ITC 3/49
Information Technology and Control
Vol. 49 / No. 3 / 2020
pp. 421-437
DOI 10.5755/j01.itc.49.3.25918

Unsupervised Text Feature Learning via Deep Variational Auto-encoder

Received 2020/04/24 | Accepted after revision 2020/06/14

http://dx.doi.org/10.5755/j01.itc.49.3.25918

HOW TO CITE: Liu, G., Xie, L., Chen, C.-H. (2020). Unsupervised Text Feature Learning via Deep Variational Auto-encoder. *Information Technology and Control*, 49(3), 421-437. https://doi.org/10.5755/j01.itc.49.3.25918

Unsupervised Text Feature Learning via Deep Variational Auto-encoder

Genggeng Liu, Lin Xie, Chi-Hua Chen*
College of Mathematics and Computer Science; Fuzhou University; Room 108, Building 2, No. 2, Xueyuan Road, Fuzhou City, Fujian Province, China; phone: +86 18359183858; e-mail: chihua0826@gmail.com

***Corresponding author:** chihua0826@gmail.com

Dimensionality reduction plays an important role in the data processing of machine learning and data mining, which makes the processing of high-dimensional data more efficient. Dimensionality reduction can extract the low-dimensional feature representation of high-dimensional data, and an effective dimensionality reduction method can not only extract most of the useful information of the original data, but also realize the function of removing useless noise. In this paper, an unsupervised multilayered variational auto-encoder model is studied in the text data, so that the high-dimensional feature to the low-dimensional feature becomes efficient and the low-dimensional feature can retain mainly information as much as possible. Low-dimensional feature obtained by different dimensionality reduction methods are used to compare with the dimensionality reduction results of variational auto-encoder (VAE). Compared with other dimensionality reduction methods, the classification accuracy of VAE on different data sets is improved by at least 0.21% and at most 3.7%.

KEYWORDS: Machine learning, dimensionality reduction, text classification, variational auto-encoder, unsupervised feature learning.

1. Introduction

As a feature dimensionality reduction method in machine learning and deep learning, unsupervised learning aims to extract useful feature information from unlabeled data which not only can be directly applied to the recognition, classification and prediction system, but also can provide initial training values for supervised learning [4, 23, 46]. Although currently like deep convolutional neural networks (CNNs)

such a supervised learning method has achieved good results in the application [22], but whether its performance is good or bad depends on the amount of marked training sample[15, 19, 24, 27], and collecting and marking these data is very difficult [32, 47].

The Internet has many different kinds of large data [6], and more than half of the data is text data. Therefore, an unsupervised feature learning method has a decisive influence in information processing [21, 36, 39]. Taking classification algorithms as an example, text data is often semi-structured or unstructured [13], then the structured text feature dimension can be reach thousands of dimensions, which not only leads to high resource consumption in the classification algorithms, but also leads to extract inaccurate information from the document, resulting in poor classification performance [16, 44, 48]. Therefore, the most critical element in improving the accuracy and efficiency of text classification is dimensionality reduction [28, 37, 38, 40].

Dimensionality reduction is specifically used to reduce the data dimension from high dimensional m to much lower dimensional d than m. Dimensionality reduction can not only improve the efficiency of subsequent calculation, remove the irrelevant features or noise features, but also enables better interpretation of data in lower dimensions. In dimensionality reduction techniques, linear dimensionality reduction method and nonlinear dimensionality reduction method are two main components. In the linear dimensionality reduction, principle component analysis (PCA) [14] and linear discriminant analysis (LDA) [20] are two main traditional methods.

Common nonlinear dimensionality reduction includes locally linear embedding (LLE) [29], laplacian eigenmaps (LE) [1], multidimensional scaling (MDS) and isometric feature mapping (Isomap) [7, 34].

The concept of deep learning was proposed in 2006 and proves that multi-layer neural networks have better feature learning ability than shallow neural networks [17]. The hidden layer can be considered as the feature extraction layer, and the output of the hidden layer can be used as dimensionality reduction. Restricted boltzmann machine (RBM) [18, 26] and auto-encoder (AE) [2, 30] are the most common neural networks models used for dimensionality reduction. In recent years, AE has many improved models that increase the constraints on hidden layers, making hidden layer expressions different from input layers [42]. Variational auto-encoder (VAE) as an improved model of the AE model is proposed in [20]. As a generation model, VAE uses a set of data to train the model, and its output is data generated by the decoder similar to the input [43].

In this paper, using the unsupervised VAE method, the dimensionality of the text data was reduced and the dimension vector of the resulting text from the hidden layer data was extracted as the low dimensional feature representation. In conclusion, there are two novel applications:

1. An unsupervised neural network model is used to reduce the dimensionality of high dimensional and sparse text vectors.
2. Unsupervised VAE is applied to the dimensionality reduction of text vectors.

The dimensionality reduction models are used to reduce the dimensionality of the high-dimensional features of the public datasets, and the reduced-dimensional data is used to train the k-nearest neighbor (kNN), support vector machine (SVM) and random forest (RF) classifiers. The comparative experiment results show the feasibility of VAE in feature dimensionality reduction.

In the following sections, Section 2 reviews some related work about text representation algorithm and method of dimensional reduction. In Section 3, the structure and theoretical derivation of VAE and the proposed dimensional reduction algorithm using VAE for text data are presented. In Section 4, the effectiveness of the VAE is verified in text classification experiment. Finally, this paper is concluded in Section 5.

2. Related Work

Unstructured text data is transformed into vectors through text representation and then reduced in dimension. A text representation algorithm and some classical dimensionality reduction methods are introduced in the rest of this section.

2.1. Text Representation

For processing unstructured text data, Term Frequency-Inverse Document (TF-IDF) is used to convert it into institutionalized vectors that can be processed subsequently in this paper [8]. TF-IDF consists of TF

and IDF. TF is the frequency with which a word appears in an article from the document.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{1}$$

where, $TF_{i,j}$ represents the frequency of the word i in document j, $n_{i,j}$ indicates the number of occurrences of the word i appears in document j and $\Sigma_k n_{k,j}$ represents the number of words contained in document j, k is every word in document j. For example, in an article with a total of 1,000 words, the word 'feature' appears 35 times, so its word frequency is 0.035.

IDF reflects the frequency of a word appearing in all texts. If a word appears in many texts, its IDF value should be low which indicates the importance of a word in a text. The IDF of a word can be obtained by dividing the total number of files by the number of files containing the term, and then taking the logarithm of the resulting quotient.

$$\mathrm{IDF}_i = \log \frac{|Doc|}{|Doc_i| + 1}, \tag{2}$$

where, IDF_i represents the inverse document frequency of the word i, $|Doc|$ represents the total number of texts in the corpus and $|Doc|$ represents the total number of texts containing word i in the corpus. Finally, the TF-IDF of the word i in document j is calculated as the follow formula.

$$\mathrm{TFIDF}_{i,j} = TF_{i,j} \times IDF_i. \tag{3}$$

In this paper, $T_{i,j}$ is used to replace $TFIDF_{i,j}$.

2.2. Principal Component Analysis

PCA has the properties in terms of maximum separability which makes the projection of the sample points on the hyperplane after dimensionality reduction as separate as possible [12]. For a sample $X = \{x_i\}_{i=1}^m \in \mathbb{R}^{m \times n}$ (m is the data dimension and n is the number of data) in a given m dimensional space, the projection of sample point x_i onto the hyperplane in new space is $W^T x_i$. To keep the projections of all sample points as separate as possible, the variance of projected sample points $\sum_{\mathrm{i}} W^T x_i x_i^T W$ should be maximized.

$$\begin{aligned} &\max_W \quad tr\left(W^T X X^T W\right). \\ &s.t. \quad W^T W = I \end{aligned} \tag{4}$$

where, I is the identity matrix and $tr(\cdot)$ represents the trace of the matrix.

The following formula can be obtained by using the Lagrange multiplier method:

$$XX^T w_i = \lambda_i w_i. \tag{5}$$

Therefore, as long as the eigenvalue λ_i of the covariance matrix XX^T is decomposed, the maximum d features are the corresponding eigenvectors.

2.3. Multiple Dimensional Scaling

MDS guarantees that the distance of all data point pairs in low dimensional space is equal to the distance in high dimensional space. Suppose that given n instances, the distance matrix $D \in \mathbb{R}^{n \times n}$ in the original space can be calculated in Euclidean Distance formula. The element D_{ij} of the i-th row and the j-th column represents the distance between the i-th instance and the j-th instance. Transform the data into the d-dimensional space and get the representation $Z \in \mathbb{R}^{d \times n}$ of all sample points in d, where $z_i^T \in \mathbb{R}^d$ denotes the i-th instance, and the distance of any two instances in the d-dimensional space is equal to the distance in the original space [9].

The inner product matrix $B = Z^T Z \in \mathbb{R}^{n \times n}$ is obtained by the following formula:

$$b_{ij} = -\frac{1}{2}\left(D_{ij}^2 - \frac{1}{n}\sum_{j=1}^n D_{ij}^2 - \frac{1}{n}\sum_{i=1}^n D_{ij}^2 + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n D_{ij}^2\right). \tag{6}$$

By the eigenvalue decomposition of matrix B, the final low-dimensional feature can be obtained.

2.4. Isometric Mapping

The basic starting point of Isomap is that it is misleading to calculate the linear distance directly in the high-dimensional space after the low-dimensional manifold is embedded into the high-dimensional space, because the linear distance in the high-dimensional space is unreachable on the low-dimensional embedded manifold [34].

Isomap is similar to the MDS. For calculating the distance matrix D, the distance between the adjacent k reachable points of each point x_i is the Euclidean Distance, and the distance between x_i and other unreachable points is set as infinite. There is a connection between the neighboring points in the graph, and there is no connection between the non-neighboring points. Then, the problem of calculating the distance matrix D is transformed into the shortest path problem between the two points on the neighbor graph. The famous shortest path problem algorithm is Dijkstra or Floyd algorithm. Finally, the desired low-dimensional features are obtained by inputting the distance matrix D into the MDS algorithm.

2.5. Locally Linear Embedding

For Isomap, it tries to keep the distance between neighboring samples from each other, while LLE tries to keep the relationship between the samples in the relationship of samples in the neighborhood [16]. The goal of LLE is to keep the relationship of sample reconstruction in high dimensional space in low dimensional space.

$$x_i = w_{ij}x_j + w_{ik}x_k + w_{il}x_l, \tag{7}$$

where, the coordinates of x_i can be reconstructed by a linear combination of the coordinates of its neighborhood point x_{j}, x_{k} and x_l in high dimensional space. LLE method ensures the relationship will hold in low-dimensional space.

For data $X = \{x_i\}_{i=1}^{\mathrm{m}} \in \mathbb{R}^{m\times n}$, LLE calculates the linear reconstruction coefficient W through the neighborhood set Q_{j} of each point x_{j}:

$$\begin{aligned} &\max_{W} \quad \sum_{i=1}^{m}\left\| x_i - \sum_{j\in Q_i} w_{ij}x_j \right\|_2^2. \\ &s.t. \quad \sum_{j\in Q_i} w_{ij} = 1 \end{aligned} \tag{8}$$

In addition, for $x_j \notin Q_i, w_{ij} = 0$.

Low-dimensional data are obtained by eigenvalue decomposition of the matrix M:

$$M = (I-W)^T (I-W). \tag{9}$$

2.6. Laplacian Eigenmaps

LE is a local perspective to build relationships between data. The base idea is that if the two data instances x_i and x_{j} are very similar, then x_i and x_{j} should be as close as possible in the target subspace after dimensionality reduction [31]. The purpose of neighborhood preserving embedding (NPE) is to search for neighborhood construction on stream data which is similar to LLE.

LE selects neighborhood for the entire spatial sample with two methods which are ϵ-neighborhoods and k-nearest-neighbors. LE uses a thermonuclear method with a kernel width $t > 0$ [25], W is a similar matrix, and the weights are set as

$$W_{ij} = e^{-\frac{\left\|x_i - x_j\right\|^2}{t}}. \tag{10}$$

The lower dimensional vectors can be obtained by solving the problem of generalized eigenvalues of Laplace diagrams.

$$Ly = \lambda Gy, \tag{11}$$

where, G is a diagonal matrix, $G_{ii} = \sum_j w_{ij}$, and $L = G - W$.

2.7. Laplacian Eigenmaps

RBM is a modeling method based on energy function. The energy of the joint configuration of visible variable v and hidden variable h is

$$E(v,h) = -a^T v - b^T h - h^T W v, \tag{12}$$

where, W is a weight matrix, a represents the bias of hidden unit and b represents the bias of visible unit. v and h satisfy the joint probability formula.

$$P(v,h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}, \tag{13}$$

2.8. Auto-encoder

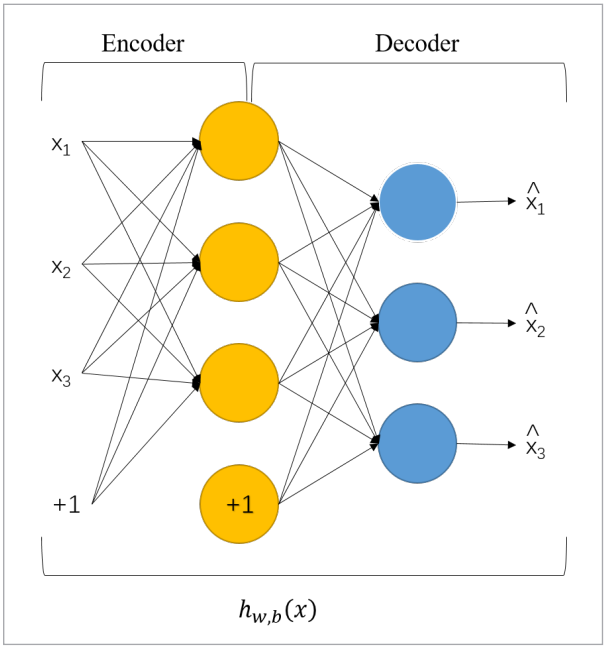
As a super-supervised feature learning method based on deep learning [41], AE can be regarded as a three-layer neural network structure: input layer,

hidden layer and output layer, as shown in Figure 1. In Figure 1, x_i represents the input node, $\hat{x}_i$ represents the output node, "+1" represents the bias item, and $h_{W,b}$ represents the approximate output of the input data after the three-layer network structure. $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m}$ are a set of unlabeled input data (m represents the data dimension and n represents the number of data), and the objective function of AE is as follows to minimize the reconstruction error [33].

$$\arg\min \sum_{i=1}^{n} \left\| X_i - \widehat{X}_i \right\|^2 . \tag{14}$$

The hidden layer neurons of the AE are much smaller than the input layer and the output layer, which allows low dimensional features to be extracted in the hidden layer.

Figure 1
Auto-Encoder



In summary, the structure of the AE and the dimensionality reduction process are shown below. The AE has a three-layer neural network structure of input layer, hidden layer, and output layer which are represented as encoding layer, hidden layer and decoding layer, respectively. The neurons of the input layer and the output layer of the AE are both set to m and the neurons of the hidden layer are set to d ($d \ll m$). Then, encoder layer encodes the network input $\{x_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$ to obtain the hidden layer output $\{z_i\}_{i=1}^{d} \in \mathbb{R}^{d\times n}$, and decoder layer reconstructs the output of $\{z_i\}_{i=1}^{d} \in \mathbb{R}^{d\times n}$ back to the spatial dimension of the original data to obtain the reconstructed $\{\hat{x}_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$. Finally, the output $\{\hat{x}_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$ is approximated to the input $\{\hat{x}_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$ continuously, so that the low-dimensional features of the data can be obtained.

2.9. Conclusion

In unsupervised feature dimension reduction methods, the methods can be divided into linear and nonlinear methods. Among them, PCA described in this section is the linear method.

Nonlinear dimensionality reduction can divided into two types of dimensionality reduction which are preserving local features and preserving global features. Preserving local features methods in dimensionality reduction involves reconstruction weight and using the collar graph. LLE abandons the global optimal dimensionality reduction of all samples, but guarantees local optimum. LLE is a dimensionality reduction method based on reconstructed weights and LE is based on collar graph. LE guarantees that the relevant points in the dimensionality reduction space (the points connected in the collar graph) are as close as possible, so that the method keeps the original data structure unchanged after dimensionality reduction. There are MDS, Isomap and neural network in dimensionality reduction method for preserving global features. After dimensionality reduction, both MDS and Isomap keep the distance between the samples and the original distance unchanged.

RBM consists of a visible layer and a hidden layer, which is a stochastic neural network model. Similar to a general feedforward neural network, RBM is not connected between neurons in the same layer, and adjacent layers are completely connected. The hidden layer can be considered as the feature extraction layer, and the output of the hidden layer can be used as dimensionality reduction. A symmetric AE network is also composed of neural networks, the output layer reconstructs the input layer so that the network can encode the data. After the training is completed, the output of the hidden layer can be used as the dimen-

sionality reduction as well as the hidden layer of the RBM. AE shows that if the hidden layer of the model can also reconstruct the input data at this time, the hidden layer data is sufficient to represent the input data. Then, the output of the hidden layer can be considered as an effective feature that is automatically learned from the model. Table 1 summarizes the relationship between these methods.

Table 1
Feature dimension reduction methods

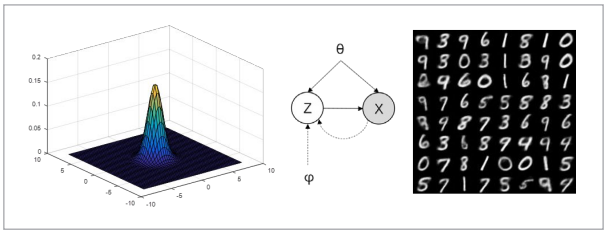
Linear	PCA	
Nonlinear	Preserving local features	LLE, LE
	Preserving global features	MDS, Isomap
	Neural network	RBM, AE

All of the above seven methods have been used to reduce the dimensionality of the data. In this paper, they will be compared by text data dimensionality reduction experiment. In addition, the Section 3 also proposed to adopt VAE to reduce the dimension of sparse text data, this method will also be tested by comparative experiments.

3. Methodology

VAE is a generation model that generates data based on a certain distribution z. As shown in Figure 2, by limiting z to satisfy a certain Gaussian distribution, then obtaining z from the Gaussian distribution $p(z)$, and finally generating the data from the distribution $p_\theta(x \mid z)$. The theory proves that the data of arbitrary distribution $p(\mathrm{x})$ can be generated by the hidden variable z satisfying the Gaussian distribution through the neural network. As shown in Figure 2, in the probabilistic graph model of VAE, the generated model $p_\theta(x \mid z)$ (the solid line $z \to x$) is equivalent to the decoder and the recognition model $q_\varphi(z \mid x)$ (the dotted line $x \to z$) is equivalent to the encoder.

Figure 2
The probabilistic graph model of VAE



3.1. Objective Function

In general, in order to make the generated data $\hat{x}$ the most similar to the original data x, the distribution $p_\theta(x)$ should be maximize. Then, the maximum likelihood method is used to maximize the following likelihood functions:

$$p_\theta(x) = \int p_\theta(x \mid z)\, p_\theta(z)\, dz, \tag{15}$$

where, $p_\theta(x \mid z)$ represents the reconstruction of the original data x from the hidden variable z. The prior distribution of the hidden variable z is represented by $p_\theta(z)$. The hidden variable z is obtained from the original data, and the process is represented by $p_\theta(z \mid x)$. Since $p_\theta(z \mid x)$ is relatively difficult to calculate, the VAE replaces the real posterior $p_\theta(z \mid x)$ with an approximate posterior $q_\varphi(z \mid x)$ obeying the Gaussian distribution. In order to measure the similarity between the two distributions, Kullback-Leibler divergence (KL divergence) is often used to measure the distance between two random distributions, and when two random distributions are the same, their KL divergence is zero. Then $\mathbb{E}_{q_\varphi(z|x)}$ represents $\sum q_\varphi(z \mid x)$, i.e.

$$\begin{aligned} & D_{KL}(q_\varphi(z \mid x) \| p_\theta(z \mid x)) \\ & = \mathbb{E}_{q_\varphi(z|x)} \log \frac{q_\varphi(z \mid x)}{p_\theta(z \mid x)} \\ & = \mathbb{E}_{q_\varphi(z|x)} \left[\log q_\varphi(z \mid x) - \log p_\theta(z \mid x) \right] \\ & = \mathbb{E}_{q_\varphi(z|x)} \left[\log q_\varphi(z \mid x) - \log p_\theta(x \mid z) - \log p_\theta(z) \right] \\ & + \log p_\theta(x) \end{aligned}. \tag{16}$$

Therefore,

$$\begin{aligned} & \log p_\theta(x) - D_{KL}(q_\varphi(z \mid x) \| p_\theta(z \mid x)) \\ & = \mathbb{E}_{q_\varphi(z|x)} \left[-\left(\log q_\varphi(z \mid x) + \log p_\theta(z) \right) + \log p_\theta(x \mid z) \right]. \end{aligned} \tag{17}$$

The KL divergence is not negative,

$$D_{KL}(q_\varphi(z \mid x) \| p_\theta(z \mid x)) \geq 0 \tag{18}$$

$$\begin{aligned} & L(\theta, \varphi; x) \\ & = \mathbb{E}_{q_\varphi(z|x)} \left[-\left(\log q_\varphi(z \mid x) + \log p_\theta(z) \right) + \log p_\theta(x \mid z) \right]. \end{aligned} \tag{19}$$

Then according to the above formula, the inequality is derived:

$$\log p_\theta(x) \geq L(\theta,\varphi;x). \tag{20}$$

Maximizing the logarithmic likelihood function $\log p_\theta(x)$ to lead the posterior distribution $q_\varphi(z\mid x)$ to be close to the true posterior distribution $p_\theta(z\mid x)$, which means that $D_{KL}(q_\varphi(z\mid x)\,\|\,p_\theta(z\mid x))$ is close to 0. In VAE, $L(\theta,\varphi;x)$ is considered to be the lower bound of the variation of $\log p_\theta(x)$. In order to optimize $\log p_\theta(x)$ and $D_{KL}(q_\varphi(z\mid x)\,\|\,p_\theta(z\mid x))$, the loss function of VAE can be obtained from the lower bound of the variation, i.e.

$$\begin{aligned} &L(\theta,\varphi;x)\\ &=\mathbb{E}_{q_\varphi(z|x)}\left[-\left(\log q_\varphi(z\mid x)+\log p_\theta(z)\right)+\log p_\theta(x\mid z)\right]\cdot\\ &=-D_{KL}(q_\varphi(z\mid x)\,\|\,p_\theta(z))+\mathbb{E}_{q_\varphi(z|x)}\left[\log p_\theta(x\mid z)\right] \end{aligned} \tag{21}$$

In the loss function $L(\theta,\varphi;x)$ of VAE: $D_{KL}(q_\varphi(z\mid x)\,\|\,p_\theta(z\mid x))$ is regularizer, and $\mathbb{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right]$ is reconstruction error. The process of optimizing the loss function $L(\theta,\varphi;x)=\sum_{i=1}^{N}L\left(\theta,\varphi;x^{(i)}\right)$ is equivalent to optimizing the two parts regularizer and reconstruction error above.

– Regularization terms

$p_\theta(z)$ obeys the Gaussian of $N(0;I)$ and $q_\varphi(z\mid x)$ obeys the Gaussian of $N\left(\mu,\sigma^2\right)$, conforms to the formula.

$$-D_{KL}(q_\varphi(z\mid x^i\|\,p_\theta(z))=\frac{1}{2}\sum_{j=1}^{d}(1+\log(\sigma_j^i)^2-(\mu_j^i)^2-(\sigma_j^i)^2). \tag{22}$$

j is the dimension of z.

– Reconstruction error term

To evaluate the expectation of the function $f(z)$ relative to $q_\phi(z\mid x^i)$, the Monte Carlo evaluation is used here.

In the distribution $z^{i,l}\sim q_\phi(z\mid x^i)$, sample L' hidden variables $z^{i,l}$, $l=1,2,3,\cdots,L'$ (L' is usually 1), and then average the $f(z)$. In this way, a concrete formula of the reconstructed error term can be obtained.

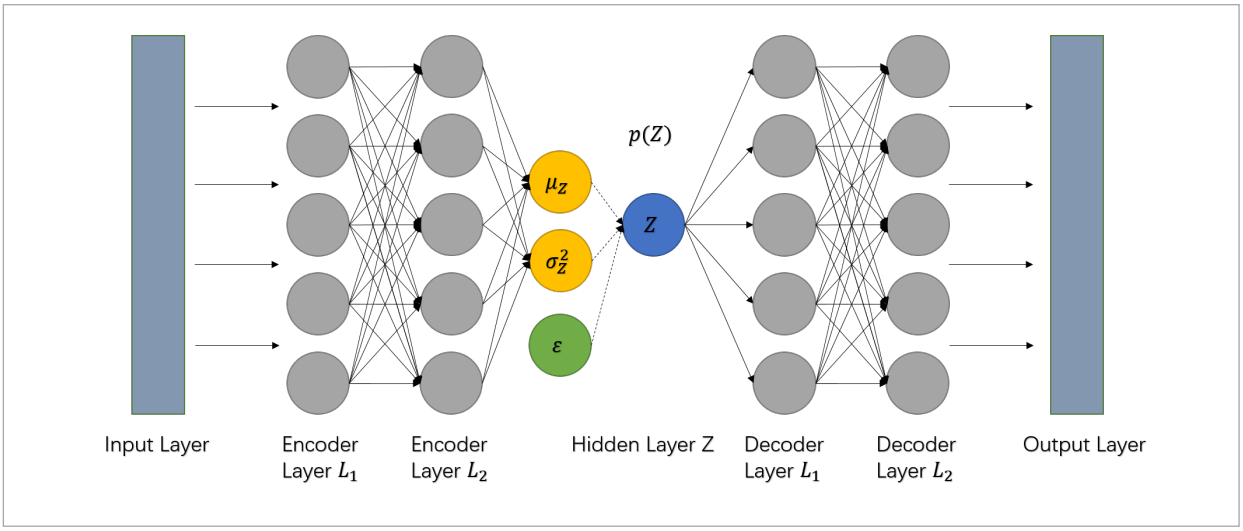
$$E_{q_\varphi(z|x^i)}[\log p_\theta(x^i|z)]=\frac{1}{L'}\sum_{l=1}^{L}\log p_\theta(x^i|z^l)=\log p_\theta(x^i|z). \tag{23}$$

Since z is not derivable, $\mu+\varepsilon\cdot\sigma$ is used to convert z into derivable μ and σ. Finally, the reconstruction error term is derived as shown in the following equation.

$$\log p_\theta(x|z)=-\sum_{i=1}^{n}\left(\left(\frac{1}{2}\left\|\frac{x^i-\mu_i}{\sigma_i}\right\|\right)+\log\left(\sqrt{x\pi}\sigma_i\right)\right). \tag{24}$$

Finally, the structure of the whole VAE neural network is shown in the Figure 3. The first half of the model is the encoder and the second half of the model is the decoder. The dotted line indicates sampling and the solid line indicates forward propagation. Among

Figure 3
Variational Auto-Encoder



them, the encoding layer represents the process of extracting the characteristics of text data, the hidden layer is the VAE sampling process, and the decoding layer represents the process of restoring the sampled data to the original data.

3.2 .Optimization Algorithm

Through the derivation process, it can be obtained that as a variational self-encoding of an auto-encoder structure, the encoding layer is a process of extracting features of text data. The hidden layer first calculates the mean and variance of the output of the coding layer, and then generates a Gaussian distribution for sampling. The decoding layer is a process of reconstructing the sampled data. Finally, the VAE feature extraction algorithm generated by the variational lower bound $L(\theta,\varphi;x)$ is trained with the stochastic gradient descent method, as detailed in Algorithm 1.

3.3 Text Feature Dimension for Deep VAE

In the first step, unstructured text data cannot be directly used as input to the VAE network for which text data need to be vectorized. Before that, a series of preprocessing is required for text data, including lemmatization, deletion of stop words and removal of low-frequency words. Lemmatization refers to the transformation of morphological words such as singular, plural and tense into prototypes. Stop word is a very common word that is not strongly related to a particular domain terminology. Stop words contain no information and should be deleted. Low frequency words refer to words that only appear in a few texts and have no practical meaning to most texts, so they need to be removed.

And then, the clean secondary sequence obtained after pretreatment is transformed by using TF-IDF to obtain the TF-IDF value corresponding to each word. For the text vector of an article, it is represented as the long vector composed of all words in the m-dimensional corpus. The corresponding value of words appearing in article j is the TF-IDF value obtained through calculation, while the corresponding value of words not appearing is 0 e.t.

$$X_j = \left\{T_{1,j}, 0, T_{3,j}, T_{4,j}, 0, T_{6,j}, 0, \cdots, 0, T_{m,j}\right\}. \quad (25)$$

Through the above process, the text vector $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$ corresponding to n articles are obtained. Finally, input $X \in \mathbb{R}^{m\times n}$ into Algorithm 1 to get the final d-dimensional feature representation of $X \in \mathbb{R}^{m\times n}$. This means that $X \in \mathbb{R}^{m\times n}$ will be used as input, and a VAE model based on input will be trained according to Algorithm 1. When the iteration ends and the loss converges, $X \in \mathbb{R}^{m\times n}$ will be input into the trained model again, and finally the hidden layer z will be extracted as the output. The output z at this time is the low-dimensional feature of $Z \in \mathbb{R}^{d\times n}$. This concept is illustrated in Figure 4.

Algorithm 1. VAE is trained by stochastic gradient descent. Use the variational lower bound $L(\theta,\varphi;x)$

Input: Training data $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$,
learning rate α, minibatch M, training epochs T.
Number of neurons: input layer neurons = m,
1st hidden encoder layer neurons = m_1,
2nd hidden encoder layer neurons = m_2,
hidden layer z neurons = d,
1st hidden decoder layer neurons = m_2,
2nd hidden encoder layer neurons = m_1,
output layer neurons = m.

Output: Z (Input $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$ into the trained encoder to obtain the low-dimensional feature $Z = \{z_i\}_{i=1}^{d} \in \mathbb{R}^{d\times n}$)

1: Random initialization parameter θ, φ

2: **for** epoch = 1:T **do**

3: Sample M samples randomly from data set $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m\times n}$: X^M

4: Sample ε from the noise distribution $N(0;I)$: ε

5: g is updated according to $\nabla_{\theta,\varphi} L(\theta,\varphi;x^M,\varepsilon)$

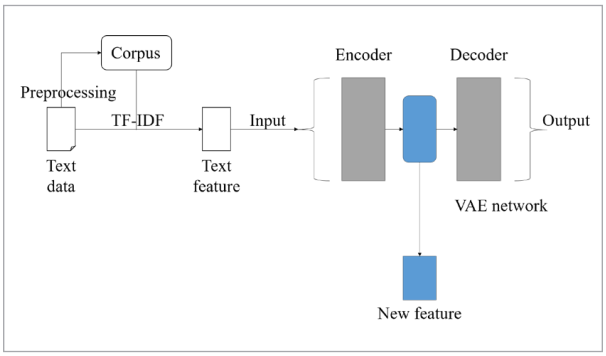
6: Using gradients g to update parameters θ, φ

7: Using gradients parameters φ to update encoders' weight matrix W_q. Using gradients parameters θ to update decoders' weight matrix W_p

8: **end for**

9: **return** The low-dimensional feature Z is obtained by the trained encoder network $q_\varphi(z \mid x)$ of the data set X

Figure 4
Text feature dimension for VAE



4. Experimental Results and Discussion

In this section, the VAE algorithm is used to reduce the dimensionality of 3 datasets and compare it with 7 classical dimensionality reduction methods. Then, the text data is generated as a labeled text vector (X, Y), where $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m \times n}$, labels $Y = y_i \in \mathbb{R}^n$. The VAE network is first trained using the text vector X, and then the encoder output $Z = \{z_i\}_{i=1}^{d} \in \mathbb{R}^{d \times n}$ is taken as the low dimensional feature of the text vector X. Finally, VAE is compared with other feature dimension reduction methods in the classification experiment.

In addition, this paper uses the Tensorflow framework to design and implement all models, and uses the GPU version to accelerate the training time of the model. Complete all deep learning work under the following configuration:

CPU: Inter(R) Core(TM) i7-7700HQ CPU@2.80GHz, memory: 24.0G DDR4, graphics card: NVIDIA GeForce GTX 1060 3.0G, tensorflow-gpu: 1.5.

All the images generated by the experiment were drawn using matlab2016.

4.1. Dataset Summarization

Experiments are contained three public text datasets, such as text data and gene expression.

Basehock has 1993 data and each line in data represents a text feature vector is 4862 dimension.

DBWorld contains 64 e-mails collected from DBWorld mailing list [50]. DBWorld has 64 data and each data has 3721 dimension vectors.

In RCV1_4Class dataset, there are 9,625 documents with 29,992 distinct words and including 4 categories [5]. It means that RCV1_4Class has 9625 data and each data has 29992 dimension vectors.

These datasets are described in Table 2.

Table 2
Data description

ID	Database	#Instances	#Features	#Classes
1	DBWorld	64	3721	2
2	Basehock	1993	4862	2
3	RCV1_4Class	9625	29992	4

4.2. Experimental Setting

To verify the efficiency of VAE, the VAE is compared with the following seven feature dimensional reduction methods mentioned in the related work chapter. They are PCA, MDS, LLE, LE, Isomap, RBM and AE. Figure 5 shows the comparison of dimension reduction by the above methods in the Swiss roll dataset.

In order to more fairly compare the performance of different feature dimensionality reduction methods, the following parameters need to be preset. All dimensionality reduction methods set the reduced dimensions of all datasets to {20,30,40,50,60,70,80,90,100} and the best classification results are recorded as the final results.

The experimental results show the best classification accuracy using kNN, SVM and RF classification algorithms with given low-dimensional feature numbers. As a classic classification algorithm, kNN's main idea is to select the k points closest to the test point distance, and then predict the category with the most occurrences as the category to which the current test point belongs. The SVM was originally proposed in [35], which finds a partition hyperplane in space based on dataset $X = \{x_i\}_{i=1}^{m} \in \mathbb{R}^{m \times n}$, so as to separate the data of different categories. RF is an algorithm that integrates multiple trees through the idea of ensemble learning and

Figure 5
A comparison of several feature dimensional reduction methods in the Swiss roll dataset

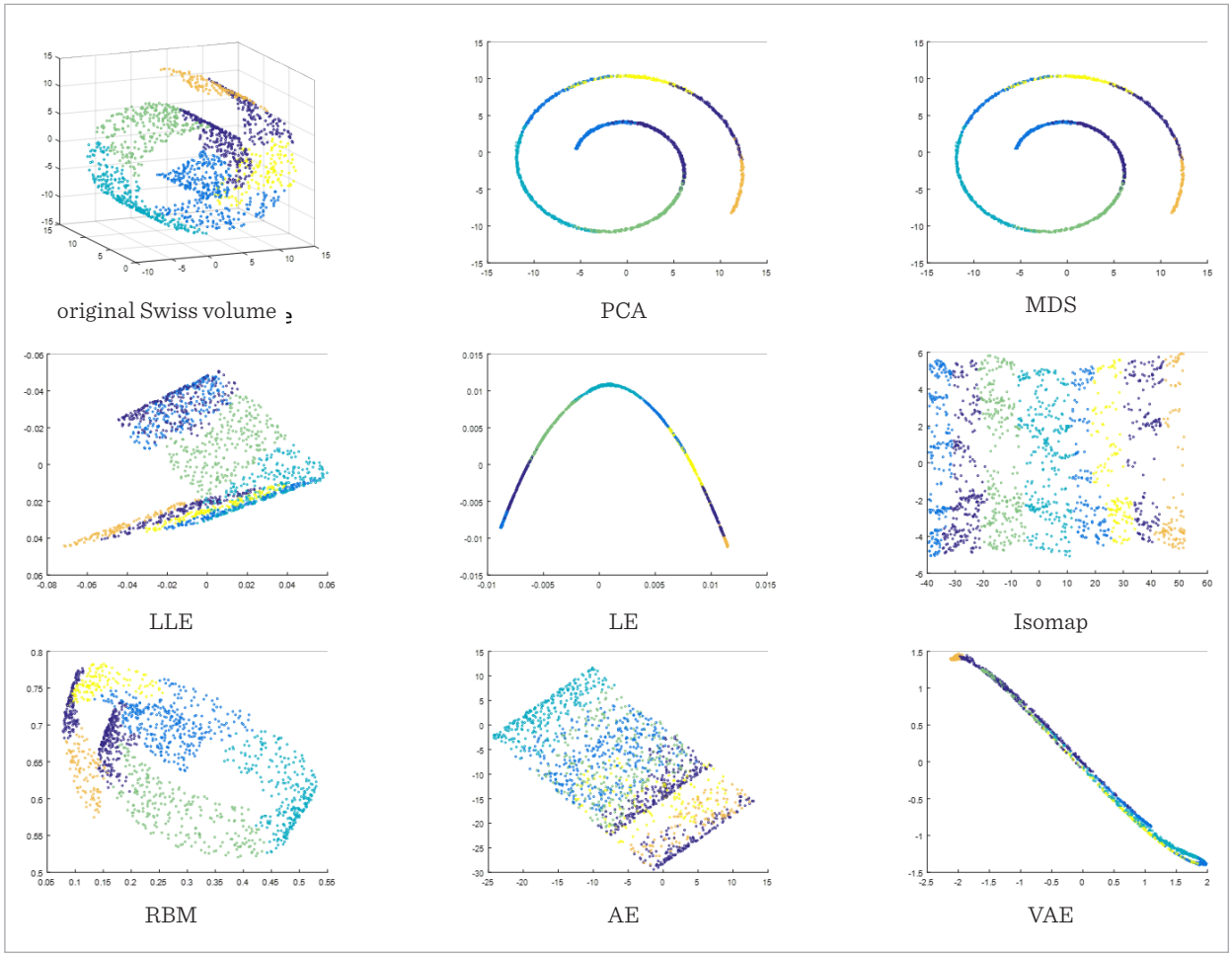
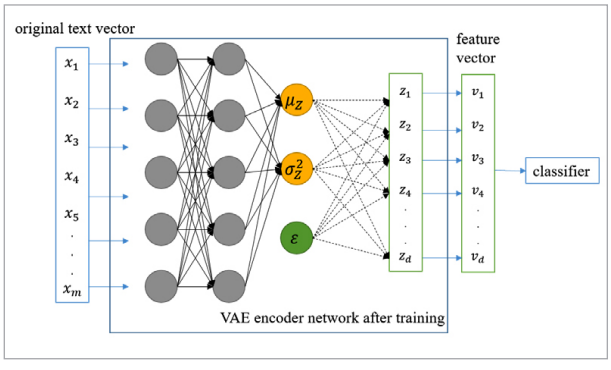


Figure 6
VAE dimensionality reduction classification process. After reducing the dimension using VAE, the low-dimensional feature vector (hidden layer) is used as the input of the classifier, and the classification result is used to measure the dimensionality reduction effect of the VAE



its basic unit is decision tree. The essence of RF is an extension of Bagging that each decision tree is a classifier and for an input sample [3, 11], N trees will have N classification results. RF integrates all classification voting results and specifies the category with the most votes as the final output [49].

For kNN classification algorithm, k is set as 5 and the number of decision trees in the RF is set to 30. By comparing the classification accuracy, it is proved that most of the original features will be preserved after dimensionality reduction by VAE. Figure 6 shows VAE dimensionality reduction process.

4.3. Evaluation Metric

The datasets are subjected to different dimensionality reduction methods to obtain new low-dimension-

al datasets. The performance of these new datasets are compared by inputting kNN, RF and SVM. Since the evaluation of the classification results is very important, this paper uses the evaluation method as described below. There are positive and negative samples in the binary classification problem [45].

In order to verify the feature dimensionality reduction performance of different methods, accuracy, precision, recall and F1-score are used as evaluation indicators.

By using ten cross tests, ten classification results can be obtained at last. So that, the classification accuracy shown at last is composed of average value and standard deviation (*mean*% ± *std*%), and the best results are highlighted in bold.

4.4. Sensitivity of the Parameters

In VAE, the effect of dimensionality reduction and the performance of the final classification effect will be affected by different parameter settings. In the following pages, some parameters of the VAE model in Algorithm 1 are changed to analyze the dimensionality reduction effect and classification results of all datasets. As can be seen in Figure 7, although the global losses of the different datasets is different, the global losses of all datasets begins to converge after about 20 iterations. It can be seen from the experiment in Figure 7 that the text data converges rapidly. Therefore, in the following experiment, the number of iterations for training the VAE model is fixed at 20, and DBWorld dataset uses full batch size and the other datasets use minibatch that the small batch data are set to 200.

The implementation of VAE network in this paper is based on a multi-layer symmetric AE network, the effect of encoder hidden layers' neurons and decoder hidden layers' neurons on accuracy need to be studied. For Algorithm 1, the m_1 is set as {100,150,200,250,300, 350,400,450,500}, the m_2 is set as {100,150,200,250, 300,350,400,450,500}, and set the hidden layer dimension z to 100. Figure 8 shows the relationship among classification accuracy, encoder hidden layers' neu-

Figure 7

Relationship among global loss, number of iterations and hidden dimension. The global losses of all datasets begin to stabilize after approximately 20 iterations

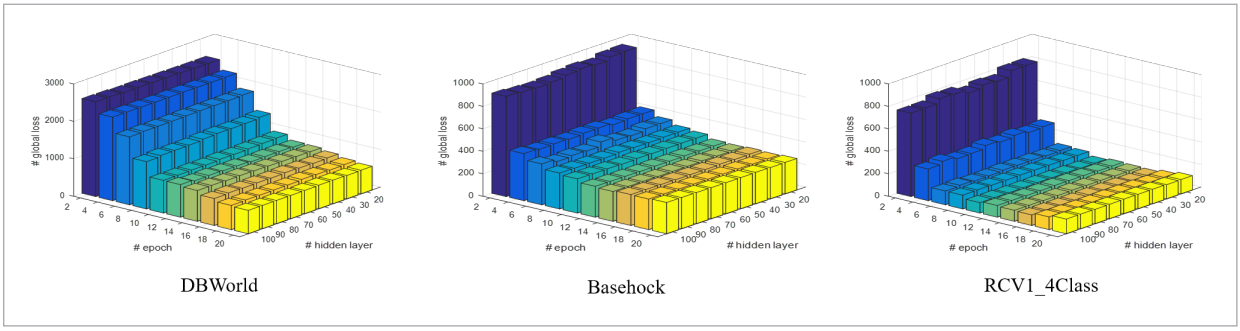


Figure 8

Relationship among classification accuracy, encoder hidden layers' neurons and decoder hidden layers' neurons

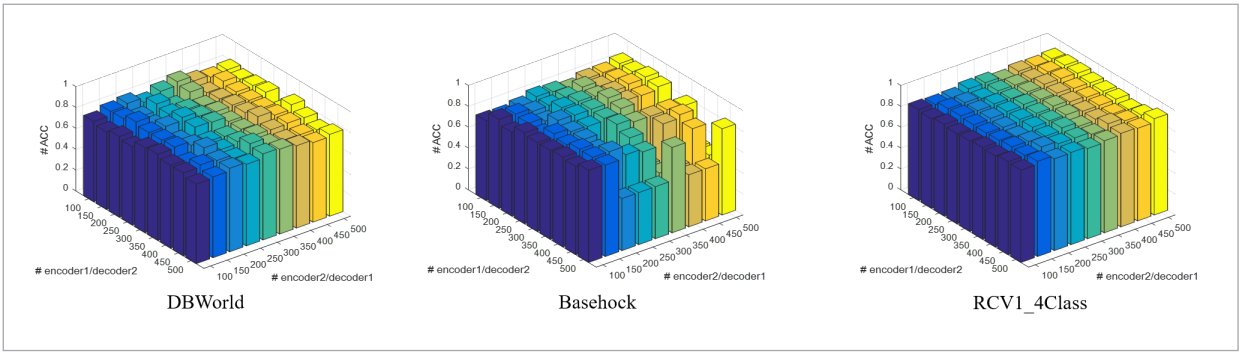
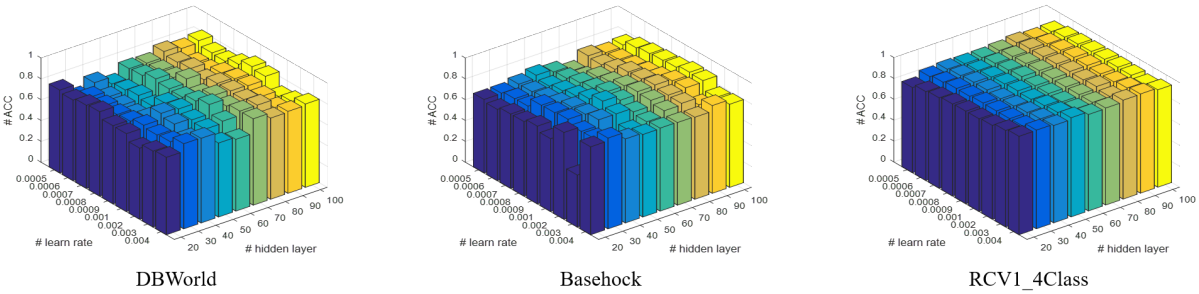


Figure 9
Relationship among classification accuracy, number of hidden layers and learning rate



rons and decoder hidden layers' neurons. It can be seen from Table 3 that different datasets have different sensitivity to network structure. This may be due to the size of the datasets, infer from Tables 2 and Table 3, it can be concluded that for more dimensional data, more neurons are needed to extract features better.

Table 3
Parameter under the highest accuracy

Database	Encoder1	Encoder2	Decoder1	Decoder2
DBWorld	100	350	350	100
Basehock	250	350	350	250
RCV1_4Class	500	500	500	500

Then, in order to continue learning the effect of learning rate α on accuracy refer to Table 3 and the α is set as {0.0005,0.0006,..., 0.0009,0.001,0.002, 0.003,0.004} and the hidden layer dimension is set to {20,30,40,50,60,70,80,90,100}. Figure 9 shows the relationship among classification accuracy, number of hidden layers and learn rate. It can be seen from Figure 9 that different datasets have different sensitivity to learning rate. The classification accuracy of DBWorld and Basehock fluctuate randomly with the change of learning rate, while the classification accuracy of RCV1_4Class is hardly affected by learning rate.

4.5. Experimental Results

Table 4 shows the classification effect of all different dimensional reduction algorithms on different datasets by *k*NN classifier. Tables 5 and 6 show the classification effect by SVM and RF, two categories of DBWorld and Basehock datasets, and four categories of RCV1_4Class dataset obtained the highest classification accuracy in the three classification algorithms. In *k*NN classification algorithm, the classification results of accuracy obtained according to the above datasets order are 90.75%, 94.76% and 95.12%. In SVM classification algorithm, the classification results of accuracy are 92.86%, 97.55%, and 95.63%. In RF classification algorithm, the classification results of accuracy are 89.19%, 97.34%, and 95.48%. In particular, in the RCV1_4Class dataset, VAE obtained the accuracy rates of 95.12%, 95.63% and 95.48% in the three classifiers respectively, which are much higher than the classification result of other dimension reduction methods. In the following experiments, three other evaluation indicators of the classification algorithm are also tested, it can be found that the dimensionality reduction of the text data has little deviation in the four evaluations.

The results of the classification experiments show that VAE performs better than other dimensionality reduction methods in sparse and discrete datasets. VAE performed well in both the two and four categories. In the DBWorld dataset, even though there are only 64 datasets, VAE still achieves the highest results in the case of less training data, and VAE has achieved a greater improvement than AE. In contrast, in Basehock and RCV1_4Class datasets with high data dimensions and large data volumes, VAE can also achieve the best classification results.

The text features are transformed from the original high-dimensional and high-sparse vector into the low-dimensional dense vector. VAE conduct dimensionality reduction on the text data, through the clas-

Table 4
Classification results of different datasets by *k*NN

Evaluation metric	Accuracy			Recall		
method/dateset	DBWorld	Basehock	RCV1_4Class	DBWorld	Basehock	RCV1_4Class
PCA	90.54±11.11	83.67±2.33	91.97±1.19	93.11±7.94	89.08±1.60	93.43±0.97
MDS	89.29±8.83	84.08±2.96	91.88±0.74	92.28±7.48	89.35±1.99	93.39±0.61
LLE	73.21±17.00	84.12±1.47	81.14±1.43	80.94±10.9	89.39±0.98	84.64±1.19
LE	66.43±17.17	91.06±1.52	88.39±1.16	70.56±7.31	94.02±1.07	90.57±0.99
Isomap	64.46±19.48	85.57±2.31	89.09±0.87	71.94±10.67	90.39±1.43	91.10±0.75
RBM	65.36±21.08	80.93±2.93	55.19±1.75	71.39±11.72	87.22±1.89	64.06±1.45
AE	63.57±15.50	89.81±3.13	82.16±1.14	69.28±7.05	93.21±2.11	85.50±0.95
VAE	**90.75±11.23**	**94.76±1.27**	**95.12±0.72**	**93.24±9.57**	**96.48±0.84**	**95.71±0.66**
Evaluation metric	Precision			F1-Score		
PCA	94.33±7.30	89.14±1.52	93.41±0.98	91.84±9.85	89.02±1.55	93.40±0.97
MDS	91.83±7.54	89.46±1.86	93.31±0.63	91.15±7.58	89.30±2.00	93.32±0.62
LLE	84.35±13.69	89.39±0.97	84.79±1.17	76.31±14.78	89.32±0.96	84.67±1.18
LE	62.79±16.00	94.00±1.03	90.50±0.89	62.59±10.8	93.99±1.03	90.50±0.95
Isomap	62.54±20.61	90.71±1.53	91.12±0.75	63.28±15.61	90.27±1.54	91.05±0.76
RBM	65.87±18.37	87.51±1.92	63.96±1.45	63.71±15.03	87.14±1.98	63.76±1.43
AE	62.24±17.11	93.17±2.13	85.56±1.03	61.55±10.91	93.15±2.10	85.50±0.99
VAE	**94.98±9.54**	**96.5±0.86**	**95.95±0.63**	**94.10±9.56**	**96.48±0.85**	**95.81±0.64**

Table 5
Classification results of different datasets by SVM

Evaluation metric	Accuracy			Recall	
method/dateset	DBWorld	Basehock	RCV1_4Class	DBWorld	RCV1_4Class
PCA	90.46±7.17	81.43±2.23	85.78±0.90	90.28±4.97	88.12±0.76
MDS	90.54±8.83	86.37±1.54	85.78±1.26	91.28±10.48	88.15±0.96
LLE	86.61±10.25	89.07±2.23	74.26±1.24	88.33±11.40	79.19±0.92
LE	84.11±15.58	94.81±13.80	87.49±1.02	88.83±11.49	89.74±0.76
Isomap	75.89±13.58	89.96±2.39	89.36±1.16	79.39±13.26	91.26±0.97
RBM	78.39±19.52	81.38±2.22	46.31±1.88	83.33±14.64	55.49±1.52
AE	86.43±14.36	96.55±1.38	88.89±1.06	90.33±10.64	90.83±0.88
VAE	**92.86±12.98**	**97.55±0.93**	**95.63±0.57**	**92.00±10.43**	**96.32±0.50**
Evaluation metric	Precision			F1-Score	
PCA	90.11±5.11	87.70±1.34	88.38±0.83	90.53±5.77	88.18±0.79
MDS	90.77±13.1	90.87±1.03	88.34±0.94	89.83±11.95	88.19±0.96
LLE	86.72±12.39	92.73±1.57	79.84±0.91	86.31±11.69	79.21±0.93
LE	86.67±13.66	96.54±0.92	89.95±0.83	84.65±14.54	89.80±0.80
Isomap	78.94±16.24	93.27±1.59	91.31±0.97	76.84±14.04	91.26±0.97
RBM	84.89±15.27	87.53±1.50	56.67±1.98	82.63±14.84	55.40±1.68
AE	90.89±10.03	97.66±0.99	91.07±0.86	88.89±11.86	90.93±0.88
VAE	**90.91±10.53**	**98.34±0.60**	**96.26±0.50**	**91.45±10.66**	**96.28±0.50**

Table 6
Classification results of different datasets by RF

Evaluation metric	Accuracy			Recall		
method/dateset	DBWorld	Basehock	RCV1_4Class	DBWorld	Basehock	RCV1_4Class
PCA	86.19±18.16	86.80±1.52	92.64±0.48	86.25±18.43	86.77±1.47	92.44±0.53
MDS	84.52±17.51	86.35±1.29	92.90±0.79	84.58±17.79	86.30±1.23	92.75±0.84
LLE	83.33±16.19	85.90±2.78	83.76±1.25	79.00±21.00	85.91±2.80	83.35±1.32
LE	83.10±18.76	93.18±1.66	91.08±0.67	83.75±17.84	93.14±1.57	90.77±0.72
Isomap	78.10±18.77	85.35±2.31	91.51±0.72	78.75±18.68	85.41±2.32	91.25±0.75
RBM	73.57±20.80	76.11±2.94	31.44±1.15	73.17±20.78	76.14±2.92	26.43±0.55
AE	65.95±17.50	91.97±1.95	78.77±1.34	63.17±16.38	92.01±1.88	78.05±1.32
VAE	**89.19±17.84**	**97.34±0.95**	**95.48±0.58**	**86.83±18.34**	**97.34±0.92**	**95.14±0.67**
Evaluation metric	Precision			F1-Score		
PCA	88.00±16.99	86.83±1.50	92.52±0.43	85.17±19.51	86.76±1.50	92.47±0.49
MDS	86.75±16.46	86.43±1.24	92.76±0.72	83.45±18.80	86.29±1.28	92.74±0.78
LLE	81.92±19.00	85.88±2.78	84.72±1.03	78.21±20.87	85.87±2.79	83.63±1.28
LE	86.58±16.99	93.25±1.74	90.99±0.62	82.19±19.58	93.15±1.66	90.85±0.67
Isomap	80.9±18.44	85.42±2.41	91.45±0.71	77.40±19.33	85.32±2.32	91.32±0.73
RBM	69.92±29.46	76.30±3.02	37.82±6.97	69.17±25.61	76.03±2.91	15.94±0.74
AE	64.83±20.75	92.08±2.06	79.13±1.20	60.28±18.50	91.95±1.95	78.38±1.31
VAE	**89.08±17.67**	**97.37±0.96**	**95.27±0.58**	**87.94±18.53**	**97.33±0.95**	**95.19±0.63**

sification experiment can be concluded that VAE can well extract the semantic information in the text. Infer from this method of text dimensionality reduction greatly reduces the time and space cost of subsequent classification experiments.

In summary, VAE is feasible for dimensionality reduction of text data.

5. Conclusions and Future Work

In this paper, some datasets perform unsupervised dimensionality reduction by using VAE. Without labels, VAE learned the low-dimensional representation from the high-dimensional text data. Through several comparison experiments, VAE still performs well in dimensionality reduction of text datasets. The classification accuracy of VAE on different data sets is improved by at least 0.21% and at most 3.7%. Under the appropriate number of iterations and parameters, the ability of VAE to reconstruct text data and reduce feature dimension has good effect. However, in data classification, the elements in the sample are assumed to be independent of each other. How to improve classification standards has not been further explored. In fact, the classification effect is not ideal for samples with only reduced dimensions. In the future work, VAE will adapt to more datasets in the dimensionality reduction and have better performance in text classification.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. 61877010, 11501114, 61906043, and 11901100), Fujian Natural Science Funds (No. 2019J01243), Funds of Education Department of Fujian Province (No. JAT190026) and Fuzhou University (Nos. 510872/GXRC-20016, 510930/XRC-20060, 510730/XRC-18075, 510809/GXRC-19037, 510649/XRC-18049 and 510650/XRC-18050).

References

1. Belkin, M., Niyogi, P. Laplacian Eigenmaps for Dimen-sionality Reduction and Data Representation. Neural Com-putation, 2014, 15(6), 1373-1396. https://doi.org/10.1162/089976603321780317
2. Bourlard, H., Kamp, Y. Auto-association by Multilayer Perceptrons and Singular Value Decomposition. Biological Cybernetics, 1988, 59(4-5), 291-294. https://doi.org/10.1007/BF00332918
3. Bengio, Y., Courville, A. A., Vincent, P. Representation Learning: A Review and New Perspectives. IEEE Transac-tions on Pattern Analysis & Machine Intelligence, 2013, 35(8), 1798-1828. https://doi.org/10.1109/TPAMI.2013.50
4. Breiman, L. Bagging Predictors. Machine Learning, 1996, 24(2), 123-140. https://doi.org/10.1007/BF00058655
5. Cai, D., He, X., Zhang, W. V., Han, J. Regularized Locality Preserving Indexing via Spectral Regression. Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management (CIKM'07), 2007, 741-750. https://doi.org/10.1145/1321440.1321544
6. Cappella, J. N. Vectors Into the Future of Mass and Inter-personal Communication Research: Big Data, Social Media, and Computational Social Science. Human Communication Research, 2017, 43(4), 545. https://doi.org/10.1111/hcre.12114
7. Chen, L., Buja, A. Local Multidimensional Scaling for Non-linear Dimension Reduction, Graph Drawing, and Proximi-ty Analysis. Publications of the American Statistical Asso-ciation, 2009, 104(485), 209-219. https://doi.org/10.1198/jasa.2009.0111
8. Chen, R. C., Liang, J., Pan, R. Using Recursive ART Net-work to Construction Domain Ontology Based on Term Frequency and Inverse Document Frequency. Expert Sys-tems with Application, 2008, 34(1), 488-501. https://doi.org/10.1016/j.eswa.2006.09.019
9. Cox, M. A. A., Cox, T. F. Multidimensional Scaling. Journal of the Royal Statistical Society, 2001, 46(2), 1050-1057.
10. Cui, Z., Xu, B., Zhang, W., Jiang, D., Xu, J. CLDA: Feature Selection for Text Categorization Based on Constrained LDA. International Conference on Semantic Computing (ICSC 2007), 2007, 702-712.
11. Cutler, A., Cutler, D. R., Stevens, J. R. Random Forests. Machine Learning, 2004, 45(1), 157-176. https://doi.org/10.1007/978-1-4419-9326-7_5
12. Ding, Y., Ma, X., Wang, Y. Health Status Monitoring for ICU Patients Based on Locally Weighted Principal Compo-nent Analysis. Computer Methods and Programs in Bio-medicine, 2018, 156, 61-71. https://doi.org/10.1016/j.cmpb.2017.12.019
13. Feldman, R., Sanger, J. The Text Mining Handbook: Ad-vanced Approaches in Analyzing Unstructured Data, Cam-bridge University Press, 2007. https://doi.org/10.1017/CBO9780511546914
14. Fleming, M. K., Cottrell, G.W. Categorization of Faces Us-ing Unsupervised Feature Extraction. IEEE 1990 IJCNN International Joint Conference on Neural Networks, 1990, 65-70. https://doi.org/10.1109/IJCNN.1990.137696
15. Gelšvartas, J., Simutis, R., Maskeliūnas, R. User Adaptive Text Predictor for Mentally Disabled Huntington's Patients. Computational Intelligence and Neuroscience, 2016, 2016, 3054258:1-3054258:6. https://doi.org/10.1155/2016/3054258
16. He, S., Shin, D. H., Zhang, J., Chen, J., Sun, Y. Full-View Area Coverage in Camera Sensor Networks: Dimension Reduction and Nearoptimal Solutions. IEEE Transactions on Vehicular Technology, 2016, 65(9), 7448-7461. https://doi.org/10.1109/TVT.2015.2498281
17. Hinton, G. E., Salakhutdinov, R. R. Reducing the Dimen-sionality of Data with Neural Networks. Science, 2006, 313(5786), 504-507. https://doi.org/10.1126/science.1127647
18. Huang, L., Wang, L. Accelerated Monte Carlo Simulations with Restricted Boltzmann Machines. Physical Review B, 2017, 95(3), 035105. https://doi.org/10.1103/PhysRevB.95.035105
19. Kapočiūtė-Dzikienė, J., Damaševičius, R. Intrinsic Evalua-tion of Lithuanian Word Embeddings Using WordNet. Arti-ficial Intelligence and Algorithms in Intelligent Systems, 2018, 764, 394-404. https://doi.org/10.1007/978-3-319-91189-2_39
20. Kingma, D. P., Welling, M. Auto-encoding Variational Bayes. Conference Proceedings: Papers Accepted To the In-ternational Conference on Learning Representations, 2013.
21. Kowalczyk, M., Buxmann, P. Big Data and Information Processing in Organizational Decision Processes. Business & Information Systems Engineering, 2014, 6(5), 267-278. https://doi.org/10.1007/s12599-014-0341-5

22. Krizhevsky, A., Sutskever, I., Hinton, G.E. Imagenet Clas-sification with Deep Convolutional Neural Networks. Communications of the ACM, 2017, 84-90. https://doi.org/10.1145/3065386

23. Lee, H., Grosse, R., Ranganath, R., Ng, A. Y. Convolu-tional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. International Con-ference on Machine Learning, ACM International Confer-ence Proceeding Series, 2009, 382, 609-616. https://doi.org/10.1145/1553374.1553453

24. Liu, J., Li, C., Yang, W. Supervised Learning via Unsuper-visedSparseAutoencoder.IEEEAccess,2018,6,73802-73814. https://doi.org/10.1109/ACCESS.2018.2884697

25. Luo, F., Guo, W., Yu, Y., Chen, G. A Multi-Label Classifi-cation Algorithm Based on Kernel Extreme Learning Ma-chine. Neurocomputing, 2017, 260, 313-320. https://doi.org/10.1016/j.neucom.2017.04.052

26. Montúfar, G. Restricted Boltzmann Machines: Introduction and Review. CoRR, vol.abs/1806.07066, 2018.

27. Napoli, C., Tramontana, E., Sciuto, G. L., Wozniak, M., Damasevicius, R., Borowik, G. Authorship Semantical Identification Using Holomorphic Chebyshev Projectors. 2015 Asia-Pacific Conference on Computer Aided System Engineering, 2015, 232-237. https://doi.org/10.1109/APCASE.2015.48

28. Polap, D. Analysis of Skin Marks Through the Use of In-telligent Things. IEEE Access, 2019, 7, 149355-149363. https://doi.org/10.1109/ACCESS.2019.2947354

29. Roweis, S. T., Saul, L. K. Nonlinear Dimensionality Reduc-tion by Locally Linear Embedding. Science, 2000, 290(5500), 2323-2326. https://doi.org/10.1126/science.290.5500.2323

30. Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning Representations by Back-Propagating Errors. Cognitive Modeling, 1986, 323(6088), 399-421. https://doi.org/10.1038/323533a0

31. Shang, T. T., Jia, Y. C., Wen, Y., Hong, S. Polarimetric Dimensionality Reduction for SAR Image Classification. IEEE Transactions on Geoscience & Remote Sensing, 2011, 50(1), 170-179. https://doi.org/10.1109/TGRS.2011.2168532

32. Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., Wang, Z. A Novel Feature Selection Algorithm for Text Categoriza-tion. Expert Systems, 2007, 33(1), 1-5. https://doi.org/10.1016/j.eswa.2006.04.001

33. Sun, Z., Yu, Y. Sparse Coding Extreme Learning Machine for Classification. Neurocomputing, 2017, 261, 50-56. https://doi.org/10.1016/j.neucom.2016.06.078

34. Tenenbaum, J. B., Silva, D., V, Langford, J. C. A Global Geometric Framework for Nonlinear Dimensionality Re-duction. Science, 2000, 290(5500), 2319-2323. https://doi.org/10.1126/science.290.5500.2319

35. Vapnik, V., Golowich, S. E., Smola, A. J. Support Vector Method for Function Approximation, Regression Estima-tion and Signal Processing. Advances in Neural Information Processing Systems, 1997, 9, 281-287.

36. Wang, S., Guo, W. Robust Co-Clustering via dual Local Learning and High-Order Matrix Factorization. Knowledge-Based Systems, 2017, 138, 176-187. https://doi.org/10.1016/j.knosys.2017.09.033

37. Wang, S., Guo, W. Sparse Multigraph Embedding for Mul-timodal Feature Representation. IEEE Transactions on Mul-timedia, 2017, 19(7), 1454-1466. https://doi.org/10.1109/TMM.2017.2663324

38. Wang, S., Pedrycz, W., Zhu, Q., Zhu, W. Subspace Learning for Unsupervised Feature Selection via Matrix Factori-zation. Pattern Recognition, 2015, 48(1), 10-19. https://doi.org/10.1016/j.patcog.2014.08.004

39. Wang, S., Zhu, W. Sparse Graph Embedding Unsupervised Feature Selection. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 48(3), 329-341. https://doi.org/10.1109/TSMC.2016.2605132

40. Wlodarczyk-Sielicka, M., Polap, D. Automatic Classifica-tion Using Machine Learning for Non-Conventional Ves-sels on Inland Waters. Sensors, 2019, 19(14), 3051. https://doi.org/10.3390/s19143051

41. Xia, Y., Wang, J. Low-Dimensional Recurrent Neural Net-workbased Kalman Filter for Speech Enhancement. Neural Networks, 2015, 67, 131-139. https://doi.org/10.1016/j.neunet.2015.03.008

42. Xie, J., Fang, Y., Zhu, F., Wong, E. Deepshape: Deep Learned Shape Descriptor for 3d Shape Matching and Re-trieval. IEEE Conference on Computer Vision and Pattern Recognition, 2015, 1275-1283.

43. Xie, L., Liu, G., Lian, H. Deep Variational Auto-Encoder for Text Classification. Proceedings of 2019 IEEE Interna-tional Conference on Industrial Cyber Physical Systems, 2019, 737-742. https://doi.org/10.1109/ICPHYS.2019.8780129

44. Yang, X., Zhang, T., Xu, C. Cross-Domain Feature Learn-ing in Multimedia. IEEE Transactions on Mul-

timedia, 2015, 17(1), 64-78. https://doi.org/10.1109/TMM.2014.2375793

45. Yang, Y., Pedersen, J. O. A Comparative Study on Feature Selection in Text Categorization. Proceedings of the Four-teenth International Conference on Machine Learning, 1997, 412-420
46. Ye, D., Chen, Z. A New Approach to Minimum Attribute Reduction Based on Discrete Artificial Bee Colony. Soft Computing, 2015, 19(7), 1893-1903. https://doi.org/10.1007/s00500-014-1371-0
47. Zhang, H., Shang, X., Luan, H., Wang, M., Chua, T. Learning from Collective Intelligence: Feature Learning Using Social Images and Tags. ACM Transactions, 2016, 13(1), 1:1-1:23. https://doi.org/10.1145/2978656
48. Zhu, Q., Li, X., Conesa, A., Pereira, C. GRAM-CNN: A Deep Learning Approach with Local Context for Named Entity Recognition in Biomedical Text. Bioinformatics, 2018, 34(9), 1547-1554. https://doi.org/10.1093/bioinformatics/btx815
49. Zhong, S., Chen, T., He, F., Niu, Y. Fast Gaussian Kernel Learning for Classification Tasks Based on Specially Struc-tured Global Optimization. Neural Networks, 2014, 57(9), 51-62. https://doi.org/10.1016/j.neunet.2014.05.014
50. Zhou, X., Yue, H., Li, G. Text Categorization Based on Clustering Feature Selection. Procedia Computer Science, 2014, 31(31), 398-405. https://doi.org/10.1016/j.procs.2014.05.283