MULTI-TASK DEEP NEURAL NETWORK FOR MULTI-LABEL LEARNING

Yan Huang, Wei Wang, Liang Wang, Tieniu Tan

Center for Research on Intelligent Perception and Computing (CRIPAC)
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100190, China
{yhuang, wangwei, wangliang, tnt}@nlpr.ia.ac.cn

ABSTRACT

This paper proposes a multi-task deep neural network (MT-DNN) architecture to handle the multi-label learning problem, in which each label learning is defined as a binary classification task, i.e., a positive class for "an instance owns this label" and a negative class for "an instance does not own this label". Multi-label learning is accordingly transformed to multiple binary-class classification tasks. Considering that a deep neural nets (DNN) architecture can learn good intermediate representations shared across tasks, we generalize one classification task of traditional DNN into multiple binary classification tasks through defining the output layer with a negative class node and a positive class node for each label. After a similar pretraining process to deep belief nets, we redefine the label assignment error of MT-DNN and perform the back-propagation algorithm to fine-tune the network. To evaluate the proposed model, we carry out image annotation experiments on two public image datasets, with 2000 images and 30,000 images respectively. The experiments demonstrate that the proposed model achieves the state-of-the-art performance.

Index Terms— Multi-Task Learning, Deep Neural Network, Multi-Label Learning, Image Annotation

1. INTRODUCTION

Learning good features is very important to computer vision and pattern recognition tasks. Taking object recognition for an example, an object in human visual system can be represented by low-level features and high-level features, e.g., Gabor-like edges, object parts. Many efforts have been put forward to train hierarchical models which contain multiple levels of feature extractors. Recently, deep neural network (DNN) as a typical hierarchial model has attracted much attention again since Hinton et al. [1] propose an efficient learning algorithm for so-called deep belief nets (DBN). The variants of DNN have been applied in various fields with its innate advantages ([2], [3], [4], [5]). Lee et al. [2] propose a convolutional DBN for unsupervised learning of hierarchical representation which achieves better performance in image classification and speaker identification. By combining multiple sources with shared hidden representation, Srivastava et al. [3] propose multimodal deep neural network to learn representations for text and image.

In this paper, we propose a multi-task deep neural network (MT-DNN) architecture to handle the multi-label learning problem based on the conclusion that deep architecture can learn good intermediate representations shared across tasks [6]. As we know, multi-label learning generally assigns multiple labels to an instance simultaneously. Each label assignment can be seen as predicting whether the instance owns the label or not. So we transform multi-label learning into multiple single-label assignment tasks by regarding single-label assignment as a binary classification problem. The traditional DNN can be extended for multi-task learning by defining the output layer containing positive class nodes and negative class nodes for each label learning. After defining the architecture of MT-DNN, we employ unlabeled data to pretrain MT-DNN so as to obtain good intermediate representations for multiple binary classification tasks. The pretraining can also provide a good initialization to MT-DNN. Finally, we sum each task's label assignment error as the whole error of MT-DNN, and perform the back-propagation algorithm to fine-tune MT-DNN.

The proposed method has three merits. First, pretraining a multi-task deep neural network can exploit large amounts of unlabeled data to obtain good intermediate representations for all the tasks. Second, unlike many rank-based multi-label learning algorithms [7] which need strategies to determine classification threshold functions, the proposed MT-DNN can automatically assign a set of labels to each instance. Third, MT-DNN naturally models the label dependencies in multi-label learning. The experimental results on two public datasets in Section 4 further verify these merits of our method.

2. BACKGROUND

In this section, we introduce deep belief nets (DBN) and multi-label learning which are the bases of MT-DNN. Par-

ticularly, DBN provides a principled pretraining strategy for MT-DNN.

2.1. Deep Belief Nets

Deep belief nets (DBN) are composed of several restricted boltzmann machines (RBM). A restricted boltzmann machine (RBM) is an undirected graph with a visible layer and a hidden layer. Each visible node is connected to each hidden node. When all the nodes in both layers are binary-valued, the energy function of this model is defined as follows:

$$E(v,h) = -v^T W h - b_1 v - b_2 h \tag{1}$$

where v and h are respectively the visible and hidden nodes, W is the weight matrix between visible nodes and hidden nodes, b_1 and b_2 are respectively the visible biases and hidden biases.

The input in our model is real-valued data, which can not be well modeled by binary visible nodes. We replace the binary RBM by a Gaussian RBM whose visible nodes are linear with Gaussian noise, the energy function becomes:

$$E(v,h) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_i \sum_j \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_j b_j h_j \tag{2}$$

Where $\{W, b_i, b_j\}$ are model parameters, σ_i is the standard deviation of the Gaussian noise for visible unit i.

Based on the energy function above, the joint probability distribution of all the nodes is defined as:

$$P(v,h) = \frac{1}{Z}\exp(-E(v,h)) \tag{3}$$

where Z is a normalization factor that scales $P(v,h)$ to [0,1].

The parameters$\{W, b\}$ are all trained by minimizing the negative log-likelihood $-\sum_h \log P(v,h)$ via gradient descend, the gradient can be efficiently approximated by using contrastive divergence (CD)[8].

Several RBMs or Gaussian RBMs are stacked together to form a deep architecture named deep belief nets, which is a generative model of powerful representability. In a DBN, the nodes between two adjacent layers are fully-connected, but no connection between the nodes in the same layer. Hinton et al. [1] propose an efficient algorithm for pretraining DBN. Pretraining consists of greedily training adjacent two layers as an RBM or Gaussian RBM.

2.2. Multi-Label Learning

Multi-label learning ([7], [9], [10], [11]) is a special kind of supervised learning where each instance can belong to multiple classes. It is a generalized version of multi-class learning where each instance is restricted to belong to only one class.

Let $\mathcal{X}$ denote the instance set and $\mathcal{Y} = \{1, 2, ..., L\}$ denote the label set. Given the training set $\{(x_1, Y_1), ..., (x_n, Y_n)\}$

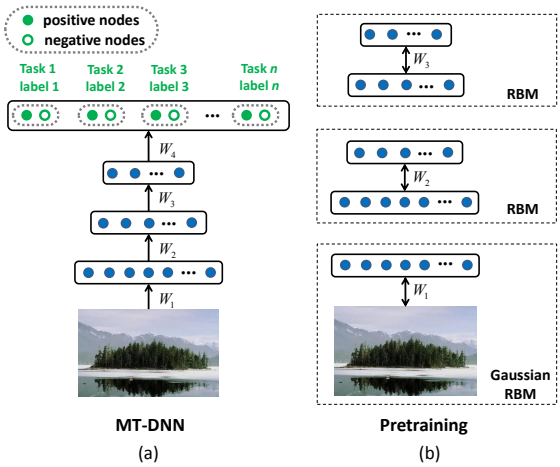

Fig. 1. The proposed multi-task deep neural network (MT-DNN) and its pretraining.

where $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$, the goal of multi-label learning is to learn a multi-label classifier $f : \mathcal{X} \to 2^L$ from the training dataset. In this paper, we apply deep neural network to implement this multi-label classifier.

3. ALGORITHM

3.1. MT-DNN for Multi-label Learning

It is generally considered that deep neural network can learn good intermediate representations shared across multiple tasks from large amounts of unlabeled data, while multi-label learning can be transformed to multiple single-label learning tasks by defining single-label learning as predicting whether an instance owns the label or not. Given these considerations above, we propose multi-task deep neural network (MT-DNN) for multi-label learning.

We illustrate the architecture of a five-layer MT-DNN in Fig.1 (a), which contains an real-valued input layer, three binary hidden layers for representation learning, an output layer for label assignment. The output layer consists of multiple pairwise nodes corresponding to multiple single-label learning, the green solid nodes for the positive classes $\{c_l\}_{l=1,...,L}$ and the green circle nodes for the negative classes $\{\overline{c_l}\}_{l=1,...,L}$. The parameters of these layers are $\{W_i\}_{i=1,...,4}$, respectively. For a simple notation, the biases b_1, b_2 will be ignored below.

Given an instance x, the multi-label classifier f of MT-DNN is

$$f(x) = g(W_4^T(g(W_3^T g(W_2^T g({W_1}^T x))))) \tag{4}$$

Here $g(x)$ is the sigmoid function $\frac{1}{1+e^{-x}}$. In order to determine the l-th label, we need to compare $f_{c_l}(x)$ and $f_{\overline{c_l}}(x)$ corresponding to nodes c_l and $\overline{c_l}$ as follows:

$$\begin{cases} x \text{ owns label } l & \text{if } f_{c_l}(x) \geq f_{\overline{c_l}}(x) \\ x \text{ does not own label } l & \text{if } f_{c_l}(x) < f_{\overline{c_l}}(x) \end{cases} \quad (5)$$

We will pretrain and fine-tune MT-DNN to learn the parameters $\{W_i\}_{i=1,\ldots,4}$ below.

Pretraining Similar to [12], we pretrain MT-DNN with unlabeled data to learn intermediate representations and also provide a good initialization for the network. First, we combine the input layer and the first hidden layer together as a Gaussian RBM, and train the parameters W_1 with contrastive divergence. The conditional probability of the first hidden-layer nodes will be used as the input of the second hidden layer, which is denoted as:

$$p(h = 1|v) = g(W^T v) \quad (6)$$

Then, we combine the first hidden layer and the second hidden layer as a binary RBM, and train the parameters W_2 in a similar way to Gaussian RBM. We repeat this process for the third hidden layer. Although MT-DNN is pretrained greedily, it has been demonstrated that the procedure will not decrease the log-probability of the input data [1].

Fine-tuning After pretraining, we need to fine-tune MT-DNN with labeled data by backpropagating the derivatives of label assignment error. Considering multi-label learning as a multi-task learning problem, we define the whole assignment error of MT-DNN as the summation of each label assignment error.

Take the l-th label assignment error as an example. We first normalize the two outputs $f_{c_l}(x)$ and $f_{\overline{c_l}}(x)$ to obtain the probability of "an instance owns label l" p_l:

$$p_l = \frac{\exp(f_{c_l}(x))}{\exp(f_{c_l}(x)) + \exp(f_{\overline{c_l}}(x))} \quad (7)$$

then compute the cross-entropy as the l-th label assignment error E_l:

$$E_l = -[q_l \log p_l + (1 - q_l) \log(1 - p_l)] \quad (8)$$

where $q_l \in Y$ is the truth label from the training set. Summing over all the label assignment errors, we obtain the whole assignment error E of MT-DNN:

$$E = \sum_{l=1}^{L} E_l \quad (9)$$

Finally, we compute the derivatives of E over $\{W_i\}_{i=1,\ldots,4}$ and perform the back-propagation algorithm to fine-tune MT-DNN.

4. EXPERIMENTAL RESULTS

In order to quantitatively evaluate the performance of MT-DNN, we carry out some experiments in terms of image annotation on two public datasets. We compare the experimental results of MT-DNN with those of two state-of-the-art multi-label learning methods, namely ML-KNN [9] and BP-MLL [7]. It should be noted that although BP-MLL also utilizes neural networks for multi-label learning, our method differs from it in three aspects: 1) the output layer in our method is redefined for each label assignment, 2) our method can exploit unlabeled data through pretraining, 3) our method does not need a threshold function to determine the assigned labels.

4.1. Experiments on the Natural Scene Dataset

In this experiment, we predict labels for a natural scene image dataset provided by Zhang et al. [9], which contains 2,000 natural scene images. All the 5 labels of these images are *desert*, *mountains*, *sea*, *sunset*, and *trees*. The images which have more than one label (e.g., desert+mountains) cover 20% of the dataset, and the average number of labels for each image is 1.3. Fig. 2 shows two images of this dataset. As we can see, mountains and trees can be assigned to Fig. 2(a), and sunset and sea can be assigned to Fig. 2(b).

(a) (b)

Fig. 2. Two samples of the natural scene images [9].

The architecture of our four-layer MT-DNN is designed like this: the input layer, two hidden layers and output layer have 294, 300, 400, 10 nodes, respectively (i.e., a 294-300-400-10 MT-DNN). Hamming loss [9] is chosen as the evaluation criterion, which computes how many times an instance-label pair is miss-classified. The smaller the hamming loss, the better the method. Ten-fold cross-validation is performed in this dataset, which is also used in [9]. The experimental results are listed in Table 1. We can see that our method achieves the lowest hamming loss, both in mean and deviation, which indicates that our method is much more effective and stable.

Table 1. The results on the natural scene dataset.

Methods	Hamming Loss
BP-MLL [7]	0.271 ± 0.016
ML-KNN [9]	0.169 ± 0.071
Our method	$\mathbf{0.157 \pm 0.008}$

4.2. Experiments on the NUS Dataset

We also carry out image annotation experiments on a relatively larger dataset to verify the effectiveness of our method. The used dataset is called NUS-WIDE-OBJECT provided by

Chua et al. [13], which contains 30,000 images and 31 concepts. The images in this dataset are exacted from the photo sharing web site Flickr.com. The concepts of these images are various, such as *boats*, *cars*, *flags*, *horses*, *sky*, *sun*, *tower*, *plane* and *zebra*. Fig. 3 illustrates two sample images of the NUS dataset. As can be seen, sky and plane are two concepts which can be assigned to Fig. 3(a).

(a) (b)

Fig. 3. Two examples of the NUS images [13].

The architecture of our four-layer MT-DNN on this dataset is designed like this: the input layer, two hidden layers and output layer have 634, 3000, 4000, 62 nodes, respectively (i.e., a 634-3000-4000-62 MT-DNN). We do not perform ten-fold cross-validation and can not compute the standard deviation of hamming loss in this dataset, because the training set and testing set have been given explicitly (17,927 training images and 12,073 testing images) [13]. From the experimental results in Table 2, we can see that our method achieves better performance than two state-of-the-art methods (ML-KNN [9], BP-MLL [7]) on this dataset.

Table 2. The results on the NUS dataset.

Methods	Hamming Loss
BP-MLL [7]	0.0442
ML-KNN [9]	0.0348
Our method	**0.0323**

5. CONCLUSION

This paper has transformed multi-label learning to multiple binary classification tasks, and proposed a deep neural network architecture to handle this kind of multi-task problem. Experimental results on image annotation demonstrate that the proposed model has achieved the state-of-the- art performance. It can be seen that deep neural network can be a good choice to perform multi-task learning through defining the output layer with multiple different aims. In the future, we will extend MT-DNN to handle multiple instances and propose a deep neural network architecture for multi-instance multi-label learning.

6. ACKNOWLEDGMENTS

This work is jointly supported by National Natural Science Foundation of China (61175003, 61135002, 61202328), Hundred Talents Program of CAS, National Basic Research Program of China (2012CB316300), the Strategic Priority Research Program of CAS (XDA06030300), and National Key Technology R&D Program (2011BAH11B01).

7. REFERENCES

[1] G. E. Hinton and S. Osindero, "A fast learning algorithm for deep belief nets," *Neural Computation*, 2006.

[2] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," *International Conference on Machine Learning*, 2009.

[3] N. Srivastava and R. Salakhutdinov, "Learning representations for multimodal data with deep belief nets," *International Conference on Machine Learning*, 2012.

[4] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, 2009.

[5] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech, and Language Processing*, 2012.

[6] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, 2009.

[7] M. L. Zhang and Z. H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," *IEEE Trans. on Knowledge and Data Engineering*, 2006.

[8] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, 2002.

[9] M. L. Zhang and Z. H. Zhou, "Ml-knn: a lazy learning approach to multi-label learning," *Pattern Recognition*, 2007.

[10] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang, "Correlative multi-label video annotation," *ACM International Conference on Multimedia*, 2007.

[11] Z. Zha, X. Hua, T. Mei, J. Wang, G. Qi, and Z. Wang, "Joint multi-label multi-instance learning for image classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, 2006.

[13] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: A real-world web image database from national university of singapore," *ACM International Conference on Image and Video Retrieval*, 2009.