Pattern Recognition 74 (2018) 15–24

Contents lists available at ScienceDirect

Pattern Recognition

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

Revisiting multiple instance neural networks

Xinggang Wang*, Yongluan Yan, Peng Tang, Xiang Bai*, Wenyu Liu

School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 12 April 2017
Revised 24 July 2017
Accepted 23 August 2017
Available online 31 August 2017

Keywords:
Multiple instance learning
Neural networks
Deep learning
End-to-end learning

ABSTRACT

Of late, neural networks and Multiple Instance Learning (MIL) are both attractive topics in the research areas related to Artificial Intelligence. Deep neural networks have achieved great successes in supervised learning problems, and MIL as a typical weakly-supervised learning method is effective for many applications in computer vision, biometrics, natural language processing, and so on. In this article, we revisit Multiple Instance Neural Networks (MINNs) that the neural networks aim at solving the MIL problems. The MINNs perform MIL in an end-to-end manner, which take bags with a various number of instances as input and directly output the labels of bags. All of the parameters in a MINN can be optimized via back-propagation. Besides revisiting the old MINNs, we propose a new type of MINN to learn bag representations, which is different from the existing MINNs that focus on estimating instance label. In addition, recent tricks developed in deep learning have been studied in MINNs; we find *deep supervision* is effective for learning better bag representations. In the experiments, the proposed MINNs achieve state-of-the-art or competitive performance on several MIL benchmarks. Moreover, it is extremely fast for both testing and training, for example, it takes only 0.0003 s to predict a bag and a few seconds to train on MIL datasets on a moderate CPU.

1. Introduction

Multiple Instance Learning (MIL) was originally proposed for drug activity prediction [1]. Now it has been widely applied to many domains and is an important problem in machine learning. Many multimedia data have the Multiple Instance (MI) structure; for example, a text article contains multiple paragraphs, an image can be divided into multiple local regions, and a gene expression data contains multiple genes. MIL is useful to processing and understanding MI data.

Multiple instance learning is a kind of Weakly-Supervised Learning (WSL). Each sample is in the form of labeled bags, composed of a wide diversity of instances associated with input features. The aim of MIL, in a binary task, is to train a classifier to predict labels of testing bags, which is based on the assumption that a positive bag contains at least one positive instance, whereas a bag is negative if it is only constituted of negative instances. Thus, the crux of MIL is to deal with the ambiguity of the labels of the instances, especially in positive bags that have plenty of cases with different compositions.

There are many algorithms have been proposed to solve the MIL problem. According to the survey by Amores [2], MIL algorithms are in three folds: instance-space paradigm, bag-space paradigm, and embedded-space paradigm. Instance-space paradigm learns the instance classifier and performs bag classification by aggregating the responses of instance-level classifier. Bag-space paradigm exploits bag relations and treats bag as a whole; in particular, it calculates bag-to-bag distance/similarity; then the nearest neighbor or Bayesian classifier carries out bag classification based on the distances/similarities. Embedded-space paradigm embeds a bag into a vocabulary-based feature space to obtain a compact representation for the bag, for example, a vector representation; then classical classifiers can be applied to solve the bag classification problem.

Deep neural networks have been applied to solve many machine learning problems. For supervised learning, there are several kinds of neural networks. Deep Belief Networks (DBN) [3] use unsupervised pre-training and take a fixed length vector as input for feature learning, regression, and classification. Deep Convolutional Neural Networks (CNN) [4,5] take images as input and have dominated many computer vision problems. Deep Recurrent Neural Networks (RNN) [6] and Long Short Term Memory (LSTM) networks [7] take sequential data as input, such as text and speech, and are good at dealing with sequence prediction problems. Usually, training these deep networks requires a huge number of fully labeled data, that is, each training sample/instance needs a label.

* Corresponding author.
E-mail addresses: xgwang@hust.edu.cn (X. Wang), yongluanyan@hust.edu.cn (Y. Yan), pengtang@hust.edu.cn (P. Tang), xbai@hust.edu.cn, xiang.bai@gmail.com (X. Bai), liuwy@hust.edu.cn (W. Liu).

http://dx.doi.org/10.1016/j.patcog.2017.08.026
0031-3203/© 2017 Elsevier Ltd. All rights reserved.

However, in MIL, only bag-level labels are given. Meanwhile, MI data have a more complex structure which is a set of instances in various size. Also, MI data is different from the sequential data mentioned above, since there is no order information between instances. These problems make it difficult to deal with the MIL problem by conventional neural networks.

Before the raising of deep learning, some research studies were trying to solve the MIL problem using neural networks. In the year of 2000, Ramon and Raedt [8] firstly proposed a Multiple Instance Neural Network (MINN). The network estimates instance probabilities before the last layer and calculates bag probability using a convex max operator (i.e., log-sum-exp). The network was trained using back-propagation. Then, Zhang and Zhou [9] also proposed a multiple instance network that calculates bag probability by directly taking the max of instance probabilities.

A MINN takes a bag with multiple instances as input. Instance-level representation is gradually learned layer by layer guided by bag-level supervision. To inject the bag-level representation, there are two different network architectures. Following the naming style in a classical MIL study [10], we name the two networks as mi-Net and MI-Net, which aim at dealing with the MIL problem in instance-space paradigm and embedded-space paradigm [2], respectively. In mi-Net, there are instance classifiers in the each layer. We can obtain instance predictions for both training and testing bags, which is an appealing property in some applications. Different from MI-Net, there is no instance classifier. It directly builds a fixed-length vector as the bag representation and then learns bag classifier. Compared with mi-Net, MI-Net can obtain better bag classification accuracy. The previous studies are in the mi-Net category. We newly propose MI-Net in this article.

A key component in MINN is MIL Pooling Layer (MPL), which aggregates either instance probability distribution vectors or instance feature vectors into a bag probability/feature vector. It bridges MI data with conventional neural networks. As it must be differentiable, there are a few choices, such as max pooling, mean pooling, and log-sum-exp pooling. These pooling methods are compared and discussed in the experiments section. Besides MIL pooling layer, we use fully-connected layers with non-linear activations for instance feature learning. In MIL benchmarks, instance features are hand-crafted and raw data of instances are given. Even so, it is beneficial to do feature transformation guided by the bag-level supervision. Finally, for MI-Net, we use a fully-connected layer with only one neuron to match the predicted bag label with ground-truth in training.

Training neural networks using complex MI data is a challenging task. To learn good instance feature, we have tried to adopt various recent progresses of deep learning in MINN, such as dropout [11], Rectified Linear Unit (ReLU) [12], Deeply Supervised Nets (DSN) [13] and Residual Connections [14]. We find DSN is the most effective one because DSN can fuse the hierarchical features to make a better decision. Besides, residual connections are also helpful in MINNs.

To summarize, we revisit the problem of solving MIL using neural networks (MINNs), which are ignored in current MIL research community. Our experiments show that MINNs is very effective and efficient. Different from most MIL algorithms, MINNs optimize instance feature learning, bag feature learning, instance classification, and bag classification in a fully end-to-end manner via back-propagation. This article focuses on MINNs with comprehensive studies on MIL benchmarks. The main contributions of this article include two extremely fast and scalable methods for MIL, mi-Net, and MI-Net, and introducing deep supervision and residual connections for MIL.

The rest of this article is organized as follows. Section 2 briefly reviews previous studies on MIL. In Section 3, we propose end-to-end MIL networks. Our experimental results are presented on several MIL benchmarks in Section 4. Some discussions of experimental setups are presented in Section 5. Finally, in Section 6, we conclude the article with some future studies.

2. Related work

The previous MIL works based on neural networks were mainly proposed by Zhou et al. and Ramon et al. in [8,9,15,16]. and Raedt [8] introduced the use of a log-sum-exp function as the convex max to calculate bag probabilities from instance probabilities. Zhou and Zhang [9] changed to a different loss function and directly applied max function. Zhang and Zhou [15] improved multiple instance neural networks by feature selection using Diverse Density and PCA. Zhang and Zhou [16] showed that ensemble methods could be integrated with multiple instance neural networks. Subsequently, solving MIL using neural networks has been ignored in machine learning research. This article revisits this problem, proposes some new network structures, and investigates some of the recent neural network tricks. The idea of using neural networks for solving MIL problem has been studied in some computer vision studies, such as [17,18]. Wu et al. [17] proposed a deep MIL which uses max pooling to find positive instances/patches for image classification and annotation. Pinheheiro et al. [18] used log-sum-exp pooling in deep CNN for weakly supervised semantic segmentation. The studied mi-Net follows the path of these two works [17,18]. are applications of mi-Net. Thus, it is not necessary to compare to them in the experiments. In addition, in this article, we study the variants of mi-Net that utilize deep supervision and focus on more general MIL problems. Besides integrating MIL into deep neural networks, Wang et al. proposed a method to combine MIL with support vector machine using a relaxed MIL constraint [19] and applied this for object discovery. However, they pay more attention to vision applications (e.g., image classification, image annotation, and semantic segmentation, etc.), which are based on convolutional image features. Meanwhile, they always fine-tune neural network models pre-trained on other much larger datasets such as ImageNet [20]. Moreover, they only focus more on instance-space MIL. We focus on applying MINNs for more general MIL problems. Notice that for general MIL problems, there are no available large datasets for pre-training such as computer vision, which makes it harder to train MINNs efficiently. As we will show in experiments, training [8,9,17] like MINNs on such small scale MIL benchmarks directly cannot obtain satisfactory results. To solve this problem, We show many tricks to train our networks from the start on MIL benchmarks with limited training data, and have achieved many inspiring results. Meanwhile, we have investigated both mi-Net and MI-Net, and experiments have shown that MI-Net outperforms mi-Net in more cases.

Learning effective representation from (weakly-supervised) data, especially MIL, has received a lot of attention as it helps solve a range of real applications [21–25]. Till date, numerous MIL methods have been proposed to either develop effective MIL solvers or apply MIL to solve real application problems [26,27]. A comprehensive survey of MIL algorithms and applications can be found in [2]. Here, we focus on a brief review of the most recent MIL algorithms, especially the ones related to deep neural networks and feature learning. From the view of embedded-space paradigm for MIL, the most recent method is the scalable MIL algorithm, which solves MIL using Fisher Vector (FV) coding [28], called miFV [17]. miFV transforms instance feature into high-dimensional space using an unsupervised learned Gaussian Mixture Model (GMM) and FV coding. The proposed MI-Net learns instance feature using deep multiple instance supervision. In addition, MI-Net achieves better bag classification accuracy and is much faster than miFV.

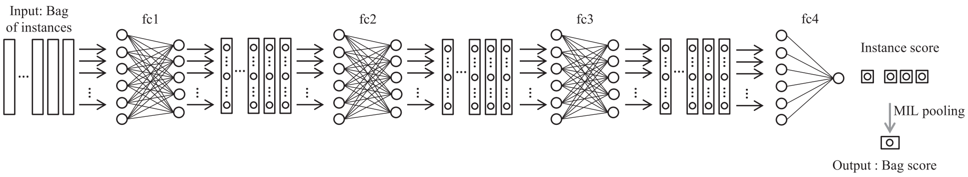

Fig. 1. A mi-Net with four fully-connected layers. The number of output of fully-connected layers are 256, 128, 64, and 1, respectively. The last layer is a MIL Pooling Layer with instance probabilities as input and bag probability as output.

3. Multiple instance neural networks

In this section, we firstly introduce the formulation of MIL, then give various networks for MIL, and lastly study the MIL pooling methods and training loss.

3.1. Notations

Here we first review the definition of MIL. Given a set of bags $X = \{X_1, X_2, \ldots, X_N\}$ and instance features of ith bag $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im_i}\}, x_{ij} \in \mathbb{R}^{d \times 1}$, where N and m_i denote the number of bags and the number of instances in bag X_i respectively. Suppose $Y_i \in \{0,\ 1\}$ and $y_{ij} \in \{0,\ 1\}$ are the label of bag X_i and instance x_{ij} separately, where 1 means positive and 0 means negative. In MIL, only bag labels are given during training, and there are two MIL constraints:

- If bag X_i is negative, then all instances in X_i will be negative, that is, if $Y_i = 0$, then all $y_{ij} = 0$;
- If bag X_i is positive, then at least one instance in X_i will be positive, that is, if $Y_i = 1$, then $\sum_{j=1}^{m_i} y_{ij} \geq 1$.

The most challenging issue in MIL is that the instance label is not given. In MINNs, there are two strategies: the first one is to infer instance label in the network, that is, placing instance probabilities of being positive as a hidden layer in the network; the second one is to learn bag representation in the network and directly carry out bag classification without calculating instance probability. The first strategy had been studied in [8,9,17]. The second strategy is newly proposed in this article. In the following sub-sections, we will give the descriptions of MINNs.

Let us consider a setting of a single bag X_i with multiple instances x_{ij} that is passed through a MINN. A MINN is made out of L layers, each of which contains a non-linear transformation $H^{\ell}(\cdot)$, where ℓ indexes the layer. $H^{\ell}(\cdot)$ can be a composite of operations such as inner product (or fully-connection), rectified linear units (ReLU) [29], or proposed MIL pooling. We denote the output of the ℓth layer of an instance x_{ij} as x_{ij}^{ℓ}.

3.2. Mi-Net: an instance-Space MINN

Initially, we review traditional multiple instance neural networks [8,9,17], which are named as mi-Net. As shown in Fig. 1, each instance in a bag is first fed into several fully-connected (FC) layers with an activation function (in this article we use four FC layers with the ReLU activation [29]). We obtain the instance feature denoted as x_{ij}^{L-2} in the $(L-2)$th layer and the instance probability denoted as p_{ij}^{L-1}. p_{ij}^{L-1} is a scalar in the range of [0, 1] and is inferred from x_{ij}^{L-2} individually. In the last layer, there is a MIL Pooling Layer (described in Section 3.6), which takes instance probabilities as input and outputs bag probability denoted as $P^L(X_i)$.

These first $L-2$ layers can learn some more semantic instance features compared with original x_{ij} (higher layer corresponding to higher semantic features). After learning these instance features, a FC layer which only has one neuron with a sigmoid activation is used to predict the positiveness of instances.

Unlike traditional neural networks, for mi-Net, we only have bag labels for training, but instance labels are not available. To address this problem, we treat the instance labels as latent variables and infer them during the network training. We design a layer to aggregate instance scores into bag score. Here, a MIL Pooling Layer is used to aggregate these instance scores into the final positiveness of bag.

The MIL pooling method satisfies the MIL constraints: If a bag is positive, there should be at least one instance with large positiveness. Otherwise, all instances in the bag should have low positiveness. As the pooling layer is integrated into the neural network, the pooling function should be differentiable. The typical MIL pooling is introduced in Section 3.6. In summary, the mi-Net can be formulated as:

$$\begin{cases} x_{ij}^{\ell} = H^{\ell}(x_{ij}^{\ell-1}), \\ P_i^L = M^L(p_{ij|j=1\ldots m_i}^{L-1}). \end{cases} \tag{1}$$

In mi-Net, the formulation of the last two layers is: $P_i^L = M^L(p_{ij|j=1\ldots m_i}^{L-1})$. P_i^L is the bag probability and the M^L is a MIL operator. Thus, the neurons of the second to last layer (i.e. the $(L-1)$th layer) represent the instance probabilities.

3.3. MI-Net: a new embedded-Space MINN

We propose a series of new multiple instance neural networks that do not rely on inferring instance probability. The networks directly learn bag representation and produce better bag classification accuracy. These methods belong to the category of embedded-space MIL algorithms defined in survey [2]. Following the naming style in [10], we name this network as MI-Net.

In Fig. 2, we show a plain MI-Net with three fully-connected layers and one MIL Pooling Layer. The change of network structure leads the network to focus on learning bag representation, rather than predicting instance probability. No matter how many input instances there are, the MIL Pooling Layer aggregates them into one feature vector as a bag representation. Finally, a FC layer with only one neuron and sigmoid activation takes the bag representation as input and predicts bag probability. This plain MI-Net is formulated as:

$$\begin{cases} x_{ij}^{\ell} = H^{\ell}(x_{ij}^{\ell-1}), \\ X_i^{\ell} = M^{\ell}(x_{ij|j=1\ldots m_i}^{\ell-1}). \end{cases} \tag{2}$$

The difference between MI-Net and mi-Net. Firstly, we may compare Figs. 1 and 2 to find the difference between mi-Net and MI-Net. In mi-Net, there are some nodes representing instance scores. In MI-Net, there are no instance scores; instead, it contains a bag feature vector after the blue arrow. From the view of feature learning, mi-Net focuses on learning instance representation; while, MI-Net learns both instance representation and bag representation. We have a clear motivation of designing MI-Net. Since mi-Net predicts an instance score based on the individual instance and the bag score depends on the instance scores, bag classification will fail if the instance classifiers make mistake. Our motivation of MI-Net is to obtain a richer representation for a bag by

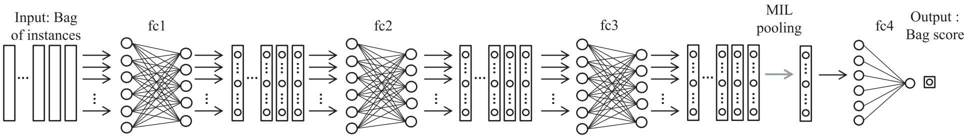

Fig. 2. The proposed MI-Net with three fully-connected layers and one MIL pooling layer. The number of output of fully-connected layers are 256, 128, and 64, respectively.

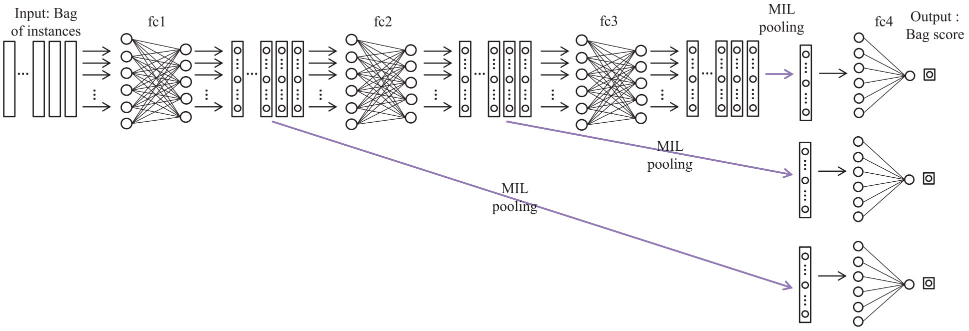

Fig. 3. The proposed MI-Net with deep supervision. There are three fully-connected layers for learning instance features which are in the size of 256, 128, and 64, respectively. There are three MIL Pooling Layers for generating bag feature, and the bag features are connected to the bag label via a fully-connected layer with one neuron, respectively.

aggregating all instance features and then give more robust prediction based on the bag representation. Since multi-instance learning is weakly-supervised learning problem; either explicitly or implicitly inferring instance probabilities, it always has a risk to make mistake. However, in MI-Net, it only focuses on the bag classification task; thus, the weakly-supervised MIL problem becomes a fully-supervised bag classification problem. This is the reason why MI-Net tends to give better bag classification accuracy as shown in the experiments. However, there is a limitation in MI-Net; it is not able to give the instance probability. In the applications that require instance probability, MI-Net is not appropriate.

3.4. MI-Net With deep supervision

Inspired by the Deeply-Supervised Nets (DSN) [13], we add deep supervisions in MI-Net as shown in Fig. 3. That is, each middle FC layer that can learn instance features, is followed by a MIL pooling layer and a FC layer for predicting bag score. During training, the supervision is added to each level. In addition, during testing, we compute the mean score for each level. The MI-Net with deep supervision is formulated as:

$$\begin{cases} x_{ij}^{\ell} = H^{\ell}(x_{ij}^{\ell-1}), \\ X_i^{\ell,k} = M^{\ell}(x_{ij|j=1\dots m_i}^{k}), k \in \{1,2,3\}, \end{cases} \tag{3}$$

where the index k in $X_i^{\ell,k}$ means we learn multiple bag features from all levels of instance features by MIL pooling. MI-Net with deep supervision can utilize multiple hierarchies to get better bag classification accuracy. It can be interpreted from two perspectives: (1) In training, instance feature in bottom layers can receive better supervision; and (2) in testing, we can average multiple bag probabilities to get a more robust bag label. In this article, we set the weights of different levels equally.

3.5. MI-Net with residual connections

Recently, deep residual learning was proposed in [14] and showed an impressive improvement in image recognition by utilizing very deep neural networks. We study the residual connections in MI-Net as shown in Fig. 4. A MI-Net with residual connections is formulated as:

$$\begin{cases} x_{ij}^{\ell} = H^{\ell}(x_{ij}^{\ell-1}), \\ X_i^{1} = M^{\ell}(x_{ij|j=1\dots m_i}^{1}), \\ X_i^{\ell} = M^{\ell}(x_{ij|j=1\dots m_i}^{\ell}) + X^{\ell-1}, \ell > 1. \end{cases} \tag{4}$$

Different from the original residual learning in [14] which learns representation residuals using convolution, batch normalization, and ReLU, we learn the bag representation residuals via fully-connected layers, ReLU, and MIL pooling. In the end of the network, final bag representation is connected to the bag label via a FC layer with one neuron and sigmoid activation.

3.6. MIL pooling methods

As referred before, we use a MIL Pooling Layer to get bag scores or bag representations. In this article, we use three popularly used MIL pooling methods: max pooling, mean pooling, and log-sum-exp (LSE) pooling, as shown in Eq. (5), where f_i is the input, o is the output, m is the number of input, and r is a hyper-parameter. All these methods satisfy the constraints referred in Section 3.2. The LSE [30] is a smooth version and convex approximation of the max function. The hyperparameter r controls the smoothness of approximation. That is, it is more approximate to max when r is large and more approximate to mean when r is small.

$$\begin{cases} \text{max}: & M^{\ell}(x_{ij|j=1\dots m_i}^{\ell-1}) = \max_j x_{ij}^{\ell-1}, \\ \text{mean}: & M^{\ell}(x_{ij|j=1\dots m_i}^{\ell-1}) = \dfrac{1}{m_i}\sum_{j=1}^{m_i} x_{ij}^{\ell-1}, \\ \text{LSE}: & M^{\ell}(x_{ij|j=1\dots m_i}^{\ell-1}) = r^{-1}\log\left[\dfrac{1}{m_i}\sum_{j=1}^{m_i}\exp(r\cdot x_{ij}^{\ell-1})\right]. \end{cases} \tag{5}$$

3.7. Training loss

For both mi-Net and MI-Net, we can obtain the bag scores. Here, we define the loss function during training. As we are aiming at predicting labels of bags, it is natural to choose the cross entropy loss function, as in Eq. (6), where S_i is the bag score of ith bag. This loss is added to each bag scores for deep supervision.

$$\text{Loss}(S_i, Y_i) = -\{(1-Y_i)\log(1-S_i) + Y_i \log S_i\}. \tag{6}$$

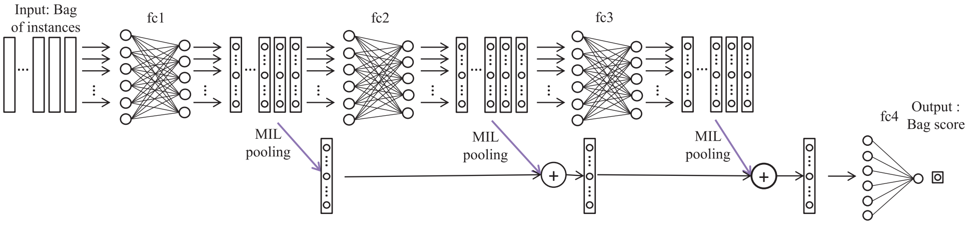

Fig. 4. The proposed MI-Net with residual connections. The first fully-connected layer produces a bag feature vector. The latter fully-connected layers learn the residuals of bag representation. The size of all fully-connected layers is 128.

As all parts of the proposed MINNs are differentiable, we can train these networks by standard back-propagation with Stochastic Gradient Descent (SGD).

4. Experiments

In this section, we perform experiments to test mi-Net, MI-Net, and its variations on different MIL benchmarks, including drug activity prediction, automatic image annotation, and text categorization.

4.1. Datasets

Drug Activity Prediction. MUSK [1] datasets are used to predict whether a drug molecule can bind well to target protein. Each molecule is exhibited as multiple shapes, which are described as 166-dimension features. In the MIL problem, we can regard a molecule as a bag and represent different shapes belonging to the same molecule as instances of this bag. 476 instances are included in MUSK1, which is divided into 47 positive bags and 45 negative bags, whereas 6598 instances are included in MUSK2, which is divided into 39 positive bags and 63 negative bags.

Automatic Image Annotation. The Elephant, Fox, and Tiger datasets [10], are all composed of 100 positive bags from the target class of animal images and 100 negative bags randomly chosen from other class of animal images. Here, an image is represented as a bag, which contains a set of regions we called instances in MIL problems. When searching for a target object, we use this network to obtain the keywords of images. Moreover, each image is represented by 2 to 13 instances, which have 230-dimension features that describe the color, texture, and shape in regions of an image.

Text Categorization. Besides the above datasets, we validate our method on some text categorization dataset since the text is naturally a MI data and MIL is effective to solve this problem. Here, we take 20 datasets derived from the 20 Newsgroups corpus [31]. In each category, 100 bags are included among which half of the bags are positive, and the rest of bags are negative. Each positive bag contains 3% posts from the target class and the rest of posts are from other categories, whereas the instances of negative bags are all randomly drawn from other categories. In addition, each instance is represented by the top 200 Term Frequency, Inverse Document Frequency (TF-IDF) features.

Detailed characteristics of these datasets are summarized in Table 1.

4.2. Experimental setup

These neural networks contain four fully-connected (FC) layers and the first three FC layers are followed by a dropout layer (0.5 dropout ratio). As referred in Section 3, we present the performance of the proposed MIL approaches: (1) In mi-Net, we learn instance scores from four FC layers and aggregate instance scores into bag scores to predict the label of the bag via MIL Pooling Layer. (2) In MI-Net, the input instances are aggregated into bag representation by the first three FC layers and a MIL Pooling Layer, and then the last FC layer is used to predict the bag probability. (3) In MI-Net with Deep Supervision (MI-Net with DS), different from MI-Net, each intermediate FC layer is followed by a MIL Pooling Layer and FC layer to predict bag scores. The loss function of MI-Net with DS sums up all intermediate entropy losses to do back-propagation with SGD for training, and the average of all bag scores is used for testing. (4) In MI-Net with Residual Connections (MI-Net with RC), residual connections are set between each intermediate bag representation, followed by a FC layer to obtain bag score.

In experiments, we use default hyper-parameters given as follows. As for the numbers of neurons in FC layers, there are 256, 128, 64, and 1 in mi-Net, MI-Net, and MI-Net with DS, whereas there are 128, 128, 128, and 1 in MI-Net with RC. Weights of FC layers are all initialized using a glorot-uniform distribution [32]. Biases are all initialized to be 0. For different datasets, suitable values are set for the learning rate, weight decay, momentum and the MIL pooling functions (in Eq. (5)) are searched using cross-validations on training data, which are given in the configuration file of our code. All networks are trained with SGD, and one bag is input as a batch for training and testing. Moreover, regarding the training and testing time, for example, it takes only 0.0003 second to predict a bag and 0.0008 second to train on MUSK1 dataset on a moderate CPU, which is comparable to miVLAD [33] and miFV [33] (for training and testing, it costs 0.018 second and less than 0.001 second, respectively), and much more efficient than other classical MIL methods including mi-Graph [31], mi-SVM [10], MI-SVM [10], MI-Kernel [34], and EM-DD [35]. Our code is written in Python, based on Keras [36], and all of our experiments are run on a PC with Inter(R) i7-4790K CPU (4.00GHZ) and 32GB RAM. The code for reproducing results will be available upon acceptance.

4.3. Experimental results

In this subsection, we give the results of the studied MINNs and compare to the state-of-the-arts on the tasks of drug activation prediction, automatic image annotation, and text categorization. The results are shown in Tables 2, 3, and 4, respectively. To avoid the bias in the datasets, we follow the standard evaluation protocol and the studied MINNs using 10-fold cross-validations for 5 times. The average bag classification accuracy and its standard deviation are reported. The results of the compared methods are quoted from the original publications contain both average bag classification accuracy and standard deviation. There is an exception that the standard deviations of mi-SVM [10], MI-SVM [10], and MI-Kernel [34] are not available in their original papers.

In the tables of results, the best performance on each dataset is in bold, whereas the second best is in italic. From these results, we have some observations as follows. (1) MINNs including mi-Net, MI-Net, and the variants of MI-Net obtain competi-

Table 1
Detailed characteristics of the datasets. "# positive" ("#negative") presents the number of positive(negative) bags used in each round. For Text category dataset, because it contains 20 sub-datasets, we present three of them in it.

Dataset	# attribute	# bag			# instance
		positive	negative	total	
MUSK1	166	47	45	92	476
MUSK2	166	39	63	102	6598
Elephant	230	100	100	200	1391
Fox	230	100	100	200	1320
Tiger	230	100	100	200	1220
alt.atheism	200	50	50	100	5443
comp.graphics	200	49	51	100	3094
comp.os.ms-windows.misc	200	50	50	100	5175

Table 2
Comparison results (*mean* ± *std*) of different methods for bag classification on MUSK1 and MUSK2 (task: drug activity prediction).

Dataset	MUSK1	MUSK2
mi-SVM [10]	0.874	0.836
MI-SVM [10]	0.779	0.843
MI-Kernel [34]	0.880	0.893
EM-DD [35]	0.849 ± 0.098	0.869 ± 0.108
mi-Graph [31]	0.889 ± 0.073	**0.903 ± 0.086**
miVLAD [33]	0.871 ± 0.097	0.872 ± 0.095
miFV [33]	**0.909 ± 0.089**	0.884 ± 0.094
mi-Net	0.889 ± 0.088	0.858 ± 0.110
MI-Net	0.887 ± 0.091	0.859 ± 0.102
MI-Net with DS	0.894 ± 0.093	0.874 ± 0.097
MI-Net with RC	0.898 ± 0.097	0.873 ± 0.098

Table 3
Comparison results (*mean* ± *std*) of different methods for bag classification on Fox, Tiger, and Elephant (task: localized content-based image retrieval).

Dataset	Fox	Tiger	Elephant
mi-SVM [10]	0.582	0.784	0.822
MI-SVM [10]	0.578	0.840	0.843
MI-Kernel [34]	0.603	0.842	0.843
EM-DD [35]	0.609 ± 0.101	0.730 ± 0.096	0.771 ± 0.097
mi-Graph [31]	0.620 ± 0.098	**0.860 ± 0.083**	0.869 ± 0.078
miVLAD [33]	0.620 ± 0.098	0.811 ± 0.087	0.850 ± 0.080
miFV [33]	0.621 ± 0.109	0.813 ± 0.083	0.852 ± 0.081
mi-Net	0.613 ± 0.078	0.824 ± 0.076	0.858 ± 0.083
MI-Net	0.622 ± 0.084	0.830 ± 0.072	0.862 ± 0.077
MI-Net with DS	**0.630 ± 0.080**	0.845 ± 0.087	**0.872 ± 0.072**
MI-Net with RC	0.619 ± 0.104	0.836 ± 0.083	0.857 ± 0.089

tive results with the state-of-the-arts on the MUSK datasets. (2) In the task of image classification on the Fox, Tiger and Elephant datasets, the best one of MINNs, MI-Net with DS, wins two of three datasets and has a very similar average accuracy over the three datasets comparing to the previous state-of-the-art method, which is the mi-Graph method. Note that mi-Graph can make use of the contextual information between image patches (instances), but the studies MINNs do not model the contextual information. In addition, the three datasets are in small size. The "small data" is more suitable to use traditional shallow models to analyze rather than the proposed deep representation learning method. (3) In the task of text classification on the 20 Newsgroups datasets which are among the largest MIL datasets, MINNs outperform the other methods by a large margin. The results show the superiority of MINNs and also imply that MINNs are beneficial from larger size of the training data. (4) The proposed embedded-space network MI-Net is more competitive than the previous instance-space network mi-Net [8,9]. (5) In five MIL benchmarks (MUSK datasets and Animal datasets), MI-Net with DS achieves best results compared with other methods, which verifies that network with deep supervision is more robust to predict bag label. (6) Additionally, MI-Net with RC also obtains good results on these five benchmark datasets. In

Table 4
Comparison results (*mean* ± *std*) of different methods for bag classification on 20 Newsgroups (task: text categorization).

Dataset	MI-Kernel [34]	miGraph [31]	miFV [33]	mi-Net	MI-Net	MI-Net with DS	MI-Net with RC
alt.atheism	0.602 ± 0.039	0.655 ± 0.040	0.848 ± 0.119	0.758 ± 0.124	0.776 ± 0.101	**0.860 ± 0.134**	0.858 ± 0.099
comp.graphics	0.470 ± 0.033	0.778 ± 0.016	0.594 ± 0.140	**0.830 ± 0.145**	0.826 ± 0.134	0.822 ± 0.123	0.828 ± 0.118
comp.windows.misc	0.510 ± 0.052	0.631 ± 0.015	0.615 ± 0.172	0.658 ± 0.134	0.678 ± 0.101	0.716 ± 0.112	**0.720 ± 0.120**
comp.ibm.pc.hardware	0.469 ± 0.036	0.595 ± 0.027	0.665 ± 0.147	0.772 ± 0.134	0.778 ± 0.129	**0.792 ± 0.155**	0.784 ± 0.145
comp.sys.mac.hardware	0.445 ± 0.032	0.617 ± 0.048	0.660 ± 0.157	0.746 ± 0.127	0.792 ± 0.113	0.794 ± 0.138	**0.810 ± 0.133**
comp.window.x	0.508 ± 0.043	0.698 ± 0.021	0.768 ± 0.155	0.746 ± 0.145	0.786 ± 0.111	0.812 ± 0.135	**0.820 ± 0.098**
misc.forsale	0.518 ± 0.025	0.552 ± 0.027	0.565 ± 0.146	0.580 ± 0.135	0.652 ± 0.128	0.686 ± 0.119	**0.696 ± 0.119**
rec.autos	0.529 ± 0.033	0.720 ± 0.037	0.667 ± 0.166	0.746 ± 0.142	0.774 ± 0.121	0.776 ± 0.129	**0.792 ± 0.127**
rec.motorcycles	0.506 ± 0.035	0.640 ± 0.028	0.802 ± 0.144	0.716 ± 0.118	0.762 ± 0.116	**0.868 ± 0.119**	0.856 ± 0.133
rec.sport.baseball	0.517 ± 0.028	0.647 ± 0.031	0.779 ± 0.148	0.808 ± 0.139	0.856 ± 0.113	0.874 ± 0.122	**0.880 ± 0.117**
rec.sport.hockey	0.513 ± 0.034	0.850 ± 0.025	0.823 ± 0.137	0.860 ± 0.129	0.862 ± 0.085	0.912 ± 0.111	**0.918 ± 0.088**
sci.crypt	0.563 ± 0.036	0.696 ± 0.021	0.760 ± 0.146	0.608 ± 0.132	0.694 ± 0.142	**0.812 ± 0.166**	0.796 ± 0.140
sci.electronics	0.506 ± 0.020	0.871 ± 0.017	0.555 ± 0.156	0.932 ± 0.099	0.930 ± 0.088	0.926 ± 0.084	**0.938 ± 0.091**
sci.med	0.506 ± 0.019	0.621 ± 0.039	0.783 ± 0.125	0.792 ± 0.110	0.818 ± 0.106	**0.848 ± 0.110**	0.842 ± 0.108
sci.space	0.547 ± 0.025	0.757 ± 0.034	0.818 ± 0.131	0.694 ± 0.124	0.752 ± 0.112	**0.818 ± 0.137**	0.810 ± 0.136
soc.religion.christian	0.492 ± 0.034	0.590 ± 0.047	0.814 ± 0.138	0.718 ± 0.130	0.782 ± 0.113	0.820 ± 0.38	**0.822 ± 0.124**
talk.politics.guns	0.477 ± 0.038	0.585 ± 0.060	0.747 ± 0.150	0.596 ± 0.140	0.652 ± 0.117	**0.780 ± 0.119**	0.762 ± 0.101
talk.politics.mideast	0.559 ± 0.028	0.736 ± 0.026	0.793 ± 0.135	0.774 ± 0.103	0.794 ± 0.127	**0.842 ± 0.142**	0.824 ± 0.120
talk.politics.misc	0.515 ± 0.037	0.704 ± 0.036	0.697 ± 0.152	0.602 ± 0.108	0.654 ± 0.135	**0.776 ± 0.140**	0.736 ± 0.104
talk.religion.misc	0.554 ± 0.043	0.633 ± 0.035	0.739 ± 0.151	0.700 ± 0.171	0.700 ± 0.114	0.758 ± 0.123	**0.764 ± 0.120**
average	0.515	0.679	0.726	0.737	0.766	**0.815**	0.813

Table 5
The influence of different pooling methods for MI-Net with DS on five MIL benchmarks.

Pooling method	MUSK1	MUSK2	Fox	Tiger	Elephant
max	**0.894 ± 0.093**	**0.874 ± 0.097**	**0.630 ± 0.080**	0.826 ± 0.087	0.870 ± 0.072
mean	0.886 ± 0.105	0.858 ± 0.110	0.615 ± 0.078	**0.845 ± 0.087**	0.867 ± 0.083
LSE	0.891 ± 0.111	**0.874 ± 0.101**	0.625 ± 0.104	0.840 ± 0.083	**0.872 ± 0.072**

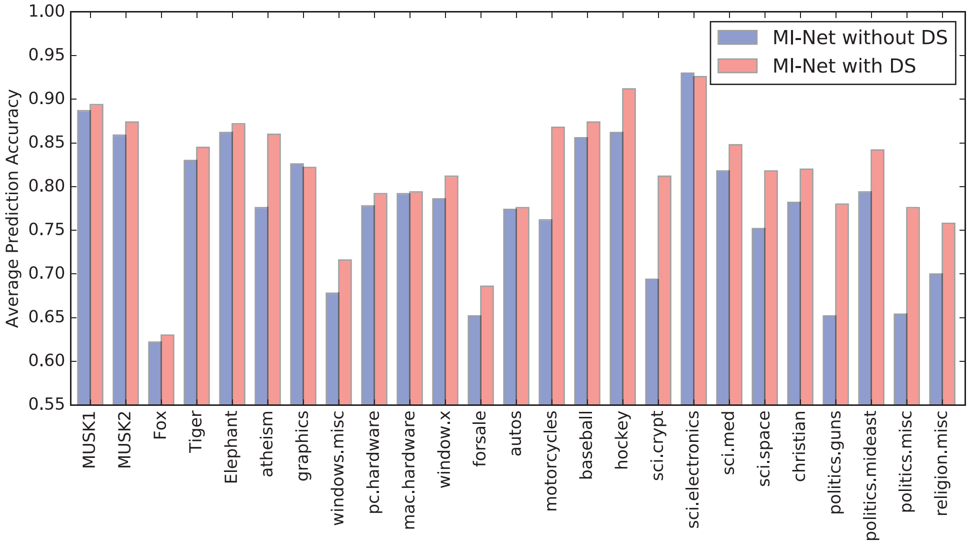

Fig. 5. The influence of deep supervision for MI-Net on five MIL benchmarks, where DS means deep supervision.

the 20 Newsgroups dataset, MI-Net with DS achieves superior performance, and results of MI-Net with RC are slightly worse than results of MI-Net. The average accuracy of all 20 datasets at evaluation indicates that MI-Net and its two variations outperform the other five competing algorithms, including MI-Kernel [34], mi-Graph [31], miFV [33], and mi-Nets. The observations in (5) and (6) suggest that we may choose different network tricks in different applications/problems, which is consistent with the common practices in the research of deep learning (different datasets may require different tricks).

5. Discussion

In this section, we study the influence of different pooling methods, deep supervision, residual connections, as well as the width and depth of MINNs.

5.1. The influence of different pooling methods

The pooling layer is a critical component to achieve multiple instance neural networks. There are three typical pooling methods, max pooling, mean pooling, and LSE pooling. On the MUSK datasets and Animal datasets, we give the results of MI-Nets with different pooling methods in Table 5. The results show that the three pooling methods obtain similar classification accuracy. More specifically, max pooling and LSE pooling methods work slightly better than mean pooling. However, in the 20 Newsgroups datasets, only max pooling method gives satisfying results, and mean pooling and LSE pooling do not coverage to normal bag classification accuracy. In summary, we recommend using max pooling method together with MINNs.

5.2. The influence of deep supervision

To illustrate the effectiveness of deep supervision, we compare MI-Net with deep supervision to MI-Net without deep supervision directly in Fig. 5 using histograms. Deep supervision helps to improve the bag classification accuracy in 23 datasets out of total tested 25 datasets. In the 2 datasets with no improvement, deep supervision hurts accuracies very slightly. We can conclude that deep supervision helps to learn better bag feature hierarchies in MI-Nets. In addition, deep supervision is computationally efficient in testing. It is a useful trick for training MI-Nets.

5.3. The influence of residual connections

We compare MI-Net with residual connections and MI-Net without residual connections in Fig. 6. The figure shows that residual connections improve bag classification accuracies in 23 datasets out of the total 25 datasets. In the Elephant and Tiger datasets, the residual connections method slightly hurts the performance. Like the deep supervision method, the residual connections method also has a positive impact on learning good bag representation in MI-Nets. Though deep supervision and residual connections look different, the theory behind them is the same. The common theory behind is to make the MIL supervision information easier to propagate to the early layers and help to learn better bag representation in early layers.

5.4. The influence of deeper and wider MINNs

As aforementioned, for mi-Net, MI-Net, and MI-Net with its variations, the number of layers and neurons for each layer are fixed when training and testing. In Tables 2, 3, and 4, both the proposed networks have four FC layers and there are 256, 128, 64, and 1 neurons for FC layers in MI-Net with DS respectively whereas there are 128, 128, 128, and 1 neurons in MI-Net with RC. It is very necessary to study deeper and wider MINNs, since in deep learning the deeper and wider neural network may get better performance. In this subsection, we report the results of the proposed MI-Net with DS and MI-Net with RC with different layer numbers and neuron number values on five MIL benchmarks, respectively.

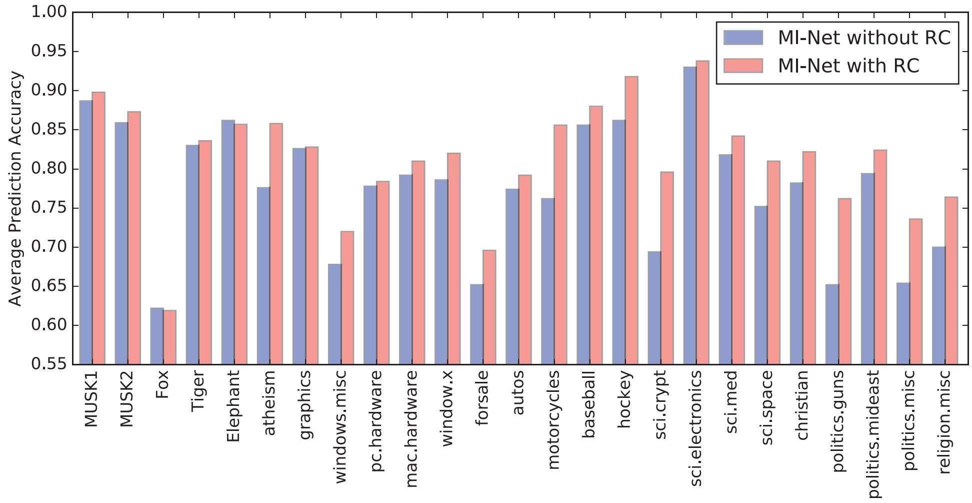

Fig. 6. The influence of residual connections for MI-Net on five MIL benchmarks, where RC means residual connections.

Table 6
The influence of depth and width for MI-Net with DS on five MIL benchmarks, where numbers in brackets show the number neurons for each FC layer.

Structure	MUSK1	MUSK2	Elephant	Fox	Tiger
(256, 256, 256, 1)	**0.898 ± 0.086**	0.853 ± 0.018	0.842 ± 0.071	0.629 ± 0.093	0.826 ± 0.080
(256, 256, 128, 1)	0.881 ± 0.084	**0.877 ± 0.116**	0.844 ± 0.700	0.602 ± 0.091	0.836 ± 0.091
(256, 128, 64, 1)	0.894 ± 0.093	0.874 ± 0.097	**0.872 ± 0.072**	**0.630 ± 0.080**	**0.845 ± 0.087**
(128, 128, 128, 1)	0.887 ± 0.094	0.871 ± 0.106	0.840 ± 0.069	0.616 ± 0.117	0.836 ± 0.087
(128, 128, 64, 1)	0.866 ± 0.103	0.859 ± 0.099	0.845 ± 0.071	0.602 ± 0.097	0.836 ± 0.076
(64, 64, 64, 1)	0.891 ± 0.097	0.857 ± 0.106	0.861 ± 0.072	0.592 ± 0.096	0.824 ± 0.081
(256, 256, 128, 128, 64, 1)	0.892 ± 0.086	0.873 ± 0.108	0.844 ± 0.069	0.627 ± 0.109	0.835 ± 0.083
(256, 256, 256, 256, 256, 1)	0.884 ± 0.099	0.853 ± 0.112	0.838 ± 0.078	0.609 ± 0.086	0.835 ± 0.092

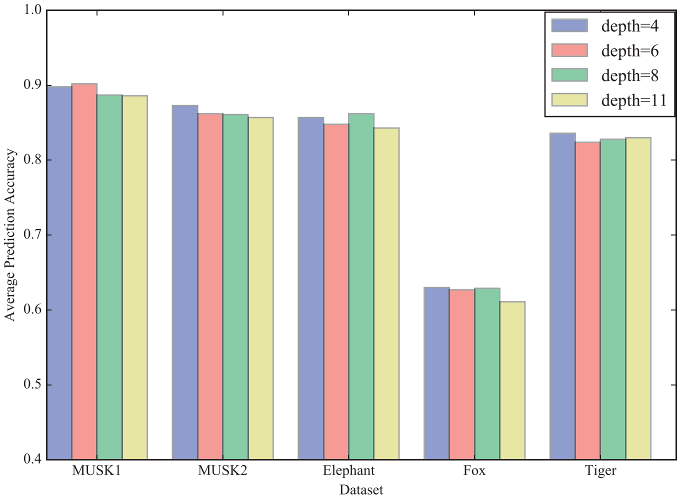

Fig. 7. Comparisons of depth for MI-Net with RC on five MIL benchmarks.

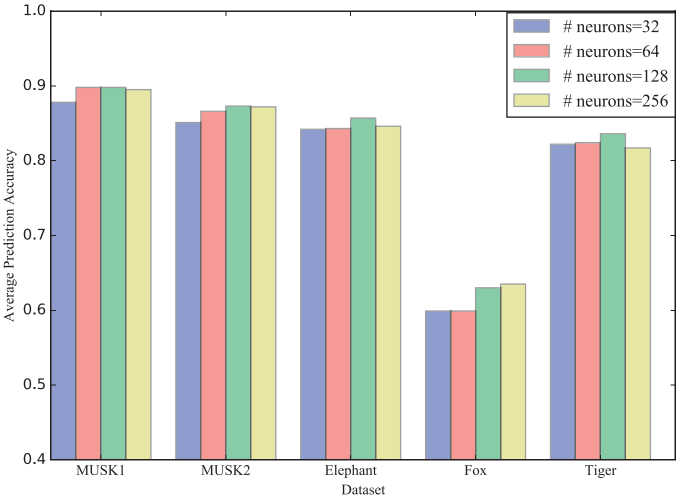

Fig. 8. Comparisons of width for MI-Net with RC on five MIL benchmarks.

The depth and width analysis results of MI-Net with DS on five MIL benchmarks are presented in Table 6. The neuron number of the last FC layer is fixed to 1 to output bag scores. As shown in Table 6, MI-Net with DS can achieve the best performance in most cases when the depth is 4, and each FC layer has 256, 128, 64, and 1 neurons respectively. Although results of the deeper and wider network are superior to the shallower and thinner one on some datasets, the advantage of the deeper and wider network is not obvious to boost the performance.

As referred in Section 3.5, the neuron numbers of FC layers should be of the same value to build residual connections except for the last FC layer. Fixing the width of MI-Net with RC, we only change the depth of the network. In Fig. 7, the results of different depths on five MIL benchmarks are similar. Then we fix the depth of MI-Net with RC to 4 during discussing the influence of width on MI-Net with RC. Fig. 8 illustrates that the wider network is not necessary to boost the performance. In addition, MI-Net with RC may get worse performance when it is too thin.

This observation is not consistent with the performance of deeper and wider neural networks to solve other problems. However, regarding the size of MIL datasets is much smaller than the modern deep learning datasets and the features are fixed and hand-crafted, it is reasonable the accuracies get saturated when MINNs are in the depth of 4 and have 128 neurons per layer.

6. Conclusion

In this study, we revisit the problem of end-to-end learning of MINNs and propose a series of novel MINNs with the state-of-the-art performance. Different from the existing MINNs, our method focuses on bag-level representation learning instead of instance-level label estimating. Experiments show that our bag-level networks show superior results on several MIL benchmarks compared with the instance-level networks. Moreover, we integrate the most popular deep learning tricks (deep supervision and residual connections) into our networks, which can boost the performance further. Moreover, our method only takes about 0.0003 s for testing (forward) and 0.0008 s for training (backward) per bag, which is very efficient.

Acknowledgements

We thank the constructive suggestions from the anonymous reviewers. This work was supported by Young Elite Sponsorship Program by CAST (No. YESS 20150077), CCF-Tencent Open Research Fund and National Natural Science Foundation of China (NSFC) (nos. 61503145, 61573160, and 61572207).

References

[1] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1) (1997) 31–71.
[2] J. Amores, Multiple instance classification: review, taxonomy and comparative study, Artif. Intell. 201 (2013) 81–105.
[3] G. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[4] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[5] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012, pp. 1097–1105.
[6] R.J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Comput. 1 (2) (1989) 270–280.
[7] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[8] J. Ramon, L. De Raedt, Multi instance neural networks, in: Proceedings of the ICML-2000 Workshop on Attribute-Value and Relational Learning, 2000, pp. 53–60.
[9] Z.-H. Zhou, M.-L. Zhang, Neural networks for multi-instance learning, in: Proceedings of the International Conference on Intelligent Information Technology, Beijing, China, 2002, pp. 455–459.
[10] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: NIPS, 2002, pp. 561–568.
[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, JMLR 15 (1) (2014) 1929–1958.
[12] V. Nair, G. Hinton, Rectified linear units improve restricted Boltzmann machines, in: ICML, 2010, pp. 807–814.
[13] C.Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: AISTATS, 2015, pp. 562–570.
[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv:1512.03385 (2015).
[15] M.-L. Zhang, Z.-H. Zhou, Improve multi-instance neural networks through feature selection, Neural Process. Lett. 19 (1) (2004) 1–10.
[16] M. Zhang, Z. Zhou, Ensembles of multi-instance neural networks, in: International Conference on Intelligent Information Processing, Springer, 2004, pp. 471–474.
[17] J. Wu, Y. Yu, C. Huang, K. Yu, Deep multiple instance learning for image classification and auto-annotation, in: CVPR, 2015, pp. 3460–3469.
[18] P.O. Pinheiro, R. Collobert, From image-level to pixel-level labeling with convolutional networks, in: CVPR, 2015, pp. 1713–1721.
[19] X. Wang, Z. Zhu, C. Yao, X. Bai, Relaxed multiple-instance SVM with application to object discovery, in: ICCV, 2015, pp. 1224–1232.
[20] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009, pp. 248–255.
[21] C. Xu, D. Tao, C. Xu, Multi-view learning with incomplete views, IEEE Trans. Image Process. 24 (12) (2015) 5812–5825.
[22] Y. Wang, C. Xu, S. You, D. Tao, C. Xu, Cnnpack: Packing convolutional neural networks in the frequency domain, NIPS, 2016.
[23] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. (2016).
[24] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, iprivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning, IEEE Trans. Inf. Forensics Secur. (2016).
[25] W. Ren, K. Huang, D. Tao, T. Tan, Weakly supervised large scale object localization with multiple instance learning and bag splitting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 405–416.
[26] M. Qiao, L. Liu, J. Yu, C. Xu, D. Tao, Diversified dictionaries for multi-instance learning, Pattern Recognit. 64 (2017) 407–416.
[27] H. Zhao, J. Cheng, J. Jiang, D. Tao, Multiple instance learning via distance metric optimization, in: Image Processing (ICIP), 2013 20th IEEE International Conference on, IEEE, 2013, pp. 2617–2621.
[28] J. Sánchez, F. Perronnin, T. Mensink, J.J. Verbeek, Image classification with the fisher vector: theory and practice, IJCV 105 (3) (2013) 222–245.
[29] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks., in: Aistats, 15, 2011, p. 275.
[30] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge university press, 2004.
[31] Z.H. Zhou, Y.Y. Sun, Y.F. Li, Multi-instance learning by treating instances as non-iid samples, in: ICML, 2009, pp. 1249–1256.
[32] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: AISTATS, 2010, pp. 249–256.
[33] X.S. Wei, J. Wu, Z.H. Zhou, Scalable algorithms for multi-instance learning, IEEE Trans. Neural Netw. Learn. Syst. PP (99) (2016) 1–13.
[34] T. Gärtner, P.A. Flach, A. Kowalczyk, A.J. Smola, Multi-instance kernels, in: ICML, 2, 2002, pp. 179–186.
[35] Q. Zhang, S.A. Goldman, EM-DD: An improved multiple-instance learning technique, in: NIPS, 2001, pp. 1073–1080.
[36] F. Chollet, Keras, 2015, (https://github.com/fchollet/keras).

Xinggang Wang is an assistant professor of School of Electronics Information and Communications of Huazhong University of Science and Technology. He received his Bachelors degree in communication and information system and Ph.D. degree in computer vision both from Huazhong University of Science and Technology. From May 2010 to July 2011, he was with the Department of Computer and Information Science, Temple University, Philadelphia, PA., as a visiting scholar. From February 2013 to September 2013, he was with the University of California, Los Angeles, as a visiting graduate researcher. He is a reviewer of IEEE Transaction on Cybernetics, Pattern Recognition, Computer Vision and Image Understanding, Neurocomputing, CVPR, ICCV and ECCV etc. His research interests include computer vision and machine learning.

Yongluan Yan is a master student in the School of Electronics Information and Communications, Huazhong University of Science and Technology (HUST). She received her B.S. degree from HUST in 2016. Her research interests include Computer Vision and Machine Learning. In particular, she focuses on multiple instance learning.

Peng Tang is a Ph.D. student in the School of Electronics Information and Communications, Huazhong University of Science and Technology (HUST). He received his B.S. degree from HUST in 2014. He is a reviewer of Neurocomputing. His research interests include Computer Vision and Machine Learning. In particular, he focuses on mid-level representation for image understanding.

Xiang Bai received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications, HUST. He is also the Vice-director of the National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition and intelligent systems.

Wenyu Liu received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1986 and the M.S. and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is currently a Professor and Associate Dean of the Department of Electronics and Information Engineering, HUST. His current research interests include multimedia information processing and computer vision.