Egyptian Informatics Journal xxx (xxxx) xxx

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Review

Hybridized classification algorithms for data classification applications: A review

F. Sherwani [a], B.S.K.K. Ibrahim [b], Muhammad Mujtaba Asad [c]

[a] Department of Computer Science, FAST-National University of Computer and Emerging Sciences, Karachi, Sindh, Pakistan
[b] School of Mechanical, Aerospace and Automotive, Faculty of Engineering, Environment and Computing, Coventry University, Priory St, Coventry CV1 5FB, United Kingdom
[c] Department of Education, Sukkur IBA University, Airport Road, Sukkur, Sindh, Pakistan

A R T I C L E I N F O

Article history:
Received 22 October 2019
Revised 27 February 2020
Accepted 24 July 2020
Available online xxxx

Keywords:
Recurrent Neural Networks (RNNs) algorithm
Back Propagation Neural Network (BPNN)
Deterministic algorithms
Machine learning
Levenberg-Marquardt (LM)

A B S T R A C T

Machine-based classification usually involves some computer programs, known as algorithms, developed using several mathematical formulations to accelerate the automated classification process. Along with the exponential increase in the size and computational complexity of the data today, such optimized, robust, agile and reliable computational algorithms are required which can efficiently carry out these conforming classification tasks. In this review paper, deterministic optimization techniques have been analysed that are efficiently employed for machine learning applications. In this review, systematic literature review approach has been adopted in which 200 research articles were downloaded from which 100 latest articles has been selected based on the most commonly employed neural networks' techniques. Moreover, the reported neural networks techniques based on Back Propagation Neural Network (BPNN), Recurrent Neural Networks (RNNs) Algorithm and Levenberg-Marquardt (LM) with several hybridized classification algorithms based on optimization techniques have been indicated that are commonly used to optimize and benefit the classification process.

Contents

1. Introduction

Machine Learning (ML) is a sub-branch of Artificial Intelligence (AI) that deals with the interpretation of raw data into meaningful information [1,2]. With the humongous growth in the volume of data available today, it is required to design and implement such compliant data classification techniques that can tackle and facili-

E-mail address: fahad.sherwani@nu.edu.pk (M.M. Asad)
Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2020.07.004
1110-8665/© 2020 Published by Elsevier B.V.

tate efficient and competent data analysis, accordingly. Some of the most common existing techniques that are being employed to implement the aforementioned systems include Support Vector Machines (SVMs), Artificial Neural Networks (ANNs) etc [3–5]. Most of the advanced classification techniques are combined or hybridized with optimization methods today to further improve the classification accuracy and reduce the computational time in such classification systems [6,7].

Topologically, exploration of the optimal solution to any given problem in a computational scenario is similar to searching for a tiny pin in a dark room [1,2]. While, this search can be highly probabilistic and significant, it may also be computationally very expensive as well [8,9]. However, if it is known that the pin may lie in some of the specific areas in that room, it may very well decrease the overall computational cost and increase the probability to find the object [10,11]. This scenario complies with the gradient ascent or gradient descent techniques. Such techniques allow the computational algorithm to identify the potential search areas and the search is conducted in those areas to ensure symmetrical results during each run [3,4]. BPNN, ERNN and LM algorithms are some of the most popular gradient descent techniques used in neural networks.

2. The Back-propagation (BP) neural networks algorithm

Back-propagation (BP) Neural Networks algorithm may be referred to as one of the widely employed prevalent optimization algorithm that is applied on Neural Networks (NNs) aimed at expediting the convergence of neural network to global optimum [8–11]. BPNN inherits the fundamental principles of Artificial Neural Networks (ANNs) that follow the learning process and cognition skills of humans [12–15].

The structure of BPNN algorithm is kindred to that of ANNs which includes of one layer of input neurons, single or multiple hidden layers and one output layer [10,14,16]. It is structured on a fully connected architecture where every node in one layer is connected to all the other nodes in the succeeding layers as depicted in Fig. 1. BP-based neural networks learn by back propagating the errors in the output layers in order to measure the errors in the hidden layers.

BPNN has been employed widely in variety of applications because of its highly elastic nature and learning abilities [6,10,14,17]. Principally, the fundamental purpose of the process of learning is to reduce the difference between the predicted output O_k and the actual output t_k through the optimization of the

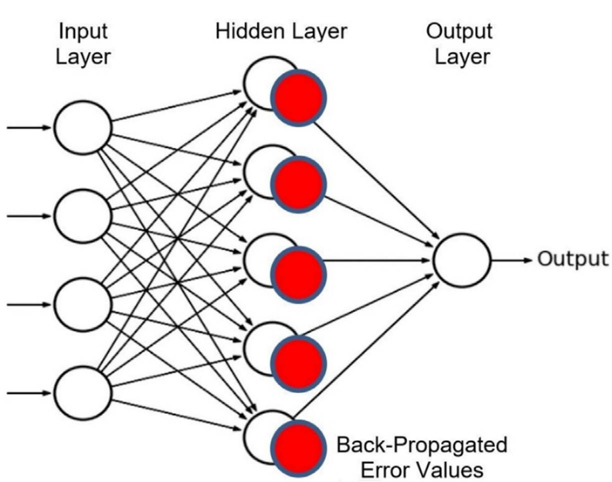

Fig. 1. Basic structure of back-propagation neural networks.

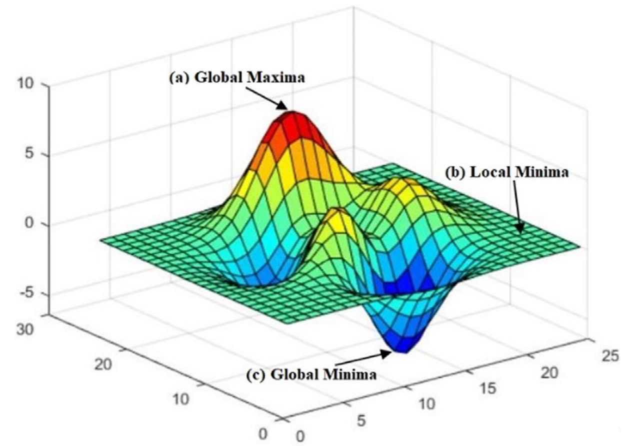

Fig. 2. 3D visualization of error function.

weights w^* in the network. The Error function is defined as [7,18–20].

$$E = \sum_{k-1}^{n} (t_k - O_k)^2 \quad (1)$$

where;n: Output Neuronst_k: Desired k^{th} outputO_k: Actual k^{th} output

Fig. 2 demonstrates the 3D visualization of the error function in weight space. As shown in Fig. 2, the point (a) at which the error function is the highest is called as the global maxima, point (c) at which the error function is the smallest is known as the global minima, whereas, every other minima is referred to as local minima.

The error function can be defined as a non-linear function of weights in networks within excess of one layer and it may possess numerous local minima. The gradient of total error function E relative to weights can be denoted by using the following equation:

$$\nabla E(\omega) = 0 \quad (2)$$

Eq. (2) can be used to calculate the network error by equating the difference of actual output of network against the anticipated output [21–23]. The error is propagated back through the network as a reference, after it is computed, to perform the weight adjustment [24–26]. This process continues until the maximum epoch or the target error is achieved by the network [26–29]. Although, BPNN employs local learning gradient descent technique, nonetheless it is prone to problems like slow learning or even network stagnancy.

Hence, initial parameters such as weights and biases, network topology, activation function, learning rate and momentum coefficient are very crucial to decide and should be calculated accordingly [30–32]. These parameters can cause the network convergence to slow down and can also create network stagnancy if they are not selected appropriately [33–35]. Numerous techniques and concepts have been proposed and reported in literature in order to overcome the aforementioned issues. Some of the literature related to BPNN is highlighted in the Table 1.

The appropriate selection of initial values of weights is crucial for BPNN algorithm to perform better and introduce agility in the network convergence towards global optimum [36–38]. Momentum coefficient is another BPNN parameter which is used to minimize the oscillations in the trajectory by inducing a portion of the preceding change in weight [39–41]. The inclusion of the coefficient of momentum helps to regulate the path of gradient descent by avoiding changes that are caused due to local irregularities [42–44]. Hence, it is vital to minimize any variations that are produced because of the error surface [45]. In the early 90's, Back-

Table 1
Brief literature on back-propagation neural networks.

Year	Author(s)	Contributions
1989	Lari-Najafi et al.,	Indicated the use of large initial weights for increasing the learning rate of the BPNN network.
1990	Kolen & Pollack	Proved the sensitivity of BPNN to initial weights and suggested the use of weights initialized with small random values.
1992	Qiu et al.,	Suggested several adaptive momentum techniques methods to overcome static momentum problem, such as momentum step and dynamic selection of momentum.
1994	Swanston, Bishop & Mitchell	Proposed Simple Adaptive Momentum (SAM) for further improving the performance of BP-neural networks.
1996	Thimm, Moerland, & Fiesler	Relationship between learning rate, momentum, and activation function was mapped.
2008	Mitchell	Improved SAM by scaling the momentum after considering all the weights in each part of the Multi-layer Perceptron (MLP). This method has been observed to improve convergence speed to the global minima (Mitchell, 2008).
2009	Shao & Zheng	Introduced a new Back Propagation momentum Algorithm (BPAM) with dynamic momentum coefficient.
2001	Ye	Ye claimed that the constant learning is unable to answer the search for the optimal weights resulting in the blind search.
2002	Yu & Liu	Presented back propagation and acceleration learning algorithm (BPALM) with adaptive momentum and learning rate.
2007	Nawi	Proposed adaptive leaning rate with adaptive momentum and adaptive gain parameter.
2012	Hamid	Suggested novel adaptive leaning rate and adaptive momentum techniques to speed-up the convergence rate in conventional BP-based neural networks.
2015	Wang et al.,	Proposed a hybridized BPNN with a combination of Adaptive Differential Evolution (ADE) Algorithm to address the issue of falling in local minima in conventional BPNN by applying ADE to optimize initial weights and thresholds of BPNN algorithm, followed by a thorough search for the optimal weights and thresholds by BPNN.
2018	Zhaoyang ye & Moon Keun Kim	Proposed a combination of the Levenberg–Marquardt and back-propagation (LM-BP) neural network by combining the gradient descent and Quasi-Newton method to enhance the accuracy of predictions, resulting in agile convergence speed and improved overall performance. It is achieved through adjustment between the steepest gradient descent method and the Gauss–Newton method in an adaptive manner to optimize the weights in order to ensure effective network convergence.

Ppropagation with Fixed Momentum (BPFM) showed its ability in convergence to global minima but later it was found that BPFM performs when the error gradient and the last change in weights are in parallel, which leads towards the network stagnancy or even failure. Hence, it was concluded and recommended that there should be a phenomenon of adaptiveness in the momentum coefficient feature [46,47]. In accordance to the above findings, various approaches to induce adaptive momentum were proposed in the previous literature such as momentum step and dynamic selection of the momentum rate to overcome the fixed momentum problem [48,49]. Swanston, Bishop and Mitchell proposed Simple Adaptive Momentum (SAM) for further improving the performance of BPNN [50] in 1994. In this technique, the momentum term is increased to accelerate the convergence if the change in the weights is in the same direction and vice versa. SAM improved the overall performance of conventional BP algorithm considerably where it costs lower computational overheads and converged in significantly less iterations. Far along the way in 2008, Mitchell updated SAM by scaling the momentum after analyzing all the weights in each part of Multi-layered Perceptron (MLP) networks. This method was found to be helpful in improving the speed of convergence to the global minima [51,52]. Besides momentum, another parameter that greatly affects the performance of BPNN is the learning rate. In the earlier studies, the usual value of learning rate was kept constant. In 2001, Ye claimed that the constant learning is unable to answer the search for the optimal weights resulting in the blind search [53,54]. To avoid more trials and errors with the network training, Yu and Liu (2002) introduced back propagation and acceleration learning method (BPALM) with adaptive momentum and learning rate to answer the problem of fixed learning rate [55,56]. More recently, Hamid introduced adaptive momentum and leaning rate to accelerate the convergence in conventional BPNN. After the experimentation process, it was concluded that too little learning rate can slow down the network convergence while too big learning rate can lead the network towards less optimal solutions. So, a learning rate should be selected very carefully to make the network perform efficiently [57,58]. Besides other factors effecting the performance of BPNN, an activation function represents an output node that is showing some synapses or nothing at all. Its basic function is to limit the range of the output neuron. It generates an output value for a node in a predefined range as the closed unit interval [0, 1] or alternatively [−1, 1] which can be a linear or non-linear function [17]. In this study, the logistic sigmoid activation function is used which limits the amplitude of the output in the range of [0, 1]. The activation function for the *jth* node is given in the Equation below;

$$a_{net,j} = \sum_{i=1}^{1} w_{ij} O_i + \theta_j \tag{3}$$

$$O_j = \frac{1}{1 + e^{-c_j a_{net,j}}} \tag{4}$$

O_j: output of the jth unit.O_i: output of the ith unit.w_{ij}: link weight from unit i to unit j.$a_{net,j}$: net input activation function for the jth unit.θ_j: bias for the jth unit.c_j: gain of the activation function.

In earlier studies the value for gain parameter in the activation function was reserved fixed. But later on, it was realized that the gain parameter can greatly influence the slope of the activation function. In 1996, a relationship between learning rate, momentum, and activation function was mapped [57]. In their findings, it was indicted that rate of learning and the gain of the activation function are exchangeable and better results can be obtained with the variable gain parameter. Thimm's theory of changing the gain of the activation is equivalent to learning rate, and momentum is further verified by Edom, Jung and Sirisena in 2003, when they automatically tuned gain parameter with the fuzzy logic. [58,59] used the adaptive gain parameter in back propagation with conjugate gradient method. Further, the previous approach and suggested an adaptive gain parameter improved based on adaptive momentum and adaptive leaning rate. The proposed Back Propagation Gradient Descent with Adaptive gain, adaptive momentum, and adaptive learning (BPGD-AGAMAL) algorithm showed significant enhancement in the performance of BPNN on classification datasets [60–62]. Despite inheriting the most stable multi-layered architecture, BPNN algorithm is not suitable for dealing with the temporal datasets due to its static mapping routine [63,64]. In-order to use a temporal dataset on BPNN, small dimensions of the pattern vectors must be equal otherwise BPNN is rendered useless. [36] proposed a hybridized BPNN with a combination of Adaptive Differential Evolution (ADE) Algorithm and named it as ADE-BPNN. This algorithm addressed the issue

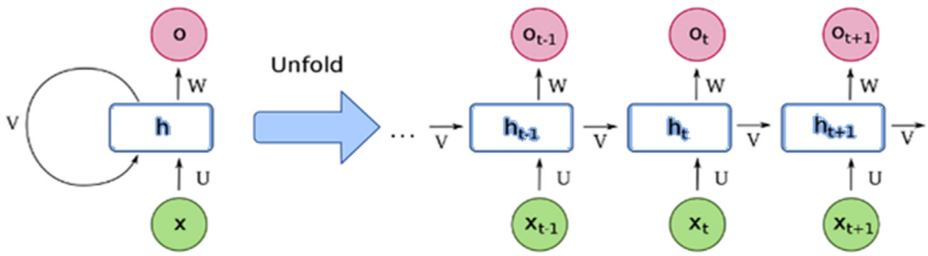

Fig. 3. Recurrent neural network architecture.

of falling into the local minima in conventional BPNN by applying ADE to optimize the global initial connection weights and thresholds of BPNN, followed by BPNN to thoroughly search for the optimal weights and thresholds. [65–67] proposed a combination of the Levenberg–Marquardt and back-propagation (LM-BP) neural network by combining the gradient descent and Quasi-Newton method to enhance the accuracy of predictions, resulting in agile convergence speed and improved overall performance. It is achieved through adaptive adjustment between the steepest gradient descent method and the Gauss–Newton method to optimize the weights in order to ensure effective network convergence.

3. The recurrent neural networks (RNNs) algorithm

Recurrent Neural Networks (RNNs) algorithm can be referred to as a subclass of ANNs which is structured based directed graph sequence architecture along the nodes as shown in Fig. 3. RNNs are known for their ability to exhibit dynamic temporal behaviour for time sequences. RNNs can map both temporal and spatial datasets and has short term memory to remember the past event thus highly influencing the output vectors [38,68]. It means that, RNNs possess the capability to store previous changes made to any node in the network layers, which can then be utilized later. Due to this learning elasticity, RNN have been deployed in several fields such as; simple sequence recognition, Turing machine learning, pattern recognition, forecasting, optimization, image processing, and language parsing etc [39,69]. In the earlier years of ANN's inception, fully recurrent neural networks were quite popular. Usually, RNNs are classified as fully recurrent or partially recurrent based on their functionalities.

Some of the examples are back propagation through time (BPTT) and Recurrent back propagation (RBP). The fundamental working principle of BPTT is that of unfolding [70] it is a training method for fully recurrent network which allows back propagation to train an unfolded feed-forward non-recurrent version of the original network. Once trained, the weights from any layer of the unfolded network are passed onto the recurrent network for temporal training [71]. BPTT is quite inefficient in training long sequences. Also, error deltas make a big change for each weight after they are folded back requiring a greater memory requirement. If, a larger time step is used, it diminishes the error effect called vanishing gradient thus making it totally infeasible to be applied on any dataset [72]. Unlike BPTT, Recurrent Back Propagation (RBP) possesses similarities to the master or slave network of Lapedes and Farber, the difference being it simple in terms of architecture [27]. In RBP network, the back propagation is protracted directly to train fully recurrent neural network. In this method, all the units are assumed to have continually evolving states [73]. Pineda (1987) used RBP on temporal XOR with 200 patterns and found it to consume a lot of time.

Also, BPTT and RBP are off linen training methods and not suitable for long sequences due to more time consumption. In 1989, Williams used online training of RNN in which the weights are updated while the network is running, and the error is minimalized after every time step rather than at the completion of the sequence. Hence, recurrent networks are able to learn those tasks that involve information retention over fixed or indefinite duration of time periods [74].

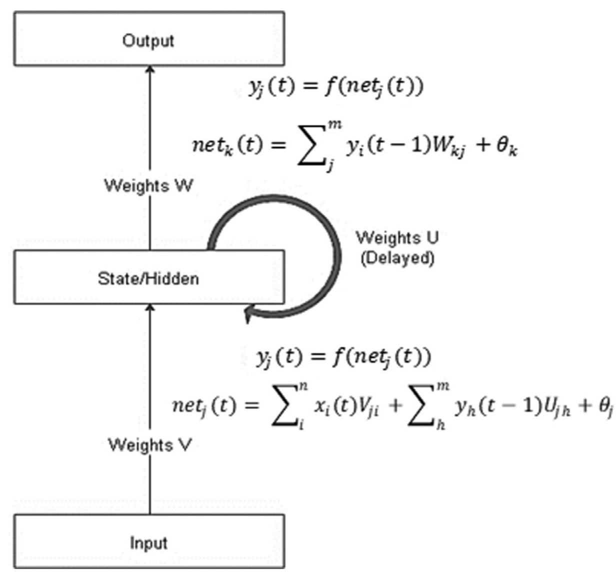

Fig. 4. An Elman recurrent neural network [35].

In partial recurrent neural network, recurrence in feed forward neural network is produced by feeding back the network outputs as additional input units [10] or delayed hidden unit outputs [75]. An Elman Recurrent Neural Network (ERNN) is a relatively simple structure proposed by Elman to train a network whose connections are largely feedforward with a careful selection of feedback context layer's units to hidden units. The context layer nodes store the previous inputs to hidden layer's nodes. The context values are used as extra inputs to hidden layers, resulting in an open memory of one-time delay [76]. Three layered ERNN structure is used in this research as given in the Fig. 4. In ERNN, each layer has its own index variable:

$$net(t) = \sum_i^n x_i(t)V_{ji} + \theta_i \tag{5}$$

Where, n is the number of inputs, and θj is the bias. In an ERNN, the input vector is spread in a similar manner like feed-forward networks that propagates through each layer with some weights. Whereas, the input vector is combined with the previous state activation in RNN, through an additional recurrent weight layer, U;

$$y_j(t) = f(net_j(t)) \tag{6}$$

$$net_j(t) = \sum_i^n x_i(t)V_{ji} + \sum_h^m y_h(t-1)U_{jh} + \theta_j \tag{7}$$

Where, f denotes the network output function and m denoted the number of states. The network output is attained through the current state and the output weights, W;

$$y_k(t) = g(net_k(t)) \tag{8}$$

$$net_k(t) = \sum_j^m y_i(t-1)W_{kj} + \theta_k \tag{9}$$

Where, g represents the network output function and W_{kj} denotes the weights from the hidden layer k to output layer j.

During early 1990′s, ERNN has been found to have a sufficient generalization capability and has successfully predicted the stock points in Tokyo stock exchange. ERNN also takes advantage of the parallel hardware architecture, and it has shown faster capability to learn complex patterns such as natural language processing [19]. In medical field, it is found beneficial in dynamic mapping of the electroencephalographic (EEG) signals classification with high accuracy during clinical trials [77].

Later, a similar ERNN technique was used for Doppler ultrasound signal classification using Lyapunov exponents and again high accuracy was achieved [78]. Based on the optimization provided by ERNN, (Xing, 2015) has recently applied ERNN to solve real time price estimation problems in the power grid with great success [79]. Despite all these achievements ERNN algorithms face the initial weight dilemma and gets stuck in local minima or slow convergence. In-order to avoid local minima and slow convergence in ANN, a second order derivative based Levenberg-Marquardt (LM) algorithm was introduced [80]. [9] proposed a hybridized Elman neural network with a combination of Genetic Algorithm (GA) for forecasting energy consumption in public buildings. GA was used to optimize the weight matrix within ERNN. It was concluded that the proposed technique provided better results in terms of optimization of the energy consumption forecasting. Table 2 enlists the literature related to ERNN.

Table 2
Brief literature on Elman recurrent neural networks.

Year	Author(s)	Contributions
1987	Pineda	Pineda used RBP on temporal XOR with 200 patterns and found it to consume a lot of time.
1989	Williams	In 1989, Williams used online training of RNN in which the weights are updated while the network is running, and the error is minimized after every time step rather than at the completion of the sequence.
1990	Elman	An ERNN network is a relatively simple structure proposed by Elman to train a network whose connections are largely feedforward with a careful selection of feedback context layer's units to hidden units.
1990	Kamijo and Tanigawa	ERNN has been found to have a sufficient generalization capability and has successfully predicted the stock points in Tokyo stock exchange.
1991	Jordan et al.,	Feed forward neural network is produced by feeding back the network outputs as additional input units.
2008	Übeyli	ERNN technique was used for Doppler ultrasound signal classification using Lyapunov exponents and again high accuracy was achieved.
2014	Cho et al.,	Representations of learning phrase using RNN encoder-decoder for statistical machine translation.
2015	He et al.,	Xing (2015) has recently applied ERNN to solve real time price estimation problems in the power grid with great success.
2018	L.G.B. Ruiz et al.,	Proposed a hybridized Elman neural network with a combination of Genetic Algorithm (GA) for forecasting energy consumption in public buildings. GA was used to optimize the weight matrix within ERNN.

4. The Levenberg-Marquardt (LM) neural networks algorithm

BPNN algorithm is commonly known as the steepest descent method but it also suffers from slow convergence problems. Therefore, an intermediary algorithm that utilizes the gradient descent and Gauss-Newton (GN) methods is introduced. This algorithm is known as Levenberg-Marquardt (LM) which is comparatively more robust than the GN technique due to its capability to converge even in highly complex optimization problems [81]. The fundamental working principle of LM algorithm is that it keeps shifting to the steepest descent algorithm while waiting for the proper local curvature in order to compute a quadratic approximation. After that, it transforms into a very similar form of the Gauss–Newton algorithm to accelerate the convergence to global minima, significantly [82]. LM uses Hessian matrix for approximation of error surface. Assume that the error function is:

$$E(t) = \frac{1}{2}\sum_{i=1}^{N} e_i^2(t) \tag{10}$$

$\nabla(t) = l^T(t)e(t)(11)$

$\nabla^2(t) = j^T(t)j(t)(12)$ where, $\nabla E\ (t)$; denotes the Gradient descent,$\nabla^2\ E\ (t)$; represents the Hessian matrix of E $(t)l\ (t)$; Symbolizes the Jacobian matrix

In GN method,

$\nabla w = -[l^T(t)J(t)]^{-1} I(t)e(t)(13)$

$$J(t) = \begin{bmatrix} \frac{\partial v_1(t)}{\partial t_1} & \frac{\partial v_1(t)}{\partial t_2} \cdots & \frac{\partial v_1(t)}{\partial t_n} \\ \frac{\partial v_2(t)}{\partial t_1} & \frac{\partial v_2(t)}{\partial t_2} \cdots & \frac{\partial v_2(t)}{\partial t_n} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{\partial v_n(t)}{\partial t_1} & \frac{\partial v_n(t)}{\partial t_2} \cdots & \frac{\partial v_n(t)}{\partial t_n} \end{bmatrix} \tag{14}$$

Whereas, for the LM algorithm as a variant of GN;

$$W(k\ +\ 1) = w(k) - \left[j^T(t)J(t) + \lambda I\right]^{-1} j(t)e(t) \tag{15}$$

Where λ is the damping factor, λ > 0 and is a constant, I is the identity matrix. The algorithm will mimic the Gauss-Newton approach for a small value of λ that ensures rapid convergence to global minima. Whereas, when parameter λ is large, the Equation (15) achieves the gradient descent (with learning rate 1/λ).

Even though, LM inherits both the speed and the stability of the Gauss-Newton and the BPNN methods, respectively. Nevertheless, conventionally, the LM algorithm stays computationally expensive because the inverse Hessian matrix needs to be computed repeatedly within a single epoch to update the network weights. Consequently, LM is generally regarded as impractical for large datasets as the computational cost of Hessian matrix inversion may render as CPU overhead. Also, the Jacobian matrix that is required to be stored for computation is P × M × N dimensional, where P represents the number of patterns, M denotes the number of outputs, and N characterized the number of weights. Hence, the cost of memory to store Jacobian matrices may well be too huge to be employed practically for applications involving datasets with large volumes [44]. In 1994, Marquardt algorithm for nonlinear least squares was presented and was later combined with the backpropagation algorithm to carry out feed-forward neural networks' training. The algorithm was tested on optimization function problems and benchmarked against the conjugate gradient algorithm and a variable learning rate algorithm. It was found during the simulations that the Marquardt algorithm proved to be more efficient than any other techniques when the network weights are limited to a few hundred [83]. In 2002, two second-order algorithms for the training of feed-forward neural networks. The Levenberg Marquardt (LM) method used for nonlinear least squares problems incorporated an additional adaptive momentum term. The simulation results on large scale datasets show that their implementation models had better success rate than the conventional LM and other gradient descent methods [84]. Later in 2005, LM algorithm implemented to determine the sensation of smell through the use of an electronic nose. Their research showed that the LM algorithm is a suitable choice for odor classification and it performs better than the old BP algorithm. [84] optimized the LM algorithm by calculating the Quasi-Hessian matrix and gradient vector directly, thus eliminating the need for storing the Jacobian matrix as it was replaced with a vector operation. The removal of Jacobian Matrix caused less memory overheads during simulations on large datasets. The simulation results found that this unconventional LM algorithm can perform better than the simple LM with less memory and CPU overheads. [85] solved the limitation of memory problem in LM training by disregarding the Jacobian matrix multiplication and storage while computing the Quasi-Hessian matrix and gradient vector directly. A novel Forward Accumulation Through Time (FATT) algorithm was presented by [86] in order to compute the Jacobian matrix multiplication with in the LM algorithm to effectively train RNNs using the LM algorithm. [89–91] introduced an improved implementation of a truncated singular value decomposition (TSVD)-based LM algorithm for generating multiple realizations of reservoir models conditioned to production data. It has been observed in the proposed research that the TSVD-based LM algorithm converges to appropriate estimates because it gradually resolves the important features of the true model. Recently, another technique to tackle the flat-spot problem in conventional LM algorithm has been proposed through the compression of neuron weights which subsequently pushes neuron activation out of saturated region towards the linear region [93,94]. This algorithm (LM-WC) avoids the training failures and ensures convergence of network through the adjustment of adaptable compression parameter. Some of the literature related to LM is highlighted in Table 3.

Table 3
Brief literature on Levenberg-Marquardt neural networks.

Year	Author(s)	Contributions
1944	Levenberg	This algorithm is known as Levenberg-Marquardt (LM) which is comparatively more robust than the GN technique due to its capability to converge even in highly complex optimization problems.
1994	Hagan & Menhaj	The Marquardt algorithm for nonlinear least squares was presented and later combined with the back-propagation algorithm for training feed-forward neural networks.
2002	Ampazis & Perantonis	Presented two algorithms of second-order for the training of feed-forward neural networks.
2005	Kermani et al.,	Implemented LM algorithm to determine the sensation of smell through the use of an electronic nose.
2007	Wilamowski et al.,	Optimized the LM algorithm by calculating the Quasi-Hessian matrix and gradient vector directly, thus eliminating the need for storing the Jacobian matrix as it was replaced with a vector operation.
2010	Wilamowski, B. M., & Yu, H.	The limitation of memory problem in LM training was solved by disregarding the Jacobian matrix multiplication and storage while computing the Quasi-Hessian matrix and gradient vector directly.
2014	Xingand Fu et al.,	A new Forward Accumulation Through Time (FATT) algorithm was presented by [31] in order to compute the Jacobian matrix multiplication with in the LM algorithm to effectively train RNNs using the LM algorithm.
2016	Shirangi & Alexandre	Introduced an improved implementation of a truncated singular value decomposition (TSVD)-based LM algorithm for generating multiple realizations of reservoir models conditioned to production data
2018	James S. Smith et al.,	Levenberg-Marquardt with weight compression (LM-WC) algorithm was proposed in this research to tackle the flat-spot issue in typical LM algorithm and to ensure early convergence of Neural Network.

5. Discussion

Artificial Intelligence (AI) and Machine Learning (ML) are one of the most active research areas today due to the higher industrial demand for in the field of control systems and robotics. Similarly, AI and ML research domains are being investigated for the provision of such improved techniques that can benefit computer science field. The applications of AI and ML range from basic robotics to highly sophisticated industries based on their technical needs. There are various fundamental approaches that form the bases of these intelligent systems which has been discussed in the review article. Whereas, optimization of each task is inevitable in such systems to facilitate the efficiency of overall operation. Moreover, the deterministic algorithms are such mathematical programs that allow a system to identify the optimal search areas within an optimization problem in order to ensure symmetrical result based on the industrial demand to perform specific information or function. Besides, the domain of data classification is the most important type of machine learning technique which deals with the classification of large, computationally and handle the complex industrial and organizational datasets. Classification of these huge datasets using existing techniques lead to higher computational times and decreased accuracy which is also one of the major challenges while handling such datasets. Recently, several hybridized classification algorithms based on optimization techniques are proposed and commonly used to optimize and benefit the classification process. Bio-inspired metaheuristic algorithms are most commonly used for such hybridized techniques because of their versatile exploration and exploitation capabilities. Therefore, this review paper has discussed major and mostly implemented and utilized deterministic optimization techniques that are employed to implement the data classification in ML applications, industrial and organization settings in this modern era of fourth industrial revolution. These techniques include BPNN, ERNN and LM algorithms which are some of the most widespread gradient descent techniques used in neural networks these days based on their nature and specific industrial demand. Whereas, the most prominent improvements that have been made to the techniques, from their conventional forms, were also discussed in this review paper. For performing an effective classification of complex datasets, it is extremely important to develop such classification techniques that can reduce the computational time and improve the classification accuracy. Therefore, to reduce the computational times and enhance classification accuracy, hybridized classification techniques using optimization algorithms have observed to be most appropriate.

6. Conclusion

This brief literature review also highlighted that the deterministic optimization based gradient descent systems are considered as effective to implement the neural network-based data classification in ML applications. Moreover, it is concluded that although, there have been many improvements in these techniques that have been proposed in previous literature, even then, there is still a lot of scope and need to further improve these techniques in order to cope up with the ever increasing supply of highly complex data these days. The prevailing issues in machine-based hybridized classification techniques limit the potential of automated classification systems in high-dimensional classification problems. In order to perform and assist efficient classification for such datasets, it is crucial to develop such classification techniques that can significantly reduce the computational times and improve the classification accuracy for such applications. Hence, to reduce the computational times in hybridized classification techniques using optimization algorithms and improve the overall classification accuracy, it is inevitable to improve the optimization capability of the traditional optimization algorithms.

Acknowledgement

This work was supported by Malaysian International Scholarship (MIS) under Ministry of Education Malaysia (MOE).

References

[1] Buchbinder N, Feldman M. Deterministic algorithms for submodular maximization problems. ACM Trans Algorith (TALG) 2018 Jun 16;14(3):32.
[2] Sergeyev YD, Kvasov DE, Mukhametzhanov MS. Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms. Math Comput Simul 2017 Nov;1(141):96–109.
[3] Socher R, Manning CD, Ng AY. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In: proceedings of the NIPS-2010 deep learning and unsupervised feature learning workshop 2010 Dec 10, Vol. 2010, p. 1–9.
[4] Chan TM. Improved deterministic algorithms for linear programming in low dimensions. ACM Trans Algorith (TALG) 2018 Jun 16;14(3):30.
[5] Hagger-Johnson G, Harron K, Goldstein H, Aldridge R, Gilbert R. Probabilistic linkage to enhance deterministic algorithms and reduce data linkage errors in hospital administrative data. J Innov Heal Informat 2017;24(2):234–46.
[6] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE conference on 2017 Jul 21, p. 3462–71, IEEE.
[7] Wang G, Guo L, Duan H. Wavelet neural network using multiple wavelet functions in target threat assessment. Sci World J 2013;2013.
[8] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323(6088):533.
[9] Chang P, Yang J, Yang J, You T. Respiratory signals prediction based on particle swarm optimization and back propagation neural networks. In: AIP conference proceedings 2017 Apr 28, Vol. 1834, No. 1, p. 040049, AIP Publishing.
[10] Nawi NM, Khan A, Rehman MZ. A New Optimized Cuckoo Search Recurrent Neural Network (CSRNN) Algorithm. In: The 8th international conference on robotic, vision, signal processing & power applications 2014, p. 335–41, Springer, Singapore.
[11] Wang B, Hobbs BF. Flexiramp market design for real-time operations: Can it approach the stochastic optimization ideal?. In: Power and Energy Society General Meeting (PES), 2013 IEEE 2013 Jul 21, p. 1–5, IEEE.
[12] Gong Q, Li Y, Peng Z. Power management of plug-in hybrid electric vehicles using neural network based trip modeling. In: American control conference, 2009. ACC'09. 2009 Jun 10, p. 4601–6), IEEE.
[13] Das D, Pratihar DK, Roy GG, Pal AR. Phenomenological model-based study on electron beam welding process, and input-output modeling using neural networks trained by back-propagation algorithm, genetic algorithms, particle swarm optimization algorithm and bat algorithm. Appl Intell 2018 Sep 1;48(9):2698–718.
[14] Nawi NM, Ghazali R, Salleh MN. Predicting patients with heart disease by using an improved back-propagation algorithm. J Comput 2011;3(2):53–8.
[15] Nawai N, Shariff MN. Loan repayment problems in microfinance programs that using individual lending approach: a qualitative analysis.
[16] Nawi NM. Computational issues in process optimisation using historical data. Swansea University (United Kingdom); 2007.
[17] Nawi NM, Ransing RS, Ransing MR. A new method to improve the gradient based search direction to enhance the computational efficiency of back propagation based Neural Network algorithms. In: Modeling & simulation, 2008. AICMS 08. Second Asia international conference on 2008 May 13, p. 546–52; IEEE.
[18] Nawi NM, Rehman MZ, Ghazali MI. Noise-induced hearing loss prediction in Malaysian industrial workers using gradient descent with adaptive momentum algorithm. Int Rev Comput Softw 2011 Sep;6(5):740–8.
[19] Abdul Hamid N. The effect of adaptive parameters on the performance of back propagation, Doctoral dissertation, Universiti Tun Hussein Onn Malaysia.
[20] Hamid NA, Nawi NM, Ghazali R, Salleh MN. Solving local minima problem in back propagation algorithm using adaptive gain, adaptive momentum and adaptive learning rate on classification problems. In: International journal of modern physics: conference series 2012, Vol. 9, p. 448–55); World Scientific Publishing Company.
[21] Fkirin MA, Badwai SM, Mohamed SA. Change detection using neural network in Toshka area. In: Radio science conference, 2009. NRSC 2009. National 2009 Mar 17, p. 1–10; IEEE.
[22] Qiu G, Varley MR, Terrell TJ. Accelerated training of backpropagation networks by using adaptive momentum step. Electron Lett 1992 Feb 13;28(4):377–9.
[23] Thimm G, Moerland P, Fiesler E. The interchangeability of learning rate and gain in backpropagation neural networks. Neural Comput 1996 Feb 15;8(2):451–60.
[24] Zhang D, Martinez N, Lindholm G, Ratnaweera H. Manage sewer in-line storage control using hydraulic model and recurrent neural network. Water Resour Manage 2018;32(6):2079–98.
[25] Zhang H, Liu F, Phanikumar MS, Meerschaert MM. A novel numerical method for the time variable fractional order mobile–immobile advection–dispersion model. Comput Math Appl 2013;66(5):693–701.
[26] Zhang L, Hu X, Wang Z, Sun F, Dorrell DG. Fractional-order modeling and State-of-Charge estimation for ultracapacitors. J Power Sources 2016;15(314):28–34.
[27] Zhang L, Yang T, Jin R, He X. O (logT) projections for stochastic optimization of smooth and strongly convex functions. In: International conference on machine learning 2013 Feb 13, p. 1121–9).
[28] Zhang S, Yu Y, Wang H. Mittag-Leffler stability of fractional-order Hopfield neural networks. Nonlinear Anal Hybrid Syst 2015;1(16):104–21.
[29] Zhang Y, Wang S, Ji G. A comprehensive survey on particle swarm optimization algorithm and its applications. Math Probl Eng 2015;2015.
[30] Wilamowski BM, Yu H. Improved computation for Levenberg–Marquardt training. IEEE Trans Neural Networks 2010;21(6):930–7.
[31] Fu X, Li S, Fairbank M, Wunsch DC, Alonso E. Training recurrent neural networks with the Levenberg–Marquardt algorithm for optimal control of a grid-connected converter. IEEE Trans Neural Networks Learn Syst 2014;26(9):1900–12.
[32] Shirangi MG, Emerick AA. An improved TSVD-based Levenberg–Marquardt algorithm for history matching and comparison with Gauss-Newton. J Petrol Sci Eng 2016;143:258–71.
[33] Smith JS, Wu B, Wilamowski BM. Neural network training with Levenberg–Marquardt and adaptable weight compression. IEEE Trans Neural Networks Learn Syst 2018;30(2):580–7.
[34] Ruiz LGB, Rueda R, Cuéllar MP, Pegalajar MC. Energy consumption forecasting based on Elman neural networks with evolutive optimization. Expert Syst Appl 2018;92:380–9.
[35] Boden M. A guide to recurrent neural networks and backpropagation. The Dallas Project; 2001.
[36] Wang L, Zeng Y, Chen T. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Syst Appl 2015;42(2):855–63.
[37] Ye Z, Kim MK. Predicting electricity consumption in a building using an optimized back-propagation and Levenberg–Marquardt back-propagation neural network: case study of a shopping mall in China. Sustain Cities Soc 2018;42:176–83.
[38] Maru A, Dutta A, Kumar KV, Mohapatra DP. Effective software fault localization using a back propagation neural network. In: Computational intelligence in data mining, p. 513–26; 2020, Springer, Singapore.
[39] Mukherjee A, Jain DK, Goswami P, Xin Q, Yang L, Rodrigues JJ. Back propagation neural network based cluster head identification in MIMO sensor networks for intelligent transportation systems. IEEE Access 2020.
[40] Park D, Cha J, Kim M, Go JS. Multi-objective optimization and comparison of surrogate models for separation performances of cyclone separator based on CFD, RSM, GMDH-neural network, back propagation-ANN and genetic algorithm. Eng Applications Comput Fluid Mech 2020;14(1):180–201.
[41] Li Q, Wang J, Tao H, Zhou Q, Chen J, Fu B, et al. The prediction model of warfarin individual maintenance dose for patients undergoing heart valve replacement, based on the back propagation neural network. Clin Drug Invest 2020;40(1):41–53.
[42] Das SR, Mishra D, Parhi P, Debata PP. Mutual fund investment method using recurrent back propagation neural network. In: Advances in intelligent computing and communication, p. 330–7); 2020, Springer, Singapore.
[43] Yu L, Jin W, Zhou J, Li X, Zhang Y. Optimal extraction bioactive components of tetramethylpyrazine in Chinese herbal medicine jointly using back propagation neural network and genetic algorithm in R language. Pakistan J Pharmaceut Sci 2020;33(1).
[44] Zhang Y, Wang Y, Luo G. A new optimization algorithm for non-stationary time series prediction based on recurrent neural networks. Future Generation Comp Syst 2020;102:738–45.
[45] Smyl S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. Int J Forecast 2020;36(1):75–85.
[46] Liu H, Liu C, Wang JT, Wang H. Predicting coronal mass ejections using SDO/HMI Vector Magnetic Data Products And Recurrent Neural networks. Astrophys J 2020;890(1):12.
[47] Nisioti E, Thomos N. Design of capacity-approaching low-density parity-check codes using recurrent neural networks. arXiv preprint; 2020. arXiv:2001.01249.
[48] Song Z, Yang F, Schonfeld P, Li J, Pu H. Heuristic strategies of modified Levenberg–Marquardt algorithm for fitting transition curves. J Surv Eng 2020;146(2):04020001.
[49] Wali AS, Tyagi A. Comparative study of advance smart strain approximation method using Levenberg-Marquardt and Bayesian regularization backpropagation algorithm. Mater Today: Proc 2020;21:1380–95.

[50] Kaur H, Kaur M. Fault classification in a transmission line using Levenberg–Marquardt algorithm based artificial neural network. In: Data communication and networks, p. 119–35); 2020, Springer, Singapore.
[51] Ahookhosh M, Fleming RM, Vuong PT. Finding zeros of Hölder metrically subregular mappings via globally convergent Levenberg–Marquardt methods. Optimiz Methods Softw 2020:1–37.
[52] Yang N, Shi F. Speech emotion recognition based on back propagation neural network. In: Information technology and intelligent transportation systems: proceedings of the 3rd international conference on information technology and intelligent transportation systems (ITITS 2018) Xi'an, China, September 15–16, 2018, Vol. 314, p. 216; 2019, January, IOS Press.
[53] Jiang J, Hu G, Li X, Xu X, Zheng P, Stringer J. Analysis and prediction of printable bridge length in fused deposition modelling based on back propagation neural network. Virtual Phys Prototyp 2019;14(3):253–66.
[54] Zhang H, Li Y, Zhang H. Risk early warning safety model for sports events based on back propagation neural network machine learning. Saf Sci 2019;118:332–6.
[55] Li S, Li S, Zhao H, An Y. Design and implementation of state-of-charge estimation based on back-propagation neural network for smart uninterruptible power system. Int J Distribut Sens Networks 2019;15 (12):1550147719894526.
[56] Feng X, He P, Zhang H, Yin W, Qian Y, Cao P, et al. Rice seeds identification based on back propagation neural network model. Int J Agric Biol Eng 2019;12 (6):122–8.
[57] Guo B, Zhen J. Coherent signal direction finding with sensor array based on back propagation neural network. IEEE Access 2019;7:172709–17.
[58] Ying X, Tang B, Zhou C. Nursing scheme based on back propagation neural network and probabilistic neural network in chronic kidney disease. J Med Imag Heal Informat 2020;10(2):416–21.
[59] Wang Y, Wang Y, Qin H, Ji H. Security risk assessment for connected vehicles based on back propagation neural network. In: CICTP 2019, p. 5733–45; 2019.
[60] Pakpahan HS, Haviluddin H, Nurpadillah DI, Islamiyah I, Setyadi HJ, Widagdo PP. A Sundanese characters recognition based on backpropagation neural network approach. In: 2019 international conference on electrical, electronics and information engineering (ICEEIE), Vol. 6, p. 250–4); 2019, October. IEEE.
[61] Zhang J, Zhao S, Wang Y, Zhu X. Improved social emotion optimization algorithm for short-term traffic flow forecasting based on back-propagation neural network. J Shanghai Jiaotong Univ (Sci) 2019;24(2):209–19.
[62] Li R, Wu Z, Li L, Cai D, Huang L, Tao J, et al. Simulation of fish migration at different water depths based on backpropagation neural network. Appl Ecol Environ Res 2019;17(1):437–49.
[63] Chen Y, An J. A novel prediction model of PM2. 5 mass concentration based on back propagation neural network algorithm. J Intell Fuzzy Syst 2019; Preprint:1–9.
[64] Fan L, Li C, Lan Z, Liu L. Big data analysis of smart grid based on back-propagation neural network algorithm in Hadoop environment. In: IOP conference series: earth and environmental science, Vol. 227, No. 3, p. 032025; 2019, February. IOP Publishing.
[65] Wang Z, Zhang J, Wang J, He X, Fu L, Tian F, et al. a back propagation neural network based optimizing model of space-based large mirror structure. Optik 2019;179:780–6.
[66] Pan L, Feng X, Sang F, Li L, Leng M, Chen X. An improved back propagation neural network based on complexity decomposition technology and modified flower pollination optimization for short-term load forecasting. Neural Comput Appl 2019;31(7):2679–97.
[67] Wang J, Anisetti M, Jeon G. Reconstruction of missing color-channel data using a three-step back propagation neural network. Int J Mach Learn Cybern 2019;10(10):2631–42.
[68] Chiba Z, Abghour N, Moussaid K, El Omri A, Rida M. New anomaly network intrusion detection system in cloud environment based on optimized back propagation neural network using improved genetic algorithm. Int J Commun Networks Inform Security 2019;11(1):61–84.
[69] Farooq MS, Khan MA, Abbas S, Athar A, Ali N, Hassan A. Skin detection based pornography filtering using adaptive back propagation neural network. In: 2019 8th international conference on information and communication technologies (ICICT), p. 106–112; 2019, November. IEEE.
[70] Chanda PB. Analysis on efficient handwritten document recognition technique using feature extraction and back propagation neural network approaches. In: Emerging technologies in data mining and information security. Singapore: Springer; 2019. p. 157–68.
[71] Ramesh VP, Baskaran P, Krishnamoorthy A, Damodaran D, Sadasivam P. Back propagation neural network based big data analytics for a stock market challenge. Commun Stat-Theory Methods 2019;48(14):3622–42.
[72] Shi F, Chen G, Wang Y, Yang N, Chen Y, Dey N, Sherratt RS. Texture features based microscopic image classification of liver cellular granuloma using artificial neural networks. In: 2019 IEEE 8th joint international information technology and artificial intelligence conference (ITAIC), p. 432–9; 2019, May. IEEE.
[73] Sornam M, Prabhakaran M. Logit-based artificial bee colony optimization (LB-ABC) approach for dental caries classification using a back propagation neural network. In: Integrated intelligent computing, communication and security. Singapore: Springer; 2019. p. 79–91.
[74] Jiang JW, Chen Z, Wang YH, Peng T, Zhu SL, Shi LM. Parameter estimation for PMSM based on a back propagation neural network optimized by chaotic artificial fish swarm algorithm. Int J Comput Commun Control 2019;14(6).
[75] Yu W, Zhao F. Prediction of critical properties of biodiesel fuels from FAMEs compositions using intelligent genetic algorithm-based back propagation neural network. Energy Sources Part A 2019:1–14.
[76] Singh RS, Saini BS, Sunkaria RK. Arrhythmia detection based on time–frequency features of heart rate variability and back-propagation neural network. Iran J Comput Sci 2019;2(4):245–57.
[77] Vlachas PR, Pathak J, Hunt BR, Sapsis TP, Girvan M, Ott E, Koumoutsakos P. Forecasting of spatio-temporal chaotic dynamics with recurrent neural networks: a comparative study of reservoir computing and backpropagation algorithms. arXiv preprint arXiv:1910.05266; 2019.
[78] Matayoshi J, Cosyn E, Uzun H. Using recurrent neural networks to build a stopping algorithm for an adaptive assessment. In: International conference on artificial intelligence in education, p. 179–84; 2019, June. Springer, Cham.
[79] Kumar AD, Thodupunoori H, Vinayakumar R, Soman KP, Poornachandran P, Alazab M, et al. Enhanced domain generating algorithm detection based on deep neural networks. In: Deep learning applications for cyber security. Cham: Springer; 2019. p. 151–73.
[80] Payani A, Fekri F, Alregib G, Mohandes M, Deriche M. Compression of seismic signals via recurrent neural networks: lossy and lossless algorithms. In: SEG technical program expanded abstracts 2019, p. 4082–6; 2019. Society of Exploration Geophysicists.
[81] Liu Y. Novel volatility forecasting using deep learning–long short term memory recurrent neural networks. Expert Syst Appl 2019;132:99–109.
[82] Zhou Y, Yue H, Kong Q, Zhou S. Hybrid event detection and phase-picking algorithm using convolutional and recurrent neural networks. Seismol Res Lett 2019;90(3):1079–87.
[83] Luo J, Wu J, Zhao S, Wang L, Xu T. Lossless compression for hyperspectral image using deep recurrent neural networks. Int J Mach Learn Cybern 2019;10 (10):2619–29.
[84] Zhang Z, Kag A, Sullivan A, Saligrama V. Equilibrated recurrent neural network: neuronal time-delayed self-feedback improves accuracy and stability. arXiv preprint arXiv:1903.00755; 2019.
[85] Madahana MC, Ekoru JE, Mashinini TL, Nyandoro OT. Mine workers threshold shift estimation via optimization algorithms for deep recurrent neural networks. IFAC-PapersOnLine 2019;52(14):117–22.
[86] Allen-Zhu Z, Li Y, Song Z. On the convergence rate of training recurrent neural networks. In: Advances in neural information processing systems, p. 6673–85; 2019.
[89] Camero A, Toutouh J, Alba E. A specialized evolutionary strategy using mean absolute error random sampling to design recurrent neural networks. arXiv preprint arXiv:1909.02425; 2019.
[90] Wu D, Lu H, Hu M, Zhao B. Independent random recurrent neural networks for infrared spatial point targets classification. Appl Sci 2019;9(21):4622.
[91] Eslamibidgoli MJ, Mokhtari M, Eikerling MH. Recurrent neural network-based model for accelerated trajectory analysis in AIMD simulations. arXiv preprint arXiv:1909.10124; 2019.
[93] Duneja A, Puyalnithi T, Vankadara MV, Chilamkurti N. Analysis of inter-concept dependencies in disease diagnostic cognitive maps using recurrent neural network and genetic algorithms in time series clinical data for targeted treatment. J Ambient Intell Hum Comput 2019;10(10):3915–23.
[94] Rosenberg I, Shabtai A, Elovici Y, Rokach L. Defense methods against adversarial examples for recurrent neural networks. arXiv preprint arXiv:1901.09963; 2019.

Further reading

[87] Livieris IE. Forecasting economy-related data utilizing weight-constrained recurrent neural networks. Algorithms 2019;12(4):85.
[88] Rezk NM, Purnaprajna M, Nordström T, Ul-Abdin Z. Recurrent neural networks: an embedded computing perspective. arXiv preprint arXiv:1908.07062; 2019.
[92] ElSaid A, Benson S, Patwardhan S, Stadem D, Desell T. Evolving recurrent neural networks for time series data prediction of coal plant parameters. In: International conference on the applications of evolutionary computation (Part of EvoStar), p. 488–503; 2019, April. Springer, Cham.