Applied Soft Computing 49 (2016) 399–406

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

ECG assessment based on neural networks with pretraining

Vicent J. Ribas Ripoll[a,*], Anna Wojdel[a], Enrique Romero[b], Pablo Ramos[c], Josep Brugada[c]

[a] Custom Software and Electronics, Marie Curie 8, 08042 Barcelona, Spain
[b] Soft Computing (SOCO) Research Group, Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Edifici Omega, Campus Nord, 08034 Barcelona, Spain
[c] Hospital Clínic, Universitat de Barcelona, Villarroel, 170, 08036 Barcelona, Spain

ARTICLE INFO

Article history:
Received 20 October 2015
Received in revised form 22 June 2016
Accepted 4 August 2016
Available online 10 August 2016

Keywords:
Neural networks
Pretraining
Restricted Boltzmann machines
Deep learning
Electrocardiography
Cardiology

ABSTRACT

In this paper, we present a new automatic screening method to assess whether a patient from ambulatory care or emergency should be referred to a cardiology service. This method is based on deep neural networks with *pretraining* and takes as an input a raw ECG signal without annotation.

This work is based on a prospective clinical study that took place at Hospital Clínic in Barcelona between 2011–2012 and recruited 1390 patients. For each patient, we recorded a 12-lead ECG and the diagnosis was conducted by the cardiology service at the same hospital. Normal, borderline normal and normal variant ECGs were labelled as *normal* and the rest as *abnormal*.

Our deep neural networks with pretraining were tested through cross-validation with a cohort of 416 test patients. The performance of our model was compared against other standard classification methods such as neural networks without pretraining, Support Vector Machines, Extreme Learning Machines, k-Nearest Neighbours and a professional classification algorithm certified for medical use that annotates the raw ECG signals prior to classification.

The resulting best classifier was a pretrained neural network with three hidden layers and 700 units in every layer. This network yielded an accuracy of 0.8552, a sensitivity of 0.9176 and a specificity of 0.7827. The best alternative classification method was a Support Vector Machine with a Gaussian kernel, which yielded an accuracy of 0.8476, a sensitivity of 0.9446 and a specificity of 0.7346. The professional classification algorithm yielded an accuracy of 0.8407, a sensitivity of 0.8558 and a specificity of 0.8214.

Neural networks with pretraining automatically obtain a representation of the input data without resorting to any annotation and, thus, simplify the process of assessing normality of ECG signals. The results that we have obtained are slightly better than those obtained with the professional classification system and, for some network configurations, they can be considered as exchangeable.

Neural networks with pretraining open up a promising line of research for the automatic assessment of ECG signals that may be used in the future in clinical practice.

1. Introduction

Cardiovascular disease (CVD) remains the main cause of death worldwide. According to the most recent statistics of the World Health Organisation (WHO), CVD mortality rates are expected to range between 246 deaths for 100,000 population in 2015 to 264 for 100,000 population in 2030 [1].

The early detection of CVD through ambulatory care services may significantly decrease time to treatment in hospital cardiology services and, consequently, improve patient outcomes while improving service productivity and keeping costs under control.

Electrocardiography (ECG) is a fast and reliable method for assessing cardiac function in general and arrhythmias in particular. However, the assessment of ECGs is challenging and diagnoses must be confirmed/checked by an expert cardiologist. During patient triage at an emergency service or an ambulatory cardiac examination, it is possible that some abnormalities may go undetected or that a normal ECG might be considered as pathological. Therefore, an accurate ECG triage tool is required to improve the decision process prior to referring a patient to a cardiology service. This triage tool must be accurate and present good sensitivity and specificity.

* Corresponding author.
E-mail address: vribas@tecnocse.com (V.J. Ribas Ripoll).

http://dx.doi.org/10.1016/j.asoc.2016.08.013
1568-4946/

Modern ECG equipment often include algorithms performing some kind of automatic interpretation of the electrocardiogram. Normally, these algorithms interpret the annotated ECG in order to predict the state of health of the patient (i.e. they classify the ECG using a set of extracted features from it).

The accuracy of the classifier depends on the signal quality and the extracted feature set. There are various methods available for building classification models for ECGs including Artificial Neural Networks [2–5], Support Vector Machines [6–8] or Hidden Markov Models, [9,10]. Other methods resort to k-Nearest Neighbour clustering for ECG assessment [11–14], Gaussian Mixture Models, Probabilistic Neural Networks [15] and Classification and Regression Trees [16,17]. It is important to note that these works target specific conditions such as atrial fibrillation, arrhythmia, myocardial infarction and heartbeat profiling.

Despite the fact that a lot of research has been done in ECG assessment, we believe that decision support for patient referral still needs to be addressed. For this reason, we designed a clinical study to obtain a challenging and generic database for training and testing our algorithms. The clinical relevance is also assessed through expert assessment and compared against the baseline provided by an automatic diagnostics system routinely used in clinical practice.

In this paper, we present a new screening/triage method to assess whether a patient should be referred to a cardiology service due to a pathological ECG signal regardless of their base pathology. This pathology shall be later assessed by the expert cardiologists.

In order to achieve this goal, we resort to a complex dataset that combines a myriad of heart pathologies with normal ECGs and ECGs of patients that recovered from an illness such as a heart attack. The ECGs from these patients present abnormalities that cannot be considered as pathological and thus are difficult to be classified as normal by machine learning models. This dataset reflects the reality faced by a doctor in terms of different pathologies from a big tertiary hospital in Barcelona providing assistance to over 1M users.

The method proposed here is based on deep neural networks with pretraining and takes as an input the raw ECG signal (i.e. without annotation) for further simplicity. Deep networks have recently emerged as powerful models due to their ability to excel on challenging tasks in different problems and settings [18–20]. The pretraining of our deep networks was performed with Restricted Boltzmann Machines.

This paper extends the results presented in [2]. In that paper we evaluated the importance of pretraining with shallow networks in order to obtain actionable results with a raw dataset without annotation. What we propose here is to test deep networks with and without pretraining to address the more ambitious objective of ECG screening that may be used in clinical practice. In this paper we take as a baseline other methods that have been used for ECG screening in the state-of-the-art such as Support Vector Machines (SVMs), Extreme Learning Machines (ELMs) and k-Nearest Neighbours (kNNs) and a professional classifier (Hannover Expert System – HES).

We tested different deep neural network architectures to determine the best network topology and configuration. The performance of these models was compared to other methods such as neural networks without pretraining, SVMs, ELMs and kNNs. In addition, the proposed models were further tested against the HES (® Corscience GmbH, Germany), a state-of-the-art ECG assessment algorithm used in clinical practice that annotates the raw ECG signal prior to classification. Out of the models tested in this paper, the best classifier in terms of accuracy was a pretrained neural network with three hidden layers and 700 units in every layer, which yielded an accuracy of 0.8552, a sensitivity of 0.9176 and a specificity of 0.7827.

This paper is organized as follows: Section 2 gives an overview of neural networks, Restricted Boltzmann Machines and the other models that we used for comparison. In Section 3, we present the dataset used in our experiments and show the experimental results for each model and configurations tested. In Section 4, we discuss the results and main limitations of this work. Section 5 presents the conclusions and gives an account of the main limitations of this study together with a short outline for future work.

2. Methods

2.1. Neural networks

A neural network is a set of simple interconnected processing units, where every connection has an associated weight. Neural networks are trained to optimize a certain cost function, which depends on the weights and the data, such as the cross-entropy or the sum-of-squares error. These cost functions are usually differentiable and the training process is guided by its derivatives with respect to the weights. The most widely used algorithm to compute these derivatives is Back-propagation [21].

Typically, the units of a neural network are structured in layers: the input layer, one or more hidden layers and the output layer. Usually, two adjacent layers are fully connected, and there are no connections between non-adjacent layers. The computation starts in the input layer, which propagates the data through the hidden layers to the output layer. Every unit receives the outputs of the units in the previous layer and transforms this information through simple computations according to its weights. The combination of these simple computations may result in very complex functions [22].

2.2. Shallow neural networks, deep neural networks and pretraining

The number of hidden layers of a neural network is a parameter of the model. Until recently, most successful neural networks were shallow, i.e., they had one or two hidden layers.[1] In fact, deeper architectures consistently yielded worse results, or at most similar, than shallow ones [23], since they converged to local minima with worse generalization. Therefore, deep networks were considered more difficult to train than shallow networks.

In the last years, however, deep networks have emerged as powerful models [24,25,23,26], outperforming shallow networks in different problems and settings [20]. One of the keys of this success is weight initialization. Typically, the standard training of a neural network starts with random weights, but in many cases this is not suitable for deep networks, that need more complex initialization procedures [27].

In this regard, many successful deep neural networks are trained in two steps: an unsupervised pretraining followed by a supervised fine-tuning [27]. The pretraining step is used to find a good set of initial weights. Fine-tuning is equivalent to the standard training procedure of a neural network, but in the case of deep learning it starts from the set of weights found by the pretraining step. This scheme has been successfully applied to different problems including regression [28] and classification [29].

Typically, unsupervised pretraining procedures try to capture information about the input data. An explanation about why unsupervised pretraining works is that unsupervised pretraining restricts the initial weights to belong to particular regions of weights that may capture some structure of the input distribution.

[1] These networks correspond to the classical neural network implementations found in the literature.

The solutions obtained during fine-tuning will be relatively close to the initial ones found in pretraining and thus result in better generalization than random initial weights [27].

One of the most common approaches for pretraining (and the first breakthrough in deep learning) is described in [24,25] and is based on an unsupervised generative model called Restricted Boltzmann Machine (RBM) [30]. Other pretraining algorithms are based on auto-encoders [23,26]. In the following we will describe some basic features of RBMs, which are the pretraining approach that we used in this study.

2.3. Restricted Boltzmann machines

RBMs are energy-based probabilistic models. In these models, a probability distribution is defined from an energy function of the form:

$$P(\boldsymbol{x}, \boldsymbol{h}) = \frac{e^{-\text{Energy}(\boldsymbol{x},\boldsymbol{h})}}{Z},$$

where $\boldsymbol{x}$ are input variables and $\boldsymbol{h}$ correspond to the hidden variables introduced to increase the expressive power of the model. The normalization factor Z is called the partition function:

$$Z = \sum_{\boldsymbol{x},\boldsymbol{h}} e^{-\text{Energy}(\boldsymbol{x},\boldsymbol{h})}.$$

There are different kinds of RBMs depending on the input variables. The original implementation of RBMs has binary inputs and binary hidden variables, with the following energy function [30]:

$$\text{Energy}(\boldsymbol{x}, \boldsymbol{h}) = -\boldsymbol{b}'\boldsymbol{x} - \boldsymbol{c}^t\boldsymbol{h} - \boldsymbol{h}'\boldsymbol{W}\boldsymbol{x}. \tag{1}$$

For real-valued conditional Gaussian input variables and binary hidden variables, the energy function suggested in [31] is the following:

$$\text{Energy}(\boldsymbol{x}, \boldsymbol{h}) = \frac{1}{2}\left|\left|\frac{\boldsymbol{x}-\boldsymbol{b}}{\sigma}\right|\right|^2 - \boldsymbol{c}'\boldsymbol{h} - \boldsymbol{h}'\boldsymbol{W}\frac{\boldsymbol{x}}{\sigma}. \tag{2}$$

The RBMs with an energy function as defined in Eq. (2) are often referred as Gaussian–Bernoulli RBMs.

The main characteristics of the energy functions in (1) and (2) are twofold [32]. First, $P(\boldsymbol{h}|\boldsymbol{x})$ and $P(\boldsymbol{x}|\boldsymbol{h})$ have a very simple expression that allows to work with RBMs similarly to standard neural networks. Second, and more important, since both $P(\boldsymbol{h}|\boldsymbol{x})$ and $P(\boldsymbol{x}|\boldsymbol{h})$ factorize, it is possible to compute $P(\boldsymbol{h}|\boldsymbol{x})$ and $P(\boldsymbol{x}|\boldsymbol{h})$ in one step. Thus, it possible to efficiently perform Gibbs sampling [33] over these probability functions. This is the basis of Contrastive Divergence (CD) [34], which is the most common learning algorithm for RBMs. The section below provides a short overview of CD.

2.4. Contrastive divergence

When we have a parametric probabilistic model, the training of RBMs aims at maximizing the log-likelihood of a given dataset $\boldsymbol{X}$:

$$\text{LogLikelihood}(\boldsymbol{X}) = \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \log P(\boldsymbol{x}_i)$$

The derivative of this log-likelihood is

$$\frac{\partial \log P\left(\boldsymbol{x};\theta\right)}{\partial\theta} = -E_{P(\boldsymbol{h}|\boldsymbol{x})}\left[\frac{\partial \text{Energy}\left(\boldsymbol{x},\boldsymbol{h}\right)}{\partial\theta}\right] + E_{P(\tilde{\boldsymbol{x}})}\left[E_{P(\boldsymbol{h}|\tilde{\boldsymbol{x}})}\left[\frac{\partial \text{Energy}(\tilde{\boldsymbol{x}},\boldsymbol{h})}{\partial\theta}\right]\right].$$

The first term in the expression above can be efficiently computed, but the second term is computationally intractable. Therefore, it is not possible to calculate the exact derivative of the log-likelihood in a reasonable time. In [34], CD was proposed as an efficient method for training RBMs.[2] CD_n estimates the derivative of the log-likelihood as

$$\frac{\partial \log P(\boldsymbol{x}_i;\theta)}{\partial\theta} \sim eq - E_{P(\boldsymbol{h}|\boldsymbol{x}_i)}\left[\frac{\partial \text{Energy}\left(\boldsymbol{x}_i,\boldsymbol{h}\right)}{\partial\theta}\right] + E_{P(\boldsymbol{h}|\boldsymbol{x}_i^n)}\left[\frac{\partial \text{Energy}(\boldsymbol{x}_i^n,\boldsymbol{h})}{\partial\theta}\right]$$

where $\boldsymbol{x}_i^n$ is the last sample of the Gibbs chain of length n starting from $\boldsymbol{x}_i \in \boldsymbol{X}$:

$$\boldsymbol{h}_i^1 \sim P(\boldsymbol{h}|\boldsymbol{x}_i); \quad \boldsymbol{x}_i^1 \sim P(\boldsymbol{x}|\boldsymbol{h}_i^1); \cdots \boldsymbol{h}_i^n \sim P(\boldsymbol{h}|\boldsymbol{x}_i^{n-1}); \quad \boldsymbol{x}_i^n \sim P(\boldsymbol{x}|\boldsymbol{h}_i^n).$$

This computation can be performed efficiently because, as previously mentioned, $P(\boldsymbol{h}|\boldsymbol{x})$ and $P(\boldsymbol{x}|\boldsymbol{h})$ factorize. In practice, it has also been observed that, even with $n = 1$, CD_1 works well [34,24,23,32].

2.5. Support vector machines

In this paper, the performance of the methods presented above was compared in their ability to predict ECG normality against the soft margin SVM [35]. In the SVM approach, the objective is to obtain a hyper-surface separating the training points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with labels $\mathbf{Y} = \{y_1, \ldots, y_N\}$ into two disjoint sets, one for each class studied. The separating hyper-surface is required to have maximum margin. Soft margin SVMs [35] let some points fall on the incorrect side of the margin boundary by introducing a penalty that increases with the distance from this margin (i.e., the greater the misclassification, the bigger the error). In one-norm soft margin SVMs, this is achieved by solving the following quadratic problem:

$$\underset{\alpha}{\text{argmax}}\left(\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\alpha^t\mathbf{K}\alpha\right) \quad \text{s.t. } 0 \leqslant \alpha_i \leqslant C \text{ and } \sum_{i=1}^{N}\alpha_i y_i = 0. \tag{3}$$

where $K_{ij} = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$ and $k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function. With these definitions, the output of a SVM has the same structure than that of a neural network where the hidden-layer weights are a subset of the training set (the support vectors) and the output-layer weights are the solution of Eq. (3). Parameter C controls the trade-off between the penalty and the size of the margin and has a strong influence in the number of support vectors.

2.6. Extreme learning machines

The proposed methods were also compared with ELMs [36]. The main idea underlying ELMs is to use random weights in the hidden layers and compute the output-layer weights with quadratic techniques. These models are based on the results in [37], where it is shown that neural networks with random weights in the hidden layer are universal approximators for different activation functions.

The original algorithm for ELMs can be described as follows. Given a set of training examples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, their labels $\mathbf{Y} = \{y_1, \ldots, y_N\}$, a hidden-layer activation function φ and an *a priori* fixed number N_h of hidden units:

[2] RBMs are a special case of a product-of-experts model.

(1) Randomly assign hidden-layer weights and biases ($\boldsymbol{\omega}_i$, b_i), $i = 1, \ldots, N_h$.
(2) Compute the $N \times N_h$ hidden-layer output matrix $\mathbf{H}$ ($\mathbf{H}_{ji} = \varphi(\mathbf{x}_j, \boldsymbol{\omega}_i, b_i)$).
(3) Calculate the output-layer weight vector $\boldsymbol{\lambda} = \mathbf{H}^{\dagger}\mathbf{Y}$, where $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is the pseudo-inverse (aka Moore-Penrose generalized inverse) of the hidden-layer output matrix $\mathbf{H}$.

With these definitions, the output of the network for the training set is $\mathbf{H}\boldsymbol{\lambda}$, which minimizes the sum-of-squares cost for the assigned hidden-layer weights.

3. Results

3.1. Dataset description

Our study was conducted at the Cardiology service of Hospital Clínic in Barcelona (Spain) during years 2011 and 2012. The study was approved by the clinical research ethics committee (reference 2011/7044). The need for informed consent was waived.

The dataset used in our study consists of 12-lead digital ECGs recorded in resting position (with GE and Custo Med equipment). All subjects included in our study were selected at random from all cardiology patients without any patient-specific screening. No discrimination was made in regards to age, gender, or base pathology. The only criteria used in our ECG selection was related to its readability in terms of noise levels from the recorded sample. ECG selection was performed by cardiologists from the same service. The dataset includes a single ECG for each patient, which is at least 10 s long.

In total, our dataset consists of 12-lead ECG signals from 1390 subjects (749 male and 641 female). The age of the subjects ranged between 1 month and 94 years with a median of 63 years old and inter-quantile range (IQR) of 23 years.

During the recruitment phase of our study, each ECG recording was carefully analysed and a complete diagnosis given by the team of cardiologists at Hospital Clínic. Each report contains all abnormalities detected on each ECG recording. The diagnosis was given as a textual description and was further labelled according to a predefined list of codes. In our case, we used the ECG diagnosis codes used by the Mayo Clinic, as presented in [38].

Following this procedure, the cardiology service applied 61 different diagnosis codes to describe the heart conditions affecting the patients recruited in our study. Table 1 summarizes the pathologies occurring in the study listed in 13 groups related to their general characteristics. Number of cases in each table row represents the occurrences of all conditions (diagnosis codes) joined in each descriptive group. The most prevalent conditions in our dataset are: sinus bradycardia (code 2A, 154 cases), atrial fibrillation (code 2S, 109 cases) and complete right bundle branch block (code 7B, 101 cases). These three conditions are highlighted in their corresponding groups in Table 1.

Table 1
Number of ECG diagnoses belonging to each of the descriptive groups making occurrence in our dataset. Highlighted conditions are the most prevalent in our data set (complete right bundle branch block belongs to the marked group with asterisk).

Diagnosis group	Cases
sinus: arrhythmia, **bradycardia**, tachycardia	209
atrial: ectopic rhythm, tachycardia, premature complexes	36
atrial flutter and **atrial fibrillation**	120
ventricular escape or premature complexes	63
AV conduction abnormalities	89
intra-ventricular conduction disturbances (*)	249
P wave abnormalities	56
QRS low voltage, QRS axis deviation	48
ventricular hypertrophy	31
myocardial infraction	126
ST-, T-, and U-wave abnormalities	235
pacemaker rhythm and malfunction	44
other (e.g. Brugada syndrome)	15
ECGs with diagnostic code 1A (normal)	758
ECGs with diagnostic code 1B (borderline)	14

The classification of each ECG in the *normal* or *abnormal* group was based on the final diagnosis provided by the cardiologists. All ECGs with diagnostic codes containing 1A (Normal ECG) or 1B (borderline normal ECG or normal variant) were labelled as *normal* and the rest of ECGs as *abnormal*. In total, our dataset contained 772 *normal* and 618 *abnormal* recordings.

It is important to emphasise that a single ECG can be affected by more than one pathological condition. Thus, the complete description of the ECG pattern provided by the cardiology service may include several diagnosis codes. In fact, most of our reports contain more than one diagnosis code (up to 8 in the most extreme cases). This does not only apply to the ECG patterns labelled as abnormal but also to the ECGs considered as normal (18% of normal ECGs were additionally marked with some other diagnostic codes). The most important co-occurrences affecting the ECGs that have been labelled as normal are sinus arrhythmias, sinus bradycardia and tachycardia, atrial or ventricular premature complexes and ST, or T-wave abnormalities like early repolarization.

The myriad of pathologies available in the database and the presence of non-pathological wave abnormalities in the ECGs labelled as normal, result in a very challenging dataset for evaluating the methods presented in this paper. In particular, we have evaluated the accuracy, sensitivity and specificity for each model.

3.2. Signal processing

ECGs were recorded at two different sampling frequencies (1000 Hz and 500 Hz). Therefore, data rates were unified through resampling all ECGs to 250 Hz (i.e. we applied a resampling factor of 2 or 4 depending on the original sampling frequency) after filtering the signal with an anti-aliasing Butterworth filter. After this processing, the baseline was removed by firstly applying a 20th order low-pass Butterworth filter with a cut-off frequency of 150 Hz to all ECG channels. Other baseline elements like respiration and other low-frequency artefacts were removed by fitting a 6th order polynomial to each ECG channel.

After this pre-processing, the QRS complexes for each channel were detected through automatic peak detection. After that, a 700 ms asymmetric window for each beat is obtained by taking 500 ms of the signal after the R peak and 200 ms before. Finally, we averaged the windows calculated previously to remove noise and obtain a clean ECG pulse for each channel (i.e. each pulse has 175 samples). The mean pulses for each channel were concatenated into a vector and used as an input for training and evaluating our models.[3] In summary, the dataset used to construct the classifiers had 1390 examples and 2100 variables.

3.3. Models tested and experimental setting

We assessed the performance of shallow and deep neural networks with and without RBM-based pretraining with CD. Since our input variables were real-valued and Gaussian, pretraining was performed with Gaussian-Bernoulli RBMs, whose energy function is defined in Eq. (2) above. The hidden units in the rest of layers were logistic. Networks were pretrained with stochastic gradient ascent (mini-batches of size 32), estimating the derivative of the

[3] Note that we have 12 channels × 175 samples = 2100 samples.

Table 2
Three best mean test results for each network architecture with pretraining.

Pretraining	Hidden layers	Hidden units	Acc.	Sens.	Spec.
y	1	700	0.8538	0.9149	0.7827
y	1	500	0.8528	0.9117	0.7843
y	1	600	0.8523	0.9068	0.7890
y	2	700	0.8542	0.9041	0.7963
y	2	500	0.8542	0.8838	0.8199
y	2	300	0.8538	0.8923	0.8089
y	3	700	**0.8552**	**0.9176**	**0.7827**
y	3	600	0.8545	0.9009	0.8005
y	3	400	0.8528	0.9041	0.7932
y	4	700	0.8492	0.8923	0.7990
y	4	800	0.8489	0.9000	0.7895
y	4	600	0.8487	0.9059	0.7822

Table 3
Three best mean test results for each network architecture without pretraining.

Pretraining	Hidden layers	Hidden units	Acc.	Sens.	Spec.
n	1	300	0.8230	0.8604	0.7796
n	1	500	0.8220	0.8950	0.7372
n	1	100	0.8218	0.8689	0.7670
n	2	700	0.8281	0.8892	0.7571
n	2	500	0.8242	0.8802	0.7592
n	2	600	0.8237	0.8919	0.7445
n	3	700	0.8196	0.8959	0.7309
n	3	500	0.8191	0.8752	0.7539
n	3	600	0.8145	0.8477	0.7759
n	4	600	0.8196	0.8734	0.7571
n	4	800	0.8189	0.8622	0.7686
n	4	700	0.8179	0.8599	0.7691

log-likelihood with CD_1 (see [32] for further details). The initial weights were drawn from a zero-mean normal distribution with standard deviation 0.1. The RBMs were trained for 50 epochs and a momentum of 0.8. An additional weight decay term was also added. The learning rate and the weight decay factor were selected through a grid search (the learning rate varying in {0.001, 0.005, 0.01, 0.05, 0.1} for every layer and the weight decay factor in {0.001, 0,01, 0.1}). When no pretraining was performed, the initial random weights were the starting point for fine-tuning. A softmax output layer was added prior to fine-tuning and its weights were also initially drawn from a zero-mean normal distribution with standard deviation 0.1. The fine-tuning step was done with stochastic gradient descent (mini-batches of size 32), computed with standard back-propagation and a cross-entropy objective function (see [39] for further details). The learning rate, momentum and weight decay factor were set to 0.01, 0.9 and 0.001, respectively. The fine-tuning was also performed for 50 epochs. Regarding the architectures, we tested networks varying their number of hidden layers from one to four with the same number of hidden units in every layer, from 100 to 900.

Neural network classifiers were also compared with other standard classification methods such as one-norm soft margin SVMs with a Gaussian kernel [35], ELMs [36] and kNNs. For SVMs, the parameter γ of the Gaussian kernel and the C value were selected through a grid search ($\gamma \in \{0.0001, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1\}$ and $C \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$). For ELMs, the same number of hidden units were tested as we did for the standard networks. ELMs were also implemented with logistic units. The gain factor γ of the logistic function was also selected through a grid search with the same values than for SVMs. For kNNs, k varied from 1 to 9. The performance of our ECG assessment algorithms was tested using cross-validation. The data set was randomly split with balanced classes into a training dataset with 974 subjects (70% of data) and a test set with 416 patients (30% of data). This procedure was repeated 10 times (with different random partitions), and implemented inside the grid search for tuning the parameters of each model. Before each cross-validation iteration, data was scaled to zero mean and unit variance.

3.4. ECG assessment

We have ranked the performance of our classifiers in terms of accuracy, sensitivity and specificity. Tables 2 and 3 show the best results that we obtained during our grid search with cross-validation for the neural network models. More precisely, the three best mean test set results for every number of hidden layers trained, with and without pretraining, are shown in Tables 2 and 3 respectively. The best network that we obtained with this procedure was a pretrained neural network with three layers and 700 hidden units in each layer. This model yielded an accuracy of 0.8552 a sensitivity of 0.9176 and a specificity of 0.7827. The best network without pretraining obtained an accuracy of 0.8281, a sensitivity of 0.8892 and a specificity of 0.7571.

Table 4
Best mean test results for the rest of the models tested.

Model	Parameters	Acc.	Sens.	Spec.
SVMs	$\gamma = 0.001$, $C = 0.1$	0.8474	0.9446	0.7346
ELMs	$\gamma = 0.005$, 100 hidden units	0.8068	0.8883	0.7720
kNNs	$k = 3$	0.7738	0.8162	0.6084
HES	–	0.8407	0.8558	0.8214

The results of the rest of the models are shown in Table 4. The SVMs with a Gaussian kernel yielded an accuracy of 0.8474, a sensitivity of 0.9446 and a specificity of 0.7346. The accuracy that we obtained with ELMs was 0.8068, a sensitivity of 0.8883 and a specificity of 0.772. Finally, kNNs yielded an accuracy of 0.7738 a sensitivity of 0.8162 and a specificity of 0.6084.

The performance of our classifiers was also compared against HES. This algorithm is FDA and CE marked for medical use. For our dataset, the HES algorithm yielded an accuracy of 0.8407, a sensitivity of 0.8558 and a specificity of 0.8214 (see Table 4). From Table 2, we see that a neural network with two layers of 500 hidden units and pretraining yields an accuracy of 0.8542, a sensitivity of 0.8838 and a specificity of 0.8199, which is almost exchangeable with HES.

Training times were different for every method. Typically, neural networks were the most expensive ones (with differences depending on the number of hidden layers and whether pretraining was used or not). As an example, Table 5 shows the training times of one run for the models with 700 hidden units, from one to four hidden layers, with and without pretraining. As it can be seen, adding one hidden layer roughly increases between 40% and 25% the computational cost, depending on the number of layers (the increment is smaller when more hidden layers are present). The pretraining procedure takes slightly more than half of the whole

Table 5
Reference training times of neural networks models with 700 hidden units in every layer, with and without pretraining.

Pretraining	Hidden layers	Hidden units	Training times
y	1	700	216.70 s
y	2	700	307.42 s
y	3	700	401.61 s
y	4	700	494.56 s
n	1	700	90.40 s
n	2	700	130.23 s
n	3	700	176.49 s
n	4	700	225.09 s

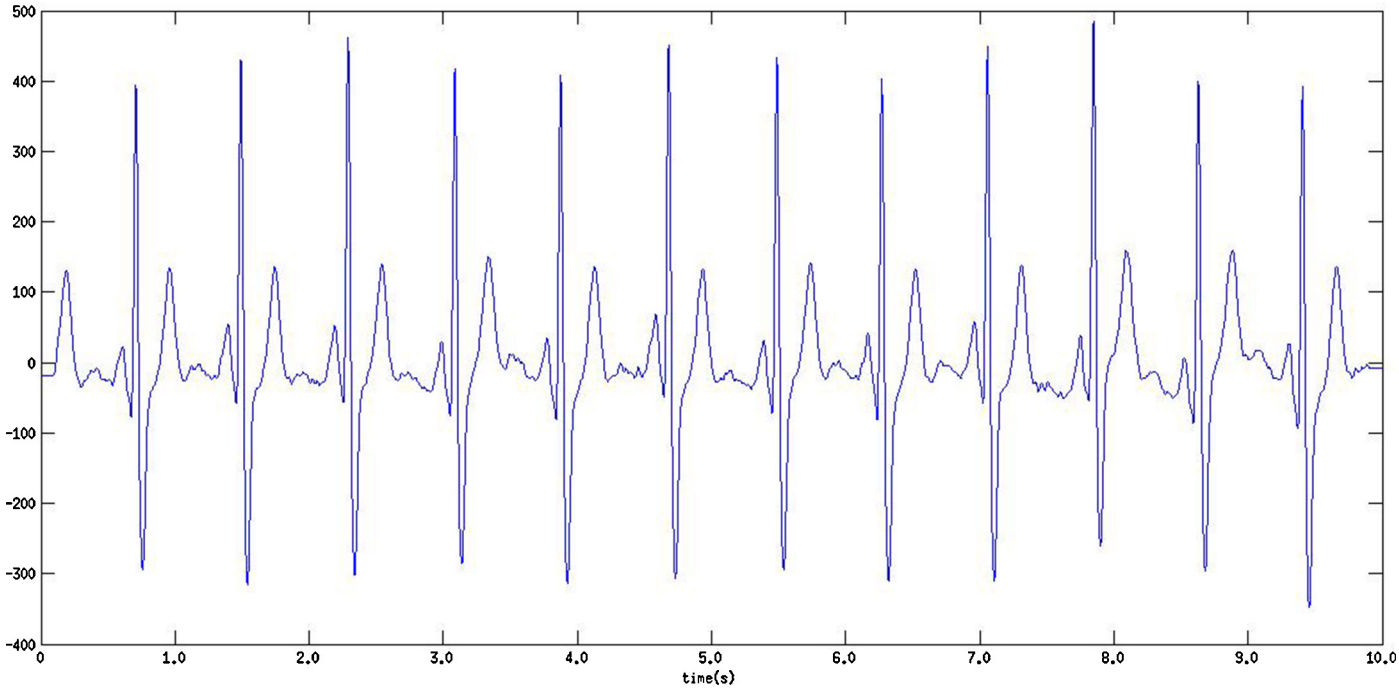

Fig. 1. ECG signal with complete right bundle branch block and left anterior fascicular block. The *x* axis corresponds to time in seconds and the *y* axis to potentials measured in *mV*.

running time. For this data set, the training times of SVMs were one order of magnitude smaller, and ELMs and kNNs were two order of magnitude faster than neural networks models. Testing times, in contrast, were very similar among all methods (around one second for all methods).

4. Discussion

In this paper, neural networks with different architectures were trained with RBM-based pretraining and applied to a data set of 1390 pre-processed 12 lead ECGs. The deep networks presented

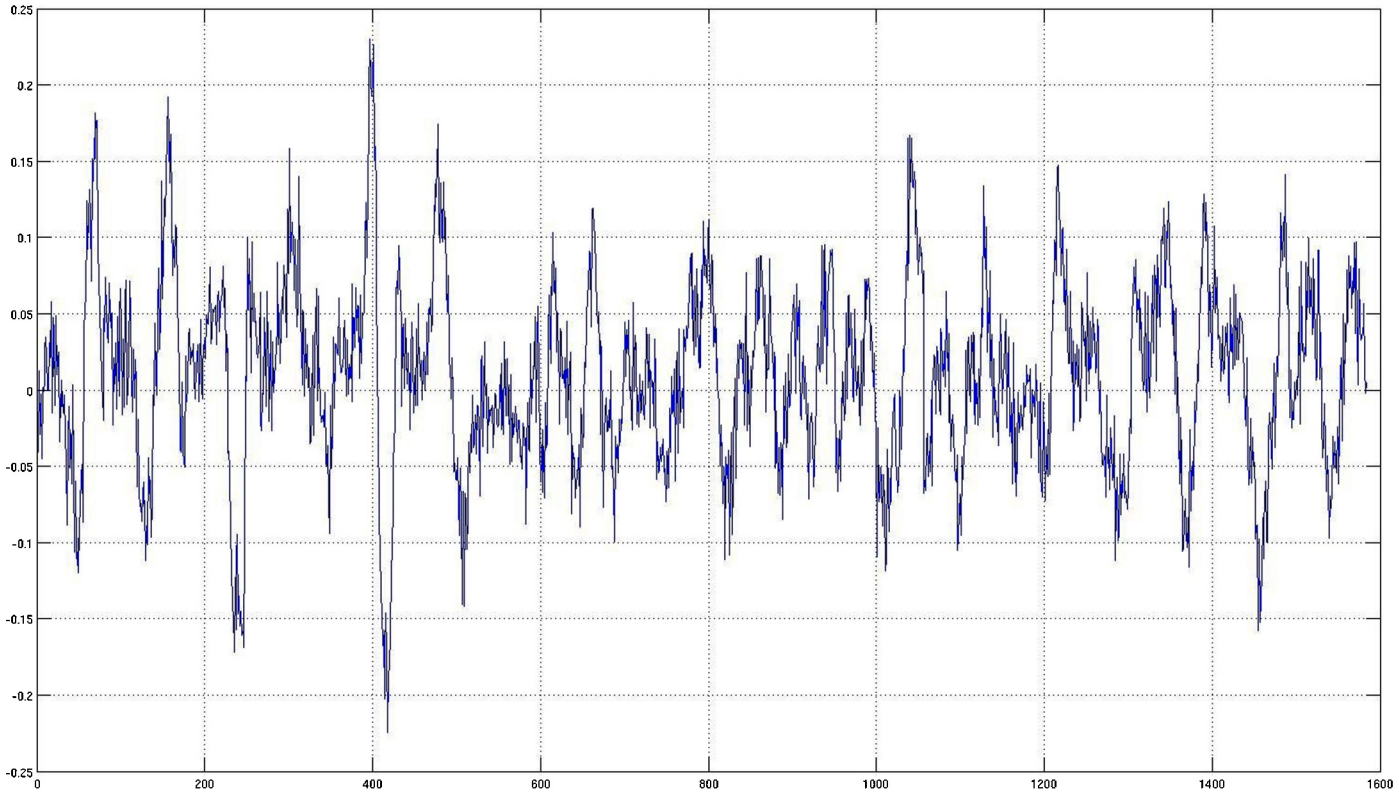

Fig. 2. First layer weights of a deep network with pretraining. The patterns obtained are very similar to ECG segments. This underlying structure, from a signal processing standpoint, could be understood as bench of adapted filters to the input signals (i.e. output maximized correlation).

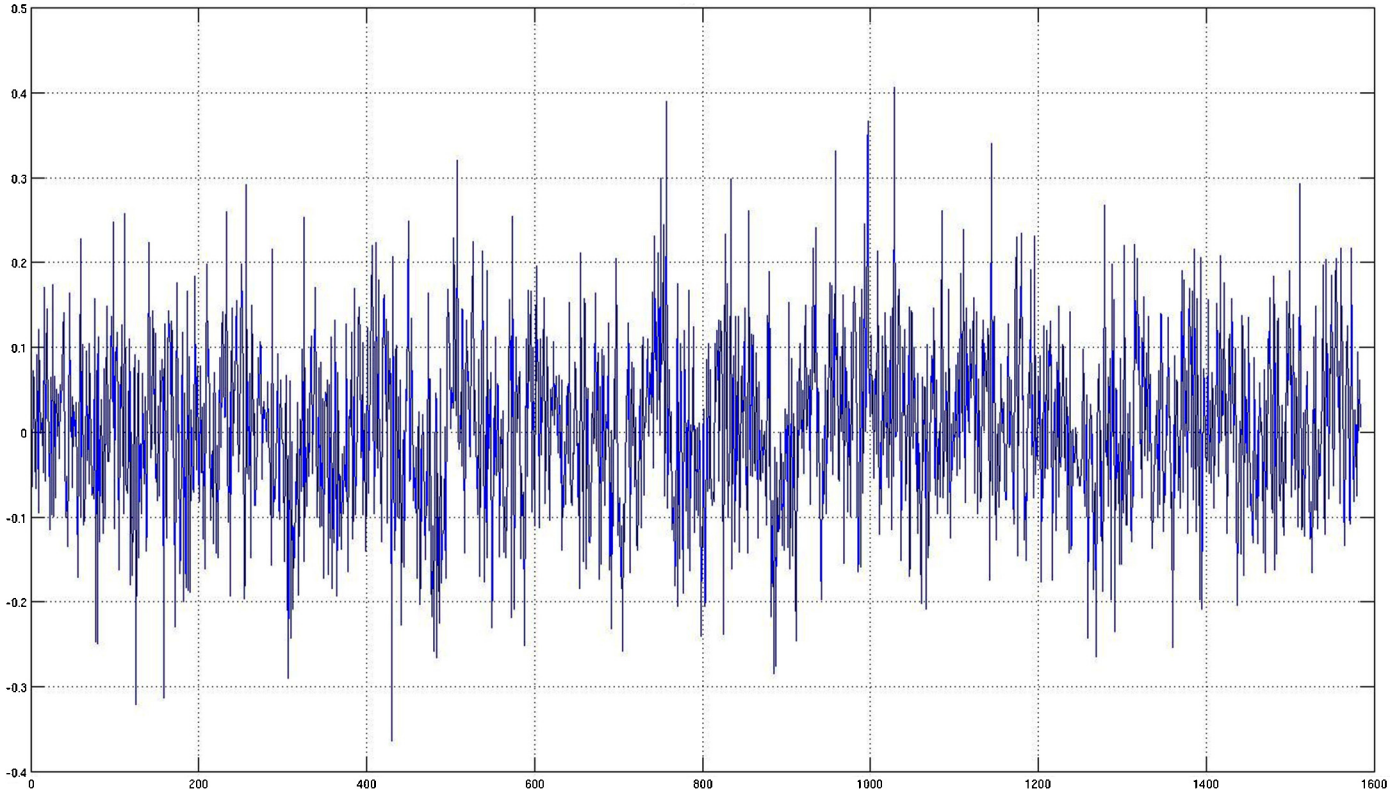

Fig. 3. Shallow network weights without pretraining. These output weights convey no information about the input signal.

here provide accurate and medically actionable results for ECG screening while keeping an acceptable balance between the different parameters of interest (accuracy, sensitivity and specificity). The generalization capabilities of the networks studied in this paper are acceptable and outperform other standard methods such as HES, neural networks without pretraining, SVMs, ELMs and kNNs.

The pretraining procedure clearly plays an important role in improving the accuracy of the classifiers developed and helps to balance the rate between true and false positives. However, this improvement in accuracy, sensitivity and specificity comes at a high computational cost for training the deep neural networks.

Increasing the number of layers and hidden units over those providing the best results does not improve the performance of the resulting classifier. The optimum results from our grid search are presented in the tables below.

The role of pretraining is better understood by inspecting the weights learned by the networks (the weights store the knowledge that the neural network has learned from the data). Fig. 1 depicts an example of ECG signal and in Fig. 2, we show the weights associated to one hidden unit in the network pretrained with the raw data that gave the best results. As it can be seen, the neural network has managed to obtain ECG sensitive neurons capable of discriminating pathological traits in the ECG signal. For the sake of comparison, we also present in Fig. 3 the weights of the same neural network without pretraining. It is apparent that the 'learned' traits are much noisier than those obtained with pretraining.

SVMs are easier to implement and train than neural networks but present a slightly lower accuracy and specificity when compared against the best deep neural network obtained (0.8476 vs. 0.8552 and 0.7346 vs. 0.7827). The SVM approach shall be considered as a baseline comparison for future work since it also provides a good alternative to the professional classification algorithm that we evaluated in this paper.

All methods presented in this paper have as an input a raw 12-lead ECG signal. This further simplifies the system preprocessing but with the limitation that equivalent sampling rates and analysis window lengths must be ensured throughout the whole processing and analysis.

The main limitation in our study comes from the signal processing techniques that we have used. More specifically, the asymmetric windowing and averaging of QRS complexes results in a loss of information about heart rhythm that may be useful for assessing some syndromes or life-threatening events. For example, in our test dataset, we misclassified three atrial fibrillations and a couple of premature complexes. Despite the fact that the results are very promising, we believe that there is still space for improving the classification by using other deep learning techniques such as convolutional networks that can take as an input the 12 ECG channels without any processing or combining deep networks with other signal processing techniques such as statistical power spectral analysis that better capture signal variability. We leave these approaches as future work.

5. Conclusions

Assessing the normality of ECGs is an important task in the cardiology work-flow. Some illnesses and syndromes are very difficult to diagnose even for trained cardiologists. For this reason, a system that simplifies this process and supports the decision of referring a patient to a cardiology service may be very useful in different clinical settings such as ambulatory care, emergency, or internal medicine.

Sensitivity and specificity are important measures of performance when predicting ECG normality because timely referral to a cardiology service may result in better outcomes for high-risk patients. Moreover, better decisions result in better productivity

and workload control in the already overwhelmed cardiology services around the world. The deep neural networks presented in this paper might address this issue by providing a simple yet accurate system that can be used in medical practice for referring to the cardiology service those patients that really need assistance (i.e. improved detection rates whilst keeping the number of false positives under control).

Finally, the results obtained with these models are promising if compared with professional classification methods such as HES. The deep networks presented here may be useful for ECG screening in clinical practice. However, a word of caution must be given and further multi-centric tests are required before a screening system as the one we present here can be used in a real clinical setting.

Acknowledgements

This project has been partially funded by the Shockomics project, under the EU FP7 framework.

References

[1] Global health observatory data repository, http://www.who.int, (accessed 18.07.13).
[2] V.J. Ribas, A. Wojdel, P. Ramos, E. Romero, J. Brugada, Assessment of electrocardiograms with pretraining and shallow networks, in: Computing in Cardiology, vol. 41, IEEE, 2014, pp. 1061–1064.
[3] N. Murthy, M. Meenakshi, Comparison between ANN-based heart stroke classifiers using varied folds data set cross-validation, in: L.C. Jain, P.L. Srikanta, I. Nikhil (Eds.), Intelligent Computing, Communication and Devices, vol. 308 of Advances in Intelligent Systems and Computing, Springer, India, 2015, pp. 693–699.
[4] M. Mitra, R.K. Samanta, Cardiac arrhythmia classification using neural networks with selected features, Procedia Technol. 10 (2013) 76–84, First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA).
[5] D. Patra, M.K. Das, S. Pradhan, Integration of FCM, PCA and neural networks for classification of ECG arrythmias, IAENG Int. J. Comput. Sci. 36 (2005) 1–5.
[6] M. Moavenian, H. Khorrami, A qualitative comparison of artificial neural networks and support vector machines in ECG arrhythmias classification, Expert Syst. Appl. 37 (2010) 3088–3093.
[7] A. Yildiz, M. Akm, M. Poyraz, An expert system for automated recognition of patients with obstructive sleep apnea using electrocardiogram recordings, Expert Syst. Appl. 38 (2011) 12880–12890.
[8] S. Karpagachelvi, M. Arthanari, M. Sivakumar, Classification of ECG signals using extreme learning machine, Comput. Inf. Sci. 4 (2011) 42–52.
[9] J. Dumont, A.I. Hernandez, J. Fleureau, G. Carrault, Modelling temporal evolution of cardiac electrophysiological features using hidden semi-Markov models, in: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2008, pp. 165–168.
[10] R.V. Andreão, S.M.T. Muller, J. Boudy, B. Dorizzi, T.F. Bastos-Filho, M. Sarcinelli-Filho, Incremental HMM training applied to ECG signal analysis, Comput. Biol. Med. 38 (2008) 659–667.
[11] I.C.G. Gómez-Herrero, V. Krasteva, I. Jekova, A. Gotchev, K. Egiazarian, Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification, Med. Eng. Phys. 28 (2006) 876–887.
[12] A. Lanatá, G. Valenza, C. Mancuso, E.P. Scilingo, Robust multiple cardiac arrhythmia detection through bispectrum analysis, Expert Syst. Appl. 38 (2011) 6798–6804.
[13] S. Kiranyaz, T. Ince, J. Pulkkinen, M. Gabbouj, Personalized long-term ECG classification: a systematic approach, Expert Syst. Appl. 38 (2011) 3220–3226.
[14] R.J. Martis, U.R. Acharya, H. Prasad, K.Ch. Chua, C.M. Lim, J.S. Suri, Application of higher order statistics for atrial arrhythmia classification, Biomed. Signal Process. Control 8 (2013) 888–900.
[15] D. Giri, U.R. Acharya, R.J. Martis, S.V. Sree, T.-C. Lim, V.I.A. Thajudin, J.S. Suri, Automated diagnosis of coronary artery disease affected patients using LDA, PCA, ICA and discrete wavelet transform, Knowl. Based Syst. 37 (2013) 274–282.
[16] J. Fayn, A classification tree approach for cardiac ischemia detection using spatiotemporal information from three standard ECG leads, IEEE Trans. Biomed. Eng. 58 (2011) 95–102.
[17] L. Pecchia, P. Melillo, M. Bracale, Remote health monitoring of heart failure with data mining via CART method on HRV features, IEEE Trans. Biomed. Eng. 58 (2011) 800–804.
[18] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3642–3649.
[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, vol. 24, 2012, pp. 1106–1114.
[20] Y. Bengio, Deep learning of representations: looking forward, in: Statistical Language and Speech Processing, Springer, 2013, pp. 1–137.
[21] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[22] S. Haykin, Neural Networks. A Comprehensive Foundation, 2nd ed., Prentice Hall International, 1999.
[23] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in Neural Information Processing Systems, vol. 19, MIT Press, 2007, pp. 153–160.
[24] G.E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[25] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[26] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: 25th International Conference on Machine Learning, 2008, pp. 1096–1103.
[27] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11 (2010) 625–660.
[28] J.C. Ruiz-Rodríguez, A. Ruiz-Sanmartín, V. Ribas, J. Caballero, A. García-Roche, J. Riera, X. Nuvials, M. de Nadal, O. de Sola-Morales, J. Serra, J. Rello, Innovative continuous non-invasive cuffless blood pressure monitoring based on photoplethysmography technology, Intensive Care Med. 39 (9) (2013) 1618–1625.
[29] Q.V. Le, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, A.Y. Ng, Building high-level features using large scale unsupervised learning, in: International Conference on Machine Learning (ICML), ICML/Omnipress, 2012.
[30] P. Smolensky, Chapter 6: Information processing in dynamical systems: foundations of harmony theory, in: D.E. Rumelhart, J.L. McClelland (Eds.), in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, MIT Press, 1986, pp. 194–281.
[31] R.R. Salakhutdinov, G.E. Hinton, Using deep belief nets to learn covariance kernels for Gaussian processes, in: Advances in Neural Information Processing Systems, vol. 20, MIT Press, 2008, pp. 1249–1256.
[32] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (2009) 1–127.
[33] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 721–741.
[34] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (2002) 1771–1800.
[35] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, UK, 2000.
[36] G. Huang, G.B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32–48.
[37] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (2006) 879–892.
[38] J.G. Murphy, M.A. Lloyd, Mayo Clinic Cardiology: Concise Textbook, Taylor & Francis, 2006.
[39] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: Neural Networks: Tricks of the Trade, Second ed., Springer, Berlin, Heidelberg, 2012, pp. 437–478.