Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Battle damage assessment based on an improved Kullback-Leibler divergence sparse autoencoder*

Zong-feng QI[1], Qiao-qiao LIU[2], Jun WANG[2, 3], Jian-xun LI[†‡2]

([1]*State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, Luoyang 471003, China*)

([2]*MOE Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China*)

([3]*Luoyang Electronic Equipment Test Center of China, Luoyang 471000, China*)

[†]E-mail: lijx@sjtu.edu.cn

Received July 4, 2016; Revision accepted Jan. 23, 2017; Crosschecked Dec. 20, 2017

Abstract: The nodes number of the hidden layer in a deep learning network is quite difficult to determine with traditional methods. To solve this problem, an improved Kullback-Leibler divergence sparse autoencoder (KL-SAE) is proposed in this paper, which can be applied to battle damage assessment (BDA). This method can select automatically the hidden layer feature which contributes most to data reconstruction, and abandon the hidden layer feature which contributes least. Therefore, the structure of the network can be modified. In addition, the method can select automatically hidden layer feature without loss of the network prediction accuracy and increase the computation speed. Experiments on University of California-Irvine (UCI) data sets and BDA for battle damage data demonstrate that the method outperforms other reference data-driven methods. The following results can be found from this paper. First, the improved KL-SAE regression network can guarantee the prediction accuracy and increase the speed of training networks and prediction. Second, the proposed network can select automatically hidden layer effective feature and modify the structure of the network by optimizing the nodes number of the hidden layer.

Key words: Battle damage assessment; Improved Kullback-Leibler divergence sparse autoencoder; Structural optimization; Feature selection

https://doi.org/10.1631/FITEE.1601395 **CLC number:** TP391.4; E917

1 Introduction

Battle damage assessment (BDA), which provides real-time diagnoses for emergency repair on the battlefield, is a vital premise of battlefield repair (Zhang *et al.*, 2012; Li and Huang, 2014; Sun and Li, 2016). Generally, BDA makes real-time assessments of the degree of damage to the combat equipment, so the damaged parts of the combat equipment can be repaired in time based on these assessments. In this way, combat equipment can continue the scheduled combat missions. According to previous experiments, BDA is beneficial to accomplish combat missions by attaching importance to assessment and repair of damaged equipment on the battlefield. BDA research has attracted attention from almost all nations. Damaged equipment always becomes an obstacle on the battlefield and will be permanently lost if it is not repaired in time. On the other hand, damaged equipment will be used in battle again and continue to carry out missions if repairs are made in time.

Traditional BDA methods prefer to assess battle damage by constructing battle models based on multi-agent systems. However, it is generally difficult to construct accurate physical models (Ma ZJ *et al.*, 2007; Ma XM *et al.*, 2016). Recently, many achievements have been made in BDA research. Yong (2004) put the dynamic factor of a battle environment into the model based on complex adaptive system theory (Ding *et al.*, 2016). This research improved the

[‡] Corresponding author

[*] Project supported by the National Basic Research Program (973) of China (No. 61331903) and the National Natural Science Foundation of China (Nos. 61175008 and 61673265)

ORCID: Zong-feng QI, http://orcid.org/0000-0001-7031-8477

simulation platform of the intercommunication process between the equipment on the battlefield and the battle environment. Simulation results revealed the inner relationships among the damage mechanism, battle environment, and conditions. Based on data mining and reasoning techniques, battle damage was assessed synthetically by combining two kinds of useful information: the assessment of pre-strike forecasts and expert experience; the battle damage information collected on the battlefields (Ma *et al.*, 2007). Furthermore, a new approach was proposed, which effectively enhanced the accuracy of assessments and decreased the uncertainty of decisions by updating the conditional probabilities of the model dynamically. Chen *et al.* (2011) predicted the level of battle damage based on the fuzzy theory and a Bayesian method. Li and Huang (2014) analyzed the indexes and variables of the damage using a Bayesian network. These studies revealed that data-driven methods, instead of complex battle damage models, are playing increasingly important roles in BDA research.

For decades, researchers have successfully applied different learning theories to pattern classification and regression problems (Hastie *et al.*, 2009). Artificial neural network (ANN), shallow machine learning, and deep learning are methods that can make predictions. The main idea of these methods is to train the learning networks using data samples, so that the networks can extract and mine useful information from a large dataset. Before 2006, machine learning algorithms were always based on shallow learning networks which contained no more than one hidden layer, such as linear regression, logistic regression, Bayesian networks, support vector machines, and decision trees (Hosmer and Lemeshow, 2005; Jensen and Nielsen, 2007; Vens *et al.*, 2008; Seber and Lee, 2012; Qin *et al.*, 2014). Because it lacks hidden layers, shallow machine learning does not always perform well in extracting high-level features, leading to unsatisfactory classification or regression results. In 2006, deep learning led pattern classification techniques to a new platform in learning procedures (Zhao *et al.*, 2015). Generally, deep learning methods include convolution neural networks (CNNs) and stacked autoencoders (Vincent *et al.*, 2010; Wen *et al.*, 2016). A novel two-level hierarchical feature-learning framework has been proposed based on deep CNNs, which is simple but effective (Song *et al.*, 2016). Currently, researchers focus on designing machine learning algorithms that can extract more useful features from raw data in the field of deep learning. These algorithms, such as the restricted Boltzmann machine, can be exploited as a fundamental procedure that extracts one level of features, and then the features are used as the input for the next level. These algorithms produce non-linear representations of features, which can be stacked and then form a deep structure. Actually, the deeper the network, the more abstract the features (Rifai *et al.*, 2011). For machine learning, learning principles are crucial in guiding the learning of intermediate representations. Positive learning principles should extract as much useful information as possible from the training samples. The de-noising autoencoder (DAE) and contractive autoencoder (CAE) were developed based on a normal autoencoder (Vincent *et al.*, 2010; Rifai *et al.*, 2011). DAE and CAE extract robust features using different learning principles. DAE prefers to encourage robust reconstructed data, while CAE encourages the robustness of intermediate representations explicitly. The sparse autoencoders (Jiang *et al.*, 2015) with different penalties such as the L1 norm, L2 norm, Student-*t*, and Kullback-Leibler divergence (KL-divergence), can obtain sparse representations of the input data, which improves the prediction performance compared to a normal autoencoder.

The autoencoder regularized with an L2 norm tends to learn fewer sparse features and prefers to avoid overfitting. The KL-divergence autoencoder can ensure its performance by learning sparse features. In this study, the proposed method assesses battle damage using a novel KL-divergence sparse autoencoder (KL-SAE). A complex mathematical expression in high dimensionality is employed to reveal the statistical characteristics of the imagery (Ding *et al.*, 2016). In complex problems, nonlinear feature extraction approaches are superior to linear ones. In contrast to most existing methods which use linear transformations to achieve feature extraction, a nonlinear feature extraction approach is applied to the proposed method.

2 Tolerance fluctuation analysis

Deep learning networks including autoencoders have been successfully applied in artificial intelligence for image, language, and text recognition. However, deep learning networks are always regarded as a 'black box', which means that only the input and output layers have explicit physical meanings, and the hidden layers are not interpretable. Moreover, the relationship between the layers is not explicable. In this study, we use a sparse autoencoder to select the features of the hidden layer through unsupervised learning. By carrying out a feature selection, it can be explained clearly which hidden features make a greater contribution to the values that we want to evaluate or assess.

KL-SAE is an expansion of a normal autoencoder. It includes two processes: encoding and decoding (Fig. 1).

If the input data is x and the activation value of the jth unit in the hidden layer of KL-SAE is $h_j(x)$, $\hat{\rho}_j$ is defined as

$$\hat{\rho}_j = \frac{1}{K}\sum_{t=1}^{m} h_j(x^{(t)}), \tag{1}$$

where $\hat{\rho}_j$ means the average activation value of the jth unit in the hidden layer, K is the number of samples, m is the number of hidden layer neurons, and ρ is a constant called 'sparse index', which is a predetermined value. The value of ρ is between 0 and 1. Generally, ρ is about 0.1, which limits $\hat{\rho}_j$ of the training data to about 0.1. If limiting $\hat{\rho}_j=\rho$, the penalty term of the KL-divergence will be added to the optimization function. The mathematical expression of the KL-divergence is

$$\sum_{j=1}^{s_2}\mathrm{KL}(\rho \,\|\, \hat{\rho}_j) = \sum_{j=1}^{s_2}\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}, \tag{2}$$

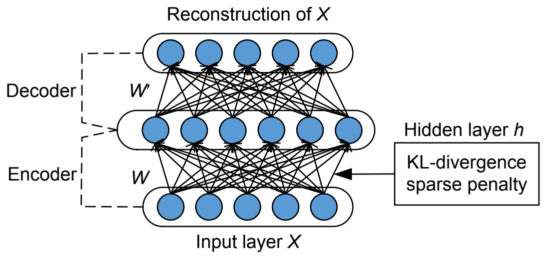

Fig. 1 Kullback-Leibler divergence sparse autoencoder model

where s_2 is the number of units in the hidden layer. The subscript j represents the jth unit in the hidden layer. Namely,

$$\mathrm{KL}(\rho \,\|\, \hat{\rho}_j) = \rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}. \tag{3}$$

The KL-divergence function has the following properties: if $\hat{\rho}_j=\rho$, $\mathrm{KL}(\rho \,\|\, \hat{\rho}_j)$ equals 0; the KL-divergence value increases with the increase of the difference between $\hat{\rho}_j$ and ρ.

We set ρ=0.2, 0.5, and 0.8, and the variation tendency of $\mathrm{KL}(\rho \,\|\, \hat{\rho}_j)$ along with $\hat{\rho}_j$ is shown in Fig. 2.

It is shown that $\mathrm{KL}(\rho \,\|\, \hat{\rho}_j)$ achieves the minimum value 0 when $\hat{\rho}_j=\rho$, and tends to be ∞ when $\hat{\rho}_j$ is close to 0 or 1. Thus, minimizing $\mathrm{KL}(\rho \,\|\, \hat{\rho}_j)$ will make the average activation value of each hidden unit close to a constant.

Based on the reconstruction error penalty of the normal autoencoder, the whole optimization function of the KL-SAE is

$$J_{\text{KL-sparese}}(\theta) = \frac{1}{N}\sum_{x^{(t)}} L(x^{(t)}, \hat{x}^{(t)}) + \frac{\lambda}{N}\sum_{j}\mathrm{KL}(\rho \,\|\, \hat{\rho}_j), \tag{4}$$

where $L(x^{(t)}, \hat{x}^{(t)})$ is the reconstruction error, N is the number of samples, and λ is the penalty parameter.

3 Improved KL-divergence sparse autoencoder

3.1 Local receptive field

ANNs are inspired by neuroscience. The networks attempt to imitate the activities of the multilayer neurons in the sensory cortex. Due to the fact that the connections of two neuron layers are adjacent, optimizing the ANN is time-consuming. At the same time, a full connection does not accord with the local connections of the biological neurons. According to research of the visual cortex cells of cats, the concept of a receptive field was first proposed by Hubel and Wiesel (1962). The brain of an animal contains large quantities of neurons. The spatial relationships among these neurons are local. When perceiving one image from the outer world, a local receptive field is used. One neuron does not perceive all the information of an image but a local field of the image. Namely, only a few sensory cortex neurons are activated in a local field of the image, while most of the sensory cortex neurons are inactivated.

Because of the KL-divergence sparse penalty, the values of most of the hidden neurons for KL-SAE are close to 0. If those neurons are ignored, they are similar to the inactivated sensory cortex neurons.

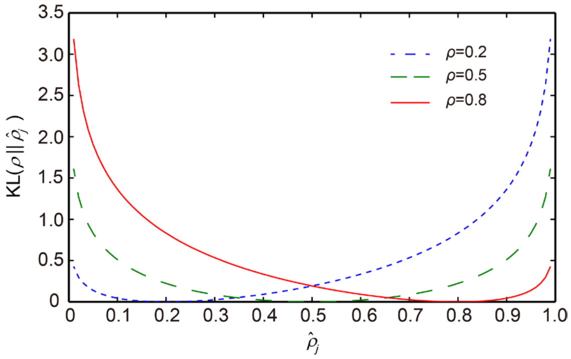

Fig. 2 Relationship between KL($\rho\|\hat{\rho}_j$) and $\hat{\rho}_j$ with ρ=0.2, 0.5, and 0.8

3.2 Improved KL-SAE regression model

According to the above analysis, a new regression method using the improved KL-SAE is proposed. The improved KL-SAE regression model is an unsupervised learning model, which gains hidden high-level feature representations. The regression layer attached to the KL-SAE is a supervised learning method that estimates the output values. Fig. 3 shows the improved KL-SAE regression model. The hidden representation of the KL-SAE will be regarded as the input for the regression layer. The hidden layer of the KL-SAE is a high-level feature representation of raw input data that can be used as the pre-extracted features for other machine learning algorithms. The regression layer outputs final estimation values of each input.

3.3 Algorithms of the improved KL-SAE regression model

First, the KL-SAE (in the lower box in Fig. 3) will be trained with an unsupervised learning method. The hollow neuron in Fig. 3 means that the improved KL-SAE regression model can extract sparse features. Then the sparse features in the hidden layer will be used as the input to the regression layer to do supervised training.

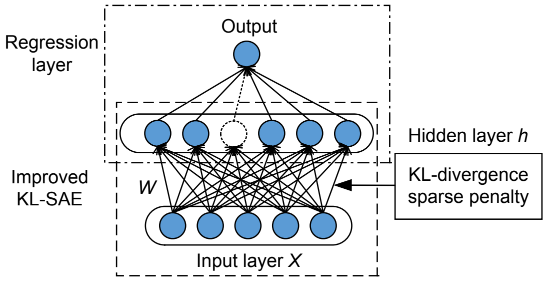

Fig. 3 Improved KL-SAE regression model

KL-SAE: Kullback-Leibler divergence sparse autoencoder. References to color refer to the online version of this figure

The algorithm of traditional KL-SAE is shown in Algorithm 1.

Algorithm 1 Traditional KL-SAE

```
1  Set the value of sparse constant $\rho$ ($0<\rho<1$);
2  Set the number of input neurons for KL-SAE equal to
   the dimension of one sample as $n$;
3  Set the number of neurons of the hidden layer $m$
   greater or less than or equal to the input neurons $n$;
4  Input the samples into KL-SAE and train KL-SAE;
5  Change the values of $\rho$ and $m$, repeat steps 1–5, and a
   different KL-SAE model will be obtained.
```

Based on the traditional KL-SAE, an algorithm of the improved KL-SAE regression model is summarized in Algorithm 2.

Algorithm 2 Improved KL-SAE

```
1  Set the value of the sparse constant $\rho$ ($0<\rho<1$);
2  Set the number of input neurons for KL-SAE equal to
   the dimension of one sample as $n$;
3  Set the number of neurons of the hidden layer $m$
   greater or less than or equal to the input neurons $n$;
4  Input the samples into KL-SAE and train KL-SAE;
5  Obtain the hidden features of all samples and count the
   number of neurons whose values are less than $\rho$ in each
   dimension of the hidden features as count$_i$ ($i$=1, 2, ...,
   $m$);
6  Compare the values of count$_i$ with each other to find
   the $k$ maximum values ($k$<<$m$);
7  Ignore the $k$-dimensional features and keep the rest of
   the hidden features;
8  Obtain the trained improved KL-SAE model;
9  Change the values of $\rho$ and $m$, repeat steps 1–8, and
   different improved KL-SAE models will be obtained.
```

Based on the improved KL-SAE, an algorithm of the regression model is summarized in Algorithm 3.

Algorithm 3 Improved KL-SAE regression model

```
Set the value of the sparse constant $\rho$ ($0<\rho<1$);
Set the number of input neurons for KL-SAE equal to the dimension of one sample as $n$;
Set the number of neurons of the hidden layer $m$ greater or less than or equal to the input neurons $n$;
Input the samples into KL-SAE and train KL-SAE;
Obtain the hidden features of all samples and count the number of neurons whose values are less than $\rho$ in each dimension of the hidden features as $\text{count}_i$ ($i$=1, 2, ..., $m$);
Compare the values of $\text{count}_i$ with each other to find the $k$ maximum values ($k$<<$m$);
Ignore the $k$-dimensional features and keep the rest of the hidden features;
Train the regression layer using the features obtained from step 7 and obtain a completely trained regression model;
Change the values of $\rho$ and $m$, repeat steps 1–8, and different improved KL-SAE models will be obtained.
```

In general, the initial value of k is 1 in step 6 of Algorithm 3, and it is increased by 1 the next time. When k=1, only one dimension of the hidden features will be abandoned. Since the ignored feature is not an important feature, the prediction performance will not be weakened without it. The rest of the features are main features that can guarantee the prediction performance and speed up the training observably. Thus, the improved KL-SAE regression model has the capability of automatically selecting the main hidden features to optimize the structure of the network.

Additionally, when ρ remains constant, the more features it abandons, the faster the model will be trained along with the increase of k. However, the larger k is, the more information will be lost, which will decrease the prediction performance of the network. The proper value of k will be found by a few rounds of experiments. This value can guarantee the performance.

When k remains constant, the less sparse the constant ρ is, the lower the average activation of the hidden neurons will be. This situation will lead to more zeros of hidden neuron values, which means sparser features are extracted. The proper value of ρ can be found by a few rounds of experiments to extract the hidden features that are appropriately sparse. The learned regression model will be verified by testing the data sets.

3.4 Benchmark data sets

Three benchmark data sets from the University of California-Irvine (UCI) machine learning repository are used to verify the proposed improved KL-SAE regression model. The attribute information of the UCI data sets is shown in Table 1. The results of the experiments using the UCI data sets to compare different regression models, support vector regression (SVR), KL-SAE+SVR, and the improved KL-SAE+SVR, are shown in Table 2. The KL-SAE regression model is not always better than SVR. The improved KL-SAE regression model steadily performs better than the other two regression models.

Remark 1 (Computational complexity) The computational complexity of the KL-SAE+SVR and the improved KL-SAE+SVR is almost the same. However, compared to SVR, the complexity of the proposed method is slightly increased, due to the introduction of the KL-SAE regression model.

Furthermore, due to the increase of algorithm complexity, the algorithm performance has been improved, including the training time and prediction time per sample.

Remark 2 (Convergence property) Compared to SVR, the KL-SAE+SVR and improved KL-SAE+SVR have a better convergence property, which can be demonstrated in the smaller feature dimension and the shorter training time.

The size of the training sample set is also an important factor. When the sample set is large enough, like combined cycle power plant (CCPP) which has 4784 samples, the convergence property is better (smaller root mean square error (RMSE) and mean absolute error (MAE)).

4 BDA based on the improved KL-SAE regression model

4.1 Description of BDA

An effective BDA is a vital step in battle tactics and command. It is defined as a real-time and precise assessment of target-equipment damage evaluation after the equipment being struck by military forces. The BDA research can be divided into three stages (Cao and Zhang, 2014). Before the 20th century, BDA research relied almost exclusively on real experiments under firepower for adjustment purposes, which was expensive. Along with the increasing scale of firepower, the costs of BDA studies also increased. In the second stage, BDA system modeling based on the theory of complex adaptive systems (Yong, 2004) was applied to BDA research. However, it was not easy to construct an accurate mathematical model and solve the model. It can even be impossible to do this. To avoid constructing a complex physical model, the third stage of BDA research is based on data collected from a real battlefield. These methods use information theory and data mining to assess the grades of damage. The effect of the damage on the target's function includes five grades in total: completely destroyed, severe damage, moderate damage, light damage, and no damage (Li and Huang, 2014). This method becomes just a qualitative classification problem to assess the category of battle damage, which cannot obtain a real value of the battle damage due to low precision in BDA. In this study, a novel regression approach is proposed to quantitatively calculate the exact value of the battle damage based on the improved KL-SAE regression model, which can optimize the structure of the network through feature selection.

Table 1 Attribute information of the UCI data sets

Data set	Number of features	Number of samples	Maximum	Minimum
Forest fire	12	517	1090.84	0
CCPP	4	9568	1	0
Servo	4	168	1	0

UCI: University of California-Irvine; CCPP: Combined cycle power plant

4.2 BDA data source

The data set for BDA in this study comes from a study that modeled battle damage based on multi-agent (Yong, 2004). In this physical model, the battle damage system is considered as a complex adaptive system. To simulate the battlefield, the following conditions of the battle damage are considered: (1) considering only the battle damage of the equipment; (2) equipment damage occurs during the battle without considering the probability of the equipment being discovered, the firing rate, or the hit rate of the enemy weapons; (3) not considering the secondary damage caused by damaged fragments; (4) not considering the safety of humans.

Based on these conditions, the model imitates a battle between red equipment and blue equipment in a complex battlefield, and investigates the battle damage of the red equipment and blue equipment. The battlefield environment is represented by a 50×50 matrix, the coordinates of which represent different terrains. Each coordinate has only a value for terrain like roads, rivers, obstacles, and so on. The combat equipment in red and in blue is on some of the coordinates of the battlefield. The red and the blue travel to their own destinations and hit each other during the whole battle. The factors that affect equipment damage include the coefficient of resistance, range of fire, battlefield environment, strategic targets, and tactics. The attribute information of the BDA data is shown in Table 3.

4.3 BDA case study

The number of data samples is 40000. The proposed BDA method in Sections 3.2 and 3.3 will be applied in this section.

Table 2 Experiment results of the UCI data sets

Data set	Regression model	Dimension of features	Training time (s)	Prediction time of one sample (μs)	RMSE	MAE
Forest fire (344 sample sets)	SVR	12	0.015	1.55×10^{-5}	30.13	15.53
	KL-SAE+SVR	11	0.006	1.47×10^{-5}	27.87	8.72
	Improved KL-SAE+SVR	11	0.006	1.14×10^{-5}	27.92	8.71
CCPP (4784 sample sets)	SVR	4	0.141	5.63×10^{-5}	0.023	0.016
	KL-SAE+SVR	3	0.3171	4.83×10^{-5}	0.036	0.027
	Improved KL-SAE+SVR	3	0.112	4.47×10^{-5}	0.022	0.013
Servo (84 sample sets)	SVR	4	0.003	3.81×10^{-6}	0.207	0.17
	KL-SAE+SVR	3	5.22×10^{-4}	3.75×10^{-6}	0.176	0.117
	Improved KL-SAE+SVR	3	2.0×10^{-4}	2.99×10^{-6}	0.216	0.122

UCI: University of California-Irvine; CCPP: Combined cycle power plant; RMSE: root mean square error; MAE: mean absolute error

Table 3 Attribute information of the BDA data

Name of attribute	Data type	Data range	Remark
Time	Int	>0	Feature
Coefficient of resistance	Int	1–10	Feature
Range of fire	Int	1–10	Feature
Environment of battlefield	Int	1–4	Feature
Strategic target			
$x_{\min}$	Int	0–50	Feature
$y_{\min}$	Int	0–50	Feature
$x_{\max}$	Int	0–50	Feature
$y_{\max}$	Int	0–50	Feature
Present coordinate x	Int	0–50	Feature
Present coordinate y	Int	0–50	Feature
Tactics	Int	0 and 1	Feature
Damage	Float	0–1	Label

BDA: battle damage assessment

First, the BDA method uses 11-dimensional features to estimate the damage value using support vector regression. In this experiment, the whole data set is divided into the training set and the testing set. Half of the data randomly selected are training samples and the rest are testing samples. The kernel function of the support vector machine is chosen as a radial basis function (RBF). The terminating condition of iteration is set as the number of iterations reaching 1000 or the training error being less than 10^{-7}. After a 10-fold cross-validation, the ideal penalty parameter C=10, the kernel function parameter γ=0.1, the geometric margin ρ=0.001, and the number of support vectors, $\mathrm{Sv}_{\mathrm{count}}$=910. The training time consumed is 2.32 s and the prediction time for one single sample is 74.26 µs. On the test set, RMSE=0.19 and MAE=0.16.

Then, an improved KL-SAE with six input neurons, six hidden neurons, and six output neurons is constructed. The sigmoid function is chosen as the activation function. Before training, the parameters (W and b) are initialized as random and sparse constants, which are set as 0.01. The activation of hidden neurons is obtained by encoding the input neurons with the sigmoid function and sparse constraint. The hidden neurons have the ability to reconstruct the input data, so the hidden features are another representation or encoding of the input data. Tables 4 and 5 show the encodings (values of hidden neurons) of 10 training samples and 10 testing samples. The hidden neurons whose values are less than the sparse constant are regarded as suppressed neurons, and those whose values are greater than the sparse constant are considered as activated neurons.

Because of the sparse penalty, there exist many suppressed hidden neurons, which are called ‘sparse features’. The results of the improved KL-SAE regression model are: RMSE=0.21 and MAE=0.18. The training time consumed is 2.25 s and the prediction time for one single sample is 82.88 µs.

The numbers of suppressed hidden neurons and activated ones for all the training samples are shown in Fig. 4. Most of the hidden neurons are suppressed. This indicates that the input data is represented by a nonlinear combination of a set of basis neurons. Moreover, fewer basis neurons can reconstruct the input data under the constraint of sparse penalty.

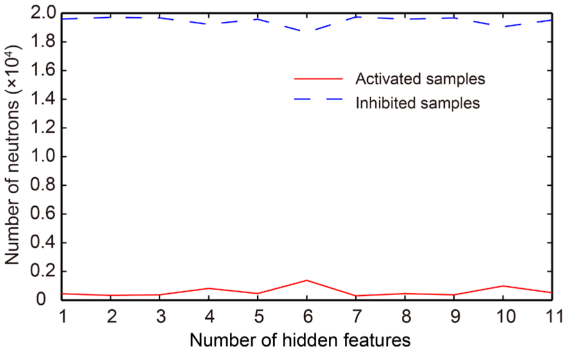

Fig. 4 The numbers of suppressed hidden neurons and activated ones of all training samples

After counting the suppressed times of each neuron, the feature that is suppressed most will be abandoned and the other features will be retained. The experiment results on the screened features are: RMSE=0.18 and MAE=0.15. The training time consumed is 2.05 s and the prediction time for one single sample is 62.86 µs. The results of the three experiments are summarized in Table 6.

All of the results of three algorithms show small values of RMSE and MAE. The first experiment uses SVR directly by using raw data to predict battle damage. The second experiment uses SVR with the sparse hidden features generated by KL-SAE to predict battle damage. This algorithm performs worse than the first algorithm in terms of RMSE and MAE. The last experiment gets rid of the features that make less contribution to reconstruction, but it does not lose precision with low-dimensional features. Namely, the

Table 4 Encodings (value of hidden neurons) of 10 training samples

Hidden neuron	Value of hidden neurons									
	1	2	3	4	5	6	7	8	9	10
1	1.7258×10^{-5}	1.0000	6.0474×10^{-6}	1.0000	1.1686×10^{-7}	1.0000	3.3350×10^{-7}	1.0000	6.4446×10^{-9}	1.0000
2	0.5049	1.0000	0.4943	1.0000	0.3956	1.0000	0.4058	1.0000	0.3138	1.0000
3	1.5825×10^{-6}	0.0036	1.5873×10^{-6}	0.0069	1.2890×10^{-6}	0.0086	1.2851×10^{-6}	0.0086	1.0436×10^{-6}	0.069
4	1.0677×10^{-35}	3.4888×10^{-44}	2.7780×10^{-35}	3.9546×10^{-44}	6.4519×10^{-34}	7.7996×10^{-44}	2.4800×10^{-34}	2.0291×10^{-43}	5.7601×10^{-6}	6.9633×10^{-43}
5	0.8382	0.9845	0.8366	0.9871	0.8224	0.9878	0.8241	0.9876	0.8091	0.9866
6	0.2104	1.0000	0.2013	1.0000	0.1360	1.0000	0.1427	1.0000	0.0942	1.0000
7	8.3635×10^{-14}	0.5474	6.6260×10^{-14}	0.9082	1.5131×10^{-14}	0.9447	1.9099×10^{-14}	0.9311	4.3613×10^{-15}	0.8311
8	0.0028	0.2924	0.0033	0.4615	0.0045	0.5504	0.0038	0.5920	0.0052	0.5880
9	8.9414×10^{-31}	2.4648×10^{-7}	6.2674×10^{-31}	3.0162×10^{-52}	8.3018×10^{-31}	5.4966×10^{-53}	1.1844×10^{-30}	3.8528×10^{-53}	1.5689×10^{-30}	1.0387×10^{-52}
10	0.6648	1.4648×10^{-7}	0.6705	3.9247×10^{-8}	0.7747	2.5738×10^{-8}	0.7702	2.6407×10^{-8}	0.8499	4.2389×10^{-8}
11	0.0013	1.1949×10^{-4}	0.0022	2.5775×10^{-4}	0.0095	4.7223×10^{-4}	0.0057	7.9729×10^{-4}	0.0247	0.0012

improved KL-SAE regression model optimizes the structure of the network while guaranteeing the prediction accuracy and clearly reduces the time and storage overhead.

In addition, different values of the sparse constant will affect the prediction results. In this experiment, the numbers of input and hidden neurons of the improved KL-SAE regression model will be confirmed and the sparse constant will vary from 0.01 to 0.1 to explore the variation trend in MRSE and MAE (Fig. 5). The sparser the hidden features are, the less time-consuming the method will be. However, if the sparse constant is too small, it will lead to a decrease in prediction accuracy.

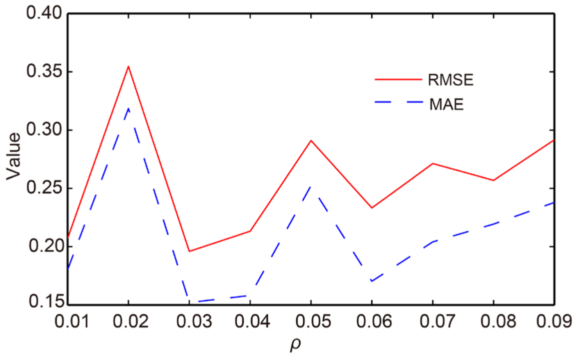

Fig. 5 Variation of MRSE and MAE with the sparse constant ρ

RMSE: root mean square error; MAE: mean absolute error

5 Conclusions

A novel regression approach is proposed based on an improved KL-SAE regression model, which remedies the disadvantages in structural optimization and feature selection of traditional deep learning networks. The improved KL-SAE regression model can extract sparse features in hidden layers through a sparse constraint, and the features obtained can be used to improve the performance of the autoencoder. The proposed algorithm based on the improved KL-SAE regression model is applied to BDA. The

Table 5 Encodings (value of hidden neurons) of 10 testing samples

Hidden neuron	Value of hidden neurons				
	1	2	3	4	5
1	1.9442×10^{-7}	1.0000	7.0279×10^{-9}	1.0000	2.8501×10^{-9}
2	0.9032	1.0059×10^{-18}	0.9830	2.8068×10^{-20}	0.9859
3	0.9744	0.0323	0.9928	0.0397	0.9952
4	0.0522	0.9827	0.1298	0.9964	0.1910
5	5.5053×10^{-18}	6.1563×10^{-13}	3.0851×10^{-19}	2.3683×10^{-13}	1.1812×10^{-19}
6	0.0868	0.8755	0.0982	0.9282	0.1076
7	1.3327×10^{-19}	1.5151×10^{-40}	2.3007×10^{-51}	6.6465×10^{-41}	6.2694×10^{-52}
8	2.2585×10^{-10}	0.9998	2.1596×10^{-10}	1.0000	3.0194×10^{-10}
9	3.6417×10^{-44}	3.3546×10^{-49}	5.3054×10^{-45}	1.9277×10^{-50}	2.2371×10^{-45}
10	0.9446	0.6586	0.9540	0.6381	0.9561
11	0.6973	0.2092	0.7216	0.1849	0.7256

Hidden neuron	Value of hidden neurons				
	6	7	8	9	10
1	1.0000	6.2710×10^{-10}	1.0000	1.1558×10^{-9}	1.0000
2	9.6547×10^{-21}	0.9966	1.1660×10^{-20}	0.9883	4.9442×10^{-20}
3	0.0552	0.9970	0.0808	0.9968	0.1237
4	0.9985	0.2036	0.9990	0.2719	0.9991
5	9.0822×10^{-14}	4.5155×10^{-20}	3.4774×10^{-14}	4.5227×10^{-20}	1.3293×10^{-14}
6	0.9443	0.1013	0.9494	0.1178	0.9461
7	2.1226×10^{-41}	1.4577×10^{-52}	5.7840×10^{-42}	1.7084×10^{-52}	1.3448×10^{-42}
8	1.0000	1.4771×10^{-10}	1.0000	4.2214×10^{-10}	1.0000
9	4.1830×10^{-51}	1.8330×10^{-45}	1.7638×10^{-51}	9.4328×10^{-46}	1.4452×10^{-51}
10	0.6388	0.9600	0.6501	0.9581	0.6717
11	0.1793	0.7408	0.1823	0.7295	0.1941

Table 6 Results of three experiments

Parameter	Value		
	SVR	KL-SAE +SVR	Improved KL-SAE +SVR
Dimension of features	11	11	10
Training time (s)	2.32	2.25	2.05
Prediction time of one sample (μs)	74.26	82.88	62.86
RMSE	0.19	0.21	0.18
MAE	0.16	0.18	0.15

improved KL-SAE regression model keeps the features that are effective and important for data reconstruction, and abandons the invalid features. According to experimental results, the improved KL-SAE regression model is superior in important feature selection and prediction when compared to the reference methods.

References

Cao, S.C., Zhang, F., 2014. Review of battle damage assessment. *Mil. Econ. Res.*, (8):53-56 (in Chinese).

Chen, X., Li, L., Liu, D., 2011. Battle damage level prediction on fuzzy theory and Bayesian method. IEEE Conf. on Robotics, Automation and Mechatronics, p.295-298. https://doi.org/10.1109/RAMECH.2011.6070499

Ding, Y., Li, N., Zhao, Y., *et al.*, 2016. Image quality assessment method based on nonlinear feature extraction in kernel space. *Front. Inform. Technol. Electron. Eng.*, **17**(10):1008-1017. https://doi.org/10.1631/FITEE.1500439

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning (2nd Ed.). Springer, New York, USA. https://doi.org/10.1007/978-0-387-84858-7

Hosmer, D.W., Lemeshow, S., 2005. Applied Logistic Regression (2nd Ed.). John Wiley & Sons, New York, USA. https://doi.org/10.1002/0471722146

Hubel, D.H., Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.*, **160**(1):106-154.

Jensen, F.V., Nielsen, T.D., 2007. Bayesian Networks and Decision Graphs. Springer, New York, USA.

https://doi.org/10.1007/978-0-387-68282-2

Jiang, N., Rong, W.G., Peng, B.L., *et al*., 2015. An empirical analysis of different sparse penalties for autoencoder in unsupervised feature learning. Int. Joint Conf. on Neural Networks, p.1-8. https://doi.org/10.1109/IJCNN.2015.7280568

Li, C.H., Huang, J., 2014. The application of Bayesian network in battle damage assessment. IEEE Int. Conf. on Software Engineering and Service Science, p.529-532. https://doi.org/10.1109/ICSESS.2014.6933622

Ma, X.M., Ding, P., Yan, W.D., 2016. Warship-damage assessment based on Bayesian networks. *Ordnance Ind. Autom*., **35**(6):72-75 (in Chinese). https://doi.org/10.7690/bgzdh.2016.06.017

Ma, Z.J., Shi, Q., Li, B., 2007. Battle damage assessment based on Bayesian network. 8th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, p.388-391. https://doi.org/10.1109/SNPD.2007.421

Qin, F.W., Li, L.Y., Gao, S.M., *et al*., 2014. A deep learning approach to the classification of 3D CAD models. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **15**(2):91-106. https://doi.org/10.1631/jzus.C1300185

Rifai, S., Vincent, P., Muller, X., *et al*., 2011. Contractive auto-encoders: explicit invariance during feature extraction. 28th Int. Conf. on Machine Learning, p.833-840.

Seber, G.A.F., Lee, A.J., 2012. Linear Regression Analysis (2nd Ed.). John Wiley & Sons, New York, USA. https://doi.org/10.1002/9780471722199

Song, G.H., Jin, X.G., Chen, G.L., *et al*., 2016. Two-level hierarchical feature learning for image classification. *Front. Inform. Technol. Electron. Eng.*, **17**(9):897-906. https://doi.org/10.1631/FITEE.1500346

Sun, G.L., Li, J., 2016. Battle damage assessment based on attribute weighted Bayesian classification. *Ship Electron. Eng*., **36**(1):29-32 (in Chinese). https://doi.org/10.3969/j.issn.1672-9730.2016.01.009

Vens, C., Struyf, J., Schietgat, L., *et al*., 2008. Decision trees for hierarchical multi-label classification. *Mach. Learn*., **73**:185-214. https://doi.org/10.1007/s10994-008-5077-3

Vincent, P., Larochelle, H., Lajoie, I., *et al*., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res*., **11**(12):3371-3408.

Wen, M.F., Hu, C., Liu, W.R., 2016. Heterogeneous multimodal object recognition method based on deep learning. *J. Cent. South Univ.* (*Sci. Technol.*), **47**(5):1580-1586 (in Chinese). https://doi.org/10.11817/j.issn.1672-7207.2016.05.018

Yong, L.Y., 2004. Modeling in Battle Damage Based on Multi-agent. MS Thesis, Harbin University of Science and Technology, Harbin, China (in Chinese).

Zhang, C., Shi, Q., Liu, T.L., *et al*., 2012. Study on battle damage level prediction using hybrid-learning algorithm. 4th Int. Conf. on Computational and Information Sciences, p.65-68. https://doi.org/10.1109/ICCIS.2012.298

Zhao, Z.Y., Li, Y.X., Yu, F., *et al*., 2015. Improved deep learning algorithm based on extreme learning machine. *Comput. Eng. Des*., **36**(4):1022-1026 (in Chinese). https://doi.org/10.16208/j.issn1000-7024.2015.04.036