MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

Transfer Learning Method using Multi-Prediction Deep Boltzmann Machines for a small scale dataset

Yoshihide Sawada, Kazuki Kozuka
Panasonic Corporation, 3-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan

{sawada.yoshihide, kozuka.kazuki}@jp.panasonic.com

Abstract

In this article, we propose a transfer learning method using the multi-prediction deep Boltzmann machine (MPDBM). In recent years, deep learning has been widely used in many applications such as image classification and object detection. However, it is hard to apply a deep learning method to medical images because the deep learning method needs a large number of training data to train the deep neural network. Medical image datasets such as X-ray CT image datasets do not have enough training data because of privacy. In this article, we propose a method that re-uses the network trained on non-medical images (source domain) to improve performance even if we have a small number of medical images (target domain). Our proposed method firstly trains the deep neural network for solving the source task using the MPDBM. Secondly, we evaluate the relation between the source domain and the target domain. To evaluate the relation, we input the target domain into the deep neural network trained on the source domain. Then, we compute the histograms based on the response of the output layer. After computing the histograms, we select the variables of the output layer corresponding to the target domain. Then, we tune the parameters in such a way that the selected variables respond as the outputs of the target domain. In this article, we use the MNIST dataset as the source domain and the lung dataset of the X-ray CT images as the target domain. Experimental results show that our proposed method can improve classification performance.

1 Introduction

In recent years, deep learning (DL) has been widely used in the fields of machine learning and pattern recognition [1, 2, 3] because of its high recognition performance. DL methods train the deep neural network with a large number of parameters using a large number of training data. For example, Le et al. [2] train 1 billion parameters using 10 million training images, and Krizhevsky et al. [3] train 60 million parameters using 1.2 million training images.

On the other hand, medical image datasets such as X-ray CT image datasets do not have enough data for training the deep neural networks because of privacy. Therefore, many applications including computer aided diagnosis (CAD) systems use conventional sophisticated features [4, 5]. In this article, we propose a method that combines the DL and the transfer learning method for a small number of training data. It should be noted that the source domain (training data of non-medical images) has a large number of data and the target domain (training data of medical images) has a small number of data.

Transfer learning is a method that re-uses knowledge about the source domain to solve the target task [6]. For example, Oquab et al. [7] trained convolutional neural network (CNN) with the ImageNet [8] as the source domain. After training the CNN, they re-use the parameters from the input layer on the mid-level hidden layer. Then, they add a new layer and tune the parameters using the target domain. In their article, they show that their proposed method outperformed other methods.

In this article, we propose a new transfer learning method using DL. Figure 1 shows the outline of our proposed method. Let $\boldsymbol{x}_s$ ($\boldsymbol{x}_t$) be the sample of the source (target) domain and let y_s (y_t) be the label corresponding to $\boldsymbol{x}_t$ ($\boldsymbol{x}_t$). Let D_s be the deep neural network trained on the source domain $\{\boldsymbol{x}_s\}$, and let $\boldsymbol{w}_s^{(i\to o)}$ be the parameters from the input layer to the output layer of D_s. Our proposed method firstly trains the deep neural network D_s. Secondly, we evaluate the relation between the source domain $\{\boldsymbol{x}_s\}$ and the target domain $\{\boldsymbol{x}_t\}$. To evaluate the relation, we input the target domain $\{\boldsymbol{x}_t\}$ into D_s. Next, we compute the histograms based on the response of the output layer of D_s. After computing the histograms, we select the variables of the output layer that relate to the target domain $\{\boldsymbol{x}_t\}$. Finally, we tune the parameters in such a way that the selected variables respond as the outputs of the target domain $\{\boldsymbol{x}_t\}$.

The difference between our proposed method and Oquab's method is the constraint. Oquab's method constrains the network by using the parameters $\boldsymbol{w}_s^{(i\to m)}$, which denote the parameters from the input layer to the mid-level hidden layer of D_s. On the other hand, our proposed method constrains by using all parameters $\boldsymbol{w}_s^{(i\to o)}$. This means that our proposed method adds the constraint harder than Oquab's method. Therefore, we think that our proposed method is suitable for avoiding overfitting when you have a small scale target domain.

We evaluated our proposed method by using the MNIST handwritten character dataset [9] as the source domain and the lung dataset of the X-ray CT images as the target domain. CT images contain many slices, and the lung lesions are life-threatening. Thus, the CAD system for lung lesion needs high classification performance. We adopted the multi-prediction deep Boltzmann machine (MPDBM) [10] as the DL method. Multi-prediction means the procedure includes prediction of any subset of the variables given the comple-

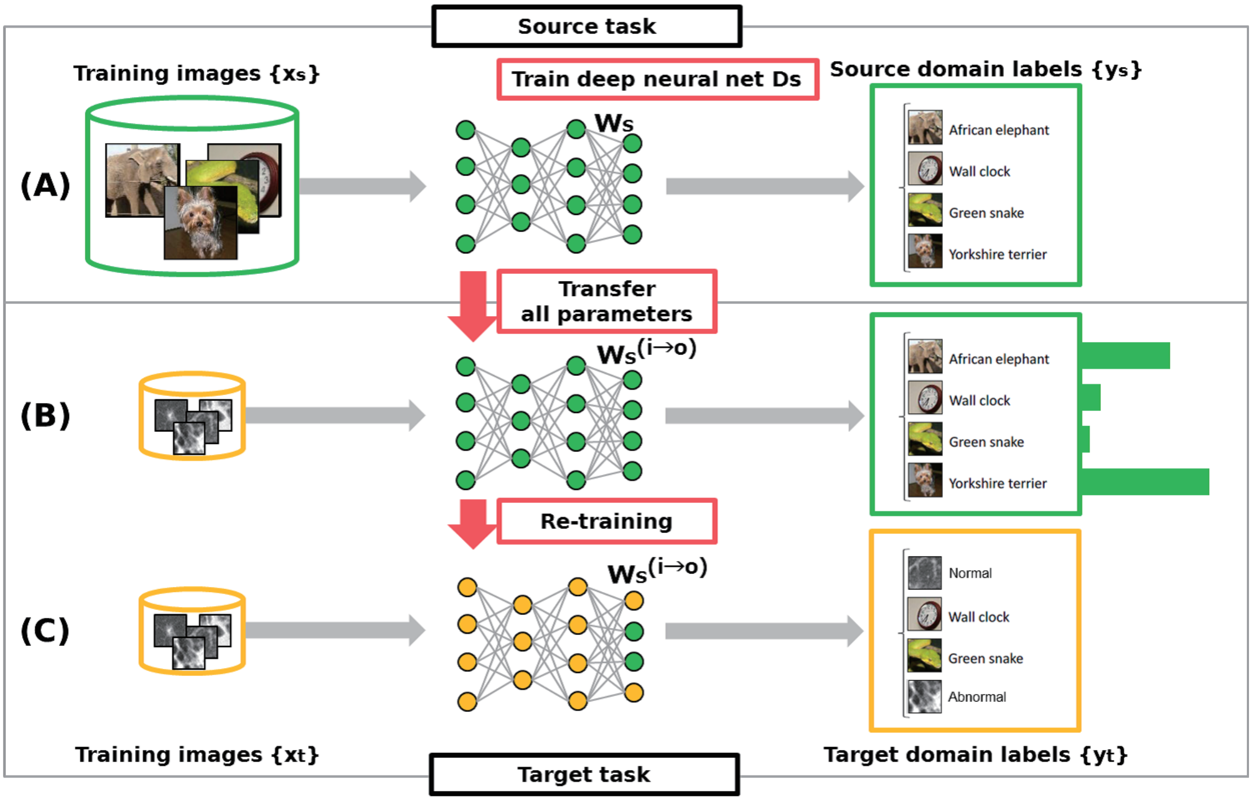


Figure 1. Outline of our proposed method. We re-use all parameters trained on source domain. (A): training deep neural network D_s, (B): evaluating relation between source domain and target domain, (C): tuning $\boldsymbol{w}_s^{(i\to o)}$ based on relation.

ment of that subset of variables [10]. The MPDBM does not require greedy layerwise pretraining and outperforms the standard DBM [10]. We experimentally evaluated the relationship between the classification rate and the number of layers T that transfer from the source domain to the target domain. Experimental results showed that as the number of transferred layers T becomes larger, the classification performance becomes higher.

2 Proposed Method

In this article, we propose a method that combines the MPDBM and the transfer learning. Our proposed method re-uses all parameters to avoid overfitting.

2.1 Multi-Prediction Deep Boltzmann Machine

We train the deep Boltzmann machine using the following objective function [10],

$$J(\{(\boldsymbol{x},y)\},\boldsymbol{w}) = -\sum_{(\boldsymbol{x},y)\in\{\boldsymbol{x},y\}}\sum_i \log \hat{p}^*(O_{S_i},\boldsymbol{w}), \quad (1)$$

where $\boldsymbol{x}$ is the input vector, y is the label, $O=[\boldsymbol{x},y]^\top$, and $\boldsymbol{w}$ is the parameter. O_{S_i} is the subset of the variables in O, and $\hat{P}^*(O_{S_i},\boldsymbol{w})$ is the following mean-field approximation,

$$\hat{p}^*(O_{S_i},\boldsymbol{w}) = \arg\min_{\hat{p}} KL(\hat{p}(O_{S_i},\boldsymbol{w})||p(O_{S_i},\boldsymbol{w}|O_{-S_i})), \quad (2)$$

where O_{-S_i} is the subset of the variables in O except for O_{S_i}, and $KL(.||.)$ is the KL-divergence. $p(O_{S_i},\boldsymbol{w}|O_{-S_i})$ is the conditional probability distribution of $p(O,\boldsymbol{w})$,

$$p(O,\boldsymbol{w}) = \frac{1}{Z_e}\exp(-E(O,\boldsymbol{w})) \quad (3)$$

where Z_e is the partition function, and $E(O,\boldsymbol{w})$ is the energy function of the deep Boltzmann machines.

When we train the MPDBM, we use the mini-batch stochastic gradient descent (SGD) on (1) [10].

2.2 Transfer Learning method using MPDBM

We explain the transfer learning method using the MPDBM for a small scale target domain.

Let $\boldsymbol{w}_s$ be the parameters trained on the source domain, let N_s and N_t be the number of labels of the source and the target domain ($N_s \geq N_t$), and let M_s and M_t be the number of training samples of the source and the target domain ($M_s \gg M_t$).

Figure 1 shows the outline of our proposed method and the algorithm is as follows:

1. Source task step:
   (a) Initialize the parameters $\boldsymbol{w}_s$.
   (b) Minimize $J(\{(\boldsymbol{x}_s,y_s)\},\boldsymbol{w}_s)$.

2. Target task step:

Table 1. Comparison of classification performance with respect to method for selecting appropriate variables $\mathcal{V}^*$. $v^*(1) = 1, v^*(2) = 2$ represents variables selected based on (7), and $v^*(1) = 8, v^*(2) = 9$ represents randomly selected ones.

	Performance (%)
$v^*(1) = 8, v^*(2) = 9$	97.5
$v^*(1) = 1, v^*(2) = 2$	99.6

Table 2. Comparison of classification performance with respect to different structures.

	Performance (%)
$(784, 500, 500, 10)$	99.6
$(784, 500, 50, 10)$	99.3

(a) Evaluate relation between the source domain and the target domain.

(b) Select the appropriate variables of the output layers that relate to the target domain.

(c) Minimize $J(\{(\boldsymbol{x}_t, y_t)\}, \boldsymbol{w}_s^{(i\to o)})$.

At the source task step, we train D_s by using the MPDBM described in section 2.1. Then, we re-use all parameters of D_s for the target task. For re-using all parameters, we evaluate the relation between the source and target domain by computing the histograms of each label. The histogram of l's label is as follows ($l = 1, 2, \cdots, N_t$):

$$p_l(v) = \frac{1}{Z_h} h_l(v), \tag{4}$$

where v $(= v_1, v_2, \cdots, v_{N_s})$ is the output variable of D_s, and Z_h is the partition function, and

$$h_l(v) = \sum_{j=1}^{M_t(l)} h_l(v|\boldsymbol{x}_{t,j}), \tag{5}$$

where $M_t(l)$ is the number of samples of l's label, and $h_l(v|\boldsymbol{x}_{t,j})$ is the output probability given $\boldsymbol{x}_{t,j}$. In this article, we use the following approximation.

$$h_l(v|\boldsymbol{x}_{t,j}) = \begin{cases} 1, \ \max_k v_k, \\ 0, \ \text{otherwise}, \end{cases} \tag{6}$$

where $k = 1, 2, \cdots, N_s$.

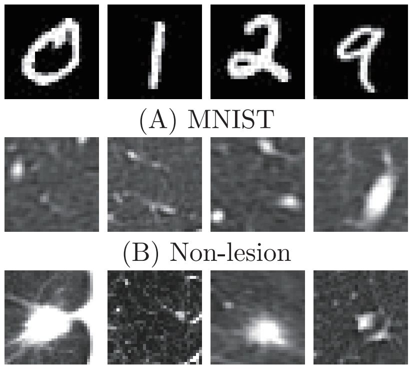
By using $p_l(v)$, we select the appropriate variables of each label $\mathcal{V}^* = \{v^*(l) | l = 1, 2, \cdots, N_t\}$ as follows:

$$\begin{aligned} &\mathcal{V}^* = \arg\max_{\mathcal{V}} \textstyle\sum_{l=1}^{N_t} p_l(v(l)), \\ &s.t. \ \ v^*(l) \neq v^*(l'), \ (l \neq l'), \end{aligned} \tag{7}$$

where $\mathcal{V} = \{v(l) | l = 1, 2, \cdots, N_t\}$.

After selecting $\mathcal{V}^*$, we re-train D_s in such a way that $\mathcal{V}^*$ respond as the outputs of each label of the target. It should be noted that the re-training of D_s corresponds to the training of the deep neural network using the initial parameters $\boldsymbol{w}_s^{(i\to o)}$.

3 Experimental Results


(A) MNIST

(B) Non-lesion

(C) Lesion

Figure 2. Examples of dataset.

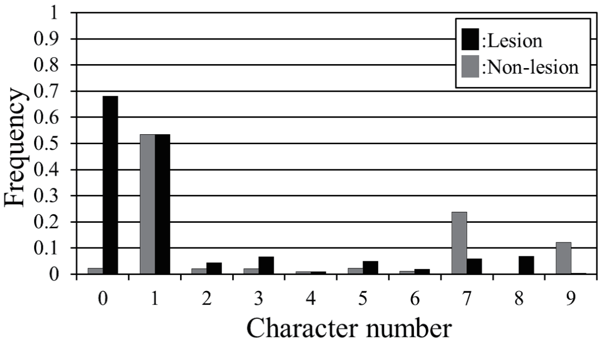


Figure 3. The histograms $p_l(v)$ of D_s = $(784, 500, 500, 10)$.

We evaluated the classification performance by using the MNIST handwritten character dataset [9] as the source domain and the lung dataset of the X-ray CT images as the target domain. Figure 2 shows examples. Fig.(A) represents the examples of MNIST, Fig.(B) represents non-lesion images, and Fig.(C) represents lesion images. The size of these images is 28×28 pixels, and the determination of lesion or non-lesion was based on diagnosis by radiologists.

We used $M_s = 60000$ and $N_s = 10$ (character number from "0" to "9"), and $M_t = 2000$ and $N_t = 2$ (lesion or non-lesion). The number of samples of each label is $M_s(1) = M_s(2) = \cdots = M_s(10) = 6000$, and $M_t(1) = M_t(2) = 1000$. l's label of the source domain represents the character "$l - 1$", 1's label of the target domain represents "lesion", and 2's label represents "non-lesion". As the test dataset, we used 140 images of lesions and 140 images of non-lesions. These test images are not included in the training dataset.

3.1 Effectiveness study of relation evaluation

Figure 3 shows the histograms of the relation. The black bar represents the histogram of the lesions and the gray bar represents the histogram of the non-lesions. When we computed these histograms, we used D_s with 784 units in the input layer, 500 units in the first and the second hidden layer, and 10 units in the output layer. In this article, $D_s = (784, 500, 500, 10)$. As shown in this figure, the highest relation of the lesion images is the character "0" ($v^*(1) = 1$) and the non-lesion images is the character "1" ($v^*(2) = 2$).

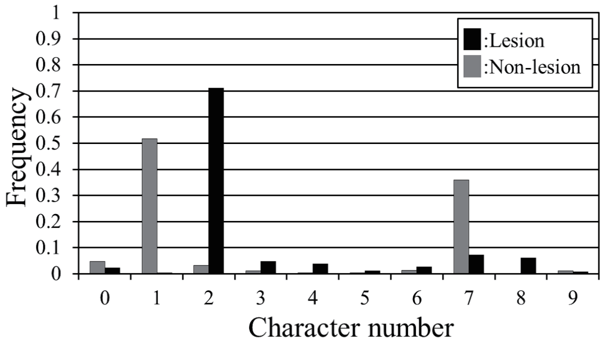


Figure 4. Histograms $p_l(v)$ of $D_s = (784, 500, 50, 10)$.

Table 3. Comparison of classification performance with respect to transferring value T.

	Performance (%)
$T = 0$	93.2
$T = 1$	98.2
$T = 2$	98.5
$T = 2$ (Adding a new layer)	98.9
$T = 3$ (Proposed)	99.6

Table 1 shows the comparison of the classification performance with respect to the method for selecting the appropriate variables $\mathcal{V}^*$. $v^*(1) = 1$ and $v^*(2) = 2$ were selected by (7), and $v^*(1) = 8$ and $v^*(2) = 9$ were selected randomly. As shown in this table, the result based on (7) outperformed the randomly selected one.

Next, we compared the performance of the other structures. Figure 4 and table 2 show the histograms and the classification performance using $D_s = (784, 500, 50, 10)$. Compared to these results, the appropriate variables v^* and the classification performance changed depending on D_s.

These results indicate the importance of evaluating the relation between the source and the target domain.

3.2 Comparison study of classification performance

Table 3 shows the comparison of the classification performance with respect to the number of transferred layers T. For example, $T = 0$ represents no transfer $\boldsymbol{w}_s$, and $T = 3$ represents to transfer all parameters. It is noted that the deep neural network of $T = 0, T = 1, T = 2$, and $T = 3$ trained on the same hidden layers ($D_s = (784, 500, 500, 10)$). On the other hand, $T = 2$ (adding a new layer) corresponds to Oquab's method [7], and we added 500 units to the third layer ($D_s = (784, 500, 500, 500, 10)$).

Our proposed method outperformed other methods. This implies that using all parameters trained for other tasks will improve the classification performance of the target task if you have a small-scale dataset.

4 Conclusion and Future Work

We propose a transfer learning method for a small number of target samples. Firstly, we trained a deep neural network D_s on the MNIST dataset. For training D_s, we used the multi-prediction deep Boltzmann machine (MPDBM). Secondly, we computed the histograms based on the response of the output layer of D_s to evaluate the relation between the MNIST and the medical image dataset. After computing the histograms, we selected the appropriate variables of the output layers that relate to the MNIST dataset. Then, we tuned the parameters of D_s in such a way that the selected variables respond as lesion or non-lesion.

Experimental results showed that selecting the variables based on the relation is effective, and our proposed method outperformed the classification performance. Future work is to compare the classification performance by using other source domains and try to use another method for training D_s.

References

[1] Y. Bengio, "Learning deep architectures for ai," Foundations and trends® in Machine Learning, vol.2, no.1, pp.1–127, 2009.

[2] Q.V. Le, "Building high-level features using large scale unsupervised learning," Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference onIEEE, pp.8595–8598 2013.

[3] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, pp.1097–1105, 2012.

[4] Y. Sawada, T. Oku, H. Hotani, J. WU, T. TAKEDA, and Y. WATANABE, "Improved detection of tumors in fdg-pet/ct images based-on single-class classifier," International Forum on Medical Imaging in Asia 2009 (IFMIA 2009), vol.108, pp.229–234, 2009.

[5] K. Kozuka, K. Takata, K. Kondo, M. Kiyono, M. Tanaka, and T. Sakai, "Development of lung ct image-retrieval system based on imaging findings and an image-selection interface," IEEE EMBC 2013, Annual International Conference, p.1, Springer, 2013.

[6] S.J. Pan and Q. Yang, "A survey on transfer learning," Knowledge and Data Engineering, IEEE Transactions on, vol.22, no.10, pp.1345–1359, 2010.

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," Computer Vision and Pattern Recognition, 2014. CVPR 2014. Proceedings of the 2014 IEEE Computer Society Conference on, pp.1–8, 2014.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference onIEEE, pp.248–255 2009.

[9] 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.

[10] I. Goodfellow, M. Mirza, A. Courville, and Y. Bengio, "Multi-prediction deep boltzmann machines," Advances in Neural Information Processing Systems, pp.548–556, 2013.