RESEARCH

ECG Annotation and Diagnosis Classification Techniques

Yan Yan* and Lei Wang

*Correspondence:
yan.yan@siat.ac.cn
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Xueyuan, Shenzhen, China
Full list of author information is available at the end of the article
†Equal contributor

Abstract

First part title: Text for this section.

Second part title: Text for this section.

Keywords: sample; article; author

Background

The heart is comprised of myocardium which rhythmically contract and thus drive the circulation of blood throughout the human body. A wave of electrical current passes through the entire heart, which triggers myocardial contraction [1]. Electrical propagation spreads over the whole heart in a coordinated pattern generate changes on the body surface potentials which can be measured and illustrated as an electrocardiogram (ECG, or sometimes EKG). Metabolic abnormalities (a lack of oxygen, or ischemia etc.) and pathological changes of the heart engender variety of ECG, consequently ECG analysis has been a routine part of any complete medical evaluation or healthcare applications.

Automated ECG analysis provides indispensable assist in clinical monitoring, a large number of approaches have been proposed for the task, basically the diagnosis of arrhythmic and further the inspection of heart rate variability or heart turbulence analysis [2]. Lots of ECG annotation and diagnosis classification techniques had been proposed in industrial circles and academic communities. As the general steps in a classification problem in a machine learning task, the ECG classification includes data collection, preprocessing, feature extraction, and classification with a classifier. Most of literatures described models which were combined by different classifier with features which extracted from different feature extraction algorithms. The ECG classification methods develops at the same pace with the development of classification theories in machine learning and pattern recognition. Because of the particularity in medical data collection and data annotation, the developments in ECG classification and detection were not as flourishing as the similar research topics like speech recognition, natural language processing and image processing etc.

In this chapter, we first introduce the basic elements and procedures in a typical ECG classification task, then we would review the proposed literatures of ECG classification, in the last we would introduce a new method in unsupervised learning for ECG classification.

Technology Roadmap

ECG classification methods had been developed for decades. With the development of theories in machine learning and data mining, lots of algorithms had been adopted in this domain. Before the review about the methods, it is quite necessary to mention the common experiment settings and data sets, as well as the framework.

ECG Acquisition

Acquiring and storing ECG data were the base for a analyzing task. Errors might be creep into an analysis at any possible stage, thus not only the hardware acquisition system, but also the transmission and storage should be carefully designed. The explantation for the acquisition field could be found in [3]. A raw data acquisition task related the digital signal processing and hardware design knowledges would not be further discussed in this chapter, in [4] a typical ECG signal acquisition process was illustrated.

As for the signal acquiring process, different kinds of sample rate might be involved, for common ECG acquisition device the sample rate would be 128Hz, 250Hz, 340Hz or 500Hz, even higher. However, even in murine studies, a sampling rate of 2 kHz is considered sufficiently high [5]. Arbitrary resizing would be an ideal procedure to handle with the different sampling rate from different data source to build the datasets for mining and analysis.

ECG Signal Preprocessing

Before the segmentation and feature extraction process, the ECG signals were preprocessed. As in the procedure of collecting ECG signals, in addition to the ECG signals, the baseline wander (caused by Perspiration, respiration and body movements), power line interference and muscle noise were recorded as well, which had been described in lots of literatures [6]. When the filtering methods were proposed and adopted in the preprocessing, the desired information should not be altered. The ECG typically exhibits persistent features like P-QRS-T morphology and average RR interval, and non-stationary features like individual RR and QT intervals, long-term heart rate trends [1]. Possible distortions caused by filtering should be quantified in these features.

The filtered ECG signals then were segmented into individual heartbeat waveforms depends on the detected R peaks in a classification task. The ECG segmentation can be seen as the decoding procedure of an observation sequence in terms of beat waveforms [7]. Dynamic time warping [8], time warping [9], Bayesian framework [10], hidden Markov models[7], weighted diagnostic distortion [11], morphology and heartbeat interval based methods [12] and genetic methods [13] had been used in this sub-task. The state accuracy rate was close to 100%, which would be accurate enough in most online and offline applications.

ECG Feature Extraction and Classification

After the segmentation for the ECG records, we got plenty of ECG waveform samples with variety categories. Since different physiological disorder may reflect on different type of abnormal heartbeat rhythms. For the task of classification, it is quite important to determine the classes which would be used. In the early literatures, there were no unified class labels for an ECG classification problem. As in the

open database MIT-BIH arrhythmia database annotations [14, 15], the class label system was build with five beat classes recommended by ANSI/AAMI EC57:1998 standard, i.e., normal beat, ventricular ectopic beat (VEB), supraventricular ectopic beat (SVEB), fusion of a normal and a VEB, or unknown beat type were used in most literature on the classification problems instead of early diversity sub class labels, which could be appropriate in the task since the widely acceptance.

Supervised learning Methods in ECG classification

To be added....

Unsupervised learning Methods in ECG classification

To be added....

Deep Learning in ECG Classification: A Preliminary Study Based on Deep Sparse Autoencoder

Deep learning methods attempt to learn feature hierarchies as higher-level features are formed by the composition of lower-level features. The electrocardiography interpretation has been judged by the medical professionals, which was based on the abstractions of the perceptible features. In this model we consider the higher-level abstractions as the perceptible features, with whose composition the medical professionals can make arrhythmia judgement. The deep architecture automatic learning method is especially important for high-level abstractions, which human often do not know how to specify explicitly in terms of raw sensory input [16]. As [17] discussed, deep learning methods are bused on learning internal representations of data, another important advantage they offer is the ability to naturally leverage: (a) unsupervised data and (b) data from similar tasks to boost performance on large and challenging problems that routinely suffer from a poverty of labelled data. In the electrocardiography classification problem, we got plenty of unsupervised data, and the labelled data was limited as well, so it is a spontaneously idea to adapt deep learning method in this classification problem.

Deep Neural Networks

The artificial neural network had been widely used in different applications, the basic 3-layer model (with only one hidden layer) is a fairly shallow network which means only shallow features can be learned via the structure. Deep neural networks were the structures in which we have multiple hidden layers, with which we can compute much more complex features from the input. Each hidden layer computes a non-linear transformation of the previous layer, a deep network can have significantly greater representational power (i.e., can learn significantly more complex functions) than a shallow one. A typical deep neural network structure makes no different from the normal multi layer neural network.

Autoencoders and Sparsity

An autoencoder is trained to encode the input x into some representation $c(x)$ so that the inputs can be reconstructed from that representation. High-dimensional data can be converted to low-dimensional codes by training a multilayer neural

network with a small central layer to reconstruct high-dimensional input vectors and such "autoencoder" networks works better than principal components analysis as a tool to reduce the dimensionality of data [18]. Principal Component Analysis (PCA) is a linear reduction technique that seeks projection of the data into the directions of highest variability [19], while autoencoders do the same task in a different way with a wider scope (PCA is method that assumes linear systems where as autoencoders do not). Since in the neural network the hidden layer is nonlinear, the autoencoder behaves differently from PCA, which has the ability to capture multi-modal aspects of the input distribution (the representation of the input). The related literature experiments reported in [20] suggest that in practice, when trained with stochastic gradient descent, nonlinear autoencoders with more hidden units than inputs (called overcomplete) yield useful representations (in the sense of classification error measured on a network taking this representation in input). A farther defence of autoencoder can be accessed from [21]. As the theory illustrated, the electrocardiography signal representations can be learned via the autoencoder structure and learning algorithms.

The structures and learning algorithms used were illustrated in lots of literatures [19, 22]. Here we impose a sparsity constraint on the hidden units to guarantee the representations expression ability. So for the neuron in the neuron network would be "active" if its output value is close to 1, or as being "inactive" if its output value is close to 0 due to the adopted sigmoid activation function. Here $a_j^{(2)}(x)$ denote the activation of hidden unit j in the autoencoder with the given input of x. Fatherly, let

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})] \tag{1}$$

be the average activation of hidden unit j (averaged over the training set). Approximately enforce the constraint:

$$\hat{\rho}_j = \rho \tag{2}$$

where ρ is a sparsity parameter, typically a small value close to zero (such as $\rho = 0.05$), which means the average activation of each hidden neuron j to be close to zero (0.05 for instance).

The overall cost function of neural network is denoted by $J(W, b)$ which was defined by:

$$J(W,b) = [\frac{1}{m}\sum_{i=1}^{m}(\frac{1}{2}\|h_{W,b}(x^{(i)}) - y^{(i)}\|^2)] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}{W_{ji}^{(l)}}^2 \tag{3}$$

as the first term in the definition of $J(W, b)$ is an average sum-of-squares error term. The second term is a regularization term that tends to decrease the magnitude

of the weights, and helps prevent overfitting. The definition of λ, s, l etc. would be explained in detail in the appendix part. To satisfy the constraint of sparsity, an extra penalty term to the optimisation objective that penalised $\hat{\rho}_j$ deviating significantly from ρ. The Kullback-Leibler (KL) divergence:

$$\sum_{j=1}^{s_2} KL(\rho||\hat{\rho}) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_j} \tag{4}$$

is chosen as the penalty term. KL-divergence is a standard function for measuring how different two different distributions are. So in the autoencoder neural network training, the cost function of $J_{sparse}(W,b)$ was defined as:

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} KL(\rho||\hat{\rho}_j) \tag{5}$$

β denotes the weight of the sparsity penalty term. The above theories were cited from the recent research literatures [23] and open source [24] on the topic of deep learning.

Representation Learning

The autoencoder base on neuron network had been used to learn representations (features) from unlabelled data. Autoencoders have been used as building blocks to build and initialize a deep multi-layer neural network. The training procedure would be [21]:

1 Train the first layer as an autoencoder to minimise some form for reconstruction error of the raw input. This is unsupervised.
2 The hidden units' outputs of the autoencoder are now used as input for another layer, also trained to be an autoencoder. Here unlabelled representations were used as well.
3 Iterates as in 2) to initialize the desired number of additional layers.
4 Take the last hidden layer output as input to a supervised layer and initialize its parameters (either randomly or by supervised training, keeping the rest of the network fixed).
5 Fine tune all the parameters of this deep architecture with respect to the supervised criterion. Alternately, unfold all the autoencoders into a very deep autoencoder and fine-tune the global reconstruction error.

The greedy layer-wise approach for pretraining a deep network works by training each layer in turn as explained in step 2). Assume $a^{(n)}$ as the deepest activation of the autoencoder network, then $a^{(n)}$ is a higher level representation than any lower layers, which contains what we interested in. Then the higher level representations (the corresponding features in the traditional artificial selected features) can be used as the classifier input.

Fine-tuning and Classifier

For the training method of stacked autoencoders, when the parameters of one layer are being trained, parameters in other layer are kept fixed. In order to achieve better result, fine-tuning using backpropagation can be used to improve the model performance by tuning the parameters of all layers are changed at the same time after the layer-wise train phase. After the fine-tuning process the optimised network structure would learn a good representation of the inputs, which can be used as the features similar to the traditional methods. The cardiac arrhythmia classification problem is a multi-classs classification problems where the class label y may take more than two possible values.So the softmax regression is selected as the supervised learning algorithm which would be adapted as the classifier in conduction with the deep network.

Softmax regression model was generalized from the logistic regression. Similar to the logistic regression, the training set

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\} \tag{6}$$

of m labelled examples, the input features are $x^{(i)} \in \Re^{(n+1)}$ (with x_0 corresponding to there intercept term). The labels are denoted by

$$y^{(i)} \in \{1, 2, 3, \ldots, k\} \tag{7}$$

which means k classes. Given a test input x, the hypothesis to estimate the probability that $p(y = j|x)$ for each value of $j = 1, \ldots, k$. I.e., the probabilities of the class labels taking on the k different possible values are estimated.

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} \tag{8}$$

$$= \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{9}$$

in which $\theta_1, \theta_2, \ldots, \theta_k \in \Re^{(n+1)}$ are parameters of the model. The term $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}$ was normalizes the distribution, so that it sums to one.

The cost function adopted for softmax regression is:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}] \tag{10}$$

where $1\{\cdot\}$ is the indicator function. There is no known closed-form way to solve for the minimum of $J(\theta)$, and an iterative optimisation algorithm synch as gradient descent of L-BFGS could be used for the minimal value (some other iterative optimisation algorithms were mentioned in [25]). So the cost function and iteration equations would be:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\}\log\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{jk}^2(\lambda > 0) \quad (11)$$

and

$$\bigtriangledown_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[x^{(i)}(1\{y^{(i)} = j\} - p(y^{(i)} = j|x^{(i)}; \theta))] + \lambda\theta_j(\lambda > 0) \quad (12)$$

By minimizing $J(\theta)$ with respect to θ, the softmax regression classifier would work properly for the classification task.

Experiments and Results

Datasets Preparation

As illustrated in the above section, the preprocessing and segmentation had been described. In the preprocessing stage, filtering algorithms were adapted to remove the artefact signals from the ECG signal. The signals include baseline wander, power line interference, and high-frequency noise. The segmentation method was based on the program of Laguna et al.[1] was adapted, which also had been validated by other related work [12]. The experiment was based on three datasets:

1 Ambulatory electrocardiography database were used in this study, which includes recordings of 100 subjects with arrhythmia along with normal sinus rhythm. The database contains 100 recordings, each containing a 3- lead 24-hour long electrocardiography which were bandpass filtered at 0.1-100Hz and sampled at 128Hz. In this study, only the lead I data were adapted after preprocessing in the classification task. The reference average heart beats for each sample has 97,855 beats for the 24-hour long recording, and the reference arrhythmia average is 1,810 beats which were estimated by a commercial software (this statistics aim to indicate the existence for arrhythmia samples, which should not be consider as a experiment preset).

2 The MIT-BIH Arrhythmia Database [23] contains 48 half- hour recordings each containing two 30-min ECG lead signals (lead A and lead B), sampled at 360Hz. As well only the lead I data were used in the proposed method. In agreement with the AAMI recommended practice, the four recordings with paced beats were removed from the analysis.Five records randomly selected were used to verify the real time application. The remaining recordings were divided into two datasets, with small part of which were used as the training set of the fine- tuning process (details would be described in the following part).

[1] "ecgpuwave", check the website of Physionet

3 The MIT-BIT Long-term Database is also used in this study for training and verification, which contains 7 long-term ECG recordings (14 to 22 hours each), with manually reviewed beat annotations and sampled at 128Hz. Similarly, the 7 recordings were divided into two datasets, with part used as the fine-tuning training set.

After the segmentation for the ambulatory ECG database, three batches of heartbeat samples listed in Table 1 were acquired for the classification task.

Table 1 Samples after Segementation

Ambulatory ECG Database (AECG)	MITBIH-AR	MITBIH-LT
9,785,500	100,687	667,343

As for the pretraining, fine-tuning for our proposed task and comparison, we divided all the samples into three groups: the pretraining group as DS1, the fine-tuning group as DS2 and test group as DS3 (illustrated in Table 8). Samples are chosen randomly from the original AR and LT database, the details of the sample class would be described in the experiment result analysis.

Table 2 Samples Dataset Settings

Dataset	DS1	DS2	DS3
Useage	Pretraining	Fine-tuning	Test
Source (samples)	AECG (9,785,500)		
	AR (50,193)	AR (33,663)	AR (16,831)
	LT (587,347)	LT (50,000)	LT (30,000)
Total	10,423,040	83,633	46,831

Classification Workflow

The stack autoencoder use multilayer "encoder" network to transform high dimensional data into low dimensional code, similarly a "decoder" network can be adopted to recover from the code, which we previously described. For the one-hidden-layer autoencoder input layer and hidden layer, the output was set equal to the input, starting with random weights in the one-hidden layer neural networks, they can be trained together by minimizing the discrepancy between the original input data and the reconstruction. The gradients were obtained by using chain rule of back-propagate error derivatives, the decoder means the raw input can be reconstructed by the learned feature with the trained weight. With large initial weights, autoencoders typically find poor local minima; with small initial weights, the gradients in the early layers are tiny, making it infeasible to train autoencoders with many hidden layers. After learning the feature and network weight in the first layer, we can add hidden layer one by one to get deeper representations, as well the learned weight can be used to reconstruct the input. When training the weight of layer 2, we take the weight in layer 1 fixed replace random initialize because the learned weights are close to a good solution, which means training the parameters of each layer individually while freezing parameters for the remainder of the model. In the experiment, we adopted 2-hidden-layer, 3-hidden- layer, 4-hidden-layer stacked autoencoder for the test and verification.

Fine tuning is a strategy that widely used in deep learning, which can be used to greatly improve the performance of a stacked autoencoder. After pretraining

multiple layers of feature detectors, the model is "unfold" to produce encoder and decoder networks that initially use the same weights [37]. The weights learned can be used for classification implementation after adding one classifier after the feature layer. In this study, a softmax classifier was added (Figure 3). In the fine-tuning initialization, the parameters learned in the autoencoder pretraining were used, and the weights W and biases b of softmax classifier (the last layer of the network) were initialized randomly. The training set of DS2 were used in the supervised learning pretraining while the backpropagation algorithm as usual of multi-layer perceptrons to minimize the output prediction error has been adopted.

Classifier Performance Assessment

After the pretraining and fine-tuning process, the deep network parameters were acquired. Then we use the parameters and the test data set DS3 to predict the class of samples. It is necessary to mention that in DS2 and DS3, the labelled data used in pretraining and fine-tuning were divided randomly, which satisfy the requirement of Holdout cross-validation scheme so that the test results were meaningful for the classification task performance improvement.

Table 3 Samples Dataset Settings

Dataset	DS1	DS2	DS3
Useage	Pretraining	Fine-tuning	Test
Source (samples)	AECG (9,785,500) AR (50,193) LT (587,347)	 AR (33,663) LT (50,000)	 AR (16,831) LT (30,000)
Total	10,423,040	83,633	46,831

The following statistical parameters of test performance were used in the study:

1 Specificity: number of correctly classified normal beats over total number of normal beats.
2 Sensitivity: number of correctly classified abnormal beats over total number of the given abnormal beats.
3 Overall classification accuracy: number of correctly classified beats over number of total beat.

Results

As previously mentioned, we adopted three different layer strategies for the classification task. In the 2-hidden-layer autoencoder network, we got a accuracy of 99.33%. For the N class the specificity is 99.76%, the sensitivity of S class is 80.08%, the sensitivity of V class is 98.13%, the sensitivity of F class is 85.48% as illustrated in Table VI.

Table 4 Test Result for 2-Hidden-Layer Autoencoder Network

		Algorithm label					
		N	S	V	F	Q	T
Reference label	N	41,965	39	45	13	6	42,068
	S	91	398	6	2	0	497
	V	63	3	3,940	5	4	4,015
	F	23	0	13	212	0	248
	Q	2	1	0	1	0	3

The test accuracy is about 99.33%.

In the 3-hidden-layer autoencoder network, we got a accuracy of 99.07%. For the N class the specificity is 99.64%, the sensitivity of S class is 75.14%, the sensitivity of V class is 97.58%, the sensitivity of F class is 80.33% as illustrated in Table VI.

Table 5 Test Result for 3-Hidden-Layer Autoencoder Network

		Algorithm label					
		N	S	V	F	Q	T
Reference label	N	41,721	66	66	19	0	41,872
	S	120	405	13	1	0	539
	V	74	10	4,073	17	0	4,174
	F	27	2	19	196	0	244
	Q	2	0	0	1	0	2

The test accuracy is about 99.07%.

In the 4-hidden-layer autoencoder network, we got a accuracy of 99.34%. For the N class the specificity is 99.74%, the sensitivity of S class is 82.29%, the sensitivity of V class is 98.31%, the sensitivity of F class is 87.71% as illustrated in Table 6.

Table 6 Test Result for 4-Hidden-Layer Autoencoder Network

		Algorithm label					
		N	S	V	F	Q	T
Reference label	N	41,778	38	48	17	5	41,886
	S	93	460	3	1	2	559
	V	52	1	4,067	11	6	4,137
	F	15	0	13	214	2	244
	Q	1	0	1	1	1	5

The test accuracy is about 99.34%.

Comparison with Other Work

Different kinds of performance assessment criteria had been adopted in the ECG arrhythmia classification problem. In the comparison part, we adopt several ordinary indicators for the performance assessment, which brought in the above sections. The accuracies, N-class specificities (N-spe), S-class sensitivities (S-sen), V-class sensitivities (V-sen) and the F-class sensitivities(F-sen) in Table 7 are presented for the comparison. The percentages are calculated from the literatures' test results, in which some of the classes are ignored like [**?**], we use a $*$ symbol to represent the result are not available. In Table 7, we use the highest value (2 to 4 hidden layers based structures) for the verification which illustrated in "proposed" line.

Table 7 Comparisons with Other Work

Approaches	Accuracy	N-spe	S-sen	V-sen	F-sen
Proposed	**99.34%**	**99.76%**	82.29%	**98.31%**	87.71%
Mar[**?**]	84.63%	84.85%	82.90%	86.72%	51.55%
Chazal[**?**]	86.19 %	86.86%	**83.83%**	77.74%	**89.43%**
Melgani[**?**]	90.52%	89.12%	*[a]	89.97%	*
Jiang [**?**]	94.51%	98.73%	50.59%	86.61%	35.78%

[a] * means the results were not available.
[b] The listed percentages are based on the previous described rules.

Through the comparisons in Table 7, we can see that the proposed method offers better accuracy of classification problem. Since accuracy in lots of the literatures are good enough, the verification parameter depends on mainly on the normal class detection, but on contrary with these methods, this approach provided better performance in other kind of arrhythmia waveforms classes. Especially in the ventricular

ectopic beat sensitivity, a quite large improvement had been made by the proposed method.

Deep Learning in ECG Classification: A Two-lead ECG Classification Based on Deep Belief Network

A restricted Boltzmann machine learning algorithm were proposed in the two-lead heart beat classification problem. A restricted Boltzmann machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs. In this part a deep belief network was constructed and the RBM based algorithm was used in the classification problem.

The Deep Belief Network and Classifier

The Restricted Boltzmann Machine

The restricted Boltzmann Machine is a stochastic neural network with strong unsupervised learning ability. In the RBM network structure, each visible unit is connected to the hidden units without visible-visible or hidden-hidden connections. There were no connections between the visible and hidden layers. The visible units were independent then the Gibbs sampling method could be used to approximate the probability distribution. It consists of one layer of visible units with input $X = (v_1, v_2, ..., v_n)$, one layer of hidden units with output $Y = (h_1, h_2, ..., h_m)$ and two bias units whose states were always on and a way to adjusting the value of each unit.

Boltzmann machine is based on statistical mechanics. The energy function $E(v, h)$ of an RBM was defined as:

$$E(v, h|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_i h_i - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \tag{13}$$

v and h present the state vectors of the visible and hidden layers, a_i. b_j and W_{ij} are parameters , define $\theta = \{W_{ij}, a_i, b_j\}$. So based on the energy function, the distribution of v and h is:

$$P(v, h|\theta) = \frac{e^{-E(v,h|\theta)}}{Z(\theta)}, Z(\theta) = \sum_{v,h} e^{-E(v,h|\theta)} \tag{14}$$

The purpose of RBM is to learn the optimal θ, according to the probability distribution, the maximum likelihood function is defined as:

$$\theta^* = arg\max_{\theta} L(\theta) = arg\max_{\theta} \sum_{t=1}^{T} \log P(v^{(t)}|\theta) \tag{15}$$

$$L(\theta) = \sum_{t=1}^{T} \left(\log \sum_{h} exp[-E(v^{(t)}, h|\theta)] - \log \sum_{v}\sum_{h} [-E(v, h|\theta)] \right) \tag{16}$$

To get the optimal θ^*, stochastic gradient descent[**?**] method was used to maximum the likelihood function $L(\theta)$. The partial derivative of the parameters is shown below:

$$\begin{aligned}
\frac{\partial \log P(v|\theta)}{\partial W_{ij}} &= \langle v_i h_i \rangle_{data} - \langle v_i h_i \rangle_{model}, \\
\frac{\partial \log P(v|\theta)}{\partial a_i} &= \langle v_i \rangle_{data} - \langle h_i \rangle_{model}, \\
\frac{\partial \log P(v|\theta)}{\partial b_j} &= \langle h_i \rangle_{data} - \langle h_i \rangle_{model}.
\end{aligned} \tag{17}$$

$\langle . \rangle_P$ denotes the distribution about P. $\langle . \rangle_{data}$ is easy to be calculated when the training samples were defined. $\langle . \rangle_{model}$ could not be resolved directly, but approximated by Gibbs sampling. Here we use the contrastive divergence(CD)[**?**] algorithm proposed by Hinton in 2002, with which would achieve better results by only one step of Gibbs sampling .

Classifier and the Training of Multi-layer RBM

This model generalized logistic regression [**?**] in classification missions which would be useful in heartbeats arrhythmia classification problems. The softmax model is a kind of supervised learning method in conjunction with the deep belief network.

Supposing m samples in the training set :

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\} \tag{18}$$

the inputs were vectors $x^{(i)}$ corresponding to the features space. The labels are denoted by $y^{(i)}$ corresponding to the arrhythmia classes of the inputs. The cost function of softmax regression with a weight decay term was defined as:

$$\begin{aligned}
J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}] \\
+\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{jk}^2 (\lambda > 0)
\end{aligned} \tag{19}$$

and the partial derivative of the parameters were:

$$\begin{aligned}
\bigtriangledown_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[x^{(i)}(1\{y^{(i)} = j\} - p(y^{(i)} = j|x^{(i)}; \theta))] \\
+\lambda\theta_j (\lambda > 0)
\end{aligned} \tag{20}$$

To train an optimum classifier, an optimization gradient descent algorithm called L-BFGS was used same as what had been done in the autoencoder classification.

Figure 1 shows that RBMs can be stacked and trained with a greedy manner to form a deep belief network(DBN)[**?**, **?**]. In the last layer, a softmax classifier is

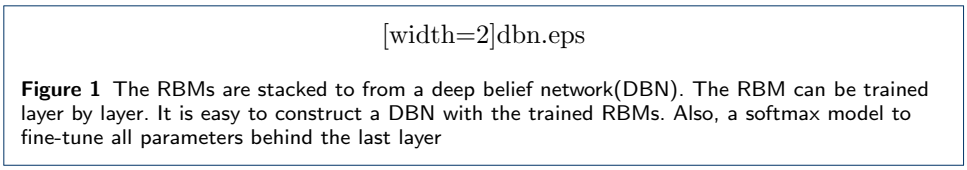

Figure 1 The RBMs are stacked to from a deep belief network(DBN). The RBM can be trained layer by layer. It is easy to construct a DBN with the trained RBMs. Also, a softmax model to fine-tune all parameters behind the last layer

connected with the DBN. DBN is the graphical model of a hierarchical architecture. The five layers network shows in Figure1 are used to the heartbeat classification. The procedures were:

1 Train the first layer as an RBM which models the raw input X as a visible layer.
2 After training the RBM, representations of the input were obtained.
3 Train the next layer as an RBM which models the transformed data as a visible layer.
4 Iterate step 2 and 3 for the desired number of layers.

Finally the RBMs are combined to a DBN with the softmax model. Fine-tuning then was used as the supervised method to minimize the likelihood function and improve the adaptability. Here also the L-BFGS algorithm was used.

Combined optimization algorithm for multi-lead classifiers

The ECG wavelet transform performs differently in different channels by the waveforms. Each heartbeat channel shows diversely due to the P,QRS-complex and T wave constituent. Taking those into consideration, multi-lead ECG classification is significant improved by sample voting method. So a weight optimization method is proposed in two leads ECG signal classification. The method can be generally used in multi-lead ECG data classification.

For each classifier trained with distinct lead, a reliability value is denoted as the accurate rate of the classifier. let γ represent the reliability value. Then the classifiers' reliability is defined as $\gamma_1, \gamma_2, \cdots$

Using the testing samples to assess the result. The statistical matrix:

$$ClassSTST = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{12} & \cdots & C_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{pmatrix} \tag{21}$$

In the statistical matrix, there are n classes and C_{11} represents the class I which is classified as class I, C_{12} represents class I is classified as II, etc. The diagonal values are the correct classification. The purpose is to increase the diagonal values, so we adopt weights of the outputs for each classifier. The weights of the first classifier is $W_1 = (w_{11}, w_{12}, \cdots, w_{1n})$, the second were $W_2 = (w_{21}, w_{22}, \cdots, w_{2n})$. The constraint condition is $\sum_{i=1}^{2} w_{ik} = 1$.

First initial the weight to a mean value. Each sample has an output vector $O = o_1, o_2, \cdots, o_n$, the class is decided by the maximum value. Adding the weight of the value, the output is $(o_1 w_1, o_2 w_2, \cdots, o_n w_n)$. If the label of the sample is l, we would like to maximize $o_1 w_{11} + o_1 w_{21}$ while correspondingly minimize others. Through this,

the l class accurate is promoted while the false negative rate is also increased. The optimization algorithm would find a balance between the two weights of all testing samples. Accordingly, the optimal function is defined as:

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} x \exp^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{22}$$

To find the maximum value of x, the derivative of the equation is:

$$F^{'}(x) = 0, x = \frac{\mu}{2} + \sqrt{\frac{\mu^2}{4} + \sigma^2} \tag{23}$$

Here, μ is denoted as initial weights. On the basis of the statistical matrix , counting the difference of the correct (diffcort) and false negative (diffflneg) quantities of the two classifiers. If the difference of the difference of the correct and false negative greater than zero, $\sigma^2 = \sqrt{\frac{diffcort - difffflsneg}{totalnumber}}$, so updating the corresponding class weight of the first classifier as $w_1 = \frac{\mu}{2} + \sqrt{\frac{\mu^2}{4} + \sigma^2}$. Else, $\sigma^2 = \sqrt{\frac{difffflsneg - diffcort}{totalnumber}}$, updating the weight of the second classifier as $w_2 = \frac{\mu}{2} + \sqrt{\frac{\mu^2}{4} + \sigma^2}$. Finally normalize the weights, the optimal combined value is $\gamma_1 o_1 w_1 + \gamma_2 o_2 w_2$. Generally, every two classifiers can be used to optimize the multi-lead ECG classification.

Experiment and Results

Preprocessing and Segmentation

Here we do the same process as the above sparse autoencoder network diid. The experiment data sets changed due to the new problem. The preprocessing include two main parts: the ECG data filtering and heartbeat segmentation. The filtering task is removing the artifact signal from the ECG signal, which includes the baseline wander, power line interference and high-frequency noise. The massive unlabelled data we collected is extracted and resampled from 128Hz to 360Hz.

In heartbeat segmentation process, average samples of each beat is 277 samples. To get more information , we allow partially overlap and a window with a length of 340 data points in one beat was defined, the R peak of the wave is located at 141st point. Most annotations of the MIT-BIH arrhythmia database is lied the R-wave. For the dataset we collect, a high accurate algorithm has been explored to determine the R pick and then divide into the heartbeat segments according to the R pick.

Training and Fine-tuning

The goal of RBM learning is to maximize the product of the probabilities. The parameters of the network can be initialized by the constructor. This option is useful when an RBM is used as the building block of the deep belief network, in which case the weight matrix and the hidden layer bias is shared with the corresponding layer of the network. The active function of the nodes is sigmoid function. The data is batched to train the RBM layer by layer. A single-step contrastive divergence(CD-1) is used in the gradient descent procedure. After calculating the partial derivative, the weights and bias is updated.

After the learning process, the RMBs can be used to initialize a deep belief network. Standard backpropagation algorithm can be applied to fine-tune the model. That can significantly improve the performance of the DBN. The fine- turning process is a supervised learning procedure, so at the last layer, a multi-class model called softmax is connected to classify the ECG data. Then using the fine-turning method to minimize the cost function.

Experiment Results

In the experiments, we adopted multi-lead ECG signal based on the restricted Boltzmann machine for the classification task. The MIT-BIH arrhythmia database is divided into three parts. Half of the beat is added to training the RBMs, one-third is applied to fine-tune the network and the left is used to test the model. In the 3 hidden layers deep belief network, the classifier outperforms in terms of sensitivity (SPR) 99.35%, specificity (SPC) 95.18% and accuracy rate (ACC) 98.25% using the first channels. Using the second channel we get the accuracy (ACC) of 97.43%, sensitivity (SPR) 98.90% and specificity (SPC) 93.36%. At convergence of the optmization process, the combining method achieves the accuracy (ACC) of 98.83%, sensitivity (SPR) 99.83% and specificity (SPC) 96.05%.

Table 8 Test result of three hidden Layers deep belief network using the first lead

		Algorithm classified label										
		NORMAL	LBBB	RBBB	ABERR	PVC	FUSION	NPC	APC	FLWAV	VESC	NESC
Original label	NORMAL	12,289	3	2	3	15	9	0	37	2	0	0
	LBBB	7	1,369	0	0	6	0	0	1	0	0	0
	RBBB	4	0	1,179	0	3	0	0	4	0	0	0
	ABERR	7	1	0	12	5	0	0	0	1	0	0
	PVC	18	2	0	2	1,104	8	0	2	2	0	0
	FUSION	19	0	0	1	6	97	0	0	0	0	2
	NPC	5	0	1	0	0	0	4	1	0	0	1
	APC	58	2	9	0	2	0	0	382	0	0	2
	FLWAV	10	0	0	1	8	0	0	0	58	0	0
	VESC	2	0	0	0	1	0	0	0	0	24	0
	NESC	6	0	1	0	0	0	0	1	0	0	15
	AESC	3	0	0	0	0	0	0	0	0	0	0

The test accuracy of the first lead is 98.247%.

For we collect large amount of ECG data from the hospital which contains lots of normal beats and abnormal beats, the learning method of restricted Boltzmann machine is used to learning the features from the massive data by an unsupervised way. Then the RBMs is adapted to build a deep belief network. The optimization algorithm we propose improves the accuracy with multilead ECG signal. The heartbeat classification also has been studied by other algorithm, including hidden Markov model(HMM) [**?**], support vector machine(SVM) [**?**][**?**], independent component analysis(ICA) [**?**], Artificial neural networks(ANN) that use varied features extracted from the discreet signal data. Different performance assessment criteria has been adopted. In this comparison, we evaluate the performance on the indicators which put forward in the above sections: Sensitivity(TPR), specificity(SPC) and overall Accuracy(ACC). The $*NR$ symbol represents the result is not reported.

Table 9 Test result of three hidden layers deep belief network using second lead

		Algorithm classified label										
		NORMAL	LBBB	RBBB	ABERR	PVC	FUSION	NPC	APC	FLWAV	VESC	NESC
Original label	NORMAL	12,233	7	3	3	57	13	1	34	12	0	6
	LBBB	12	1,366	0	0	5	0	0	0	0	0	0
	RBBB	5	0	1,169	0	5	0	0	9	2	0	0
	ABERR	7	0	1	10	6	0	1	0	1	0	0
	PVC	56	3	1	0	1,048	15	0	4	11	0	0
	FUSION	22	0	0	0	5	97	0	0	0	0	1
	NPC	1	0	1	0	0	0	10	0	0	0	0
	APC	60	4	8	0	11	2	0	368	1	0	0
	FLWAV	7	0	0	0	8	0	0	1	61	0	0
	VESC	2	0	1	0	2	0	0	0	2	20	0
	NESC	6	0	1	0	1	0	0	0	0	0	15
	AESC	2	0	0	0	0	0	0	0	0	0	0

The test accuracy of the second is 97.433%.

Table 10 Test result of combination optimal algorithm with two leads

		Algorithm classified label										
		NORMAL	LBBB	RBBB	ABERR	PVC	FUSION	NPC	APC	FLWAV	VESC	NESC
Original label	NORMAL	12,348	0	0	0	11	2	0	7	1	0	0
	LBBB	3	1,377	0	0	3	0	0	0	0	0	0
	RBBB	1	0	1,182	0	2	0	0	5	0	0	0
	ABERR	8	0	0	16	2	0	0	0	0	0	0
	PVC	16	0	0	0	1,108	10	0	1	3	0	0
	FUSION	22	0	0	0	4	99	0	0	0	0	0
	NPC	3	0	0	0	0	0	9	0	0	0	0
	APC	58	2	5	0	1	0	0	389	0	0	2
	FLWAV	2	0	0	0	6	0	0	0	69	0	0
	VESC	3	0	0	0	1	0	0	0	0	23	0
	NESC	11	0	1	0	0	0	0	1	0	0	11
	AESC	3	0	0	0	0	0	0	0	0	0	0

The test accuracy of the combining leads is 98.829%.

Table 11 Comparisons with others' works

Approaches	Accuracy(ACC)	Sensitivity(TPR)	Specificity(SPC)
Proposed	98.83%	99.83%	96.05%
Tadejko[?]	97.82%	99.70%	93.10%
Banerjee[?]	97.60%	97.30%	98.80%
Can[?]	99.71%	*NR	*NR
Osowski[?]	96.06%	98.10%	95.53%

[a] *NR means the results were not reported.
[b] The listed percentages are based on the assessment rules.

The Table 11 shows the performance of different models that used for the heartbeat classification, the proposed method offers a high accuracy of classification.

All the annotations in the MIT-BIH arrhythmia database is use in our study and Osowski[**?**] only selects 7 types. Can[**?**] get the highest accuracy but with a price of rejecting 2054 heartbeats(2.4% rejections). By comparing with others' experience, our approach provided higher performance in heartbeat classification without complex wavelet transform algorithms.

Coclusions

Text

References

References

1. Clifford, G.D., Azuaje, F., McSharry, P., *et al.*: Advanced Methods and Tools for ECG Data Analysis. Artech House, Boston (2006)
2. Mar, T., Zaunseder, S., Martinez, J.P., Llamedo, M., Poll, R.: Optimization of ecg classification by means of feature selection. Biomedical Engineering, IEEE Transactions on **58**(8), 2168–2177 (2011)
3. Clifford, G.D., Azuaje, F., McScharry: Advanced Tools for ECG Analysis. http://www.ecgtools.org/
4. Silva, C.V., Philominraj, A., del Río, C.: A DSP Practical Application: Working on ECG Signal. INTECH Open Access Publisher, ??? (2011)
5. Ai, H., Cui, X., Tang, L., Zhu, W., Ning, X., Yang, X.: [studies on the time domain and power spectrum of high frequency ecg in normal mice]. Sheng li xue bao:[Acta physiologica Sinica] **48**(5), 512–516 (1996)
6. Blanco-Velasco, M., Weng, B., Barner, K.E.: Ecg signal denoising and baseline wander correction based on the empirical mode decomposition. Computers in biology and medicine **38**(1), 1–13 (2008)
7. Andreão, R.V., Dorizzi, B., Boudy, J.: Ecg signal analysis through hidden markov models. Biomedical Engineering, IEEE Transactions on **53**(8), 1541–1549 (2006)
8. Vullings, H., Verhaegen, M., Verbruggen, H.: Automated ecg segmentation with dynamic time warping. In: Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE, pp. 163–166 (1998). IEEE
9. Vullings, H., Verhaegen, M., Verbruggen, H.B.: Ecg segmentation using time-warping. In: Advances in Intelligent Data Analysis Reasoning About Data, pp. 275–285. Springer, ??? (1997)
10. Sayadi, O., Shamsollahi, M.: A model-based bayesian framework for ecg beat segmentation. Physiological Measurement **30**(3), 335 (2009)
11. Zigel, Y., Cohen, A., Katz, A.: The weighted diagnostic distortion (wdd) measure for ecg signal compression. Biomedical Engineering, IEEE Transactions on **47**(11), 1422–1430 (2000)
12. De Chazal, P., O'Dwyer, M., Reilly, R.B.: Automatic classification of heartbeats using ecg morphology and heartbeat interval features. Biomedical Engineering, IEEE Transactions on **51**(7), 1196–1206 (2004)
13. Gacek, A., Pedrycz, W.: A genetic segmentation of ecg signals. Biomedical Engineering, IEEE Transactions on **50**(10), 1203–1208 (2003)
14. Mark, R., Schluter, P., Moody, G., Devlin, P., Chernoff, D.: An annotated ecg database for evaluating arrhythmia detectors. In: IEEE Transactions on Biomedical Engineering, vol. 29, pp. 600–600 (1982)
15. Moody, G.B., Mark, R.G.: The mit-bih arrhythmia database on cd-rom and software for use with it. In: Computers in Cardiology 1990, Proceedings., pp. 185–188 (1990)
16. Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S., Vincent, P.: The difficulty of training deep architectures and the effect of unsupervised pre-training. In: International Conference on Artificial Intelligence and Statistics, pp. 153–160 (2009)
17. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167 (2008). ACM
18. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
19. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, ??? (2012)
20. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., *et al.*: Greedy layer-wise training of deep networks. Advances in neural information processing systems **19**, 153 (2007)
21. Bengio, Y.: Learning deep architectures for ai. Foundations and trends® in Machine Learning **2**(1), 1–127 (2009)
22. Bishop, C.M., *et al.*: Pattern Recognition and Machine Learning vol. 4. springer New York, ??? (2006)
23. Zou, W., Zhu, S., Yu, K., Ng, A.Y.: Deep learning of invariant features via simulated fixations in video. In: Advances in Neural Information Processing Systems, pp. 3212–3220 (2012)
24. Ng, A., Ngiam, J., Foo, C.Y., Mai, Y., Suen, C.: UFLDL Tutorial. http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial (2010)
25. Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Le, Q.V., Ng, A.Y.: On optimization methods for deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 265–272 (2011)