



DensePILAE: a feature reuse pseudoinverse learning algorithm for deep stacked autoencoder

Jue Wang^{1,2} · Ping Guo³ · Yanjun Li⁴

Received: 15 May 2021 / Accepted: 26 August 2021 / Published online: 11 September 2021
© The Author(s) 2021

Abstract

Autoencoder has been widely used as a feature learning technique. In many works of autoencoder, the features of the original input are usually extracted layer by layer using multi-layer nonlinear mapping, and only the features of the last layer are used for classification or regression. Therefore, the features of the previous layer aren't used explicitly. The loss of information and waste of computation is obvious. In addition, faster training and reasoning speed is generally required in the Internet of Things applications. But the stacked autoencoders model is usually trained by the BP algorithm, which has the problem of slow convergence. To solve the above two problems, the paper proposes a dense connection pseudoinverse learning autoencoder (DensePILAE) from reuse perspective. Pseudoinverse learning autoencoder (PILAE) can extract features in the form of analytic solution, without multiple iterations. Therefore, the time cost can be greatly reduced. At the same time, the features of all the previous layers in stacked PILAE are combined as the input of next layer. In this way, the information of all the previous layers not only has no loss, but also can be strengthened and refined, so that better features could be learned. The experimental results in 8 data sets of different domains show that the proposed DensePILAE is effective.

Keywords Dense connection · Feature reuse · Pseudoinverse learning · Pseudoinverse learning autoencoder

Introduction

With the development of the Internet of things (IoT), people can obtain all kinds of data anytime and anywhere through various types of sensors. It lays the foundation for the application of deep learning. With the continuous growth of data volume, deep learning technology has been applied in many fields of IoT, such as smart cities [16,17], intelligent transportation [20,27], healthcare [4,15,29], and so on. Deep neural network (DNN) is formed by stacking of shallow neu-

ral network, that is, the output of the former layer is used as the input of the latter layer. Then DNN is trained by error back propagation (BP) algorithm. This paradigm has been used in a large number of applications and has achieved very good results. Deep autoencoder [9] is a typical application.

The deep autoencoder utilizes the excellent feature reconstruction capability of the autoencoder to learn features. Many variants of deep autoencoder have been proposed, such as stacked autoencoder (SAE) [12], deep denoising autoencoder [21]. In addition, it has been applied in many fields, such as remote sensing image recognition [30] and anomaly detection [5].

Since the deep autoencoder is usually trained by BP algorithm, the two notorious problems of BP algorithms, local minima and slow convergence, are present in the training process of the network. In this way, the network not only has a long training time, but also cannot obtain an optimal solution. Especially in the application of the IoT, DNN needs to run on resource-constrained devices, which often have low computing power and can not adapt to a large amount of computing. To overcome the shortcomings mentioned above, researchers have proposed many non-BP based methods, such as pseu-

✉ Ping Guo
pguo@bnu.edu.cn

Jue Wang
jwang86@foxmail.com

¹ School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

² School of Space Information, Space Engineering University, Beijing 101416, China

³ School of System Science, Beijing Normal University, Beijing 100875, China

⁴ School of Information, Shanxi University of Finance and Economics, Taiyuan 030012, China

doinverse learning algorithm (PIL) [6,22] and random vector functional link (RVFL) [18].

RVFL is only a single hidden layer feedforward neural network. In order to introduce RVFL into deep learning, dRVFL was proposed in [14]. This method can extract features quickly and get a satisfactory performance. However, its disadvantage is that the features of each layer are obtained by random projection, which is difficult to understand. PILAE [25] is proposed by combining PIL with autoencoder. It uses PIL to train autoencoder. PILAE is a unsupervised feature learning method, and could exactly learn feature. Therefore, interpretability of PILAE is more acceptable than RVFL.

However, stacked PILAE and all other deep autoencoder have information loss problems. Autoencoder is a unsupervised feature learning method by setting the input equal to the output. To avoid trivial solutions, there are usually multiple bottleneck layers, which could force the network to learn abstract compression. The narrower the layer is, the greater the compression is. As the increase of network depth, feature is becoming more and more abstract, and information loss is becoming more and more serious [31]. The learning ability of the model is affected by information loss. To strength learning ability of model, loss information should be supplied.

In deep CNN networks, feature reuse is used to reduce the information loss. ResNet [10] introduces an identity connection, which integrates linear activation and nonlinear activation. Each residual block reuses the information of the previous layer. DenseNet [13] further extends the input range of the identity connection. The output of each layer is used as the input of the subsequent layer. The previously extracted features are preserved in the later layers. The problem of loss of information is alleviated by feature reuse. Inspired by DenseNet, dense connection is introduced to PILAE. The feature reuse is used to solve the problem of information loss in autoencoder, and a new feature learning approach is proposed, namely, dense connection pseudoinverse learning autoencoder (DensePILAE). The input of each layer in DensePILAE is the concatenation of all previous layer outputs, and the new feature is learned by reconstructing all historical features. By reusing the features learned in all the previous layers, more lossless and compressed features are extracted. It could get better accuracy with fewer parameters. In this paper, we make the following contributions: (1) aiming at the problem of information loss in stacked PILAE, a dense connection PILAE method is proposed; (2) the effectiveness of DensePILAE is analyzed from the perspective of feature reuse; (3) experiments are carried out on 8 data sets, and the accuracy, the area under curve, time cost and parameter sensitivity are analyzed to verify the effectiveness of DensePILAE. The rest of the paper is organized as follows: In “Related work”, we briefly review the related works of this paper. Then we detail the basic theory of PILAE and the proposed DensePILAE in “DensePILAE”. In “Experi-

ments and discussion”, we conduct experiments and present the comparison and analysis. Finally, we give our conclusions in “Discussion”.

Related work

Feature reuse

Feature reuse is an important concept, which is proposed by Bengio in seminal paper [1]. Feature reuse can be achieved by depth of network. Through the multi-layer nonlinear transformation, the input is compressed, so the current deep network can be seen as a way of feature reuse. This is an implicit feature reuse. Another feature reuse is to directly input the output of the front non-adjacent layer into the current layer by crossing connection, which can be regarded as an explicit feature reuse. ResNet introduces the concept of residual blocks, which is essentially combination of the previous layer feature and the current layer feature. This is the reuse of the previous layer feature. DenseNet go one step further by concatenate the output of all the previous layers as input for subsequent layers. Not only the feature of the previous layer is reused, but also the features of all layers before the previous layer are reused. Therefore, the subsequent layers could make use of the knowledge learned from all previous layers. Deep layer aggregation [28] extends the way feature reuse. It does not simply concatenate the features of the previous layer or all of the previous features, but selectively reuse them. Two effective reuse methods are proposed, namely, iterative reuse and hierarchical reuse.

Similar feature reuse methods have also been used in fully connected networks. The deep stacking network (DSN) [3] reuses the predictions of all previous layers. In DSN, the reused features have lower dimensionality, whose representation ability is limited. In addition, for some simple samples, the prediction results of each layer will be more consistent, which leads to a lot of simple redundant information. Different from the reuse of DSN, DensePILAE reuses the hidden layer of autoencoder, which has larger dimension and includes richer information. The feature reuse of sequence data is studied in ResInNet [19], which is applied in the traffic prediction of Internet of things.

Non-BP based fast learning network

As a training method, BP algorithm has been widely used in the training of deep neural network and has become the most popular training method. However, there are two notorious shortcomings, local minimum and slow convergence rate, which are also widely criticized. To avoid using the BP algorithm, many network architectures are proposed, such as RVFL [18], PIL [6–8]. The weight of the network is

obtained by solving the analytical solution. The differences between PIL and RVFL are the network structure and the initialization method of weight. PIL adopts standard single layer feed-forward neural network (SLFN), while RVFL adds direct connection between input layer and output layer. For the weight between input layer and hidden layer, PIL adopts pseudoinverse or random, while RVFL adopts random value.

After years of development, many variants of PIL and RVFL have been developed, such as PILAE [25], LR-PILAE [26], CPILer [24], D-RVFL [11], dRVFL [14], SP-RVFL [32]. PILAE is proposed by applying PIL to the training of autoencoder. LR-PILAE is proposed to solve the problem of automatic selection of network structure by using low rank constraint. CPILer uses graph Laplace regularization to solve the robustness problem of AutoML system. In [23], The combination of PILAE and AdaBoost is used to solve the problem of driving stress recognition. Deep random vector functional link (D-RVFL) [11] is a multi-layer RVFL network by stacking. The deep RVFL (dRVFL) [14] is another multi-layer RVFL network. The dRVFL uses RVFL as the basic building block. Except for the first layer, the enhancement unit of each layer is obtained by multiplying the previous layer enhancement unit by a random weight. The enhancement units of all layers are concatenated together as the enhancement unit of the dRVFL, and then the weight of the output layer is determined by the least squares method.

DensePILAE

In this section, we will introduce the basic theory of PILAE and our proposed DensePILAE.

Basic theory of PILAE

The pseudoinverse learning algorithm (PIL) [6–8] is a fast training method for a single hidden layer feed-forward neural network. It uses the random or pseudoinverse of input data to initialize the weight between the input layer and the hidden layer, and the weight between the hidden layer and the output layer can be obtained in the form of an analytical solution.

Given training set $\mathbf{D} = \{\mathbf{X}, \mathbf{Y}\}$, the weight between the input layer and the hidden layer is represented as \mathbf{W}_{in} , and the weight between the hidden layer and the output layer is represented as \mathbf{W}_{out} . \mathbf{W}_{in} is initialized by random or pseudoinverse of input matrix \mathbf{X} , the hidden layer output \mathbf{H} is

$$\mathbf{H} = f(\mathbf{X}\mathbf{W}_{in}), \quad (1)$$

$f(\cdot)$ is activation function. Learning problems can be expressed as

$$\min_{\mathbf{W}_{out}} \|\mathbf{H}\mathbf{W}_{out} - \mathbf{Y}\|^2, \quad (2)$$

We can get the analytical solution of \mathbf{W}_{out} by solving the pseudoinverse of \mathbf{H} :

$$\mathbf{W}_{out} = \mathbf{H}^+ \mathbf{Y}. \quad (3)$$

The pseudoinverse of \mathbf{H} is

$$\mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T. \quad (4)$$

The autoencoder is essentially a three-layer neural network. The biggest difference is that a constraint is added. That is to make the input and output equal

$$\mathbf{Y} = \mathbf{X}. \quad (5)$$

Wang et al. [25] proposed PILAE using PIL training autoencoder. The weight of the encoder \mathbf{W}_e is initialized by random or pseudoinverse of input matrix \mathbf{X} . According to formulas Eqs. (3), (4) and (5), the weight of the decoder \mathbf{W}_d can be obtained as

$$\mathbf{W}_d = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{X}. \quad (6)$$

To avoid the ill-conditioned problem and enhance the generalization ability of the network, the L_2 regularization constraint is adopted for the decoder weight in PILAE. The weight formula of the decoder can be rewritten as

$$\mathbf{W}_d = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{X}, \quad (7)$$

where $\lambda > 0$ is the regularization parameter. Since the autoencoder is a symmetrical structure, to reduce the risk of overfitting, weight tied is used to reduce the number of parameters, then the weight of the encoder will be updated to

$$\mathbf{W}_e = \mathbf{W}_d^T. \quad (8)$$

Recalculating the output of hidden layer with new weights of encoder, we can get the feature.

Because the learning ability of single PILAE is limited, several PILAEs are stacked to learn. However, with the increase of depth, the performance improvement of stacked PILAE is not obvious. The reason is that to avoid identity mapping, the constraint of forced dimension reduction is added in the network structure of each PILAE. Therefore, there are a lot of necklace layers in stacked PILAE. Although the features are refined with the increase of depth, partial information is also lost. Therefore, it leads to the increase of model error.

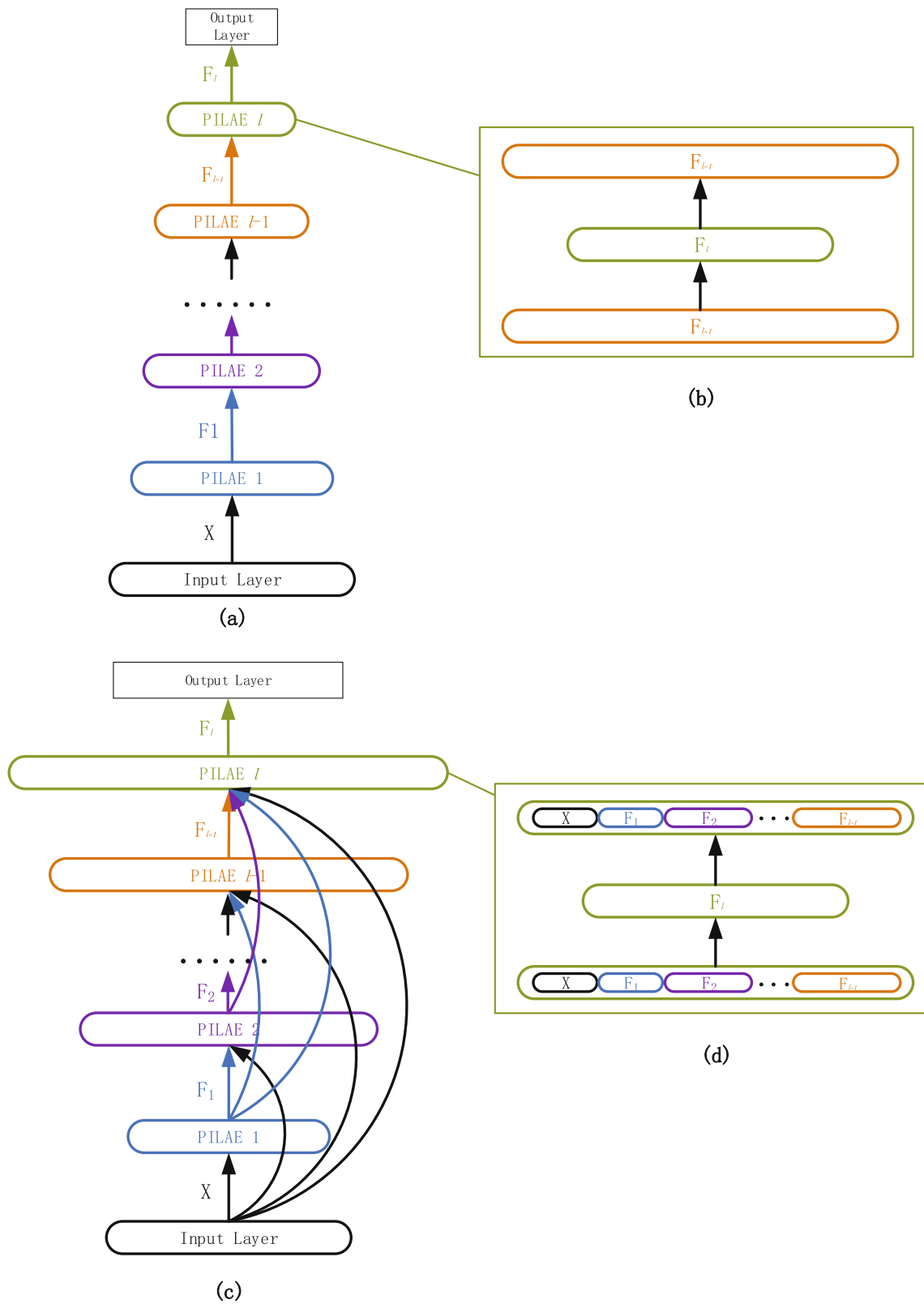


Fig. 1 Network structure of stacked PILAE and DensePILAE with four layers. **a** The structure of stacked PILAE. **b** l -th layer of stacked PILAE. **c** The structure of DensePILAE. **d** The l -th layer of DensePILAE

DensePILAE

To this end, we concatenate the output of all the previous layers as input to the subsequent layer. Figure 1 illustrates network structure of DensePILAE. The input of the l th layer is

$$\mathbf{D}_l = [\mathbf{X}, \mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_{l-1}], \quad (9)$$

where \mathbf{F}_i is the extracted feature of i th layer. According to Eq. (1), the hidden output \mathbf{H}_l is

$$\mathbf{H}_l = f(\mathbf{D}_l \mathbf{W}_{el}), \quad (10)$$

where \mathbf{W}_{el} is the random weight of the encoder in l th layer. According to Eqs. (7) and (8), the weight of the decoder in l th layer can be calculated as

$$\mathbf{W}_{dl} = (\mathbf{H}_l^T \mathbf{H}_l + \lambda \mathbf{I})^{-1} \mathbf{H}_l^T \mathbf{D}_l. \quad (11)$$

The weight of the l th layer encoder is obtained by weight tied, the feature extracted \mathbf{F}_l by the l th layer autoencoder can be calculated as

$$\mathbf{F}_l = f(\mathbf{D}_l \mathbf{D}_l^T \mathbf{H}_l ((\mathbf{H}_l^T \mathbf{H}_l + \lambda \mathbf{I})^{-1})^T). \quad (12)$$

DensePILAE is implemented by applying feature reuse to stacked PILAE. It has two advantages. One is that the lost information can be directly supplemented by identity connections, thus the error of the model is reduced. Another advantage is that the supplementary information comes from the features of low layers that have been learned, so there is no need to design new modules to learn the lost information.

On the whole, DensePILAE is a combination of width reuse and depth reuse. Layer by layer stacking realizes the feature reuse in depth perspective. It is implicit reuse. The concatenated feature realizes the feature reuse in width perspective. It is explicit reuse. Feature reuse reduces the error of the network and improves the feature learning ability of the network.

Experiments and discussion

Data set

To verify the validity of our proposed method, several experiments are performed on 8 public data sets in several fields, including MNIST, USPS, BA, Yale, ORL, COIL-20, COIL-100 and NORB data set. The MNIST, USPS and BA data set are handwritten font recognition data set. The Yale and ORL data set are face recognition data set. The COIL-20, COIL-

Table 1 Details of data sets

Data set	Samples	Dimensions	Classes
MNIST	4000	748	10
BA	1404	320	36
USPS	9298	256	10
ORL	400	1024	40
Yale	165	1024	15
COIL-20	1440	1024	20
COIL-100	7200	1024	100
NORB	48,600	2048	5

100 and NORB are object recognition data set. The data sets are described in detail as follows (Table 1):

- **MNIST** The Mixed National Institute of Standards and Technology (MNIST) is a handwritten digital identification data set, which contains a total of ten numbers from 0 to 9. MNIST has a total of 70,000 images, of which 60,000 images are trained and 10,000 images are tested. Each image is a 28×28 pixel grayscale image. In the experiment, we randomly selected 400 images for each class to form our experimental data set. The experimental data set contains a total of 4000 images.
- **BA** The Binary Alphanumeral (BA) data set includes 1404 samples, and each sample is a image, whose size is 20×16 . There are 36 categories, including numbers from 0 to 9 and letters from A to Z. Each category have 39 samples.
- **USPS**: The US Postal (USPS) handwritten digital data set includes 8-bit gray-scale images from “0” to “9”. The data set consists of 9298 images. The dimension of every image is 256.
- **ORL** The ORL data set is a face data set produced by Olivetti research laboratory in Cambridge University. There are 40 people, and each one has 10 different images. The size of each image is 92×112 . To compare with other methods, each image is subsampled to a size of 32×32 .
- **Yale** The Yale data set is created by Yale University. It consists of 165 samples from 15 different people. The samples of the same person has different lighting, expression or posture. Compared with ORL face database, the samples collected in Yale database contain more obvious changes of illumination, expression, posture and occlusion.
- **COIL-20** The Columbia object image library (COIL) contains two data sets, COIL-20 and COIL-100. It can be used for target and pose recognition. The COIL-20 data set contains images of 20 objects from different angles. The size of each image is 128×128 . To compare with

Table 2 Performance comparison in terms of ACC (%)

Data set	RVFL	stacked PILAE	dRVFL	DensePILAE
MNIST	86.03 ± 0.61	90.55 ± 1.05	88.02 ± 2.10	92.25 ± 1.50
BA	61.25 ± 4.16	68.63 ± 2.61	63.22 ± 3.68	73.31 ± 3.00
USPS	92.98 ± 1.11	95.17 ± 0.69	95.17 ± 0.69	96.25 ± 0.70
Yale	80.33 ± 10.38	81.33 ± 8.33	87.33 ± 7.57	82.33 ± 12.12
ORL	97.50 ± 2.24	98.25 ± 1.95	98.25 ± 2.51	98.75 ± 1.68
COIL-20	98.18 ± 1.38	99.93 ± 0.21	99.09 ± 0.63	99.94 ± 0.19
COIL-100	87.91 ± 0.97	96.48 ± 0.55	90.55 ± 1.07	99.34 ± 0.22
NORB	87.73 ± 1.21	83.40 ± 1.05	88.23 ± 8.92	92.99 ± 0.74

The best performance is shown in bold

Table 3 Performance comparison in terms of AUC (%)

Data set	RVFL	stacked PILAE	dRVFL	DensePILAE
MNIST	97.66 ± 0.31	99.31 ± 0.17	97.85 ± 0.52	99.50 ± 0.13
BA	95.60 ± 0.48	98.01 ± 0.46	96.11 ± 0.97	98.59 ± 0.25
USPS	99.18 ± 0.28	99.71 ± 0.10	99.36 ± 0.17	99.82 ± 0.07
Yale	96.37 ± 2.65	95.80 ± 2.98	98.82 ± 0.62	97.18 ± 2.17
ORL	99.88 ± 0.23	99.98 ± 0.03	99.87 ± 0.28	100.00 ± 0.00
COIL-20	99.97 ± 0.03	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
COIL-100	99.20 ± 0.22	99.99 ± 0.00	99.20 ± 0.20	100.00 ± 0.00
NORB	97.43 ± 0.34	97.47 ± 0.26	96.71 ± 4.40	99.42 ± 0.10

The best performance is shown in bold

other methods, each image is subsampled to a size of 32×32 .

- *COIL-100* The COIL-100 data set contains 100 objects from different angles. The size of each image is 128×128 . To compare with other methods, each image is subsampled to a size of 32×32 .
- *NORB* The NYU object recognition benchmark (NORB) contains 5 classes, namely, animals, humans, airplanes, trucks and cars. There are 9720 images in each category. The SmallNORB [2] is used in the experiment. The size of each image is 32×32 .

Table 4 Performance comparison in terms of time cost on feature learning (measured by second)

Data set	RVFL	stacked PILAE	dRVFL	DensePILAE
MNIST	0.02	0.54	0.08	0.13
BA	0.04	0.43	0.08	0.12
USPS	0.04	0.09	0.07	0.12
Yale	0.02	0.58	0.08	0.11
ORL	0.03	0.27	0.08	0.17
COIL-20	0.02	0.77	0.11	0.18
COIL-100	0.02	0.72	0.06	0.15
NORB	0.04	1.03	0.07	1.53

Compared methods

We compare the proposed DensePILAE with three non-BP methods, stacked PILAE [25], RVFL [18], and dRVFL [14]. Stacked PILAE is a forward learning algorithm that uses PIL to quickly train SAE. In RVFL, the input layer is directly connected with the output layer. Therefore, RVFL is a special single hidden layer feed-forward neural network. The dRVFL is an extension of RVFL in the depth direction. Its characteristic is that only the weight of the last layer is obtained by learning, and the weights of all the previous layers are generated by random projection.

Experiment settings

To compare the different methods fairly, the number of neurons in the hidden layer of RVFL is set to 100. The number of neurons all hidden layers of dRVFL and DensePILAE is all set to 100, and the number of layers is set to 10. The width and depth of stacked PILAE is set by cross validation. The activation functions of all methods is sigmoid function. The regularization parameter λ is selected in the range of $\{2^{-6}, 2^{-4}, 2^{-2}, 2^0, 2^2, 2^4, 2^6, 2^8, 2^{10}\}$. To reduce the randomness and contingency as much as possible, the final experimental results are obtained by 10-fold cross validation. Our experiments are performed on a Geforce GTX 1080 GPU.

Performance comparison and analysis

To verify the effectiveness of DensePILAE, we first report the accuracy (ACC) and the area under curve (AUC) of DensePILAE and other methods on 8 data sets. Among them, DensePILAE gets the best ACC and AUC on 7 data sets. In other words, DensePILAE outperformed other methods on 87.5% data sets. The ACC of DensePILAE is more than 99% on COIL-20 and COIL-100 data set. The AUC of DensePILAE is 100% on ORL, COIL-20 and COIL-100 data set. Table 2 shows the average values of ACC for the compared method and our proposed method, and Table 3 shows the average AUC of the compared method and our proposed method. The results are the average values of tenfold cross validation.

Comparison between stacked PILAE and DensePILAE In Tables 2 and 3, we can find that DensePILAE achieves the best results on all 8 data sets. Specifically, the accuracy of DensePILAE is improved more obviously on NORB, BA and COIL-100 data sets, where the improvements of ACC reach 9.59%, 4.68% and 2.86%, respectively. However, the improvement is smaller on ORL and COIL-20 data sets, only 0.5% and 0.01%. The AUC of DensePILAE is significantly improved on NORB and Yale data sets, where the improvements of ACC reach 1.95% and 1.38%, respectively. The results of experiments show that the feature reuse can significantly improve the feature extraction ability of network, and make the network helpful to extract more generalized features.

Comparison between dRVFL and DensePILAE In Tables 2 and 3, we can find that DensePILAE get the best results on 7 data sets and is defeated on Yale data set. Specifically, the accuracy of DensePILAE is significantly improved on BA, COIL-100, NORB and MNIST data sets, reaching 10.09%, 9.79%, 4.76% and 4.23%, respectively. In the COIL-20 and ORL data sets, the accuracy of improvement is weak, less than 1%. The AUC of DensePILAE, respectively, increases by 2.71%, 2.48% and 1.65% on NORB, BA and MNIST data sets. However, the improvement on USPS is weak, only 0.46%. The results show that compared with the features obtained by random projection with dRVFL, the features obtained by pseudoinverse have stronger discrimination ability. In addition, it also shows that the feature reuse by densely connect can extract better features even for simple network structure.

Time analysis

The time cost is an important criteria to evaluate the performance of the model. Feature learning takes up most of the time cost. We report the time of feature learning on 8

data sets in Table 4. It can be seen from the table that the order of feature learning speed from fast to slow is RVFL, dRVFL, DensePILAE and stacked PILAE. Except NORB data set, DensePILAE is slightly slower than dRVFL. This is because the weights of each layer of dRVFL except output layer don't need to be learned, is only set to random projection of input. However, the weights of DensePILAE need to be learned. DensePILAE is faster than stacked PILAE. As the depth increases of DensePILAE, the input of every PILAE is increasing in DensePILAE, but the hidden width is fixed. Because the features of low layers are reused, the width of hidden layer could be set smaller. In stacked PILAE, the width of hidden is closely related to the width of input, which is usually larger. The width of stacked PILAE is larger than that of DensePILAE, so the time cost of stacked PILAE is large.

Parameter sensitivity analysis

In neural networks, the selection of parameters plays an important role in the network performance. In DensePILAE, regularization parameter and the number of hidden neurons are two important hyperparameters. We use grid search method to analyze the influence of two parameters on the performance of DensePILAE. The search range of regularization parameters λ is from 2^{-6} to 2^{12} . Each sample point is four times the previous one. Therefore, the selected regularization parameters are $\{2^{-6}, 2^{-4}, 2^{-2}, 2^0, 2^2, 2^4, 2^6, 2^8, 2^{10}, 2^{12}\}$. The number of neurons in the hidden layer is selected from 10 to 100, and the interval between the two samples is 10. Therefore, the selected numbers of hidden neurons H are $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$.

We have carried out parameter sensitivity analysis experiments on 8 data sets. Figures 2 and 3 are the experimental results of ACC and AUC, respectively. On NORB, MNIST, COIL-100 and USPS data sets, when the regularization parameter is small, ACC and AUC will gradually reach the best performance with the increase of width in hidden layer. The best ACC values are 92.99%, 92.58%, 99.35% and 96.25%, respectively. In addition, the best AUC values are 99.42%, 99.51%, 100.00% and 99.83%, respectively. On Yale and ORL data sets, larger regularization parameters lead to better ACC and AUC. The width of hidden layer has limited influence on the final results. The best ACC values are 83.33% and 98.75%, respectively. The best AUC values are 97.96% and 100.00%, respectively. For COIL-20 data set, when the regularization parameter is small, as long as the number of hidden layer neurons exceeds threshold, the perfect learning can be obtained. The best ACC and AUC are 99.94% and 100%, respectively. For BA data set, although ACC and AUC will increase slowly with the increase of width in the hidden layer, the regularization parameter have great influence. Therefore, it must be selected carefully.

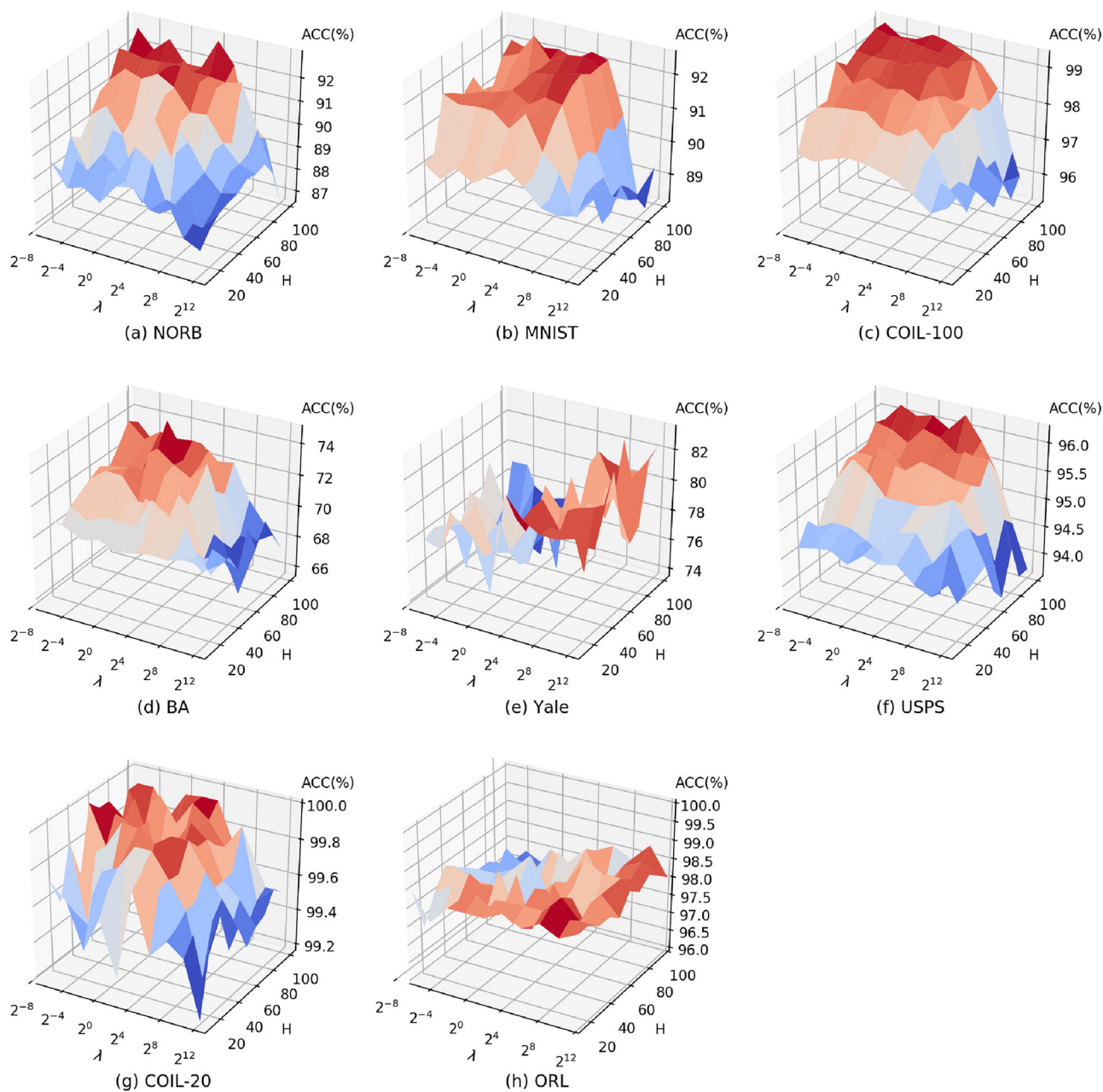


Fig. 2 Performance of DensePILAE with different parameters in terms of ACC(%)

In a word, ACC and AUC can be improved with the increase of the number of neurons in the hidden layer. The larger number of hidden neurons will contribute to get better results. The regularization parameter has a more important impact on the performance of the model. For most data sets, smaller regularization parameter will get an acceptable result. However, if you want to get the best result, you need to choose it carefully.

Discussion

There are significant differences between dRVFL and DensePILAE in feature reuse and feature learning. The dRVFL reuses the features of the all previous layer in the last layer. However, the DensePILAE reuses the features of all previous layers in every layer. Therefore, the feature learned from every layer in DensePILAE is the comprehensive utilization of historical information. It can be seen from Tables 2 and 3 that DensePILAE can obtain better results than dRVFL. In addition, the features of hidden layer are obtained by random projection in

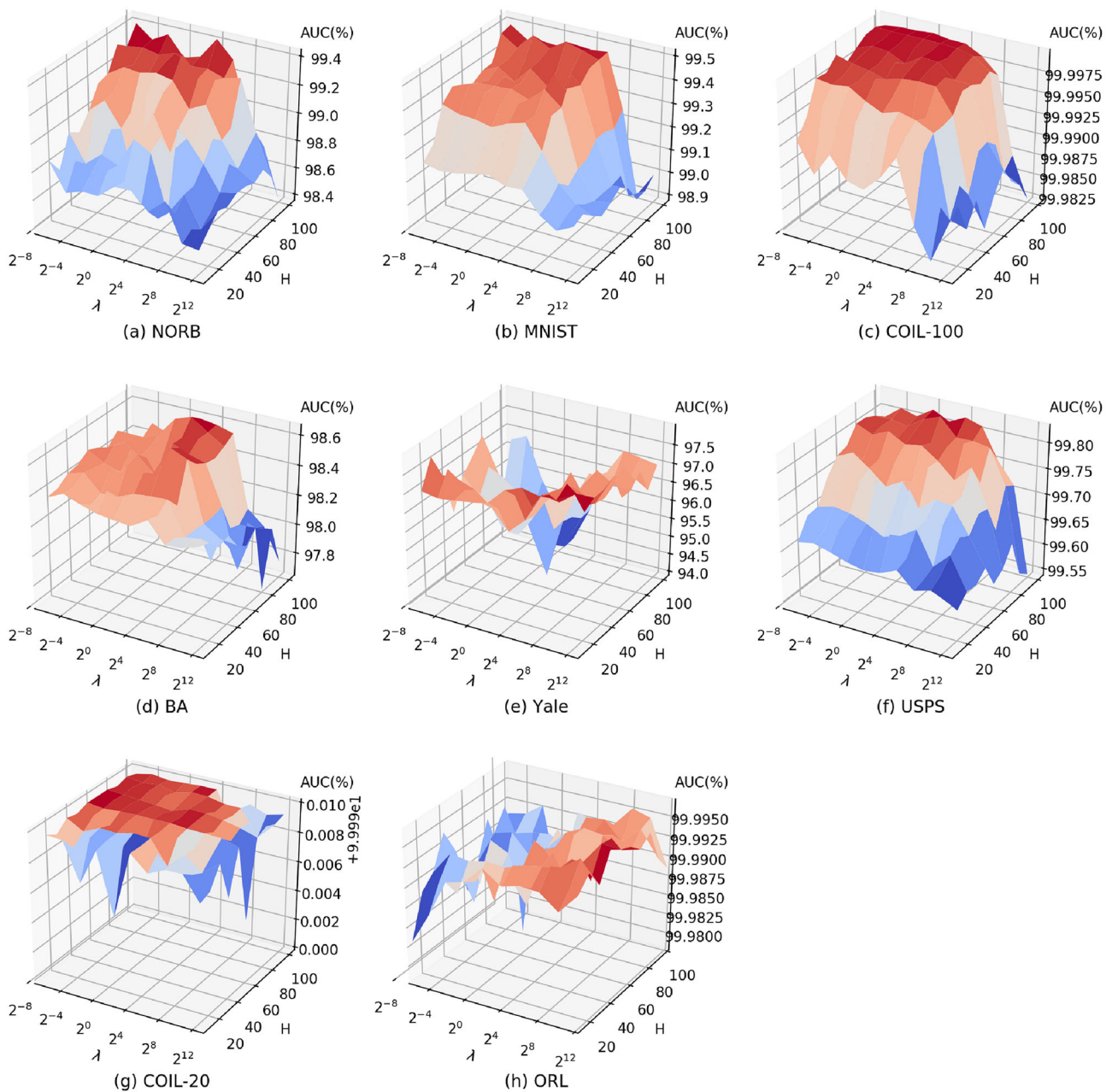


Fig. 3 Performance of DensePILAE with different parameters in terms of AUC(%)

dRVFL, so dRVFL is similar to width learning network. However, the hidden layer features in DensePILAE are obtained by pseudoinverse learning. Therefore, it can be seen from Table 4 that the speed of feature learning in DensePILAE is slightly slower.

Conclusion

In this paper, a dense connection pseudoinverse learning autoencoder based on feature reuse is proposed. The method

can reuse the information of the middle layer faster and better, and the learned features have a stronger discriminating ability. Our method can be seen as a combined implementation of explicit reuse and implicit reuse. The explicit reuse of features is realized by crossing connections, and the implicit reuse of features is realized by multi-layer stacking. In addition, the method can not only greatly shorten the feature extraction time of the network, but also effectively avoid the gradient explosion and gradient vanished problems. The experimental results show that the proposed method has comprehensive performance compared with the other non-BP

based methods. This is because the feature reuse makes up for the loss of information and reduces the error of network. Moreover, this strategy can also be applied to other non-BP based learning networks to further improve the performance of the network. In addition to image classification, DensePI-LAE can be applied in many scenarios. In the future, we will apply DensePI-LAE to object detection and fault detection.

Funding This work is supported by Department of science and technology of Shanxi Province (No. 201901D211415).

Availability of data and materials All data sets used in this paper are all public data sets. The COIL-20 and COIL-100 are released by Columbia University, and the URL is “<http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>” and “<http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>” respectively. The ORL data set is released by Cambridge University, and its URL is “<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>”. The Yale data set is released by Yale University, and its URL is “<http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html>”. The URL of MNIST data set is “<http://yann.lecun.com/exdb/mnist/>”. The URL of BA data set is “<https://cs.nyu.edu/protect/unhbox\voidb@x\penalty\M\roweis/data/>”. The Libsvm is the source of the USPS and NORB data set, and its URL is “<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>”.

Declarations

Conflict of interest The authors declare that there is no conflict of interest in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 35(8):1798–1828
- Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol* 2(3):1–27
- Deng L, Yu D, Platt J (2012) Scalable stacking and learning for building deep architectures. In: *IEEE international conference on acoustics, speech and signal processing*, pp 2133 – 2136
- Divya R, Peter JD (2021) Smart healthcare system-a brain-like computing approach for analyzing the performance of detection2 and PoseNet models for anomalous action detection in aged people with movement impairments. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-021-00319-8>
- Gong D, Liu L, Le V, Saha B, Mansour MR, Venkatesh S, Hengel AVD (2019) Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection. In: *2019 IEEE/CVF international conference on computer vision (ICCV)*
- Guo P, Chen CLP, Sun Y (1995) A exact supervised learning for a three-layer supervised neural network. In: *Proceedings of 1995 international conference on neural information processing*
- Guo P, Lyu MR (2001) Pseudoinverse learning algorithm for feedforward neural networks. In: *Advances in neural networks and applications*, pp 321–326
- Guo, P., Lyu, M.R.: A pseudoinverse learning algorithm for feedforward neural networks with stacked generalization applications to software reliability growth data. *Neurocomputing* 56(1), 101–121 (2004)
- Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: A review. *Neurocomputing* 187:27–48
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*
- Henriquez PA, Ruz GA (2018) Twitter sentiment classification based on deep random vector functional link. In: *2018 international joint conference on neural networks (IJCNN)*. IEEE, Rio de Janeiro
- Hinton GE (2006) Reducing the Dimensionality of Data with Neural Networks. *Science* 313(5786):504–507
- Huang G, Liu Z, Weinberger KQ, van der Maaten L (2017) Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*
- Katuwal R, Suganthan PN, Tanveer M (2019) Random vector functional link neural network based ensemble deep learning. [arXiv:1907.00350](https://arxiv.org/abs/1907.00350)
- Liu C, Cao Y, Luo Y, Chen G, Vokkarane V, Yunsheng M, Chen S, Hou P (2018) A New Deep Learning-Based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure. *Ieee Transactions on Services Computing* 11(2):249–261
- Liu R, Tang F, Wang Y, Zheng S (2021) A modified NK algorithm based on BP neural network and DEMATEL for evolution path optimization of urban innovation ecosystem. *Complex Intell Syst*
- Mohammadi M, Alfuqaha A (2018) Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges. *IEEE Communications Magazine* 56(2):94–101
- Pao YH, Takefuji Y (1992) Functional-link net computing: theory, system architecture, and functionalities. *IEEE Computer* 5:76–79
- Sun X, Gui G, Li Y, Liu RP, An Y (2019) ResInNet: A Novel Deep Neural Network With Feature Reuse for Internet of Things. *IEEE Internet of Things Journal* 6(1):679–691
- Tian Y, Li P (2015) Predicting short-term traffic flow by long short-term memory recurrent neural network. In: *IEEE international conference on smart city/socialcom/sustaincom*
- Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on machine learning*. ACM Press, Helsinki
- Wang J, Guo P, Xin X (2018) Review of pseudoinverse learning algorithm for multilayer neural networks and applications. In: *International symposium on neural networks*. Springer, pp 99–106
- Wang K, Guo P (2021) An ensemble classification model with unsupervised representation learning for driving stress recognition using physiological signals. *IEEE Transactions on Intelligent Transportation Systems* 22(6):3303–3315
- Wang K, Guo P (2021) A Robust Automated Machine Learning System with Pseudoinverse Learning. *Cognitive Computation* 13(3):724–735

25. Wang K, Guo P, Luo AL (2016) A new automated spectral feature extraction method and its application in spectral classification and defective spectra recovery. *Monthly Notices of the Royal Astronomical Society* 465(4):4311–4324
26. Wang K, Guo P, Xin X, Ye Z (2017) Autoencoder, low rank approximation and pseudoinverse learning algorithm. In: 2017 IEEE international conference on systems, man, and cybernetics. IEEE Press, pp 948–953
27. Wei X, Li J, Yuan Q, Chen K, Zhou A, Yang F (2019) Predicting fine-grained traffic conditions via spatio-temporal LSTM. *Wirel Commun Mob Comput*
28. Yu F, Wang D, Shelhamer E, Darrell T (2018) Deep layer aggregation. In: 2018 IEEE/CVF conference on computer vision and pattern recognition. IEEE, Salt Lake City, UT
29. Zhang F, Mao ZJ, Huang Y, Xu L, Ding G (2018) Deep learning models for EEG-based rapid serial visual presentation event classification. *Journal of Information Hiding and Multimedia Signal Processing* 9:177–187
30. Zhang L, Jiao L, Ma W, Duan Y, Zhang D (2019) PolSAR image classification based on multi-scale stacked sparse autoencoder. *Neurocomputing* 351:167–179
31. Zhang R, Isola P, Efros AA (2017) Split-brain autoencoders: unsupervised learning by cross-channel prediction. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Honolulu, HI
32. Zhang Y, Wu J, Cai Z, Du B, Yu PS (2019) An unsupervised parameter learning model for RVFL neural network. *Neural Networks* 112:85–97

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.