Adaptive and Dynamic Knowledge Transfer in Multi-task Learning with Attention Networks

Tao Ma and Ying Tan(✉)

Key Laboratory of Machine Perception (MOE), Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
{pku_mark,ytan}@pku.edu.cn

Abstract. Multi-task learning has shown promising results in many applications of machine learning: given several related tasks, it aims to generalize better on the original tasks, by leveraging the knowledge among tasks. The knowledge transfer mainly depends on task relationships. Most of existing multi-task learning methods guide learning processes based on predefined task relationships. However, the associated relationships have not been fully exploited in these methods. Replacing predefined task relationships with the adaptively learned ones may lead to superior performance as it can avoid the misguiding of improper predefinition. Therefore, in this paper, we propose Task Relation Attention Networks to adaptively model the task relationships and dynamically control the positive and negative knowledge transfer for different samples in multi-task learning. To evaluate the effectiveness of the proposed method, experiments on various datasets are conducted. The experimental results demonstrate that the proposed method outperforms both classical and state-of-the-art multi-task learning baselines.

Keywords: Representation learning · Multi-task learning · Knowledge transfer · Attention networks

1 Introduction

Multi-task learning (MTL) aims to generalize better on the original tasks, by leveraging the shared knowledge among tasks [15]. In MTL, positive knowledge transfer leads to improved performance, because: 1) more related information is incorporated into the target tasks, which benefits the feature learning process to obtain better feature representations; 2) the incorporated information of positively related tasks acts as regularizer to avoid the risks of over-fitting. The knowledge transformer mainly depends on the task relationships. Therefore, how to appropriately model task relationships and how to control the knowledge transfer among tasks are crucial in MTL.

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 1–13, 2020.
https://doi.org/10.1007/978-981-15-7205-0_1

With the recent advances in deep learning, MTL with deep neural networks has been used widely and successfully across many applications of machine learning, from natural language processing [7] to computer vision [11]. In most of these existing methods, the task relationships guiding the learning process are generally predefined, and the knowledge transfer among tasks relies on the sharing of hidden layers. Despite they have achieved promising results, but there are still several challenges for further improving the performance of MTL methods.

The first challenge is adaptively learning the task relationships instead of relying on predefined relationships. For some multi-task learning problems with complex task associations, the pre-definition based on limited human knowledge requires costly human efforts. Besides, if there are not adequate efforts for sophisticated pre-definitions, there is likely to be negative knowledge transfer because of the misguiding of improperly predefined task relationships [8,16]. Since an improper pre-definition of task relationships may result in negative knowledge transfer, it is essential for MTL methods to adaptively and appropriately model the task relationships with learning-based modules.

The second challenge is controlling the knowledge transfer with the dynamically learned task relationships instead of the fixed ones. The relationships among tasks are not constantly fixed, but vary slightly from different samples. However, in most of the methods relying on pre-definition, the task relationships are fixed [9], even in some MTL methods equipped with adaptively learning modules [14]. A concrete example is, in the works of [21], the task relationships are determined by the inputs X and outputs Y of training data. In the testing phase, the model directly performs predictions on the learned relationships and the inputs X_{test}, i.e., the relationships are fixed for the testing data. However, the task relationships in different samples may not be necessarily consistent. Therefore, dynamically modeling these relationships based on different inputs may lead to superior performance.

To address these challenges, in this paper, we propose Task Relation Attention Networks (TRAN) to adaptively capture the task relationships and dynamically control the knowledge transfer in MTL. Specifically, TRAN is an attention-based model to adaptively capture the task relationships via task correlation matrix according to their inputs and specify the shared feature representations for different tasks. Since the task relationships are adaptively learned by TRAN during the learning process, it is replacing the predefined relationships. And TRAN relies on the inputs, therefore, it can dynamically model the correlations for different samples.

To evaluate the effectiveness of the proposed method, the experiments on various datasets are conducted. The experimental results demonstrate the proposed method can outperform both classical and state-of-the-art MTL baselines. The contributions of this paper can be summarized as follows:

- This paper proposes Task Relation Attention Networks (TRAN) to adaptively learn the task relationships to replace the predefined ones.
- The proposed method can dynamically control the knowledge transfer in multi-task learning based on the adaptively learned task relationships.

– This paper provides an explicable learning-based framework for multi-task learning to learn the shared feature representations for different tasks.

2 Related Works

2.1 Multi-task Learning

Multi-task Learning (MTL) provides an efficient framework for leveraging task relationships to achieve knowledge transfer and leads to improved performance.

The typical MTL architectures were sharing the bottom layers for all tasks and split top layers for different tasks, proposed by [3]. Afterwards, there have been some attempts to design partly shared architectures between tasks, instead of only sharing the bottom hidden layers. For leveraging the task relationships in MTL, there are some recent examples. The cross-stitch networks [14] learned an optimal combination of task-specific representations for each task using linear units. The tensor factorization model [20] generated the hidden layer parameters for each task. The multi-gate mixture-of-experts model [13] using gating mechanism to capture the task differences and implicitly model the task relationships. In the works of [21], they applied attention networks to statically capture the task relationships.

Compared to the typical MTL methods, these works performed better feature learning for shared and task-specific representations and achieved better performance. However, there are still some limitations: the method of [14] can hardly expand to a large number of tasks; the method of [20] relies on a specific application scenario; the method of [13] applies linear gates to implicitly model the task relationships and performs poor efficiency as the number of experts increases; the method of [21] captures the task relationships statically.

2.2 Attention-Based Neural Networks

The key idea of attention mechanism is mainly based on the human visual attention, which has been successfully applied in various applications, such as natural machine translation [12] and text summarization [1].

Graph attention networks [18] was proposed for feature learning of graph-structured data, based on the self-attention mechanism. [10] applied the self-attention networks for time series warping. [17] performed self-attention on sentence encoding models, called Transformer, dispensing with recurrence and convolutions entirely [17]. Based on the Transformer, a language representation model called BERT was proposed [5]. In this paper, we attempt to adaptively model the task relationships in MTL with self-attention networks. Based on the learned relationships, the knowledge transfer among tasks can be dynamically controlled and each task can obtain a better shared feature representation, leading the MTL method to better performance.

3 Method

3.1 Problem Statement for Multi-task Learning

Given a single task, which can be regression or classification, the formal definition is as follows:

$$\hat{\boldsymbol{Y}} = \mathrm{M}(\boldsymbol{X}) \approx \mathrm{E}\left(\boldsymbol{Y}|\boldsymbol{X}\right), \tag{1}$$

where $\boldsymbol{X}$ represents the inputs, $\boldsymbol{Y}$ represents the ground truth values, $\hat{\boldsymbol{Y}}$ is the predicted values, $\mathrm{E}(\cdot)$ is the mathematical expectation and $\mathrm{M}(\cdot)$ is the model.

In MTL, assuming that there are K tasks, the problem can be described as follows:

$$\begin{aligned}\left(\hat{\boldsymbol{Y}}_1, \hat{\boldsymbol{Y}}_2, ..., \hat{\boldsymbol{Y}}_K\right) &= \mathrm{MTL}\left(\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K\right) \\ &\approx \mathrm{E}\left(\boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_K|\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K\right),\end{aligned} \tag{2}$$

where $\mathrm{MTL}(\cdot)$ is the multi-task learning method, $\boldsymbol{X}_i, \boldsymbol{Y}_i, \hat{\boldsymbol{Y}}_i$ are respectively the inputs, labels and predictions of each task. In some real-world scenarios, the inputs of different tasks can be the same, i.e., $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K = \boldsymbol{X}_s$.

3.2 Task Relation Attention Networks

We apply attention networks to model the task relationships to help shared feature learning, called Task Relation Attention Networks (TRAN). Given K tasks, the inputs are a set of task features, $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K\}, \boldsymbol{X}_i \in \mathrm{R}^n$, n is the dimensionality of features.

Given task i and j, there is a shared attention network a_t measuring the attention correlations $\boldsymbol{e}_{ij}$ between two tasks, processed as follows:

$$\boldsymbol{e}_{ij} = a_t(\boldsymbol{W}_i\boldsymbol{X}_i||\boldsymbol{W}_j\boldsymbol{X}_j),\ i, j = 1, 2, ..., K, \tag{3}$$

where $\boldsymbol{W}_i \in \mathrm{R}^{n\times d}$ and $\boldsymbol{W}_j \in \mathrm{R}^{n\times d}$ represent the encoding networks, modeling the original inputs into high-level latent representations with the dimensionality of d, and $||$ is the concatenation operation.

The attention weights for task i are normalized by softmax function to obtain the associated relationships between other tasks and task i, $(\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, ..., \boldsymbol{\alpha}_{iK})$, processed as:

$$\boldsymbol{\alpha}_{ij} = \mathrm{softmax}_i(\boldsymbol{e}_{ij}) = \frac{\exp(\boldsymbol{e}_{ij})}{\sum_{k=1}^{K}\exp(\boldsymbol{e}_{ik})}, j = 1, 2, ..., K. \tag{4}$$

The attention networks a_t are implemented with fully-connected neural networks with the activation function of LeakyReLU. And the learning process of attention networks can be described as:

$$\boldsymbol{\alpha}_{ij} = \frac{\exp\left[a_t(\boldsymbol{W}_i\boldsymbol{X}_i||\boldsymbol{W}_j\boldsymbol{X}_j)\right]}{\sum_{k=1}^{K}\exp\left[a_t(\boldsymbol{W}_i\boldsymbol{X}_i||\boldsymbol{W}_k\boldsymbol{X}_k)\right]}. \tag{5}$$

The learned attention weights for target task i reflect the correlations between other tasks and task i. And all attention weights compose the task correlation matrix $\boldsymbol{A}$.

$$\begin{aligned} \boldsymbol{A}_i &= (\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, \ldots, \boldsymbol{\alpha}_{iK}), \ i = 1, 2, \ldots, K, \\ \boldsymbol{A} &= (\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_K)^T. \end{aligned} \tag{6}$$

The task-specific representations for task i is $\boldsymbol{s}_i$, the combination of all task latent representations with their attention weights for task i, as follows:

$$\begin{aligned} \boldsymbol{s}_i &= \boldsymbol{A}_i \left(\boldsymbol{W}_1\boldsymbol{X}_1, \boldsymbol{W}_2\boldsymbol{X}_2, \ldots, \boldsymbol{W}_K\boldsymbol{X}_K\right)^T \\ &= \sum_{j=1}^{K} \boldsymbol{\alpha}_{ij}\boldsymbol{W}_j\boldsymbol{X}_j, \ i = 1, 2, \ldots, K. \end{aligned} \tag{7}$$

We perform multi-head attention mechanism on TRAN, which allows H independent attention networks to learn the attention weights in parallel and applies linear transformation $\boldsymbol{W}_H = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_H)$ to combine them. The final task-specific representation for task i is processed as follows:

$$\begin{aligned} \boldsymbol{s}_i &= \boldsymbol{W}_H(\boldsymbol{A}_i^1, \ldots, \boldsymbol{A}_i^H)^T(\boldsymbol{W}_1\boldsymbol{X}_1, \ldots, \boldsymbol{W}_K\boldsymbol{X}_K)^T \\ &= \sum_{h=1}^{H}\sum_{j=1}^{K} \boldsymbol{w}_h\boldsymbol{\alpha}_{ij}^h\boldsymbol{W}_j\boldsymbol{X}_j, \ i = 1, 2, \ldots, K. \end{aligned} \tag{8}$$

The illustration of feature learning is presented in Fig. 1.

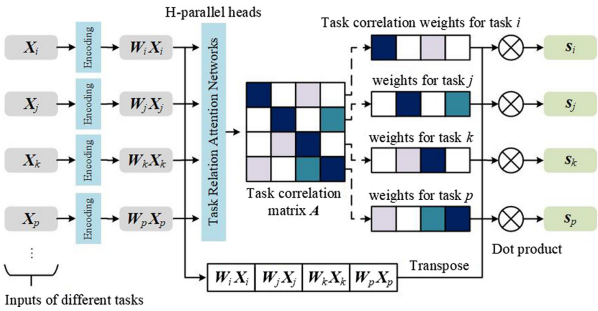


Fig. 1. Illustration of the feature learning with Task Relation Attention Networks.

3.3 Prediction Layer

For multi-task prediction, each task is equipped with a feed-forward sub-layer to convert the final task-specific representations to the predicted values. Each feed-forward sub-layer consists of two layers: the first one $\boldsymbol{W}_i^e$ is a fully-connected neural network with ReLU activation and skip-connection for embedding the final representations; the second one $\boldsymbol{W}_i^p$ is a linear transformation for prediction. The formal equation is described as:

– For regression tasks,

$$\hat{\boldsymbol{Y}}_i = \boldsymbol{W}_i^p \left(\boldsymbol{W}_i^e \boldsymbol{s}_i + \boldsymbol{s}_i \right). \tag{9}$$

– For classification tasks,

$$\hat{\boldsymbol{Y}}_i = \text{softmax} \left(\boldsymbol{W}_i^p \left(\boldsymbol{W}_i^e \boldsymbol{s}_i + \boldsymbol{s}_i \right) \right). \tag{10}$$

3.4 Objective Function

For multi-task learning, all tasks are jointly trained by optimizing a joint loss function L_{joint}. Given the inputs $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K\}$ and labels $\boldsymbol{Y} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_K\}$, the joint loss function is defined as:

$$L_{joint}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{K} \lambda_i L_i(\boldsymbol{X}_i, \boldsymbol{Y}_i) + \beta_W ||\boldsymbol{W}||_F^2 + \beta_A ||\boldsymbol{A} - \boldsymbol{I}||_F^2, \tag{11}$$

where the first item is the combination of the task-specific losses L_j with their loss weights λ_j; the second item is the regularization for all the trainable parameters $\boldsymbol{W}$; the third item is the regularization for the learned attention correlation matrix $\boldsymbol{A}$ to ensure the auto-correlations of tasks. For each task, the task-specific loss is defined as:

– Mean squared error (MSE) for regression tasks,

$$L_i(\boldsymbol{X}_i, \boldsymbol{Y}_i) = \text{MSE}(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i). \tag{12}$$

– Cross entropy for classification tasks,

$$L_i(\boldsymbol{X}_i, \boldsymbol{Y}_i) = \text{CrossEntropy}(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i). \tag{13}$$

4 Experiments

4.1 Dataset

The performance of the proposed method is evaluated on three datasets: Census-income dataset, FashionMnist dataset and Sarcos dataset.

– **Census-income dataset**: The Census-income dataset is from UCI Machine Learning Repository [2]. It is extracted from the 1994 Census database, which contains 299,285 instances of demographic information for American adults. We construct two multi-task learning problems based on 40 features.
  - Task 1: predict whether the income exceeds \$50K;
    Task 2: predict whether this person is never married.
  - Task 1: predict whether the education level of this person is at least college;
    Task 2: predict whether this person is never married.

- **FashionMnist dataset**: The samples in FashionMnist are 28×28 grayscale images with 10-class labels [19], similar to Mnist. We construct a multi-task learning problem: Task 1 is the original 10-class classification task; Task 2 is predict if the objects are shoes, or female products, or another type. All task shares the same inputs.
- **Sarcos dataset**: This is a regression dataset [4] where the goal is to predict the torque measured at each joint of a 7 degrees-of-freedom robotic arm, given the current state, velocity, and acceleration measured at each joint (7 torques for 21-dimensional inputs). Following the procedure of [4], we have 7 regression tasks, where each task is to predict one torque.

4.2 Baselines

The baseline methods to be compared with are as follows:

- **LASSO**: This is the classic linear method, learning each task independently with L1-norm regularization.
- **Bayesian**: Another linear method, learning each task independently with Bayesian inference.
- **Neural Networks (NN)**: For regression tasks and general classification tasks, we apply fully-connected neural networks with one hidden layer. For image classification, we apply single-layer convolutional neural networks.
- **Shared-bottom MTL**: This is a typical MTL method, where all tasks share the bottom hidden layers and have top sub-layers for prediction. In this method, the task relationships are predefined and fixed.
- **L2-Constrained MTL**: This is a classical MTL method [6], where the parameters of different tasks are shared softly by an L2-constraint. Given two tasks, the prediction of each task can be described as:

$$\hat{\boldsymbol{Y}}_1 = f(\boldsymbol{X}_1, \theta_1), \ \hat{\boldsymbol{Y}}_2 = f(\boldsymbol{X}_2, \theta_2), \tag{14}$$

  where θ_1, θ_2 are the parameters of each task. And the objective function of multi-task learning is:

$$L_1\left(\boldsymbol{Y}_1, \hat{\boldsymbol{Y}}_1\right) + L_2\left(\boldsymbol{Y}_2, \hat{\boldsymbol{Y}}_2\right) + \alpha||\theta_1 - \theta_2||_2^2, \tag{15}$$

  where α is a hyper-parameter. This method models the task relationships with the magnitude of α.
- **Cross-stitch Networks (CSN)**: This is a deep learning based MTL method [14]. The knowledge is shared between tasks by a linear units, call cross-stitch. Given two tasks, $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are the outputs of previous hidden layers of each task, and the outputs of cross-stitch are described as:

$$\begin{bmatrix} \tilde{\boldsymbol{h}}_1 \\ \tilde{\boldsymbol{h}}_2 \end{bmatrix} = \begin{bmatrix} \alpha_{11} \ \alpha_{12} \\ \alpha_{21} \ \alpha_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{bmatrix}, \tag{16}$$

  where $\alpha_{ij}, \ i, j = 1, 2$ are trainable linear parameters representing the knowledge transfer.

- **Multi-gate Mixture-of-Experts (MMoE)**: It adopts the multi-gate mixture-of-experts structure to MTL [13]. In this method, there is a group of expert networks as the bottom layers, and the top task-specific layers can utilize the experts differently with gating mechanism.
- **Multiple Relational Attention Networks (MRAN)**: It was recently proposed [21], applying attention networks to model multiple types of relationships in MTL. However, the task relationships in this method are statically modeled. The relationships are determined in the training phase, and in the testing phase, the relationships are fixed. Our method is able to dynamically capture the task relationships, and we will discuss about the differences between this method and ours.

4.3 Performance Comparison

The proposed method is Task Relation Attention Networks (TRAN), and MH-TRAN means it is equipped with multi-head mechanism. Note that, because the code and some datasets of baseline MRAN are not released yet, we only compare its performance in the available Sarcos dataset, which is reported in their paper.

Overall Comparison. The performance comparison on Census-income, FashionMnist and Sarcos datasets are presented in Table 1, 2 and 3 respectively. From all the tables, we can observe that both TRAN and MH-TRAN outperform other methods on all tasks, and MH-TRAN outperforms TRAN on most tasks. The average relative improvements of MH-TRAN in all tasks are 4.68% (L2-Norm), 5.78% (CSN) and 2.45% (MMoE) for Census-income dataset, 2.94% (L2-Norm), 1.68% (CSN) and 2.85% (MMoE) for FashionMnist dataset and 67.95% (L2-Norm), 55.09% (CSN), 40.50% (MMoE) and 25.29% (MRAN) for Sarcos dataset.

Comparison Between Different MTL Methods. Compare with TRAN, both CSN and MMoE are based on linear models, and MMoE model task relationships implicitly, while our method is based on attention networks to explicitly model task relationships. The fact that TRAN outperforms CSN and MMoE, indicating the advantages of our method. Besides, compared with statically modeling task relationships (MRAN), TRAN is able to dynamically control the knowledge transfer, and the fact that TRAN outperforms MRAN demonstrates the effectiveness of this key component. Although MRAN includes different types of relationships, our method still outperforms it with the relative improvements of 25.29%.

4.4 Analysis for Key Components

Task Relationships and Knowledge Transfer. We illustrate the task correlations learned by TRAN in Fig. 2.

In overall, all tasks are strongly correlated to themselves. And for different target tasks, the contributions of the other tasks vary a lot, e.g., the relationships in Sarcos dataset in Fig. 2(c). We can observe the differences between traditional methods and TRAN. In traditional methods, the task correlations are predefined

and equal for each task, however, TRAN captures their differences. In Sarcos dataset, for task 7, the contribution of task 1 is apparently less than the others. If the method relies on the pre-definition of equal task correlations, there may exist negative knowledge transfer hurting the performance. From the performance comparison, we can observe that TRAN outperforms the traditional methods with pre-definition, which demonstrates the effectiveness of adaptively capturing the task correlations.

Table 1. Performance comparison on the Census-income dataset in terms of AUC.

Method	AUC/Census-income dataset			
	Income	Marital	Education	Marital
LASSO	0.8849	0.9367	0.7695	0.9367
Bayesian	0.9267	0.8604	0.8209	0.8718
NN	0.8904	0.9387	0.8179	0.8841
Shared-bottom	0.8997	0.9379	0.8201	0.8973
L2-Norm	0.8967	0.9398	0.8174	0.9366
CSN	0.8913	0.9401	0.8219	0.8996
MMoE	0.9298	0.9523	0.8444	0.9422
TRAN	0.9412	0.9693	0.8566	**0.9792**
MH-TRAN	**0.9501**	**0.9721**	**0.8613**	0.9789

Table 2. Performance comparison on the FashionMnist dataset in terms of accuracy.

Method	Accuracy/FashionMnist	
	10-class	3-class
LASSO	0.8691	0.8879
Bayesian	0.8698	0.8981
NN	0.9032	0.9103
Shared-bottom	0.9044	0.9182
L2-Norm	0.9031	0.9173
CSN	0.9107	0.9324
MMoE	0.9079	0.9142
TRAN	0.9180	0.9497
MH-TRAN	**0.9207**	**0.9533**

For Census-income dataset, we have two multi-task learning problems, and marital task appears in both group I and II accompanied with different tasks. As the performance comparison in Table 1, marital task of TRAN in group II

performs better than the one in group I. And from the illustration, we can observe that the task correlations in group II are stronger than the correlations in group I. This indicates there are more positive knowledge transfer in group II, which contributes to the improved performance.

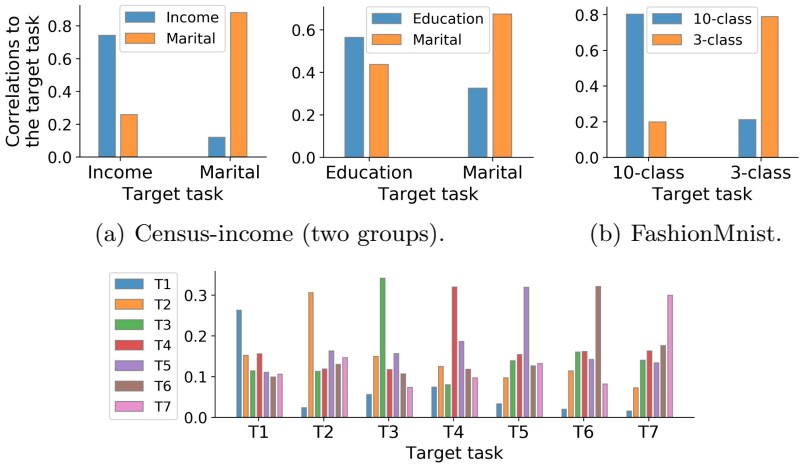
In order to verify our observation, we assess the practical strengths of task relationships in group I and II, because in general, stronger task correlations imply that there are more positive knowledge transfer. According to the works of [13], the Pearson correlations of the labels of different tasks can be used as the quantitative indicator of task relationships, because the Pearson correlations of labels are positively correlated to the strength of task relationships. The Pearson correlation in group I is 0.1784, and the one in group II is 0.2396. This indicates that there is supposed to be more positive knowledge transfer in group II, corresponding to our observation. This demonstrates that TRAN does capture the practical task correlations and control the positive knowledge transfer to help improve the performance.

Dynamically Control the Knowledge Transfer. The task relationships are not fixed, but vary slightly from different samples. We aim to dynamically capture the task relationships from different samples using TRAN.

We randomly select 8 samples from the testing samples of Sacros dataset, and provide an illustration of their dynamically learned task correlations in Fig. 3. From the correlations, we can observe that there is a slight variety in the task relationships in different samples. This demonstrate that TRAN does capture the dynamic task relationships, and the performance comparison in Table 1, 2 and 3 indicates TRAN controls the knowledge transfer to improve the performance using the dynamically learned relationships.

Table 3. Performance comparison on the Sarcos dataset in terms of RMSE.

Method	RMSE/Sarcos dataset						
	T_1	T_2	T_3	T_4	T_5	T_6	T_7
LASSO	5.848	5.159	3.153	3.501	0.410	0.952	0.733
Bayesian	5.576	4.763	3.014	3.119	0.366	0.910	0.661
NN	5.534	4.936	3.002	3.375	0.393	0.919	0.704
S-bottom	4.476	4.396	2.172	3.184	0.396	0.873	0.724
L2-Norm	4.773	4.507	2.255	3.254	0.382	1.110	0.721
CSN	3.774	3.456	1.749	1.885	0.315	0.553	0.403
MMoE	3.094	2.329	1.328	1.335	0.284	0.431	0.358
MRAN	2.879	1.895	1.041	0.829	0.154	0.261	0.236
TRAN	1.955	1.388	0.833	0.798	0.132	**0.242**	0.214
MH-TRAN	**1.937**	**1.371**	**0.793**	**0.771**	**0.122**	0.245	**0.212**



(a) Census-income (two groups). (b) FashionMnist.

(c) Sarcos dataset.

Fig. 2. Illustration of the task correlations learned by TRAN.

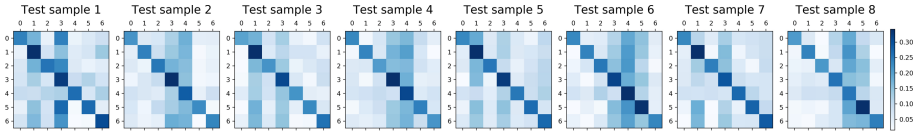


Fig. 3. Illustration of the dynamic task relationships on the Sarcos dataset. We randomly select 8 samples from the testing dataset and visualize their attention correlation matrices.

5 Conclusion

In this paper, we propose Task Relation Attention Networks to adaptively capture the task relationships, replacing the pre-defined ones in traditional MTL methods. Based on the learned relationships, the positive and negative knowledge transfer can be dynamically balanced in different samples. As a result, a better task-specific representation is obtained and leads to improved performance. In addition, the learned correlation matrix presents the dynamic transfer pattern, making the MTL method more explicable. To evaluate its performance, we conduct experiments on various datasets, including regression and classification tasks. Both classical and state-of-the-art MTL methods are employed to provide benchmarks. The experimental results and analyses demonstrate the effectiveness of our method, and its advantages over other methods.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (GrantNos.: 61673025, 61375119), and the National Key R&D Program of China (Grant Nos.: 2018AAA0100300, 2018AAA0102301), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China (Grant No. 2015CB352302).

References

1. Allamanis, M., Peng, H., Sutton, C.: A convolutional attention network for extreme summarization of source code. In: International Conference on Machine Learning, pp. 2091–2100 (2016)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
3. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
4. Ciliberto, C., Rudi, A., Rosasco, L., Pontil, M.: Consistent multitask learning with nonlinear output relations. In: Advances in Neural Information Processing Systems, pp. 1986–1996 (2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Duong, L., Cohn, T., Bird, S., Cook, P.: Low resource dependency parsing: cross-lingual parameter sharing in a neural network parser. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 2: Short Papers. vol. 2, pp. 845–850 (2015)
7. Hashimoto, K., Xiong, C., Tsuruoka, Y., Socher, R.: A joint many-task model: growing a neural network for multiple NLP tasks. arXiv preprint arXiv:1611.01587 (2016)
8. Kaiser, L., et al.: One model to learn them all. arXiv preprint arXiv:1706.05137 (2017)
9. Li, Y., Fu, K., Wang, Z., Shahabi, C., Ye, J., Liu, Y.: Multi-task representation learning for travel time estimation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1695–1704. ACM (2018)
10. Lohit, S., Wang, Q., Turaga, P.: Temporal transformer networks: joint learning of invariant and discriminative time warping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12426–12435 (2019)
11. Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., Feris, R.: Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5334–5343 (2017)
12. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
13. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1930–1939. ACM (2018)
14. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3994–4003 (2016)

15. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
16. Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, pp. 242–264. IGI Global (2010)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
18. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 vol. 1, no. 2 (2017)
19. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
20. Yang, Y., Hospedales, T.: Deep multi-task representation learning: a tensor factorisation approach. arXiv preprint arXiv:1605.06391 (2016)
21. Zhao, J., Du, B., Sun, L., Zhuang, F., Lv, W., Xiong, H.: Multiple relational attention network for multi-task learning. In: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (eds.) Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, 4–8 August 2019, pp. 1123–1131. ACM (2019). https://doi.org/10.1145/3292500.3330861