Neural Networks 127 (2020) 160–167

Contents lists available at ScienceDirect

Neural Networks

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

Graph Convolution Networks with manifold regularization for semi-supervised learning

M. Tavassoli Kejani [a], F. Dornaika [b,c,*], H. Talebi [d]

[a] University of Isfahan, Isfahan, Iran
[b] University of the Basque Country UPV/EHU, San Sebastian, Spain
[c] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[d] Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 24 November 2019
Received in revised form 4 April 2020
Accepted 16 April 2020
Available online 23 April 2020

Keywords:
Graph-based semisupervised learning
Graph Convolution Networks (GCN)
Label prediction
Manifold regularization
Semisupervised image classification

ABSTRACT

In recent times, Graph Convolution Networks (GCN) have been proposed as a powerful tool for graph-based semi-supervised learning. In this paper, we introduce a model that enhances label propagation of Graph Convolution Networks (GCN). More precisely, we propose GCNs with Manifold Regularization (GCNMR). The objective function of the proposed GCNMR is composed by a supervised term and an unsupervised term. The supervised term enforces the fitting term between the predicted labels and the known labels. The unsupervised term imposes the smoothness of the predicted labels of the whole data samples. By learning a Graph Convolution Network with the proposed objective function, we are able to derive a more powerful semi-supervised learning. The proposed model retains the advantages of the classic GCN, yet it can improve it with no increase in time complexity. Experiments on three public image datasets show that the proposed model is superior to the GCN and several competing existing graph-based semi-supervised learning methods.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

In many real-world problems, obtaining a large number of labeled samples can be very challenging if not impossible. The reason is twofold. First, these labeled samples can be very scarce. Second, for some typical applications (biometrics, biology) it is very expensive to obtain labels in a manual way. Thus, getting labels can be very expensive in terms of time, money, and human labor (Dai & Van Gool, 2016; El Traboulsi, Dornaika, & Assoum, 2015; Gong, et al., 2016). On the other hand, unlabeled samples are easily accessible. The use of unlabeled data is an excellent way which can provide a significant influence and a considerable extension of application for learning methods. Semi-supervised learning (SSL) aims at building models by exploiting labeled and unlabeled data. Such methods benefit from labeled samples that explicitly contribute to the discrimination capacity of the final model, and from unlabeled ones to maintain the intrinsic structure of data.

Nowadays, SSL-based methods become a hot topic in machine learning (Guo, Zou, & Tan, 2020; Liu, Zhao, & Kong, 2019; Traboulsi, Dornaika, & Ruichek, 2018). Thanks to semi-supervised learning (SSL), image classification has seen remarkable progress (Ashfaq, Wang, Huang, Abbas, & He, 2017; Dornaika & El Traboulsi, 2016, 2017; Liu et al., 2019; Zhang, Zhang, Li, Jing, & Lv, 2019).

Graph-based learning methods can be used by many real-world applications (Nie, Wang, Wang, & Li, 2020). Graph-based methods exploits data structure in discovering the sought models. Besides, these approaches can encode the relationship between the labeled or unlabeled samples data. Many data dimensionality reduction methods rely on graphs. In graph-based learning field, several effective algorithms were developed in the past: Locally Linear Embedding (LLE) (Roweis & Saul, 2000), Laplacian Eigenmap (LE) (Belkin & Niyogi, 2002), ISOMAP (Tenenbaum, De Silva, & Langford, 2000), and Flexible Semi-Supervised Embedding (Dornaika & El Traboulsi, 2016) are graph-based non-linear embedding algorithms. Manifold Regularized Deep Learning algorithm (MRDL) (Yuan, Mou, & Lu, 2015) proposed a deep architecture to learn the high-level features for scene recognition in an unsupervised fashion. In Hou, Nie, Li, Yi, and Wu (2014), the authors proposed a Joint Embedding Learning and Sparse Regression (JELSR) for unsupervised feature selection. Semi-supervised Discriminant Embedding (SDE) (Yu, Zhang, Domeniconi, Yu, & You, 2012) is the semi-supervised extension of Local Discriminant Embedding (LDE) (Chen, Chang, & Liu, 2005). It is a linear projection method that is based on manifold smoothness and a regularizer that controls the complexity of learning. Semi-Supervised Discriminant Analysis (SDA) (Cai, He, & Han, 2007)

* Corresponding author at: University of the Basque Country UPV/EHU, San Sebastian, Spain.
E-mail address: fadi.dornaika@ehu.eus (F. Dornaika).

https://doi.org/10.1016/j.neunet.2020.04.016
0893-6080/© 2020 Elsevier Ltd. All rights reserved.

extends the classic Linear Discriminant Analysis (LDA) (Martínez & Kak, 2001) by adding a geometrically based regularization term in the objective function of LDA.

Semi-supervised methods benefit from labeled samples and unlabeled samples. The labeled samples explicitly contribute to the discrimination ability of the final model, and the unlabeled ones maintain the geometrical structure of data.

In recent times, Graph Convolutional Neural Networks (GCNNs) achieved remarkable success and were widely applied in graph analysis (Defferrard, Bresson, & Vandergheynst, 2016; Hamilton, Ying, & Leskovec, 2017; Kipf & Welling, 2016; Kipf & Welling, 2017; Klicpera & Gunnemann, 2019; Velickovic, et al., 2017; Velickovic, Fedus, Lió, & Bengio, 2019; Xu, Hu, Leskovec, & Jegelka, 2019). As for graph-based semi-supervised learning, Graph Convolutional Network (GCN) deploys convolution operation defined on graphs to extract features (Kipf & Welling, 2017; Yu, Yang, & Zhang, 2019). By doing so the GCN model overcomes the shortcomings of the conventional methods by propagating features based on the graph structure through multiple layers. The semi-supervised learning algorithm adopted by the GCN algorithm takes as input a data set together with its associated graph, and outputs the label matrix for the whole samples. The GCN model can be thought like a special deep neural net in which the data matrix is the input. The loss function of this net is given by the cross-entropy loss between the ground-truth labels and the predicted ones. On a number of benchmarks of graph-based semi-supervised classification, GCN outperforms state-of-the-art methods both in accuracy and efficiency. However, the GCN model only focuses on the fitness between the ground-truth labels and the predicted ones. Indeed, it ignores the manifold structure that is implicitly encoded by the graph, which is an important cue in semi-supervised learning field.

In this paper, we propose a Graph Convolutional Network with Manifold Regularization (GCNMR). Our proposed model exploits data-driven graphs in two ways. First, it integrates feature propagation over graphs. Second, it ensures that estimated labels satisfy the manifold regularization. Hence, our proposed GCNMR exploits another source of information, and has the opportunity to output more accurate labels.

The remainder of the paper is structured as follows. Section 2 introduces the notations and definitions used in this paper, and briefly reviews some related works. Section 3 presents the proposed GCNMR model. Section 4 reports and analyzes some experimental results obtained with three public image datasets. Finally, Section 5 concludes the paper. In this paper, capital bold letters denote matrices and small bold letters denote vectors.

2. Notation and preliminaries

In this section, we introduce the notations adopted in the paper. We define the data matrix by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}] \in \mathbb{R}^{d\times(l+u)}$, where $\mathbf{x}_i\,|_{i=1}^{l}$ and $\mathbf{x}_i\,|_{i=l+1}^{l+u}$ are the labeled training samples and unlabeled test samples, respectively, l and u being the total numbers of labeled train samples and unlabeled test samples data, respectively, and d being the sample dimension. Let $N = l + u$ be the total number of samples data and n_c be the total number of labeled samples in the cth class. The labeled train samples are denoted by the matrix $\mathbf{X}_l = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l] \in \mathbb{R}^{d\times l}$. The label of each sample $\mathbf{x}_i$ is denoted by $y_i \in 1, 2, \ldots C$, where C is the total number of classes. The unlabeled (test) samples are denoted by the matrix $\mathbf{X}_u = [\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_{l+u}] \in \mathbb{R}^{d\times u}$. Let $\mathbf{S} \in \mathbb{R}^{(l+u)\times(l+u)}$ be the graph similarity matrix associated with the data matrix $\mathbf{X}$ where $S(i, j)$ represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., $S(i, j) = sim(\mathbf{x}_i, \mathbf{x}_j)$. The function $sim(., .)$ can be any symmetric function that measures the similarity between two samples. This can be given by the Cosine or the Gaussian Kernel.

Table 1
Main notations used in the paper.

Notation	Description
d	Dimensionality of original data
N	Number of data samples
l	Number of labeled samples
u	Number of unlabeled samples
C	Number of classes
n_c	Number of labeled cth class samples
$\mathbf{x}_i \in \mathbb{R}^d$	The i-th original data sample
$y_i \in \{1, 2, \ldots, C\}$	The label of $\mathbf{x}_i$
$\mathbf{X}_l = [\mathbf{x}_1, \ldots, \mathbf{x}_l] \in \mathbb{R}^{d\times l}$	Labeled train samples matrix
$\mathbf{X}_u = [\mathbf{x}_{1+1}, \ldots, \mathbf{x}_{l+u}] \in \mathbb{R}^{d\times u}$	Unlabeled test samples matrix
$\mathbf{X} = [\mathbf{X}_l, \mathbf{X}_u] \in \mathbb{R}^{d\times(l+u)}$	Original data matrix
$\mathbf{S}$	Graph similarity matrix
$\mathbf{F} \in \mathbb{R}^{N\times C}$	Label matrix
$\mathbf{D}$	Diagonal matrix
$\mathbf{I}$	Identity matrix
$\mathbf{L}$	Laplacian matrix

The Laplacian of the similarity matrix $\mathbf{S}$ is denoted by $\mathbf{L}$ and is given by $\mathbf{L} = \mathbf{D} - \mathbf{S}$ where $\mathbf{D}$ is a diagonal matrix whose elements are the row or column(since the matrix is symmetric) sums of $\mathbf{S}$ matrix. The normalized Laplacian matrix $\mathbf{L}$ is defined by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{1/2}$ where $\mathbf{I}$ denotes the identity matrix. Table 1 summarizes the notations used in the paper.

3. Related work: Graph-based semi-supervised learning

Semi-supervised learning aims to use a small amount of labeled data and a large amount of unlabeled data to learn a classifier. A common assumption in semi-supervised learning is the manifold assumption that data distributes in a manifold structure and nearby samples have similar output values.

3.1. Graph-based label propagation methods

The semi-supervised learning methods using graph-based label propagation received much attention in the last decade. All of them impose that samples with high similarity should share similar labels. They differ by the regularization term as well as by the loss function used for fitting label information associated with the labeled samples. All of these methods use the graph similarity matrix and the initial labels of some samples. Some recent label propagation algorithms (they can also be called classifiers Sousa, Rezende, & Batista, 2013) are: Gaussian Fields and Harmonic Functions (GFHF) (Zhu, Ghahramani, Lafferty, et al., 2003), Local and Global Consistency (LGC) (Zhou, Bousquet, Lal, Weston, & Schölkopf, 2004), Laplacian Regularized Least Square (LapRLS) (Belkin, Niyogi, & Sindhwani, 2006), Robust Multi-class Graph Transduction (RMGT) (Liu & Chang, 2009), Flexible Manifold Embedding (FME) (Nie, Xu, Tsang, & Zhang, 2010), and Kernel Flexible Manifold Embedding (KFME) (El Traboulsi et al., 2015). These techniques can be either transductive (defined for training samples only) or inductive (defined for both training and unseen samples).

The Multi-view Learning with Adaptive Neighbors (MLAN) method (Nie, Cai, Li, & Li, 2018) learns similarity graph with adaptive neighbors. It learns weight of views. In these methods, the graph and label propagation are jointly performed. The method proposed The work presented in Nie, Tian, Wang, and Li (2019) learns a unified graph and weights from a priori individual graphs. The proposed method allows the learned unified graph to be away from the convex hull of the individual graphs. In an iterative process, the method estimates the unknown labels, the unified graph and the individual weights.

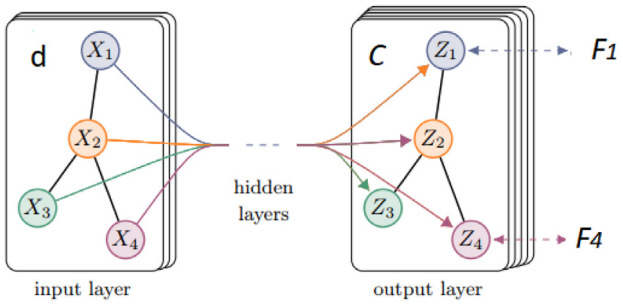


Fig. 1. Schematic illustration of multi-layer Graph Convolutional Network (GCN) for semi-supervised learning with four samples (Kipf & Welling, 2016). The number of input channels (feature dimensionality) is d. The output layer is the matrix of labels. It has C channels where C is the number of classes. The graph structure (edges shown as black lines) is shared over layers, the label distribution of the ith sample is denoted by the row vector $\mathbf{F}_i \in \mathbb{R}^C$.

3.2. Graph-based embedding methods

In this category of methods, the labeled and unlabeled samples are jointly used in order to estimate a discriminant projection that induce a distance metric. Once the data are represented in the new space, a simple classifier is invoked in order to predict the label of the testing samples. In practice, very often the used classifier is given by the nearest neighbor classifier. Some typical methods are given below:

Semi-supervised Discriminant Analysis (SDA) By adding a geometrically based regularizer, Cai et al. extend LDA into its semi-supervised version: SDA (Cai, He, & Han, 2007). This method seeks a projection matrix that projects samples to a reduced subspace wherein samples having the same label are as close as possible, keeping at the same time the intrinsic geometric structure of the data.

Semi-supervised Discriminant Embedding (SDE) SDE (Huang, Liu, & Pan, 2012) aims to estimate a projection matrix that can transform data samples from their high dimensional space to a reduced subspace and contributing, at the same time, to the discrimination of the different classes.

Sparsity Preserving Discriminant Analysis (SPDA) Like SDA, SPDA (Qiao, Chen, & Tan, 2010) can be considered as an extension of LDA. Instead of using Laplacian smoothness criterion related to the whole labeled and unlabeled data, SPDA proposes to use the Sparsity Preserving property.

Exponential Semi-supervised Discriminant Embedding ESDE (Dornaika & Traboulsi, 2017) was proposed in order to overcome the Small Sample Size problem as well as to emphasize the discrimination property by enlarging distances between samples that belong to different classes.

Semi-Supervised Online Kernel Matrix Factorization (SSOKMF) (Vanegas, Beltran, Escalante, & Gonzalez, 2019) performs a semantic embedding by finding a non-linear mapping to a low-dimensional semantic space. The non-linear modeling is learned with kernel methods with a learning-in-a-budget strategy.

4. Review of Graph Convolutional Networks (GCN)

The Graph Convolutional Networks algorithm (Kipf & Welling, 2017) presents an approach for semi-supervised learning on graph-structured data. it can estimate the labels of unlabeled samples in a deep fashion. In order to have a self-contained description, this section will briefly present GCN. More details about GCNs can be found in Kipf and Welling (2017) and Yu et al. (2019).

The basic idea of GCN is to use deep neural networks that perform the linear and nonlinear mapping of data representation in order to provide the labels. In each layer, the data features are propagated using the graph edges. This operation is equivalent to replacing each sample by the linear combination of its neighboring nodes. It motivates the choice of convolutional architecture via a localized first-order approximation of spectral graph convolutions. The GCN is used for semi-supervised classification on graph-structured data, such as citation networks or on a knowledge graph dataset.

For semi-supervised label propagation, a two-layer GCN model has the following form:

$$\mathbf{F} = f(\mathbf{X}, \mathbf{S}) = \text{softmax}\left(\hat{\mathbf{S}}\sigma(\hat{\mathbf{S}}\mathbf{X}^{\mathbf{T}}\mathbf{W}^{(0)})\mathbf{W}^{(1)}\right) \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{d\times N}$ denotes the input data with N samples and d dimensions, $\mathbf{S} \in \mathbb{R}^{N\times N}$ denotes the graph matrix associated with the data, and $\hat{\mathbf{S}}$ is the renormalized graph matrix. It is given by $\hat{\mathbf{S}} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{S}+\mathbf{I})\hat{\mathbf{D}}^{-\frac{1}{2}}$ and $\tilde{D}_{ii} = \sum_j (\mathbf{S}+\mathbf{I})_{ij}$. $\sigma(.)$ denotes an element-wise activation function, such as $ReLU(.) = max(0, .)$. It is given by the rectified linear activation function ReLU. The softmax operator applied on a $N\times C$ matrix $\mathbf{A}$ in Eq. (1) transforms each row in $\mathbf{A}$ to a probability distribution. $\mathbf{W}^{(0)} \in \mathbb{R}^{d\times H}$ is an input-to-hidden weight matrix for a hidden layer with H features, and $\mathbf{W}^{(1)} \in \mathbb{R}^{H\times C}$ is a hidden-to-output weight matrix with C features. C denotes the number of classes. $\mathbf{F} \in \mathbb{R}^{N\times C}$ is the matrix of labels, and represents the unknown output of the GCN.

The deep neural network is learned by imposing that the estimated labels of the labeled samples are as close as possible to their desired value. This is achieved by minimizing a cross-entropy loss function that is usually used in classification problems. This loss is given by:

$$\mathcal{L}\left(\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(L)}\right) = -\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\, log(F_{ik}) \tag{2}$$

where F_{ik} is the estimated probability of the ith sample to be in class k, and y_{ik} is the ground-truth probability (either 0 or 1). l denotes the total number of labels samples. Since y_{ik} is equal to one for only one value of k. The above loss can be simplified to:

$$\mathcal{L}\left(\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(L)}\right) = -\sum_{i=1}^{l} log(F_{ik})$$

According to the above loss function, whenever the network outputs a value for F_{ik} that is far from the desired value (here it is one) then the ith sample imposes a positive penalty in the total loss.

The neural network weights $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are trained using gradient descent. In Kipf and Welling (2017), the authors perform batch gradient descent using the full dataset for every training iteration.

Fig. 1 illustrates a GCN with an arbitrary number of layers. The dataset consists of four samples.

5. Proposed model: Graph Convolution Networks with manifold regularization

The GCN which is presented in the previous section only exploits label fitting and discards the fact that label and unlabeled data are on a hidden manifold that can be captured by the data graph. This motivates us to jointly use the cross entropy loss (a supervised term) and the manifold regularization loss (an unsupervised term) in order to learn an enhanced GCN model. In our proposal, the role of the graph is twofold. First, it performs feature propagation at each layer. Second, it ensures that the estimated labels (the final output of the deep net) satisfy manifold regularization.

In our proposed model, the deep neural network is trained by minimizing the following global loss function:

$$\mathcal{L}\left(\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(L)}\right) = -\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\, log(F_{ik}) + \frac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \|\mathbf{F}_{i*} - \mathbf{F}_{j*}\|^2 S_{ij} \tag{3}$$

where $\mathbf{F}i*$ denotes the ith row in the label matrix $\mathbf{F}$ which the output of the deep net. y_{ik} is the ground-truth label distribution of the ith sample. l denotes the number of labeled samples. λ is a trade-off parameter that balances the two terms. The first term is the loss associated with the GCN model. The second term is the label smoothness regularization. Thus, minimizing the first term ensures that the estimated label of a labeled sample is as close as possible to its ground-truth label. On the other hand, minimizing the second terms enforces the label smoothness over the graph. It is obvious that if a pair of samples are very similar (i.e., their S_{ij} is large), then the minimization of the second term tends to produce two labels vectors $\mathbf{F}i*$ and $\mathbf{F}j*$ that are close to each other since the quantity $\|\mathbf{F}_{i*} - \mathbf{F}_{j*}\|^2 S_{ij}$ should as small as possible.

In the literature of spectral analysis, we have the following equality:

$$\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \|\mathbf{F}_{i*} - \mathbf{F}_{j*}\|^2 S_{ij} = Trace(\mathbf{F}^T \mathbf{L}\, \mathbf{F})$$

where $Trace(.)$ denotes the trace of a matrix. Based on the above equality, the loss function in Eq. (3) can be written as:

$$\mathcal{L}(\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(L)}) = -\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\, log(F_{ik}) + \lambda\, Trace(\mathbf{F}^T \mathbf{L}\, \mathbf{F}) \tag{4}$$

In the above loss function, the label matrix $\mathbf{F}$ is linked to the deep net parameters by:

$$\mathbf{F} = f(\mathbf{X}, \mathbf{S}, \mathbf{W}^{(0)}, \mathbf{W}^{(1)}, \ldots) = \text{softmax}\left(\cdots\left(\hat{\mathbf{S}}\, \sigma(\hat{\mathbf{S}}\, \mathbf{X}^{\mathbf{T}}\, \mathbf{W}^{(0)})\, \mathbf{W}^{(1)}\right)\cdots\right) \tag{5}$$

If the output size of all non-final layers is fixed to H, then the total number of the net parameters will be $d.H + H.H + \cdots + H.C$. The sought deep net has $N.C$ outputs given by the elements of the label matrix $\mathbf{F}$. The derivative of the loss function with respect to the element F_{ik} is given by:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial F_{ik}} &= -\frac{y_{ik}}{F_{ik}} + 2\,\lambda\,[\mathbf{L}\,\mathbf{F}]_{ik} \quad \text{if } \mathbf{x}_i \text{ is labeled} \\ &= 2\,\lambda\,[\mathbf{L}\,\mathbf{F}]_{ik} \qquad otherwise \end{aligned} \tag{6}$$

6. Performance evaluation

This section presents the performance of the proposed model on different kinds of image data which include scene, and face datasets. For comparison, we use three public image datasets: Extended Yale, PF01, and Caltech101.

6.1. Image datasets

We used the following datasets:

1. **Extended YALE Face Dataset:** The cropped version that contains 38 individuals has been used in our experiments. The images of the cropped version contain illumination variation and facial expression variation. The images are in gray scale, and we have rescaled them to 32 × 32 pixels in our experiments. The reported results are obtained using semi-supervised learning with 9, 14, and 20 images from each class as labeled samples and the remaining are used as unlabeled samples.
2. **PF01**[1]: It contains the true-color face images of 103 people, 53 men and 50 women, representing 17 different images (1 normal face, 4 illumination variations, 8 pose variations, 4 expression variations) per person. All the people in the dataset are Asians. There are three kinds of systematic variations, such as illumination, pose, and expression variations in the dataset. The images are resized to 32 × 32 pixels. The reported results are obtained using semi-supervised learning with 5, 8, and 12 images from each class as labeled samples and the remaining are used as unlabeled samples.
3. **Caltech101 Dataset**[2]: The Caltech101 dataset used in this paper is the one that contain images of objects belonging to 101 classes. The full Caltech dataset which consists of 256 classes can be found at Griffin, Holub, and Perona (2007). It is a well-known challenging set which contains a set of images of complicated backgrounds. We used a cropped version of the original Caltech dataset which consists of 2,020 images, 20 images for each one of the 101 classes. The reported results are obtained using semisupervised learning with 3, 6, and 9 images from each class as labeled samples and the remaining are used as unlabeled samples. We adopt the deep features provided by the ResNet-50 (He, Zhang, Ren, & Sun, 2016) convolutional neural network. This is a 50 layer convolutional neural network that is trained on the ImageNet database. Using this network, we are able to extract the image representation in the Average Pooling layer. The latter is considered as the image descriptor that has 2048 dimensional vector.

6.2. Experimental setup

We compare the proposed framework GCNMR with some state-of-the-art graph-based semi-supervised methods. These methods belong to three categories of semi-supervised learning: (i) shallow models that perform label propagation over graphs, (ii) shallow models that perform data projection followed by a simple classification, and (iii) deep neural networks (GCN) that predict the labels through a cascade of layers that perform feature propagation over the graph.

We compare the proposed framework GCNMR with the state-of-the-art graph-based semi-supervised algorithm GCN (Kipf & Welling, 2017) and other semi-supervised algorithms including Gaussian Random Field (GRF) (Zhu et al., 2003), Kernel Flexible Model Embedding (KFME) (El Traboulsi et al., 2015), Multi-view Learning with Adaptive Neighbors (MLAN) (Nie et al., 2018), Semi-Supervised Discriminant Embedding (SDE) (Yu et al., 2012), and Exponential Semi-Supervised Discriminant Embedding (ESDE) (Dornaika & Traboulsi, 2017). For a fair comparison, the MLAN semi-supervised classification algorithm is used with a single descriptor that is used by all other methods. In the above three datasets, nodes are images and edges represent pairwise similarities. We treat the links as undirected edges using KNN graph where the number of neighbors is set to 10. The similarity of two images is set to a Gaussian function whose variance is set to the average of all squared distances in the dataset.

Since the GCN method and the proposed GCNMR method rely on a deep neural net they are trained using similar setups. Both models were implemented in TensorFlow (Abadi, et al., 2016). The learning rate is fixed to 0.01, the dropout rate to 0.3, L_2

[1] https://sites.google.com/site/postechimlab2012/databases/face-database-2001

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

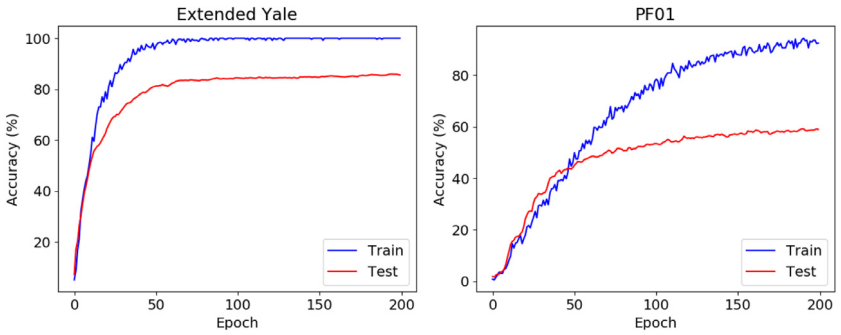


Fig. 2. Classification accuracy (%) obtained by the proposed method for different training epochs. **(a)** Extended Yale dataset. **(b)** PF01 dataset.

Table 2
Recognition performance (Mean recognition accuracy ± Standard deviation %) on Extended Yale dataset over five different random splits.

Extended Yale			
Method	Labeled samples		
	q = 9	q = 14	q = 20
SDE	76.79 ± 5.20	78.87 ± 3.99	83.46 ± 3.73
ESDE	76.65 ± 3.84	84.44 ± 7.00	86.95 ± 3.84
GRF	72.35 ± 2.90	75.73 ± 2.90	78.01 ± 2.66
KFME	72.96 ± 7.81	80.36 ± 5.10	86.11 ± 3.24
MLAN	77.37 ± 3.39	79.68 ± 2.99	81.52 ± 1.67
GCN	78.39 ± 8.50	84.57 ± 3.59	87.61 ± 2.69
GCNMR (Proposed)	**81.95** ± 7.34	**85.65** ± 2.45	**88.36** ± 2.27

Table 3
Recognition performance (Mean recognition accuracy ± Standard deviation %) on PF01 dataset over five different random splits.

PF01			
Method	Labeled samples		
	q = 5	q = 8	q = 12
SDE	49.28 ± 3.8 3	60.10 ± 4.81	56.49 ± 11.96
ESDE	50.89 ± 4.58	61.27 ± 6.29	63.55 ± 12.42
GRF	45.08 ± 3.95	52.44 ± 5.68	50.09 ± 10.85
KFME	49.97 ± 4.35	57.65 ± 6.77	55.66 ± 11.26
MLAN	47.55 ± 4.29	55.39 ± 6.87	53.05 ± 11.60
GCN	50.26 ± 4.69	58.81 ± 7.96	60.37 ± 9.90
GCNMR (Proposed)	**54.15** ± 5.08	**62.63** ± 6.47	**64.74** ± 9.75

regularization weight to zero. We use two convolutional layers and 256 hidden units.

The proposed method has one extra hyper parameter: λ. We set this parameter to a subset of values belonging to $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. We report the recognition accuracy (best average recognition rate) of all methods. All results are obtained with five random splits of the data into a labeled set and an unlabeled set. For labeled sets, different percentages are considered for every image dataset.

Each compared method has a few hyper-parameters. These are searched using a grid search, where each hyper-parameter has 5 values. For each method, its corresponding model is selected based on the best performance it can provide. This strategy is adopted by all compared methods. Furthermore, all compared methods were used on the same splitting into labeled images and unlabeled images. This ensures fair comparisons between all methods.

6.3. Method comparison

Tables 2, 3, and 4 illustrate the average classification rate in % (together with its standard deviation over the five random splits) using different graph-based semi supervised methods for Extended Yale, PF01, and Caltech101 datasets, respectively. For each dataset, three different amounts of labeled samples were used. In each table, there are three columns that correspond to three numbers of labeled images per class. For all results, the number of epochs is limited to 200.

Table 4
Recognition performance (Mean recognition accuracy ± Standard deviation %) on Caltech101 dataset over five different random splits.

Caltech101			
Method	Labeled samples		
	q = 3	q = 6	q = 9
SDE	61.65 ± 2.75	77.90 ± 1.51	83.22 ± 0.92
ESDE	72.23 ± 0.88	79.23 ± 0.41	81.96 ± 0.73
GRF	76.96 ± 1.67	80.01 ± 0.59	81.87 ± 0.68
KFME	77.76 ± 1.37	80.90 ± 0.69	83.42 ± 0.59
MLAN	76.61 ± 1.35	79.82 ± 0.89	81.35 ± 1.05
GCN	77.94 ± 0.61	82.12 ± 0.47	83.93 ± 0.49
GCNMR (Proposed)	**78.73** ± 0.73	**82.43** ± 0.64	**84.07** ± 0.64

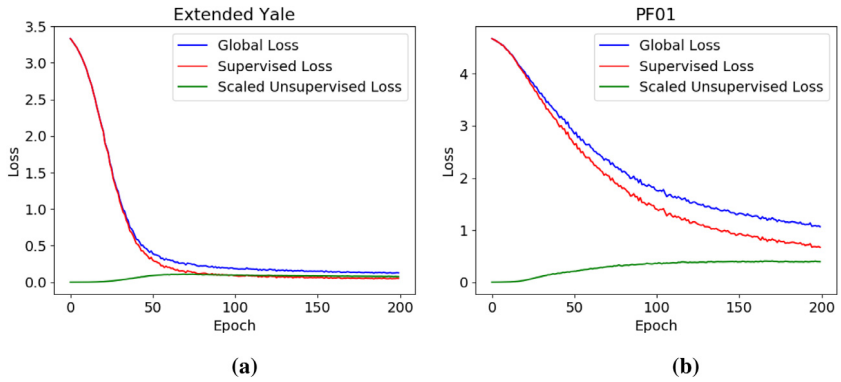
6.4. Convergence

Since the learning of the proposed model GCNMR is based on gradient descent, it is interesting to study the dynamic of learning. To this end, we consider the Extended Yale and PF01 datasets. Fig. 2.a illustrates the classification accuracy obtained on the Extended Yale dataset for different training epochs. At each number of epochs, the associated learned network is used in order to estimate the labels of both the training and test data samples. The number of labeled images per class is fixed to 9. Fig. 2.b illustrates the classification accuracy obtained on the PF01 dataset for different training epochs. The number of labeled images per class is fixed to 5.

Fig. 3 illustrates the global loss function (Eq. (4)) (blue curve), the supervised loss ($-\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\, log(F_{ik})$) (red curve) and the scaled unsupervised loss ($\lambda\, Trace(\mathbf{F}^T\mathbf{L}\mathbf{F})$) (green curve) as a function of the epoch number. Fig. 3.a corresponds to the Extended Yale dataset. Fig. 3.b corresponds to PF01 dataset. For both datasets, λ was set to 10^{-6}. As it can be seen, during the first stages of learning the supervised loss decreases rapidly.

6.5. Effect of the trade-off between supervised loss and manifold regularization loss

The proposed model has λ as a trade-off parameter that determines the proportion of the unsupervised loss in the total loss. Fig. 4 illustrates the classification accuracy as a function of λ on the test data of two datasets: Extended Yale and PF01. From this



(a) **(b)**

Fig. 3. Global loss function (blue curve), supervised loss (red curve) and scaled unsupervised loss (green curve) as a function of the epoch number. **(a)** Extended Yale dataset. **(b)** PF01 dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

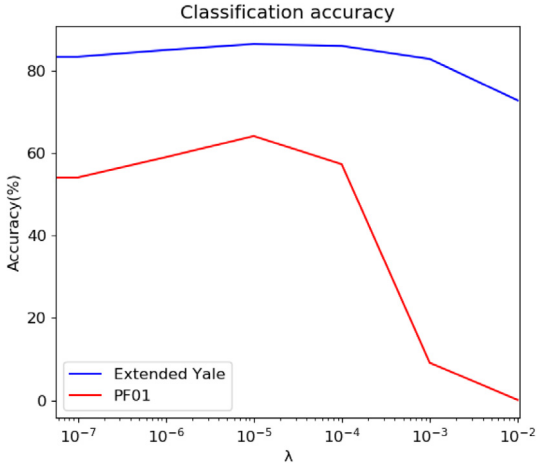


Fig. 4. Classification accuracy of the proposed model GCNMR as a function of the trade-off parameter, λ. The results are obtained with the Extended Yale and PF01 datasets.

figure, it can be easily seen that the optimal range of values for both datasets is $[10^{-6}, 10^{-4}]$.

6.6. Effect of number of layers

In this section, we study the effect of the number of layers used by the deep neural network of the proposed model GCNMR. Fig. 5 illustrates the training and test accuracy as a function of the number of layers for the Extended Yale, PF01, and Caltech101 datasets. In these experiments, the number of epochs is fixed to 400.

6.7. Analysis of results

From the obtained results depicted in the previous tables and figures, we can draw the following conclusions:

- Our proposed model has achieved very good performance in all three datasets: Extended Yale, PF01, and Caltech101.
- In all configurations, the performance of the proposed GCNMR model was better than that of the GCN model. For the first two datasets, the superiority of the proposed method is obvious. This suggests that the use of manifold regularization in the global loss function can enhance the estimation of the unknown labels. In the experiments associated with the

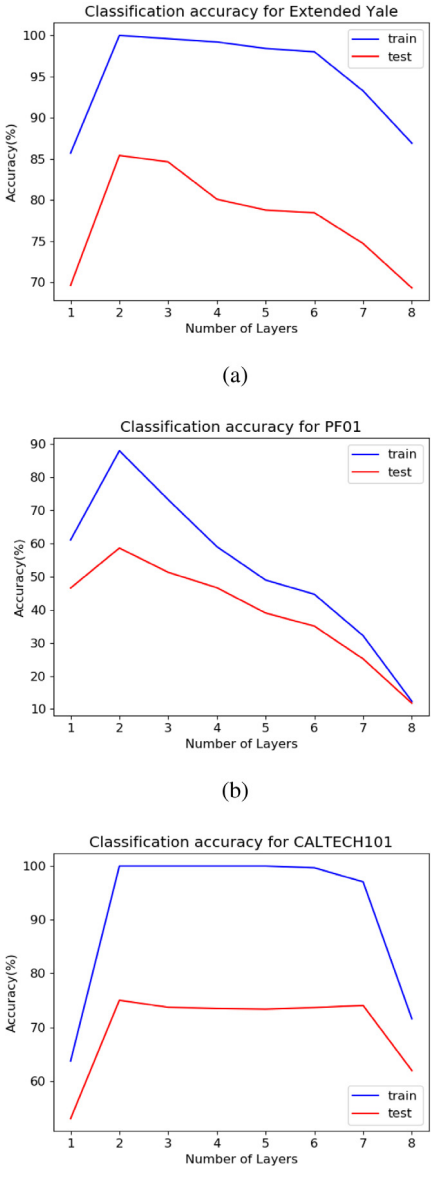


Fig. 5. Classification accuracy as a function of the number of layers.

PF01 dataset, the improvement of the GCNMR model over the GCN model can be more than 4%.
- The learning process of the proposed GCNMR model can be achieved in 200–400 epochs.
- In the experimental setup used by the proposed GCNMR model, using more than two layers has not improved the performance of the semi-supervised classification.

7. Conclusion

We introduced an enhanced Graph Convolution Network model for semi-supervised learning. The model adopts a global loss that integrates supervised and unsupervised information. The unsupervised information is imposed by manifold regularization. The deep neural network of the proposed model has the same structure as the classic GCN. Thus, virtually the training and testing complexity is the same. However, the loss function of the proposed model includes an additional information retrieved from both labeled and unlabeled samples. Our proposed model achieves better generalization ability on unlabeled data.

The proposed method inherits the advantages of semi-supervised learning. It will also benefit the general frameworks of deep graph-based semi-supervised learning in the sense that these ones can have some performance improvement. Future work will investigate two tracks. On the one hand, we will consider other unsupervised loss functions in the global loss function. On the other hand, we will explore ways that are able to transform the GCNMR to a learner able to estimate the labels of unseen samples. In other words, we investigate an inductive GCNMR.

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.

Acknowledgment

This work was funded by the Spanish Ministerio de Ciencia, Innovacion y Universidades, Programa Estatal de I+D+i Orientada a los Retos de la Sociedad, RTI2018-101045-B-C21.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint.

Ashfaq, R. A. R., Wang, X.-Z., Huang, J. Z., Abbas, H., & He, Y.-L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, *378*, 484–497.

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems* (pp. 585–591).

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research (JMLR)*, *7*, 2399–2434.

Cai, D., He, X., & Han, J. (2007). Semi-supervised discriminant analysis. In *2007 IEEE 11th international conference on computer vision* (pp. 1–7).

Cai, D., He, X., & Han, J. (2007). Semi-supervised discriminant analysis. In *IEEE International conference on conputer vision*.

Chen, H.-T., Chang, H.-W., & Liu, T.-L. (2005). Local discriminant embedding and its variants. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on (vol. 2)* (pp. 846–853). IEEE.

Dai, D., & Van Gool, L. (2016). Unsupervised high-level feature learning by ensemble projection for semi-supervised image classification and image clustering. arXiv preprint arXiv:1602.00955.

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).

Dornaika, F., & El Traboulsi, Y. (2016). Learning flexible graph-based semi-supervised embedding. *IEEE Transactions on Cybernetics*, *46*(1), 206–218.

Dornaika, F., & El Traboulsi, Y. (2017). Matrix exponential based semi-supervised discriminant embedding for image classification. *Pattern Recognition*, *61*, 92–103.

Dornaika, F., & Traboulsi, Y. E. (2017). Matrix exponential based semi-supervised discriminant embedding. *Pattern Recognition*, *61*, 92–103.

El Traboulsi, Y., Dornaika, F., & Assoum, A. (2015). Kernel flexible manifold embedding for pattern classification. *Neurocomputing*, *167*, 517–527.

Gong, C., Tao, D., Maybank, S. J., Liu, W., Kang, G., & Yang, J. (2016). Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, *25*(7), 3249–3260.

Griffin, G., Holub, A., & Perona, P. (2007). Caltech-256 object category dataset.

Guo, H., Zou, H., & Tan, J. (2020). Semi-supervised dimensionality reduction via sparse locality preserving projection. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems* (pp. 1024–1034).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hou, C., Nie, F., Li, X., Yi, D., & Wu, Y. (2014). Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, *44*(6), 793–804.

Huang, H., Liu, J., & Pan, Y. (2012). Semi-supervised marginal fisher analysis for hyperspectral image classification. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *1*, 3.

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. In *NIPS workshop on bayesian deep learning*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*.

Klicpera, A. B. J., & Gunnemann, S. (2019). Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*.

Liu, W., & Chang, S.-F. (2009). Robust multi-class transductive learning with graphs. In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on* (pp. 381–388). IEEE.

Liu, J., Zhao, M., & Kong, W. (2019). Capped l2, 1-norm regularization on kernel regression for robust semi-supervised learning. In *2019 IEEE symposium on product compliance engineering - Asia*.

Martínez, A. M., & Kak, A. C. (2001). Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(2), 228–233.

Nie, F., Cai, G., Li, J., & Li, X. (2018). Auto-weighted multi-view learning for image clustering and semi-supervised classification. *IEEE Transactions on Image Processing*, *27*(3), 1501–1511.

Nie, F., Tian, L., Wang, R., & Li, X. (2019). Multiview semi-supervised learning model for image classification. *IEEE Transactions on Knowledge and Data Engineering*.

Nie, F., Wang, Z., Wang, R., & Li, X. (2020). Submanifold-preserving discriminant analysis with an auto-optimized graph. *IEEE Transactions on Cybernetics*.

Nie, F., Xu, D., Tsang, I. W.-H., & Zhang, C. (2010). Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, *19*(7), 1921–1932.

Qiao, L., Chen, S., & Tan, X. (2010). Sparsity preserving discriminant analysis for single training image face recognition. *Pattern Recognition Letters*, *31*(5), 422–429.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Sousa, C., Rezende, S., & Batista, G. (2013). Influence of graph construction on semi-supervised learning. In *European conferene on machine learning* (pp. 160–175).

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.

Traboulsi, Y. E., Dornaika, F., & Ruichek, Y. (2018). Semi-supervised two phase test sample sparse representation classifier. *Knowledge-Based Systems*, *160*, 16–27.

Vanegas, J. A., Beltran, V., Escalante, H. J., & Gonzalez, F. A. (2019). Transductive non-linear semantic embedding for multi-class classification. *Pattern Recognition Letters*.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

Velickovic, P., Fedus, W., Ló, P., & Bengio, Y. (2019). Deep graph infomax. In *ICLR*.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *ICLR*.

Yu, S., Yang, X., & Zhang, W. (2019). Pkgcn: prior knowledge enhanced graph convolutional network for graph-based semi-supervised learning. *International Journal of Machine Learning and Cybernetics*, *10*(11), 3115–3127.

Yu, G., Zhang, G., Domeniconi, C., Yu, Z., & You, J. (2012). Semi-supervised classification based on random subspace dimensionality reduction. *Pattern Recognition*, *45*(3), 1119–1135.

Yuan, Y., Mou, L., & Lu, X. (2015). Scene recognition by manifold regularized deep learning architecture. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(10), 2222–2233.

Zhang, J., Zhang, P., Li, B., Jing, L., & Lv, T. (2019). Semisupervised feature extraction based on collaborative label propagation for hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 1–5.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, *16*(16), 321–328.

Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. vol. 3, (pp. 912–919).