Available online at www.dergipark.ulakbim.gov.tr/beuscitech/

Journal of Science and Technology

E-ISSN 2146-7706

A study of artificial neural network training algorithms for classification of cardiotocography signals

Zafer CÖMERT [a]*, Adnan Fatih KOCAMAZ [b]

[a] Bitlis Eren University, Department of Computer Engineering, Bitlis, Turkey

[b] İnönü University, Department of Computer Engineering, Malatya, Turkey

ARTICLE INFO

Article history:
Received 13 September 2017
Received in revised form 24 October 2017
Accepted 06 November 2017

Keywords:
Biomedical Signal Processing
Fetal Heart Rate
Artificial Neural Network
Training Algorithm
Classification

ABSTRACT

Cardiotocography (CTG) containing of fetal heart rate (FHR) and uterine contraction (UC) signals is a monitoring technique. During the last decades, FHR signals have been classified as normal, suspicious, and pathological using machine learning techniques. As a classifier, artificial neural network (ANN) is notable due to its powerful capabilities. For this reason, behaviors and performances of neural network training algorithms were investigated and compared on classification task of the CTG traces in this study. Training algorithms of neural network were categorized in five group as Gradient Descent, Resilient Backpropagation, Conjugate Gradient, Quasi-Newton, and Levenberg-Marquardt. Two different experimental setups were performed during the training and test stages to achieve more generalized results. Furthermore, several evaluation parameters, such as accuracy (ACC), sensitivity (Se), specificity (Sp), and geometric mean (GM), were taken into account during performance comparison of the algorithms. An open access CTG dataset containing 2126 instances with 21 features and located under UCI Machine Learning Repository was used in this study. According to the results of this study, all training algorithms produced rather satisfactory results. In addition, the best classification performances were obtained with Levenberg-Marquardt backpropagation (LM) and Resilient Backpropagation (RP) algorithms. The GM values of RP and LM were obtained as 89.69% and 86.14%, respectively. Consequently, this study confirms that ANN is a useful machine learning tool to classify FHR recordings.

1. Introduction

Vital activities of fetuses, such as respiratory and nutrition, directly depend on the placenta, so there is a strong relationship between the development of fetuses and placenta activities. This relation can be observed noninvasively during both of the antepartum and more importantly intrapartum periods using cardiotocography (CTG) (Grivell et al., 2010). CTG is a monitoring technique used routinely for recording of fetal heart rate (FHR) and uterine contraction (UC) signals during pregnancy and delivery. These biophysical signals are called as non-stress test (NST) and contraction stress test (CST) recorded simultaneously. The tests are used for investigating the functional state of the fetal autonomic nervous system depending on the respiratory function of the placenta (Cesarelli et al., 2007). In the last trimester of pregnancy (i.e., after the 28th week), CTG is applied more commonly, especially in high-risk pregnancies. In addition, CTG is a routine procedure for assessment of the fetal state and has been used since the end of the 1960s by obstetricians.

The first guidelines for the interpretation of CTG have been announced by the different institutions, such as International Federation of Gynecology and Obstetrics (FIGO) (Ayres-de-Campos et al., 2015) and the National

* corresponding author:
Tel.: +90 5067814052
E-mail address: comertzafer@gmail.com

Institute of Child Health and Human Development (NICHD) (Tongsong et al., 2005). As a result of the existing guidelines, the shapes and changes of FHR signals, such as baseline, acceleration, deceleration, and variability have been defined as morphological features that reflect the characteristic features of the signals (Pinas and Chandraharan, 2016). Although there is not a single standard agreed by all of the society on the interpretation of CTG, the computer-aided studies have been developed considering the mentioned guidelines (Nunes and Ayres-de-Campos, 2016).

FHR signals have been recently classified as normal, suspicious and pathological (Jezewski et al., 2010). For this particular purpose, many machine learning techniques such as Random Forest (RF) (Tomáš et al., 2013), Naïve Bayes (Menai et al., 2013), Extreme Learning Machine (ELM) (Cömert et al., 2016; Cömert and Kocamaz, 2017a; Ravindran et al., 2015), Logistic Regression (LR) (Huang and Yung-Yan, 2012), Support Vector Machine (SVM) (Ocak, 2013), Decision Trees (DT) (Karabulut and Ibrikci, 2014), Adaptive Boosting (AdaBoost) (Yang and Zhidong, 2017) and radial basis function network (RBFN) (Sahin and Subasi, 2015) were employed in the previous works. In this context, it was observed that neural networks have given robust and promising results (Cömert and Kocamaz, 2016). It is a well-known fact that the phase of training a neural network is highly consistent with unconstrained optimization theory and various algorithms have attempted to speed up training steps. Moreover, heuristics approaches, such as momentum or variable/adaptive learning rate have been employed to accelerate neural network training (Hagan et al., 2014).

It is aimed to compare performances of the training algorithms employed by the networks having the same topology in order to determine the most efficient and fastest training algorithms on CTG signals classification task considering 12 neural network's training algorithms. ANNs produced remarkable results due to its powerful pattern recognition and classification capabilities. The behaviors and performances of ANN training algorithms were compared to each other in terms of the performance metrics, training times (TT), epochs, and mean square error (MSE).

2. Material and Methods

2.1. Dataset Description

Before training of a network, it is a challenge to see that how much data is required. This situation (amount of data required) directly depends on the complexity of the problem. In practice, many problems require a large amount of data, but it should be noticed that the size of a dataset is closely related to the selection of the number of neurons in the neural network. In summary, a sufficient amount of data must be used for training a network (Amato et al., 2013). All variations of possible known of the problem area should be enclosed to the dataset. Sufficient data representation to a system is necessary for obtaining a robust and reliable network. The data may occur continuous, discrete or a mixture of both (Basheer and Hajmeer, 2000).

During the evaluation of networks, a dataset should be partitioning into three subsets. First one is training set which is used to update weights and biases according to output values of network and targets. The second one is validation which is used to measure of network generalization and is used to stop training before overfitting occurs. The last one is testing which provides an independent measure of network performance, using random indices and is used to predict future performance of the network. In general, the training set makes up approximately 70% of full dataset, with validation and testing making up approximately 15% each.

CTG dataset located under the UCI Machine Learning Repository was used in this study (Lichman, 2013a). The digital CTG signals were transmitted from electronic fetal monitoring devices to computer by using the serial port. Afterward, the software called as SisPorto 2.0© acquired signals and calculated the diagnostic features automatically (Ayres-de-campos et al., 2000). 2126 samples in the dataset are described with 21 features that 8 of them are continuous, and 13 are discrete. Also, all signals were classified by three expert obstetricians. Of these recordings, 1655 represent normal, 295 suspects, and 176 pathological. The summary of UCI CTG dataset is given in Table 1.

Table 1. The summary of UCI CTG Dataset (Lichman, 2013)

Symbol	Feature information
LB	FHR baseline (beats per minute)
AC	# of accelerations per second
FM	# of fetal movements per second
UC	# of uterine contractions per second
DL	# of light decelerations per second
DS	# of severe decelerations per second
DP	# of prolonged decelerations per second
ASTV	Percentage of time with abnormal short-term variability
MSTV	Mean value of short-term variability
ALTV	Percentage of time with abnormal long-term variability
MLTV	Mean value of long-term variability
Width	Width of FHR histogram
Min	Minimum of FHR histogram
Max	Maximum of FHR histogram
Nmax	# of histogram peaks
Nzeros	# of histogram zeros
Mode	Histogram mode
Mean	Histogram mean
Median	Histogram median
Variance	Histogram variance
Tendency	Histogram tendency
NSP	Fetal state class (Normal, Suspicious, Pathological)

The whole features described in Table 1were used as the input to the networks.

2.2. Feature Transform

The feature transform covers two independent steps that are feature extraction and feature selection. Feature selection is used to reduce the dimension of input space and isolates redundant or irrelevant information from to input space (Chudacek et al., 2008). The size of input vector must be kept as small as possible when designing a network. In this case, computation cost decreases, whereas the performance of network increases. At the same time, this process prevents the overfitting (Onnia et al., 2001).

A series of events, such as determining the mean FHR level, transient events such as accelerations and decelerations and FHR variability (FHRV) are detected in feature extraction stage to recognize FHR patterns. The determination of the basic morphological features is crucial in terms of automatic analysis as well as the clinical management (Ayres-de-Campos et al., 2015). It should be emphasized that a correct classification is based on the selection of prominent features which are used to identify possible diagnoses.

Table 2. A summary of CTG classification criteria according to FIGO (Ayres-de-Campos et al., 2015)

	Normal pattern	Suspicious pattern	Pathological pattern
Baseline	110 – 160 bpm	160 – 170 bpm or 100 – 110 bpm	<100 bpm or > 170 bpm
Variability	5-25 bpm	5 – 10 bpm for more than 40 minutes	Persistence of heart rate variability less than 5 bpm for more than 40 minutes.
Other	**For pathological patters** • Severe variable decelerations or severe repetitive early decelerations • Prolonged decelerations • Late deceleration • A sinusoidal pattern		

In this section, a brief overview of prominent features of CTG is presented. Baseline FHR is described over a period of 5 or 10 minutes in the case that accelerations and decelerations are absent (Sundar et al., 2012). The baseline level range between 110 and 160 bpm is accepted as normal, on the other hand, extremely accelerations that above 170 bpm and decelerations that below 100 bpm are referred as pathological situations: tachycardia and bradycardia, respectively. Accelerations are specified as a good health status for fetus while decelerations are pointed as the symptom of fetal distress (Czabanski et al., 2012). Also, variability is substantial to make a decision. Variability is described as amplitude oscillations around baseline heart rate. A summary of CTG classification criteria according to FIGO is given in Table 2.

2.3. Designing of Neural Networks

ANN, which has a powerful connection between the input and output variables, is a mathematical model that reflects learning and generalization ability of human neural architecture (Amato et al., 2013). ANN can be employed to solve various real-world problems, such as any complex functional approximation, pattern classification or clustering, forecasting, and image completion. Therefore, ANNs are evaluated as a valuable computational model (Günther and Fritsch, 2010). ANNs consist of the input layer and the output layer, furthermore, the layer(s) between input and output layers are referred to hidden layer that may be one or more, helps to capture nonlinearity and is not directly observed. In theory, ANNs can be contained an arbitrary number of input and output variables. However, it must be noted that the number of variables and computational cost is entirely proportional (Hu and Hwang, 2001). The number of neurons per layers, training algorithms, epochs, maximum training time, performance values, gradient, and validation checks can be set before training of an ANN, so it can be expressed that ANN is very flexible and versatile tool.

Computing Environment: The computing environment is a workstation which has Intel Xeon CPU E5-2687W v3 3.10 GHz and 32 GB of RAM memory using MATLAB® (2016a) software.

Architecture of the Network: Choosing the appropriate network type and architecture is a challenge to solve the studied problem. Pattern recognition-classification network, which deals with classifying inputs into a set of target categories, was chosen for this study. Initially, the topology of the networks was established as$\{21, 10, 3\}$. It indicates that the dimension of the layers consists of 21 input variables, 10 nodes in a hidden layer, and 3 output nodes respectively. The general structure of network is shown in Figure 1. Tangent sigmoid transfer function and softmax transfer function were used in the hidden layer and output layer, respectively. In the pattern recognition problems, log-sigmoid or tangent sigmoid transfer functions are used commonly. For function approximation or regression problems, the mean square error (mse) works well because of the target values are continuous. Otherwise, cross-entropy, which is a performance index, is proposed for classification problems, because of that the targets take on discrete values. In general, softmax transfer function and cross-entropy performance function are used together. As shown in Figure 1, the tangent sigmoid and softmax transfer functions were chosen in this study.

Training Concepts: A network can be trained by two alternative concept either incremental or batch training.

Incremental training is also known as the example-by-example model and is habitually chosen with dynamic networks such as an adaptive filter. However, it can be applied to static networks. The weights are updated in each iteration immediately. This concept includes small storage, but a first bad example may force the search in the wrong direction (Basheer and Hajmeer, 2000). As for batch training concept, the weights are completely updated after all the inputs are presented to the network. Moreover, this concept is more efficient in the MATLAB® environment (Demuth et al., 2010). For this reason, batch training style was used in this study.

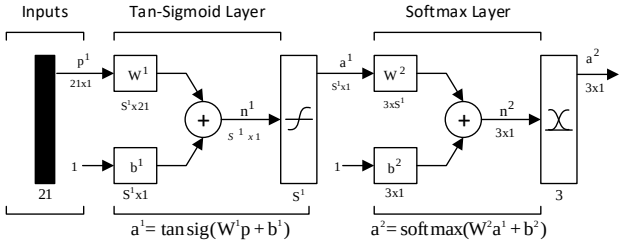

Figure 1. The general structure of the networks. The input and output of network comprised of matrices dimension $21x1$ and $3x1$ respectively. Herein, p^i indicate ith input instance, n^i indicates net the function of ith layer. S, indicates number of neurons in hidden layer (e.g. $S^1 = 10$, it means that there are 10 neurons/nodes in hidden layer), a^i indicates output value of ith layer.

2.4. Training Algorithms

A training algorithm finds a decision function that updates the weights of the network. There are many variations of the training algorithms. The several of those are given in Table 3 and were discussed in the scope of this study. It is a difficult task to estimate which training algorithm will produce the best results (Sundar et al., 2012). The algorithms update the network weights and biases to map correctly arbitrary inputs to outputs. In this study, the training algorithms were collected into five group as Gradient Descent, Conjugate Gradient, Quasi-Newton, Resilient Backpropagation, and Levenberg-Marquardt algorithms. The gradient is expressed as technical term backpropagation (BP), which backwardly propagates the error between the network output and the desired output. The aim of BP is to optimize the weights so that the neural network can learn how to estimate properly varying inputs to outputs. In the BP networks, the data is fed to forward, and there is no feedback (Hagan et al., 2014).

Gradient Descent Algorithms (GDAs): BP learning algorithms provide needed and desired weights. The standard BP algorithm is gradient descent backpropagation (GD) that is the batch steepest descent training algorithm and aims to decline network error as rapidly as possible. One iteration of GD algorithm defines as in (1). In GD, the weights are changed in proportion to negative of an error derivate on each weight. GD is the most straightforward implementation of BP. Eq. 1 is repeated until the network achieves a point of convergence.

$$x_{k+1} = x_k - a_k g_k \quad (1)$$

Between (1) and (11), current weights and biases are denoted by a vector of x_k, g_k is current gradient of the error with respect to the weight vector, a_k is the learning rate (or it can be expressed as length of step size), x_{k+1} is a new weight vector, and k represents the number of iterations proceeded by the methods.

Table 3. ANN training algorithms with their group number (G) employed in this study

G	Acr.	Description
1	GD	Gradient descent backpropagation
1	GDA	Gradient descent with adaptive learning rate backpropagation
1	GDM	Gradient descent with momentum backpropagation
1	GDX	Gradient descent with momentum and adaptive learning rate backpropagation
2	RP	Resilient Backpropagation
3	CGF	Conjugate gradient backpropagation with Fletcher-Reeves restarts
3	CGP	Conjugate gradient backpropagation with Polak/Ribiére restarts
3	CGB	Conjugate gradient with Powell/Beale restarts
3	SCG	Scaled conjugate gradient backpropagation
4	BFGS	BFGS Quasi-Newton backpropagation
4	OSS	One-step secant backpropagation
5	LM	Levenberg-Marquardt backpropagation

Choosing an applicable learning rate is crucial to understand the stability of algorithms. Choosing of a proper learning rate may prevent instability or slow convergence. When the learning rate is selected as enough small, it exhibits an ideal behavior in this case. However, this situation leads to more expensive computational cost and requires a longer training phase. On the other hand, when the learning rate is chosen too large, it promotes oscillation in the error surface, or may cause a jump completely over the global minimum, or is trapped at the shallow local minimum (Wythoff, 1993). Furthermore, the size of learning rate affects whether the network is stable or not. The learning rate is unchanged during the training of the GD algorithms, and these algorithms often converge slowly, trapped in a local minimum and may not give the desired performance. The different algorithms were purposed, such as Gradient Descent with Adaptive Learning Rate Backpropagation (GDA), Gradient Descent with Momentum Backpropagation (GDM), and Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation (GDX) to overcome these shortcomings. It is possible to achieve more successful results by using these heuristic techniques. The learning rate is adaptive in GDA. In the first step, initial network output and error are calculated,

and later on, new weights are adjusted depending on current learning rate in each epoch. If new error surpasses the old error, the new weights are shifted. Otherwise, the new weights are updated. The possible largest learning rate is desired without triggering oscillation in GDA. This situation provides an extremely rapid learning for networks. Unlike the GDA, GDM uses a variable which is called momentum. Momentum coefficient utilizes past weights changes as well as the current error. If adaptive learning rate and momentum are combined, it is possible to derivate relatively more accurate results. So, GDX consists of a combination of GDA and GDM.

Resilient Backpropagation (RP): RP is a heuristic learning algorithm that improved the convergence speed by using the only sign of the derivative, not magnitude of the derivative of the error function for the weight update as shown in (2). RP reduces the number of learning steps and other adaptive parameters, according to GDAs, and it computes local learning scheme easily (Riedmiller and Braun, 1993).

$$\Delta x_k = -sign\left(\frac{\Delta E_k}{\Delta x_k}\right)\Delta k \tag{2}$$

Δx_k is used to denote the changes of current weights vector, ΔE_k is used to denote error function E at k, and Δk is used to denote the increase in bias.

Conjugate Gradient Algorithms (CGAs): Conjugate gradient algorithms (CGAs), which can be evaluated as one class of optimization methods, are much more efficient than GDAs having a low memory requirement and providing fast convergence. However, it tends to be unstable in large-scale problems occasionally (Møller, 1993). Also, CGAs are practical for minimizing functions of very many variables since the storing of any matrices is not necessary (Powell, 1977). The whole CGAs work by searching in steepest descent direction (Hagan et al., 2014) which is negative of the gradient as given in (3).

$$p_0 = -g_0 \tag{3}$$

In the next step, a series calculations as given in (4) are made for a line search (El-Nabarawy et al., 2013).

$$x_{k+1} = x_k + a_k p_k \tag{4}$$

Herein, the search direction is indicated withp_k. The next search direction is selected according to (5) and depends on previous search direction.

$$p_k = -g_k + \beta_k p_{k-1} \tag{5}$$

The computation of constant β_k is different in CGAs. Fletcher-Reeves (FR) update is used by Conjugate Gradient Backpropagation with Fletcher-Reeves Restarts (CGF), and Polak-Ribiére (PR) update is used by Conjugate Gradient Backpropagation with Polak/Ribiére Restarts (CGP) (Saini and Soni, 2002). The formulations of the methods are given in (6) and (7), respectively. According to computational experiments, PR performs better than FR. Especially PR method seems to be preferable compared to others methods (Luenberger et al., 1984).

$$\beta_{k-1} = \frac{g_k^T + g_k}{g_{k-1}^T + g_{k-1}} \tag{6}$$

$$\beta_{k-1} = \frac{(g_k - g_{k-1})^T g_k}{g_{k-1}^T + g_{k-1}} \tag{7}$$

The search direction resets at regular intervals in CGAs. When the condition occurs in (8), the search direction resets to the negative of the gradient in Conjugate Gradient with Powell/Beale Restarts (CGB), thereby the efficiency of the training is increased (Powell, 1977).

$$|g_{k-1}g_k| \geq 0.2\ \|g_k\| \tag{8}$$

Scaled Conjugate Gradient Backpropagation (SCG) is the last algorithm in this group and uses second order information from feedforward neural network such as Levenberg-Marquardt (LM) algorithm. It avoids the time-consuming line search at each iteration (Møller, 1993).

Quasi-Newton Algorithms (QNAs): QNAs are similar in fast optimization to CGAs and can be considered as the basic local method using second-order information (Battiti, 1992). The computation cost of the algorithms is more expensive, dense, and complex when compared to CGAs. The weights are updated according to the Newton method given in (9). Quasi-Newton is based on Newton method, but it does not require calculation of second derivatives so that it is called Quasi-Newton (or secant) methods (Dennis Jr and Schnabel, 1996).

$$x_{k+1} = x_k - H_k^{-1} g_k \tag{9}$$

H_k is the Hessian matrix (second derivatives) of the performance index at current values of the weights and biases. The new weighs x_{k+1} is computed as a function of the gradient and the current weight x_k in the Newton algorithms.

BFGS Quasi-Newton backpropagation (BFGS) algorithm necessitates high computation and storage. For this reason, it is recommended for many networks with a small number of weights (Azar, 2013). On the other hand, one-step secant backpropagation (OSS) method fills a gap between CGAs and QNA. OSS does not store the entire Hessian matrix, requires less storage, and computation per epoch than the BFGS (Hagan et al., 2014).

Levenberg-Marquardt Algorithm (LM): LM is agreed as a standard technique for solving nonlinear least squares

problems. It occurs a combination of gradient descent and Gauss-Newton method. LM exhibits adaptive behavior according to the distance of solution so that it can be guaranteed the solution in many cases (Marquardt, 1963). When BP is gradient descent, the algorithm is far from the solution and it is quite slow (Hagan and Menhaj, 1994). Conversely, in the case that BP is Gauss-Newton, the algorithm is close to correct one. In LM, computation of the approximate Hessian given in (10) is done slightly, and the gradient is computed in the manner given in (11).

$$H = J^T J \tag{10}$$

$$g = J^T e \tag{11}$$

where J and e indicate the Jacobian matrix and a vector of network errors, respectively. LM algorithm uses this approximation in the manner given in (12) such as Newton.

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{12}$$

In summary, GDAs update the weights and biases in the direction of the negative gradient of the performance function. Unlike GDAs, CGAs search steepest descent direction along conjugate directions. QNAs converge faster than CGAs and give better-generalized results. However, the calculations may take a long time. The conjugate gradient and Quasi-Newton only use the first derivative of the function. Therefore, these methods are regularly preferred in applications when only the first derivative is known or when higher derivatives are very expensive to calculate.

3. Results and Discussion

3.1. Performance Evaluation Criteria

In this study, primarily, the performances of the networks were analyzed by using confusion matrix that consists of four prognostic indices which are True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). In this scope, TP and TN represent the numbers of samples predicted correctly, whereas FP and FN represent the numbers of samples predicted incorrectly. In addition, confusion matrix can help to calculate several performance measures, some of which are described as follow:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

Accuracy (ACC) expresses the overall performance of the model and is calculated as in (13).

$$Se_M = \frac{\sum_{i=1}^{M} \frac{TP_i}{TP_i + FN_i}}{M} \tag{14}$$

Sensitivity (Se), it is also known as recall, only focuses on positive examples and predictions. In general, Se is used to measure the model success on positive class. Here, M shows the number of classes.

$$Sp_M = \frac{\sum_{i=1}^{M} \frac{TN_i}{TN_i + FP_i}}{M} \tag{15}$$

Similarly, specificity (Sp) is used to measure the model performance on negative class and the calculation of Sp is shown in (15).

$$GM_M = \sqrt{Se_M Sp_M} \tag{16}$$

The distribution of data between classes may be unbalanced in most of cases, in which case the geometric mean (GM) metric becomes extremely useful to achieve more objective results.

Another useful metric for describing the performance of a classification model is Receiver Operating Characteristic (ROC) curve. It is a threshold-independent measure and is a fundamental tool for diagnostic test evaluation. A particular decision threshold is corresponding to a point on ROC curve, and it determines a true positive/false positive pair. The area under this curve (AUC) is a useful index to compare classifiers. It represents the distinction between the diagnostic groups. At this point, it is important to mention k -fold cross-validation method in order to evaluate a model. When there is a small amount of data for training a model, this approach becomes beneficial. In this method, the whole dataset is divided into k part in order to be used as the training and test sets. The specified $k - 1$ part is used for training and the rest is used for testing in k rounds. Thus, the performance metrics are reported by taking average values of the metrics. In this study, k was adjusted to 10.

3.2. Experimental Results

The data set comprises of 1655 normal, 295 suspicious and 176 pathological samples. The distribution of 2126 samples with 21 features on first three principal components is shown in Figure 2. As seen in Figure 2, it is unlikely that the dataset can be linearly separated.

In the first experimental setup, the topology of the networks was configured as 21 input, one hidden layer

which comprises of 10 hidden nodes, and three output nodes. The training times of the networks were assessed in a large range in 1 to 60 s and it was limited to 10 s. 10-fold cross-validation was applied, and the validation set was not used in the experiments. Based on these parameters, the results of 10 fold cross-validation with 1 repetition are given in Table 4. In addition to the performance metrics, AUC values of pathological samples, training time (TT), and the number of epoch of training and mean square error (MSE) were taken into account in the evaluation stage. As seen in Table 4, all training algorithms yielded rather satisfactory results. However, it is clear that RP algorithm was superior to others. ACC of 93.60%, Se of 88.42% and Sp of 90.98% were obtained using RP algorithm. Also, LM algorithm was determined as the fastest algorithm with the average training time of 9.47 s and the average epoch of 346.4. CGAs performed more effective results than GDAs whereas QNAs were superior to CGAs. Although the performance results of GDAs are seemed as weak, indeed this is not true due to the restriction of the training parameters, especially training time. If the training times of the networks are expanded, the performances of these group algorithms can be improved.

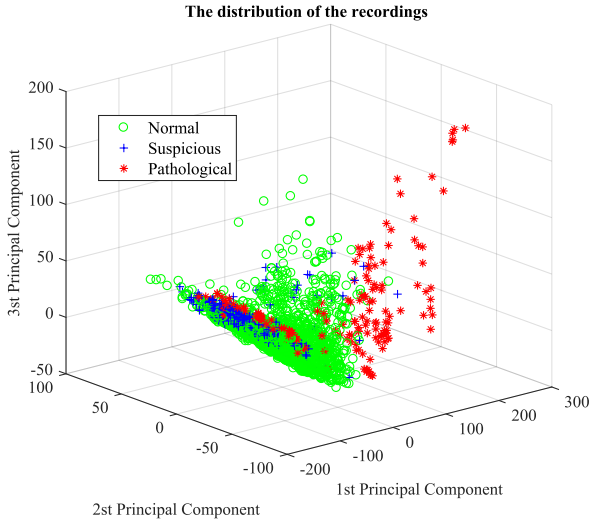

Figure 2. The distribution of 2126 recordings with 21 features on the first three principal components analysis axes

In the second experimental setup, the dataset was divided into three sets that are training (70%), validation (15%), and test (15%). The network topology was not changed. Furthermore, the training time was adjusted to 10 s and random data division were employed. The experiment was repeated 100 times, and obtained results are summarized in Table 5.

According to the results of the second experimental setup, LM algorithm was superior to others with ACC of 91.27%, Se of 82.36%, and Sp of 87.02%. Although the performance results of the training algorithms showed a decline, as seen in Table 5, the training times and epochs were decreased significantly. For example, the numbers of the epochs in the training of LM and RP algorithms were reduced from 364.4 to 18.9 and 5707.3 to 54.1 respectively. Despite that, the AUC value for pathological samples of LM algorithm increased from 0.9713 to 0.9877. It is clear that using the validation set in the training process has made it possible to carry out a faster training. Consequently, both of the first and second experimental results confirmed that LM and RP algorithms exhibited an efficient performance on detection of pathological samples. Also, the comparison of the performance metrics for these experiments are given in Figure 3.

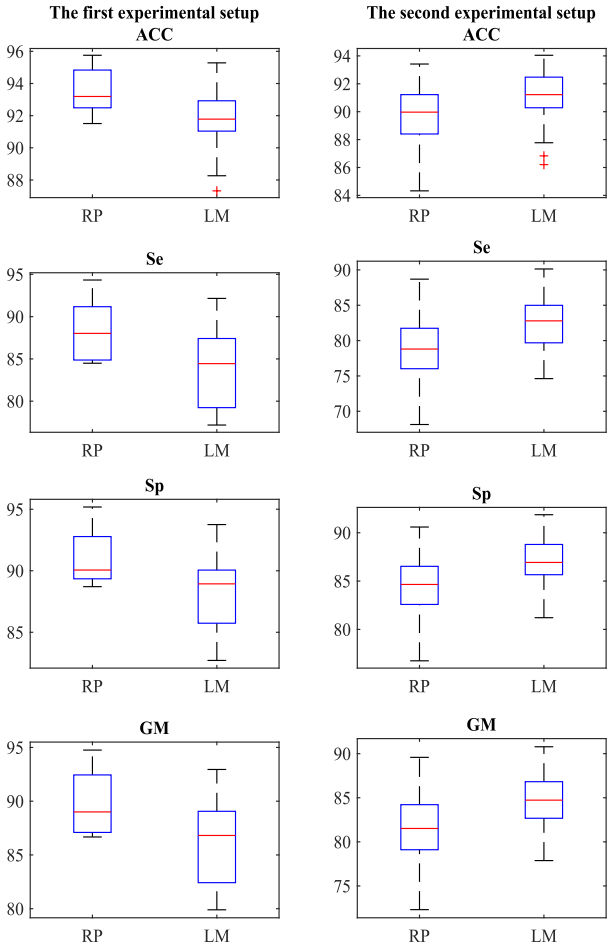

Figure 3. The comparison of the results of the first and second experiments

Table 4. Classification results of the first experiment

Alg.	ACC [%]	Se [%]	Sp [%]	GM [%]	AUC	TT [s]	Epoch	MSE
GD	86.97 (±2.59)	66.16 (±5.57)	77.87 (±3.52)	71.76 (±4.53)	0.9607 (±0.0089)	10.04 (±0.088)	5924.7 (±583.4)	0.1100 (±0.01)
GDA	92.01 (±2.11)	84.93 (±4.87)	88.43 (±3.51)	86.66 (±4.17)	0.9914 (±0.0057)	10.08 (±0.104)	5716.1 (±103.2)	0.0747 (±0.01)
GDM	86.60 (±2.58)	65.41 (±6.33)	77.16 (±4.15)	71.02 (±5.30)	0.96089 (±0.0215)	10.08 (±0.103)	5817.3 (±122.2)	0.1140 (±0.01)
GDX	92.62 (±2.40)	86.16 (±4.14)	89.36 (±3.51)	87.74 (±4.47)	0.9944 (±0.0043)	10.08 (±0.103)	5763.9 (±124.1)	0.0745 (±0.02)
RP	93.60 (±1.43)	88.42 (±3.60)	90.98 (±2.26)	89.69 (±2.89)	0.9912 (±0.0055)	10.17 (±0.211)	5707.3 (±114.3)	0.1630 (±0.06)
CGF	90.92 (±1.76)	83.43 (±4.67)	87.16 (±2.91)	85.27 (±2.91)	0.9889 (±0.0079)	10.09 (±0.103)	1979.1 (±51.0)	0.3280 (±0.13)
CGP	91.39 (±2.66)	83.91 (±4.87)	87.58 (±3.24)	85.72 (±4.00)	0.9811 (±0.0160)	10.04 (±0.014)	1984.1 (±33.5)	0.2699 (±0.09)
CGB	91.06 (±1.55)	83.14 (±3.21)	87.24 (±2.14)	85.16 (±2.63)	0.9816 (±0.0108)	10.09 (±0.110)	1952.1 (±74.4)	0.3013 (±0.08)
SCG	90.02 (±2.10)	81.07 (±6.47)	85.63 (±3.49)	83.30 (±4.98)	0.9738 (±0.0145)	10.08 (±0.104)	3788.5 (±100.5)	0.5088 (±0.17)
BFGS	91.01 (±1.91)	84.26 (±4.38)	87.86 (±2.64)	86.04 (±3.49)	0.9829 (±0.0146)	10.33 (±0.577)	613.5 (±18.0)	0.3345 (±0.16)
OSS	92.38 (±1.58)	85.33 (±2.30)	88.99 (±1.65)	87.14 (±1.87)	0.9864 (±0.0104)	10.09 (±0.104)	2108.2 (±61.0)	0.1367 (±0.05)
LM	91.54 (±2.30)	84.16 (±4.85)	88.19 (±3.22)	86.14 (±3.99)	0.9713 (±0.0327)	9.47 (±1.738)	346.4 (±65.6)	0.0544 (±0.01)

Mean (± standard deviation)

Table 5. Classification results of the second experiment

Alg.	ACC [%]	Se [%]	Sp [%]	GM [%]	AUC	TT [s]	Epoch	MSE
GD	87.15 (±2.06)	67.04 (±5.06)	78.43 (±3.13)	72.50 (±4.14)	0.9659 (±0.0180)	10.03 (±0.03)	6089.1 (±43.4)	0.1080 (±0.0116)
GDA	88.29 (±1.90)	72.82 (±4.77)	81.33 (±2.86)	76.95 (±3.83)	0.9731 (±0.0128)	0.2599 (±0.0574)	139.8 (±8.1)	0.0975 (±0.0124)
GDM	87.03 (±2.14)	66.55 (±5.38)	78.11 (±3.27)	72.08 (±4.39)	0.9650 (±0.0150)	10.03 (±0.03)	6091.6 (±45.2)	0.1091 (±0.0115)
GDX	89.12 (±2.52)	76.95 (±8.46)	83.41 (±5.016)	80.08 (±6.97)	0.9752 (±0.0466)	0.9752 (±0.06)	171.1 (±22.4)	0.0852 (±0.0242)
RP	89.85 (±1.87)	78.84 (±4.02)	84.51 (±2.716)	81.62 (±3.35)	0.9872 (±0.0073)	0.1228 (±0.06)	54.1 (±24.2)	0.0806 (±0.0142)
CGF	89.74 (±1.99)	79.26 (±4.46)	84.75 (±3.05)	81.96 (±3.74)	0.9854 (±0.0081)	0.9854 (±0.08)	35.9 (±11.1)	0.0800 (±0.0140)
CGP	89.67 (±1.50)	79.10 (±3.90)	84.63 (±2.46)	81.81 (±3.17)	0.9861 (±0.0071)	0.2101 (±0.08)	34.7 (±11.5)	0.0797 (±0.0109)
CGB	89.82 (±1.65)	79.23 (±3.87)	84.85 (±2.54)	81.99 (±3.20)	0.9816 (±0.0101)	0.2075 (±0.08)	34.1 (±10.7)	0.8047 (±0.0111)
SCG	89.39 (±1.64)	78.17 (±4.58)	84.05 (±2.76)	81.05 (±3.66)	0.9856 (±0.0076)	0.1386 (±0.06)	40.8 (±11.7)	0.0816 (±0.0111)
BFGS	89.11 (±1.53)	78.33 (±3.69)	83.87 (±2.42)	81.05 (±3.03)	0.9835 (±0.0106)	0.4802 (±0.15)	39.4 (±11.7)	0.0835 (±0.0107)
OSS	90.06 (±1.69)	79.52 (±4.06)	85.08 (±2.63)	82.25 (±3.33)	0.9856 (±0.0076)	0.2874 (±0.09)	56.0 (±19.5)	0.0778 (±0.0106)
LM	91.27 (±1.55)	82.36 (±3.69)	87.02 (±2.36)	84.65 (±3.012)	0.9877 (±0.0069)	0.5515 (±1.12)	18.9 (±4.3)	0.0432 (±0.0067)

Table 6. Comparison of studied ANN training algorithms and previously reported models

References	Methods	Software	# Features	# Classes	Sp	Se
(Sahin and Subasi, 2015)	Random Forest, 10-fold cross validation	WEKA	21	2	99.7	94.1
(Tomáš et al., 2013)	Random Forest, 50% train -50% test, correlation feature selection	WEKA	7	3	84.6*	89.8*
(Ocak, 2013)	Genetic Algorithm, Support Vector Machine	MATLAB	13	2	99.3	100
(Yılmaz, 2016)	Generalized regression neural network, 10 fold cross validation	MATLAB	21	3	83.1	92.6
(Cömert and Kocamaz, 2017b)	Artificial Neural Network, 10-fold cross validation	MATLAB	21	2	99.7	97.9
This paper	ANN with RP algorithm, 10-fold cross validation	MATLAB	21	3	88.4	90.9
This paper	ANN with LM algorithm, 70% train - 15% validation – 15% test, 100 trial	MATLAB	21	3	82.4	87.0

***** The metric which was calculated according to information given in the paper. **#**: the number of

3.3. Discussion

A comparison which considers the performance metrics, the number of features used to feed classifiers, the number of classes, simulation platform and methods was carried out between the training algorithms of ANN and previously reported works and the comparison is presented in Table 6. As can be seen from Table 6, comparing the performance of the training algorithms and the related works is a very difficult task due to the variable parameters, such as the employed methods, the number of features, the number of classes and simulation platform. (Sahin and Subasi, 2015) and (Cömert and Kocamaz, 2017b) addressed this classification task on two classes indicated as normal and pathological by using different methods and simulation programs. A rather satisfactory results over 99% of Se and Sp were reported by the researchers. (Tomáš et al., 2013) and (Ocak, 2013) employed feature selection algorithms to reduce the dimension of the feature set before classification stage. On the other hand, the problem was dealt with by Ocak and Tomas et al. as binary and multi classification task. In addition, (Yılmaz, 2016) focused three different type of ANN in the study and achieved the best results using generalized regression neural network.

4. Conclusion

CTG is a significant biomedical signal since it carries vital information on fetal state. The early detection of stressful conditions for the fetus from this traces may prevent adverse events such as cerebral palsy. However, as can be seen from the literature, there is a high variability in the interpretation of these trace among observers. Computerized analysis of CTG is admitted as the most promising way to tackle these drawbacks. For this reason, we focused on a robust and efficient classifier, ANN. More specifically, we investigated the behaviors and performances of 12 training algorithms of ANN in 5 groups with two different experimental setups. The experiments were performed on an open access dataset consisting 1655 normal, 295 suspicious and 176 pathological samples. The comparison of each algorithm to others was realized considering performance metrics, AUC, TT, Epoch and MSE. We achieved the most efficient results with RP and LM algorithms. Se, Sp, QI, AUC values of LM algorithm were obtained as 84.16%, 88.19%, 86.14% and 0.9713, respectively whereas these values of RP algorithm were obtained as 88.42%, 90.98%, 89.69% and 0.9912, respectively. A comparison between the results of LM-RP algorithms and previously reported works was realized in the last part of this study.

In future works, combinations of the feature selection and different machine algorithms will investigate to provide a more effective model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

Amato, F. L. A., Peña-Méndez E. M., et al., 2013. Artificial Neural Networks in Medical Diagnosis, Journal of Applied Biomedicine 11(2), 47–58.

Ayres-de-Campos, D., Bernardes J., Garrido, A., et al., 2000. SisPorto 2.0: A Program for Automated Analysis of Cardiotocograms. Journal of Maternal-Fetal Medicine 9(5), 311–318.

Ayres-de-Campos D., Spong, C. Y., and Chandraharan, E., 2015. FIGO consensus guidelines on intrapartum fetal monitoring: Cardiotocography. International Journal of Gynecology & Obstetrics 131(1), 13–24.

Azar, A.T., 2013. Fast neural network learning algorithms for medical applications. Neural Computing and Applications, Springer 23(3–4), 1019–1034.

Basheer, I. A., and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of Microbiological Methods 43(1), 3–31.

Battiti, R., 1992. First-and second-order methods for learning: between steepest descent and Newton's method. Neural computation, MIT Press 4(2), 141–166.

Cesarelli, M., Romano M., Bifulco, P., et al., 2007. An algorithm for the recovery of fetal heart rate series from CTG data. Computers in Biology and Medicine 37(5), 663–669.

Chudacek, V., Spilka, J., Rubackova B., et al., 2008. Evaluation of feature subsets for classification of cardiotocographic recordings. Computers in Cardiology (CinC), 845–848.

Cömert, Z. and Kocamaz, A. F., 2016. Evaluation of Fetal Distress Diagnosis during Delivery Stages based on Linear and Nonlinear Features of Fetal Heart Rate for Neural Network Community. International Journal of Computer Applications, New York, USA: Foundation of Computer Science (FCS), NY, USA 156(4), 26–31.

Cömert, Z., and Kocamaz, A. F., 2017a. Cardiotocography analysis based on segmentation-based fractal texture decomposition and extreme learning machine. 25th Signal Processing and Communications Applications Conference (SIU), 1–4.

Cömert, Z., and Kocamaz, A. F., 2017b. Comparison of Machine Learning Techniques for Fetal Heart Rate Classification. Acta Physica Polonica A 132(3), 451-454.

Cömert, Z., Kocamaz, A. F., and Gungor, S., 2016. Cardiotocography signals with artificial neural network and extreme learning machine. 24th Signal Processing and Communication Application Conference (SIU).

Czabanski, R., Jezewski J., Matonia, A., et al., 2012. Computerized analysis of fetal heart rate signals as the predictor of neonatal acidemia. Expert Systems with Applications 39(15), 11846–11860.

Demuth, H., Beale, M., and Hagan, M., 2010. Neural Network Toolbox TM 6 User's Guide. Network, the MathWorks, Inc.

Dennis, J. J.E., and Schnabel, R. B., 1996. Numerical methods for unconstrained optimization and nonlinear equations. SIAM.

El-Nabarawy, I., Abdelbar, A. M., and Wunsch, D.C., 2013. Levenberg-Marquardt and Conjugate Gradient methods applied to a high-order neural network. Neural Networks (IJCNN), the 2013 International Joint Conference, 1–7.

Grivell, R. M., Alfirevic, Z., Gyte, G., M., et al., 2010. Antenatal cardiotocography for fetal assessment. The Cochrane database of systematic reviews, England (1), 1–48.

Günther, F., and Fritsch, S., 2010. neuralnet: Training of neural networks. The R journal 2(1), 30–38.

Hagan, M. T., and Menhaj, M. B., 1994. Training feedforward networks with the Marquardt algorithm. IEEE transactions on Neural Networks, IEEE 5(6), 989–993.

Hagan, M.T., Demuth, H. B., Beale, M. H., et al., 2014. Neural network design, Martin Hagan.

Hu, Y. H., and Hwang, J. N., 2001. Handbook of neural network signal processing. CRC press.

Huang, M. L., and Yung-Yan, H., 2012. Fetal distress prediction using discriminant analysis, decision tree, and artificial neural network. Journal of Biomedical Science and Engineering 5, 526–533.

Jezewski, M., Czabanski, R., Wrobel, J., et al., 2010. Analysis of Extracted Cardiotocographic Signal Features to Improve Automated Prediction of Fetal Outcome. Biocybernetics and Biomedical Engineering 30(4), 29–47.

Karabulut, E. M., and Ibrikci, T., 2014. Analysis of Cardiotocogram Data for Fetal Distress Determination by Decision Tree Based Adaptive Boosting Approach. Computer and Communications 2(7), 32–37.

Lichman, M., 2013. {UCI} Machine Learning Repository. Available from http://archive.ics.uci.edu/ml.

Luenberger, D. G., Ye, Y., and others, 1984. Linear and nonlinear programming. Springer.

Marquardt, D., 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics, Society for Industrial and Applied Mathematics 11(2), 431–441.

Menai, M., Mohder, F., and Al-mutairi, F., 2013. Influence of Feature Selection on Naïve Bayes Classifier for Recognizing Patterns in Cardiotocograms. Journal of Medical and Bioengineering 2(1), 66–70.

Møller, M. F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural networks, Elsevier 6(4), 525–533.

Nunes, I. and Ayres-de-Campos, D., 2016. Computer analysis of foetal monitoring signals. Best Practice & Research Clinical Obstetrics & Gynaecology 30, 68–78.

Ocak, H., 2013. A medical decision support system based on support vector machines and the genetic algorithm for the evaluation of fetal well-being. J Med Syst 37(2), 9913.

Onnia, V., Tico, M., and Saarinen, J., 2001. Feature selection method using neural network. International Conference on Image Processing, 513–516.

Pinas, A., and Chandraharan, E., 2016. Continuous cardiotocography during labour: Analysis, classification and management. Best Practice & Research Clinical Obstetrics & Gynaecology 30, 33–47.

Powell, M. J. D., 1977. Restart procedures for the conjugate gradient method. Mathematical programming, Springer 12(1), 241–254.

Ravindran, S., Jambek, A. B., Muthusamy, H., et al., 2015. A novel clinical decision support system using improved adaptive genetic algorithm for the assessment of fetal well-being. Computational and mathematical methods in medicine, United States 2015, 283532.

Riedmiller, M., and Braun, H., 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Neural Networks, 1993. IEEE International Conference, 586–591.

Sahin, H., and Subasi, A., 2015. Classification of the cardiotocogram data for anticipation of fetal risks using machine learning techniques. Applied Soft Computing 33, 231–238.

Saini, L. M., and Soni, M. K., 2002. Artificial neural network-based peak load forecasting using conjugate gradient methods. IEEE Transactions on Power Systems, IEEE 17(3), 907–912.

Sundar, C., Chitradevi, M., and Geetharamani, G., 2012. Classification of cardiotocogram data using neural network based machine learning technique. International Journal of Computer Applications 47(14).

Tomáš, P., Krohová, J., Dohnálek, P., et al., 2013. Classification of cardiotocography records by random forest. 36th International Conference on Telecommunications and Signal Processing (TSP), 620–923.

Tongsong, T., Iamthongin, A., Wanapirak, C., et al., 2005. Accuracy of fetal heart-rate variability interpretation by obstetricians using the criteria of the National Institute of Child Health and Human Development compared with computer-aided interpretation. Journal of Obstetrics and Gynaecology Research, Australia 31(1), 68–71.

Wythoff, B. J., 1993. Backpropagation neural networks: a tutorial. Chemometrics and Intelligent Laboratory Systems, Elsevier 18(2), 115–155.

Yang, Z. and Zhidong, Z., 2017. Fetal State Assessment Based on Cardiotocography Parameters Using PCA and AdaBoost. 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics.

Yılmaz, E., 2016. Fetal State Assessment from Cardiotocogram Data Using Artificial Neural Networks. Journal of Medical and Biological Engineering, Springer Berlin Heidelberg, 1–13.