IET Systems Biology

Research Article

Cancers classification based on deep neural networks and emotional learning approach

ISSN 1751-8849
Received on 14th March 2018
Revised 23rd May 2018
Accepted on 29th May 2018
E-First on 30th July 2018
doi: 10.1049/iet-syb.2018.5002
www.ietdl.org

Noushin Jafarpisheh[1], Mohammad Teshnehlab[1] ✉

[1]Department of Electrical Engineering, K.N. Toosi University of Technology, Tehran, Iran

✉ E-mail: teshnehlab@eetd.kntu.ac.ir

Abstract: In the present era, enormous factors contribute to causing cancer. So cancer classification cannot rely only on doctor's thoughts. As a result, intelligent algorithms concerning doctor's help are inevitable. Therefore, the authors are motivated to suggest a novel algorithm to classify three cancer datasets; colon, ALL-AML, and leukaemia cancers. Their proposed algorithm is based on the deep neural network and emotional learning process. First of all, by applying the principal component analysis, they had a feature reduction. Then, they used deep neural as a feature extraction. Then, they implemented different classifiers; multi-layer perceptron, support vector machine (SVM), decision tree, and Gaussian mixture model. In the end, because in the real world, especially when working on systems biology, unpredictable events, and uncertainties are undeniable, the robustness of their model against uncertainties is important. So they added Gaussian noise to the input features of the first encoder in each dataset, then, they applied the stacked denoising method. Experimental results disclosed that, generally, using emotional learning increased the accuracy. In addition, the highest accuracy was gained by SVM, 91.66, 92.27, and 96.56% for colon, ALL-AML, and leukaemia, respectively. However, GMM led to the lowest accuracy. The best accuracy gained by GMM was 60%.

1 Introduction

According to Uppu *et al.* [1, 2] due to the high number of features, classical machine learning algorithms are not applicable in cases such as cancer classification. In fact, due to the vanishing of the gradient in the training phase, a multi-layer artificial neural network [3, 4] cannot be trained by classical approaches [5]. Bioinformatics, which implements machine learning approaches for solving problems such as cancer classification, is not exempted from this issue either. Therefore, approaches such as the deep neural networks (DNNs) [6, 7] are widely used in their different aspects and, according to Ravi *et al.* [8], in health informatics. As a matter of fact, the deep learning structure is the most suitable one when working on the big data.

One of the most controversial issues in the datasets containing gene expression features is their low number of samples and a high number of features. So feature selection methods should be used to reduce the number of features. Also, these reduced features should be able to express whole features [9].

Selecting features makes it possible to know which of them are more involved in a specific disease [10, 11]. However, as it was claimed in [10], due to a high range of features, the method of selecting them is one of the most challenging issues in bioinformatics. Also, analysing some of these papers published recently reveals that for classifying cancerous and non-cancerous data, at first, a feature selection method should be used, and then, different types of classifiers should be applied.

DNN is new in bioinformatics and is a useful tool for selecting features that are more effective in a matter. In this paper, principal component analysis (PCA) and DNN are applied as feature selection methods. We are motivated to use this strategy because according to Fakoor *et al.* [11], PCA is a method for selecting features that without eliminating the significant features, it reduces the number of features. In addition, deep learning generates abstractions of features. So at first, PCA is implemented to reduce the number of features, and then its outputs are fed to the first encoder of DNN structure. Finally, several classifiers are used to classify cancerous and non-cancerous data.

In [10], it is mentioned that we should use some other pre-processing methods to achieve a better result. Although apparently, those methods increase computational complexity. In this paper, it will be observed that if we implement the right structure, there will be no need for those pre-processing methods.

2 Related work

There have been different papers that applied various feature selection methods. Some of them used datasets similar to these papers. In [12], the support vector machine (SVM) and the mutual information (MI) were applied for gene classification and identifying the informative genes, respectively. Authors in that paper claimed that MI method makes it possible to define a subset of genes. That paper used the colon tumour dataset and allocated 61/62 samples to train and 1/62 to test the performance of the system. This procedure was iterated 62 times to make sure each of the samples was used for evaluating the performance of the system. The best mean accuracy rate achieved by the SVM linear classifier was 67.74%. In [11], the colon dataset is same as ours and for enhancing cancer diagnosis and classification deep learning was used. At first, PCA was applied to reduce and eliminate the irrelevant features. To develop cancer classifiers, the deep learning method was applied. In that paper, accuracy obtained for the colon dataset was 83.33%. In [13], the acute lymphoblastic leukaemia - acute myelogenous leukaemia (ALL-AML) dataset is similar to this paper, and particle swarm optimisation (PSO) and *K*-nearest neighbourhood (*K*-NN) were implemented as gene selection. In that work, *K* has been selected adaptively by running the programme for ten times. Each time, the number of selected genes was considered, and the number of *K* in *K*-NN was calculated. Then, *K* was selected based on the highest test accuracy and the lowest numbers of genes selected. After selecting the genes and determining *K*, the SVM classifier with different kernel functions including linear, radial basis function (RBF) with sigma = 1, polynomial with order three and quadratic were applied. The best result was 97.05888% (33/34) as the test accuracy.

3 Background

To increase the robustness of representation in the DNN, many variations of autoencoders including denoising autoencoder, contractive autoencoder, sparse autoencoder, and stacked autoencoder have been applied that each of which has its pros and cons [8]. Here, we briefly introduce variations which were not implemented in this paper. In addition, emotional leaning and the reason which motivated us to apply this method will be discussed.

3.1 Sparse autoencoder

Sparse autoencoder leads to a better classification. When the numbers of neurons in the hidden layer are more than the numbers of inputs, a constraint should be considered for the learning phase. This constraint can be sparse. In fact, we consider that a neuron is inactive for most of the time. We suppose a neuron is active when its output is one; otherwise, zero. In (1), $a_j^{(2)}$ is the output of the jth neuron in the hidden layer when x is the input. Here, $a_j^{(2)}$ can be one or zero and m shows the total number of training data. So $\hat{\rho}_j$ is the average activation of the jth neuron. In (2), ρ is the sparsity parameter which is usually near to zero. We are willing to achieve (2). For achieving this goal, we consider a penalty that $\hat{\rho}_j$ goes near the ρ. This is shown in (3)

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

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

$$\sum_{j=1}^{n_1}\mathrm{KL}\left(\rho \,||\, \hat{\rho}_j\right) = \sum_{j=1}^{n_1}\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{3}$$

On the basis of the concept of divergence of Kullback–Leibler, this penalty is between two Bernoulli random variables with averages between ρ and $\hat{\rho}_j$. The divergence of Kullback–Leibler is a standard function for measuring the quantity of two different distributions. This penalty function has this feature that divergence of Kullback–Leibler of two distributions is zero when two distributions are equal. Otherwise, $\hat{\rho}_j$ goes near the ρ monotonically increasing. The new cost function follows as:

$$J_{\text{sparse}}(\boldsymbol{W}, b) = J(\boldsymbol{W}, b) + \beta\sum_{j=1}^{n_1}\mathrm{KL}\left(\rho \,||\, \hat{\rho}_j\right) \tag{4}$$

where β is the sparsity learning rate of the penalty function J ($\boldsymbol{W}$, b) and has been shown in the equation below:

$$J(\boldsymbol{W}, b) = \frac{1}{2}e^2 \tag{5}$$

where e is the reconstruction error in the output layer of the DNN [14].

3.2 Contractive autoencoder

Contractive autoencoder [15] is used for boosting robustness of representation among small changes in training samples [8]. The cost function is expressed in the equation below:

$$J_{\text{CAE}}(\boldsymbol{W}, b) = J(\boldsymbol{W}, b) + \lambda\sum_{j=1}^{n_1}\|J_{\text{f}}(x)\|_{\text{F}}^2 \tag{6}$$

where $\|J_{\text{f}}(x)\|_{\text{F}}^2$ is Frobenius norm of Jacobean and λ is used for controlling the strength of regularisation. The sum square of the partial derivative of extracted features is shown in the equation below:

$$\|J_{\text{f}}(x)\|_{\text{F}}^2 = \sum_{i,j}\left(\frac{\partial h_j(x)}{\partial x_i}\right)^2 \tag{7}$$

where h_j is the output of the jth neuron and x is the input that is mapped to the hidden layer by a non-linear function [15].

3.3 Emotional learning

People who have a high emotional intelligence have a better communication with others and their environment. They try to obtain emotional experience from other people. As a result, besides perception of their emotion, they can easily understand others' emotion. So they will have a better interaction with others. In addition, these people are more successful in controlling their emotions. Having high emotional intelligence indicates individual experiences against negative and positive events. These experiences are inherent of a human being that will appear by learning. These people try to apply their experience and others' experience for enhancing their decision.

Since artificial neural networks have been inspired by real neural nets in our body, we applied the mentioned concept in our work. The cost function has been considered based on current and former errors of the network. This strategy has a more appropriate effect on the learning process of the network. In fact, using former error which contains previous information of the network leads to have a better learning process, increasing the speed of the convergence to the optimal point, and achieving a higher accuracy.

4 Methodology

4.1 Contributions

This paper proposes a structure for classifying three cancer datasets to increase the accuracy of results. These three cancer datasets had not experienced the proposed algorithm before.

Furthermore, because system boogies are at risk of uncertainties and unpredictable events, we added Gaussian noise to the input features of the first autoencoder to increase the robustness of our model. Then, for the first time, by stacked denoising method, we tried to omit the effect of noise added to the datasets.

In addition, in order to increase the accuracy gained by each classifier, we proposed emotional learning. In this method, previous errors or former information of the network is accomplished in the training phase. This leads to achieving higher accuracy for each classifier.

4.2 Cancer datasets

Three cancer datasets including colon, ALL-AML, and leukaemia cancers were retrieved from [16, 17]. All three datasets include gene expression. We assessed our proposed approach and methodology through these three datasets. We selected these three cancer datasets because the range of the features that each dataset contains is markedly different. The features of the colon cancer dataset have a wide range of variations. The value of whole features is between 5.8163 and 20,903. This dataset is considerably non-linear. Samples in ALL-AML and leukaemia have a more limited number of features compared with the colon cancer dataset. The value of features is approximately between −8 and 8.5 in ALL-AML dataset. Unlike the colon and ALL-AML cancer datasets, features in leukaemia dataset are discrete, and the value of features is just −2 or 0.

It is worth mentioning that when the dataset is non-linear, we should increase the number of features in the first autoencoder. Doing this increases the diversity of the feasible space for searching the stable point. This increment is achievable via determining the weight matrixes in the autoencoder layer.

Table 1 presents the number of samples and their characteristics in each dataset. It can be clearly seen that these three datasets contain a high dimension of features. On this occasion, if we use the conventional neural network, the number of hidden layers will be increased. As a result, weight matrixes in the first hidden layers, near the input layer, will not be updated. It is the reason why a classical neural network cannot be applied. This implies the necessity of feature reduction and extraction which is done by PCA and DNN in this paper.

Table 1 Number of whole samples and number of samples in each class for different cancer datasets

Dataset	Number of samples	Number of features	Number of samples in each dataset
colon cancer [16]	62	2000	40 (tumour) and 22 (normal)
ALL-AML [17]	72	7129	47 (ALL) and 25 (AML)
leukaemia [17]	72	7070	47 (−1)25 (1)

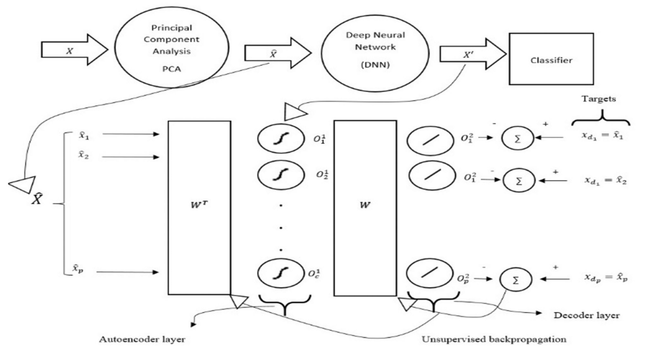

Fig. 1 *General overview of autoencoder and decoder structures in DNN*

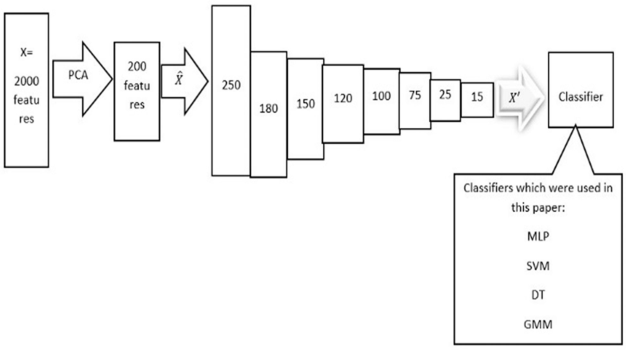

Fig. 2 *Number of features selected in each autoencoder. This structure is fixed for all three cancer datasets that are used here. Only the number of whole features (x) will be different for each dataset*

4.3 Experimental setup

According to Khodayar and Teshnehlab [5], the autoencoder is known as a non-linear method that is suitable for highly non-linear datasets. Fig. 1 shows the strategy that is considered in this paper. It illustrates that PCA reduces the number of features from $\boldsymbol{X}$ to $\hat{\boldsymbol{X}}$. Then, DNN extracts features from $\hat{\boldsymbol{X}}$ features and generates X' features. After, different classifiers (MLP, SVM, DT [18], and GMM) are applied in order to classify $\boldsymbol{X}'$ features. In [10], it was claimed that in some datasets there are some problems such as data imbalance or datum shift; therefore, pre-processing steps are required. However, in this paper, regardless of how a dataset is balanced or imbalanced the proposed method can be applied, and there is no need for those pre-processing steps. Consequently, computational complexity will be reduced. However, it is subject to choosing the appropriate structure for DNN. Here, we used a similar structure (Fig. 1) for all the datasets.

Fig. 2 depicts the dimension of weight matrix ($\boldsymbol{W}$) or the number of neurons in the hidden layer (250, 180, 150, 120, 100, 75, 25, and 15). Also, these values demonstrate numbers of extracted features from each autoencoder layer. Since ALL-AML and leukaemia datasets are not as non-linear as the colon cancer dataset, there is no need to increase the number of features in the first encoder layer. It is the reason why we just increase the number of features in the first encoder layer in the colon cancer dataset (Fig. 2). To choose the number of autoencoders and the number of extracted features, each time, the number of neurons in the autoencoder layer is considered randomly. Then, the error is calculated. After that, we choose the numbers of neurons that lead to the lowest error. Finally, different classifiers are applied.

To train the autoencoders, (8) should be considered in this methodology. Equations (9)–(12) show the forward neural network equations. In addition, (10) and (12) show that tangent sigmoid and linear functions are considered the activation functions in the autoencoder and decoder layers, respectively. Then, an optimisation method should be considered for updating the weight matrix that connects neurons in the hidden layer to the output layer $\boldsymbol{W}^2$. Here, the gradient descent has been applied. To do this, as it is shown in Fig. 1, the output of each encoder should be compared with its target. The targets are same as the inputs of the encoder layer (unsupervised learning). Then, the error will be calculated (13). *J* index in (13) refers to the *j*th error. Then, sum square error is defined as the cost function (14a). Additionally, in order to increase the performance of the training phase, we applied emotional learning. The cost function for this learning algorithm is defined by (14b), where k_1 and k_2 are coefficients that determine the effect of current and former errors. Here, we considered 1 and 0.5 for k_1 and k_2, respectively. Next, the calculated error will be back propagated to update $\boldsymbol{W}^2$ through a chain rule [(15a) and (15b)]. After updating $\boldsymbol{W}^2$, we should transpose this matrix, and substitute it with $\boldsymbol{W}^1$ which connects input layer to neurons in the hidden layer.

The first step is the feed-forward algorithm, which is illustrated in the equations below:

$$\boldsymbol{W}^2 = \boldsymbol{W}; \quad \boldsymbol{W}^1 = \boldsymbol{W}^{\rm T} \tag{8}$$

$$\mathbf{net}^1(k) = \boldsymbol{W}^{\rm T}(k)\hat{\boldsymbol{X}}(k) \tag{9}$$

The outputs of autoencoder layers are calculated through the equation below:

$$\boldsymbol{O}^1(k) = \text{tansig}\big(\mathbf{net}^1(k)\big) \tag{10}$$

$$\mathbf{net}^2(k) = \boldsymbol{W}(k)\boldsymbol{O}^1(k) \tag{11}$$

The outputs of decoder layers are determined by a linear function in the equation below:

$$\boldsymbol{O}^2(k) = \mathbf{net}^2(k) \tag{12}$$

The second step is the learning phase presented by (13)–(15a) and (15b)

$$\boldsymbol{e}_j(k) = X_{d_j}(k) - \boldsymbol{O}^2(k) = \boldsymbol{X}_{d_j}(k) - (\boldsymbol{W}(k)\,\boldsymbol{O}^1(k)) \tag{13}$$

Moreover, (14a) and (14b) are considered as the cost functions

$$E_1(k) = \frac{1}{2}\sum_{j=1}^{p} \boldsymbol{e}_j^2(k) \tag{14a}$$

$$\begin{aligned} E_2(k) &= \frac{1}{2}\sum_{j=1}^{p} \boldsymbol{r}_j^2(k) = \frac{1}{2}\sum_{j=1}^{p}\big(k_1\boldsymbol{e}_j(k) + k_2\dot{\boldsymbol{e}}_j(k)\big)^2 \\ &= \frac{1}{2}\sum_{j=1}^{p}\big((k_1 + k_2)\boldsymbol{e}_j(k) - k_2\boldsymbol{e}_j(k-1)\big)^2 \end{aligned} \tag{14b}$$

Classical back propagation (see (15a)) .

Emotional back propagation

$$\begin{aligned} \Delta\boldsymbol{W}(k) &= -\eta\frac{\partial E_2(k)}{\partial \boldsymbol{W}(k)} = -\frac{\partial E_2(k)}{\partial \boldsymbol{r}(k)}\frac{\partial \boldsymbol{r}(k)}{\partial \boldsymbol{e}(k)}\frac{\partial \boldsymbol{e}(k)}{\partial \boldsymbol{O}^2(k)}\frac{\partial \boldsymbol{O}^2(k)}{\partial \boldsymbol{W}(k)} \\ &= \eta\,(k_1 + k_2)\boldsymbol{r}(k)\boldsymbol{O}^1(k) \end{aligned} \tag{15b}$$

where η is the learning rate that should be $0 < \eta \le 1$ for keeping the learning process stable.

$$\begin{aligned}\Delta \boldsymbol{W}(k) &= -\eta\frac{\partial E_1(k)}{\partial \boldsymbol{W}(k)} = -\frac{\partial E_1(k)}{\partial \boldsymbol{e}(k)}\frac{\partial \boldsymbol{e}(k)}{\partial \boldsymbol{O}^2(k)}\frac{\partial \boldsymbol{O}^2(k)}{\partial \boldsymbol{W}(k)} = -\eta(\boldsymbol{e}(k))(-1)\boldsymbol{O}^1(k)\\ &= \eta\boldsymbol{e}(k)\boldsymbol{O}^1(k)\end{aligned} \tag{15a}$$

Table 2 Number of training and testing samples in each dataset

Dataset	Number of samples for training	Number of samples for testing
colon cancer	44	18
ALL-AML	55	22
leukaemia	55	22

Table 3 Results of classifying colon cancer dataset with 18 test samples

Classifier	Mean of results (*E*1)	*E*2	Variance of results (*E*1)	*E*2	Best result (*E*1)	*E*2
MLP	64.45%(12)	67.78%(12)	3.3778	2.1778	15	15
SVM	89.44%(16)	91.66%(16)	0.77	0.72	17	18
DT	70.56%(13)	73.34%(13)	5.79	3.73	16	16
GMM	47.78%(9)	54.44%(10)	17.6	16.18	13	17

Table 4 Results of classifying ALL-AML cancer dataset with 22 test samples

Classifier	Mean of results (*E*1)	*E*2	Variance of results (*E*1)	*E*2	Best result (*E*1)	*E*2
MLP	64.54%(14)	65.45%(14)	7.0667	3.8222	18	17
SVM	88.63%(19)	92.27%(20)	2.9444	0.9	21	22
DT	72.27%(16)	74.09%(16)	6.5444	6.0111	19	21
GMM	26.82%(6)	29.09%(6)	15.8778	12.0444	13	11

Table 5 Results of classifying leukaemia cancer dataset with 22 test samples

Classifier	Mean of results (*E*1)	*E*2	Variance of results (*E*1)	*E*2	Best result (*E*1)	*E*2
MLP	86.36%(19)	87.22%(19)	2.011	1.12	21	21
SVM	95.45%(21)	96.56%(21)	1.8222	0.84	22	22
DT	81.82%(18)	83.18%(18)	2.011	5.57	21	22
GMM	50.09%(13)	51.82%(11)	12.2667	19.38	16	18

After updating weight matrixes in each autoencoder, extracted features will be considered as the input features of each classifier. In each classifier, we divide the samples into two groups: the train samples and the test samples (unseen data). The number of train samples and test samples for each cancer dataset are in Table 2. Here, we use a supervised learning algorithm to classify features because the actual classification for each dataset is available.

5 Simulation results and discussion

We obtained the experimental results by running the simulation programmes ten times with ten different initial values of weight matrixes. Then, the average of the results was reported in Tables 3–5. Since the number of test samples assigned to each dataset is not the same, the titles of these tables were reported regarding the number of test samples. In all tables, $E1$ and $E2$ refer to (14a) and (14b). At first glance, it can be clearly seen that SVM [19] has the highest accuracy in all of the three cancer datasets. In the second column, it can be observed that the GMM classifier gives the highest variance among other classifiers. The SVM classifier, on the other hand, gives the lowest variance. Reporting the variance of the results should be taken into account, for it somehow presents the reliability of results obtained in each simulation. For example, if the variation in results is ignorable, the accuracy obtained in one simulation can be sufficient for our decisions. Moreover, there will be no need for running the programme several times. In fact, we know that in another programme running, the accuracy will not be significantly different. Nevertheless, the previous papers that worked on these datasets did not report the variation of results. Some of those papers, also, did not report the number of times that programmes were simulated. Here, in contrast, we report the variance of results in a ten-time simulation. Hereby, we can claim that if we run the programme several times (as in this paper); we can reduce the effect of some parameters that are selected randomly such as the initial value of weight matrixes. In the last column of these tables, the highest number of true answers is illustrated. It can be observed that the SVM classifier can classify all test samples without any errors at least in one programme running. Furthermore, MLP can classify the leukaemia cancer dataset with the highest accuracy and colon cancer with the lowest accuracy. In addition, DT and GMM classify leukaemia better than other datasets. Generally, we cannot discuss the performance of GMM because of the high variability of results obtained by this classifier.

Overall, according to the experimental results, the more datasets are non-linear and include a wider range of features, the better to apply SVM. In addition, GMM classifier can lead to a good performance if the distribution of data is normal. However, our datasets do not have normal distributions. Therefore, implementation of this classifier cannot have a good performance, and it is not suitable. In all of the three cancer datasets, the SVM classifier leads to a higher accuracy than MLP, because MLP is based on tests or experience. However, SVM minimises structure risk [20]. This classifier determines boundaries that have the longest distance to samples in each category. Therefore, what it takes into consideration is minimising of the risks in the structure. In addition, SVM classifies samples faster than MLP.

Since the existence of unpredictable events such as noise is undeniable, we should increase the robustness of our method against those events. Therefore, in the last section of this paper, we consider the stacked denoising method. Moreover, we compare the obtained results when the input features are pure and when the Gaussian noise is added to them. Tables 6–8 disclose the results and accuracy achieved by applying different classifiers using the stacked denoising method. In general, the stacked denoising method is more successful when SVM classifier is applied in ALL-

Table 6 Results of classifying colon cancer dataset by stacked denoising method with 18 test samples

Classifier	Mean of results (*E*1)	*E*2	Variance of results (*E*1)	*E*2	Best result (*E*1)	*E*2
MLP	60.56%(11)	70%(13)	2.9889	4.2667	13	16
SVM	86.67%(16)	90%(16)	3.16	2.4	18	18
DT	72.22%(13)	76.67%(14)	6.0444	2.84	16	16
GMM	44.44%(8)	53.89%(10)	17.3889	25.34	15	17

Table 7 Results of classifying ALL-AML cancer dataset by stacked denoising method with 22 test samples

Classifier	Mean of results (*E*1)	*E*2	Variance of results (*E*1)	*E*2	Best result (*E*1)	*E*2
MLP	60.00%(13)	64.55%(14)	0.6222	3.0667	14	17
SVM	88.18%(19)	92.27%(20)	0.9333	1.1222	21	22
DT	65.91%(14)	74.09%(16)	1.8333	7.7889	20	20
GMM	35%(8)	35.05%(8)	16.2333	8.2667	15	12

Table 8 Results of classifying leukaemia cancer dataset by stacked denoising method with 22 test samples

Classifier	Mean of results (*E*1)	*E*2	Variance of results (*E*1)	*E*2	Best result (*E*1)	*E*2
MLP	81.82%(18)	86%(19)	3.4333	3.21	21	22
SVM	94.1%(21)	94.54%(21)	1.12	0.62	22	22
DT	77.27%(17)	83.63%(18)	3.5111	2.27	21	21
GMM	54.45%(10)	60%(13)	10.9444	8.84	15	20

AML and leukaemia cancer datasets. It means the mean accuracies obtained by this classifier are not different a lot before and after adding noise and applying stacked denoising method. For the colon cancer dataset, the variation of results obtained by MLP classifier, before and after adding noise and using the stacked denoising method, decreases. Also, for ALL-AML cancer dataset, MLP, SVM, and GMM lead to a lower variance. In ALL-AML and leukaemia cancer datasets, applying the SVM classifier leads to a lower variance of results when noise is added and the stacked denoising method is applied.

In all classifiers and cancer datasets when we consider emotional learning, higher accuracy of results is achievable. It is due to the fact that emotional learning applies former information for training the network.

6 Conclusion

Dealing with big data is challenging, especially while we are working on low numbers of samples and a high number of features. Cancers datasets including gene expression profiles are the examples. In these cases, classical machine learning algorithms are not applicable due to the existence of any problems such as vanishing of the gradient descent. DNN makes it possible to overcome these problems.

In this paper, we proposed a novel approach for cancer classification via PCA and DNN as the feature reduction and feature extraction. Then, we applied four types of classifiers –MLP, SVM, DT, and GMM – to categorise the extracted features. In the next stage, we used three cancer datasets; colon, ALL-AML, and leukaemia to assess our method through them. The proposed method had a fixed structure, which means that the number of extracted features by DNN was the same in all cancer datasets. We selected these three datasets because the features they carry have different ranges of variations. The colon cancer dataset includes features with the highest number of variations. However, the features in ALL-AML cancer dataset contain a limited value of variations. The features in the last cancer dataset – leukaemia – consist of only −2 or 0. Then, in order to increase the robustness of our method against uncertainties, we added Gaussian noise to the input features of the first autoencoder and used the stacked denoising method, which was not used in the previous papers with the same datasets used here. Moreover, in parallel with the steps mentioned above, we proposed emotional learning. In this learning algorithm, former and current errors are accomplished to form the cost function. Therefore, in the training phase, the network uses the past and current information to be trained. As a result, the network will learn the pattern better, the speed of convergence to the optimal point will be increased, and higher accuracy will be obtained. Overall, among all classifiers, SVM led to the highest accuracy, and GMM resulted in the lowest one.

For future work, an interpretable state-of-the-art algorithm for the DNN can be proposed. The DNN algorithm used here is derived from neural network, which is a black-box structure. The novel approach can be such as a grey box rather than a black one. This will be achievable by the combination of deep learning structures and interpretable machine learning algorithms.

7 References

[1] Uppu, S., Krishna, A., Gopalan, R.P.: ‘Rule-based analysis for detecting epistasis using associative classification mining’, *Netw. Model. Anal. Health Inf. Bioinf.*, 2015, **4**, (1), pp. 1–19

[2] Uppu, S., Krishna, A., Gopalan, R.P.: ‘A deep learning approach to detect SNP interactions’, *J. Softw.*, 2016, **11**, (10), pp. 965–975

[3] Grolinger, K., Heureux, A.L., Capretz, M.A., *et al.*: ‘Energy forecasting for event venues: big data and prediction accuracy’, *Energy Build.*, 2016, **112**, pp. 222–233

[4] Cruz, J.A., Wishart, D.S.: ‘Applications of machine learning in cancer prediction and prognosis’, *Cancer Inf.*, 2006, **2**, pp. 59–77

[5] Khodayar, M., Teshnehlab, M: ‘Robust deep neural network for wind speed prediction’. 2015 Fourth Iranian Joint Congress Fuzzy and Intelligent Systems (CFIS), Iran, 2015, pp. 1–5

[6] Uppu, S., Krishna, A., Gopalan, R.P.: ‘Towards deep learning in genome-wide association interaction studies’, PACIS, Taiwan, 2016, p. 20

[7] L'heureux, A., Grolinger, K., Elyamany, H.F., *et al.*: ‘Machine learning with big data: challenges and approaches’, *IEEE Access*, 2017, **5**, pp. 7776–7797

[8] Ravi, D., Wong, C., Deligianni, F., *et al.*: ‘Deep learning for health informatics’, *IEEE J. Biomed. Health Inf.*, 2017, **21**, (1), pp. 4–21

[9] Mundra, P., Rajapakse, J.: ‘SVM-RFE with MRMR filter for gene selection’, *IEEE Trans. Nanobiosci.*, 2010, **9**, (1), pp. 31–37

[10] Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., *et al.*: ‘A review of microarray datasets and applied feature selection methods’, *Inf. Sci.*, 2014, **282**, pp. 111–135

[11] Fakoor, R., Ladhak, F., Nazi, A., *et al.*: ‘Using deep learning to enhance cancer diagnosis and classification’, *Conf. Mach. Learn.*, 2013, **28**, pp. 1–7

[12] Vanitha, C.D.A., Devaraj, D., Venkatesulu, M.: ‘Gene expression data classification using support vector machine and mutual information-based gene selection’, *Procedia Comput. Sci.*, 2015, **47**, pp. 13–21

[13] Kar, S., Sharma, K.D., Maitra, M.: ‘Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive *K*-nearest neighborhood technique’, *Expert Syst. Appl.*, 2015, **42**, (1), pp. 612–627

[14] Ng, A.: ‘Sparse autoencoder’, *CS294A Lect. Notes*, 2011, **72**, pp. 1–19

[15] Rifai, S., Vincent, P., Muller, X., *et al.*: ‘Contractive auto-encoders explicit invariance during feature extraction’. Proc. Int. Conf. Machine Learning, Bellevue, WA, USA, June 2011, pp. 833–840

[16] ‘SIPTA homepage’. Available at http://leo.ugr.es/elvira/DBCRepository/index.html, accessed February 2017

[17] ‘Datasets|feature selection @ ASU’. Available at http://featureselection.asu.edu/datasets.php, accessed February 2017

[18] Lakshmipathy, T., Ranganathan, G.: ‘Significance of data mining techniques in disease diagnosis and biological research – a survey’, *IIOAB J.*, 2016, **7**, (1), pp. 284–292

[19] Kourou, K., Exarchos, T.P., Exarchos, K.P*., et al.*: ‘Machine learning applications in cancer prognosis and prediction’, *Comput. Struct. Biotechnol. J.*, 2015, **13**, pp. 8–17

[20] Zanaty, E.: ‘Support vector machines (SVMs) versus multilayer perception (MLP) in data classification’, *Egypt. Inf. J.*, 2012, **13**, (3), pp. 177–183