Deep learning model construction for a semi-supervised classification with feature learning

Sridhar Mandapati¹ · Seifedine Kadry² · R. Lakshmana Kumar³ · Krongkarn Sutham⁴ · Orawit Thinnukool⁵

Received: 12 July 2021 / Accepted: 10 December 2021
© The Author(s) 2022

Abstract

Several deep models were proposed in image processing, data interpretation, speech recognition, and video analysis. Most of these architectures need a massive proportion of training samples and use arbitrary configuration. This paper constructs a deep learning architecture with feature learning. Graph convolution networks (GCNs), semi-supervised learning and graph data representation, have become increasingly popular as cost-effective and efficient methods. Most existing merging node descriptions for node distribution on the graph use stabilised neighbourhood knowledge, typically requiring a significant amount of variables and a high degree of computational complexity. To address these concerns, this research presents DLM-SSC, a unique method semi-supervised node classification tasks that can combine knowledge from multiple neighbourhoods at the same time by integrating high-order convolution and feature learning. This paper employs two function learning techniques for reducing the number of parameters and hidden layers: modified marginal fisher analysis (MMFA) and kernel principal component analysis (KPCA). The MMFA and KPCA weight matrices are modified layer by layer when implementing the DLM, a supervised pretraining technique that doesn't require a lot of information. Free measuring on citation datasets (Citeseer, Pubmed, and Cora) and other data sets demonstrate that the suggested approaches outperform similar algorithms.

Keywords Deep learning · Feature learning · Semi-supervised classification · Deep architecture

Introduction

Deep learning has revealed to be profoundly productive in a comprehensive variety of jobs, including speech recognition [1], natural language processing [2], and computer vision [3]. It presents us with the aesthetically pleasing durable framework for extracting useful characteristics and excellent tensor information from a Euclidean structure. However, there is a vast amount of data in the real world, such as social interactions [4], bio-molecules [5], and journal citations [6] that can be naturally modelled by graphs. As a result, expanding the convolution network to graph data is crucial and interesting in improving graph-based activities. Several efforts to apply convolutions to graphs data have recently been made [7, 8].

A graph is a non-linear data form. Many other graphs exist in real life and can be abstracted into graph form, including communications infrastructure, process states, social networks, subway networks, and many others. It is critical to

✉ Orawit Thinnukool
orawit.t@cmu.ac.th

Sridhar Mandapati
mandapati12@gmail.com

Seifedine Kadry
skadry@gmail.com

R. Lakshmana Kumar
research.laksha@gmail.com

Krongkarn Sutham
krongkarn.s@cmu.ac.th

¹ Department of Computer Applications, R. V. R & J.C College of Engineering, Chowdavaram, Guntur, India

² Faculty of Applied Computing and Technology (FACT), Noroff University College, Kristiansand, Norway

³ Centre of Excellence for Artificial Intelligence and Machine Learning, Hindusthan College of Engineering and Technology, Coimbatore, India

⁴ Department of Emergency Medicine, Faculty of Medicine, Chiang Mai University, Chiang Mai, Thailand

⁵ Research Group of Embedded Systems and Mobile Application in Health Science, College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand

acquire appropriate graph modelling from node and edge attributes and the topological graph formation to exploit the rich knowledge of graph-structured information. In response to an effectiveness of deep learning models on grid-structured information, a slew of graph neural network [9] models has been suggested. Graph convolution [10] consolidates and regenerates characteristic Vectors from the local environment, a key component of neural graph networks. Graph neural networks acquire a node demonstration at the low dimension subspace where neighbouring nodes at a graph have the exact representation by combining graph structural features with node attributes based on Laplace smoothing [11]. Researchers have suggested graph level pooling systems [12] which compact a representation of the node into a global function vector for study a demonstration for a whole graph. In several subsequent graphs processing jobs, for example graph classification [14], node classification [13], etc., the graph or node demonstration acquired through neural graph network have attained up to date efficiency.

Node classification [6, 13] is a crucial activity on graph data that primarily determines node classes based on node characteristics and graph topology. GCNs are frequently used to define and recognise the description of each node. Because of a large quantity of graph information with high labelling costs, some node classification options are semi-supervised. For instance, at the citation system, where node describe edges and publications depict citation connections, classifying semi-supervised nodes is to decide the mark of each publication based on a limited amount of labelled data [15]. This paper proposes a semi-supervised node classification with feature learning.

Numerous character recognition approaches are presented for acquire a low dimension interpretation of high dimension information to overcome a curse of dimensionality. Most of them can be educated with limited data and use closed-form solutions or convex optimisation as their primary learning strategies. One of the supervised feature learning techniques using a framework for graph embedding is marginal Fisher analysis (MFA) [16]. It utilises the penalty graph for inter-class separability also the intrinsic graph for intraclass compactness. By using generalised eigenvalue decomposition, the best MFA solution can be found.

This paper suggests a novel deep learning solution built on feature learning for address existing DL algorithm problems while also incorporating the benefits of feature learning. To initialise this deep architecture, two function learning MFA and KPCA [17] layers are used instead of random initialisation. First, this paper creates a non-linear weight matrix used for plan a input information into the high dimensional space, thus increasing an architecture's power. The MFA and KPCA layers are then used to layer-by-layer (LbL) study a lower-dimensional demonstrations of the information. Lastly, a final feature layer is attached to the softmax

layer. The proposed work is compared with deep learning models on graph dataset (include PubMed, Citeseer, and Cora) and other dataset. Results demonstrate that DLM-SSC outperforms other deep learning models.

This paper's significant participation is as follows:

- A novel deep architecture called DLM is suggested for semi-supervised node labelling. The first secret layer of DLM has twice as many neurons as the input layer. Four layers of attribute learning methods are then used to study low-dimensional demonstrations of input information. Finally, nodes are classified using a multiclass classifier.
- Two feature learning methods, MMFA and KPCA, are used to reduce the parameters. Three hidden layers are added to DLM based on these selected features.
- On semi-supervised classification tasks, comprehensive analyses in citation and publication datasets (Cora, Pubmed, Citeseer) show DLM-SSC perform better than benchmark techniques in contexts of precision.

The rest of the paper is arranged like this: the next section examines associated studies on deep learning techniques, node classification, and feature learning in a brief literature review. The key methods are discussed in the following section. Furthermore, the next section delves the proposed strategy. Following section defines the evaluation metrics and introduces the experimental design. In the next section, an experimental outcomes are evaluated, also finally, the last section provides the conclusion of this paper.

Related works

This segment examines some previous research that applies to this thesis. Node classification, function learning, and deep learning architecture are also included.

Node classification

Gong and Ai [13] recommend a block Adaptive Graph Convolutional Neural Networks for efficiently study descriptions of a specific node for node ordering jobs. This analysis develops a neighbourhood adaptive kernel, a convolutional kernel abstracted from a spreading process, for acquire also combine appropriate neighbourhood node data for all nodes in a very targeted manner. N-GCN is a proposal by Abu-El-Haija et al. [18] for semi-supervised node classification using multi-scale graph convolution. By practising individual situations of GCNs over node sets identified at distinct intervals in irregular positions, it seeks an aggregate of the problem outputs that optimises the distribution goal. The DEMO-Net architecture is proposed by Wu et al. [19]. It is a generalised

graph neural network model built on the premise which nodes by a corresponding degree worth have a similar graph convolution. A multi-task graph convolution feature with a specific degree value is provided to help explain the node demonstration.

Liu et al. [20] suggest a different higher-order GCN with multi-scale community clustering to semi-supervised node labelling. MNPooling and high-order convolution are two primary categories. MNPooling proposes three knowledge aggregation approaches for integrating data from multiple neighbourhoods while preserving graph topology. In high-order convolution, weight sharing restricts the range of dimensions. [15] introduced a different GCN for semi-supervised data analysis on charts. This system represents a softmax, a discriminable capsule sheet, symmetrical grinding layers, coarsening layers, with SEGCN layers.

Li and Pi [21] proposes a deep neural network (DNN) technique for node labeling called DNNNC at the sense of DL. Create a positive pointwise reciprocal knowledge matrix from an adjacent matrix. The information is feed into a DNN with deep stacked sparse auto-encoders with a softmax layer that is very well equipped for node labelling in the DNN framework and can attain a node interpretation whilst encodes rich non-linear semantic with structural data. Molokwu et al. [22] proposed a new approach for processing and retrieving valuable data from OSN systems to assist at node categorisation also group identification activities. It uses an edge sampling technique to take advantage of social graph features by studying each actor's history concerning nearby nodes to create vector-space embeddings for each actor. Madhawa et al. [23] looked at using an adaptive learning algorithm to increase node classification performance on attributed graph labels.

Li et al. [43] present a novel semi-supervised learning technique combining dynamic graph learning and self-paced learning. The graph learning approach proposed by Kang et al. [44], preserves both the local and global structure of data. The global structure is captured using the self-expressiveness of samples, while the local structure is respected using an adaptive neighbour approach.

Feature learning

In feature learning models, dimensionality compression is crucial because it eases solving problems like visualising high dimension information also minimising the dimensionality curse. There are three types learning: supervised (models are trained using labeled data), unsupervised (trained using unlabelled data) and semi-supervised (combine label and unlabelled data). For semi-supervised learning, Liu et al. [29] suggest a novel hierarchical, streamlined graph-based coarse feature extraction method. This method combines learning with local structure, sparse approximation, also

mark spread to improve information dimension reduction. A PCA definition of minimising least squares calculation faults are regularised by graph drawing, which combines different local manifold embedding approaches into a generalised scheme to preserve global with regional low dimension subspace [30]. Unlike a PCA method, this method's simplified least squares solution believes in information sharing and the instance penalty at all data points. Gou et al. [31] suggest a dimensionality reduction technique based on a graph called Discriminative globality also locality protecting graph implanting through scheming good locality and globality keeping graph buildings. Bidirectional edges weights are recently described in the constructed graphs, taking into account mathematical sharings of all point of an edge with a class bias. Rajabzadeh et al. [32] suggest a novel supervised dimensionality reduction approach that optimises the recently developed with proficient goal task for study each class changes.

In correlation to an individual conversion, this strategy catches more discriminative knowledge from each group of data. Many feature learning models offer an efficient solution for dimensionality reduction implementations. Feature learning models can underperform for vast, complicated issues. The advantages of deep architectures with feature learning are combined to offer the new supervised node classification approach for the DL algorithm.

Deep learning architecture

There are a few tasks on deep architecture-based feature learning models. To address the scene recognition process, Yuan et al. [24] presented the enhanced multi-layer learning technique. This model learns all visual identification features in the not supervised way. Kejani et al. [25] present a model that improves GCN label propagation. It is made up of two terms: supervised and unsupervised. A supervised word enforces the appropriate word among an expected with the identified label. An unsupervised word requires that a scheduled label of all data samples be smooth. Zhu et al. [26] suggest the deep architecture and flexible embedding based on a graph that digs deep into the data's structural detail. A multiple geometrical organisation of information is included in this deep architecture.

Trigeorgis et al. [27] suggested a deep semi-Non-negative matrix factorisation that can acquire hidden interpretations from unknown parameters in a dataset. This model has been learned to consider low-dimensional representations, which are more likely to cluster. Ngiam et al. [28] created an unsupervised paradigm for understanding feature representations through various modalities called an adaptive architecture. They believed that multi-modality representation learning is superior to single modality representation learning, citing intense video and audio data sets as evidence. Niloy et al.

[45] suggests deep dive for improving classification performance.

Preliminaries

Node classification

When data is interpreted as a graph, node classification plays a vital role in learning problems. The nodes in graph G are called V nodes, and the edges connecting them are called E edges. A graph's edges can also be directional. In functional implementations, for example, recommendation system [33], applied chemistry [34], with social network research [35], node classification is commonly used. The attributed graph $G = (V, E)$ and N nodes are provided like an adjacency matrix $\text{Adj}_{\text{Mat}} \in R^{N \times N}$ and the node attribute matrix $\text{Att}_{\text{Mat}} \in R^{N \times F}$ in a node classification problem. The number of attributes is denoted by the letter F . The graph's adjacency matrix, with each entry a_{ij} indicating the edge weight among i and j . The Adj_{Mat} is characterised as

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

If a graph is not directed, an adjacency matrix A is symmetric. A degree matrix is the diagonal matrix explained as $D = \{d_1, d_2, \dots, d_N\}$, here diagonal element d_i is the row total of an adjacency matrix such that

$$d_i = \sum_{j=1}^N a_{ij}. \quad (2)$$

Each node v_i has a real-valued feature vector $x_i \in R^{N \times F}$, and v_i owned one of the C class labels.

Graph neural networks (GNN)

Graph neural networks is a kind of neural network that is constructed for training by characteristic graphs. GNN techniques [11] attain modern efficiency in a node categorisation issue, which is the noteworthy enhancement of the recently utilised embedding algorithm [36]. The ability of GNNs to model both structural information and node attributes together sets them apart from previous versions. In theory, all GNN models have a message passing mechanism that spreads a node's feature information to its neighbours. To developing attributes for the various attribute space, most GNN architectures employ a learnable parameter matrix. In most cases, two or three of these layers are combined with a non-linear function.

GCN include the input layer, last perceptron layer with numerous hidden layers. Provided the adjacency matrix

Adj_{Mat} with the input feature matrix $X(0) = X$, GCN perform a subsequent layer-wise spread at hidden layers like the equation

$$X^{k+1} = \sigma \left(D^{-\frac{1}{2}} \text{Adj}_{\text{Mat}} D^{-\frac{1}{2}} X^k W^k \right). \quad (3)$$

Here $D = \text{diag}(d_1, d_2, \dots, d_n)$ is a diagonal matrix with $k = 0, 1, \dots, K-1$ and $d_i = \sum_{j=1}^n a_{ij}$, $W^k \in R^{d_k \times d_{k+1}}$, $d_0 = p$ is the layer particular weight matrix requiring to be learned.

$\sigma(\cdot)$ is denoted like the activation method, for example $\text{ReLU}(\cdot) = \max(0, \cdot)$ and $X^{k+1} \in R^{n_{k+1} \times d_{k+1}}$ Like a consequence of activations at a k th layer. The final layer of semi-supervised classification [37] is defined as,

$$Z = \text{softmax} \left(D^{-\frac{1}{2}} \text{Adj}_{\text{Mat}} D^{-\frac{1}{2}} X^{(k)} W^{(k)} \right), \quad (4)$$

where $W^{(k)} \in R^{d_k \times o}$ With o indicates a no. of classes. A result $Z \in R^{n \times o}$ Indicates the label forecast for the entire information X at that every row Z_i indicates a label forecast for an i th node.

Marginal Fisher's analysis (MFA)

Yan et al. [38] showed many dimension techniques could be merged into the graph embedding structure. By protecting a geometric graph form from the input space to the function space, the dimension reduction methods in this scheme produce low-dimensional features. The novel supervised technique known as MFA was presented in addition to this method. MFA (marginal fisher analysis) is a non-linear problem-solving manifold learning algorithm. The key concept behind MFA is to create two graphs based on sample neighborhood connections, then create a condition for intra-class sample cohesion and inter-class sample separation based on the two graphs.

The intra-class point adjacency association is represented by the intrinsic graph G_c , in which every instance is linked to its k -nearest class neighbours. Marginal point pairs of various classes are attached at a penalty graph G_p , which depicts the inter-class marginal point adjacency association. Let $N_{k_1}(x_i) = \{x_i^1, x_i^2, \dots, x_i^{k_1}\}$ is a collection of its k_1 nearby neighbours. A weight matrix is defined as,

$$W_{c,ij} = \begin{cases} 1 & \text{if } x_i \in N_{k_1}(x_j) \text{ or } x_j \in N_{k_1}(x_i) \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

After that, an intra-class density is explained like a total of distances among all nodes also its k_1 -nearby neighbours, which belong to a similar class:

$$\begin{aligned}
& \sum_{ij} y_i - y_j^2 W_{c,ij} \\
& = 2\text{Tr}\left(V^T X(D_c - W_c)X^T V\right) \\
& = 2\text{Tr}\left(V^T X L_c X^T V\right),
\end{aligned} \tag{6}$$

where D_c is the diagonal matrix by $D_{c,ii} = \sum_j W_{c,ij}$, $L_c = D_c - W_c$ is a Laplace matrix.

Regarding every points pair (x_i, x_j) from a various class, also attach the edge among x_i and x_j if x_j is one of the x_i 's k_2 -nearby neighbours whose class labels are dissimilar from a class label of x_i .

Let $N_{k_2}(x_i) = \{x_i^1, x_i^2, \dots, x_i^{k_2}\}$ is a collection of its k_2 nearby neighbours. A weight matrix is defined as,

$$W_{p,ij} = \begin{cases} 1 & \text{if } x_i \in N_{k_2}(x_j) \text{ or } x_j \in N_{k_2}(x_i) \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

The inter-class separability is illustrated through a penalty graph that has the word

$$\begin{aligned}
& \sum_{ij} y_i - y_j^2 W_{p,ij} \\
& = 2\text{Tr}\left(V^T X(D_p - W_p)X^T V\right) \\
& = 2\text{Tr}\left(V^T X L_p X^T V\right),
\end{aligned} \tag{8}$$

where D_p is a diagonal matrix with $D_{p,ii} = \sum_j W_{p,ij}$, $L_p = D_p - W_p$ is the Laplacian matrix.

The Criterion of Marginal Fisher is explained like this:

$$\arg \min_v \frac{V^T X L_c X^T V}{V^T X L_p X^T V}. \tag{9}$$

Following the effectiveness of implementing MFA in a variety of fields, some issues remain unsolved.

- In MFA, there is a unique problem that emerges from the fact that most training samples are much smaller than the dimension of each image, a flaw known as the singular or minimal sample size challenge.
- MFA is a supervised learning approach that requires labelled information to ensure successful generalisation on test samples. However, it is simple to obtain many face images for real-world face recognition, although only some of them are manually labelled. Since there isn't enough labelled data, strictly supervised MFA can't be well trained in this case.
- Since MFA is still a linear approach in nature, it is insufficient to explain the difficulty of real face images due to variations in lighting and pose. The most widely used kernels are data-independent kernels, which may not be

Table 1 Notations and symbols

Symbols	Description
G	Graph
n	Nodes count
m	Edges count
X	Feature matrix
f	Number of features per node
Y	Training labels
c	Number of classes
A	Adjacency matrix
A_{ij}	Edge among nodes i and j
I_n	Identity matrix
D	Diagonal matrix
W	Weight matrix

compatible with the intrinsic manifold structure uncovered by unlabelled data.

Our approach

This segment illustrates the recommended strategy in detail. First, describe the intended deep learning architecture. The following explains the two feature learning modules and finally represents a multiclass classifier. Table 1 gives symbols with notations utilised in this work.

Let graph G by n nodes with m edges has an attribute matrix $X \in \mathbb{R}^{n \times f}$ by f features per node, and training labels Y , annotating a partial set of nodes with the c possible classes. Consider A indicates an adjacency matrix of G , here a non-zero entry A_{ij} specifies the edge among nodes i with j . The weight matrix W is denoted as,

$$w_{ij} = \begin{cases} 1 \times f \text{dist}(v_i, v_j) & \text{if } (v_i, v_j) \in E \\ 0 & \text{Otherwise} \end{cases}. \tag{10}$$

Here $f \text{dist}(v_i, v_j)$ denote the feature distance between v_i, v_j . If the feature matrix is a binary vector, then the Hamming space is used; otherwise, use Euclidean distance.

$$f \text{dist}(v_i, v_j) = \begin{cases} v_i \oplus v_j, & \text{if } X \in \{0, 1\}^{n \times n} \\ \sqrt{\sum_{i=1}^n (q_i, p_i)^2}, & \text{otherwise} \end{cases}. \tag{11}$$

Deep architecture model for node classification

This section discusses the proposed work's overall deep architecture. It could view as the vast paradigm for constructing information representations. Furthermore, this framework uses function training elements by several result dimension for the train a data demonstrations at all layer. Mapping

Fig. 1 Proposed architecture

methods among successive layers are attained through optimising feature learning models layer by layer. The proposed deep architectures are depicted in Fig. 1 as a high-level outline. In this diagram, three hidden layers (H_1 , H_2 and H_3) are attached to the input with output layers. A first layer is built using a random matrix W_{rm} .

The novel demonstration of the input A could write like this

$$hv_1 = \rho(A \cdot W_{rm}^T \cdot X), \quad (12)$$

where $\rho(\cdot)$ is the non-linear activation method, subsequently, characteristic training patterns are utilised to boot the next layer. Results of the following hidden layers are

$$hv_t = \rho(A \cdot W_{F_{t-1}} \cdot hv_{t-1}). \quad (13)$$

The load patterns of the second and third layers in Fig. 1 acquired from function training models are W_{MMFA} and W_{KPCA} . Finally, a softmax multiclass classifier is used as a final layer for labelling works. At an initial esoteric layer, high dimensional input information descriptions could acquire. Lower-dimensional embeddings can then be progressively known using two function learning models. Feature learning modules initialise a weight matrice of hidden layers, except for the matrix at an initial hidden layer, resulting in better results than different DL methods that use arbitrary matrix.

Modified MFA

This section describes how to build deep learning models using modified marginal Fisher analysis (MMFA). There are several advantages to using MMFA. MMFA is much more

specific for feature selection than many conventional characteristic learning models, such as LDA, since it makes no assumptions regarding the knowledge dissemination of individual group. Furthermore, the margins between classes will accurately describe the classes' separability.

To boost MFA, this paper considers using the best discriminate attributes from together unlabelled with labelled instances, as well as treating unlabelled with labelled information separately while constructing a fitness method to achieving maximum inter-class margin whilst decreasing intra-class difference.

The between-class also within-class of MFA is constructed like this:

Between-class: $W'_{ij} = W'_{ji} = 1$ if (x_i, x_j) is between the k_2 shortest pairs between the set $\{(x_i, x_j) | x_i \in X_c, x_j \notin X_c\}$.

Within-class: $W_{ij} = W_{ji} = 1$ if x_j is between the k_1 nearby neighbor of x_i at a similar class.

Where W with W' are similarity matrices that indicate within-class correlation and between-class dissimilarity, correspondingly. MFA reduces resemblance while concurrently maximising separability at low dimension space.

The following section explains the proposed modified marginal fisher analysis.

Initially, create the nearest neighbour graph. Find the k closest neighbours of data point x_i and construct the edge among x_i also its neighbours. The set of its k most immediate neighbours is denoted by $(x_i) = \{x_i^1, x_i^2, \dots, x_i^k\}$. The adjacency matrix can be written as follows:

$$A_{ij} = \begin{cases} \exp\left(-\frac{x_i - x_j^2}{\sigma^2}\right) & \text{if } x_i \in N(x_j) \text{ or } x_j \in N(x_i) \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

The weight matrix within-class and between-class is defined by,

$$W_{ij} = \begin{cases} \alpha A_{ij}, & \text{if } c_i = c_j \\ 0, & \text{Otherwise} \end{cases}, \quad (15)$$

$$W'_{ij} = \begin{cases} \beta A_{ij}, & \text{if } c_i \neq c_j \\ 0, & \text{otherwise} \end{cases}. \quad (16)$$

Here, α and β are adjustable parameter and $\alpha + \beta = 1$.

The diagonal matrix for within-class and between-class is defined by,

$$D_{ii} = \sum_j W_{ij}, \quad (17)$$

$$D'_{ii} = \sum_j W'_{ij}. \quad (18)$$

The between-class separability also within-class similarity is defined as follows:

$$S = 2\text{Tr}(A^T X(D - W)X^T A), \quad (19)$$

$$S' = 2\text{Tr}(A^T X(D' - W')X^T A). \quad (20)$$

The objective functions of MMFA are defined like,

$$W_{\text{MMFA}} = \arg \max_A \frac{\text{Tr}(A^T X(D' - W')X^T A)}{\text{Tr}(A^T X(D - W)X^T A)}. \quad (21)$$

Kernel PCA

Given the data $Y = [y_1, y_2, \dots, y_n]$, PCA get the linear subspace of dimensional d such that the entire information lies on or near to it., at a Euclidean distance. PCA solves

$$\min_{U_d, \{x_i\}} \sum_{i=1}^N y_i - U_d x_{i2}^2, \quad (22)$$

$$U_d^T U_d = I, \quad (23)$$

where U_d is the matrix of orthonormal. The best result is $x_i = U_d^T y_i$.

The optimal solution for kernel PCA is

$$\min_X K_y - X^T X_F^2, \quad (24)$$

$$X X^T = D, \quad (25)$$

where $K_y = Y^T Y$ is kernel matrix, D indicates diagonal matrix containing eigenvalues of K_y .

In several application settings, structural information implying or being implied by dependencies can benefit the dimensionality reduction task. This knowledge could encode at a graph also embodied at X by graph regularisation. Especially, assume there has the graph G on that information is

Table 2 Example of kernel types

Type	Function	Parameters
p-step random walk	$r(\lambda) = (a - \lambda)^{-p}$	$a \geq 2, p$
Diffusion	$r(\lambda) = \exp\left\{\frac{\sigma^2 \lambda}{2}\right\}$	$\sigma^2 \geq 0$
Regularized Laplacian	$r(\lambda) = 1 + \sigma^2 \lambda$	$\sigma^2 \geq 0$
Bandlimited	$r(\lambda_n) = \begin{cases} \frac{1}{\beta} n \leq B \\ \beta \text{ otherwise} \end{cases}$	$\beta, B > 0$

soft. Specifically, vectors $\{x_i\}$ that correspond to connected nodes of G are near to all others at Euclidean distance. By A indicating an adjacency matrix of G , $a_{ij} = 1$ if node i is linked by node j . A Laplacian of G is $L_G = D - A$, here D is the matrix of diagonal by entries $d_{ii} = \sum_j a_{ij}$.

The kernel PCA [17] can be rewritten as,

$$\min_X -\text{tr}(X K_y X^T) + \gamma \text{tr}(X L_G X^T), \quad (26)$$

$$X X^T = I. \quad (27)$$

The following steps are used to find the low dimensional representations.

1. Compute $r(L_G)$.
2. Compute the largest eigenvalues and corresponding eigenvectors of $K_y - \gamma r(L_G)$.
3. Collect V_d .

Here $r(\cdot)$ is the non-decreasing scalar method of eigenvalues of L_G . Table 2 shows the graph of Laplacian kernels

Implementation and data design

This segment exhibits the implementation and trial picture of the recommended work. Three graph-related data were used with the Java framework for validation and analysis of the proposed model.

Citation graphs such as CORA, PubMed, and CiteSeer [39] are widely utilised. All of that undirected graph is prepared up of nodes (documents) with edges (citations). When one text references another, an edge connects them. The characteristics of the node refer to bag-of-words features of a text material of documents.

Each node in the citation datasets corresponds to a journal article published in that journal. A citation from one article to another is symbolised through the edge among two nodes, and a label represents the subject of the article. Each node in every dataset has a binary bag-of-words (BoW) feature vector. A BoW is based on a summary of an article. As a result, provided a BoW of an article's citations and abstract

Table 3 Graph-based data set

Dataset	Nodes (N)	Edges (M)	Features (F)	Classes (C)
Cora	2708	5429	1433	7
Citeseer	3312	4732	3703	6
Pubmed	19,717	44,338	500	3

to further (probably labelled) article, a job is for forecast a topic of those articles. A graph dataset is shown in Table 3.

CiteSeer

The CiteSeer dataset is mostly made up of computer-related research publications, such as those on machine language, information processing, databases, and artificial intelligence, among other topics. The CiteSeer collection contains 3312 scientific papers that are divided into six categories (Agents, AI, DB, IR, ML, HCI). There are 4732 linkages in the citation network. A 0/1-valued word vector describes each publication in the dataset, indicating the existence or absence of the associated word from the dictionary. There are 3703 distinct terms in the dictionary.

Cora

A Cora data set is mostly made up of articles on ML, for example, reinforcement learning, probabilistic approaches, genetic algorithms, and neural networks, among other topics. The Cora dataset contains 2708 scientific publications that are divided into seven categories (neural networks—818, probabilistic methods—426, genetic algorithms—418, theory—351, case based—298, reinforcement learning—217, rule learning—180). There are 5429 linkages in the citation network. A 0/1-valued word vector describes each publication in the dataset, indicating the existence or absence of the associated word from the dictionary. There are 1433 distinct terms in the dictionary.

Table 4 Baseline methods

Method	Author	Year	Description
SemiEmb [41]	Weston et al.	2008	The non-linear embedding algorithm to the deep structure
DeepWalk [40]	Perozzi et al.	2014	Unsupervised random walks based Node embeddings with SVM classifier
Planetoid [36]	Yang et al.	2016	Graph embedding based on a semi-supervised learning structure
ChebyShev [42]	Defferrard et al.	2016	A convolutional network that extensive, conventional CNN to non-Euclidean space
GCN [37]	Kipf et al.	2017	Node embedding vector through shortening a Chebyshev polynomial for an initial order neighbourhood
NAGCN [13]	Gong et al.	2019	Semi-supervised node categorisation works using the neighbourhood adaptive graph convolutional network
D-SEGCN [15]	Jia et al.	2020	An attention method based on the deep hierarchical network for obtaining attribute dimensions
HCNP [20]	Liu et al.	2021	Semi-supervised node training works utilising a novel high-order GCN

Table 5 Result comparison of various methods

Method	Citation dataset		
	Cora	Citeseer	Pubmed
SemiEmb [41]	59	59.6	71.1
DeepWalk [40]	67.2	43.2	65.3
Planetoid [36]	75.7	64.7	77.2
ChebyShev [42]	81.2	69.8	74.4
GCN [37]	81.5	70.3	79
NAGCN[13]	82.8	72.0	79.7
D-SEGCN [15]	84.8	74	81.2
HCNP [20]	82.5	71.7	79.3
DLM-SSC	85.3	76.5	78.7

The results of the proposed system are highlighted in bold

PubMed

A PubMed dataset is mostly made up of biomedical research articles. Nodes denote texts; edges indicate citation relations. The Pubmed Diabetes dataset contains 19717 scientific publications about diabetes from the PubMed database, which are divided into three categories (diabetes mellitus—experimental, type 1, type 2). There are 44,338 linkages in the citation network. A TF/IDF weighted word vector from a vocabulary of 500 unique terms is used to describe each publication in the dataset.

All publication is represented through the 0 or 1 valued word vector representing a presence or absence of an equivalent term from a dictionary in each of the three citation datasets.

Evaluation results

This section evaluates the efficiency of this proposed technique in opposition to numerous further associated practices on three traditional citation standard data sets. To verify the

Fig. 2 Accuracy comparison **Table 6** Classification result for different learning models

Model	Citation dataset		
	Cora	Citeseer	Pubmed
DLM + MMFA	80.2	74.1	75.9
DLM + KPCA	78.5	73.8	78.4
DLM + MMFA + KPCA	85.3	76.5	78.7

The results of the proposed system are highlighted in bold

Fig. 3 Classification result for learning models

performance of the proposed DLM-SSC, various baselines, including graph embedding approaches, graph convolution approach, and high-order graph convolution, comes. Table 4 shows the baseline methods.

The performance of a proposed technique is analysed by classification accuracy. Table 5 demonstrates the accuracy comparison presented by several baseline techniques.

Figure 2 shows the accuracy comparison of various baseline methods for three citation data sets. The proposed method gives higher classification accuracy 85.3 and 76.5

Table 7 Classification result for different no of hidden layers

	No of hidden layers	Features	Dataset		
			Cora	Citeseer	Pubmed
	3	1433-700-700	85.3	76.5	78.7
	5	3703-1800-1800-900-900	80.0	73.1	76.4
	7	500-250-250-125-125-60-60	77.5	70.9	75.1

for Cora and citeseer dataset. The D-SEGCN [15] have high accuracy for PubMed data.

Table 6 and Fig. 3 shows the classification result for different learning models. From those results, the feature learning model MMFA has higher accuracy compared to the KPCA feature learning model. When combining both MMFA and KPCA feature learning models with DLM, the model increase the classification accuracy.

Table 7 shows the results for a dissimilar number of hidden layers for the three citation dataset. When increasing the hidden layers, the outcome will be reduced.

Conclusion

A novel neural network deep architecture for semi-supervised node classification is presented in this article. A supervised pretraining approach may be used to initialise the architecture. This approach employs two distinct learning methods. As a hidden layer, each function taught by MMFA and KPCA is used. These two learning algorithms efficiently learn lower dimensional of data. For experiments three publicly available citation dataset (Cora, Citeseer and Pubmed) are used. The results shows that the proposed work achieves 85.3% accuracy for Cora, 76.5% for Citeseer and 78.7% for Pubmed dataset. Extensive tests show that this proposed work executes superior to further similar approaches over small datasets. This principle would be extended to graph classification in the future, and deep active learning approaches for graph-related learning methods can be used. The baseline strategies are seen in Table 4.

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

- Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: a systematic review. *IEEE Access* 7:19143–19165
- Jelodar H, Wang Y, Orji R, Huang S (2020) Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. *IEEE J Biomed Health Inform* 24(10):2733–2742
- Sun K, Xiao B, Liu DS, Wang JD (2019) Deep, high-resolution representation learning for human pose estimation. *Proc CVPR* 2019:5693–5703
- Wang M, Du X, Shu X, Wang X, Tang J (2020) Deep supervised feature selection for social relationship recognition". *Pattern Recogn Lett* 138:410–416
- Gao W, Pooja Mahajan S, Sulam J, Gray JJ (2020) Deep learning in protein structural modeling and design. *Patterns* 1(9):100142
- Abrishami A, Aliakbary S (2019) Predicting citation counts based on deep neural network learning techniques. *J Informetr* 13(2):485–499
- Guo K et al (2021) Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Trans Intell Transp Syst* 22(2):1138–1149
- Tavassoli Kejani M, Dornaika F, Talebi H (2020) Graph convolution networks with manifold regularisation for semi-supervised learning. *Neural Netw* 127:160–167
- Kajla NI, Missen MMS, Luqman MM, Coustaty M, Mehmood A, Choi GS (2020) Additive angular margin loss in deep graph neural network classifier for learning graph edit distance. *IEEE Access* 8:201752–201761
- Gu W, Gao F, Li R, Zhang J (2021) Learning universal network representation via link prediction by graph convolutional neural network. *J Soc Comput* 2(1):43–51
- Li Q, Han Z, Wu X-M (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: *Proc. AAAI*, pp 1–8
- Atwood J, Towsley D (2016) Diffusion-convolutional neural networks [online]. <http://arxiv.org/abs/1511.02136v2>
- Gong P, Ai L (2019) Neighborhood adaptive graph convolutional network for node classification. *IEEE Access* 7:170578–170588
- Zhang M, Cui Z, Neumann M, Chen Y (2018) An end-to-end deep learning architecture for graph classification. In: *Proc. AAAI*, pp. 1–8
- Jia N, Tian X, Zhang Y, Wang F (2020) Semi-supervised node classification with discriminable squeeze excitation graph convolutional networks. *IEEE Access* 8:148226–148236
- Zhong G, Chherawala Y, Cheriet M (2013) An empirical evaluation of supervised dimensionality reduction for recognition. In: *ICDAR*, pp 1315–1319
- Shen Y, Traganitis PA, Giannakis GB (2018) Graph-adaptive non-linear dimensionality reduction. *Unknown Journal*
- Abu-El-Haija S, Kapoor A, Perozzi B, Lee L (2019) N-GCN: multi-scale graph convolution for semi-supervised node classification. In: *Proc. UAI*, pp 1–9

19. Wu J, He J, Xu J (2019) Demo-Net: degree-specific graph neural networks for node and graph classification. <http://arxiv.org/abs/1906.02319>
20. Liu X, Xia G, Lei F, Zhang Y, Chang S (2021) Higher-order graph convolutional networks with multi-scale neighborhood pooling for semi-supervised node classification. *IEEE Access* 9:31268–31275
21. Li B, Pi D (2019) Learning deep neural networks for node classification. *Expert Syst Appl* 137:324–334
22. Molokwu BC, Shuvo SB, Kar NC, Kobti Z (2020) Node classification in complex social graphs via knowledge-graph embeddings and convolutional neural network. In: *International conference on computational science—ICCS 2020*, vol 12142. Springer, pp. 183–198
23. Madhawa K, Murata T (2020) Active learning for node classification: an evaluation. *Entropy* 22(10):1164
24. Yuan Y, Mou L, Lu X (2015) Scene recognition by manifold regularized deep learning architecture. *IEEE Trans Neural Netw Learn Syst* 26(10):2222–2233
25. Kejani MT, Dornaika F, Talebi H (2020) Graph convolution networks with manifold regularisation for semi-supervised learning. *Neural Netw* 127:160–167
26. Zhu R, Dornaika F, Ruichek Y (2020) Semi-supervised elastic manifold embedding with deep learning architecture. *Pattern Recogn* 107:107425
27. George T, Konstantinos B, Stefanos Z, Schuller BW (2014) A deep semi-NMF model for learning hidden representations. In: *ICML*, pp 1692–1700
28. Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY (2011) Multi-modal deep learning. In: *ICML*, pp 689–696
29. Liu Z, Lai Z, Ou W, Zhang K, Zheng R (2020) Structured optimal graph-based sparse feature extraction for semi-supervised learning. *Signal Process* 170:107456
30. Shen X-J, Liu S-X, Bao B-K, Pan C-H, Zha Z-J, Fan J (2020) A generalised least-squares approach regularised with graph embedding for dimensionality reduction. *Pattern Recogn* 98:107023
31. Gou J, Yang Y, Yi Z, Lv J, Mao Q, Zhan Y (2020) Discriminative globality and locality preserving graph embedding for dimensionality reduction. *Expert Syst Appl* 144:113079
32. Rajabzadeh H, Jahromi MZ, Ghodsi A (2021) Supervised discriminative dimensionality reduction by learning multiple transformation operators. *Expert Syst Appl* 164:113958
33. Yu W, Qin Z (2020) Graph convolutional network for recommendation with low-pass collaborative filters. In: *Proc. ICML*, pp 1–13
34. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. In: *International conference on machine learning*, PMLR, pp. 1263–1272
35. Molokwu BC, Shuvo SB, Kar NC, Kobti Z (2020) Node classification and link prediction in social graphs using RLVEC. In: *32nd international conference on scientific and statistical database management (SSDBM 2020)*, Article 12, pp 1–10
36. Yang Z, Cohen W, Salakhudinov R (2016) Revisiting semi-supervised learning with graph embeddings. In: *Proceedings of the international conference on machine learning*, pp 40–48
37. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: *Proc. int. conf. learn. represent.*, pp 1–13
38. Yan S, Xu D, Zhang B, Zhang H-J, Yang Q, Lin S (2007) Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell* 29(1):40–51
39. Sen P, Namata G, Bilgic M, Getoor L, Galligher B, Eliassi-Rad T (2008) Collective classification in network data". *AI Mag* 29:93–93
40. Perozzi B, Al-Rfou R, and S. Skiena, DeepWalk: Online learning of social representations, in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, pp. 701–710, 2014
41. J. Weston, F. Ratle, and R. Collobert, Deep learning via semi-supervised embedding, in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, pp. 1168–1175, 2008
42. M. Defferrard, X. Bresson, and P. Vandergheynst, Convolutional neural networks on graphs with fast localised spectral filtering, in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 3844–3852, 2016

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