Revisiting Multiple Instance Neural Networks

Xinggang Wang, Yongluan Yan, Peng Tang, *Student Member, IEEE,* Xiang Bai, *Senior Member, IEEE,* and Wenyu Liu, *Senior Member, IEEE*

***Abstract*—Recently neural networks and multiple instance learning are both attractive topics in Artificial Intelligence related research fields. Deep neural networks have achieved great success in supervised learning problems, and multiple instance learning as a typical weakly-supervised learning method is effective for many applications in computer vision, biometrics, nature language processing, etc. In this paper, we revisit the problem of solving multiple instance learning problems using neural networks. Neural networks are appealing for solving multiple instance learning problem. The multiple instance neural networks perform multiple instance learning in an end-to-end way, which take a bag with various number of instances as input and directly output bag label. All of the parameters in a multiple instance network are able to be optimized via back-propagation. We propose a new multiple instance neural network to learn bag representations, which is different from the existing multiple instance neural networks that focus on estimating instance label. In addition, recent tricks developed in deep learning have been studied in multiple instance networks, we find *deep supervision* is effective for boosting bag classification accuracy. In the experiments, the proposed multiple instance networks achieve state-of-the-art or competitive performance on several MIL benchmarks. Moreover, it is extremely fast for both testing and training, e.g., it takes only 0.0003 second to predict a bag and a few seconds to train on a MIL datasets on a moderate CPU.**

***Index Terms*—Multiple instance learning, neural networks, end-to-end learning.**

I. Introduction

Multiple instance learning (MIL) was originally proposed for drug activity prediction [1]. Now it has been widely applied in many domains and becomes 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 effective to process and understand MI data.

MIL is a kind of weakly-supervised learning (WSL). Each sample is in a 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 while a bag is negative if it is only constituted of negative instances. Thus, the crux of MIL is to deal with the ambiguity of instances labels, especially in positive bags which 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 can be divided into three folds: instance-space paradigm, bag-space paradigm and embedded-space paradigm. Instance-space paradigm learns 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, bag-to-bag distance/similarity is calculated; then the nearest neighbor or Bayesian classifier is able to do bag classification. Embedded-space paradigm embeds a bag into a vocabulary-based feature space to obtain a compact representation for the bag, e.g., 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 and classification; deep Convolutional Neural Networks (CNN) [4], [5] take 2D image as input and have dominated image recognition; 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 sequential prediction. Usually, training these deep networks requires a large number of fully labeled data, i.e., each instance requires a label. However, in MIL, only bag labels can be got. Meanwhile, MI data have a more complex structure which is a set of instances. The numbers of instances are different for different bags. These problems make it hard to deal with MIL problem by conventional neural networks.

Before the raising of deep learning, there were some research works trying to solve the MIL problem using neural networks. 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 can be trained using back-propagation. Zhang and Zhou [9] also proposed a multiple instance network which calculates bag probability by directly taking the max of instance probabilities.

A MINN takes a various number of instances as input. For each instance, its representation is gradually learned layer by layer guided by multiple instance supervision. To inject multiple instance supervision, there are two different network architectures. Following the naming style in a classical MIL work [10], we name the two networks as mi-Net and MI-Net, which aim at the instance-space paradigm and embedded-space paradigm [2] respectively. In mi-Net, there are instance classifiers in the each layer. We are able to obtain instance labels for both training and testing bags, which is an appealing property in some applications. While in MI-Net, there is no instance classifier. It directly builds a fixed-length vector as the

X. Wang, Y. Yan, P. Tang, X. Bai, W. Liu were with the School of Electronics Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074 China e-mail: (xbai@hust.edu.cn).

bag representation and then learns bag classifier. Compared with mi-Net, MI-Net can obtain better bag classification accuracy. The previous works are in the category of mi-Net. We newly propose MI-Net in this paper.

A key component in MINN is MIL Pooling Layer (MPL), which aggregates either instance probability distribution vectors or instance feature vectors into a bag feature vector. It bridges MI data with conventional neural networks. Since 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 part. Besides of 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 supervision of bag labels. In the last of 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 progress of deep learning in MINN, such as dropout [11], ReLU [12], deeply supervised nets (DSN) [13] and Residual Connections [14]. We find DSN is the most effective one. This is due to DSN is able to better use hierarchical features in networks. Also, residual connections do a great job in networks.

To summarize, we revisit the problem of solving multiple instance learning using neural networks. This branch of MIL algorithm is ignored by current MIL research community. But it is highly effective and efficient. Different from most MIL algorithms, it is able to learn instance features in an end-to-end manner. This paper focuses on neural networks for end-to-end MIL with comprehensive studies on MIL benchmarks. The main contributions of this paper include two extremely fast and scalable methods for MIL, i.e., mi-Net and MI-Net, and introducing deep supervision and residual connections for MIL.

We organize the rest of this paper as follow. Section II briefly reviews previous works on MIL. In Section III, we propose end-to-end MIL networks. Our experimental results are presented on several MIL benchmarks in Section IV. Some discussions of experimental setups are presented in Section V. Finally, in Section VI we conclude the paper with some future works.

II. Related Work

Previous works on solving MIL using neural networks include [8], [9], [15], [16]. [8] introduced to use a log-sum-exp as the convex max to calculate bag probabilities from instance probabilities. [9] changed to a different loss function and directly applied max function. [15] improved multiple instance neural networks by feature selection using Diverse Density and PCA. [16] showed that ensemble methods could be integrated with multiple instance neural networks. Then, solving MIL using neural networks has been ignored in machine learning research. This paper revisits this problem, proposes new network structures, and investigates recent neural network tricks.

Multiple Instance Learning (MIL) has received a lot of attentions since it helps to solve a range of real applications. Till now, lots of MIL methods have been proposed to either develop effective MIL solvers or apply MIL to solve application problems. A comprehensive survey of MIL algorithms and applications can be found in [2]. Here, we focus on give a brief review of the most recent MIL algorithm, 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, i.e., solving MIL using Fisher Vector (FV) coding [17], which is called miFV [18]. miFV transforms instance feature into high-dimensional space using a pre-trained Gaussian mixture model and FV coding. The proposed MI-Net learns instance feature using deep multiple instance supervision. And MI-Net achieves better bag classification accuracy and is much faster than miFV.

The idea of using neural networks for solving MIL problem has been studied in some computer vision studies, such as [18], [19]. Wu et. al [18] proposed deep MIL which uses max pooling to find positive instances/patches for image classification and annotation. Pinheheiro et. al [19] used log-sum-exp pooling in deep CNN for weakly supervised semantic segmentation. The proposed mi-Net follows the path of these two works; different from them, mi-Net utilizes deep supervision, and focuses on more general MIL problems. Besides of integrating MIL into deep neural networks, Wang et. al proposed a method to combine MIL with support vector machine using a relaxed MIL constraint [20] and applied this for object discovery. However, they pay more attention on vision applications (e.g., image classification, image annotation, and semantic segmentaion, etc.), which are based on convolutional image features. Meanwhile, they always finetune neural network models pre-trained on other much larger datasets like ImageNet [21]. Moreover, they also only focus more on instance-space MIL.Compared with theirs, we focus on appling MIN structure for more general MIL problems. Notice that for general MIL problems, there are no available large datasets for pre-training like computer vision, which makes it more difficult to train MINN efficiently. We have shown many tricks to train our networks from scratch on MIL benchmarks with limited training data, and achieved many inspiring results. Meanwhile, we have investigated both mi-Net and MI-Net, and experiments have shown that the MI-Net outperforms mi-Net in more cases.

III. Multiple Instance Neural Networks

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

A. Notations

Here we first review the definition of MIL. Given a set of bags $X = \{X_1, X_2, ..., X_N\}$ and instance features of ith bag $X_i = \{x_{i1}, x_{i2}, ..., x_{im_i}\}, x_{ij} \in \mathbb{R}^{d\times 1}$, 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, i.e., 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, i.e., if $Y_i = 1$, then $\sum_{j=1}^{m_i} y_{ij} \geq 1$.

Since instance label is not given in training phase, solving the MIL problem is challenging. In MINNs, there are two strategies: the first one is to infer instance label in the network, i.e., placing instance probabilities of being positive as a hidden layer in the network; the second one is to use learn bag representation in the network and directly carry out bag classification without calculating instance probability. The first strategy has been studied in [8], [9], [18]. The second strategy is newly proposed in this paper. 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 consists of 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), or rectified linear units (ReLU) [22]. We denote the output of the ℓ^{th} layer of an instance x_{ij} as x_{ij}^ℓ.

B. mi-Net: Instance-Space MIL Algorithm

At first, we review traditional multiple instance neural networks [8], [9], [18], 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 activation function (in this paper we use four fc layers and ReLU activation). Thus, we get the instance feature denoted as x_{ij}^{L-2} in the $(L-2)th$ layer and instance probability denoted as p_{ij}^{L-1}. p_{ij}^{L-1} is a scalar in the range of $[0, 1]$. In the last layer, there is a MIL pooling layer (described in Section III-F) which takes instance probabilities as input and outputs bag probability, denoted as $P^L(X_i)$.

These first $L-1$ fc layers can learn some more semantic instance features compared with original x_iij (higher layer corresponding to higher semantic features). After learning these instance features, a fc layer which only has one neuron with sigmoid activation, is used to predict the positiveness of instances.

But 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 the positiveness of bag.

The MIL pooling method satisfies the MIL constraints: If a bag is positive, there should have at least one instance with large positiveness. Otherwise, all instances in the bag should have low positiveness. Since the pooling layer is integrated into the neural network, the pooling function should be differentiable. There typical MIL pooling will be introduced in Section III-F.

In summary, the mi-Net can be formulated as:

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

C. MI-Net: A new Embedded-Space MIL Algorithm

We propose a series of new multiple instance neural networks which 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 the survey [2]. Following the naming style in [10], we name this networks as MI-Net.

In Figure 2, we show a plain MI-Net with three fully connected layer 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. At last, 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:

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

D. MI-Net with Deep Supervision

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

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

where the index k in $X_i^{\ell,k}$ means we learn multiple bag features from all different levels of instance features by MIL pooling. MI-Net with deep supervision is able to utilize multiple hierarchies to get better bag classification accuracy. It can be interpreted from two folds: (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 paper, we set the weights of different levels equally.

E. MI-Net with Residual Connections

Recently, deep residual learning was proposed in [14] and showed the impressive improvement in image recognition by utilizing very deep neural networks. We study the residual

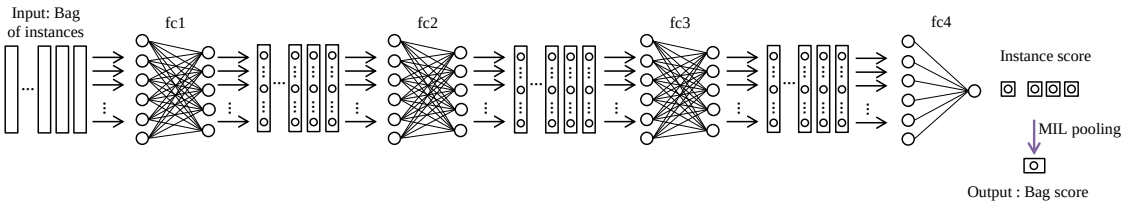

Fig. 1. A mi-Net with four fully connected layers. The number of output of fully connected layers are 256, 128, 64 and 1 respectively. The last layer is a MIL pooling layer with instance probabilites as input and bag probability as output.

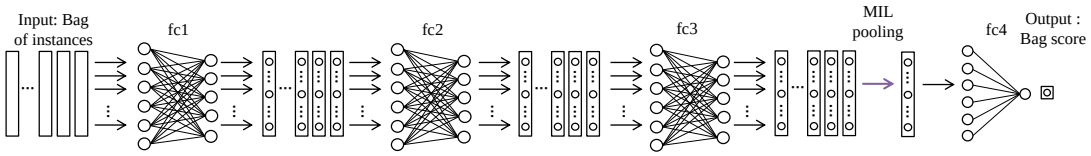

Fig. 2. The proposed MI-Net with three fully connected layers and one MIL pooling layer. The number of output of fully connected layers are 256, 128 and 64 respectively.

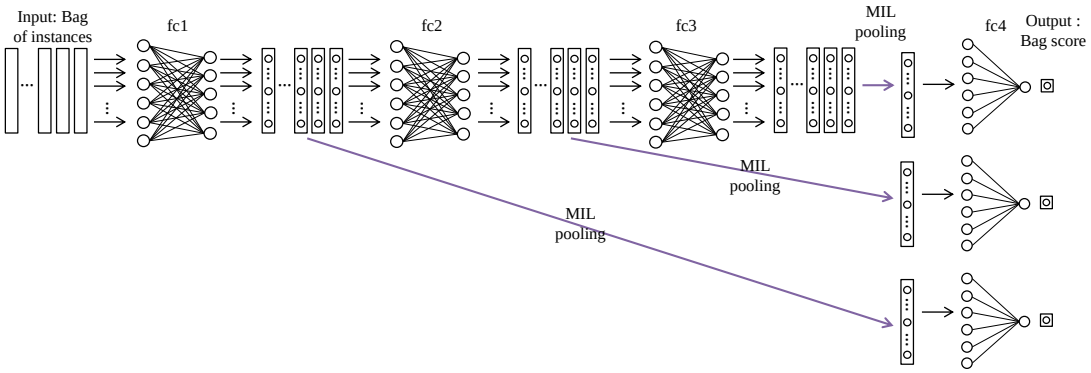

Fig. 3. The proposed MI-Net with deep supervision. There are three fully connected layers for learning instance features which are in the size of 256, 128 and 64 respectively. And there are three MIL pooling layers for generating bag feature and the bag features are connected to the bag label via a fully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

connections in MI-Net as shown in Figure 4. MI-Net with residual connections are formulated as:

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

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.

F. MIL Pooling Methods

As referred before, we use a MIL pooling layer to get patch scores or patch representations. In this paper, we use three popular 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 III-B. Actually the LSE [23] is a smooth version and convex approximation of the max function. The hyperparameter r controls how 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.

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

G. Training Loss

For both mi-Net and MI-Net, we can get the bag scores. Here we will 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

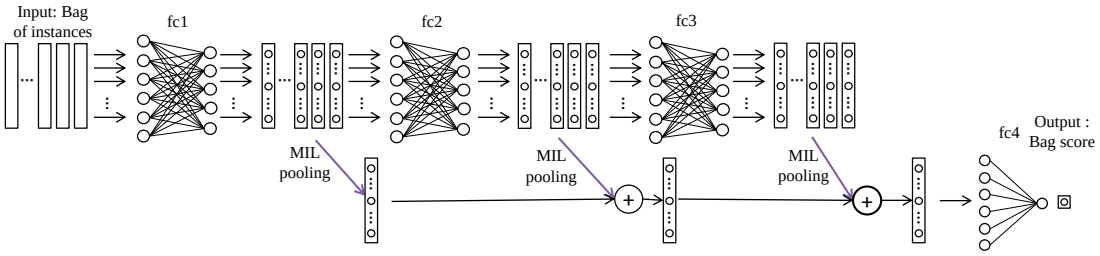

Fig. 4. The proposed MI-Net with residual connections. The first fully connected layer produces a bag feature vector. The latter fully connected layers learn the residuals of bag representation. The size of fully connected layers are all 128.

bag score of i bag. This loss is added to each bag scores level for deep supervision.

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

As all parts of our multiple instance network are differentiable, we can train these networks by standard back-propagation with Stochastic Gradient Descent (SGD).

IV. Experiments

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

A. Datasets

We test these methods on three widely-used MILbenchmarks in different applications, including drug activation prediction, automatic image annotation and text categorization. For evaluation, we run five times 10-fold cross validation and report the average results.

a) Drug Activation 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, while 6598 instances are included in MUSK2 which is divided into 39 positive bags and 63 negative bags.

b) Automatic Image Annotation: The Elephant, Fox and Tiger datasets [10], are all composed of 100 positive bags from the target class animal images and 100 negative bags randomly chosen from other class 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 are 230-dimension features that describe the color, texture, and shape in regions of an image.

c) Text Categorization: Besides the above datasets, the text categorization is another widely used application of MIL problems. Here, we take twenty datasets derived from the 20 Newsgroups corpus [26]. In each category, 100 bags are included among which half bags are positive and the rest of bags are negative. Each positive bag contains 3% posts from the target class and the rest from other categories, while the instances of negative bags are all randomly drawn from other categories. In addition, each instance is represented by the top 200 TF-IDF features.

Detailed characteristics of these datasets are summarized in Table I.

B. Experimental Setup

These neural networks contain four fully connected (fc) layers and first three fc layers are followed by a dropout layer (0.5 dropout ratio). As referred in Section III, we present the performance of the proposed multiple instance learning approaches: (1) 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) MI-Net: Input instances are aggregated into bag representation by first three fc layers and MIL pooling layer, and then use the last fc layer to predict bag probability. (3) MI-Net with Deep Supervision (MI-Net with DS): Different from MI-Net, each middle fc layer is followed by a MIL pooling layer and fc layer to compute bag scores. The loss function of MI-Net with DS sums up all middle entropy losses to do backpropagation with SGD for training, and the average of each bag score is used for testing. (4) MI-Net with Residual Connections (MI-Net with RC): Residual connections are built between each middle bag representation, and followed by a fc layer to obtain bag score.

As for the numbers of neurons in fc layers, there are 256, 128, 64, 1 in mi-Net, MI-Net and MI-Net with DS while 128, 128, 128, 1 in MI-Net with RC. Weights of fc layers are all initialized using a glorot-uniform distribution [28]. Biases are all initialized to be 0. For different datasets, the learning rate, weight decay and momentum are set suitable values that you can find in the configuration file of our code. All networks are trained with SGD, and one bag is inputted as a batch for training and testing. Moreover, about training and testing time, e.g., it takes only 0.0003 second to predict a bag and 0.0008 second to train on MUSK1 dataset on a moderate CPU. Our code is written in Python, based on Keras [29], and all of our

TABLE I
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	min	max	total
MUSK1	166	47	45	92	2	40	476
MUSK2	166	39	63	102	1	1044	6598
Elephant	230	100	100	200	3	13	1391
Fox	230	100	100	200	2	13	1320
Tiger	230	100	100	200	1	13	1220
Text(Zhou) alt.atheism	200	50	50	100	22	76	5443
Text(Zhou) comp.graphics	200	49	51	100	12	58	3094
Text(Zhou) comp.os.ms-windows.misc	200	50	50	100	25	82	5175

TABLE II
AVERAGE PREDICTION ACCURACY (IN %) OF DIFFERENT METHODS FOR BAG CLASSIFICATION ON FIVE MIL BENCHMARKS.

Name	MUSK1	MUSK2	Elephant	Fox	Tiger
mi-SVM [10]	0.780	0.702	0.822	0.582	0.784
MI-SVM [10]	0.779	0.843	0.814	0.578	0.840
EM-DD [24]	0.849	0.869	0.771	0.609	0.730
MI-Kernel [25]	0.880	0.893	0.843	0.603	0.842
MI-Graph [26]	0.900	0.900	0.851	0.612	0.819
mi-Graph [26]	0.889	**0.903**	0.868	0.616	**0.860**
miVLAD [27]	0.871	0.872	0.850	0.620	0.811
miFV [27]	**0.909**	0.884	0.852	0.621	0.813
mi-Net	0.889	0.858	0.858	0.613	0.824
MI-Net	0.887	0.859	0.862	0.622	0.830
MI-Net with DS	0.894	0.874	**0.872**	**0.630**	0.845
MI-Net with RC	0.898	0.873	0.857	0.619	0.836

experiments are running on a PC with Inter(R) i7-4790K CPU (4.00GHZ) and 32GB RAM. The code for reproducing results will be available upon acceptance.

C. Experimental Results

Experimental results are shown in Table II and Table III. The best performance of each dataset is bolded. Notice that using different pooling methods for these networks will produce different results for each dataset. Here, we choose the best one as the final result (for text categorization, the max pooling achieves the best performance consistently). And we will discuss the influence of pooling methods later. Particularly, it achieves state-of-the-art performance on Elephant, Fox, and text categorization, and nearly best accuracies on other datasets. These results demonstrate the effectiveness of these multiple instance networks. From these results, we can observe that these networks achieve highly competitive results.

We can easily find that the embedded-space network MI-Net seems more competitive than the instance-space network mi-Net, which is consistent with other MIL algorithms. In five benchmark datasets, MI-Net with DS achieves almost all best results than other methods, which verifies the network with deep supervision will be more robust to predict bag label. Additionally, MI-Net with RC also gets good results on these five benchmark datasets. In text categorization datasets, MI-Net with DS achieves the superior performance and results of MI-Net with RC is slightly worser than results of MI-Net. The average accuracy of all 20 datasets as evaluation indicates that MI-Net and its two variations outperform other five competing algorithms, including MI-Kernel [25], miGraph [26], miFV [27] and mi-Nets.

V. DISCUSSION

In this section, we discuss the influence of different pooling methods, deep supervision, residual connections on the networks. The width and depth of networks which may have impact on the performance is also considered in discussion.

A. The Influence of Different Pooling Methods

There are three pooling methods applied to these networks, including max pooling, mean pooling and LSE pooling. As referred in Section III, in embedded-space, instance features of the same bag are aggregated into the bag representation through pooling methods; in instance-space, instance scores of the same bag are aggregated into bag scores. We test the influence of different pooling methods on MI-Net with DS. From Table IV, we can observe that max pooling is preferable compared with other methods.

B. The Influence of Deep Supervision

To illustrate the effectiveness of deep supervision, we compare our MI-Net with deep supervision to the network without deep supervision, which only do MIL pooling and bag score prediction on the third fc layer. The effectiveness of deep supervision is validated on five MIL benchmark datasets, as shown in Table V. From the results, we can observe that the performance is boosted by deep supervision for all datasets

TABLE III
AVERAGE PREDICTION ACCURACY (IN %) FOR BAG CLASSIFICATION ON TEXT CATEGORIZATION.

Dataset	MI-Kernel [25]	miGraph [26]	miFV [27]	mi-Net	MI-Net	MI-Net with DS	MI-Net with RC
alt.atheism	0.602	0.655	0.848	0.758	0.776	**0.860**	0.858
comp.graphics	0.470	0.778	0.594	**0.830**	0.826	0.822	0.828
comp.windows.misc	0.510	0.631	0.615	0.658	0.678	0.716	**0.720**
comp.ibm.pc.hardware	0.469	0.595	0.665	0.772	0.778	**0.792**	0.784
comp.sys.mac.hardware	0.445	0.617	0.660	0.746	0.792	0.794	**0.810**
comp.window.x	0.508	0.698	0.768	0.746	0.786	0.812	**0.820**
misc.forsale	0.518	0.552	0.565	0.580	0.652	0.686	**0.696**
rec.autos	0.529	0.720	0.667	0.746	0.774	0.776	**0.792**
rec.motorcycles	0.506	0.640	0.802	0.716	0.762	**0.868**	0.856
rec.sport.baseball	0.517	0.647	0.779	0.808	0.856	0.874	**0.880**
rec.sport.hockey	0.513	0.850	0.823	0.860	0.862	0.912	**0.918**
sci.crypt	0.563	0.696	0.760	0.608	0.694	**0.812**	0.796
sci.electronics	0.506	0.871	0.555	0.932	0.930	0.926	**0.938**
sci.med	0.506	0.621	0.783	0.792	0.818	**0.848**	0.842
sci.space	0.547	0.757	**0.818**	0.694	0.752	**0.818**	0.810
soc.religion.christian	0.492	0.590	0.814	0.718	0.782	0.820	**0.822**
talk.politics.guns	0.477	0.585	0.747	0.596	0.652	**0.780**	0.762
talk.politics.mideast	0.559	0.736	0.793	0.774	0.794	**0.842**	0.824
talk.politics.misc	0.515	0.704	0.697	0.602	0.654	**0.776**	0.736
talk.religion.misc	0.554	0.633	0.739	0.700	0.700	0.758	**0.764**
average	0.515	0.679	0.726	0.737	0.766	**0.815**	0.813

TABLE IV
THE INFLUENCE OF DIFFERENT POOLING METHODS FOR MI-NET WITH DS ON FIVE MIL BENCHMARKS.

Pooling Method	MUSK1	MUSK2	Elephant	Fox	Tiger
max	**0.894**	**0.874**	**0.870**	**0.630**	0.826
mean	0.886	0.858	0.867	0.615	**0.845**
LSE	0.891	0.874	0.872	0.625	0.840

TABLE V
THE INFLUENCE OF DEEP SUPERVISION FOR MI-NET ON FIVE MIL BENCHMARKS, WHERE DS MEANS DEEP SUPERVISION.

Method	MUSK1	MUSK2	Elephant	Fox	Tiger
MI-Net with DS	**0.894**	**0.874**	**0.870**	**0.630**	**0.845**
MI-Net without DS	0.887	0.859	0.862	0.622	0.830

TABLE VI
THE INFLUENCE OF RESIDUAL CONNECTIONS FOR MI-NET ON FIVE MIL BENCHMARKS, WHERE RC MEANS RESIDUAL CONNECTIONS.

Method	MUSK1	MUSK2	Elephant	Fox	Tiger
MI-Net with RC	**0.898**	**0.873**	0.857	0.619	**0.836**
MI-Net without RC	0.887	0.859	**0.862**	**0.622**	0.830

and networks. Deep supervision is essential for learning good instance features in multiple instance networks.

C. The Influence of Residual Connections

In order to show the improvement of residual connections, MI-Net with Residual Connections which learns the bag representation residuals, is compared to MI-Net. As referred in Section VI, the influence of residual connections is proved on five MIL benchmark datasets. The results of MI-Net with Residual Connections are better than MI-Net without Residual Connections, except for Elephant and Tiger. Residual Connections may also have a positive impact on learning good bag representation in multiple instance networks.

D. The Influence of network depth and width

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 table II and table III, the proposed network both have four fc layers and there are 256, 128, 64, 1 neurons for fc layers in MI-Net with DS respectively while 128, 128, 128, 1 in MI-Net with RC. However, in deep learning, the deeper and wider neural network may get better performance. In this section, we will report the results of proposed MI-Net with DS and MI-Net with RC with different layer number and neuron number values on five MIL benchmarks, respectively.

The depth and width analysis results of MI-Net with DS on five MIL benchmarks are presented in Table VII. Note that,

TABLE VII
THE INFLUENCE OF DEPTH AND WIDTH FOR MI-NET WITH DS ON FIVE MIL BENCHMARKS, WHERE NUMBERS IN BRACKETS MEAN THE NUMBER NEURONS FOR EACH FC LAYER.

Structure	MUSK1	MUSK2	Elephant	Fox	Tiger
(256, 256, 256, 1)	**0.898**	0.853	0.842	0.629	0.826
(256, 256, 128, 1)	0.881	**0.877**	0.844	0.602	0.836
(256, 128, 64, 1)	0.894	0.874	**0.872**	**0.630**	**0.845**
(128, 128, 128, 1)	0.887	0.871	0.840	0.616	0.836
(128, 128, 64, 1)	0.866	0.859	0.845	0.602	0.836
(64, 64, 64, 1)	0.891	0.857	0.861	0.592	0.824
(256, 256, 128, 128, 64, 1)	0.892	0.873	0.844	0.627	0.835
(256, 256, 256, 256, 256, 1)	0.884	0.853	0.838	0.609	0.835

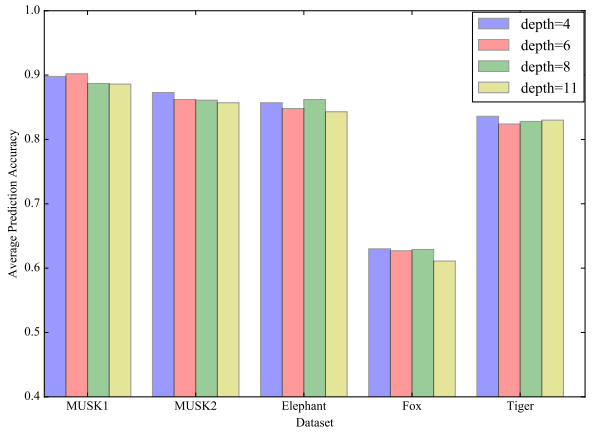

Fig. 5. Comparisons of depth for MI-Net with RC on five MIL benchmarks.

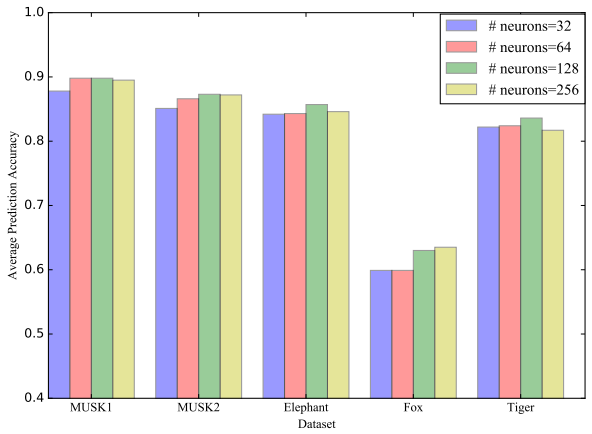

Fig. 6. Comparisons of width for MI-Net with RC on five MIL benchmarks.

the neuron number of last fc layer is fixed to 1 in order to output bag scores. As shown in Table VII, 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, 1 neurons respectively. Although results of the deeper and wider network is 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 III-E, the neuron number of fc layers should be 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 Figure 5, results on five MIL benchmarks are similar with the network deeper. So the depth of MI-Net with RC is set to 4 during discussing the influence of width on MI-Net with RC. Figure 6 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. That may be related to limited training data and simple MIL pooling methods.

VI. CONCLUSION

In this work, we propose series of novel neural network frameworks for MIL. Different from previous MIL networks, 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 network. Moreover, we intergrate the most popular deep learning tricks (deep supervision and residual connections) into our networks, which can boost the performance further. What is more, our method only takes about 0.0003 second for testing (forward) and 0.0008 second for training (backward) per bag, which is very efficient. According to these inspiring results, we believe that deep learning can also solve the traditional MIL problem well. In the future, we would like to study how to develop more effective MIL pooling methods, and how to train deeper and wider networks for MIL with limited training data.

REFERENCES

[1] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, 1997.
[2] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81–105, 2013.
[3] G. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.
[6] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] J. Ramon and 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 and 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, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2002, pp. 561–568.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] V. Nair and 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, and Z. Tu, "Deeply-Supervised Nets," in *AISTATS*, 2015, pp. 562–570.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[15] M.-L. Zhang and Z.-H. Zhou, "Improve multi-instance neural networks through feature selection," *Neural Processing Letters*, vol. 19, no. 1, pp. 1–10, 2004.

[16] M. Zhang and Z. Zhou, "Ensembles of multi-instance neural networks," in *International Conference on Intelligent Information Processing*. Springer, 2004, pp. 471–474.

[17] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek, "Image classification with the Fisher Vector: Theory and practice," *IJCV*, vol. 105, no. 3, pp. 222–245, 2013.

[18] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *CVPR*, 2015, pp. 3460–3469.

[19] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *CVPR*, 2015, pp. 1713–1721.

[20] X. Wang, Z. Zhu, C. Yao, and X. Bai, "Relaxed multiple-instance SVM with application to object discovery," in *ICCV*, 2015, pp. 1224–1232.

[21] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[22] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." in *Aistats*, vol. 15, no. 106, 2011, p. 275.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *NIPS*, 2001, pp. 1073–1080.

[25] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *ICML*, vol. 2, 2002, pp. 179–186.

[26] Z. H. Zhou, Y. Y. Sun, and Y. F. Li, "Multi-instance learning by treating instances as non-iid samples," in *ICML*, 2009, pp. 1249–1256.

[27] X. S. Wei, J. Wu, and Z. H. Zhou, "Scalable algorithms for multi-instance learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2016.

[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.

[29] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.