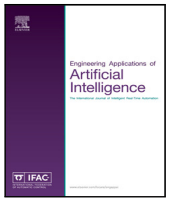
Engineering Applications of Artificial Intelligence 104 (2021) 104364

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Graph Convolutional Neural Networks with Geometric and Discrimination information

Ronghua Shang, Yang Meng *, Weitong Zhang, Fanhua Shang, Licheng Jiao, Shuyuan Yang

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xidian, Shaanxi Province 710071, China

ARTICLE INFO

Keywords:
Deep learning
Convolutional neural network
Spectral theory
Local structure
Discriminant information

ABSTRACT

In recent years, geometric deep learning methods have been proposed, which are called Graph Convolutional Neural Networks (GCNNs). GCNNs not only can extract effective features like the classical CNN, but also can effectively reflect the true geometric structure of original data. Although GCNNs consider the geometric structure of original data, they construct the same feature graph to perform graph convolution, and ignore the difference between the local structures of different samples. Therefore, a novel Graph Convolutional Neural Network with Geometric and Discrimination information (GDGCNN) is proposed, which integrates traditional machine learning ideas to further improve the performance of feature extraction. In order to exploit differences between the local structures of different samples and make full use of the geometric structure of original data, GDGCNN constructs different feature graphs for different training batches to fully exploit the local geometry of data. Moreover, the discriminant regularization is introduced into GDGCNN to effectively utilize the discriminant information contained in original data. Therefore, GDGCNN has good discriminative ability and robustness. The experimental results show that GDGCNN can perform feature extraction tasks very well, and it is superior to some existing methods for classification in terms of accuracy and F1-Score.

1. Introduction

In recent years, the research and development of big data processing and data mining have been continuously promoted with the explosive growth of data dimensions and data sizes (De Una et al., 2018; Krishnan et al., 2018; Shang et al., 2016a). For massive high-dimensional data, extracting features efficiently is needed (Shang et al., 2021). Recently, when extracting features from original data, traditional machine learning and deep leaning are the two kinds of technologies that are often based on Shang et al. (2020). For example, both methods can be used to process the handwritten digits dataset. However, both methods have their own advantages and disadvantages.

Traditional machine learning algorithms usually have certain physical meanings and interpretability, and can extract effective feature information contained in original data by simple calculations. However, these methods usually need to determine the features in advance. If the number of features is too small, it may not be able to classify it accurately, that is under-fitting. If the number of features is too large, they may pay too much attention to a certain feature in the classification process and result in high classification errors, that is over-fitting.

The concept of deep learning is derived from the study of artificial neural networks (Ptucha et al., 2019; Zhang et al., 2019; Xie et al., 2019). For example, the multi-layer perceptron with many hidden layers is a deep learning structure. Deep learning combines low-level features to form more abstract high-level representation or features, which can discover distributed feature representations of data.

Since Hinton et al. proposed the concept of deep learning in 2006 (Hinton and Salakhutdinov, 2006), deep learning has attracted more and more attention. Deep learning is a new field in machine learning, whose motivation is to build and simulate a human brain neural network for analysis and learning. It simulates the mechanism of the human brain to interpret data such as images, sounds and texts. In fact, in 1998, Lecun et al. proposed the first multi-layer structure learning algorithm, the Convolutional Neural Network (CNN) (LeCun et al., 1998). Multi-dimensional input images can be directly input into CNN, which avoids the complexity of data reconstruction in feature extraction and classification in traditional machine learning. CNN uses the convolutional layer and the pooling layer to form multiple convolution groups and extract features layer by layer, which can continuously reduce the dimensions of large-scale data in the image recognition, and finally make it easy to be trained. CNN reduces the complexity of the network with its special structure of local weight sharing, thus it has some unique advantages in speech recognition and image processing, and is widely used in many real-world applications (Pu et al., 2018; Luo et al., 2017). However, the convolution and pool operators in CNN are only used for regular grids and not suitable for graph-structured

* Corresponding author.
E-mail address: xdyangmeng@163.com (Y. Meng).

https://doi.org/10.1016/j.engappai.2021.104364
Received 6 May 2020; Received in revised form 19 June 2021; Accepted 21 June 2021
Available online 5 July 2021

data such as gene data on biological regulatory networks, user data on social networks. In order to extend CNN, many scholars combine it with spectral theory to form graph convolutional neural networks (GCNNs) (Defferrard et al., 2016; Kipf and Welling, 2016; Yang et al., 2017; Zhuang and Ma, 2018; Chen et al., 2017; Veličković et al., 2018). Thus, the applications of GCNNs are more extensive.

Although some machine learning methods are technical and elaborated, deep learning methods enable feature learning with multi-layer networks, which show advantages in many problems, especially on large-scale data. However, deep learning methods usually lack certain physical significance and explainability, and their performance needs to be improved. Therefore, traditional machine learning methods with new deep learning methods GCNNs are combined to form a new algorithm, which can have the advantages of these two kinds of methods. In view of the shortcomings of the existing GCNNs, Graph Convolutional Neural Networks with Geometric and Discrimination information (GDGCNN) is proposed. GDGCNN not only can effectively mine graph-structured data, but also can make full use of local structure information and discrimination information. Therefore, it can extract features more efficiently for further clustering and classification. The main contributions of this paper are summarized as follows:

1. GDGCNN constructs different feature graphs for different training batches to address the issue in GCNNs of neglecting the difference between the local structures of different samples in the dataset. Therefore, it can fully exploit the local geometry of original data.
2. GDGCNN effectively uses the discrimination information of original data to extract more discriminative features, and thus it has better learning ability and discriminating ability.
3. GDGCNN integrates traditional machine learning ideas into the new framework of GCNN to further improve the performance of feature extraction. The experimental results also verify that GDGCNN has higher accuracy and F1-Score in the image classification task.

The rest of this paper is organized as follows: in Section 2, the classical CNN and Graph Convolutional Neural Networks (GCNNs) is reviewed. In Section 3, the theoretical model of Graph Convolutional Neural Networks with Geometric and Discrimination information (GDGCNN) is introduced. In Section 4, the proposed algorithm is compared with the related algorithms, extensive experiments are done to prove the efficiency and effectiveness of the proposed GDGCNN. In Section 5, the paper and look forward to the future work are summarized.

2. Related work

In recent years, with the rapid development of artificial intelligence and the explosive growth of data, how to extract effective features of data has become one of the study hotspots in big data processing and data mining. Machine learning (Shang et al., 2017; Xu et al., 2019; Shang et al., 2019a), deep learning (Zhao and Kumar, 2019; He and Schomaker, 2019; Kuang et al., 2018) and spectral theory (Shang et al., 2018; Cai et al., 2007) have become hot research directions. For massive high-dimensional data, efficient feature extraction is needed (Shang et al., 2019b; Cai et al., 2010; Meng et al., 2018a) before the image clustering and classification. Before introducing our model, the classical CNN and Graph Convolutional Neural Networks (GCNNs) are reviewed in this part.

2.1. Convolutional Neural Network (CNN)

Before the emergence of CNN, SIFT, HoG or other algorithms are used to extract distinguishing features. SIFT has invariance to a certain degree of distortion, translation, rotation, angle change, brightness adjustment and other distortions, which is one of the most important image feature extraction methods in the past. Based on feature extraction methods (such as SIFT), the classifier (such as SVM) should be used for the image recognition. However, SIFT has limited ability to extract features, and the biggest challenge of early image recognition is how to organize and extract features. Convolutional neural network (CNN) was originally designed to solve the problems like the image recognition. Now, CNN is widely used not only for images and video, but also for time series signals such as audio signals, text data and so on (He et al., 2018; Liao et al., 2018).

Convolutional neural network (CNN) is a multi-layer neural network that can continuously reduce the dimensions of large-scale data in the image recognition, and finally make it easy to be trained. CNN was first proposed by Yann LeCun and successfully applied to the handwritten digits dataset MNIST (LeCun et al., 1998). The network proposed by LeCun is called LeNet-5, which is the most typical convolutional neural network consisting of convolutional layers, pooling layers and fully connected layers. The convolutional layer and the pooling layer cooperate to form a plurality of convolution groups to extract features layer by layer, and finally the classification is completed through several fully connected layers.

The input images of LeNet-5 are 32×32 grayscale images, followed by convolutional layers, pooling layers and fully connected layers. C1 is a convolutional layer with 6 feature maps with convolution kernel size is 5×5, so there are $(5 \times 5 + 1) \times 6 = 156$ parameters. S2 is a 2×2 average pooling layer for subsampling, followed by a Sigmoid activation function for nonlinear processing. C3 is the second convolutional layer with 16 feature maps with convolution kernel size is 5×5. S4 is the second pooling layer which is same as the first pooling layer S2. C5 is the third convolutional layer with 120 feature maps with convolution kernel size 5×5. As the input image size of C5 is also 5×5, it constitutes a full connection, and can be considered as a fully connected layer. F6 is a fully connected layer with 84 hidden nodes and a Sigmoid activation function. Gaussian (full) connection layer consists of Euclidean radial basis function units, which outputs the final classification results.

In summary, CNN is inspired by the concept of local receptive field, which simulates feature distinction by the convolution and local connection. By sharing the weights of convolution, CNN can reduce the amount of network parameters, and greatly reduce the training complexity. Therefore, CNN not only can avoid the over-fitting, but also can give the convolution network tolerance to translation by the local weight sharing. The pooling layer is mainly to reduce the data dimensions. Subsampling in the pool layer further reduces the amount of output parameters, and gives the model tolerance to the slight deformation, which can improve the generalization ability of the model. Finally, classification and other tasks can be accomplished by traditional neural networks.

2.2. Graph Convolutional Neural Networks (GCNNs)

With the diversification of data, researchers have found that the classical convolutional neural networks show drawbacks in many data, such as non-Euclidean structure data. Log data on telecommunication networks, gene data on biological regulatory networks, user data on social networks, or text documents on word embeddings are all important data examples on irregular or non-Euclidean domains (Defferrard et al., 2016).

The convolution and pool operators in CNN are only suitable for regular grids. To address this issue, Defferrard et al. present a formulation of CNN in the context of spectral graph theory to form graph convolutional neural networks (GCNN) (Defferrard et al., 2016), which is suitable for graph-structured data. Based on similar idea, Kipf et al. present a similar method, 1stGCNN (Kipf and Welling, 2016). The new filters defined in GCNN and 1stChebNet are localized in the graphs. The learned weights can be shared with different locations in a graph. The spectral graph convolutions operation in GCNN is K-localized since it is

a Kth order Chebyshev polynomial in the Laplacian, which can remove the require to compute the eigenvectors of the Laplacian. In addition, 1stGCNN limits $K = 1$ for the layer-wise spectral graph convolution operation in GCNN to deal with the problem of overfitting for graphs with the wide node degree distributions.

Moreover, Yang et al. proposed a graph regularized deep neural network (GR-DNN) (Yang et al., 2017), which can preserve the high-level semantics and the geometric structure within local manifold tangent space. Zhuang and Ma proposed a simple and scalable semi-supervised learning method for graph-structured data called dual graph convolutional neural network (DGCN) (Zhuang and Ma, 2018), where two convolutional neural networks are devised to embed the local-consistency-based and global-consistency-based knowledge, respectively. Chen et al. proposed a new method Stochastic Training of Graph Convolutional Networks with Variance Reduction (StoGCNN) (Chen et al., 2017), which used the historical activations of nodes as a control reduce the receptive field size for GCNN. Veličković et al. proposed an unsupervised method for learning node representations on the graph-structured data, which is called Deep Graph Infomax (DGI) (Veličković et al., 2018). DGI is based on maximizing mutual information between patch representations and the corresponding high-level summaries of graphs, which are both obtained from the established graph convolutional network architectures. The subgraphs centered around nodes of interest are summarized by the patch representations, which can be reused for downstream node-wise learning tasks. DGI can be readily applicable to both inductive and transductive learning setups since it does not rely on random walk objectives like most prior GCNNs.

These GCNNs are common methods to any geometric structures, and they have the same learning complexity and linear computational complexity as the classical CNN. However, there are some shortcomings for these GCNNs. For example, on the same dataset, they construct the same feature graph for graph convolution, so the difference between the local structures of different samples in the dataset are neglected. In addition, these existing GCNNs only use the geometric structure of the data but ignore the discrimination information, and thus they cannot extract more discriminative features.

3. Theoretical model of Graph Convolutional Neural Networks with geometric and discrimination information

In view of the high-dimensional and large-scale data in the era of big data, an efficient dimension reduction method to extract effective features from high-dimensional data is needed, and further achieve better classification performance. In this paper, traditional machine learning methods with new deep learning methods GCNNs are combined to form a new algorithm, which can have the advantages of these two kinds of methods. It not only can effectively improve the performance of feature extraction, but also has certain physical significance and explainability.

The existing GCNNs can mine the geometric structure of original data with spectral graph convolution. However, they construct the same feature graph for graph convolution, thus the differences between the local structures of different samples in the dataset are neglected when mining geometric structure. To address this issue, the proposed algorithm constructs different feature graphs for different training batches in the dataset to fully exploit the geometric structure and the difference between the local structure of different samples in original data. Moreover, the existing GCNNs ignore the discriminant information of the data, which is often used in the traditional machine learning methods. In the proposed algorithm, the discriminant regularization is introduced into GCNN, which makes the algorithm more discriminative and further improves the performance of feature extraction.

3.1. Spectral graph convolution

The spectral graph convolution instead of the regular convolution according to Defferrard et al. (2016) is used, combined with the geometric and discrimination information.

A spectral graph convolution is defined, which is calculated by multiplying $\mathbf{x} \in \Re^n$ in the input data $\mathbf{X} \in \Re^{n\times d}$ with a filter $g_\theta = diag(\boldsymbol{\theta})$ in Fourier space:

$$g_\theta * \mathbf{x} = \mathbf{U} g_\theta \mathbf{U}^T \mathbf{x}, \tag{1}$$

where $\theta \in \Re^n$ is a vector of Fourier coefficients, so g_θ is a non-parametric filter. The Laplacian operator $\mathbf{L}$ is diagonalized by the Fourier basis $\mathbf{U}$, so $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, $\mathbf{A} \in \Re^{n\times n}$ is a similarity matrix, $\mathbf{D} = [D_{ij}] \in \Re^{n\times n}$ is a diagonal matrix, $D_{ii} = \sum_j A_{ij}$, $\mathbf{U}^T\mathbf{x}$ is the graph Fourier transform of $\mathbf{x}$, $\boldsymbol{\Lambda} = diag([\lambda_0, \ldots, \lambda_{n-1}]) \in \Re^{n\times n}$, $\lambda_0, \ldots, \lambda_{n-1}$ are n eigenvalues of $\mathbf{L}$. Therefore, g_θ can be considered as a function of the eigenvalues of $\mathbf{L}$, such as $g_\theta(\boldsymbol{\Lambda})$.

Although the graph Fourier transform can be realized by Eq. (1), the computational complexity is very high, and is $O(n^2)$. In addition, the computation on the feature decomposition of $\mathbf{L}$ is very large for large-scale graphs. Faced with this problem, an effective solution is to transform $g_\theta(\boldsymbol{\Lambda})$ into a polynomial and determine its parameters. In Hammond et al. (2011), Hammond et al. proposed a solution to this problem, $g_\theta(\boldsymbol{\Lambda})$ can be well-approximated by a truncated expansion in terms of Chebyshev polynomials $T_k(\mathbf{x})$ up to Kth order:

$$g_{\theta'}(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\boldsymbol{\Lambda}}), \tag{2}$$

where $\tilde{\boldsymbol{\Lambda}} = \frac{2}{\lambda_{\max}}\boldsymbol{\Lambda} - \mathbf{I}_n$, $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{L}$, $\theta \in \Re^K$ is a vector of Chebyshev coefficients. Chebyshev polynomials can be computed recursively, such as $T_0(\mathbf{x}) = 1$, $T_1(\mathbf{x}) = \mathbf{x}$, $T_k(\mathbf{x}) = 2\mathbf{x}T_{k-1}(\mathbf{x}) - T_{k-2}(\mathbf{x})$, so the graph Fourier transform of $\mathbf{x}$ can be expressed as:

$$g_{\theta'} * \mathbf{x} \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\mathbf{L}})\mathbf{x}, \tag{3}$$

where $\tilde{\mathbf{L}} = \frac{2}{\lambda_{\max}}\mathbf{L} - \mathbf{I}_n$ is a Kth order polynomial of the Laplacian, so it is now K-localized, and it depends only on nodes that are within K steps away from the central node (Kth order neighborhood). Therefore, the computational complexity of Eq. (3) is $O(|E|)$, which is linear in the number of edges.

The pooling operation needs meaningful neighborhoods of graphs, so the similar vertices should be clustered together. Therefore, the coarsening phase of the Graclus multilevel clustering algorithm (Dhillon et al., 2007) is used, which is very effective in clustering various graphs.

3.2. Local graph regularization

Recently, manifold learning theory and spectral clustering theory have proved that constructing neighborhood graphs between discrete data points can effectively simulate the local geometric structure of the original data (Meng et al., 2018b). Therefore, in order to mine the difference between the local structures of different samples in original data, training samples are divided into batches and the training samples in each batch are constructed into neighborhood graphs to obtain the local graph regularization. The main differences between the proposed algorithm and the previous algorithms are shown in Fig. 1.

Suppose there are N training samples in the training set. The previous algorithms construct a feature graph for training, and the similarity matrix is shown in Fig. 1(a). In this paper, training samples are divided into batches and the training samples in each batch are constructed into neighborhood graphs. The obtained similarity matrix is shown in Fig. 1(b). Suppose there are n_i training samples in the training batch $\mathbf{X}_i$, each sample is d-dimensional, that is, $\mathbf{X}_i = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_i}] \in \Re^{d\times n_i}$. Construct a graph with n_i vertices, and each vertex represents a training

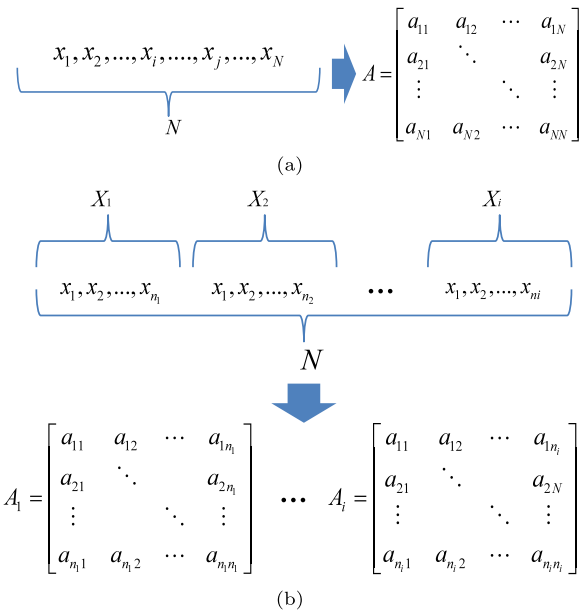


Fig. 1. Construct feature graphs. (a) The previous algorithms (b) The proposed algorithm.

sample. For each vertex x_j, find its k nearest neighbors and establish edges, and the weights on edges represent the similarity between the samples. Use $\mathbf{A}_i$ to represent the similarity matrix and use the following formula to measure the smoothness:

$$\begin{aligned}
&\frac{1}{2}\sum_{a=1}^{n_i}\sum_{b=1}^{n_i}\|\mathbf{x}_a-\mathbf{x}_b\|^2[\mathbf{A}_i]_{ab}\\
&=\sum_{a=1}^{n_i}\mathbf{x}_a^T\mathbf{x}_a[\mathbf{D}_i]_{aa}-\sum_{a=1}^{n_i}\sum_{b=1}^{n_i}\mathbf{x}_a^T\mathbf{x}_b[\mathbf{A}_i]_{ab}\\
&=Tr(\mathbf{X}_i^T\mathbf{D}_i\mathbf{X}_i)-Tr(\mathbf{X}_i^T\mathbf{A}_i\mathbf{X}_i)\\
&=Tr(\mathbf{X}_i^T\mathbf{L}_i\mathbf{X}_i),
\end{aligned}\tag{4}$$

where the Laplacian matrix is $\mathbf{L}_i=\mathbf{D}_i-\mathbf{A}_i$, and the diagonal element of the diagonal matrix $\mathbf{D}_i$ is the sum of the row elements of the matrix $\mathbf{A}_i$, i.e. $[\mathbf{D}_i]_{aa}=\sum_b[\mathbf{A}_i]_{ab}$. There are many ways to build weights in graphs, the common methods are (Cai et al., 2011): Binary (0–1) Weighting, Heat Kernel (Gaussian) Weighting, and Dot-product Weighting. The appropriate similarity matrix can be chosen according to the specific situation, so as to improve the accuracy of learning. For example, for image data, Heat Kernel (Gaussian) Weighting is usually used to measure the similarity between vertices.

GCNN constructs a neighborhood graph for the whole dataset, which can well mine the global structure information. Based on GCNN, the proposed algorithm constructs a neighborhood graph for the training samples in each training batch, so as to well mine the geometric structure between samples in each training batch, which belongs to the local structure information. Therefore, the proposed algorithm considers both the global structure information and the local structure information simultaneously, which can better mine the geometric structure of original data and further improve the performance of feature extraction.

3.3. Discriminant regularization

Discriminant information is the feature that can help to distinguish samples from other samples. In addition to the geometric structure information of data, the discriminant information of data is also important for feature selection algorithm. Ignoring the discrimination information of the data will lead to the failure to achieve better feature selection effect. Previous studies have shown that discriminant information can be used to improve the performance of feature selection algorithm (Zeng et al., 2016; Yi et al., 2011). Li et al. proposed a discriminant orthogonal nonnegative matrix factorization algorithm (Li et al., 2014), which preserves the local manifold structure and global discriminant information. Orthogonal constraint is introduced to control the sparsity of data representation. Shang et al. proposed a non negative spectral clustering algorithm based on global discriminant, which retains the global discriminant structure and geometric structure of data at the same time. The global discriminant model is kernel processed, so that the algorithm can be effectively used in nonlinear datasets. Experiments show that the strategy improves the accuracy of clustering (Shang et al., 2016b). Du et al. proposed a feature selection algorithm based on local and global discriminant learning, which uses local discriminant information, global discriminant information and geometric structure information at the same time (Du et al., 2013). Experiments show that the algorithm can also effectively select representative features. Dornaika et al. proposed a local discriminant embedding algorithm. The most relevant and discriminant features of face image are captured by integrates feature selection to improve the accuracy (Dornaika et al., 2020).

Recent studies have found that, similar to the geometric structure of original data, the discriminant information is also beneficial to improve the performance of feature extraction. Therefore, our algorithm introduces the discriminant regularization to have better discriminant ability and processing ability of data outside the samples. A local discriminant model is introduced for the training batch $\mathbf{X}_i=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_{n_i}]\in\Re^{d\times n_i}$. A label matrix is defined for the current training batch $\mathbf{T}_i=[\mathbf{t}_1,\mathbf{t}_2,\ldots,\mathbf{t}_c]\in\{0,1\}^{n_i\times c}$, where c is the number of classes of training samples, and $\mathbf{t}_j\in\{0,1\}^{n_i\times 1}$ is the label indicator vector, if $\mathbf{x}_i$ belongs to the jth class, then the ith element in $\mathbf{t}_j$ is $t_{ij}=1$, and all other elements are 0. Since most datasets are unbalanced in sample sizes, instead of directly using the label matrix $\mathbf{T}_i$, it is normalized in a similar way as in Stella and Shi (2003) and define a weighted label matrix $\mathbf{F}_i=\mathbf{T}_i(\mathbf{T}_i^T\mathbf{T}_i)^{-1/2}=\left[\frac{\mathbf{t}_1}{\|\mathbf{t}_1\|_2},\frac{\mathbf{t}_2}{\|\mathbf{t}_2\|_2},\ldots,\frac{\mathbf{t}_c}{\|\mathbf{t}_c\|_2}\right]\in\Re^{n_i\times c}$. In order to mine the discriminant information in the data, a total scatter matrix $\mathbf{V}_{it}$ is introduced, a between-cluster scatter matrix $\mathbf{V}_{ib}$ and a within-cluster scatter matrix $\mathbf{V}_{iw}$ in the local discriminant regularization, which are defined as follows:

$$\mathbf{V}_{it}=\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^T,\tag{5}$$

$$\mathbf{V}_{ib}=\tilde{\mathbf{X}}_i\mathbf{F}_i\mathbf{F}_i^T\tilde{\mathbf{X}}_i^T,\tag{6}$$

$$\mathbf{V}_{iw}=\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^T\text{-}\tilde{\mathbf{X}}_i\mathbf{F}_i\mathbf{F}_i^T\tilde{\mathbf{X}}_i^T,\tag{7}$$

where $\tilde{\mathbf{X}}_i=\mathbf{X}_i\mathbf{U}_{n_i}$ is a centralized matrix of training batch, $\mathbf{U}_{n_i}=\mathbf{I}_{n_i}-(1/n_i)\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T\in\Re^{n_i\times n_i}$ is a centralized matrix, $\mathbf{1}_{n_i}$ is a n_i-dimensional vector, whose elements are all 1.

From Yang et al. (2010), Mika et al. (1999) and Yang et al. (2011), in order to achieve a better division of the data with the discriminant information, the distance between the data in different classes needs to be made as large as possible, and the distance between the data in the same class as small as possible. The distance between the data in different classes can be represented by the between-cluster scatter matrix $\mathbf{V}_{ib}$, and the distance between the data in the same class can be represented by the within-cluster scatter matrix $\mathbf{V}_{iw}$. Therefore, the between-cluster scatter matrix $\mathbf{V}_{ib}$ needs to be maximized, and minimize within-cluster scatter matrix $\mathbf{V}_{iw}$ or the total scatter matrix $\mathbf{V}_{it}$. Inspired by Fisher discriminant analysis (Mika et al., 1999), this problem can be transformed into the following objective function:

$$\begin{aligned}
\mathbf{F}_i^*&=\underset{\mathbf{F}_i}{\operatorname{argmax}}\,Tr[(\mathbf{V}_{it}+\mu\mathbf{I}_d)^{-1}\mathbf{V}_{ib}]\\
&=\underset{\mathbf{F}_i}{\operatorname{argmax}}\,Tr[(\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^T+\mu\mathbf{I}_d)^{-1}\tilde{\mathbf{X}}_i\mathbf{F}_i\mathbf{F}_i^T\tilde{\mathbf{X}}_i^T]\\
&=\underset{\mathbf{F}_i}{\operatorname{argmax}}\,Tr[\mathbf{F}_i^T\tilde{\mathbf{X}}_i^T(\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^T+\mu\mathbf{I}_d)^{-1}\tilde{\mathbf{X}}_i\mathbf{F}_i],
\end{aligned}\tag{8}$$

where $\mathbf{I}_d$ is the identity matrix, $\mu>0$ is the regular parameter, which is fixed at 10^4 in this paper. For convenience, the maximization problem

in Eq. (8) can be transformed into the minimization problem in Eq. (9) by adding a regular term $Tr(\mathbf{F}_i^T \mathbf{U}_{n_i} \mathbf{F}_i)$:

$$\mathbf{F}_i^* = \underset{\mathbf{F}_i}{\operatorname{argmin}} Tr[\mathbf{F}_i^T \mathbf{U}_{n_i} \mathbf{F} - \mathbf{F}_i^T \tilde{\mathbf{X}}_i^T (\tilde{\mathbf{X}}_i \tilde{\mathbf{X}}_i^T + \mu \mathbf{I}_d)^{-1} \tilde{\mathbf{X}}_i \mathbf{F}_i]. \tag{9}$$

According to Lemma 1 in Mika et al. (1999), the minimization problem in Eq. (9) can be rewritten as follows:

$$\mathbf{F}_i^* = \underset{\mathbf{F}_i}{\operatorname{argmin}} Tr\{\mathbf{F}_i^T [\mathbf{U}_{n_i} (\tilde{\mathbf{X}}_i^T \tilde{\mathbf{X}}_i + \mu \mathbf{I}_d)^{-1} \mathbf{U}_{n_i}] \mathbf{F}_i\}. \tag{10}$$

3.4. Cross-entropy error

Construct a graph convolutional neural network model by stacking multiple convolutional layers in the form of Eq. (10), each layer followed by point-wise nonlinearity. Now, limit the layer-wise convolution operation to $K = 1$, which is linear with regard to $\mathbf{L}$, so the graph Laplacian spectrum is a linear function.

In this linear formula, approximate $\lambda_{\max} \approx 2$ because neural network parameters can be expected to adapt to this scale change during training. Under these approximations, Eq. (10) can be simplified as follows:

$$g_{\theta'} * \mathbf{x} \approx \theta'_0 \mathbf{x} + \theta'_1 (\mathbf{L} - \mathbf{I}_n)\mathbf{x} = \theta'_0 \mathbf{x} - \theta'_1 \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}, \tag{11}$$

where θ'_0 and θ'_1 are two parameters. In practice, in order to avoid the overfitting and minimize the number of operations in each layer (such as matrix multiplications), further limit the number of parameters and get the following expression:

$$g_\theta * \mathbf{x} \approx \theta(\mathbf{I}_n + \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2})\mathbf{x}, \tag{12}$$

where θ'_0 and θ'_1 can be combined into a single parameter $\theta = \theta'_0 = -\theta'_1$. In addition, let $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $\mathbf{I}_n + \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ can be re-normalized to $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2}$. Therefore, the forward propagation model of the two-layer graph convolutional neural network can be expressed as follows:

$$\mathbf{Z} = f(\mathbf{X}, \mathbf{A}) = \text{softmax}(\hat{\mathbf{A}} \text{Relu}(\hat{\mathbf{A}} \mathbf{X} \mathbf{W}^{(0)}) \mathbf{W}^{(1)}), \tag{13}$$

where $\mathbf{W}^{(0)} \in \Re^{C \times H}$ is the weight matrix between the input layer and the hidden layer, H is the number of feature maps in the hidden layer, and $\mathbf{W}^{(1)} \in \Re^{H \times M}$ is the weight matrix between the hidden layer and the output layer. The softmax activation function is defined as: $\text{softmax}(x_i) = \frac{1}{z} \exp(x_i)$, $z = \sum_i \exp(x_i)$. The cross-entropy error is defined as follows:

$$-\sum_{l \in y_l} \sum_{m=1}^{M} Y_{lm} In Z_{lm}, \tag{14}$$

where y_l is the sample set with labels.

Based on the above ideas, GCNN with the geometric and discrimination information are combined to form the proposed algorithm GDGCNN. GDGCNN considers the local graph regularization, the discriminant regularization and the cross entropy error generated in the GCNN forward propagation simultaneously to form the total error function of our algorithm. The weight matrices in the neural network with the full dataset are trained by performing batch gradient descent for each training iteration. The algorithm is feasible as long as the dataset is suitable for memory. Therefore, the total error of GDGCNN is shown in Fig. 2, where $\mathbf{G}_i$ and $\mathbf{S}_i$ denote the local graph regularization, and the discriminant regularization of each batch, respectively, $i = [1, 2, \ldots, n_i]$. **CE** denotes the cross entropy error generated in the GCNN forward propagation.

The overall network structure is composed of two network layers in this paper. The input of the first layer is a sparse matrix with dimension $Numberofsamples \times Numberoffeatures$. Firstly, some values in the sparse matrix are randomly deleted with a probability of 0.5. Then the input data is output by graph convolution operation. The matrix with dimension $Numberofsamples \times 16$ is obtained by using the ReLu activation function. The input of the second network layer is the $Numberofsamples \times 16$ of the output of the upper layer. Some values in the sparse matrix are randomly deleted with a probability of 0.5. Then the input data is output through graph convolution operation. Finally, the output matrix with dimension Number of samples × Number of classes is obtained. The proposed algorithm is shown in Algorithm 1.

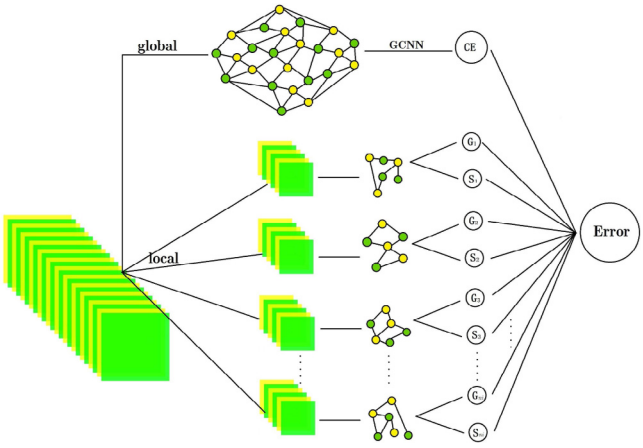


Fig. 2. The total error of GDGCNN.

Algorithm 1 Procedure of GDGCNN.

Require: Data sets,
1: Construction of model calculation chart;
2: Initialize model parameters;
3: $i = 1$;
4: **while** $i < epochs$ **do**
5: Prepare training data X and label T;
6: Calculate the predicted value P according to X;
7: Calculate the Cross-entropy Error according to the predicted value P and label T;
8: Calculate the L2 loss according to the model weight;
9: Calculate the total loss;
10: Update the model parameters according to the Cross-entropy Error;
11: $i = i + 1$;
12: **end while**
13: Prepare test data;
14: Calculate the predicted values according to the test data;
15: Determine the classes of samples according to the predicted values;
16: Output: Classes.

4. Experiments and analysis

In order to verify the performance of the proposed GDGCNN in data representation and classification,it is compared with the state-of-the-art models on lots of datasets.

4.1. Evaluation metrics

In this paper, two common evaluation metrics are used to compare the classification performance achieved by each algorithm: accuracy (ACC) and F1-Score. Before defining these two evaluation metrics, the results of the binary classification problem are explained as shown in Table 1.

As can be seen from Table 1, there are four classification results: (1) True Positive (TP), if a sample is positive and predicted to be positive; (2) False Negative (FN), if a sample is positive but predicted to be negative; (3) False Positive (FP), if a sample is negative but predicted to be positive; (4) True Negative (TN), if a sample is negative and predicted to be negative.

Table 1
Description of classification results.

		Actual Results		
		1	0	total
Predictive Results	1	True Postive (TP)	False Postive (FP)	Predictive Positive (PP)
	0	False Negative (FN)	True Negative (TN)	Predictive Negative (PN)
	total	Actual Positive (AP)	Actual Negative (AN)	All

Table 2
Experimental image datasets.

	Dimension	Train/Test	Classes
MNIST	28×28	60,000/10,000	10
Fashion	28×28	60,000/10,000	10
CIFAR10	$32 \times 32 \times 3$	50,000/10,000	10

Table 3
Experimental citation networks datasets.

	Nodes	Edges	Features	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
Pubmed	19,717	44,338	500	3

Table 4
The classification results of GDGCNN with different nearest neighbors k.

k	Accuracy-mean(%)			F1-Score-mean(%)		
	MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10
2	99.36	**96.99**	30.26	99.13	**96.58**	30.11
5	**99.56**	96.85	63.74	**99.35**	96.48	62.53
8	99.48	96.70	**74.48**	99.27	96.28	**73.29**

Accuracy(**ACC**) : The ratio of the number of samples correctly classified by the classifier to the total number of samples, so the calculation formula is as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} = \frac{\mathrm{TP+TN}}{\mathrm{AP+AN}} = \frac{\mathrm{TP+TN}}{\mathrm{PP+PN}}. \tag{15}$$

Precision: The ratio of True Postive to Predictive Positive, which is calculated as TP/(TP+FP)=TP/PP, so *Precision* is more concerned with False Postive (FP).

Recall: The ratio of True Postive to Actual Positive, which is calculated as TP/(TP+FN)=TP/AP, so *Recall* is more concerned with False Negative (FN).

F1 – Score : The harmonic mean of the precision and the recall, the calculation formula is as follows:

$$\mathrm{F1} = \frac{2 \cdot \mathit{Precision} \cdot \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP+FN+FP}}. \tag{16}$$

F1-Score is a comprehensive metric of *Precision* and *Recall*, so *Precision* and *Recall* are no longer calculated separately. In this experiment, accuracy (ACC) and F1-Score are taken as the evaluation metrics.

4.2. Compared algorithms

In order to show the performance of the proposed GDGCNN, it is compared with some common networks, including CNN (LeCun et al., 1998), GCNN (Defferrard et al., 2016), and three state-of-the-art networks 1stGCNN (Kipf and Welling, 2016), StoGCNN (Chen et al., 2017) and DGI (Veličković et al., 2018). In addition, in order to show the performance of the local graph regularization and the discriminant regularization in GDGCNN, two deformation algorithms based on GCNN are introduced. The two algorithms are summarized as follows: GGCNN introduces the local graph regularization into GCNN separately to show the effect of the local graph regularization in GDGCNN; DGCNN introduces the discriminant regularization into GCNN separately to show the effect of the discriminant regularization in GDGCNN.

4.3. Datasets

The compared algorithms mentioned in Section 4.2 on the datasets (Defferrard et al., 2016; Kipf and Welling, 2016; Chen et al., 2017) are tested. The results are shown in Tables 2 and 3.

(1) MNIST

The MNIST handwritten digital dataset is a classical deep learning entry-level dataset. MNIST contains 70,000 gray-scale images with size 28×28, in which 60,000 training images and 10,000 test images can be divided into 10 classes.

(2) Fashion

The Fashion dataset is also called Fashion-MNIST because it has the same image size and the same structure of training and test segmentation as MNIST. Fashion is an image dataset for clothing recognition, which consists of 60,000 training images and 10,000 test images of 10 classes, and each datum is a 28×28 grayscale image.

(3) CIFAR10

The CIFAR10 dataset contains 60,000 color images of 10 classes, 6,000 images per class, and each image is with size 32×32. In the 60,000 color images, 50,000 are training images and 10,000 are test images.

For each of these three image datasets, 5000 data from its training data are selected as the validation data.

(4) Cora

The Cora dataset contains 2708 machine learning publications divided into seven classes. The citation network contains 5429 links. The dictionary contains 1433 unique words. Each publication in Cora dataset is set as a 0/1-valued word vector, which indicates the absence/presence of the corresponding word from the dictionary.

(5) Citeseer

The Cora dataset contains 3327 scientific publications divided into six classes. The citation network contains 4732 links. The dictionary contains 3703 unique words. Each publication in Citeseer dataset is set as a 0/1-valued word vector, which indicates the absence/presence of the corresponding word from the dictionary.

(6) Pubmed

The Pubmed dataset contains 19,717 diabetes-related publications divided into seven classes. The citation network contains 44,338 links. The dictionary contains 500 unique words. Each publication in Pubmed dataset is described by a term frequency–inverse document frequency (TF–IDF) vector.

4.4. Parameter analysis

Classification experiments are run on some related algorithms and the proposed GDGCNN. For the sake of fairness, we use the same network framework to test these algorithms. The parameters for the other compared algorithms are stated in detail in LeCun et al. (1998), Defferrard et al. (2016), Kipf and Welling (2016) and Chen et al. (2017). Here we mainly introduce the parameters for GDGCNN. The classification results of GDGCNN with the main parameters the nearest neighbors k and batch size s are shown in Tables 4 and 5. The best classification results for each dataset are marked in bold.

From the classification results of GDGCNN with different nearest neighbors k in Table 4, it can be seen that different k should be chosen on different datasets to obtain the best classification results. In addition,

Table 5
The classification results of GDGCNN with different batch sizes s.

s	Accuracy-mean(%)			F1-Score-mean(%)		
	MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10
50	99.09	**96.70**	**74.48**	98.88	**96.28**	**73.29**
100	**99.48**	95.71	73.62	**99.27**	95.19	72.31
150	99.24	94.86	73.45	99.03	94.51	72.16

GDGCNN is not sensitive to k on MNIST and Fashion, and sensitive to k on CIFAR10.

From the classification results of GDGCNN with different batch sizes s in Table 5, it can be seen that GDGCNN is not sensitive to s and has strong robustness on these three image datasets.

4.5. Classification results and analysis

For the sake of fairness, the similar network structure is used for several compared methods. C32/C64 represents a convolutional layer with 32/64 feature maps, and GC32/GC64 represents a graph convolutional layer with 32/64 feature maps. P4 represents a pooling layer with of size and stride 4. FC512 represents a fully connected layer with 512 hidden units.

In addition, the same number of iterations on the same dataset are set for different algorithms. Table 6 shows the accuracy (ACC) results of five compared algorithms on the validation sets of the three image datasets in Table 2. Table 7 shows the F1-Score results of five compared algorithms on the validation sets of these three image datasets. the best classification results for each dataset are marked in bold and the second are marked underlined. Tables 6 and 7 numerically show the classification results of five compared algorithms on the validation sets of the three image datasets in Table 2. All these experimental results confirm the following conclusions:

- Overall, the proposed GDGCNN performs best in terms of both ACC and F1-Score results. The performance of GDGCNN is much better than CNN and GCNN in terms of these two evaluation metrics, especially on CIFAR10.
- From the comparison between GCNN and GGCNN, it can be seen that GGCNN is better than GCNN in terms of both ACC and F1-Score results. It shows that the introduction of the local graph regularization can effectively mine the local structure information of original data, which is conducive to improving the classification performance of classification algorithms.
- From the comparison between GCNN and DGCNN, it can be seen that DGCNN is better than GCNN in terms of both ACC and F1-Score results. It shows that the introduction of the discriminant regularization can effectively mine the local discriminant information of original data, which is conducive to improving the classification performance of the classification algorithms.
- From the comparison between GDGCNN and the other four algorithms, it can be seen that GDGCNN which considers both local structure information and discriminant information, can fully exploit the potential information of original data and obtain better classification results.

In order to show the classification performance of these five compared algorithms on the validation set of the three datasets more intuitively, the classification accuracy are plotted in Fig. 3, while F1-Score figure which is similar to Fig. 3, is no longer plotted. In Fig. 3, the abscissa represents the number of iterations, and the ordinate represents the classification accuracy (ACC) results.

Fig. 3 visually shows the classification accuracy of five compared algorithms on the validation sets of the three datasets. It can be seen from Fig. 3, as the increase of the number of iterations, the ACC results of five compared algorithms are on the rise. In addition, the ACC curve of the proposed GDGCNN is marked with a solid red line, which is above the other curves in most positions in Fig. 3(a), and all above the other curves in Fig. 3(b) and (c). This means that GDGCNN has the best classification performance on the three datasets. In addition, the ACC curves of GGCNN and DGCNN are mostly higher than the ACC black dotted line of GCNN in Fig. 3(a), and all above it in Figs. 3(b) and 3(c). It is shown that both the local structure information and the discriminant information can effectively mine the potential information of original data and improve the performance of the classification algorithms.

In addition, a classification comparison experiment of 1stGCNN (Kipf and Welling, 2016), StoGCNN (Chen et al., 2017), DGI (Veličković et al., 2018) and GDGCNN on three citation networks datasets (Chen et al., 2017) is performed in Table 3. The ACC results show as shown in Fig. 4.

It can be seen that the yellow one representing GDGCNN achieve the good classification performance on these three citation networks datasets. GDGCNN can achieve the best ACC results on Citeseer and Pubmed. For quantitative analysis of the ACC results, the ACC results are also presented in Table 8.

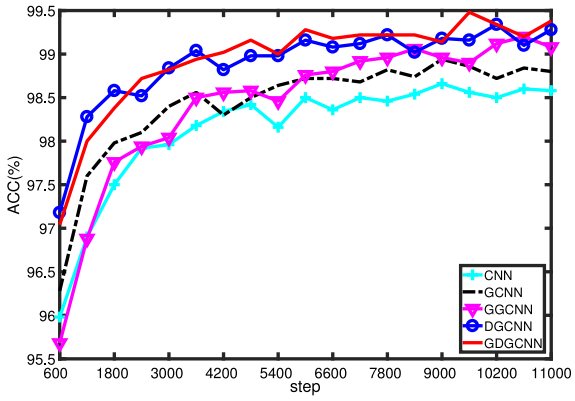
From Table 8, it can be seen that the classification results of GDGCNN are better than the other three state-of-the-art algorithms on these three citation networks datasets. The average ACC result of GDGCNN is 1.1%, 0.7%, 1.0% higher than 1stGCNN, StoGCNN and DGI respectively. Therefore, it indicates that GDGCNN can make full use of local structure information and discrimination information, which allows for the effective feature learning.

Table 6
The ACC results of five compared algorithms on the validation sets of the three datasets.

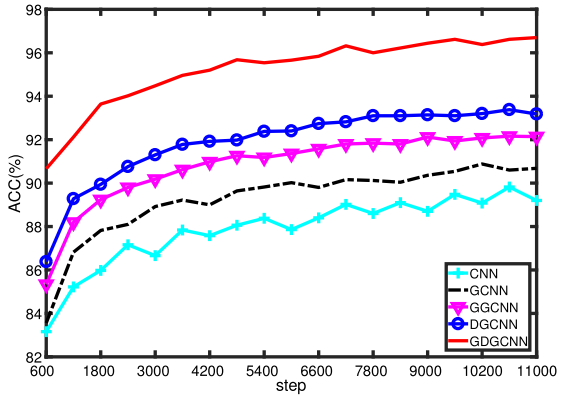
Model	Architecture	Accuracy-peak(%)			Accuracy-mean(%)		
		MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10
CNN	C32-P4-C64-P4-FC512	98.66	89.82	60.84	98.53	88.93	59.37
GCNN	GC32-P4-GC64-P4-FC512	98.94	90.88	63.98	98.78	90.32	62.22
GGCNN	GC32-P4-GC64-P4-FC512	99.20	92.16	67.08	98.98	91.88	66.13
DGCNN	GC32-P4-GC64-P4-FC512	<u>99.34</u>	<u>93.38</u>	<u>70.26</u>	<u>99.17</u>	<u>93.02</u>	<u>68.80</u>
GDGCNN	GC32-P4-GC64-P4-FC512	**99.48**	**96.70**	**74.48**	**99.27**	**96.28**	**73.29**

Table 7
The F1-Score results of five compared algorithms on the validation sets of the three datasets.

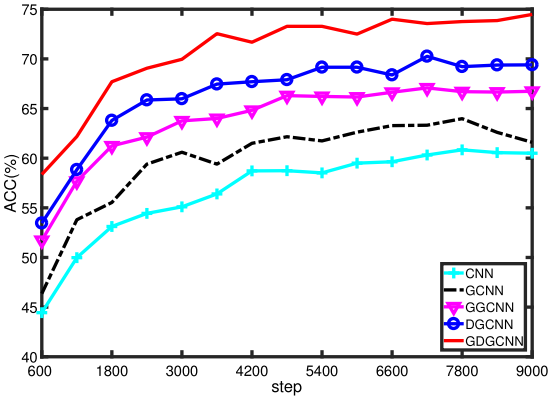
Model	Architecture	F1-Score-peak(%)			F1-Score-mean(%)		
		MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10
CNN	C32-P4-C64-P4-FC512	98.66	89.71	60.70	98.14	87.69	56.42
GCNN	GC32-P4-GC64-P4-FC512	98.94	90.85	63.99	98.78	90.19	62.04
GGCNN	GC32-P4-GC64-P4-FC512	99.20	92.05	67.11	98.98	91.70	65.95
DGCNN	GC32-P4-GC64-P4-FC512	<u>99.34</u>	<u>93.21</u>	<u>70.20</u>	<u>99.17</u>	<u>92.83</u>	<u>68.58</u>
GDGCNN	GC32-P4-GC64-P4-FC512	**99.48**	**96.60**	**74.24**	**99.27**	**96.09**	**72.99**



(a) MNIST



(b) Fashion



(c) CIFAR10

Fig. 3. The ACC results of five compared algorithms on the validation sets of the three datasets.

Table 8
The ACC results of four state-of-the-art algorithms on three citation networks datasets.

ACC(%)	Citeseer	Cora	Pubmed	Average
1stGCNN	70.3	81.5	79.0	76.9
StoGCNN	70.9	82.0	79.0	77.3
DGI	71.8	**82.3**	76.8	77.0
GDGCNN	**72.6**	82.0	**79.5**	**78.0**

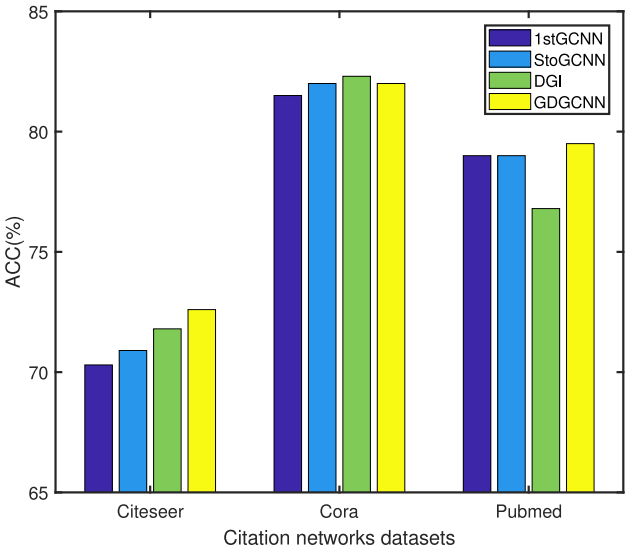


Fig. 4. The ACC results of four state-of-the-art algorithms on three citation networks datasets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

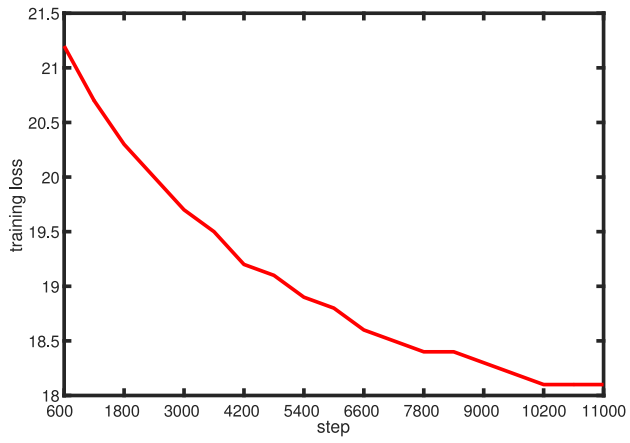
4.6. Convergence study

Here the convergence of the proposed GDGCNN on the three image datasets (MNIST, Fashion and CIFAR10) are mainly showed in Fig. 5. For each graph, the abscissa represents the number of iterations and the ordinate demotes the training loss.

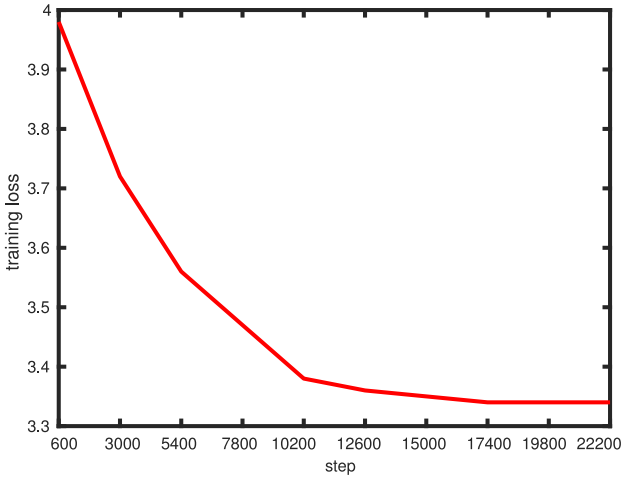
From Fig. 5, it can be seen that the training loss of GDGCNN has a decreasing trend on these three datasets, which can achieve convergence. The convergence speed on the first two datasets is slow, which needs more than 10,000 times to converge. On CIFAR 10, it can converge only about 3000 times. From the classification results in terms of the two evaluation metrics (ACC and F1-Score), it can be seen that the performance of GGCNN is better than that of GCNN, which indicates that the introduction of the local graph regularization can effectively mine the local structure information of original data and improve the classification performance of classification algorithms; the performance of DGCNN is better than that of GCNN, which indicates that the introduction of discriminant regularization can effectively mine the local discriminant information of original data and improve the classification performance of classification algorithms; GDGCNN has the best classification performance, which indicates that GDGCNN considering both local structure information and discriminant information can fully exploit the potential information of original data and improve the classification performance. In addition, from the classification results on three common citation networks datasets Cora, Pubmed, and Citeseer in detail, it can be seen that GDGCNN has the better classification performance than some state-of-the-art methods. From the convergence experiment, it can be seen that the training loss of GDGCNN has a decreasing trend and can achieve convergence on all three datasets. Although GDGCNN has some advantages over other algorithms, there are still some shortcomings, such as the longer computation time when constructing neighborhood graphs. Therefore, GDGCNN will be improved to further accelerate the construction of the neighborhood graph, and make it simpler, more practical and more efficient.

5. Summary and future work

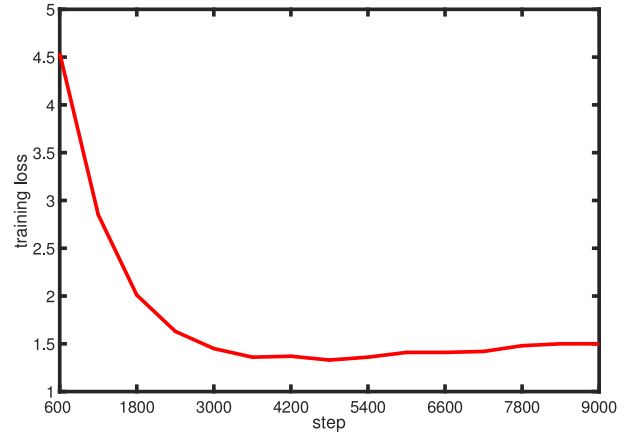
In recent years, with the rapid growth of data dimension and data volume, it continuously promotes the research and development of big data processing and data mining. The existing graph convolution network algorithms usually ignore the discrimination information and



(a) MNIST



(b) Fashion



(c) CIFAR10

Fig. 5. Convergence of GDGCNN on the validation sets of the three datasets.

the local structure differences of different samples. A Graph Convolutional Neural Network with Geometric and Discrimination information (GDGCNN) is proposed in this paper. GDGCNN is an effective graph convolutional neural network, which combines classical CNN with spectral theory to solve the graph-structured data. GDGCNN constructs different feature graphs for different training batches in the same dataset, which can not only solve the problem of ignoring the difference between the local structures of different samples in previous Graph Convolutional Neural Networks (GCNN), but also fully exploit the geometric structure of original data. GDGCNN can not only mine the graph structure data more effectively, but also make use of the local structure information and discriminant information in the original data at the same time. So it has better learning ability and discriminative ability, and can carry out more effective feature learning. GDGCNN integrates the traditional machine learning idea into the framework of graph convolution network, which has certain physical meaning and interpretability, and can further improve the performance of feature extraction.

CRediT authorship contribution statement

Ronghua Shang: Editing and revising the article, Supervision, Project administration. **Yang Meng:** Design (methodology), Investigation (performing experiments), Drafting the article. **Weitong Zhang:** Editing and revising the article. **Fanhua Shang:** Supervision, Guiding. **Licheng Jiao:** Supervision, Project administration, Funding acquisition. **Shuyuan Yang:** Supervision, Guiding.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by National Key R&D Program of China and the National Natural Science Foundation of China under Grants Nos. 61773304.

References

Cai, D., He, X., Han, J., 2007. SRDA: An efficient algorithm for large-scale discriminant analysis. IEEE Trans. Knowl. Data Eng. 20 (1), 1–12.

Cai, D., He, X., Han, J., 2010. Locally consistent concept factorization for document clustering. IEEE Trans. Knowl. Data Eng. 23 (6), 902–913.

Cai, D., He, X., Han, J., Huang, T.S., 2011. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (8), 1548–1560.

Chen, J., Zhu, J., Song, L., 2017. Stochastic training of graph convolutional networks with variance reduction. arXiv preprint arXiv:1710.10568.

De Una, D., Rümmele, N., Gange, G., Schachte, P., Stuckey, P.J., 2018. Machine learning and constraint programming for relational-to-ontology schema mapping. In: International Joint Conference on Artificial Intelligence. pp. 1277–1283.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems. pp. 3844–3852.

Dhillon, I.S., Guan, Y., Kulis, B., 2007. Weighted graph cuts without eigenvectors a multilevel approach. IEEE Trans. Pattern Anal. Mach. Intell. 29 (11).

Dornaika, F., Moujahid, A., Wang, K., Feng, X., 2020. Efficient deep discriminant embedding: Application to face beauty prediction and classification. Eng. Appl. Artif. Intell. 95, 103831.

Du, L., Shen, Z., Li, X., Zhou, P., Shen, Y.-D., 2013. Local and global discriminative learning for unsupervised feature selection. In: 2013 IEEE 13th International Conference on Data Mining. IEEE, pp. 131–140.

Hammond, D.K., Vandergheynst, P., Gribonval, R., 2011. Wavelets on graphs via spectral graph theory. Appl. Comput. Harmon. Anal. 30 (2), 129–150.

He, Y., Li, J., Song, Y., He, M., Peng, H., 2018. Time-evolving text classification with deep neural networks. In: International Joint Conference on Artificial Intelligence. pp. 2241–2247.

He, S., Schomaker, L., 2019. DeepOtsu: Document enhancement and binarization using iterative deep learning. Pattern Recognit. 91, 379–390.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Krishnan, R., Samaranayake, V., Jagannathan, S., 2018. A multi-step nonlinear dimension-reduction approach with applications to bigdata. Procedia Comput. Sci. 144, 81–88.

Kuang, Z., Yu, J., Li, Z., Zhang, B., Fan, J., 2018. Integrating multi-level deep learning and concept ontology for large-scale visual recognition. Pattern Recognit. 78, 198–214.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Li, P., Bu, J., Yang, Y., Ji, R., Chen, C., Cai, D., 2014. Discriminative orthogonal nonnegative matrix factorization with flexibility for data representation. Expert Syst. Appl. 41 (4), 1283–1293.

Liao, D., Liu, W., Zhong, Y., Li, J., Wang, G., 2018. Predicting activity and location with multi-task context aware recurrent neural network. In: International Joint Conference on Artificial Intelligence. pp. 3435–3441.

Luo, Z., Liu, L., Yin, J., Li, Y., Wu, Z., 2017. Deep learning of graphs with ngram convolutional neural networks. IEEE Trans. Knowl. Data Eng. 29 (10), 2125–2139.

Meng, Y., Shang, R., Jiao, L., Zhang, W., Yang, S., 2018a. Dual-graph regularized non-negative matrix factorization with sparse and orthogonal constraints. Eng. Appl. Artif. Intell. 69, 24–35.

Meng, Y., Shang, R., Jiao, L., Zhang, W., Yuan, Y., Yang, S., 2018b. Feature selection based dual-graph sparse non-negative matrix factorization for local discriminative clustering. Neurocomputing 290, 87–99.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.-R., 1999. Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop. pp. 41–48.

Ptucha, R., Such, F.P., Pillai, S., Brockler, F., Singh, V., Hutkowski, P., 2019. Intelligent character recognition using fully convolutional neural networks. Pattern Recognit. 88, 604–613.

Pu, J., Zhou, W., Li, H., 2018. Dilated convolutional network with iterative optimization for continuous sign language recognition. In: International Joint Conference on Artificial Intelligence. pp. 885–891.

Shang, R., Liu, C., Meng, Y., Jiao, L., Stolkin, R., 2017. Nonnegative matrix factorization with rank regularization and hard constraint. Neural Comput. 29 (9), 2553–2579.

Shang, R., Meng, Y., Liu, C., Jiao, L., Esfahani, A.M.G., Stolkin, R., 2019b. Unsupervised feature selection based on kernel fisher discriminant analysis and regression learning. Mach. Learn. 108 (4), 659–686.

Shang, R., Meng, Y., Wang, W., Shang, F., Jiao, L., 2019a. Local discriminative based sparse subspace learning for feature selection. Pattern Recognit. 92, 219–230.

Shang, R., Wang, L., Shang, F., Jiao, L., Li, Y., 2021. Dual space latent representation learning for unsupervised feature selection. Pattern Recognit..

Shang, R., Wang, W., Stolkin, R., Jiao, L., 2016a. Subspace learning-based graph regularized feature selection. Knowl.-Based Syst. 112, 152–165.

Shang, R., Wang, W., Stolkin, R., Jiao, L., 2018. Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection. IEEE Trans. Cybern. 48 (2), 793–806.

Shang, R., Xu, K., Shangl, F., Pao, L., 2020. Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection. Knowl.-Based Syst. 187 (Jan.), 104830.1–104830.15.

Shang, R., Zhang, Z., Jiao, L., Wang, W., Yang, S., 2016b. Global discriminative-based nonnegative spectral clustering. Pattern Recognit. 55, 172–182.

Stella, X.Y., Shi, J., 2003. Multiclass spectral clustering. In: Null. p. 313.

Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D., 2018. Deep graph infomax. arXiv preprint arXiv:1809.10341.

Xie, S., Hu, H., Wu, Y., 2019. Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition. Pattern Recognit. 92, 177–191.

Xu, J., Han, J., Nie, F., Li, X., 2019. Multi-view scaling support vector machines for classification and feature selection. IEEE Trans. Knowl. Data Eng..

Yang, S., Li, L., Wang, S., Zhang, W., Huang, Q., 2017. A graph regularized deep neural network for unsupervised image representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1203–1211.

Yang, Y., Shen, H.T., Ma, Z., Huang, Z., Zhou, X., 2011. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In: International Joint Conference on Artificial Intelligence Proceedings-International Joint Conference on Artificial Intelligence, Vol. 22. No. 1. pp. 1589–1594.

Yang, Y., Xu, D., Nie, F., Yan, S., Zhuang, Y., 2010. Image clustering using local discriminant models and global integration. IEEE Trans. Image Process. 19 (10), 2761–2773.

Yi, Y., Shen, H.T., Ma, Z., Zi, H., Zhou, X., 2011. L21-norm regularized discriminative feature selection for unsupervised learning. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two.

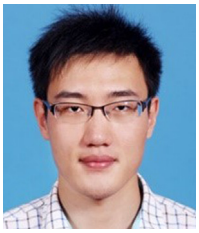
Zeng, Z., Wang, X., Zhang, J., Wu, Q., 2016. Semi-supervised feature selection based on local discriminative information. Neurocomputing 173, 102–109.

Zhang, Z., Chen, D., Wang, J., Bai, L., Hancock, E.R., 2019. Quantum-based subgraph convolutional neural networks. Pattern Recognit. 88, 38–49.

Zhao, Z., Kumar, A., 2019. A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features. Pattern Recognit. 93, 546–557.

Zhuang, C., Ma, Q., 2018. Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 499–508.

Ronghua Shang (M'09) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003 and 2008, respectively. She is currently a Professor with Xidian University. Her current research interests include optimization problems, evolutionary computation, image processing, and data mining.

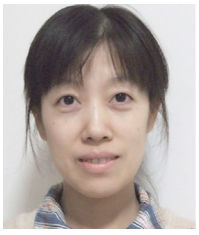


Yang Meng received the Ph.D. degree in intelligent information processing with Xidian University, Xi'an, China, in 2020. He is currently an algorithm engineer at Huawei Technologies Co., Ltd. His current research interests include machine learning and AIOps in telecom networks.

Weitong Zhang received the B.E. degree from the School of Electronic and Information Engineering, Changchun University of Science and Technology, Changchun, China, in 2013, the M.S. degree from the School of Electronics and Communication Engineering, Xidian University, Xi'an, China, in 2017, the Ph.D. degree from the School of Circuits and Systems, Xidian University, Xi'an, China, in 2021, where she is currently a lecturer with Xidian University. Her current research interests include complex networks, intelligent optimization, and deep learning.

Fanhua Shang (M'14) received the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2012. From 2016 to 2018, he was a Research Associate with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, where he was a Post-Doctoral Research Fellow. From 2012 to 2013, he was a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. He is currently a Professor with the School of Artificial Intelligence, Xidian University. His current research interests include machine learning, data mining, pattern recognition, and computer vision.

Licheng Jiao (SM'89–F'17) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. From 1990 to 1991, he was a Post-Doctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an. He has been a Professor with the School of Electronic Engineering, Xidian University. He is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education of China, Xidian University. He has led 40 major scientific research projects. He has published more than 20 monographs and a hundred articles in international journals and conferences. His current research interests include image processing, natural computation, machine learning, and intelligent information processing. Dr. Jiao is a member of the IEEE Xi'an Section Executive Committee and the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, the Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.



Shuyuan Yang (M'07–SM'14) received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in circuits and systems from Xidian University, Xi'an, China, in 2000, 2003, and 2005, respectively. She is currently a Post-Doctoral Fellow with the Institute of Intelligent Information Processing, Xidian University. Her current research interests include intelligent signal processing, machine learning, and image processing.