods for knowledge initialization, knowledge refinement and knowledge extraction [19] represent several secial constraints of ANN imposed by nonmonotonicity (ANNN).

Generally, ANNs are necessary because learning from current weak correlated data allows us to create novel models containing complex relationships, and adapt such models to new information.

This work is devoted to review the most typical implementations of ANN, essential differences in ANN architecture to describe main operational principles, the problems in the learning processes, and fields, where predictions are welcome.

1. Artificial Neural Network: architecture

All artificial neural networks ANNs are based on the concept of neurons, connections, and transfer functions. According to Kohli [20], ANN is an attempt to simulate the brain activity. Simulated ANN as very simplified brain could be used for solving some mathematical tasks according to operational principles inherited from biological neurons in synapses. As presented in biology, real tasks such as pattern recognition or data classification could be partially of fully solved after corresponding learning process. Learning in biological systems involves adjustments or readjustments to the synaptic connections that exist between the neurons.



a) single hidden layer; b) multiple hidden layers, S>1.

Fig. 1. Feedforward neural networks (FNNs).
Hidden layer (red): number of perceptrons Q,
input layer (yellow): number of units P,
output layer (green): number of units R.

Wang [4] presented several models and algorithms of ANNs and analyses of the main ideas. Complete taxonomy of two main groups such as feed-forward neural networks (FNNs) and recurrent/feedback neural networks (RNNs) was described by Jain et al [14]. Due to the link pattern, two categories of ANNs occur: a) FNNs, where graphs have no loops; b) RNNs where loops in graphs occur because of feedback links. Gómez-Ramos et al [6] expanded previous classification up to four groups of networks used as forecasting tools: 1) feed-forward neural networks (FNNs), like the Multi-Layer Perceptron (MLP), 2) recurrent/feedback neural networks (RNNs), 3) modular networks (MN), and 4) support vector machine (SVM).

Feedforward Neural Networks. FNNs, also known as MLPs, form the foundation of many deep learning models. Layers of interconnected nodes represent an feedforward network. Term *feed-forward* represents information flow in one direction from the input layer through hidden layers to the output layer:

$$input \rightarrow hidden \rightarrow output. \tag{1}$$

Fig. 1 represents schemes of FNNs in two realizations: a) single hidden layer; b) multiple hidden layers, S>1. Input layer and output layer consist of input ports, so called units, where number of units is equal to P and R respectively. Hidden layer (or layers) consist of certain number of perceptrons Q, and Q>P. This condition must be realised according to the needs of tasks and could be treated as non-mandatory condition.

Walczak et al [11] presented the problem of perceptron's quantity in hidden layer. Walczak described several empirical heuristical approaches used before in different projects.
Let us consider N_h as number of hidden nodes for an ANN, N_{in} and N_{out} are numbers of units in input and output layers respectively. Three intuitive rules for establishing the number of perceptrons in the hidden layer are presented below.

$$N_h = \frac{3}{4} \cdot N_{in} \tag{2}$$

$$N_h = \frac{1}{2} \cdot (N_{in} + N_{out}) \tag{3}$$

$$N_h = 2 \cdot N_{in} + 1 \tag{4}$$

To calculate the required number of perceptrons in the hidden layer, Bekešienė et al [21] presented several formulas that are variations of Eq.(47).

Perceptrons as artificial neurons in each layer perform a weighted sum of their inputs, followed by the application of an activation function. The weighted sum is computed using values representing the properties of link between percetrons. In hidden layer, each link has an associated value, which could be (or must be) changed during the training process. These values or weights determine the strength of influence between perceptrons.

Traditional FNNs have no memory of previous inputs and the fixed architecture. Three characteristics are important for defining the ANN [22]: a) number of hidden layers and number of nodes in each layer; b) mechanism of learning which is responsible for updating the weights of links; c) activation function used in various layers.

For FNNs, condition of stationarity assumption must be realized. FNNs assume that the underlying statistical properties of the input data do not change over time. For tasks involving non-stationary data FNNs are not suitable.

Ability to approximate any continuous function in the framework of certain conditions is one of the most important fundamental properties of FNNs, known as the *Universal Approximation Theorem* [23].

However, it is shown that the FNNs has important delimitations in network architecture: fixed input size; limited handling of time series data; sensitivity of hyperparameter.

For FNNs, a fixed number of features as input must be presented. Due to restrictions of architecture, size of input for training and for predictions must be the same. Convolutional Neural Networks (CNNs) are better suited for tasks involving image data, where the input size can vary.

For FNNs, limited handling of time series data is the serious problem. FNNs are not adapted for processing of the time series data or text item data. Long Short-Term Memory networks (LSTMs) and Recurrent Neural Networks (RNNs) are more adapted for tasks involving time series data.

Architecture of ANN could be described using following statements [24]. A set of operating interconected elements (perceptrons or nodes) represents an directed graph. Let us consider a set of input units x_i, i=1, 2, . . ., m, and a set of output units y_j, j=1, 2, . . ., n. Each node j performs a transfer function f_j of the form

$$y_j = f_j\Big(\sum_{i=0}^{m} w_{ji} x_i - \Theta_j\Big). \tag{5}$$

Connection weight between nodes i and j is denoted as w_{ji}, and threshold (or bias) of the node is denoted as Θ_j. Transfer function f_j is nonlinear, such as a heaviside, sigmoid, or Gaussian function.

For FNNs, sensitivity of hyperparameter exists in certain interval. Some adjusted parameters such as number of layers, number of perceptrons per layer, learning rate can be sensitive to the choice of presented set. Finding of the optimal set can be established after multiple simulations of learning procedure and prediction events.

Recurrent Neural Networks. RNNs are a part of ANNs designed for processing the sequential data. In a RNNs, at least one feedback connection must be present. The Hopfield model and the Boltzmann machine are two examples of RNNs. Dynamic and flexible structure of RNNs allows to handle sequences by maintaining a form of memory. Fig. 2 represents the schema of Recurrent Neural Network (RNN) where backpropagation in hidden layers was realized two times.

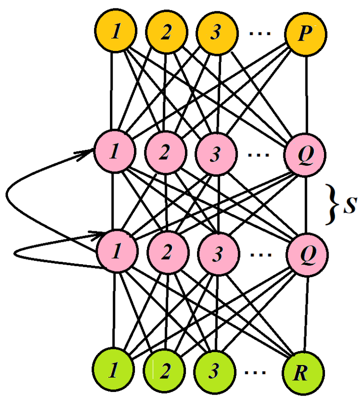


Fig. 2. Recurrent Neural Networks (RNNs).
Multiple hidden layers, S>1 (red): number of perceptrons Q,
input layer (yellow): number of units P,
output layer (green): number of units R.

Main behaviour of RNNs could be formulated as follows: hidden states for memory; special parameter sharing; flexibility for variable-length sequences [25-26].

The hidden state of the RNNs could be used as an internal memory for storage of information from previous steps. This hidden state is updated at each time step based on the current input and the previous hidden state. This realization allows the network to maintain context that evolves as the sequence progresses.

For RNNs, special parameter sharing is realized. RNNs use the same set of weights and biases across all time steps, enabling them to share parameters and learn to generalize patterns in sequences. This parameter sharing is a key aspect that allows RNNs to handle sequences of varying lengths.

Ability to process sequences of variable lengths represents the strengths of RNNs. RNNs can adapt to different lengths by dynamically updating their hidden states through time.

Limitations of RNNs could be formulated as the difficulty in capturing of long-term dependencies. However, advanced architectures like Long Short-Term Memory (LSTM) [27] and Gated Recurrent Unit (GRU) [28] can improve the modeling of longer-range dependencies in sequential data.

Modular Neural Networks. MNNs involve structuring the neural network architecture into modular components which serve a specific function. To create more complex network architecture these modules can be combined into traditional leyered structures. This design allows improving the interpretability, generalization, and efficiency of neural networks [29].

Main behaviour of RNNs could be formulated as follow: presence of domain-specific modules; high level of scalability and adaptability; interchangeable components; robustness and fault tolerance.

Depending on the specification of tasks, modular networks can be enriched by domain-specific modules (specialized modules) designed to handle particular types of data.

Due to modularity, high level of scalability and adaptable architecture is present. Networks can be scaled up by adding more modules, and modifications can be made by replacing or tweaking specific modules.

In modular networks, the individual modules are designed to be interchangeable. Due to replacing or relinking of different modules, various temporary configurations of entire network could be simulated when without redesigning the total network. This feature is very important for testing purposes [30].

The robustness and fault tolerance of neural networks could be essentialy enhanced due to presence of modularity. In case of failing of specific module, it can be adjusted without affecting the entire system. This makes modular networks more resilient to changes and uncertainties.

Variations of ANNs such as Artificial Nonmonotonic Neural Networks (ANNNs) are presented by Boutsinas et al [19]. ANNNs are a kind of hybrid learning systems that provide explanation to the trained neural networks for following goals: a) acquiring symbolic knowledge about a domain, b) improving that knowledge using a set of classifications examples, c) extracting comprehensible symbolic information.

2. Perceptron

Perceptrons are the building blocks of ANNs, and they serve as the fundamental unit for information processing. The essence of a perceptron lies in its simplicity and its ability to make binary decisions based on weighted inputs. A perceptron takes multiple binary inputs (0;1) and produces a single binary output. Each input is associated with a weight that represents the importance of that input. The perceptron computes a weighted sum of its inputs, and this sum is passed through an activation function. The most common activation function for a perceptron is a step function, where the output is 1 if the weighted sum is above a certain threshold and 0 otherwise. Learning rules for perceptrons adjust the weights based on errors in classification. The perceptron learning algorithm helps the perceptron learn the optimal weights to improve its performance on a given task.

Two types of perceptrons are significant for ANN: single-layer perceptron (SLP) and multi-layer perceptron (MLP). SLP consists of only one layer of perceptrons. For linearly separable patterns, an effective SLP learning procedure allows solving the simple not complicated tasks (predictions with high probability). MLPs consist of several hidden layers of perceptrons. Presence of multiple hidden layers allows MLPs to learn complex, non-linear patterns. Modern neural networks are constructed using MLPs. However, Gómez-Ramos et al [6] showed that the MLP has important delimitations in initialization weights.

Transfer function f_j is nonlinear, such as a heaviside, sigmoid, or Gaussian function. Heaviside step function (unit step function) usually denoted by H is a function, the value of which is zero for negative arguments and one for positive arguments:

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{6}$$

Characteristic S-shaped curve or sigmoid curve could be generated using logistic function $\sigma(x)$

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} = 1 - \sigma(-x) \tag{7}$$

or hyperbolic tangent (shifted and scaled version of the logistic function)

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{8}$$

Activation functions introduce non-linearities to the model, allowing it to capture complex relationships in the data. Common activation functions include sigmoid, hyperbolic tangent, and rectified linear unit (ReLU) [40]:

$$f(x) = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x, & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Complete classification of activation functions was described by Jain et al [13]. Nejad [12] presented expanded list of common actication functions such as exponential, hyperbolic, sine, softmax..

3. Learning procedure

The operation of ANNs is divided into two stages: learning (training) and generalization (prediction, recalling). The training of the network should be done using prepared examples such as a set of input parameters and a set of output parameters associated with multiple events. After learning procedure provided using selected learning algorithm, network parameters are adapted. Network is ready for predictions.

In ANNs, learning procedure represents modification and adjustments of the weights of links between the nodes of a graph. Learning in ANN can be classified into three categories: supervised learning, unsupervised learning, and reinforcement learning.

Feedforward neural networks (FNNs) are trained using supervised learning, where the model is provided with input-output pairs, and the weights are adjusted iteratively to minimize the difference between predicted and actual outputs. Wang [4] accented importance of training improvement because ANN will get better results with the progress of training. The backpropagation (BP) algorithm is commonly used for training. It calculates the gradient of the error with respect to the weights and adjusts the weights to minimize the error. Basheera et al [12] presented related learning rules with special emphasis on BP of ANNs. Perceptron learning algorithms were described by Jain et al [14].

Two big learning problems exist: large amounts of data and overfitting in training. Tasks of FNNs often require a large amount of labeled data to generalize well. Training deep FNNs on small datasets can lead to overfitting, where the model performs well on the training data but poorly on unseen data. Deep FNNs with many parameters can be overfitted, especially when the training data set is limited. Adjustment techniques such as dropout or weight decay are commonly used to partial solution of this problem [31].

Limitations of learning procedure for FNNs are the following: limited interpretability and computational intensity. Understanding the decision-making process of FNNs can be complex and trivial. For deep neural networks, including FNNs, realisation is presented in form of black box. It is difficult to understand why a particular prediction was made, restrict their application to areas where interpretability is critical. Training procedure of deep FNNs can be computationally intensive, requiring large resources both processing time and computing power. If resources are incomplete this can be a serious limitation for applications.

Training of RNNs involves a variation of the backpropagation algorithm titled as Backpropagation Through Time (BPTT) [32-33]. BPTT is used to update the network's parameters by considering the unfolding of the network over time, effectively treating the sequence as an unfolded computational graph. For RNNs, training procedure could be to slow down due to vanishing and exploding gradient problems. These problems arise when the gradients become too small, resulting in slow learning or ineffective updates (disappearing), or too large causing digital instability (explosion). Werbos [32] reviewed BPTT method for pattern recognition and fault diagnosis. For speech recognition and understanding, Ismail [33] proposed a new type of RRN, which contains each output unit connected to itself and is also fully connected other output devices and all hidden units. This modification of RRN was used in parallel with BPTT was assessed as promising.

The basic equations for backpropagation through time, and applications to areas like pattern recognition involving dynamic systems, systems identification, and control are discussed. Further extensions of this method, to deal with systems other than neural networks, systems involving simultaneous equations, or true recurrent networks, and other practical issues arising with the method are described. Pseudocode is provided to clarify the algorithms. The chain rule for ordered derivatives-the theorem which underlies backpropagation-is briefly discussed. The focus is on designing a simpler version of backpropagation which can be translated into computer code and applied directly by neutral network users

For MNNs, improved training and transfer learning could be organised in modular networks in comparison to others. Training of this type allows to reuse these modules in different contexts or combine them for transfer learning. This is especially useful when working on related tasks.

Farizawani et al [34] described the Conjugate Gradient (CG) technique as one of the most popular optimization practices used in ANN. Nowadays, CG technique allows improving the learning algorithm.

Training strategy sometimes could be titled asnon adequite. Walczak et al [11] describes the problem of ANN architecture when exact or approximate number of hidden units is required to model The approach consists of three steps.
1. Initially create a architecture of ANN with a very small or very large number of hidden units.
2. Train the network for some predetermined number of epochs.
3. Evaluate the error of the output nodes.

Singh et al [35] analyses the behaviour of training data. Longer time series of training samples will contain more events of different types, and hence, the generalization ability of the ANN will improve. However, for long time series, if considerable repetition of the same type of information is present, the ANN may not become "wiser", and one may be just wasting computational effort and time.

4. Predictions using ANNs

Basheera et al [12] represents history of the evolution of neurocomputing and its relation to the field of neurobiology.

Finance. Nejad [13] presented applications of ANNS by following categories: prediction, classification, data associacion, data conceptualization, data filtering. Jaiswal et al [15] used backpropagation neural network for predicting future prices by modifying these techniques as per requirements.

Classification. Nejad [12] emphasized pattern recognition as the main application areas of biological and artificial neural networks. Petridis et al [36] presented an recursive classifier of Incremental Credit Assignment (ICRA) type appropriate for online time series classifications.

Natural language processing. Feedforward neural networks (FNNs) are versatile and applied in various domains, including image and speech recognition, natural language processing [5], and regression tasks. For convolutional neural networks (CNNs) and recurrent neural networks (RNNs), FNNs are the building blocks for more complex architectures. RNNs find applications in tasks involving natural language processing (NLP), time series analysis, speech recognition, and other sequential data problems [4].

Education. Usage of ANN for educational purposes [37] shows several advantages over human teachers. ANN based education is more attractive for online learning. AI system dedicated personally allows adapting the teaching speed and satisfaction of individual needs (in context of subject). Scalability looks as the first prerogative. Availability at any time and any place looks as the second prerogative.

When learning a foreign language, language practice is very important. Speaking a foreign language with a robot can be more convenient for the beginning speaker. Implementing AI in education systems reduces the student's shame of the initial speach failure.

Autonomious education agents work best on constrained topics like math and programming where good or satisfied answers can be easily identified. Such agents will not be able to appreciate beauty poem or rate the novelty.

Industry. Bakas et al [38] presented the study which explores the use of ANNs on a heat transfer application. Yildirim et al [17] presented the study which compares non-linear regression and ANN models in predicting 10 yarn properties shaped by the influence of winding speed, quenching air temperature and/or quenching air speed during production. A multilayer perceptron ANN model was created by training 81 patterns using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm described in Ref. [39]. The hyperbolic tangent, or $\tanh$, activation function and logistic activation functions were used for the hidden and output layers respectively. ANN simultaneously predicted all of the 10 final properties of a yarn: tensile strength, tensile strain, draw force, crystallinity ratio, dye uptake based on the colour strengths, brightness, boiling shrinkage and yarn evenness, ANN predictions are more accurately than the non-linear regression model. Ali [39] described the BFGS method for large-scale optimization problems. More et al [15] presented ANN techniques to optimise and forecast mine water treatment plant parameters.

Biology, medicine. Basheera et al [12] described several practical application, where ANNs were used to model the microbial growth curves. The developed model was reasonably accurate in simulating both training and test time-dependent growth curves as affected by temperature and pH. Yacine et al [18] described a novel ANN approach based on an Adaptive Riemannian Kernel (ARK-ANN), to classify Electroencephalographic (EEG) motor imaging signals in the context of Brain Computer Interface (BCI). A multilayer perceptron is used to classify the covariance matrices of Motor Imagery (MI) signals employing an adaptive optimization of the testing set. Increasing of precision by 8.2% in comparison to the SVM based method was fixed.

OCR, vision. Possible solutions of complicated tasks related to the Optical character recognition (OCR) were described by Jain et al [14]. In 1969, Fukushima [40] described the pioneering realization of visual feature extracting network for handwritten character recognition. The design of the network was suggested by visual systems of cat. The network is composed of analog threshold elements connected in layers. Each analog threshold element receives inputs from a large number of elements in the neighboring layers and performs its own special functions. This successfully implemented method of information processing based on biological analogues inspired further constructions of complex systems.

Conclusions

1. Artificial Neural Networks (ANN) currently implementing Feedforward Neural Network (FNN) and Recurrent Neural Network (RNN) are very useful for applied tasks related to OCR, speech recognition, classification, stock market prediction, etc. This versatility is achieved due to FNN's ability to approximate any continuous function under certain conditions.
2. ANN learning procedure was demonstrated using several cuts: problems related to ANN architecture and perceptron dynamics, interpretation problems of overfitting and possible methods to improve this situation.
3. The advantages of prediction using ANNs have been demonstrated in several popular fields such as adaptive educology, classification in medicine and biology, industry, etc.

Abbreviations

AI	-	Artificial Intelligence
ARK	-	Adaptive Riemannian Kernel
ANN	-	Artificial Neural Network
ANNN	-	Artificial Nonmonotonic Neural Network
BCI	-	Brain Computer Interface
BP	-	Backpropagation
BPTT	-	Backpropagation Through Time
BFGS	-	Broyden-Fletcher-Goldfarb-Shanno
CG	-	Conjugate Gradient
CNN	-	Convolutional Neural Network
EEG	-	Electroencephalographic
FNN	-	Feedforward Neural Network
GRU	-	Gated Recurrent Unit
ICRA	-	Incremental Credit Assignment
LSTM	-	Long Short-Term Memory
ML	-	Machine Learning
MLP	-	Multi-Layer Perceptron
MNN	-	Modular Neural Network
NLP	-	Natural Language Processing
OCR	-	Optical character recognition
RNN	-	Recurrent Neural Network
SVM	-	Support Vector Machine

References

1. Soori, M.; Arezoo, B.; Dastres, R. (2023) Artificial Intelligence, Machine Learning and Deep Learning in Advanced Robotics, A Review. - *Cognitive Robotics* 3(2023)54-70. - https://doi.org/10.1016/j.cogr.2023.04.001.
2. Bartneck, C.; Lütge, C.; Wagner, A.; Welsh, S. (2021) Capter 2. What Is AI? In: An Introduction to Ethics in Robotics and AI - Springer - https://doi.org/10.1007/978-3-030-51110-4
3. Badue, C.; Guidolini, R.; Carneiro, R.; Azevedo, P.; Cardoso, V.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.; Mutz, T.; Veronese, L.; Oliveira-Santos, T.; De Souza, A. (2021) Self-driving cars: A survey - *Expert Systems with Applications* - 165(2021)113816 - https://doi.org/10.1016/j.eswa.2020.113816.
4. Wang, X. (2018) Overview of the Research Status on Artificial Neural Networks - *Advances in Social Science, Education and Humanities Research (ASSEHR)* 130(2018)351-356.
5. Sodhar, I.; Buller, A. (2020). Natural Language Processing: Applications, Techniques and Challenges. - In: Adavances In Computer Science - AkiNik Publications New Delhi, 2020. - https://doi.org/10.22271/ed.book.784-.
6. Gómez-Ramos, E.; Venegas-Martínez, F. (2013) A Review of Artificial Neural Networks: How Well Do They Perform in Forecasting Time Series? - *Analitika, Revista de analisis estadistico* 3 (2013)7-15.

7. Ahmed, R. (2017) Applications of Artificial Neural Networks in E-Learning Personalization - *International Journal of Computer Applications* 158 (2017) 2.
8. Zhou, Y.; Huang, C.; Hu, Q.; Zhu, J.; Tang, Y. (2018) Personalized learning full-path recommendation model based on LSTM neural networks - *Information Sciences* 444(2018) 135-152 - https://doi.org/10.1016/j.ins.2018.02.053.
9. Zapalska, A.; Brozik, D. (2006) Learning styles and online education - *Campus-Wide Information Systems* 23(2006) 325-335 - https://doi.org/10.1108/10650740610714080.
10. Larsen, J. (1999) Introduction to Artificial Neural Networks. - Lingby: Technical university of Denmark.
11. Walczak, S.; Cerpa, N. (2003). Artificial Neural Networks. -631-645 In: Encyclopedia of Physical Science and Technology - https://doi.org/10.1016/B0-12-227410-5/00837-1.
12. Basheera, I.A.; Hajmeerb, M. (2000) Artificial neural networks: fundamentals, computing, design, and application - *Journal of Microbiological Methods* 43(2000)3–31.
13. Nejad, A. (2020) Evolutionary Models for Adaptive Artificial Neural Networks in Accounting and Finance Trends. - Academia Eductional, 2020. - Series: Evolutionary Models for Adaptive ANNs - https://doi.org/10.198769124.
14. Jain, A.; Mao, J.; Mohiuddin, K. (1996) Artificial neural networks: a tutorial - *Computer* 29(1996)31-44 - https://doi.org/10.1109/2.485891.
15. Jaiswal, J.; Das, R. (2018) Artificial Neural Network Algorithms based Nonlinear Data Analysis for Forecasting in the Finance Sector - *International Journal of Engineering and Technology(UAE)* 7(2018)169-176 - https://doi.org/10.14419/ijet.v7i4.10.20829.
16. More, K,; Wolkersdorfer, C. (2023) Application of machine learning algorithms for nonlinear system forecasting through analytics - A case study with mining influenced water data - *Water Resources and Industry* 29(2023)100209 - https://doi.org/10.1016/j.wri.2023.100209.
17. Yıldırım, K.;, Ogut, H.; Ulcay, Y. (2017) Comparing the Prediction Capabilities of Artificial Neural Network (ANN) and Nonlinear Regression Models in PetPoy Yarn Characteristics and Optimization of Yarn Production Conditions - *Journal of Engineered Fibers and Fabrics* 12(2017)7.
18. Yacine, F.; Salah, H.; Amar, K.; Ahmad, K. (2023) A novel ANN adaptive Riemannian-based kernel classification for motor imagery - *Biomedical Physics & Engineering Express* 9(2023) 015010 - https://dx.doi.org/10.1088/2057-1976/acaca2.
19. Boutsinas, B.; Vrahatis, M.N. (2001) Artificial nonmonotonic neural networks - *Artificial Intelligence* 132(2001)1-38 - https://doi.org/10.1016/S0004-3702(01)00126-6.
20. Kohli, S.; Miglani, S.; Rapariya, R. (2014) Basics of Artificial Neural Network - *International Journal of Computer Science and Mobile Computing* 3(2014) 745-751.
21. Bekesiene, S.; Smaliukiene, R.; Vaicaitiene, R. (2021) Using Artificial Neural Networks in Predicting the Level of Stress among Military Conscripts - *Mathematics* 9(2021) 626 - https://doi.org/10.3390/math9060626.
22. Zakaria, M.; AL-Shebany, M.; Sarhan, S. (2014) Artificial Neural Network : A Brief Overview - *International Journal of Engineering Research and Applications* 4(2014)07-12
23. Kratsios, A.; Bilokopytov, E. (2020) Non-Euclidean Universal Approximation - *Advances in Neural Information Processing Systems* 33(2020).
24. Yao X. (1999) Evolving Artificial Neural Networks - *Proceedings of the IEEE* 87 (1999)1423
25. Du, K.; Swamy, M.N.S. (2014) Recurrent Neural Networks. - In: Neural Networks and Statistical Learning Springer, London, 2014. - https://doi.org/10.1007/978-1-4471-5571-3_11.
26. Rishickesh, R.; Ram Kumar, R.P.; Shahina, A.; Nayeemullah Khan, A. (2019) Identification of Duplication in Questions Posedon Knowledge Sharing Platform Quora using Machine Learning Techniques - *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 8(2019)2444 - https://doi.org/10.35940/ijitee.
27. Jurafsky, D.; Martin, J. (2023) Chapter 9. RNNs and LSTMs. - In: Speech and Language Processing. - https://web.stanford.edu/ jurafsky/slp3/.
28. Abumohsen, M.; Owda, A.; Owda, M. (2023) Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms - *Energies* 16(2023) 2283 - https://doi.org/10.3390/en16052283.
29. Auda, G.; Kamel, D. (1999) Modular neural networks: A survey - *International Journal of Neural Systems* 9(1999)129-151.
30. Shukla, A.; Tiwari, R.; Kala, R. (2010) Modular Neural Networks. - In: Towards Hybrid and Adaptive Computing. Studies in Computational Intelligence, vol 307. - Springer, Berlin, Heidelberg. - https://doi.org/10.1007/978-3-642-14344-1_14
31. Zhu, Z.; Liu, F.; Chrysos, G.; Locatello, F.; Cevher, V. (2023) Benign Overfitting in Deep Neural Networks under Lazy Training - *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, Hawaii, USA - PMLR 202, 2023.
32. Werbos, P. (1990) Backpropagation through time: what it does and how to do it - *Proceedings of the IEEE* 78(1990)1550-1560 - https://doi.org/10.1109/5.58337.
33. Ismail, S.; Ahmad, A. (2004) Recurrent neural network with backpropagation through time algorithm for Arabic recognition - *Proceedings 18th European Simulation Multiconference*, SCS Europe.
34. Farizawani, A.G.; Puteh, M.; Marina, Y.; Rivaie, A. (2020) A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches - *Journal of Physics: Conference Series* 1529(2020)022040 - https://doi.org/10.1088/1742-6596/1529/2/022040
35. Singh, S.; Sharad, J.; András, B. (2010) Training of Artificial Neural Networks using Information-Rich Data - *Hydrology* 1(2010) 40-62. - https://doi.org/10.3390/hydrology1010040.
36. Petridis, V.; Kehagias, A. (1996) A Recurrent Network Implementation of Time Series Classi
cation - *Neural Computation* 8(1996)357-372.
37. Bartneck, C.; Lütge, C.; Wagner, A.; Welsh, S. (2021) Capter 9 . Application areas of AI. - In: An Introduction to Ethics in Robotics and AI - Springer - https://doi.org/10.1007/978-3-030-51110-4
38. Bakas, I.; Kontoleon, K.J. (2021) Performance Evaluation of Artificial Neural Networks (ANN) Predicting Heat Transfer through Masonry Walls Exposed to Fire - *Applied Science* 11 (2021)11435 - https://doi.org/10.3390/app112311435.
39. Ali, M. (2021) Modified limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm for unconstrained optimization problem - *Indonesian Journal of Electrical Engineering and Computer Science* 24(2021)1027 - https://doi.org/10.11591/ijeecs.v24.i2.pp1027-1035.
40. Fukushima, K. (1969) Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements - *IEEE Transactions on Systems Science and Cybernetics* 5(1969)322-333 - https://doi.org/10.1109/TSSC.1969.300225.