Clinical Epidemiology and Global Health 7 (2019) 530–535

Contents lists available at ScienceDirect

Clinical Epidemiology and Global Health

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

Type 2 diabetes data classification using stacked autoencoders in deep neural networks

Kannadasan K, Damodar Reddy Edla*, Venkatanareshbabu Kuppili

Department of Computer Science and Engineering, National Institute of Technology, Goa, India

ARTICLE INFO

Keywords:
Classification
Deep neural networks
Autoencoders
Diabetes

ABSTRACT

Objective: This paper aims to classify the Pima Indians diabetes dataset with better accuracy and other evaluation metrics. The Deep Neural Network (DNN) framework will help to diagnose the patient in an effective way with higher accuracy.

Method: In this approach, we proposed a Deep Neural Network framework for diabetes data classification using stacked autoencoders. Features are extracted from the dataset using stacked autoencoders and the dataset is classified using softmax layer. Also, fine tuning of the network is done using backpropagation in supervised fashion with the training dataset. However, the medical diagnosis involves the risk factors of wrong prediction; hence we have used evaluation metrics such as precision, recall, specificity and F1 - score for the evaluation of our model and have achieved better results.

Results: The proposed framework is experimented on Pima Indians Diabetes data which has 768 patient records with 8 attributes for each record. We achieved classification accuracy of 86.26%.

Conclusion: A stacked autoencoders based Deep Learning framework for classification of Type 2 Diabetes data is proposed in this paper. This approach is experimented on UCI machine learning data and proved the outperformance over various existing classification methods.

1. Introduction

One of the most common chronic diseases is diabetes which affects around 415 million people around the world. Diabetes occurs because of the lack of insulin, as insulin plays a vital role in regulating blood sugar. When the body does not effectively use the insulin produced by pancreas, it causes Type 2 diabetes.[1] This results in abnormal increase of glucose level in the blood, which damages the functioning of organs such as kidney, heart, brain. Type 2 diabetes increases among people because of the changes in the food habits in the modern world.[2,3] Around 95% of the Type 2 diabetes cases are undiagnosed which leads to increased mortality rate. Global report of World Health Organization (WHO) stated that 82% of world's death is caused by non-communicable diseases in which diabetes contributes a major part.[4] Effects of diabetes can be suppressed when it is identified in the earlier stages. Hence classification is an important task in predicting and diagnosing the diabetes. So many research works has been proposed in predicting the diabetes in the past few years. However, new methods are needed to predict the diabetes accurately.[5]

Diagnosis of several diseases can be done using Artificial Intelligence (AI) techniques. Amongst, Deep Neural Network gives best performance in classification problems. In recent years, DNN has been used for diagnosing various diseases. A DNN is constructed by stacked autoencoders (SDA) with a softmax layer. Stacked autoencoder extracts the hidden features in the data by training the layers by the method which minimizes the reconstruction error. Softmax layer is used for the precise classification of data. Hence, DNN gives the combined benefits of SDA and softmax layer. DNN is superior to other conventional neural networks in the classification problem with the help of the aforementioned properties by having complex decision surface.[6,7,22–29]

2. Related work

In clinical diagnosis problems, classification plays a vital role in further treatment of the disease. Various studies have been done on the diabetes data classification using Pima Indian diabetes dataset.[3,5,8,9] Using the dataset from University of California, Irvine (UCI) machine learning repository, researchers used several methods for the classification problem and accuracy has been improved. A cascade learning system for diagnosis is proposed in Ref. 3 using Generalized Discriminant Analysis (GDA) and Least Square - Support Vector Machine (LS-SVM). Using LS-SVM, they obtained 78.21% accuracy with 10-fold

* Corresponding author.
E-mail addresses: kannadasankk@gmail.com (K. Kannadasan), dr.reddy@nitgoa.ac.in (D.R. Edla), venkatanaresh@nitgoa.ac.in (V. Kuppili).

https://doi.org/10.1016/j.cegh.2018.12.004
Received 7 October 2018; Received in revised form 6 December 2018; Accepted 8 December 2018
Available online 10 December 2018
2213-3984/

cross-validation and using GDA and LS-SVM, they have reported that 79.16% classification accuracy is attained. An expert system is proposed based on Principal Component analysis (PCA) and Adaptive Neuro-Fuzzy Inference System (ANFIS) on Pima Indian diabetes dataset for the diabetes disease diagnosis which obtained the classification accuracy of 89.47%. A general regression neural network is built for diagnosis of diabetes and achieved the classification accuracy of 80.21%.[3] Also a multilayer neural network based approach is proposed which achieved the classification accuracy of 77.08%. Abdullah Caliskan et al. proposed a simple training strategy for deep neural network classifier using L-BFGS algorithm where they performed evaluation is done with various datasets including Pima Indians diabetes dataset. They achieved 77.09% classification accuracy for Pima Indian diabetes dataset.[9] A deep convolutional neural network based approach is proposed to classify the diabetic retinopathy cases with image dataset.[18] A prediction model is built using data mining techniques using improved k-means algorithm and logistic regression in Ref. 19. Using machine learning techniques, diabetes mellitus affected patients are classified in Ref. 20. It is stated that they achieved precision and recall value of 0.770 and 0.775 respectively using Hoeffding Tree algorithm. An accuracy improvement model is proposed using Self Organizing Map clustering (SOM), PCA and Neural Networks (NN) for diabetes data set.[21] However, several studies has been proposed where the classification accuracy is between 59.5% and 77.7%.[3,10]

3. Terminologies

Deep learning belongs to a class of machine learning algorithms. Deep learning extracts features and transforms it using nonlinear functions. The learning method can be either supervised or unsupervised manner.[30] This section describes about the autoencoder and its process of training.

3.1. Autoencoder

An autoencoder has three layers as feedforward neural network, namely input layer, hidden layer and output layer. Autoencoders have the same number of neurons in the input layer and output layer, as it trains itself to reconstruct the given input.[11–13] The hidden layer of autoencoder encodes the input vector received from input layer into *code (features)*. By increasing or reducing the number of hidden layer neurons with respect to the input layer neurons, an autoencoder is trained. During the training phase, the input vector is mapped to the features. However, autoencoder tries to represent the input vector into features which are useful for data classification process.

A typical auto encoder is shown in Fig. 1. Input layer consists of D neurons and hidden layer consists of C neurons where D is the dimension of the input vector and C is the dimension of the feature vector. The autoencoder consists of two components namely encoder and decoder as shown in Fig. 2. The input of the autoencoder is the input of the encoder and output of the encoder is the output of the hidden layer of the autoencoder. To represent the input vector efficiently, it is converted to *code* (*features*) by the encoder.[11] The input of the decoder is the output of the hidden layer and the output of the decoder is the output of the autoencoder. The decoder tries to reconstruct the original input vector from the code generated by the encoder. The mapping of input to output in encoder can be given by[11]

$$c = f\,(W^T x + b) \tag{1}$$

where $W = [w_1, w_2, \ldots, w_C]$ is the weight matrix where, $wi = [w_{i1}, w_{i2}, \ldots, w_{iC}]$ which represents the weight of the link connecting neuron i to the C neurons, in the hidden layer of the network, $x = [x_1, x_2, \ldots, x_D]^T$ is the input vector in which x_i represents each attribute in the input vector, b= $[b_1, b_2, \ldots, b_C]^T$ is the bias vector where bi represents bias associated with each neuron,

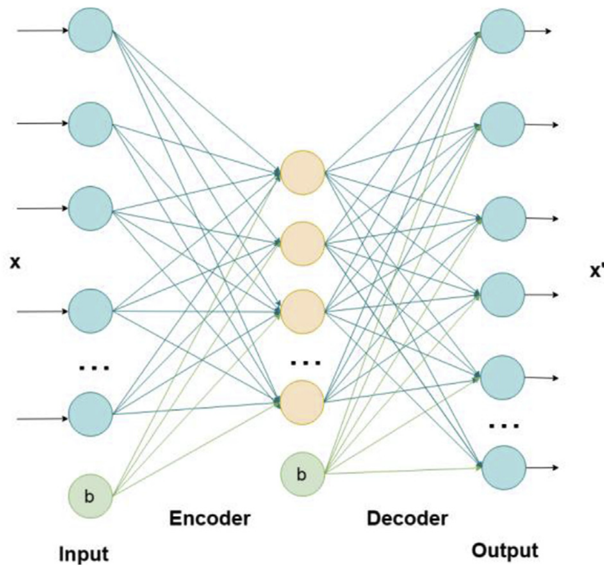

Fig. 1. An autoencoder with hidden layer consists of encoder and decoder.

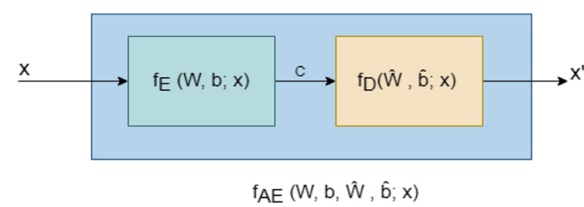

Fig. 2. An autoencoder block consists of encoder and decoder.

$c = [c_1, c_2, \ldots, c_C]^T$ is the code in which c_i represents code generated by neuron i in the hidden layer and f is the activation function of the neuron. Similarly, the mapping of input to output in decoder can be given by,[11]

$$x' = f\,(W'^{\,T} c + b') \tag{2}$$

where $W = [w'_1, w'_2, \ldots, w'_C]$ is the weight matrix which represents weight of the link connecting hidden layer neuron to the neurons in the output layer of the network, $x' = [x'_1, x'_2, \ldots, x'_D]^T$ is the reconstructed input vector in which x' represents each attribute in the input vector, b= $[b_1, b_2, \ldots, b_C]^T$ is the bias vector where b_i represents bias associated with each neuron, $c = [c1, c2, \ldots, cC]^T$ is the code generated by the encoder and f is the activation function of the neuron. Generally, the mapping in encoder can be represented as,

$$c = f_E\,(W, b; x) \tag{3}$$

where f_E is the encoder function and the mapping in decoder can be represented as,

$$x' = f_D\,(W', b'; c) \tag{4}$$

where f_D is the decoder function in the autoencoder. An autoencoder can be formed by serializing the layers of encoder and the decoder in order as shown in Fig. 2.

4. DNN for data classification

In this work, inspired by the interesting features of deep networks, we proposed a DNN based framework using stacked autoencoders for the diabetes data classification which improves all the evaluation metrics of the classification problem. The DNN classifier for diabetes dataset is built using the stacked autoencoders and the softmax layer as discussed in the previous section. The dataset has eight attributes and a class variable which are discussed in detail in the next section. The

Table 1
Parameters used for simulation.

Parameters	Values
L2Weight Regularization	0.01
Sparsity Regularization	4
Sparsity Proportion	0.05
Maximum epoch	1000
Learning rate	0.01
Loss function	Cross entropy
Training Algorithm	Scaled Conjugate Gradient

eight attributes are given as input in the input layer. The DNN built has two layers of autoencoders stacked. The network has two hidden layer each with 20 neurons. The softmax layer is appended with the last hidden layer for classification process. The output layer will give the probabilities of the diabetic and non-diabetic class for a given record. Parameters used for the simulation of the model is given in Table 1.

4.1. Training of layers

Let N input vectors taken for training of the autoencoder be $\{x_{(1)}, x_{(2)}......x_{(N)}\}$. The reconstruction of input is done by training the auto encoder as follows,[11]

$$x' = f_D(W', b'; f_E(W, b; x)) \tag{5}$$

which can be expressed as,

$$x' = f_{AE}(W, b, W', b'; x) \tag{6}$$

where f_{AE} represents the function which maps the input to output in the autoencoder.

The autoencoder is trained by minimizing the appropriate objective function which is given by total error function as,

$$E_{Total} = E_{MSE} + E_{Reg} + E_{sparsity} \tag{7}$$

where E_{MSE}, E_{Reg}, $E_{sparsity}$ represents the mean square error, regularization factor and sparsity factor respectively. The mean square error, E_{MSE} can be calculated by

$$E_{MSE} = \frac{1}{N}\sum_{i=1}^{N} e_i^2 \tag{8}$$

where e_i represents the error, which is the difference between the actual output, $x(i)$ and the observed output, $x'(i)$. The error e_i can be calculated as,

$$e_i = ||x(i) - x'(i)|| \tag{9}$$

Deep networks learns every point in the training data set thus leading to overfitting of the model. This is an issue with deep networks as it results in a poor performance of the model on a new testing data. To overcome this issue, regularization factor, E_{Reg} is considered in the objective function which can be calculated using,

$$E_{Reg} = \frac{\lambda}{2}\left(\sum_{i=1}^{C} ||w_i|| + \sum_{i=1}^{D} ||w_i'||\right) \tag{10}$$

where λ is the term for regularization of the model. Sparsity constraint allows a model to learn the interesting features from the data. Sparsity factor $E_{sparsity}$ can be calculated using,

$$E_{sparsity} = \beta\sum_{i=1}^{C} KL(\rho||\rho_j) \tag{11}$$

where β is the sparsity weight term and $KL(\rho||\rho_j)$ is the Kullback–Leibler divergence[11–13] is given by

$$KL(\rho||\rho_j) = \rho\, log\frac{\rho}{\rho_j} + (1-\rho)\frac{(1-\rho)}{(1-\rho_j)} \tag{12}$$

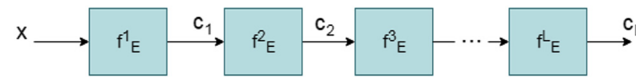

Fig. 3. Stacked autoencoder with L layers.

where sparsity parameter constant is given by ρ and ρ_j represents average activation value of jth neuron which can be calculated using[11]

$$\rho_j = \frac{1}{T}\sum_{i=1}^{T} f^{\,j}(x_{(i)}) \tag{13}$$

where $f^{\,j}(x_{(i)})$ represents the activation function of the jth neuron in the hidden layer of autoencoder.

4.2. The stacked autoencoder

The deep network using autoencoders is constructed by cascading the encoder layers as shown in Fig. 3. Recalling the mapping of autoencoders in Eq. (6) the mapping in stacked autoencoder can be expressed as,

$$f_{SAE} = f_E^{\,1}\circ f_E^{\,2}\circ f_E^{\,3}...\circ f_E^{\,L} \tag{14}$$

where the stacked autoencoder function can be represented as f_{SAE}. In each layer of stacked autoencoder the encoder function is applied. It is important to note that decoder function is not applied in all the layers.

4.3. The softmax layer

Softmax classifier is a multiclass classifier uses logistic regression which classifies the data. Softmax layer uses supervised learning algorithm which uses extended logistic regression to classify multiple classes. Thus logistic regression is the basis for the softmax classifier.[7] In multiclass classifier problem, the softmax classifier estimates the probability of each class with which the data is classified. Hence, the sum of probability of all the classes will be equal to one. The softmax function does the normalization and the exponentiation process for finding the class probabilities. The softmax layer with function f_{SC} is appended with stack autoencoder is as shown in Fig. 4.

4.4. Fine tuning

Once all the layers in the network are trained, the next stage of training the model is called fine tuning. Fine Tuning is the final step in the classification process which is used to improve the performance of the model. To minimize the classification error, the model is fine tuned with supervised learning. This stage is supervised as the target class is known and it is used for training. With the training data set, the entire network is trained similar to the training process of multilayer perceptron. In this process, only the encoder part of autoencoders is considered.

4.5. Training method for DNN classifier

Deep network classifier can be created by cascading stacked auto encoder with the softmax classifier.[11] Stacked autoencoder can have two or more autoencoders layers. Fig. 5(d) shows the DNN classifier with SAE which has two auto encoders. Let the input vector given as input to DNN be $\{x_{(1)}, x_{(2)}......x_{(D)}\}$ and the corresponding output class variables be $\{y_{(1)}, y_{(2)}......y_{(N)}\}$. Given the training input vectors, the aim

Fig. 4. A DNN framework with stacked autoencoder cascaded with softmax layer.

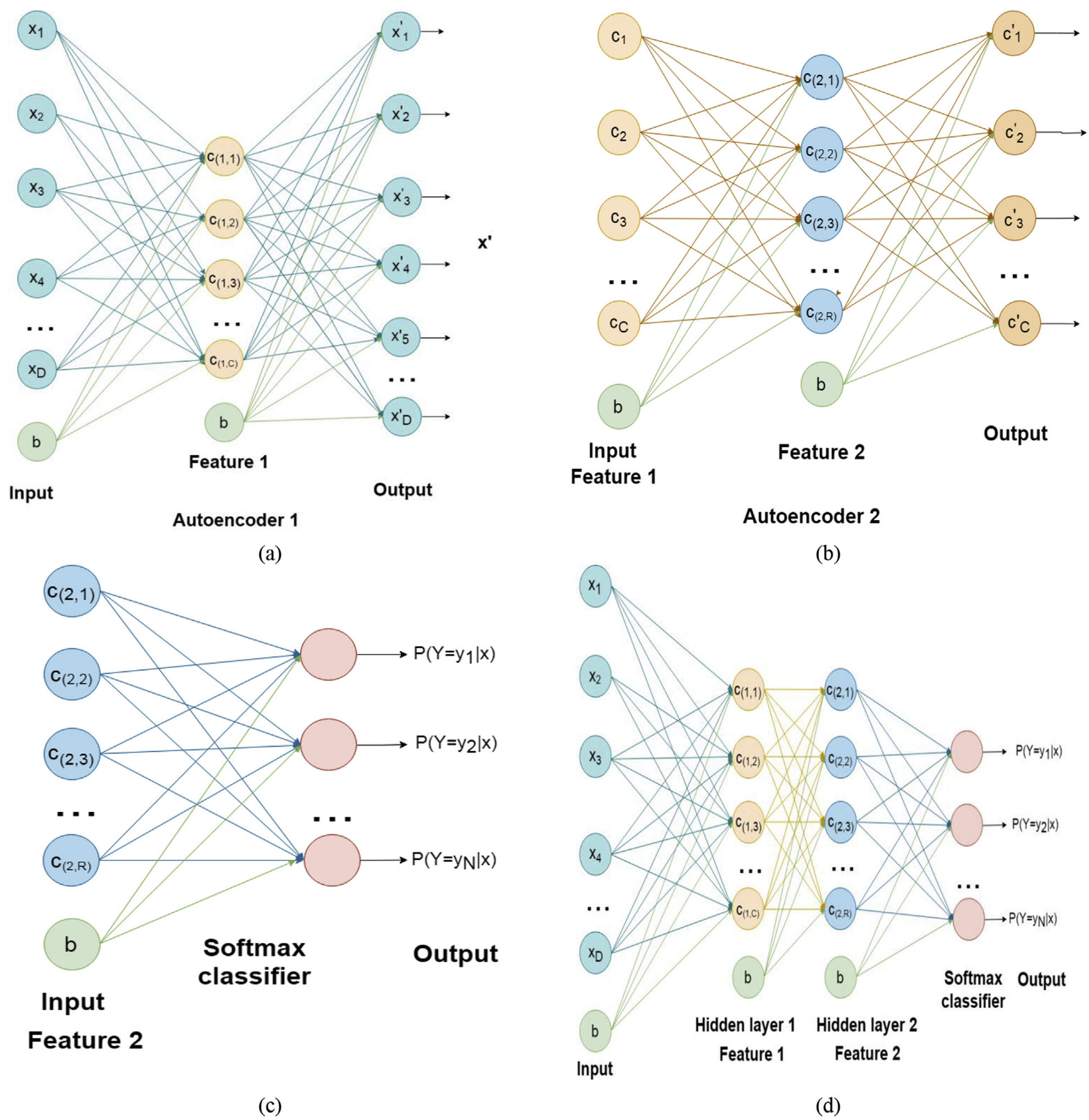

Fig. 5. (a) Network of autoencoder 1 (b) Network of autoencoder 2 (c) Softmax classifier (d) DNN.

of the training method is to tune the parameters of DNN to learn the input vectors and classify the corresponding output value with higher accuracy. The procedure for training the DNN classifier is as follows:

1. Initially, the first autoencoder layer is trained with the original input vector $\{x_{(1)}, x_{(2)}......x_{(D)}\}$ with the same as the target vector. This layer tries to reconstruct the input by extracting the features $\{c_{(1,1)}, c_{(1,2)}......c_{(1,C)}\}$ with the structure of autoencoder as shown in Fig. 5. (a).
2. The second autoencoder layer is trained by taking the output vector of the first autoencoder layer, $\{c_{(1,1)}, c_{(1,2)}......c_{(1,C)}\}$ as input vector and produces output vector $\{c_{(2,1)}, c_{(2,2)}......c_{(2,R)}\}$. The second autoencoder layer tries to reconstruct the input $c_{(1,i)}$; $i = 1,2, ...C$ as shown in Fig. 5. (b).
3. The stacked autoencoder is cascaded with the softmax classifier layer. This layer is trained by taking the second autoencoder layer's output, $c_{(2,i)}$, $i = 1,2, ...R$ as the input vector and the original class variables $\{y_{(1)}, y_{(2)}......y_{(N)}\}$ being the target vector which is obtained from the training data. The softmax classifier is shown in Fig. 5. (c).
4. Finally, to improve the classification performance of the DNN, backpropagation is employed, which is referred to as fine tuning. Fine tuning is done in a supervised manner by retraining the network with the training data.

5. Experimental analysis

The simulations are performed in MATLAB 2017b on an Intel(R) Core(TM) i3 processor with 4 GB RAM and 3.40 GHz CPU on the platform Microsoft Windows 7.

5.1. Data source

For evaluation of the DNN model benchmark data set named Pima Indians Diabetes (PID) data set was selected from the UCI machine

Table 2
Attribute information of PIMA Indians Diabetes dataset.

Attribute number	Attribute name	Minimum	Maximum
1	Number of times pregnant	0	17
2	Plasma glucose concentration a 2 h in an oral glucose tolerance test	0	199
3	Diastolic blood pressure	0	122
4	Triceps skin fold thickness	0	99
5	2-Hour serum insulin	0	846
6	Body mass index	0	67.1
7	Diabetes pedigree function	0.078	2.42
8	Age	21	81

learning repository.[14] The dataset contains 768 records of the patients with eight attributes and one class variable. The attribute information is given in Table 2. The class attribute indicates the diabetic (positive class) and non-diabetic (negative class) patient records. Among 768 patient records, 500 records belongs to negative class and 268 records belongs to positive class.

5.2. Evaluation metrics

Classification Accuracy: Classification accuracy is chosen as an evaluation metric for comparing the results produced by several methods applied in the PID dataset in the literature. Classification accuracy can be given by the equation

$$Accuracy = \frac{\sum_{i=0}^{|N|} evaluate(i)}{|N|} \tag{15}$$

$$evaluate(n) = \begin{cases} 1 \text{ if } classify(n) = cn \\ 0 \quad\quad else \end{cases} \tag{16}$$

where N is the testing dataset to be classified, |N| represents the size of the testing data set to be classified and classify(n) gives the classification result of the data item *n* by the deep network. The classification accuracy can also be represented using the parameters of confusion matrix as shown in Eq. (16).

Accuracy, Specificity, Precision, Recall, F1-score are the four metrics used for the evaluation of a method which is based on the parameters of the confusion matrix. These metrics are given by the equations given below.

$$Accuracy = \frac{TP + TN}{TN + TP + FP + FN}(\%) \tag{17}$$

$$Specificity = \frac{TN}{TN + FP}(\%) \tag{18}$$

$$Precision = \frac{TP}{TP + FP}(\%) \tag{19}$$

$$Recall = \frac{TP}{TP + FN}(\%) \tag{20}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}(\%) \tag{21}$$

5.3. Results and discussion

The deep learning model using stacked autoencoders and softmax layer used for the diabetes data classification is shown in Fig. 6. The input layer consists of eight neurons which are the eight attributes of PID dataset. The network consists of two encoders each with 20 neurons which extract the interesting features from the dataset. The softmax layer uses scaled conjugate gradient algorithm for training the model and classify the data. Fine tuning the parameters of model improves the performance by making the model to learn and extract the features.

Table 3
Comparison of various methods in the literature.

Methods	Accuracy (in %)	Authors
GRNN	80.21	K. Kayaer et al.[8]
SA	75.71 ± 4.41	H. Mohamadi et al.[15]
MLP + BP	75.8 ± 6.2	K. Polat et al.[3]
Smart	76.8	K. Polat et al.[3]
Linear Discr. Analysis	77.5	K. Polat et al.[3]
QDA	59.5	K. Polat et al.[3]
SNBa	75.4	K. Polat et al.[3]
DIPOL	92 77.6	K. Polat et al.[3]
Semi-Naive Bayes	76.0 ± 0.8	K. Polat et al.[3]
OCN2	65.1 ± 1.1	K. Polat et al.[3]
MMLa	75.5 ± 6.3	K. Polat et al.[3]
kNN	71.9	K. Polat et al.[3]
MML	75.5 ± 6.3	K. Polat et al.[3]
IB3	71.7 ± 5.0	K. Polat et al.[3]
LS-SVM	78.21	K. Polat et al.[3]
GDA-LS-SVM	79.16	K. Polat et al.[3]
Logdisc	77.7	H. Kahramanli et al.[16]
BP	75.2	H. Kahramanli et al.[16]
kNN	76.7 ± 4.0	H. Kahramanli et al.[16]
kNN	76.6 ± 3.4	H. Kahramanli et al.[16]
ASR	74.3	H. Kahramanli et al.[16]
SSV DT	73.7 ± 4.7	H. Kahramanli et al.[16]
FDA	76.5	H. Kahramanli et al.[16]
LFC	75.8	H. Kahramanli et al.[16]
Hybrid system	84.2	H. Kahramanli et al.[16]
Kohonen	72.7	H. Kahramanli et al.[16]
MLNN with LM	79.62	H. Temurtas et al.[17]
MLNN with LM	82.57	H. Temurtas et al.[17]
DNN L-BFGS	77.09	Abdullah Caliskan et al.[9]
DNN SAE	**86.26**	Proposed model

In our study, we conducted experiments with and without fine tuning for the evaluation of the performance. Fine tuning of the model is done by backpropagation method. The simulations of our study are executed for ten times and the average of the results is mentioned in the discussion. Simulation results obtained shows that fine tuning the model achieve higher accuracy. Evaluation metrics for the model with fine tuning is shown in Table 4. The classification accuracies of various methods proposed in the literature for diabetes disease diagnosis using Pima Indians diabetes dataset is shown in Table 3. In Ref. 9, authors used Multi-Layer NN with LM algorithm which gives accuracy of 77.08%. The accuracy is comparatively lesser than our proposed model. This lesser classification accuracy can be because of LM algorithm makes the model to memorize the data because of overtraining. In our model, overfitting is avoided using regularization parameter while feature extraction and training. Caliskan et al. proposed a deep learning model which uses L-BFGS algorithm for training the model. The classification accuracy obtained by their model is reported as 77.09% for the partition size 20. However their algorithm performed well for the other dataset, the results for PID dataset is lesser than all other datasets taken for the simulation in their study. Our study uses the similar model of deep learning as their model. Though, because of the training model

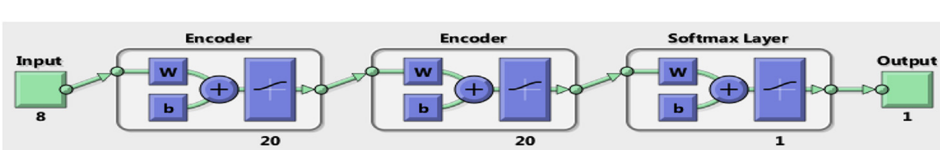

Fig. 6. Simulation of DNN

Table 4
Comparison of evaluation metrics with fine tuning.

Evaluation Metrics	DNN	DNN with fine tuning
Accuracy	83.41	**86.26**
Specificity	78.67	**83.41**
Precision	89.76	**90.66**
Recall	85.63	**87.92**
F1 - score	87.65	**89.27**

and algorithm used, the results vary significantly.

6. Conclusion

In this work, we proposed a DNN framework for diabetes classification of data using stacked autoencoders. The DNN is built using stacked autoencoders cascaded with softmax classifier. Pima Indians diabetes dataset is taken for training process of the model. Our model is compared with several neural network approaches and other state-of-art approaches in the literature. From the results it is evident that our model outperforms other model with an accuracy of 86.26%. Furthermore, our model gives precision value of 90.66% and recall of 87.92% which is quite good for the ideal classification model. Based on the experiments and observations it is concluded that the proposed DNN framework for diabetes classification can be used as powerful tool for the disease diagnosis process. Our model helps in predicting the diabetes of a patient with better accuracy, specificity, precision and recall which are important in the medical world.

References

1. Chen H, Tan C, Lin Z, Wu T. The diagnostics of diabetes mellitus based on ensemble modeling and hair/urine element level analysis. *Comput Biol Med.* 2014;50:70–75https://doi.org/10.1016/j.compbiomed.2014.04.012.
2. Mohamed EI, Linder R, Perriello G, Di ND, Poppl S, De AL. Predicting type 2 diabetes using an electronic nose-based artificial neural network analysis. *Diabetes, nutrition & metabolism.* 2002;15(4):215–221. https://doi.org/10.1051/apido:2004025.
3. Polat K, Gunes S. An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. *Digit Signal Process.* 2007;17(4):702–710https://doi.org/10.1016/j.dsp.2006.09.005.
4. World Health Organization. *Global Statistics Reports on NCDs.* 2016; 2016http://www.who.int/diabetes/global-report/en/, Accessed date: 10 September 2018.
5. Cheruku R, Edla and DR, Kuppili V, Sm-ruleminer. Spider monkey based rule miner using novel fitness function for diabetes classification. *Comput Biol Med.* 2017;81:79–92https://doi.org/10.1016/j.compbiomed.2016.12.009.
6. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science.* 2006;313(5786):504–507. https://doi.org/10.1126/science.1127647.
7. LeCun Y, Bengio Y, Hinton G. Deep learning. *nature.* 2015;521(7553):436. https://doi.org/10.1038/nature14539.
8. Kayaer K, Yıldırım T. Medical diagnosis on pima indian diabetes using general regression neural networks. *Proceedings of the International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP).* 2003; 2003:181–184 Istanbul, Turkey, June 26–29.
9. Caliskan A, Yuksel ME, Badem H, Basturk A. Performance improvement of deep neural network classifiers by a simple training strategy. *Eng Appl Artif Intell.* 2018;67:14–23https://doi.org/10.1016/j.engappai.2017.09.002.
10. Duch W. Dataset used for classification comparision of results. http://fizyka.umk.pl/kis-old/projects/datasets.html; 2018, Accessed date: 10 September 2018.
11. Ng A. Sparse autoencoder. https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf; 2011, Accessed date: 10 September 2018.
12. Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY. On optimization methods for deep learning. *Proceedings of the 28th International Conference on International Conference on Machine Learning. Omnipress, Bellevue.* 2011; 2011:265–272 Washington, USA - June 28 - July 02.
13. Poultney C, Chopra S, Cun YL, et al. Efficient learning of sparse representations with an energy-based model. *Advances in Neural Information Processing Systems,.* 2007; 2007:1137–1144.
14. Uci machine learning repository. available at http://archive.ics.uci.edu/ml/; 2018 [Online].
15. Mohamadi H, Habibi J, Abadeh MS, Saadi H. Data mining with a simulated annealing based fuzzy classification system. *Pattern Recogn.* 2008;41(5):1824–1833https://doi.org/10.1016/j.patcog.2007.11.002.
16. Kahramanli H, Allahverdi N. Design of a hybrid system for the diabetes and heart diseases. *Expert Syst Appl.* 2008;35(1-2):82–89https://doi.org/10.1016/j.eswa.2007.06.004.
17. Temurtas H, Yumusak N, Temurtas F. A comparative study on diabetes disease diagnosis using neural networks. *Expert Syst Appl.* 2009;36(4):8610–8615.
18. Wan S, Liang Y, Zhang Y. Deep convolutional neural networks for diabetic retinopathy detection by image classification. *Comput Electr Eng.* 2018;72:274–282https://doi.org/10.1016/j.compeleceng.2018.07.042.
19. Wu H, Yang S, Huang Z, He J, Wang X. Type 2 diabetes mellitus prediction model based on data mining. *Informatics in Medicine Unlocked.* 2018;10:100–107https://doi.org/10.1016/j.imu.2017.12.006.
20. Mercaldo F, Nardone V, Santone A. Diabetes mellitus affected patients classification and diagnosis through machine learning techniques. *Procedia Computer Science.* 2017;112:2519_2528. https://doi.org/10.1016/j.procs.2017.08.193.
21. Nilashi M, Ibrahim O, Dalvi M, Ahmadi H, Shahmoradi L. Accuracy improvement for diabetes disease classification: a case on a public medical dataset. *Fuzzy Information and Engineering.* 2017;9(3):345–357https://doi.org/10.10.16/j.fiae.2017.09.006.
22. Badem H, Basturk A, Caliskan A, Yuksel ME. A new efficient training strategy for deep neural networks by hybridization of artificial bee colony and limited-memory bfgs optimization algorithms. *Neurocomputing.* 2017;266:506–526https://doi.org/10.1016/j.neucom.2017.05.061.
23. Grozdic DT, Jovicic ST, Subotic M. Whispered speech recognition using deep denoising autoencoder. *Eng Appl Artif Intell.* 2017;59:15–22https://doi.org/10.1016/j.engappai.2016.12.012.
24. Yin J, Zhao W. Fault diagnosis network design for vehicle on-board equipments of high-speed railway: a deep learning approach. *Eng Appl Artif Intell.* 2016;56:250_259https://doi.org/10.1016/j.engappai.2016.10.002.
25. Xu J, Xiang L, Liu Q, et al. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE Trans Med Imag.* 2016;35(1):119–130. https://doi.org/10.1109/TMI.2015.2458702.
26. Luo C, Wu D, Wu D. A deep learning approach for credit scoring using credit default swaps. *Eng Appl Artif Intell.* 2017;65:465–470https://doi.org/10.1016/j.engappai.2016.12.002.
27. Krizhevsk A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems.* 2012; 2012:1097–1105.
28. Caliskan A, Yuksel ME, Badem H, Basturk A. A deep neural network classifier for decoding human brain activity based on magnetoencephalography. *Elektronika ir Elektrotechnika.* 2017;23(2):63–67https://doi.org/10.5755/j01.eie.23.2.18002.
29. Xu J, Luo X, Wang G, Gilmore H, Madabhushi A. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing.* 2016;191:214–223https://doi.org/10.1016/j.neucom.2016.01.034.
30. Deng L, Yu D. Deep learning: methods and applications. *Tech Rep.* 2018:198–239https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/, Accessed date: 10 September 2018.