Informatics in Medicine Unlocked 18 (2020) 100307

Contents lists available at ScienceDirect

Informatics in Medicine Unlocked

journal homepage: http://www.elsevier.com/locate/imu

Improved sparse autoencoder based artificial neural network approach for prediction of heart disease

Ibomoiye Domor Mienye [a,*], Yanxia Sun [a], Zenghui Wang [b]

[a] Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg, 2006, South Africa
[b] Department of Electrical and Mining Engineering, University of South Africa, Florida, 1709, South Africa

ARTICLE INFO

Keywords:
Sparse autoencoder
Deep learning
Unsupervised learning
ANN
Heart disease

ABSTRACT

In this paper a two stage method is proposed to effectively predict heart disease. The first stage involves training an improved sparse autoencoder (SAE), an unsupervised neural network, to learn the best representation of the training data. The second stage involves using an artificial neural network (ANN) to predict the health status based on the learned records. The SAE was optimized so as to train an efficient model. The experimental result shows that the proposed method improves the performance of the ANN classifier, and is more robust as compared to other methods and similar scholarly works.

1. Introduction

Heart disease (HD) has been classified to be among the most deadly human diseases, and its diagnosis and treatment are quite complex. Predicting heart disease is challenging but necessary since the mortality rate can be greatly reduced if the disease is detected early and preventive measures taken [1]. Therefore, accurate prediction of patient's heart disease risk is very important to reduce their associated risks of severe heart conditions [2]. To achieve this and save human lives there is need to efficiently process raw heart data for proper classification. In order to improve the performance of HD models, several researchers have used machine learning algorithms to build various models and they have achieved some success, for example in Ref. [1], the authors proposed a method to improve the prediction accuracy by detecting the important features and performing classification using a hybrid random forest. They achieved an accuracy of up to 88.47%, sensitivity of 92.8%, specificity of 88.6, and precision of 87.5%.

In [2], the authors proposed a framework to predict heart disease, where they performed feature reduction which had an impact on the performance of various classifiers they experimented on, with the support vector machine having accuracy of 88%. Similarly, in Ref. [3] a method is proposed that generates decision rules in order to effectively classify heart disease risk level, and the experimental result showed that their method achieved an accuracy of 86.7%. However, to further improve on the progress made so far, other methods need to be utilized. Deep learning has been successfully applied in several areas, especially in image and visual analysis, and in recent times, deep autoencoders have achieved superior performance in some unsupervised machine learning tasks. It is a possible solution to the challenge of heart disease prediction due to its exceptional performance in learning good feature representations in complex and large datasets.

An autoencoder basically comprises of two functions, an encoder that maps the original d-dimensional input data to an intermediate or hidden representation, and a decoder that maps the hidden representation back to a d-dimensional vector which is expected to be as close as possible to the original input of the encoder The process is called reconstruction, whereas the difference between the decoder output and encoder input is called the reconstruction error [4]. Research has shown that the classification performance can be improved when representations are learnt in a way that encourages sparsity. In sparse autoencoders, the training criterion includes a sparsity penalty on the code layer.

In this paper we present an efficient and reliable sparse autoencoder (SAE) approach to perform unsupervised feature leaning and prediction of heart disease. We focus on developing a SAE model to learn effective features from the HD dataset and then perform classification using the learned features. The model is optimized using the adaptive moment estimation (Adam) algorithm to achieve dynamic adjustment of different parameters, and a batch normalization technique is applied to avoid overfitting and to improve the performance, speed, and stability of the model. The optimized setting also ensures reconstruction error is significantly minimized. The effectiveness of the proposed method is

* Corresponding author.
E-mail address: ibomoiyem@uj.ac.za (I.D. Mienye).

https://doi.org/10.1016/j.imu.2020.100307
Received 28 January 2020; Received in revised form 26 February 2020; Accepted 2 March 2020
Available online 5 March 2020

verified by comparing with a standalone ANN, well-performing algorithms such as k-nearest neighbor (KNN), classification and regression tree (CART), Logistic regression (LR), linear discriminant analysis (LDA), and other scholarly works. The result shows that our proposed approach obtains superior classification performance.

2. Related works

This section discusses some previous works that studied and applied sparse autoencoders. In recent time autoencoders have found significant applications in various unsupervised learning tasks in several application areas. In Ref. [5] a method was proposed that combined SVM and sparse autoencoder. The rational there was that the classical SVM has limitations on large scale applications; hence, the need to use a sparse autoencoder to improve the performance. The authors used multiple layers of sparse autoencoder to perform feature learning and used the SVM for classification, thereby improving the performance of the SVM in handling large scale datasets.

In a similar research, the authors in Ref. [6] proposed a method to perform feature learning using sparse autoencoders to improve the performance of the regression model on real-valued time series data. The architecture consists of different layers of sparse autoencoders. The aim of the research was to enhance vehicular traffic flow forecasting. And in a bid to increase the accuracy of the sparse autoencoder, they proposed a cascaded model which leverages on the combination of low and high level features, and a stochastic gradient descent algorithm was employed as the regression method. Another research was conducted to carry out anomaly detection by learning anomaly scores [7]. The idea in this research was that since the reconstruction error of background data is small while that of anomaly data is relatively higher, the reconstruction error can be used as an anomaly score. Therefore, they proposed a sparse autoencoder based anomaly detection method which uses a dual concentric window.

In [8], a sparse representation based classification method was proposed using a transductive deep learning based formulation. The network comprises of a fully connected layer and a convolutional autoencoder. The fully connected layer is placed between the encoder and decoder, and its function is to find the sparse representation, whereas the autoencoder network learns effective deep features for classification. When the estimated sparse codes are used for classification of some datasets, the proposed method showed improved performance.

In [9] the authors proposed a method to add a distance constraint to stacked sparse autoencoders (SSAEs) in order to form a novel distance constraint SSAE network. The distance constraint enhances the uniqueness between target pixels and various background pixels in the feature space. Hence, by utilizing the discriminative features learned from the distance constrained SSAE, a simple detector with radial basis function kernel is obtained for background suppression. Tests conducted on two hyperspectral image show that the deep spectral features learned from the distance constraint SSAE are more distinct and perform better than many detectors.

In [10] an approach was proposed to derive a formulation that effectively determines the sparse hyper-parameter in sparse autoencoders, in addition to deriving the relationship between the average activation of hidden units and sparse hyper-parameter. The authors conducted two experiments and they obtained good performance. In Ref. [11] a new method shows where a sparse autoencoder is used for automatic modulation classification. The network was trained using a non-negativity constraint algorithm. Experimental results showed that the autoencoder with the non-negativity constraint enhances the sparsity and minimizes the reconstruction error as compared to the traditional sparse autoencoder.

3. Proposed methodology

In this section, we describe the methodology used in implementing the proposed sparse autoencoder. An autoencoder is a type of unsupervised neural network architecture that replicates its input at the output. It basically consists of an encoder and a decoder. AEs aim to learn low-level representations of the input data which are then deformed back to project the original data. The encoder maps the input to a new representation. This new representation is then decoded at the output to reconstruct the input $x^{'}$ according to Equations (1) and (2), where x is the input and z the new representation.

$$\boldsymbol{Z} = \boldsymbol{h}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{1}$$

$$\boldsymbol{X}^{'} = \boldsymbol{g}(\boldsymbol{W}^{'}z + \boldsymbol{b}^{'}) \tag{2}$$

In the above formulation, $\boldsymbol{h}$ is the activation function for the hidden layer neurons and g is for the output layer neurons, $\boldsymbol{W}$ and $\boldsymbol{W}^{'}$ are weight matrices, and $\boldsymbol{b}$ and $\boldsymbol{b}^{'}$ are the encoder and decoder bias vectors, respectively. In this paper, the sigmoid activation function is utilized, which is shown in Equation (3) instead of the others such as Relu, Tanh etc.

$$\boldsymbol{h} = \boldsymbol{g} = \frac{1}{1 + \boldsymbol{e}^{-x}} \tag{3}$$

The reconstruction error function E between the input x and reconstructed input $x^{'}$ uses the mean squared error (MSE) function shown in Equation (4).

$$\boldsymbol{E} = \frac{1}{\boldsymbol{N}} \sum_{i=1}^{N} \boldsymbol{x}_i + \boldsymbol{x}_i^{'2} \tag{4}$$

N represents the number of input samples. However, in this research a sparse autoencoder is utilized to obtain an effective low-level representation of the input data under sparse constraints. Hence, sparsity is introduced by including regularization to the cost function. Let $\widehat{\boldsymbol{p}_i}$ be the average activation of neurons in the hidden layer.

$$\widehat{\boldsymbol{p}_i} = \frac{1}{\boldsymbol{n}} \sum_{j=1}^{n} \boldsymbol{z}_i(\boldsymbol{x}_j) \tag{5}$$

From Equation (5) I, n, and j represents the ith neuron, total number of training samples, and jth training sample respectively. The average activation $\widehat{\boldsymbol{p}_i}$ approaches p that is a constant close to zero. Hence, the Kullback-Leibler (KL) divergence is used to add the regularizer to the cost function. The KL divergence is introduced to achieve sparsity.

$$\boldsymbol{\Omega}_{sparsity} = \sum_{i=1}^{d} \boldsymbol{p}\, \boldsymbol{log}\left(\frac{\boldsymbol{p}}{\widehat{\boldsymbol{p}_i}}\right) + (1 - \boldsymbol{p}) \boldsymbol{log}\left(\frac{1 - \boldsymbol{p}}{1 - \widehat{\boldsymbol{p}_i}}\right) \tag{6}$$

From Equation (6) d stands for the total number of neurons in a layer, whereas p is the sparsity proportion, which is the needed activation value. Therefore, the SAE error function now comprises of the mean square error and the regularization terms. Furthermore, in order to control the weights and prevent overfitting, L2 regularization (L2R) is introduced in the cost function.

$$\boldsymbol{\Omega}_{weights} = \frac{1}{2} \sum_{I}^{L} \sum_{j}^{N} \sum_{i}^{K} \left(\boldsymbol{w}_{ji}^{(I)}\right) \tag{7}$$

L and K represent the number of hidden layers and number of features in a sample, respectively [12]. We went further to include the weight attenuation units as seen in Equation (8). After adding the various regularization terms, i.e. Equations (6) and (7) into Equation (4) which is the reconstruction error, our cost function becomes:

$$\boldsymbol{E} = \frac{1}{\boldsymbol{N}} \sum_{n=1}^{N} \sum_{k=1}^{K} (\boldsymbol{x}_{kn} - \widehat{\boldsymbol{x}}_{kn})^2 + \boldsymbol{\lambda}^* \boldsymbol{\Omega}_{weights} + \boldsymbol{\beta}^* \boldsymbol{\Omega}_{sparsity} \tag{8}$$

There are three optimization parameters here: λ which is the coefficient for L2R and it prevents overfitting, the second parameter is β the sparsity regularization parameter, and it sets the sparsity penalty term. Lastly, p the sparsity proportion controls the needed sparsity level. The optimization parameter values for λ, β, and p are 0.0001, 0.01, and 0.5 respectively.

Furthermore, in our quest to train a robust SAE, the Adam algorithm [13] is used instead of the classical stochastic gradient descent or other variants. The Adam optimization algorithm avails us the opportunity to use a different learning rate for various parameters and to realize dynamic adjustment of various parameters by obtaining the gradient first-order moment estimate m_t and second-order moment estimate v_t shown in Equations (9)–(11).

$$m_t = \beta_1 m_{t-1} + (1-\beta_1).g_t \tag{9}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2).g_t^2 \tag{10}$$

$$g_t \leftarrow \nabla_\theta J_t(\theta_{t-1}) \tag{11}$$

Where β_1 and β_2 are the first-order exponential damping decrement and second-order exponential damping decrement respectively. Whereas g_t is the gradient of the parameters at timestep t in the cost function E above. Computer bias-corrected for m_t and v_t:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{12}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{13}$$

Update parameters:

$$\theta_{t+1} = \theta_t - \frac{\gamma}{\sqrt{\hat{v}_t} + \xi}\hat{m}_t \tag{14}$$

γ represents the update step size. ξ takes a small constant in order to stop the denominator from becoming zero. The procedure for the proposed sparse autoencoder is shown in Algorithm 1.

4. Dataset and performance indices

The HD dataset is obtained from the Kaggle website [14]. The dataset was obtained after a cardiovascular study on residents of Framingham, Massachusetts. The reason for using the Kaggle Framingham heart dataset is because it has a higher number of instances (4238) compared to the Cleveland, Hungarian, and Long Beach heart datasets which have 303, 294, and 200 instances respectively. The dataset includes patient information and it aims at predicting their 10-year risk of future coronary heart disease (CHD). It consists of 4238 samples and 16 features. Every feature is a possible risk factor, and they include behavioral, demographic, and medical risk factors. The dataset contains missing attributes. After dropping rows with missing attributes, 3656 records were left; among which 3099 were negative and 557 positive. We utilized 70-30% train-test data partitioning approach.

To effectively assess the performance of our method, some metrics including accuracy, precision, sensitivity, and F1 score were used. The various performance metrics are defined as follows:

$$\textbf{classification accuracy} = \frac{\mathbf{TP}+\mathbf{TN}}{\mathbf{TP}+\mathbf{TN}+\mathbf{FP}+\mathbf{FN}} \tag{15}$$

$$\textbf{Sensitivity} \Big/ \textbf{recall} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}} \tag{16}$$

$$\textbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FP}} \tag{17}$$

$$\mathbf{F1} = \frac{2^*\textbf{Precision}^*\textbf{Sensitivy}}{\textbf{Precision}+\textbf{Sensitivity}} \tag{18}$$

TN and TP stands for true negative and true positive, and they are the number of negative and positive patients that are classified correctly. Whereas FP and FN represents false positive and false negative and they represent the number of positive and negative patients that was wrongly predicted.

5. Experimental setup

5.1. The approach

The proposed approach involves two steps. Firstly, the dataset is preprocessed to make it suitable for building our model. After preprocessing, the dataset is then split into train and test sets. The SAE was trained using the negative instances in the train set. The rationale here is that if the model can learn accurate representation of these negative samples, when presented with unseen samples either positive or negative it should easily identify them. Our interest is to obtain the latent representation of the input learned by the SAE model and use it to train the ANN. Hence, once the SAE model is trained, the encoding part is used to create another network. This network is used to transform the train and test sets, including both positive and negative samples. This simply transforms the dataset to a low-dimensional representation dataset. The second stage is to train the ANN classifier using the

Algorithm 1. Proposed methodology for sparse autoencoder

Input:
train set x
Process:
1. Start
2. Initialize h, g, w, w', b, b'
3. Obtain the reconstruction error function E according to equation 4
4. Add sparsity regularizer to cost function according to equation 6
5. Add L2 regularization to the cost function according to equation 7
6. Train network
7. End

Output:
reconstructed representation of the input x'

transformed train set and then make predictions on the test set. The proposed method prevents any possible data leakage and overfitting.

5.2. Model parameters

The experiment was conducted using a computer that has the following specification: Intel Core i5-6300U, 2.40 GHz, and 16 GB RAM. Python was utilized as the programming language. While there is no rule of thumb to obtain the number of hidden layers and neurons in various layers of an autoencoder network, it is important for us to obtain a good network structure according to our experimental settings for optimal performance. From several experiments carried out, the experimental parameters in Table 1 provide optimal performance.

In our SAE network the layers (100, 75, 50, and 25) in encoder and decoder are symmetric around the bottleneck. The Adam optimization algorithm takes the default parameter as proposed by Kingma and Ba [13]. Also the batch normalization technique proposed by Joffe and Szegedy [15] was applied to improve the performance, speed, and stability of the SAE model. The reason for using a batch size of 32 is due to the fact that smaller batch sizes allow the model to converge faster.

6. Results and discussion

In order to demonstrate the effectiveness and performance of the features learned by our proposed sparse autoencoder, first we trained an ANN using the raw data and secondly using the learned features as shown in Table 2.

The result shows that the low-dimensional features learned by our sparse autoencoder improves the classification performance of the ANN, since the proposed method performs better than the ANN, which is a demonstration of the fact that the sparse autoencoder is capable of retaining the information in the input data while obtaining optimal low dimensional features. The model performs well on the test data, which is a major pointer to its efficiency since the model has not previously seen the data. We also carried out comparative experiments using five base classifiers including KNN, CART, Logistic regression, Naïve Bayes, and LDA. The experimental results are summarized in Table 3. From this table, it is evident that the proposed method outperforms the other algorithms.

Furthermore, the proposed approach is compared with some recent scholarly works as shown in Table 4, and it shows better performance than those reported in the literature. Lastly, as a means to further show how effective the proposed method performs, we conducted further experiments using the cervical cancer risk factors dataset [16], as shown in Table 5.

From the results obtained so far, it can be seen that the proposed method shows significant improvement compared to the other methods in terms of classification performance. And it is clear that the proposed sparse autoencoder improves the accuracy of the ANN compared to a case where the ANN alone was used to make predictions. The results also show that improved performance can be achieved not only by improving the structure of the neural network or performing hyper-parameter tuning of algorithms, but also by improving the preprocessing stage of the classification process.

Table 1
Parameters of the SAE.

Parameter	Value
Nodes in input layer	15
Neurons in 1st hidden layer	100
Neurons in 2nd hidden layer	75
Neurons in 3rd hidden layer	50
Neurons in 4th hidden layer	25
Neurons in Bottleneck layer	7
Sparse parameter	0.05
Pre-training learning rate	0.01
Batch size	32

Table 2
Performance comparison between ANN and the proposed method.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
ANN	85	72	85	78
Proposed SAE + ANN	90	89	91	90

Table 3
Performance comparison between other algorithms and the proposed method.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
KNN	81	75	81	76
CART	76	75	76	75
LR	83	84	83	77
Naïve Bayes	82	78	82	79
LDA	83	81	83	78
Proposed SAE + ANN	90	89	91	90

Table 4
Performance comparison between the proposed method and other recent scholarly works.

Author	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Mohan et al. [1]	88.47	87.5	92.8	90
Haq et al. [2]	89	–	77	–
Repaka et al. [9]	89.77	–	–	–
Latha and Jeeva [10]	85.48	–	–	–
Our approach	90	89	91	90

Table 5
Performance comparison between the proposed method and other algorithms on the cervical cancer dataset.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
SVM	93	98	95	96
Decision tree	90	93	96	94
KNN	93	98	95	96
Naïve Bayes	94	90	93	91
ANN	94	98	91	94
Proposed SAE + ANN	98	96	98	97

7. Conclusion

In this research, an improved sparse autoencoder based ANN is proposed to aid the prediction of heart disease. The sparse autoencoder was used to learn the best representation of the data while the ANN used to make predictions based on the learned records. The SAE was optimized using Adam algorithm and batch normalization applied. The accuracy of the model on test data was 90%. Compared to some traditional machine learning approaches and ANN, our proposed method showed improved performance.

Funding

This research is supported partially by South African National Research Foundation Grants (No. 112108 and 112142), and South African National Research Foundation Incentive Grant (No. 95687), Eskom Tertiary Education Support Programme (Y. Sun, Z. Wang),

Research grant from URC of University of Johannesburg.

Consent

None.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ibomoiye Domor Mienye: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Investigation, Software, Validation. **Yanxia Sun:** Investigation, Supervision, Software, Validation, Writing - review & editing. **Zenghui Wang:** Supervision, Writing - review & editing.

Acknowledgments

This research is supported partially by South African National Research Foundation Grants (No. 112108 and 112142), and South African National Research Foundation Incentive Grant (No. 95687), Eskom Tertiary Education Support Programme (Y. Sun, Z. Wang), Research grant from URC of University of Johannesburg.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.imu.2020.100307.

References

[1] Mohan S, Thirumalai C, Srivastava G. Effective heart disease prediction using hybrid machine learning techniques. IEEE Access 2019;7:81542–54. https://doi.org/10.1109/ACCESS.2019.2923707.

[2] Haq AU, Li JP, Memon MH, Nazir S, Sun R. A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms. Mobile Inf Syst Dec.2018;2018:1–21. https://doi.org/10.1155/2018/3860146.

[3] Purushottam, Saxena K, Sharma R. Efficient heart disease prediction system. Procedia Computer Science Jan. 2016;85:962–9. https://doi.org/10.1016/j.procs.2016.05.288.

[4] İrsoy O, Alpaydın E. Unsupervised feature extraction with autoencoder trees. Neurocomputing Oct.2017;258:63–73. https://doi.org/10.1016/j.neucom.2017.02.075.

[5] Ju Y, Guo J, Liu S. A deep learning method combined sparse autoencoder with SVM. In: 2015 international conference on cyber-enabled distributed computing and knowledge discovery; 2015. p. 257–60. https://doi.org/10.1109/CyberC.2015.39.

[6] Moussavi-Khalkhali A, Jamshidi M. Constructing a deep regression model utilizing cascaded sparse autoencoders and stochastic gradient descent. In: 2016 15th IEEE international conference on machine learning and applications (ICMLA); 2016. p. 559–64. https://doi.org/10.1109/ICMLA.2016.0098.

[7] Chang S, Du B, Zhang L. A sparse autoencoder based hyperspectral anomaly detection algorihtm using residual of reconstruction error. In: Igarss 2019 - 2019 IEEE international geoscience and remote sensing symposium; 2019. p. 5488–91. https://doi.org/10.1109/IGARSS.2019.8898697.

[8] Abavisani M, Patel VM. Deep sparse representation-based classification. IEEE Signal Process Lett Jun. 2019;26(6):948–52. https://doi.org/10.1109/LSP.2019.2913022.

[9] Shi Y, Lei J, Yin Y, Cao K, Li Y, Chang C-I. Discriminative feature learning with distance constrained stacked sparse autoencoder for hyperspectral target detection. Geosci Rem Sens Lett IEEE Sep. 2019;16(9):1462–6. https://doi.org/10.1109/LGRS.2019.2901019.

[10] Wan Z, He H, Tang B. A generative model for sparse hyperparameter determination. IEEE Transactions on Big Data Mar.2018;4(1):2–10. https://doi.org/10.1109/TBDATA.2017.2689790.

[11] Ali A, Yangyu F. Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints. IEEE Signal Process Lett Nov. 2017;24(11):1626–30. https://doi.org/10.1109/LSP.2017.2752459.

[12] Zia ur Rehman M, et al. Stacked sparse autoencoders for EMG-based classification of hand motions: a comparative multi day analyses between surface and intramuscular EMG. Appl Sci Jul. 2018;8(7):1126. https://doi.org/10.3390/app8071126.

[13] Kingma DP, Ba J. Adam: A method for stochastic optimization. Jan. 2017. 1412.6980 [cs].

[14] Framingham Heart study dataset [Online]. Available, https://kaggle.com/amanajmera1/framingham-heart-study-dataset. [Accessed 24 January 2020].

[15] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Mar.2015. 1502.03167 [cs].

[16] UCI machine learning repository: cervical cancer (risk factors) data set [Online]. Available, https://archive.ics.uci.edu/ml/datasets/Cervical+cancer+%28Risk+Factors%29. [Accessed 27 January 2020].