Deep Learning-Based Classification of Massive Electrocardiography Data

Yan Yan, Xingbin Qin, Jianping Fan, and Lei Wang

***Abstract*—A Big Data approach for the classification of heartbeat in electrocardiography analysis is presented. Based on massive heartbeat samples extracted from ambulatory ECG dataset, a deep neural network structure with a stacked autoencoder pretraining with fine-tuning algorithm is adopted for classification of normal heartbeat, supraventricular ectopic heartbeat, ventricular ectopic heartbeat, fusion heartbeat based on the ANSI/AAMI EC57: 1998/(R)2008 standard. The dataset proposed was mainly collected from the subjects in the division of cardiology of the hospital.The MIT-BIH arrhythmia database and MIT-BIH long term ECG database are regard as the standard reference for classifier. The sparse autoencoder algorithm is adopted for the automatical feature learning process instead of complex features extraction selection. From the massive unlabelled dataset, features learned from the raw sample were acquired by the algorithm's training process. A significant improvement of heartbeat classification from the state-of-the-art is attained for the classification task with the accuracy to 99.34%. The result shows that with two or three hidden layers and small number of hidden node can get a better performance in classification. A feedback-based system with initial parameters learned from the deep network is designed for real time classification. The system perform much better in personal ECG classification and consumed less in real time applications.**

***Index Terms*—big data, electrocardiography classification, sparse autoencoder, deep learning, real-time classification.**

I. Introduction

AN era of big data in healthcare is now under way, decades of progress in digitising medical records accumulate vast amounts of medical data, simultaneously mobile healthcare and wearable sensor technologies offer healthcare data from larger population coverage. The noninvasive, inexpensive and well-established technology of electrocardiographic signal in mobile health or personal health has the greatest popularity in heart function analysis. Automated electrocardiography classification provides indispensable assist in long-term clinical monitoring, and a large number of approaches have been proposed for the task, easing the diagnosis of arrhythmic changes as well as further inspection, e.g., heart rate variability or heart turbulence analysis [1].

Lots of algorithms have been proposed for the classification and detection for electrocardiography signals. The electrocardiography classification or detection task had been divided into two parts: the feature extraction process and classifier. Simple classifier such as linear discriminants [2] and kNN [3], more complex classifiers like neural networks [4]–[7], fuzzy inference engines [7], [8], hidden Markov model [9], [10], independent component analysis [11] and support vector machine [3], [12], [13] were also adapted by lots of researchers.

Beyond the classifier, the performance of a recognition system highly depends on the determination of extracted electrocardiography features. Time domain features, frequency domain features, and statistical measures features for six fundamental waves (PQRSTU) had been used in feature extraction process [14]. Time domain features like morphological features include shapes, amplitudes, and durations were adapted primarily in [15]–[17], frequency domain features like wavelet transformation were widely used [18], [19] stationary features like higher order statistics also had been developed. Principal component analysis [20] and Hermite functions [21] have been used in electrocardiography classification and related analysis technologies as well. Almost every single published paper proposes a new set of features to be used, or a new combination of the existing ones [1].

The results from these algorithms or models were not amenable to expert labelling, as well as for the identification of complex relationships between subjects and clinical conditions [22]. But for the ambulatory electrocardiography clinical application, as well as the normal application in daily healthcare monitoring for cardiac function or early warning of heart disease, an automated algorithm or model would have significant meaning. The application of artificial intelligence methods has become an important trend in electrocardiography for the recognition and classification of different arrhythmia types [22]. The data explosion puts forward the new request to the method of data processing and information mining.

Over the past decades computational techniques proliferated in the pattern recognition field, simultaneously the applications in electrocardiography recognition, detection and classification for relevant trends, patterns, and outliers. Most of the literatures in the electrocardiography classification task were focused on the supervised learning methods, as in unsupervised learning methods were infrequently used, which needs a lot of effort in labelling data. The MIT-BIH database [23] was the most widely used data in the

Y. Yan is with the Shenzhen Key Laboratory for Low-cost Healthcare, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China-mail: (yan.yan@siat.ac.cn).

X. Qin is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China.

J. Fan is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China.

L. Wang is with the Shenzhen Key Laboratory for Low-cost Healthcare, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China.

Manuscript received September 29, 2014; revised

classification and detection algorithm developments, while mass unlabelled electrocardiography data had been ignored due to the supervise learning approaches essential. Unsupervised learning methods become crucial in mining or analysing unlabelled data, as the unlabelled electrocardiography data accumulated. Unsupervised learning-based approaches and the application to electrocardiogram classification in literatures mainly include clustering-based techniques [21], [24], [25], self-adaptive neural network-based methods [26], [27] and some hybrid unsupervised learning systems [28].

In this paper, we adopt a big data unsupervised learning approach of sparse autoencoder based deep neural network in large unlabelled ambulatory electrocardiography dataset to learn features automatically, with which the cardiac arrhythmia with electrocardiograms classification task was proposed.

In the following sections, we will first state the experimental setup in Section 2. In Section 3 the experimental methodology we propose the system and algorithm details. Then the experimental results and the discussion are given in Section 4 and Section 5 respectively.

II. Deep Learning, AE and SAE

A. Deep Learning Methods

The backpropagation neural network architecture had been widely applied since 1989 by its multidimensional mapping ability: any L_2 function from $[0,1]^n$ to $\mathbf{R}^n$ can be implemented to any desired degree of accuracy with a three-layer backpropagation neural network [29]. Until 2006, deep architectures have not been discussed much in the machine learning literature, because of poor training and generalisation errors obtained using the standard random initialization of the parameters [30]. Great successes in speech recognition [31], image recognition [32] had been accomplished via this powerful structure due to the proposed proper training algorithms by Hinton [33]. Deep learning methods attempt to learn feature hierarchies as higher-level features are formed by the composition of lower-level features. The network structure could be first layer-wise initialized via unsupervised training and then tuned with supervised learning methods. Deep models can generate more abstract features at higher levels than the lower ones, better results could be achieved when pre-training each layer with an unsupervised learning algorithm, one layer after the other (the so layer-wised manner), starting with the first layer (that directly takes in input the observed $\mathbf{x}$) [30].

Different kinds of deep neural network architectures were proposed since the layer-wised training method was developed by Hinton. Typical structures include deep belief networks (DBNs) [34], deep Boltzmann machines (DBMs) [35], stacked autoencoders (SAEs) [36], and stacked denoising AEs [37]. The autoencoder based deep learning model and stacked autoencoder as the corresponding architecture.

B. Autoencoders

The first research on the potential benefits of unsupervised learning based pre-training might date back to 1987, in which the first unsupervised autoencoder hierarchies were proposed [38]. The lowest-level autoencoder neural network is a single hidden layer which is trained to map input patterns to themselves. One visible layer of n inputs, one hidden layer of m units and one reconstruction layer of n outputs, as well the activation functions form the one-hidden-layer autoencoder as illustrated in (a) part of Fig. 1.

In the training process, the input were mapped to the hidden layer and produce the hidden layer output $h \in \mathbf{R}^m$ which was called "encoder". The hidden layer outputs were mapped to the output layer which were the reconstruction of input layer as the right part called "decoder" (Fig. 1). An autoencoder is trained to encode the input $x \in \mathbf{R}^n$ into some representations $h \in \mathbf{R}^m$ so that the input can be reconstructed from that representation [30], the reconstructed values were denoted as $\hat{x} \in \mathbf{R}^n$. Mathematically, these two steps can be formulated as

$$h = f(W_1 x + b_1) \tag{1}$$

$$\hat{x} = f(\hat{W_1} + \hat{b_1}) \tag{2}$$

where W_1 and $\hat{W_1}$ denote the input-to-hidden and the hidden-to-output weights, b_1 and $\hat{b_1}$ denote the biases. $f(\cdot)$ denotes the activation function which could be sigmoid function, hyperbolic tangent, and rectified linear function. The autoencoder behaves differently from PCA, which has the ability to capture multi-modal aspects of the input distribution (the representation of the input) [39] due to the applying of these nonlinear activation functions.

Here we use W and b as the general parameters of the weights and biases, we use $\hat{x} = h_{W,b}$ denote the reconstruction of the input with an autoencoder, the goal of training is to minimize the error between the input x and reconstruction, i.e.,

$$\underset{W,b}{\operatorname{argmin}}\, error(x, \hat{x}) \tag{3}$$

so the parameters were W and b while the input x is given. the error function can be defined in different ways like root mean square etc. Then the general way for stochastic gradient descent to optimize the parameters were:

$$W := W - \alpha \frac{\partial error(x, \hat{x})}{\partial W} \tag{4}$$

and

$$b := b - \alpha \frac{\partial error(x, \hat{x})}{\partial b} \tag{5}$$

in which α stands for the learning rate.

After the training process, the "decoder" part would be removed while the learning features (output of the hidden layer) with weights and biases would be stored, which can be subsequently used for higher layers to produce deeper features.

C. Stacked autoencoder

Stacking the input and hidden layers of AEs together layer by layer constructs an SAE [40], the reconstruction of the input with the stacking method is illustrated in (b) of Fig. 1. Deeper features could be generated from the raw input, then the feature could be adopted in a subsequent classifier.

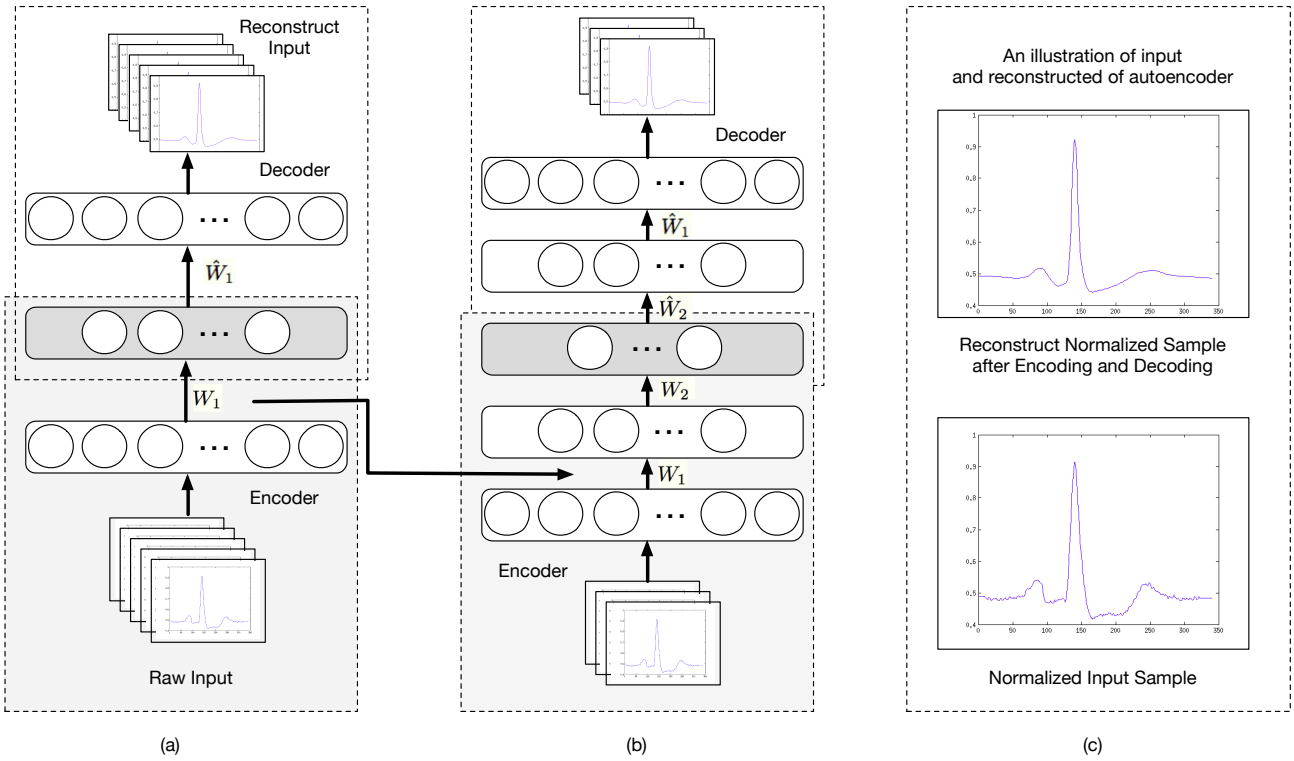

Fig. 1. The autoencoder and the staked autoencoder. (a) the one-layer autoencoder structure with the encoder and decoder for reconstruction of the raw input; (b) the multi-layer stacked autoencoder for input reconstruction; (c) an illustration of the input sample with the reconstruction.

For stacked autoencoder, we first train the first layer autoencoder, subsequently the outputs of the previous layers would be used as the input. Train the parameters of each layer individually while freezing parameters for the remainder of the model. This parameter training method was called greedy layer-wise training [36].

III. Electrocardiography Annotation

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 [22]. 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.

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 extracted by 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. Due to the particularity in medical data collection and the medical background requirements for data annotation , the developments in ECG classification and detection were not as flourishing as the similar research topics like speech recognition, natural language processing, image recognition etc.

A. Arrhythmia Categories

The MIT-BIH Arrhythmia Database was the first generally available set of standard test material for evaluation of arrhythmia detectors, and it has been used for that purpose as well as for basic research into cardiac dynamics at about 500 sites worldwide since 1980. It played an important role in stimulating manufacturers of arrhythmia analyzers to compete on the basis of objectively measurable performance. The annotations for the ECG categories adopted the ANSI/AAMI EC57: 1998/(R)2008 standard AAMI (2008), which recommended to group the heartbeats into five classes: on-ectopic beats (N as the Fig. 2 (a)); supraventricular ectopic beat (S as the Fig. 2 (b)); ventricular ectopic beat (V as the Fig. 2 (c) and (d)); fusion of a V and a N (F); unknown beat type (Q). These classes or labels are widely used in the ECG classification tasks related literatures. Generally, the normal beat, supraventricular ectopic beat and the ventricular ectopic beat categories were used much more frequently while the

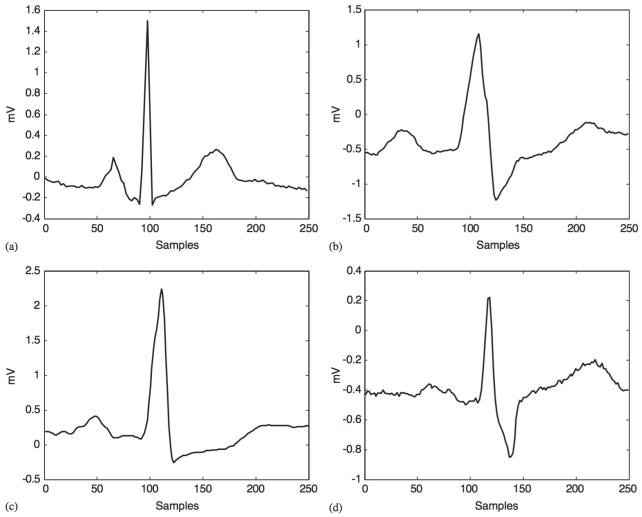

Fig. 2. Waveforms of the ECG beats: (a) Typical N label for normal beat; (b) Typical S label for congestive heart failure beat; (c) Typical V label of Ventricular tachyarryhthmia beat; (d) Typical V label of atrial fibrillation beat.

unknown beat type were abandoned because of its clinical valueless.

B. Electrocardiography Classification Tasks

IV. Classification With Deep learning Stacked Autoencoder Structures

Since the SAE based deep structure was capable to extract representations for the raw inputs, it is simpler and more convenient to extract the features for ECG classification instead of discovery or create features by researchers. Thought it should be mentioned that for clinical reasons some features from the researchers might be great of importance, the features were limited in the morphology aspect, which lacks of the ability to use some importance features in the transform-domain features or statistical representations. Another motivation for the deep features was, in the real use of the long-term ECG monitoring, the variances of the data collection devices and the difference among different objects. For example, different Holter manufactories use different collecting chips, or different analog-to-digital sampling rate, or even different digital signal filter parameters. Another factor is the physiological differences of the objects. According to these factors, using the single morphology or time domain features were insufficient, the probability distribution of a certain class is nonexclusive and has variations over multiple directions in the feature space, which makes it impossible to analyze the waveform point by point in the complicated real situation. More robust and invariant features should be extracted and used in the auto-analysis scenes. It is believed that deep architectures can potentially lead to progressively more abstract features at higher layers of feature, and more abstract features are generally invariant to most local changes of the input [41].

To tackle with these problems, with the stacked autoencoder model the deep invariant features of raw ECG samples can be learned layer by layer. Same as the traditional learning systems, after the feature extracting process, a classifier would be constructed behind the neural network to finish the classification task (Fig. 3).

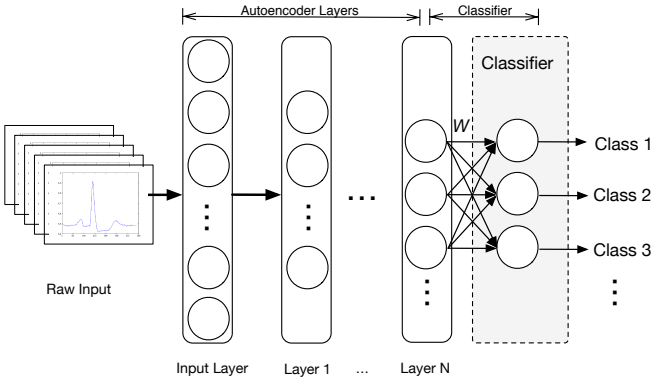

Fig. 3. The multi-layer autoencoder with a subsequent classifier.

A. Hierarchal Pre-training

1) Sparsity: In order to guarantee the representation expression ability of the model, a sparsity constraint is imposed on the hidden units. Let $a_j^{(l)}$ denotes the activation (output of the activation function $f(\cdot)$, in the input layer, $a_i^{(1)} = x_i$), then

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})] \tag{6}$$

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

$$\hat{\rho}_j = \rho \tag{7}$$

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). 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:

$$\sum_{j=1}^{s_2} KL(\rho||\hat{\rho}) = \sum_{j=1}^{s_2}\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{8}$$

is chosen as the penalty term. KL-divergence is a standard function for measuring how different two different distributions are.

2) Cost Function: The first step is to learn a deep feature for the ECG samples via pretraining the SAE hierarchically. In Section II, the outline of training method had been derived. Mathematically, the overall cost function of neural network is denoted by $J(W,b)$ which was defined by:

$$\begin{aligned} J(W,b) = [\frac{1}{m}\sum_{i=1}^{m}(\frac{1}{2}\|h_{W,b}(x^{(i)}) - y^{(i)}\|^2)] \\ +\frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}{W_{ji}^{(l)}}^2 \end{aligned} \tag{9}$$

TABLE I
AAMI CLASSES MAPPED FROM MIT-BIH ARRHYTHMIA & LONG-TERM DATABASE TYPES

AAMI heartbeat classes	N	S	V	F	Q
Description	Any heartbeat not in the S,V,F or Q class	Supraventricular ectopic beat	Ventricular ectopic beat	Fusion beat	Unknown beat
MIT-BIH heartbeat types (codes)	Normal beat (1) Left bundle branch block beat (2) Right bundle branch block beat (3) Nodal escape beat (11) Atrial escape beat (34)	Aberrated atrial premature beat (11) Nodal (junctional) premature beat (7) Atrial premature contraction (8) Premature or ectopic supraventricular beat(9)	Ventricular escape beat(10) Premature ventricular contraction (5) Ventricular flutter wave (31)	Fusion of ventricular & normal beat (6)	Paced beat (12) Unclassifiable beat (13) Fusion of paced and normal (38)
MITBIH-AR[a](100,687)	89,925[b]	2,774	7,171	802	15
MITBIH-LT(667,347)	600,197	150	64,090	2,906	0

[a] Recording 102, 104, 107, 217 were removed.
[b] The counts listed in the table may differ from other literatures [2] due to the computation need.

in the first part of which is an average sum-of-square error term: x is the input as the training examples; y denotes the output values (in the autoencoder case, y was set as $y = x$); $(x^{(i)}, y^{(i)})$ denotes the i-th training example; $W^{(l)}_{(ij)}$ denotes the weights of the connections between unit j in layer l, and unit i in layer $l+1$; $b^{(l)}_i$ denotes the bias term associated with unit i in layer $l+1$. For the regularization term which tends to decrease the magnitude of the weights and helps prevent of overfitting: λ was adopted as the weight decay parameter while n_l stands for number of layers, s_l denotes number of units in layer l. Add the sparsity parameters, in the autoencoder neural network training, the cost function of $J_{sparse}(W, b)$ was defined as:

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(\rho||\hat{\rho_j}) \tag{10}$$

β denotes the weight of the sparsity penalty term.

3) Activation Function: The nonlinear mapping activation function $f(\cdot)$ is set to be a $sigmoid$ function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{11}$$

both in the "encoder" and "decoder". While training the autoencoder, the "tied weights" are used with the greedy layer-wise approach.

4) Training: The stacked autoencoder consists multiple layers of sparse autoencoders in which the outputs of each layer is wired to the inputs of the successive layer. Sequently we use the definitions above, let $W^{(k)}, \hat{W}^{(k)}, b^{(k)}, \hat{b}^{(k)}$, denote the parameters for the k-th autoencoder. The encoding step for the stacked autoencoder is given by running the encoding step of each layer forwardly:

$$a^{(l)} = f(z^{(l)}) \tag{12}$$

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)} \tag{13}$$

The decoding step is given by running the decoding stack of each autoencoder in reverse order (in the reconstruction encoding and decoding structure):

$$a^{(n+l)} = f(z^{(n+l)}) \tag{14}$$

$$z^{(n+l+1)} = \hat{W}^{(n-l)} a^{(n+l)} + \hat{b}^{(n-l)} \tag{15}$$

Our goal is to minimize the cost function $J_{sparse}(W, b)$(with the sparse penalty term) as the function of W and b. To train the network,

After training the network, the reconstruction layers are removed, then stacked autoencoder is constructed with encoders layer by layer. The learned features which we interested in is contained within $a(n)$, which is the activations of the deepest layer of hidden units.

This vector gives us a representation of the input in terms of higher-order features, which can be used for classification problems by feeding $a(n)$ to a classifier. First initialize each parameter $W^{(l)}_{ij}$ and each $b^{(l)}_i$ to a small random value near zero, and then apply an optimization algorithm such as batch gradient descent. Then the iteration of gradient descent updates of W and b were:

$$W^{(l)}_{ij} := W^{(l)}_{ij} - \alpha \frac{\partial}{\partial W^{(l)}_{ij}} J_{sparse}(W, b) \tag{16}$$

and

$$b^{(l)}_i := b^{(l)}_i - \alpha \frac{\partial}{\partial b^{(l)}_i} J_{sparse}(W, b) \tag{17}$$

where α is the learning rate.

Given the training example (x, y)(here in autoencoder $(x, \hat{x})$), we first run forward to compute the activations of the network, as the output values were the results of hypothesis $h_{W,b}(x)$ as in equation (9). For each node i in layer l, let $\delta^{(l)}_i$ denotes the error term, then the output layer is $\delta^{(n_l)}_i$ means the difference between network's activation and y ($\hat{x}$ in autoencoder) and the hidden layer $\delta^{(l)}_i$ based on the weighted average of the error terms of the nodes that uses $a^{(l)}_i$ as an input.

B. Fine-tuning and Classification

To utilize the learned feature integrate the networks structure, we need to fine-tune the pre-trained network, which uses softmax as the output-layer activation.

Algorithm 1 The Backpropagation Algorithm

1: Random initialize the parameters and adopt the feedforward calculation to compute the activations for layers L_2, L_3 to the output layer L_{n_l}.

2: For the output layer unit i,

$$\begin{aligned}\delta_i^{(n_l)} &= \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \left\| y - h_{W,b}(x) \right\|^2 \\ &= -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})\end{aligned} \tag{18}$$

3: For the hidden layer l,

$$\begin{aligned}\delta_i^{(l)} = &\left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)}) \\ &+\beta(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i}))f'(z_i^{(l)})\end{aligned} \tag{19}$$

4: Compute the partial derivatives:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W,b;x,y) = a_j^{(l)} \delta_i^{(l+1)} \tag{20}$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W,b;x,y) = \delta_i^{(l+1)} \tag{21}$$

5: Update the parameters:

$$\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \alpha[(\frac{1}{m}\Delta \mathbf{W}^{(l)}) + \lambda \mathbf{W}^{(l)}] \tag{22}$$

$$\mathbf{b}^{(l)} = \mathbf{b}^{(l)} - \alpha[\frac{1}{m}\Delta \mathbf{b}^{(l)}] \tag{23}$$

1) Softmax model: The softmax ensures the activation of each output unit sums to 1, so that we can deem the output as a set of conditional probabilities. Softmax regression is a supervised learning algorithm, we use softmax regression as the Fine tuning treats all layers of a stacked autoencoder as one single model, so we can improve all the weight parameters in one iteration.

In the softmax regression model, the hypothesis function is set as:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} \tag{24}$$

$$= \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{25}$$

in which output a k dimensional vector denotes k estimated probabilities which stand for k different possible value of the probability. $\theta_1, \theta_2, \ldots, \theta_k \in \Re^{n+1}$ are the parameters of the model. The term $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}$ normalizes the distribution which sums to one. The cost function in softmax supervised learning is:

$$\begin{aligned}J(\theta) = -\frac{1}{m} &\left[\sum_{i=1}^{m} \sum_{j=1}^{k} 1\left\{ y^{(i)} = j \right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \right] \\ &+ \frac{\lambda}{2} \sum_{i=1}^{k} \sum_{j=0}^{n} \theta_{ij}^2\end{aligned} \tag{26}$$

in which $1\{\cdot\}$ is the indicator function. Use the iterative optimization algorithm such as gradient descent method. Taking derivatives, the gradient is:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [x^{(i)}(1\{y^{(i)} = j\} - p(y^{(i)} = j|x^{(i)};\theta))] \tag{27}$$

in which $\nabla_{\theta_j} J(\theta)$ is a vector, so that its l-th element is $\frac{\partial J(\theta)}{\partial \theta_{jl}}$ the partial derivative of $J(\theta)$ with respect to the l-th element of θ_j. Perform the update rule $\theta_j := \theta_j - \alpha \nabla_{\theta_j} J(\theta) (foreach j = 1, \ldots, k)$, we can get the parameters.

2) Fine-tuning with softmax regression: The output-layer size is set to be the same as the usual four categories of arrhythmia, and the input layer has the same size as the stacked autoencoder learned features. Since the softmax regression is implemented as a single-layer neural network, it can be merged with the former layers of networks to get a deep classifier. The fine-tuning step would use the whole combined architecture for optimization with the backpropagation algorithm as Algorithm 1 illustrated, which has a small learning rates on the former autoencoder layers due to the large amount of unsupervised learning process.

Algorithm 2 Stacked Autoencoder with Softmax Regression

1: **begin**

2:

3: **end**

V. Experimental Results

VI. Discussion and Conclusions

Acknowledgment

This study was financed partially by the National 863 Program of China (Grant No. 2012AA02A604), the Next generation communication technology Major project of National S&T (Grant No. 2013ZX03005013), the Key Research Program of the Chinese Academy of Sciences, and the Guangdong Innovation Research Team Funds for Image-Guided Therapy and Low-cost Healthcare.

References

[1] T. Mar, S. Zaunseder, J. Martinez, M. Llamedo, and R. Poll, "Optimization of ecg classification by means of feature selection," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 8, pp. 2168–2177, Aug 2011.

[2] P. de Chazal, M. O'Dwyer, and R. Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.

[3] F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 5, pp. 667–677, Sept 2008.
[4] W. Jiang and S. Kong, "Block-based neural networks for personalized ecg signal classification," *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1750–1761, 2007.
[5] T. Olmez, "Classification of ecg waveforms by using rce neural network and genetic algorithms," *Electronics Letters*, vol. 33, no. 18, pp. 1561–1562, Aug 1997.
[6] C.-W. Lin, Y.-T. Yang, J.-S. Wang, and Y.-C. Yang, "A wearable sensor module with a neural-network-based activity classification algorithm for daily energy expenditure estimation," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 5, pp. 991–998, Sept 2012.
[7] S. Osowski and T. H. Linh, "Ecg beat recognition using fuzzy hybrid neural network," *Biomedical Engineering, IEEE Transactions on*, vol. 48, no. 11, pp. 1265–1271, Nov 2001.
[8] M. Kundu, M. Nasipuri, and D. Basu, "A knowledge-based approach to ecg interpretation using fuzzy logic," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 2, pp. 237–243, Apr 1998.
[9] R. Andreao, B. Dorizzi, and J. Boudy, "Ecg signal analysis through hidden markov models," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 8, pp. 1541–1549, 2006.
[10] D. Coast, R. Stern, G. Cano, and S. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *Biomedical Engineering, IEEE Transactions on*, vol. 37, no. 9, pp. 826–836, 1990.
[11] Y. Zhu, A. Shayan, W. Zhang, T. L. Chen, T.-P. Jung, J.-R. Duann, S. Makeig, and C.-K. Cheng, "Analyzing high-density ecg signals using ica," *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 11, pp. 2528–2537, Nov 2008.
[12] A. Kampouraki, G. Manis, and C. Nikou, "Heartbeat time series classification with support vector machines," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 4, pp. 512–518, 2009.
[13] A. Khandoker, M. Palaniswami, and C. Karmakar, "Support vector machines for automated recognition of obstructive sleep apnea syndrome from ecg recordings," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 1, pp. 37–48, Jan 2009.
[14] C.-P. Shen, W.-C. Kao, Y.-Y. Yang, M.-C. Hsu, Y.-T. Wu, and F. Lai, "Detection of cardiac arrhythmia in electrocardiograms using adaptive feature extraction and modified support vector machines," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7845 – 7852, 2012.
[15] I. Jekova, G. Bortolan, and I. Christov, "Assessment and comparison of different methods for heartbeat classification," *Medical Engineering & Physics*, vol. 30, no. 2, pp. 248–257, 2008.
[16] I. Christov, G. Gomez-Herrero, I. Krasteva, I. Jekova, A. Gotchev, and K. Egiazarian, "Comparative study of morphological and time frequency ecg descriptors for heartbeat classification," *Medical Engineering & Physics*, vol. 28, no. 9, pp. 876–887, 2006.
[17] C. Ye, B. Kumar, and M. Coimbra, "Heartbeat classification using morphological and dynamic features of ecg signals," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 10, pp. 2930–2941, Oct 2012.
[18] O. T. Inan, L. Giovangrandi, and G. T. A. Kovacs, "Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2507–2515, 2006.
[19] S. Banerjee and M. Mitra, "Application of cross wavelet transform for ecg pattern analysis and classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 326–333, 2014.
[20] T. Stamkopoulos, K. Diamantaras, N. Maglaveras, and M. Strintzis, "Ecg analysis using nonlinear pca neural networks for ischemia detection," *Signal Processing, IEEE Transactions on*, vol. 46, no. 11, pp. 3058–3067, Nov 1998.
[21] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo, "Clustering ecg complexes using hermite functions and self-organizing maps," *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 7, pp. 838–848, Jul 2000.
[22] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods And Tools for ECG Data Analysis*. Norwood, MA, USA: Artech House, Inc., 2006.
[23] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13).
[24] H. Nishizawa, T. Obi, M. Yamaguchi, and N. Ohyama, "Hierarchical clustering method for extraction of knowledge from a large amount of data," *Optical Review*, vol. 6, no. 4, pp. 302–307, 1999.
[25] C. Maier, H. Dickhaus, and J. Gittinger, "Unsupervised morphological classification of qrs complexes," in *Computers in Cardiology, 1999*, 1999, pp. 683–686.
[26] S. Palreddy, W. J. Tompkins, and Y. H. Hu, "Customization of ecg beat classifiers developed using som and lvq," in *Engineering in Medicine and Biology Society, 1995., IEEE 17th Annual Conference*, vol. 1, Sep 1995, pp. 813–814 vol.1.
[27] M. Risk, J. Sobh, and J. Saul, "Beat detection and classification of ecg using self organizing maps," in *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, vol. 1, Oct 1997, pp. 89–91 vol.1.
[28] P. Tadejko and W. Rakowski, "Hybrid wavelet-mathematical morphology feature extraction for heartbeat classification," in *EUROCON, 2007. The International Conference on Computer as a Tool*, Sept 2007, pp. 127–132.
[29] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks, 1989. IJCNN., International Joint Conference on*, 1989, pp. 593–605 vol.1.
[30] Y. Bengio, "Learning deep architectures for ai," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.
[31] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
[32] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.
[33] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[34] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
[35] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.
[36] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.
[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
[38] D. H. Ballard, "Modular learning in neural networks." in *AAAI*, 1987, pp. 279–284.
[39] N. Japkowicz, S. Jose Hanson, and M. A. Gluck, "Nonlinear autoassociation is not equivalent to pca," *Neural Comput.*, vol. 12, no. 3, pp. 531–545, Mar. 2000.
[40] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 6, pp. 2094–2107, 2014.
[41] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

PLACE PHOTO HERE

Jan Doe Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.