www.impactjournals.com/oncotarget/ Oncotarget, 2017, Vol. 8, (No. 65), pp: 109646-109660

Research Paper

Tumor gene expression data classification via sample expansion-based deep learning

Jian Liu[1,*], Xuesong Wang[1,*], Yuhu Cheng[1] and Lin Zhang[1]

[1]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China

[*]Joint First Authors

Correspondence to: *Xuesong Wang,* ***email:*** *wangxuesongcumt@163.com*
Yuhu Cheng, ***email:*** *chengyuhu@163.com*

Keywords: *gene expression data; classification; sample expansion; deep learning; 1-dimensional convolutional neural network*

Received: *August 31, 2017* ***Accepted:*** *October 29, 2017* ***Published:*** *November 30, 2017*

ABSTRACT

Since tumor is seriously harmful to human health, effective diagnosis measures are in urgent need for tumor therapy. Early detection of tumor is particularly important for better treatment of patients. A notable issue is how to effectively discriminate tumor samples from normal ones. Many classification methods, such as Support Vector Machines (SVMs), have been proposed for tumor classification. Recently, deep learning has achieved satisfactory performance in the classification task of many areas. However, the application of deep learning is rare in tumor classification due to insufficient training samples of gene expression data. In this paper, a Sample Expansion method is proposed to address the problem. Inspired by the idea of Denoising Autoencoder (DAE), a large number of samples are obtained by randomly cleaning partially corrupted input many times. The expanded samples can not only maintain the merits of corrupted data in DAE but also deal with the problem of insufficient training samples of gene expression data to a certain extent. Since Stacked Autoencoder (SAE) and Convolutional Neural Network (CNN) models show excellent performance in classification task, the applicability of SAE and 1-dimensional CNN (1DCNN) on gene expression data is analyzed. Finally, two deep learning models, Sample Expansion-Based SAE (SESAE) and Sample Expansion-Based 1DCNN (SE1DCNN), are designed to carry out tumor gene expression data classification by using the expanded samples. Experimental studies indicate that SESAE and SE1DCNN are very effective in tumor classification.

INTRODUCTION

Tumors, which seriously endanger human health, are part of the major malignant diseases in the world. Early detection of the tumor is under a very important meaning for the better treatment of patients. The emergence and development of DNA microarray has promoted the research of tumor at the molecular level [1-3]. By mining the useful knowledge and information from the massive tumor gene expression data, we can have a comprehensive understanding of the nature of the tumor at the genetic level which plays an important role in promoting the clinical diagnosis and treatment of tumors as well as developing new drugs [4, 5]. Generally, gene expression data can be obtained from multiple tissue samples, including diseased samples and normal samples. By comparing the gene expression levels in diseased samples and normal samples, researchers can get a better insight into the disease pathology of the tumor [6, 7]. An urgent problem need to be addressed is how to effectively discriminate tumor samples from normal ones. To deal with this, many classification methods, such as Support Vector Machines (SVMs) [8] and Neural Networks [9]-[10], have been proposed for tumor gene expression data classification.

Among all classification methods, deep learning models show very good performance and draw more and more attention. Deep learning models have many advantages over conventional methods. On the one hand, deep learning models intrinsically learn a high level representation of the data so that avoiding laborious work [11]. On the other hand, deep structure has exponentially stronger expressive power than conventional shallow structure. Deep learning has achieved promising performance in many fields, such as computer vision, speech recognition, and natural language processing. According to [12], a review of deep learning in bioinformatics, deep learning models have also been widely used in the area of bioinformatics, including biomedical signal processing, biomedical imaging and omics. However, the application of deep learning is rare in tumor classification. The only available literature was written by Fakoor et al. [13]. Therefore, we attempt to use deep learning models to classify the tumor gene expression data.

Among a variety of deep learning models, Stacked Autoencoder (SAE) [14] is a widely used and effective method. SAE is a multi-layer neural network that reproduces the input signal as much as possible. It has been widely used in many areas, such as medical image processing [15], object recognition [16], and video classification [17]. In [13], SAE was successfully applied to the gene expression data for classifying the tumor samples. The classification process of SAE for a specific tumor type is given as follows: Firstly, the dimensionality of the feature space is reduced by using principle component analysis (PCA) due to the characteristics of the small sample problem in high-dimensional gene expression data. Secondly, other tumor gene expression data from the same platform are used as unlabeled data for feature learning since the number of samples for the specific tumor is really small. Thirdly, the weights of the features learned in the second step are tuned using the specific labeled data. Finally, the tumor gene expression data is classified. The drawback of literature [13] is that the gap between specific tumor data and other tumor gene expression data from the same platform is not considered, thus may have a negative effect on tumor classification. In this paper, we use SAE to achieve tumor classification in a different way with [13].

Convolutional Neural Network (CNN) [18] plays a dominant role in the community of deep learning models. CNN exploits spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. CNN has been demonstrated to provide better performance than other conventional methods on various vision tasks, such as face recognition [19], object detection [20], and image classification [21, 22]. In addition to vision tasks, CNN has also been applied to speech recognition [23, 24], natural language processing [25] and other fields. However, there are rare literatures on the technique with CNN for tumor classification. In this paper, CNN is considered to be applied to classify the tumor gene expression data.

CNN is the most widely used method in the field of image processing. Generally, a 2-dimensional image sample is taken as the input of CNN to implement convolution operation. However, each sample is a 1-dimensional array in tumor gene expression data which makes traditional CNN models not applicable for tumor classification. Fortunately, 1-dimensional CNN (1DCNN), a special CNN model, is proposed, and it requires the input is a 1-dimensional vector. 1DCNN has been used to analyze 1-dimensional sample in many areas. For example, Hu et al. have successfully utilized 1DCNN to process the spectral channels [26]. But the applicability of 1DCNN in the tumor gene expression data requires further study. Here, we introduced 1DCNN into tumor classification.

A large number of labeled data are usually required for training the deep learning models, including SAE and 1DCNN. However, the number of labeled samples of tumor gene expression data is quite small. For instance, there are 60 labeled samples in colon data [27] and only 20 labeled samples in breast cancer data [28]. In this paper, we propose a novel Sample Expansion (SE) method to address the problem of insufficient labeled samples. Inspired by Denoising Autoencoder (DAE) [29], a large number of labeled samples are obtained by randomly cleaning partially corrupted input many times. These labeled samples are taken as the expanded samples. Then we merged the expanded samples and untreated samples into a matrix as the training samples. This method can deal with the problem of insufficient training samples of tumor gene expression data to a certain extent. Furthermore, in order to benefit from both deep learning and SE, we suggest two deep learning-based methods, Sample Expansion-Based SAE (SESAE) and Sample Expansion-Based 1DCNN (SE1DCNN), for tumor classification. The tumor classification process is given as follows. Firstly, due to the high dimensionality of tumor gene expression data, we reduce the dimensionality of gene expression data. For each gene expression data, each feature represents one gene and has its natural meaning. Therefore, gene selection is more convincing than feature extraction in processing tumor gene expression data. Here, Infinite Feature Selection (Inf-FS) [30] is used as the dimensionality reduction strategy to select genes. Secondly, SE is implemented to expand the number of labeled samples. Finally, two deep learning models, SESAE and SE1DCNN, are utilized to achieve tumor classification based on the expanded samples. Experimental results demonstrate that SESAE and SE1DCNN are very effective in tumor gene expression data classification.

The main contributions of our work are summarized as follows. Firstly, for the first time, 1DCNN, an

Table 1: Summery of tumor gene expression datasets

Dataset	Data Labels	Number of	
		Genes	Samples
Breast cancer	1=non-IBC, 2=IBC	30006	20
Leukemia	1=MP, 2=HDMTX, 3=HDMTX+MP, 4=LDMTX+MP	12600	60
Colon cancer	1=cancer, 2= normal	2000	62

excellent deep learning model, is successfully applied to the tumor classification task. Secondly, a novel sample expansion method is proposed to deal with the problem of insufficient labeled samples when using deep learning models to implement tumor classification.

The remainder of the paper is structured as follows. In Section 2, SE is proposed and how to select cancer characteristic genes by SESAE or SE1DCNN is explained. Experimental results and discussion on tumor gene expression datasets are presented in Section 3. In Section 4, the conclusions are given.

RESULTS AND DISCUSSION

This section shows the experimental results. In this paper, microarray data was used to perform our experiment. We performed our method on three publicly available gene expression datasets, i.e., breast cancer [28], leukemia [31] and colon cancer [27]. We determined the parameters of SESAE and SE1DCNN. To demonstrate the effectiveness of SESAE and SE1DCNN for tumor classification, 1DCNN [26], traditional SAE, SAE in [13], SAE with fine tuning in [13], and Softmax/SVM were employed for comparison. In this paper, the programs are implemented by using Python language and Theano library [32] on a PC equipped with an Intel Core i7 and Nvidia GeForce GTX 980 graphics card.

Tumor gene expression datasets

We tested the proposed SESAE and SE1DCNN on three tumor datasets: breast cancer [28], leukemia [31] and colon cancer [27]. The statistics of the three datasets were summarized in Table 1. Inflammatory Breast Cancer (IBC) is a clinically defined variant of breast cancer characterized by its rapid onset and swollen, erythematous, and edematous presentation of the breast. The IBC dataset contains 30006 genes on 20 samples. There are two classes in 20 samples: 8 IBC samples and 12 non-IBC samples. Leukemia is a heterogeneous disease, usually caused by non-random chromosomal translocations that produce aberrant gene fusions or inappropriate expression of oncogenes and the prognosis for cure differs considerably among these genetic subtypes. Leukemia dataset contains 12600 genes on 60 samples. In [31], the 60 samples was processed into four classes: mercaptopurine alone (MP), high-dose methotrexate alone (HDMTX), high-dose methotrexate and mercaptopurine (HDMTX+MP), low-dose methotrexate and mercaptopurine (LDMTX+MP) and the corresponding number of samples are 12, 20, 10, 18. Colon cancer is a malignant tumor arising from the inner wall of the large intestine. In [27], colon cancer contains 2000 genes on 62 samples. There are 22 normal and 40 tumor colon samples.

Parameter determination

For each dataset, Inf-FS method was adopted as the dimensionality reduction algorithm to select genes. For fair comparison, 500 genes were selected by Inf-FS for each method. We performed 10-fold cross-validation and results were presented in terms of the average classification accuracy. In this subsection, the number of corrupted genes a was tested. For each a, we provided the parameters of SESAE and SE1DCNN on different tumor datasets. For SESAE and SE1DCNN, the choices of parameters might not be the best but effective for tumor classification.

Here, $a = 1,2,3,4,5$were tested. We tested the number of nodes of hidden layers in SESAE. Simultaneously, we also tested the number and size of convolution filters and the size of filters in max pooling. For each dataset, we took 20% samples of each class to expand the training samples and the rest 80% samples as testing samples. The parameters and classification accuracies of SESAE and SE1DCNN with different number of corrupted genes on breast cancer were summarized in Tables 2-3, respectively. In the case of $a = 1,2,3,4,5$, the number of training samples is 2505, 1255, 835, 630, 505, respectively. From Table 2, the best classification results of SESAE on breast cancer is 87.33% when $a = 1$. From Table 3, in the case of $a = 1$ and $a = 2$, SE1DCNN can reach the best performance 95.33%.

The parameters and classification accuracies of SESAE and SE1DCNN with different number of corrupted genes on leukemia dataset were summarized in Tables 4-5, respectively. When $a = 1,2,3,4,5$, the number of training samples is 6513, 3263, 2171, 1638, 1313, respectively. From Table 4, the best classification

Table 2: The parameters and classification accuracies of SESAE with different number of corrupted genes on breast cancer

Layer		*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
Hidden Layer1	Number of Nodes	50	50	50	50	50
Hidden Layer2	Number of Nodes	50	50	50	50	50
Accuracy (%)		87.33	86.67	86.00	86.67	86.00

Table 3: The parameters and classification accuracies of SE1DCNN with different number of corrupted genes on breast cancer

Layer		*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
C1 Filter	Number	11	11	5	11	11
	Size	21	21	21	21	21
M1 Filter	Size	4	4	4	4	4
C2 Filter	Number	5	5	5	5	5
	Size	21	21	21	21	21
M2 Filter	Size	4	4	4	4	4
Accuracy (%)		95.33	95.33	93.33	94.67	94.00

Table 4: The parameters and classification accuracies of SESAE with different number of corrupted genes on leukemia dataset

Layer		*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
Hidden Layer 1	Number of Nodes	30	30	30	30	30
Hidden Layer 2	Number of Nodes	30	30	30	30	30
Accuracy (%)		49.79	49.36	48.72	48.30	48.51

Table 5: The parameters and classification accuracies of SE1DCNN with different number of corrupted genes on leukemia dataset

Layer		*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
C1 Filter	Number	22	17	22	9	17
	Size	21	21	21	21	21
M1 Filter	Size	4	4	4	4	4
C2 Filter	Number	5	5	5	16	5
	Size	21	21	21	21	21
M2 Filter	Size	4	4	4	4	4
Accuracy (%)		57.87	57.02	57.24	56.17	55.96

result of SESAE is 49.79% when $a = 1$. From Table 5, the best performance of SE1DCNN is 57.87% when $a = 1$.

The parameters and classification accuracies of SESAE and SE1DCNN with different number of corrupted genes on colon cancer were summarized in Tables 6-7, respectively. When $a = 1,2,3,4,5$, the number of training samples is 6513, 3263, 2171, 1638, 1313, respectively. From Table 6, the best classification result of SESAE is 84.49% when $a = 1$. From Table 7, the best performance of SE1DCNN is 85.51% when $a = 2$.

Table 6: The parameters and classification accuracies of SESAE with different number of corrupted genes on colon cancer

Layer		*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
Hidden Layer 1	Number of Nodes	100	100	100	100	100
Hidden Layer 2	Number of Nodes	100	100	100	100	100
Accuracy (%)		84.49	83.68	83.28	83.89	83.69

Table 7: The parameters and classification accuracies of SE1DCNN with different number of corrupted genes on colon cancer

Layer		*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
C1 Filter	Number	25	5	20	12	20
	Size	21	21	21	21	21
M1 Filter	Size	4	4	4	4	4
C2 Filter	Number	20	10	7	9	5
	Size	21	21	21	21	21
M2 Filter	Size	4	4	4	4	4
Accuracy (%)		84.90	85.51	85.30	84.49	85.10

Table 8: The classification accuracies (%) of SESAE and SE1DCNN on three datasets with different values of a when all the samples are expanded

Dataset	Method	*a*=1	*a*=2	*a*=3	*a*=4	*a*=5
Breast Cancer	SESAE	99.88	99.78	99.75	99.53	99.49
	SE1DCNN	99.94	99.84	99.81	99.74	99.68
Leukemia	SESAE	99.78	99.55	99.46	99.20	98.94
	SE1DCNN	99.84	99.67	99.54	99.37	99.12
Colon Cancer	SESAE	99.96	99.94	99.93	99.86	99.86
	SE1DCNN	99.98	99.95	99.93	99.88	99.86

Furthermore, we expanded all the samples, and tested whether the corrupted samples can be correctly classified. The classification accuracies of SESAE and SE1DCNN on three datasets with different a were provided in Table 8. For breast cancer, in the case of $a = 1$, SESAE and SE1DCNN have the best results 99.88% and 99.94%, respectively. For leukemia, in the case of $a = 1$, SESAE and SE1DCNN have the best results 99.78% and 99.84%, respectively. For colon dataset, in the case of $a = 1$, SESAE and SE1DCNN have the highest accuracies 99.96% and 99.98%, respectively. The results indicate that the corrupted samples can be correctly classified and the meaningful features are successfully captured by the corrupted samples.

Comparison with other methods

To demonstrate the effectiveness of SESAE and SE1DCNN for tumor classification, traditional 1DCNN [26], traditional SAE, SAE in [13], SAE with fine tuning in [13], and Softmax/SVM were employed for comparison. SAE in [13] and SAE with fine tuning in [13] use other tumor gene expression data from the same platform to achieve feature learning since the number of labeled samples in tumor data is really small. The results were shown in Table 9. The best performance in Table 9 was indicated by bold.

On all the three datasets, SE1DCNN has better performance than all the other methods. On colon and breast cancer datasets, except for SE1DCNN, SEASE outperforms the other methods. On leukemia dataset, except for SE1DCNN and 1DCNN, SESAE have the best performance among all the 5 methods. This indicates that our SE method is very effective in classifying tumor data. Without SE method, 1DCNN outperforms traditional SAE, SAE in [13], SAE with fine tuning in [13], and Softmax/SVM on breast and colon cancer. Except for SE1DCNN,

Table 9: The classification accuracies (%) of different methods on three datasets

Methods	Breast Cancer	Leukemia	Colon Cancer
SE1DCNN	**95.33**	**57.87**	**84.90**
1DCNN	86.00	51.49	83.67
SESAE	87.33	49.79	84.49
SAE	80.67	32.55	82.07
SAE in [13]	63.33	33.71	66.67
SAE (Fine tuning) in [13]	83.33	33.71	83.33
Softmax/SVM	85.0	46.33	83.33

1DCNN has better performance than the other methods, including SESAE, on leukemia dataset. The performance of SE1DCNN and 1DCNN demonstrate that 1DCNN is a powerful method when achieving tumor classification.

METHODOLOGY

Sample expansion method

In the classification problem, it is particularly critical to obtain a good feature representation. The traditional Autoencoder (AE) can learn a useful representation by encoder. However, we cannot obtain robust features by using Autoencoder. A very different strategy is proposed by Vincent et al. to get a high-level representation: cleaning partially corrupted input, or in short denoising [29]. There are two underlying ideas in this strategy: Firstly, a good representation should be robust and stable when the input is damaged; Secondly, denoising is required to extract features that obtain useful structure in the input distribution. This denoising strategy was successfully used into Autoencoder and Denoising Autoencoder. The graphical representation of AE and DAE is described in Figure 1. In Figure 1(A), denote a vector $\boldsymbol{x}$ as an input firstly. Secondly, $\boldsymbol{x}$ is mapped to $\boldsymbol{y}$ via an encoder. Thirdly, the Autoencoder attempts to reconstruct $\boldsymbol{x}$ by decoding, $\boldsymbol{y}$ and generates the reconstruction vector $\boldsymbol{z}$. Finally, the Autoencoder calculates the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{z}$. In Figure 1(B), DAE performs some different operations compared with Autoencoders. Firstly, raw data $\boldsymbol{x}$ is stochastically corrupted to $\tilde{\boldsymbol{x}}$. In $\tilde{\boldsymbol{x}}$, each value filled with black is forced to be 0. Secondly, the corrupted data $\tilde{\boldsymbol{x}}$ is mapped to $\boldsymbol{y}$ via an encoder. Thirdly, DAE reconstructs $\boldsymbol{x}$ by decoding $\boldsymbol{y}$, and generates the reconstruction vector $\boldsymbol{z}$. Finally, DAE calculates the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{z}$ with a loss function.

The process of denoising, that is, mapping a corrupted sample back to an uncorrupted one, can be given an intuitive geometric interpretation under the so-called manifold assumption, which states that natural high dimensional data concentrates close to a non-linear low-dimensional manifold. Based on uncorrupted samples $\boldsymbol{X}$, corrupted samples $\tilde{\boldsymbol{X}}$ obtained by applying corruption process $q(\tilde{\boldsymbol{X}} \mid \boldsymbol{X})$. During denoising training, we learn a stochastic operator $p(\boldsymbol{X} \mid \tilde{\boldsymbol{X}})$ that maps a corrupted $\tilde{\boldsymbol{X}}$ back to its uncorrupted $\boldsymbol{X}$, Corrupted samples are much more likely to be outside and farther from the manifold than the uncorrupted ones. Thus stochastic operator $p(\boldsymbol{X} \mid \tilde{\boldsymbol{X}})$ learns a map that tends to go from lower probability points $\tilde{\boldsymbol{X}}$ to nearby high probability points $\boldsymbol{X}$, on or near the manifold. Note that when $\tilde{\boldsymbol{X}}$ is farther from the manifold, $p(\boldsymbol{X} \mid \tilde{\boldsymbol{X}})$ should learn to make bigger steps, to reach the manifold. Successful denoising implies that the operator maps even far away points to a small region close to the manifold. The denoising idea can thus be seen as a way to define and learn a manifold. And it can better learn a higher level representation which is rather stable and robust under corruptions of the input. The detailed interpretation of denoising idea can be founded in [29]. As a kind of natural high dimensional data, gene expression data also has the manifold structure. So the denoising idea can be used to analyze gene expression data. Experiments in [29] show that the corrupted data is very useful. There are two main reasons can explain this result: Firstly, the corrupted data can be trained to obtain smaller weight noise than non-corrupted data; Secondly, the corrupted data reduces the generation gap between the training and testing data to a certain extent [29].

Thanks to the denoising idea, in this paper, a novel Sample Expansion method is proposed to address the problem of insufficient training samples for tumor gene expression data. Denote $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ as a tumor dataset with m genes and n samples. For each sample in $\boldsymbol{X}$, SE method randomly chooses a ($a \leq m$) genes and corrupts corresponding values to 0. Supposing the locations of the corrupted genes are non-repeated, we repeat this process $floor(m/a)$ times, where $floor()$ is a function that is rounded down, and it guarantee the number of expanded samples is an integer. And each processed sample is saved for future operations. In this approach, $floor(m/a)$ expanded samples can be obtained from one sample. Similarly, $n \times floor(m/a)$ expanded samples can be obtained from all n samples. Finally, $n \times floor(m/a)$

expanded samples and n raw samples are merged into one matrix to be taken as training data.

We give the visualization of SE method in Figure 2. Denote a tumor gene expression dataset as $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ with each row representing a gene and each column representing a sample. To be more specific, here, we set a to 2. In Figure 2, the values of the corrupted genes are filled with black in expanded samples. Statistically, for the first sample in $\boldsymbol{X}$, $floor(m/2)$ expanded samples are obtained by using SE method. Including the first sample in $\boldsymbol{X}$, $floor(m/2)+1$ samples are obtained from one sample. Other samples in $\boldsymbol{X}$ are processed in the same manner. Finally, $n \times floor(m/2)+n$ samples are stored in a matrix $\boldsymbol{Y}$ that can be taken as the training data. By utilizing SE, a large number of training samples can be obtained. The expanded samples maintain the merits of the corrupted data and address the problem of insufficient training samples of gene expression data to a certain extent.

We attempt to interpret the rationality of SE from a biological perspective. Generally, the differential expression of multiple genes may be expected to result in various diseases. Moreover, the realization of a biological process usually requires interaction of multiple genes. Unfortunately, we cannot accurately determine which combinations of genes are the decisive ones we want. After being processed by SE, the non-corrupted genes in each expanded sample are taken as a combination. The corrupted samples can generate a variety of gene combinations. Some of these gene combinations may indicate distinct biological processes or different gene co-expression [33]. From this

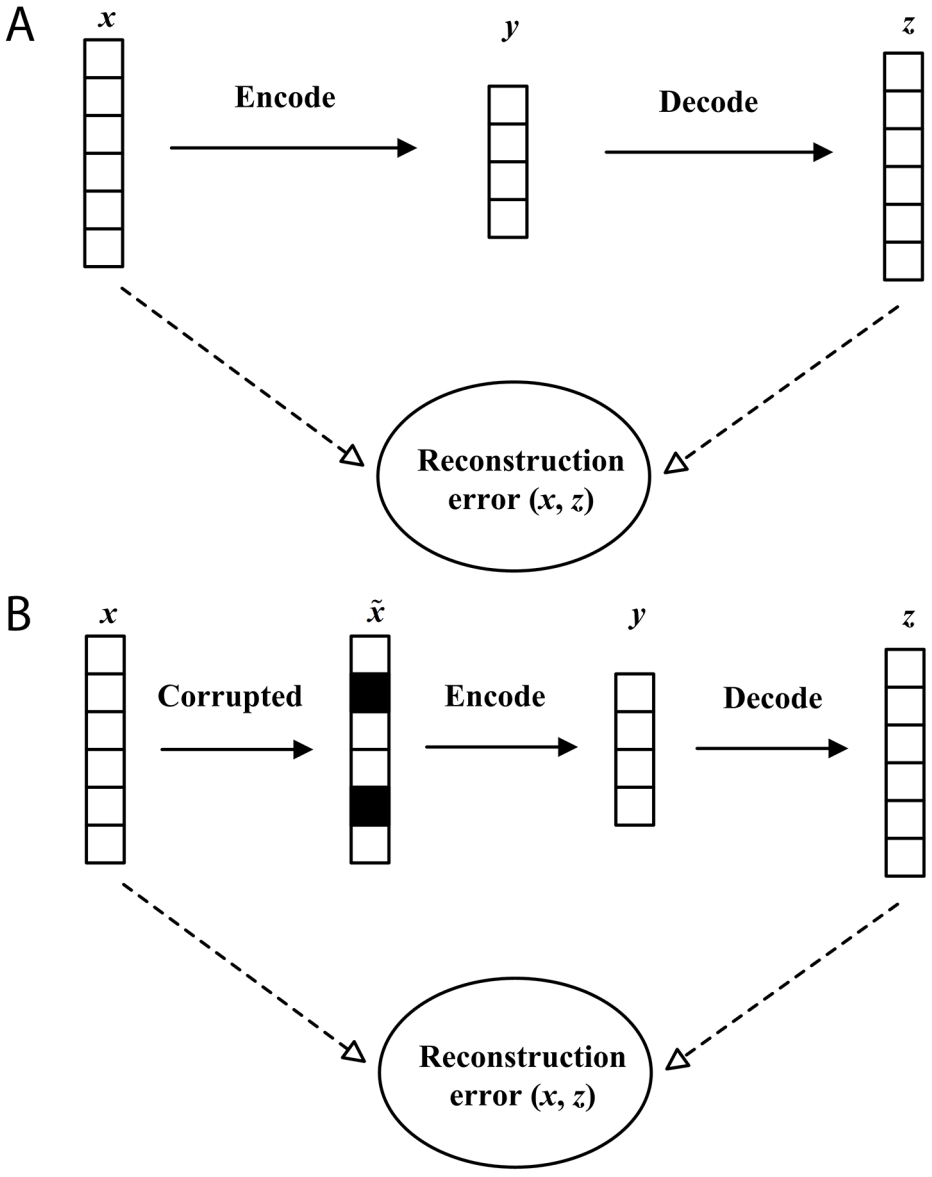

Figure 1: The graphical representation of Autoencoder **(A)** and Denoising Autoencoder **(B)**.

perspective, the class of samples can be represented more effectively by large number of gene combinations, thereby improving the classification accuracy.

Sample expansion-based SAE

An Autoencoder usually has three layers: one input layer, one hidden layer, and one reconstruction layer (See Figure 1(A)).

During training, the input $\boldsymbol{x}$ is mapped to the hidden layer and produces the latent activity $\boldsymbol{y}$. This step is called an encoder and can be formulated as follows

$$\boldsymbol{y} = f(\boldsymbol{W}_y \boldsymbol{x} + \boldsymbol{b}_y), \tag{1}$$

where $\boldsymbol{W}_y$ denotes the input-to-hidden weights, $\boldsymbol{b}_y$ denotes the bias of hidden units, and $f(\cdot)$ denotes the activation function. Here, the sigmoid function is taken as the activation function.

Then, $\boldsymbol{y}$ is mapped to a reconstruction layer by a decoder. The reconstructed value is denoted as $\boldsymbol{z}$. This step can be written as

$$\boldsymbol{z} = f(\boldsymbol{W}_z \boldsymbol{y} + \boldsymbol{b}_z), \tag{2}$$

where $\boldsymbol{W}_z$ denotes the hidden-to-output weights, $\boldsymbol{b}_z$ denotes the bias of output units.

In this paper, we hold the following constraint: $\boldsymbol{W}_y = \boldsymbol{W}_z = \boldsymbol{W}$ This can help to halve model parameters. Therefore, three groups of parameters, $\boldsymbol{W}$, $\boldsymbol{b}_y$ and $\boldsymbol{b}_z$, need to be learned.

The goal of Autoencoder is to minimize the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{z}$

$$\arg\min_{\boldsymbol{W},\boldsymbol{b}_y,\boldsymbol{b}_z} \text{cost}(\boldsymbol{x},\boldsymbol{z}), \tag{3}$$

where $cost(\boldsymbol{x},\boldsymbol{z})$ denotes the reconstruction error. The weight updating rule can be defined as

$$\boldsymbol{W} = \boldsymbol{W} - \eta \frac{\partial \text{cost}(\boldsymbol{x},\boldsymbol{z})}{\partial \boldsymbol{W}}, \tag{4}$$

$$\boldsymbol{b}_y = \boldsymbol{b}_y - \eta \frac{\partial \text{cost}(\boldsymbol{x},\boldsymbol{z})}{\partial \boldsymbol{b}_y}, \tag{5}$$

$$\boldsymbol{b}_z = \boldsymbol{b}_z - \eta \frac{\partial \text{cost}(\boldsymbol{x},\boldsymbol{z})}{\partial \boldsymbol{b}_z}, \tag{6}$$

where η denotes the learning rate.

After the model training, the learned feature lies in the hidden layer, which can be used for classification. It can also be used as the input of a higher layer to learn a deeper feature in deep learning models. The power of Autoencoder lies in the form of reconstruction-oriented training. During reconstruction, Autoencoder only uses the information in $\boldsymbol{y}$. If an Autoencoder perfectly recovers the input from $\boldsymbol{y}$, $\boldsymbol{y}$ can maintain enough information of the input. In addition, the learned nonlinear transformation in $\boldsymbol{y}$ can be regarded as a good feature extraction process. Therefore, stacking the encoders can minimize the loss of information in data. In the meantime, the abstractive and invariant information can be preserved in the deeper features. All these characteristics promote us to choose Autoencoder to extract deep features for tumor gene expression data.

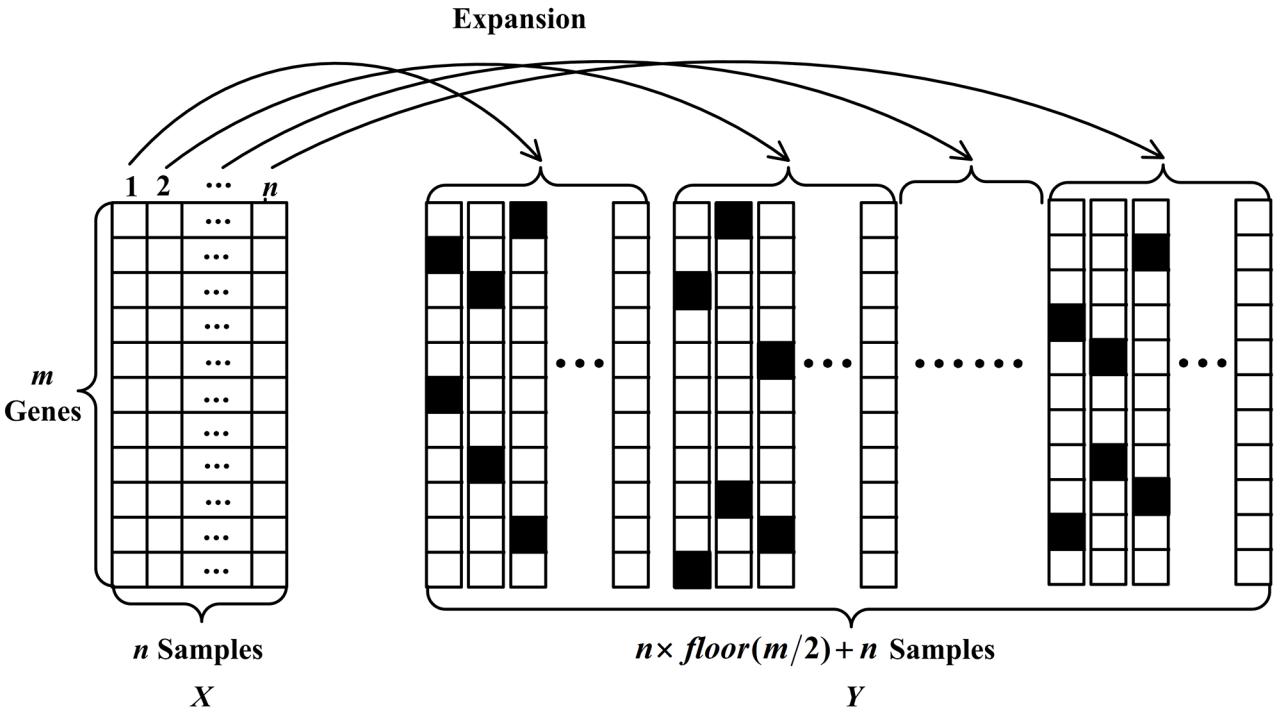

Figure 2: The schematic representation of sample expansion method.

The Stacked Autoencoder (SAE) can be constructed by stacking the input and hidden layers of Autoencoder together. The SESAE is designed by applying the SE method to SAE for tumor classification. The SESAE architecture is given in Figure 3. It consists of one input layer, two hidden layers and one output layer. SESAE first implements SE method to obtain a large number of labeled samples. Then the expanded samples and raw samples are fed into SAE. SAE first maps inputs in Input Layer to Hidden Layer 1. This step is similar with Autoencoder. After the training of Hidden Layer 1 in SAE, the inputs of subsequent layer of SAE are the output of the previous layer. We try to reconstruct the output of Hidden Layer 1according to the activity of Hidden Layer 2. After this, the decoder of the Hidden Layer 2 is cast away, and only the input-to-hidden parameters are incorporated as weights between Hidden Layer 1 and Hidden Layer 2. The subsequent classifier is also implemented as a neural network. We adopt fine-tuning strategy to adjust the parameters during training procedure. Here, we train the classifier using the back propagation method that searching for a minimum in a peripheral region of parameters initialized by the former step.

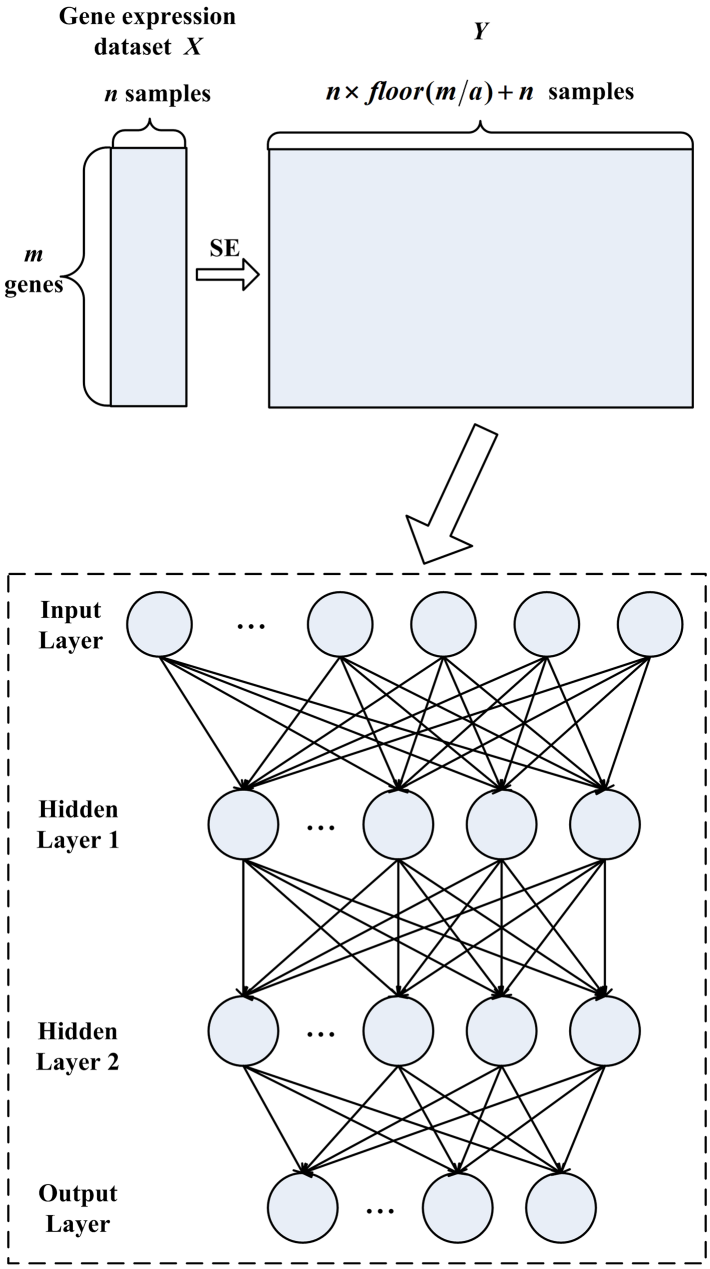

Figure 3: The SESAE architecture consisting of sample expansion process, one input layer, two hidden layers and one output layer.

Sample expansion-based 1DCNN

CNN is a classical deep learning model that exploits spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. CNN architecture consists of various combinations of convolutional layers, max pooling layers and fully-connected layers. Neurons in the same convolutional layer are sparsely connected to the neurons in the next layer and share the same weight. Weight sharing can reduce the number of trainable parameters and make CNN an effective model. The output of a convolutional layer is usually taken as the input of a max pooling layer. Max pooling layers divide the input into multiple non-overlapping windows and output the maximum value for each window. Max pooling can reduce the computation complexity for upper convolutional layers and provide translation invariance of features from the location. In the classification task, a fully-connected layer is used to integrate all the feature maps of the last pooling layer.

The training process of CNN contains two key steps: forward propagation and back propagation. The former step computes the actual classification results with current parameters while the later step updates the trainable parameters to narrow the gap between the actual classification results and the desired classification results.

Denote $\boldsymbol{x}^i$ as the output of the i-th layer and the input of the next layer. Define $\boldsymbol{x}^i$ to be

$$\boldsymbol{x}^i = f(\boldsymbol{u}^i), \tag{7}$$

with

$$\boldsymbol{u}^i = \boldsymbol{W}^i \boldsymbol{x}^{i-1} + \boldsymbol{b}^i, \tag{8}$$

where $\boldsymbol{W}^i$ is a weight matrix of the i-th layer and $\boldsymbol{b}^i$ is an additive bias vector of the i-th layer. In Eq. (7), $f(u^i)$ is the activation function of the i-th layer. In this paper, the Rectified Linear Unit (RELU) is taken as the activation function. For a classification problem with C classes and N training samples, the squared-error loss function [34] is given as

$$J^N = \frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{C}(t_c^n - y_c^n)^2, \tag{9}$$

where t_c^n is the c-th class of the n-th label, y_c^n is the value of the c-th output layer unit in response to the n-th label.

Due to the error over the whole dataset is a sum over the individual errors on each sample, the backpropagation is considered with respect to a single sample. The error function of the n-th sample is

$$J^n = \frac{1}{2}\sum_{c=1}^{C}(t_c^n - y_c^n)^2. \tag{10}$$

The errors can be regarded as sensitivities of each unit with respect to perturbations of $\boldsymbol{b}$, that is

$$\frac{\partial J}{\partial \boldsymbol{b}} = \frac{\partial J}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{b}}. \tag{11}$$

Since $\partial \boldsymbol{u}/\partial \boldsymbol{b}=1$, we can define

$$\delta = \frac{\partial J}{\partial \boldsymbol{u}}. \tag{12}$$

This derivative plays a decisive role in the back propagation from higher layers to lower layers. The following formula is used to implement the back propagation

$$\delta^i = (\boldsymbol{W}^{i+1})^T \delta^{i+1} \circ f'(\boldsymbol{u}^i), \tag{13}$$

where $\circ$ is element-wise multiplication. The sensitivities for the output layer neurons take a different form

$$\delta^{output} = f'(\boldsymbol{u}^i) \circ (\boldsymbol{y}^n - \boldsymbol{t}^n). \tag{14}$$

Finally, the delta rule is used to update weights and biases for the neurons. For the i-th layer, the weight is updated by

$$\frac{\partial J}{\partial \boldsymbol{W}^i} = \frac{\partial J}{\partial \boldsymbol{u}^i}\frac{\partial \boldsymbol{u}^i}{\partial \boldsymbol{W}^i} = (\delta^i)^T \boldsymbol{x}^{i-1}, \tag{15}$$

$$\Delta \boldsymbol{W}^i = -\eta \frac{\partial J}{\partial \boldsymbol{W}^i}, \tag{16}$$

where η is the learning rate. The bias is updated by

$$\frac{\partial J}{\partial \boldsymbol{b}^i} = \frac{\partial J}{\partial \boldsymbol{u}^i}\frac{\partial \boldsymbol{u}^i}{\partial \boldsymbol{b}^i} = \delta^i, \tag{17}$$

$$\Delta \boldsymbol{b}^i = -\eta \frac{\partial J}{\partial \boldsymbol{b}^i}. \tag{18}$$

With the increase of the number of iterations, the value of the loss function is smaller which indicates the actual output is closer to the desired output. Finally, CNN can be utilized to classify the dataset.

In image processing, the input of CNN should be a 2-dimensional image for convolution. However, each sample of gene expression data is a 1-dimensional array and the above process cannot be achieved. Therefore, 1DCNN, which asks for a 1-dimensional vector as input, is introduced in this paper. By combining the SE method and 1DCNN, we design the SE1DCNN to implement tumor classification task. In Figure 4, the designed architecture of SE1DCNN is shown. Except for the sample expansion process, 1DCNN has 7 layers: one input layer, two convolutional layers C1 and C2, two max pooling layers M1 and M2, one fully-connected layer F and one output layer. In each tumor gene expression dataset, each sample can be taken as the input of 1DCNN. In Figure 4, the input layer is a sample with m_1 genes. Suppose $\mathbf{W}_1$ with size $w_1 \times 1$ and $\mathbf{W}_2$ with size $w_2 \times 1$ are the convolutional kernels of the first and second convolutional layers C1 and C2, respectively; $\mathbf{P}_1$ with size $p_1 \times 1$ and $\mathbf{P}_2$ with size $p_2 \times 1$ are the filtering kernels of the first and second max pooling layer M1 and M2, respectively; k_1 and k_2 are the

number of kernels. After convolving the input layer, C1 contains $k_1 \times m_2 \times 1$ nodes where $m_2 = m_1 - w_1 + 1$. M1 contains $k_1 \times m_3 \times 1$ nodes where $m_3 = m_2 / p_1$. C2 contains $k_2 \times m_4 \times 1$ nodes where $m_4 = m_3 - w_2 + 1$. M2 contains $k_2 \times m_5 \times 1$ nodes where $m_5 = m_4 / p_2$. The fully-connected layer F and the output layer contain m_6 and m_7 nodes, respectively.

The classification process of SE1DCNN is the same as that of CNN. The main difference between SE1DCNN and CNN is that SE1DCNN has a sample expansion step and needs a 1-dimensional vector as input.

Tumor classification via sample expansion-based deep learning

In tumor classification task, the management of high-dimensional gene expression data requires an efficient feature selection method to individuate

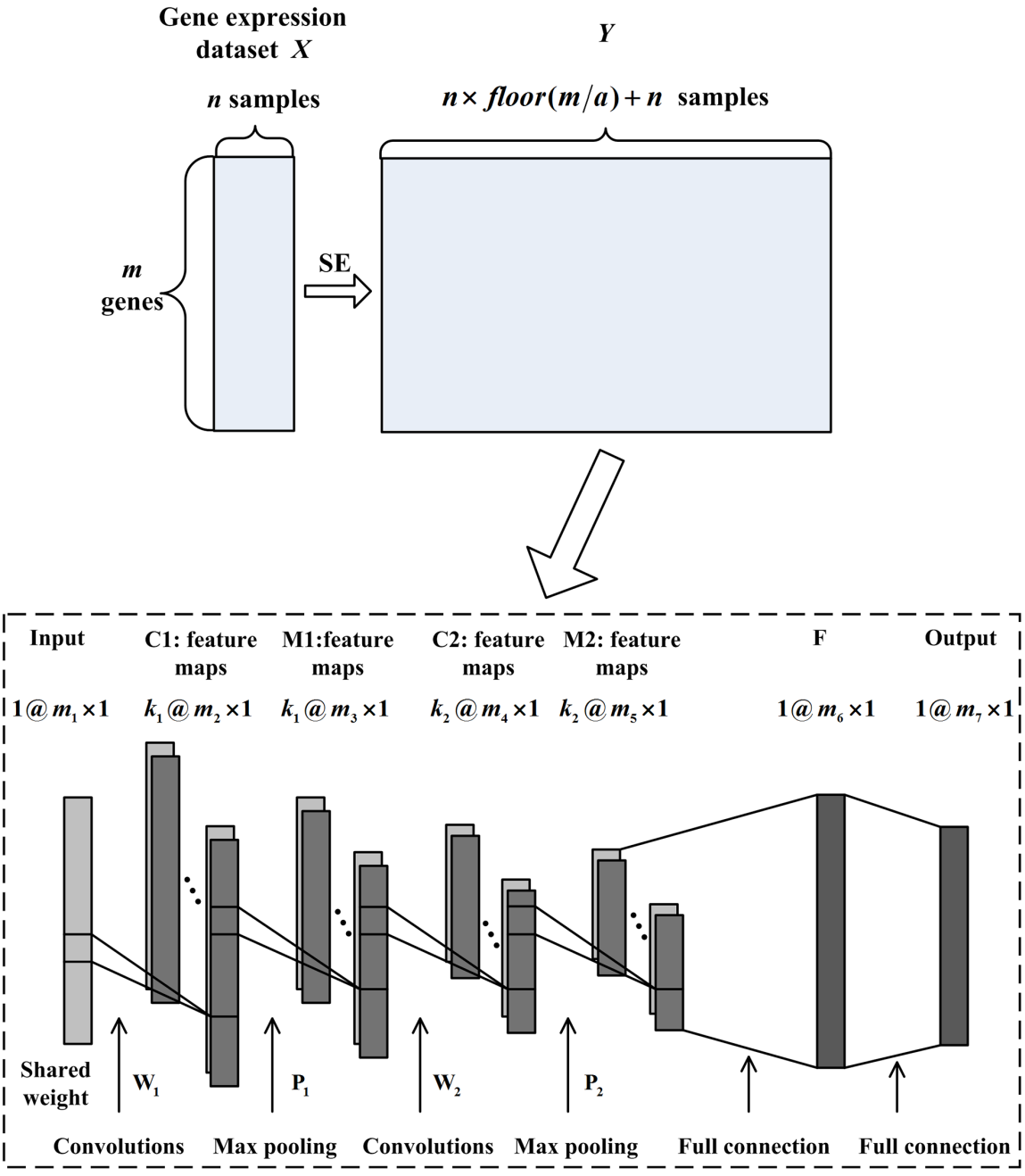

Figure 4: The SE1DCNN architecture consisting of sample expansion process, two convolutional layers, two max pooling layers and one fully-connected layer.

redundant and/or irrelevant features and avoid overfitting [35]. In [30], Roffo et al. proposed a novel unsupervised feature selection method dubbed Infinite Feature Selection (Inf-FS). The feature selection problem is mapped to an undirected fully-connected graph without label information in Inf-FS. Then a subset of features is considered as a path to connect vertices in the graph. The cost of the path, which is embedded into a cost matrix, is implemented by the combination of pairwise relationships between features and is modeled as a function of both standard deviation and Spearman's rank correlation coefficient. By construction, Inf-FS allows exploiting the convergence properties of power series of matrices, and the relevance and redundancy of one feature with respect to all the other features are calculated.

Inf-FS is an excellent feature selection method by ranking the importance individuates candidate features. The most appealing characteristic of Inf-FS is that the importance of a feature is assessed by considering all the possible subsets of all the features. In addition, the relevance and redundancy of each feature are influenced by all the other ones. Numerous experiments demonstrate that Inf-FS outperforms many classical feature selection methods, such as SVM-RFE [8], Fisher [36] and Relief-F [37]. Therefore, we adopt Inf-FS as the dimensionality reduction strategy to select genes from gene expression data.

In this paper, the framework of tumor gene expression data classification via Sample Expansion-based deep learning is given in Figure 5.

Firstly, Inf-FS is used as the dimensionality reduction strategy to select genes.

Secondly, the dimensionality reduced data is normalized. We use the following normalization

$$\hat{\boldsymbol{X}} = (\boldsymbol{X} - mean(\boldsymbol{X}))\frac{std(\hat{\boldsymbol{X}})}{std(\boldsymbol{X})} + mean(\hat{\boldsymbol{X}}), \qquad (19)$$

where $mean(\boldsymbol{X})$ is the mean of the dimensionality reduced data matrix $\boldsymbol{X}$ by row, $std(\boldsymbol{X})$ is the standard deviation of $\boldsymbol{X}$ by row, $std(\widehat{\boldsymbol{X}})$ is the standard deviation

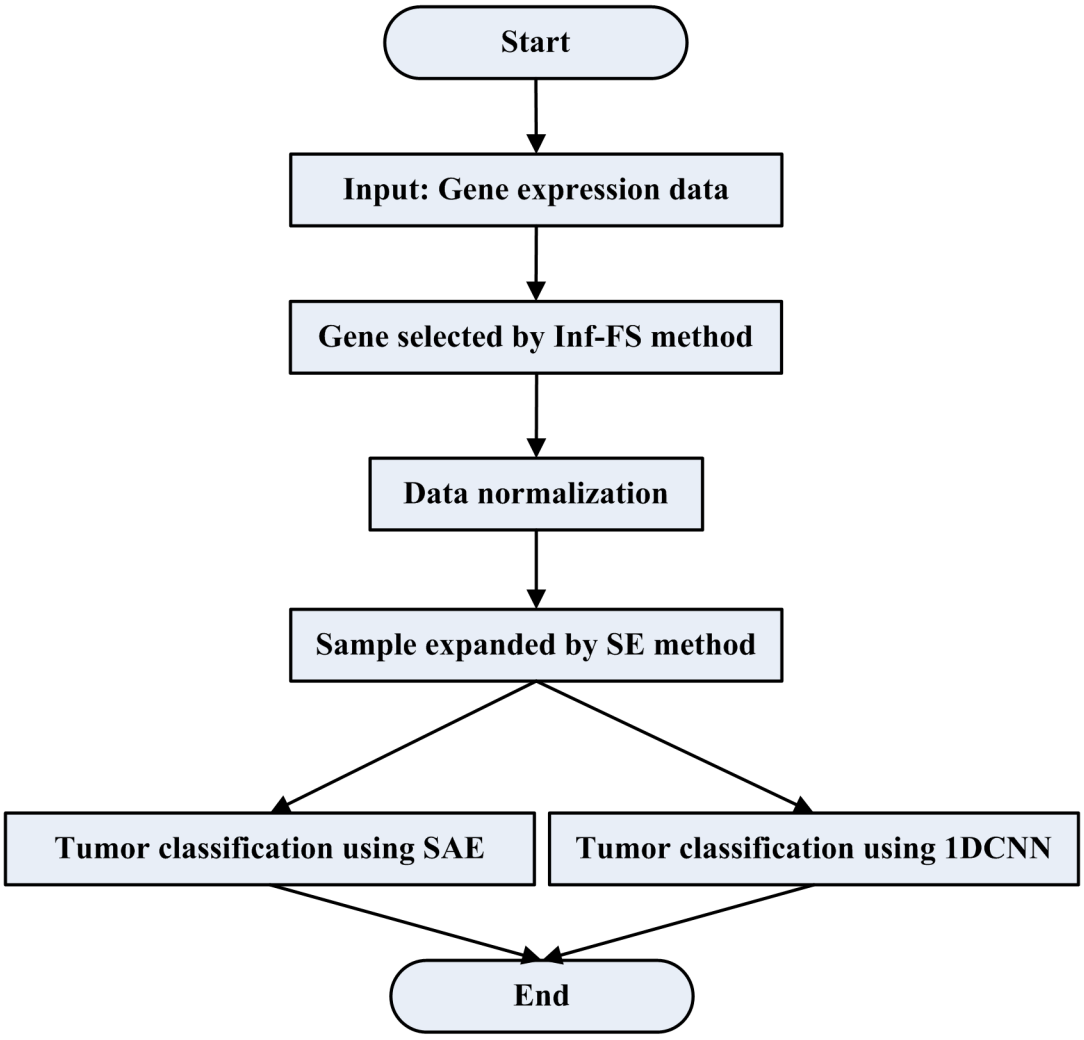

Figure 5: The flowchart of tumor classification by using SESAE or SE1DCNN.

of the expected matrix $\widehat{X}$ by row and $mean(\widehat{X})$ is the mean of the expected matrix $\widehat{X}$ by row. Here, $std(\widehat{X})$ and $mean(\widehat{X})$ are simply set to 1 and 0 respectively.

Thirdly, SE method is used to obtain a large number of training samples. The gene expression data is divided into two parts: training data and testing data. We use SE method to increase the number of labeled training data by using the separated training data.

Finally, two deep learning models, SAE and 1DCNN, are adopted to classify tumor data. The expanded and raw training data are merged into a matrix that is used as the new training data to train SAE and 1DCNN. We test SESAE and SE1DCNN by using the testing data.

CONCLUSIONS

In this paper, two sample expansion-based deep learning models, Sample Expansion-Based Stacked Autoencoder (SESAE) and Sample Expansion-Based 1D Convolutional Neural Network (SE1DCNN), are designed to classify tumor gene expression data. Firstly, since feature selection is more believable than feature extraction, an excellent feature selection method Inf-FS is used to reduce the dimensionality of gene expression data. Secondly, inspired by the denoising idea in DAE, a Sample Expansion method is proposed. The expanded samples not only have the benefits of corrupted data in DAE but also solve the problem of insufficient labeled training samples of gene expression data to a certain extent when using deep learning models. We also give an interpretation of SE method from a biological perspective. Finally, since SAE and CNN can provide excellent classification effect in many fields, the applicability of SAE and 1DCNN on gene expression data is discussed. A 4-layer SAE and a 7-layer 1DCNN are designed to classify the tumor gene expression data by using the expanded samples and raw samples.

We tested the proposed SESAE and SE1DCNN on three tumor datasets: breast cancer, leukemia and colon cancer. We first provided the parameters of SESAE and SE1DCNN on different tumor datasets with different a. This is a guide to the choice of parameters. Moreover, we expanded all samples to determine the effectiveness of the corrupted samples. The high classification accuracies of SESAE and SE1DCNN on three datasets demonstrate that the corrupted samples are very useful. Traditional 1DCNN, traditional SAE, SAE in [13], SAE with fine tuning in [13], and Softmax/SVM are employed for comparison. The classification results indicate that SE1DCNN has the best performance than the competitive methods on all the three datasets. Except for SE1DCNN and 1DCNN, SEASE outperforms the other methods on all three datasets. The performance of SESAE and SE1DCNN suggests that joint SE method and deep learning models can effectively achieve tumor gene expression data classification. The main reason is that SE method provides more and useful training samples for two deep learning models. In addition, we also find that except for SE1DCNN, 1DCNN has better performance than the other methods on leukemia dataset. And except for SE1DCNN and SESAE, 1DCNN outperforms the other methods on breast cancer and colon cancer datasets. Experimental studies on SE1DCNN and 1DCNN indicate that 1DCNN is more efficient than the other methods for tumor classification.

The limitation of this paper is mainly the explanation of SE method from a biological perspective is insufficient. In this paper, we give a short interpretation of SE method from a biological perspective. We believe that the non-corrupted genes in expanded samples can be taken as a gene combination and the corrupted samples can generate a variety of gene combinations. These gene combinations may indicate distinct biological processes or different gene co-expression. From this perspective, the class of samples can be represented more effectively by large number of gene combinations, thereby improving the performance of tumor classification. The pathogenesis of the tumor needs to be studied in future to discover the useful combinations of genes. By analyzing these combinations of genes, our SE method may give a more persuasive interpretation. In future, we will focus on the biological meaning of different combinations of genes to propose a more reasonable sample expansion strategy for tumor classification when using deep learning.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

FUNDING

This work was supported in part by the National Natural Science Foundation of China under Grant 61472424 and Grant 61772532.

REFERENCES

1. Heller MJ. DNA microarray technology: devices, systems, and applications. Ann Rev Biomed Eng. 2002; 4:129-153.
2. Sarmah CK, Samarasinghe S. Microarray gene expression: a study of between-platform association of Affymetrix and cDNA arrays. Comput Biol Med. 2011; 41:980-986.
3. Zhu L, Guo W, Deng S, Huang D. ChIP-PIT: enhancing the analysis of ChIP-seq data using convex-relaxed pair-wise interaction tensor decomposition. IEEE/ACM Trans Comput Biol Bioinform. 2016; 13:55-63.
4. Deng SP, Huang DS. An integrated strategy for functional analysis of microbial communities based on gene ontology

and 16S rRNA gene. Int J Data Min Bioinform. 2015; 13:63-74.

5. Wei P, Zhang D, Zheng C, Xia J. (2016). Cancer genes discovery based on integtating transcriptomic data and the impact of gene length. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2016; 1265-1268.
6. Tan AC, Gilbert D. Ensemble machine learning on gene expression data for cancer classification. Appl Bioinformatics. 2003; 2:S75-83.
7. Ge SG, Xia J, Sha W, Zheng CH. Cancer subtype discovery based on integrative model of multigenomic data. IEEE/ACM Trans Comput Biol Bioinform. 2016; 14:1-1.
8. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn. 2002; 46:389-422.
9. Huang DS. Systematic theory of neural networks for pattern recognition. Publishing House of Electronic Industry of China, Beijing. 1996; 201.
10. Huang DS, Du JX. A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Trans Neural Netw. 2008; 19:2099-2115.
11. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013; 35:1798-1828.
12. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Brief Bioinform. 2016.
13. Fakoor R, Ladhak F, Nazi A, Huber M. Using deep learning to enhance cancer diagnosis and classification. Proceedings of the International Conference on Machine Learning. 2013; 28.
14. Schölkopf B, Platt J, Hofmann T. Greedy layer-wise training of deep networks. International Conference on Neural Information Processing Systems. 2006; 153-160.
15. Sharma S, Umar I, Ospina L, Wong D, Tizhoosh HR. Stacked Autoencoders for Medical Image Search. arXiv.org. 2016; arXiv:1610.00320.
16. Maria J, Amaro J, Falcao G, Alexandre LA. Stacked autoencoders using low-power accelerated architectures for object recognition in autonomous aystems. Neural Process Lett. 2016; 43:445-458.
17. Liu Y, Feng X, Zhou Z. Multimodal video classification with stacked contractive autoencoders. Signal Process. 2016; 120:761-766.
18. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006; 313:504-507.
19. Taigman Y, Yang M, Ranzato MA, Wolf L. Deepface: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014; 1701-1708.
20. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014; 580-587.
21. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012; 1097-1105.
22. Ciregan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. 2012; 3642-3649.
23. Sainath TN, Mohamed AR, Kingsbury B, Ramabhadran B. Deep convolutional neural networks for LVCSR. Acoustics, speech and signal processing (ICASSP), 2013 IEEE International Conference on. 2013; 8614-8618.
24. Abdel-Hamid O, Mohamed AR, Jiang H, Penn G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. 2012; 4277-4280.
25. Kim Y. Convolutional neural networks for sentence classification. arXiv preprint. 2014; arXiv:140.85882.
26. Hu W, Huang Y, Wei L, Zhang F, Li H. Deep convolutional neural networks for hyperspectral image classification. J Sensors. 2015; 2:1-12.
27. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci. 1999; 96:6745-6750.
28. Woodward WA, Krishnamurthy S, Yamauchi H, El-Zein R, Ogura D, Kitadai E, Niwa SI, Cristofanilli M, Vermeulen P, Dirix L. Genomic and expression analysis of microdissected inflammatory breast cancer. Breast Cancer Res Treat. 2013; 138:761-772.
29. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res. 2010; 11:3371-3408.
30. Roffo G, Melzi S, Cristani M. Infinite feature selection. Proce IEEE Int ConfComput Vision. 2015; 4202-4210.
31. Cheok MH, Yang W, Pui CH, Downing JR, Cheng C, Naeve CW, Relling MV, Evans WE. Treatment-specific changes in gene expression discriminate *in vivo* drug response in human leukemia cells. Nat Genets. 2003; 34:85-90.
32. Bergstra J, Bastien F, Breuleux O, Lamblin P, Pascanu R, Delalleau O, Desjardins G, Warde-Farley D, Goodfellow I, Bergeron A. theano: deep learning on gpus with python. J Mach Learn Res. 2011; 1:1-48.
33. Deng SP, Zhu L, Huang DS. Predicting Hub Genes Associated with Cervical Cancer through Gene Co-Expression Networks. IEEE/ACM Trans Comput Biol Bioinform. 2016; 13:27-35.
34. LeCun YA, Bottou L, Orr GB, Müller KR. Efficient backprop. Neural networks: Tricks of the trade: Springer. 2012; 9-48.

35. Guyon I, Gunn S, Nikravesh M, Zadeh LA. Feature extraction: foundations and applications: Springer Science and Business Media. 2006; 207.
36. Gu Q, Li Z, Han J. Generalized fisher score for feature selection. arXiv preprint2012; arXiv:120.23725.
37. Liu H, Motoda H. Computational methods of feature selection. Chapman and Hall. 2008.