Learning Multi-Task Sparse Representation Based on Fisher Information

Yayu Zhang[1,3], Yuhua Qian[1*], Guoshuai Ma[2], Keyin Zheng[1], Guoqing Liu[1,3], Qingfu Zhang[3, 4]

[1] Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, China
[2] School of Computer Science and Technology, North University of China, Taiyuan, Shanxi, 030051, China.
[3] Department of Computer Science, City University of Hong Kong, Hong Kong, China
[4] The City University of Hong Kong Shenzhen Research Institute, Shenzhen, China
{zhang_ yayu93, jinchengqyh, maguoshuaixy}@126.com, {zhengkeyin1221, guoqingl1001}@163.com
qingfu.zhang@cityu.edu.hk

Abstract

Multi-task learning deals with multiple related tasks simultaneously by sharing knowledge. In a typical deep multi-task learning model, all tasks use the same feature space and share the latent knowledge. If the tasks are weakly correlated or some features are negatively correlated, sharing all knowledge often leads to negative knowledge transfer among. To overcome this issue, this paper proposes a Fisher sparse multi-task learning method. It can obtain a sparse sharing representation for each task. In such a way, tasks share features on a sparse subspace. Our method can ensure that the knowledge transferred among tasks is beneficial. Specifically, we first propose a sparse deep multi-task learning model, and then introduce Fisher sparse module into traditional deep multi-task learning to learn the sparse variables of task. By alternately updating the neural network parameters and sparse variables, a sparse sharing representation can be learned for each task. In addition, in order to reduce the computational overhead, an heuristic method is used to estimate the Fisher information of neural network parameters. Experimental results show that, comparing with other methods, our proposed method can improve the performance for all tasks, and has high sparsity in multi-task learning.

Introduction

Human possess an extraordinary capacity for inducing and transferring knowledge, enabling us to generalize and apply existing knowledge to new situations. However, conventional machine learning models have not adapted well to challenges such as multi-task, multi-scenario and still lack sufficient generalization ability. Taking inspiration from the knowledge transfer mechanism observed in human cognition, the concept of transfer learning (Weiss, Khoshgoftaar, and Wang 2016) has emerged as a novel learning paradigm. This paradigm facilitates the transfer of knowledge among tasks, resulting in expedited and effective model training, often yielding superior performance outcomes. Multi-task learning(MTL) (Caruana 1997) as a special case of transfer learning, which combines multiple related tasks to train the model together and improve the performance of all tasks. During the training process, knowledge is adeptly transferred across tasks, forging a dynamic learning synergy.

*The corresponding author.

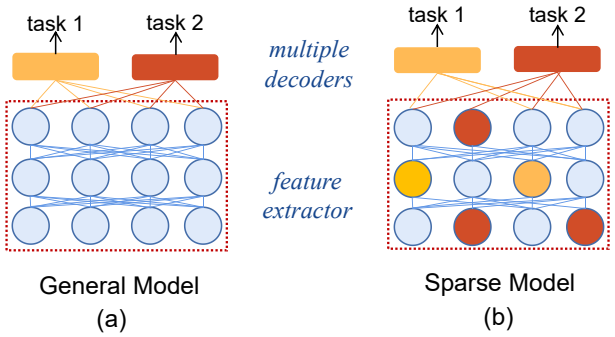


Figure 1: The two modeling approaches of multi-task learning. The layers inside the red dotted box are task-sharing layers. It can also be regarded as a feature extractor to obtain the feature representation shared by tasks. The layers at the top of the model are task-specific layers. It decodes the sharing feature to the task-specific feature space. The blue block represents the tasks sharing parameters; The yellow and red blocks are the task-specific parameters.

As shown in Fig. 1(a), the architecture of the general deep MTL model comprises a feature extractor and multiple decoders(Crawshaw 2020). The feature extractor learns the nonlinear and high-dimensional representation of multiple tasks. Subsequently, the decoder maps the output of the encoder to a task-specific representation space. This modeling approach is assumed that all tasks share identical dependencies and utilize a shared feature space. Nevertheless, this assumption often falls short in real-world scenarios, where correlations between latent features among tasks are sometimes limited. The inclusion of irrelevant shared information during multi-task training can induce negative knowledge transfer among tasks (Sun et al. 2020a). Thus, there arises a necessity to explore novel modeling approaches that can effectively address this aforementioned challenge.

Sparse modeling is assumed that all tasks share knowledge within a sparse subspace. This assumption can enhance the robustness and generalization capabilities of a multi-task learner by eliminating negative correlations and redundant features during training. Moreover, the acquired sparse model effectively reduces storage demands and can

be conveniently accommodated within embedded devices. Consequently, the sparse modeling approach has garnered increasing interest among deep multi-task learning (DMTL) researchers.

In the context of DMTL, the notion of sparsity often guides the identification of parameters that warrant sharing among tasks. Precisely, the parameters within the feature extractor are allocated across distinct tasks, as exemplified in Fig. 1(b). Knowledge transfer between tasks is confined solely to these shared parameters. However, most prevailing methodologies distribute the parameters in the feature extractor randomly. In 2020, Sun et.al. (Sun et al. 2020a) introduced the concept of sparse sharing in multi-task learning, whereby a set of masks for each task is obtained randomly, and subsequently, the optimal mask for each task is chosen for multi-task training. Another technique, known as task routing (Strezoski, Noord, and Worring 2019), procures the sparse optimization path for a task through random exclusion of the shared layer filter's output. Pascal et al. (2021) (Pascal et al. 2021) employ the maximum roaming method, introducing random variations in parameter allocation via dropout. These randomized approaches at times yield unstable, uncertain, and unexplainable experimental outcomes. The ESSR method(Zhang et al. 2023) acquires task sparse representations through evolutionary processes. However, its feasibility is restricted for extensive computer vision tasks due to the demand for substantial computational resources.

Thus, in this paper, a statistical measurement approach is adopted to derive the sparse sharing representation of tasks. This methodology surpasses the limitations of both random and evolutionary methods, providing a balanced and controlled means to enhance the efficiency and effectiveness of multi-task learning. The proposed method is called Fisher sparse multi-task learning, abbreviated as FSMTL. This approach integrates a Fisher information module into DMTL, facilitating the estimation of shared parameters' significance for each task. The task will select the K parameters with the highest Fisher information. During training, each task only optimizes these chosen parameters. The knowledge transfer among tasks takes place through the parameters jointly optimized by these tasks. In particular, we begin by formalizing the sparse model of DMTL, introducing a binary sparse variable set denoted as S within the general model. Subsequently, we formulate a bi-level optimization problem encompassing two types of variables: the network parameters and the sparse variables set. Then the neural network parameters and sparse variables are updated alternately by fixed variables. Notably, the sparse variable of task is learned based on the Fisher information of shared parameters. Ultimately, the proposed method learns a sparse representation for each task. And every task obtains a smaller inference model.

The proposed method presents several key contributions and advantages, outlined as follows:

- The sparse DMTL model is formalized, and a novel method for sparse multi-task learning is proposed. It determines which knowledge should be shared among tasks by adding a Fisher sparse module.
- The Fisher information is incorporated as a priori to attain the sparse representation of each task, with time efficiency being optimized through the utilization of an empirical estimation method instead of the conventional approach for Fisher information estimation.
- The efficacy of the proposed method is verified across three distinct types of multi-task datasets. The results show that the proposed method improves the performance of each task, and has better sparsity than related methods.

The remainder of this paper is organized as follows. Section 2 surveys different deep multi-task learning methods. Section 3 proposes the concept of multi-task sparse representation and develops a novel sparse multi-task learning method. Section 4 presents the performance evaluation of the proposed method against related methods. Finally, we draw conclusions in Section 5.

Related Work

In this section, we summarize the related works on solving task interference. We roughly divide them into the following four categories.

Multi-Task Network Architecture Learning. These method obtains the shared Architecture of the task by dividing the model structure or adding task modules. Cross-stitch networks(Misra et al. 2016) model shared representations by adding a cross-stitch unit. Single-tasking multiple tasks method decode the tasks' common representation by adding a squeezeand-excitation (SE) modulation between encoder and decoder. AdaShare(Sun et al. 2020b) is to learn which layers to excute for a given task in the multi-task network. And gumbel-softmax sampling is introduced to resolve this non-differentiability and enable direct optimization of the discrete policy using back-propagation. Stochastic filter groups method(Bragman et al. 2019) assigns the convolution kernels to task-specific and shared groups. Evolutionary architecture search(EAS)(Liang, Meyerson, and Miikkulainen 2018) develops an automated, flexible approach for evolving architectures of deep multitask networks.

Adaptive Loss Weighting. GradNorm method(Chen et al. 2018) automatically balances training in deep multitask models by dynamically tuning gradient magnitudes. Uncertainty weigh losses method(Kendall, Gal, and Cipolla 2018) proposes a principled approach to multi-task deep learning which weighs multiple loss functions by considering the homoscedastic uncertainty of each task. Dynamic weight average method(Liu, Johns, and Davison 2019) requires the numerical task loss, and therefore its implementation is far simpler. Just Pick a Sign(Chen et al. 2020) method optimizing deep multitask models with gradient sign dropout. Lin et.al. (Lin et al. 2021) propose the random weighting method, where an MTL model is trained with random loss/-gradient weights sampled from a distribution.

Trade-off Gradient Direction. The concept of multi-objective multi-task learning (MOMTL)(Sener and Koltun

2018) was first proposed by Sener and Koltum in 2018. Different from general multi-task learning problems, it mainly focuses on multi-task learning problems with conflicts. The optimization objective is to find the trade-off solution among tasks. PCGrad (Yu et al. 2020) mitigating gradient interference by altering the gradients directly. CAGrad(Liu et al. 2021a) minimizes the average loss function, while leveraging the worst local improvement of individual tasks to regularize the algorithm trajectory. Impartial multi-task learning(IMTL)(Liu et al. 2021b) is proposed to balance the weight of gradient and loss in multi-task learning. Stochastic multi-objective gradient correction (MoCo)(Fernando et al. 2023) guarantees convergence without increasing the batch size even in the nonconvex setting.

Multi-Objective Multi-Task Learning. The above methods only obtain a solution instead of a Pareto set when faced with task gradient conflicts. The Pareto multi-task learning(Lin et al. 2019) decomposes the MTL problem into multiple sub-problems with constraints to obtain a Pareto set under multiple preferences. Controllable Pareto multi-task(Lin et al. 2020) formulates the MTL as a preference-conditioned multi objective optimization problem, with a parametric mapping from preferences to the corresponding trade-off solutions. Exact Pareto optimal (EPO)(Mahapatra and Rajan 2020) develops the first gradient-based multi-objective MTL algorithm to find a preference-specific Pareto optimal solution. Continuous Pareto MTL(Ma, Du, and Matusik 2020) presents a novel and efficient method that generates locally continuous Pareto sets and Pareto fronts.

Methodology

Suppose that $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_t \in \mathcal{T}$ are t supervised learning tasks. Each task is associated with a set of training data $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t} \subset R^d \times R$, where N_t is the number of samples. When $t_1 \neq t_2$, $x_i^{t_1} = x_i^{t_2}$, it means that tasks share the same training data. MTL aims to learning a mapping $f_t : \mathcal{X}_t \to \mathcal{Y}_t$ for each task simultaneously. For general DMTL model, the mapping can divided into two parts: (1) a feature extractor $\Phi : \mathcal{X}_t \to \mathcal{Z}$, with parameters θ^{sh}, which obtain the sharing feature of tasks; (2) multiple decoders (e.g., regressor or classifier) $g_t : \mathcal{Z} \to \mathcal{Y}_t$, parameterized by θ_t, which map sharing features into different tasks. The prediction label of task $\mathcal{T}_t$ can be written in the following form

$$f_t(x) = g_t(\Phi(x_t, \theta^{sh}); \theta^t), \tag{1}$$

where $z_t = \Phi(x_t, \theta^{sh})$ is the latent representation of the input x_t and the feature mapping function Φ is learned by jointly all tasks. Information sharing among tasks is realized through θ^{sh}. The task-specific loss functions is $L_t(\theta^{sh}, \theta^t) = \frac{1}{N_t} \sum_i^{N_t} l(\hat{f}_i^t, y_i^t)$. The objective function is described as empirical risk minimization formulation:

$$\min_{\substack{\theta^{sh} \\ \theta^1, ... \theta^T}} \sum_{t=1}^{T} w_t L_t(\theta^{sh}, \theta^t), \tag{2}$$

w_t is the weight of the t-th task.

The conventional MTL paradigm mentioned above assumes that all tasks employ an identical feature space, implying uniform relevance among tasks and equal information sharing. However, this assumption is evidently unrealistic. As a remedy, we introduce the Fisher Sparse Multi-Task Learning (FSMTL) approach, designed to optimize the effectiveness of knowledge transfer among tasks.

Sparse Deep Multi-Task Learning

In the FSMTL, we assume that the tasks share information using a chosen subset of features. It means that original feature extractor Φ encompasses numerous parameters that are equal to zero. The mode of information sharing has changed from full sharing to sparse sharing. A primary objective of FSMTL is the identification of these sparse subspaces. This is achieved through the introduction of a sparse variable set, denoted as S. Furthermore, the feature mapping from input to output is restructured as follows:

$$f_t(x) = g\left(\Phi(x, S_t \odot \theta^{sh}); \theta^t\right), t = 1, ..., T, \tag{3}$$

where $\odot$ denotes element-wise multiplication. $S = \{S_1, S_2, ..., S_T\}$ is a set of task sparse variables. S_t is the sparse encoding of task t, and it is a tensor filled with 0 and 1. And the dimension of S_t is the same as θ^{sh}. If the sparse variable is given, the task's sparse representation can be derived. Sparse representations enable information sharing among tasks through distinct parameters. Next, we embed the sparse variables into neural network, and the details are described as follows.

Suppose that feature extractor Φ of neural network has P parameters, that is, $\theta^{sh} = \{\theta_1, \theta_2, ..., \theta_P\}$. And the sparse variable of task $\mathcal{T}_t$ is recorded as $S_t = \{s_1, s_2, ..., s_P\}$. Then, the gradient is calculated as follows:

$$g_{\theta_p}^t = \begin{cases} \nabla_{\theta_p} L_t, & s_p = 1 \\ 0, & s_p = 0 \end{cases}, \tag{4}$$

If $s_p = 1$, the p-th parameter of θ^{sh} is optimized by task $\mathcal{T}_t$. Conversely, if the elements of $s_p = 0$, it corresponding parameter is not optimized. By the above operation, the task is encoded within a sparse subspace. The objective function for MTL is reformulated as:

$$\min_S \min_\Theta L\left(\theta^{sh}, \theta^t, S; \mathcal{D}\right), \tag{5}$$

$\Theta = \left\{\theta^{sh}, \theta^t\right\}_{t=1}^T$. Eq. (5) is a bi-level optimization problem. Both the neural network parameters Θ and sparse variable set S are need to be optimized. In the paper, two kinds of parameters are alternately learned by fixed variable.

With the sparse variable set is provided, each task is associated with a unique optimization path. The objective at the inner level is jointly optimized through alternating iterations across all tasks. However, optimizing high-dimensional discrete task variable s_t poses computational feasibility challenges. So, the Fisher sparse module is introduced in this study to aid in learning them. It can enhance the generalization performance of a learner (Wang et al. 2023).

Fisher Sparse Module

To avoid the challenges posed by discrete optimization, the process of acquiring sparse variables can be reframed as a parameter selection problem. It tries to seek the shared parameters within the feature extractor that offer the greatest advantage to a given task. So, in the paper, the KL-divergence is used to measure the impact of shared parameters on tasks' prediction performance. The details are described as follows.

The neural network is recognized as a discriminant model (Bishop and Nasrabadi 2006). It achieves classification or regression tasks by modeling the conditional probability distribution $P(y|x, \theta)$. Here, y denotes the target variable, x denotes the input features, and θ represents the network parameters. Through introducing a slight perturbation δ to model's parameter θ, the KL-divergence, $KL(p_\theta(y|x)||p_{\theta+\delta}(y|x))$, is used to quantify the disparity between the original distribution p_θ and the perturbation distribution $p_{\theta+\delta}$. When use in DMTL, this measurement can serve as an indicator of the significance of the shared parameters θ^{sh} in determining the model's prediction performance for a specific task. Therefore, we use this latent relationship between sharing parameters and model predictive performance to obtain the sparse variable of the task.

It is well known in the field of natural gradient, assuming $\delta \to 0$, the KL-divergence can be approximate by its second order Taylor series(Martens 2020):

$$E_x \left[KL(p_\theta(y|x)||p_{\theta+\delta}(y|x))\right] = \delta^T F_\theta \delta + O(\delta^3), \quad (6)$$

where $O(\delta^3)$ denotes terms of order 3 or higher in the entries of δ. The F_θ represents the Fisher information matrix of $p_\theta(y|x)$ with respect to θ, and it is given by:

$$F_\theta = E_{Q_x}[E_{P_{y|x}}[\nabla_\theta log p_\theta(y|x) \nabla_\theta log p_\theta(y|x)^T]], \quad (7)$$

Q_x represents the target distribution of input vector x with density function $q(x)$. Formula (6) constructs the connection between KL-divergence and Fisher information. It shows that the difference between the two distributions is positively linked to Fisher information. The model exhibits heightened sensitivity towards parameters possessing greater Fisher information. In other words, the parameters with larger Fisher information carry higher significance for the model's performance.

Hence, we leverage above association to derive task's sparse variables in multi-task learning. In the context of general DMTL, the conditional distribution of task t is extended and redefined as $P(y_t|x_t; \theta^{sh}, \theta^t)$, where y_t represents the output for a specific task, x_t represents the corresponding input, and θ^{sh} and θ^t represent the shared and task-specific parameters, respectively. It density function is denoted as $p_{\Theta_t}(y_t|x_t)$, $\Theta_t = \left\{\theta^{sh}, \theta^t\right\}$. The Fisher information matrix of task $\mathcal{T}_t$ can be described as follows:

$$F_{\Theta_t} = E_{Q_{x_t}}[E_{P_{y_t|x_t}}[(\nabla_{\Theta_t} log p_{\Theta_t})(\nabla_{\Theta_t} log p_{\Theta_t})^T]]. \quad (8)$$

In practice the real $q(x_t)$ and $p(y_t|x_t)$ is unknown. Fisher information matrix is often estimated using its empirical version. The Fisher information of the multi-task learning model on task $\mathcal{T}_t$ can be calculated by the following formula.

$$\hat{F_{\Theta_t}} = \frac{1}{N_t} \sum_{x \in D_t} \sum_{c=1}^{C} p(y=c) * [\nabla_{\Theta_t} log p(y=c|x_{ti}; \theta)]^2, \quad (9)$$

where C refers to the number of categories of tasks. However, due to the large number of parameters in a neural network, it is difficult to directly calculate the Fisher information matrix. In order to improve computational efficiency, a simplified estimation approach(Sung, Nair, and Raffel 2021) is employed in this paper to approximate the Fisher information of the parameter θ.

$$\hat{F}_{\Theta_t} = \frac{1}{N_t} \sum_{x \in D_t} (\nabla_\theta log p(y=bc|x_{ti}; \Theta_t))^2 \quad (10)$$

$\{x_{ti}\}_{i=1}^{N_t}$ are N_t samples from training data D_t. bc is the prediction label of learned model. Furthermore, the Eq: (10) is faster to compute than the standard Fisher as long as more than one sample is used to approximate the expectation $E_{p_{y|x_i}}$.

In the paper, the K parameters with the largest Fisher information in the feature extractor Φ are selected as task parameters. Let θ_p be the critical value, the mapping of a parameter to sparse variables will be established:

$$s_i = \begin{cases} 1, & F_{\theta_i} \geq F_{\theta_p} \\ 0, & other \end{cases}, i = 1, \cdots, P. \quad (11)$$

In this way, the task will acquire a unique optimized path. During training, the task only updates parameters related to itself. The parameters selected by multiple tasks will be optimized by them together. The knowledge transfer among tasks occurs only for the co-optimized parameters. The Fig. A1 in the Appendix shows the optimization path of the task on the i-th epoch. By continuous alternate optimization of neural network parameters and sparse variables, the proposed method will find an optimal inference path for each task.

Algorithm 1: FSMTL Algorithm Framework

Input: $\mathcal{D} = \{D_t\}_{i=1}^T$: Multi-task dataset
Net: Base network
N_{epoch}: maximum iteration number
Output: Optimal network and sparse variable set S_D

```
Warm up Net
for i = 0 to N_epoch do
    if Satisfy the update condition then
        Updating S based on θ^sh
        Updating network parameters
    end if
end for
return Θ and S.
```

Optimization and Implementation Detail

During training, the procedure handles two types of parameters: one is the parameter of neural network, the other is

Algorithm 2: Updating Sparse Variable Set S

Input: T: The number of tasks
k: mask sparse level
$\mathcal{D} = \{D_t\}_{i=1}^{T}$: Multi-task dataset
Net: A base network
Output: S: Sparse Variable Set

1: **for** t = 1 to T **do**
2: Sample N examples D_t^N from D_t
3: Compute $g_\theta = \nabla_\theta L_t(\theta^{sh}; D_t^N)$
4: Compute g_θ^2 or $|g_\theta|$
5: Sort sharing layer parameters $\left\{g_{\theta_i}^2\right\}_{i=1}^{P}$
6: Obtain sparse variable S_t according to Eq. (11)
7: **end for**
8: **return** Sparse variable set S

sparse variable of tasks. They are updated by fixed variables. Pseudo-code are shown in algorithm 1 to algorithm 2.

Model parameter optimization. When updating Θ with fixed S_D, the optimization problem can be written as:

$$\min_{\theta_t^{sh},\theta_t} \frac{1}{N_t} \sum_{i=1}^{n} L_t(g\left(\Phi(x_i^t, S_t \odot \theta^{sh}); \theta^t\right), t = 1, ..., T \tag{12}$$

In homogeneous features MTL, the multi-task datasets are fed to the network, and the tasks' objective are optimized in turns. **In heterogeneous features MTL**, all tasks come from different feature space, that is $x_{t1} \neq x_{t_2}$. The tasks examples in min-batch are fed to network in order. And then the average of all tasks gradients is taken as the final gradient of network parameter. The task gradient is obtained by back propagation of its loss function. The details can refer to Algorithm 1 in the Appendix.

Sparse variable updating. When the network parameter is fixed, the second objective is to estimate Fisher information of feature extractor parameters for each task. Specifically, for task $\mathcal{T}_t$, we sample N examples D_t^N from D_t and input them to the network of current state. Then parameter gradient $g_\theta = \nabla_\theta p(y_c|D_t^N)$ are obtained by back propagation $L_t(\theta^{sh}, \theta_t)$. The g_θ^2 is proportional to $\hat{F}_\theta$. Therefore, within the algorithm, the elements in set $\left\{g_{\theta_i}^2\right\}_{i=1}^{P}$ to assist the selection of task parameters. Once sorted, the K parameters with the largest g_θ^2 are designated as task parameters.

Experimental Studies

Datasets

This paper conducts experiments on three multi-task datasets: DKL-mnist, CelebA, and CityScapes. They include 3, 8 and 8 tasks respectively. DKL-mnist is a heterogeneous feature multi-task dataset. CelebA and CityScapes are homogeneous feature multi-task datasets. CelebA is a multi-label dataset, with each instance belonging to multiple classes. CityScapes is a computer vision dataset used to evaluate two completely heterogeneous tasks: semantic segmentation and depth estimation. The detailed description of the datasets can be found in the Appendix.

Comparative Methods

We compare our method(FS) with 11 related methods on three multi-task datasets. There are two baseline methods (STL and MTL) and nine methods that address task interference in MTL. GradNorm and MGDA-UB methods mitigate task interference by adaptive loss weighting. CAGrad (Liu et al. 2021a) and Nash-MTL (Navon et al. 2022) mitigate task interference by gradient weighting. Random Loss method(RLW) (Lin et al. 2021) method train model via random loss/gradient weights sampled from a distribution. Squeezeand-Excitation(SE) (Maninis, Radosavovic, and Kokkinos 2019), Task Routing(TR) (Strezoski, Noord, and Worring 2019), Maximum Roming(MR) (Pascal et al. 2021), Random Sparse(RS) and Fisher Sparse(FS) divide task parameters by different ways. The detailed description of the related methods can be found in the Appendix.

Performance Metrics

Four metrics Accuracy, Precision, Recall, and F-score are used to evaluate the performance of the classification task. Intersection over Union (mIoU) and Pixel Accuracy (Pix.ACC.) are used to evaluate the performance of the semantic segmentation task. Average absolute (Abs.Err.) and relative error (Rel.Err.) are used to evaluate the performance of the depth estimation task.

Comparing with Related Methods

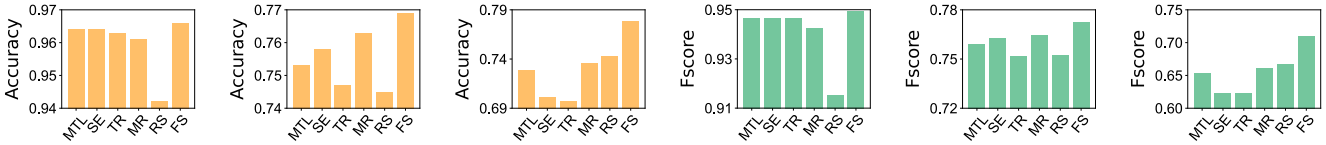
This section presents a comparative analysis of the performance of FS with related works. The experimental results are presented in Table 1 and Table 2. $\#N$ is the number of tasks. The p denotes the "keep ratio" of parameters for RS and FS. For TR and MR, the p denotes the "keep ratio" of each layer of filters' output. Each entry in the table represents the mean and standard deviation of three experiments conducted for all tasks(presented as "mean $\pm$ std"). The optimal results are highlighted in boldface font. We compare the performance of all algorithms on each task, and present the results in Fig. 2 and Fig. 3. In addition, "Statistical Significance Analysis" and "Ablation Study" can be found in the Appendix. Considering the time cost, all analytical experiments were conducted on the smaller datasets DKl-mnist and CleabA. Based on the experimental results, some findings are obtained:

(1) **The approach of sparse sharing can alleviate task interference.** The experimental results on the DKL-mnist and Cityscape datasets demonstrate that the performance of MTL is worse than STL. This suggests that task transfer in multi-task learning may contain destructive or negatively correlated components. Despite the incapacity of other compared methods to address this issue, our method also achieves good learning performance. In particular, the FS method can effectively address the issue when applied to the Cityscape dataset, using only 30% of the parameters in each iteration.

(2) **The FS can improve the performance of each task.** In Figure 2 and Figure 3, we compare the accuracy and F-score of all algorithms on each task for datasets DKL-mnist and CelebA, respectively. Results show that FS out-

Datasets	$\#N$	Model	p	Multi-Attribute Classification↑				Avg. Rank
				Accuracy	Precision	Recall	F-score	
DKL-mnist	3	STL		0.877 ± 0.002	0.807 ± 0.006	0.807 ± 0.005	0.797 ± 0.006	-
		MTL	100%	0.869 ± 0.002	0.793 ± 0.001	0.796 ± 0.002	0.784 ± 0.002	3
		CAGrad	100%	0.866 ± 0.006	0.791 ± 0 007	0.793 ± 0.006	0.780 ± 0.007	3.875
		RGW	100%	0.870 ± 0.004	0.796 ± 0.004	0.799 ± 0.004	0.786 ± 0.004	2
		Nash-MTL	100%	0.866 ± 0.001	0.788 ± 0.002	0.791 ± 0.002	0.778 ± 0.002	4.625
		SE	-	0.859 ± 0.005	0.777 ± 0.007	0.782 ± 0.006	0.767 ± 0.007	8
		TR	90%	0.859 ± 0.003	0.781 ± 0.004	0.784 ± 0.003	0.771 ± 0.003	6.5
		MR	90%	0.859 ± 0.009	0.780 ± 0.014	0.784 ± 0.012	0.770 ± 0.013	7.75
		RS	90%	0.851 ± 0.009	0.780 ± 0.010	0.787 ± 0.010	0.771 ± 0.009	6.75
		FS	90%	**0.880 ± 0.001**	**0.811 ± 0.006**	**0.813 ± 0.007**	**0.801 ± 0.007**	1
CelebA	8	STL		0.879 ± 0.002	0.629 ± 0.004	0.620 ± 0.005	0.624 ± 0.003	-
		MTL	100%	0.894 ± 0.001	0.689 ± 0.006	0.597 ± 0.003	0.632 ± 0.001	10.5
		GradNorm	100%	0.895 ± 0.004	0.693 ± 0.003	0.599 ± 0.008	0.634 ± 0.002	10
		MGDA-UB	100%	0.898 ± 0.0003	0.706 ± 0.001	0.595 ± 0.003	0.636 ± 0.002	7.5
		CAGrad	100%	0.897 ± 0.001	0.704 ± 0.004	0.597 ± 0.001	0.638 ± 0.0001	7.25
		RLW	100%	0.896 ± 0.002	0.697 ± 0.033	0.596 ± 0.006	0.635 ± 0.001	9.75
		RGW	100%	0.893 ± 0.003	0.688 ± 0.011	0.603 ± 0.004	0.635 ± 0.002	9.75
		Nash-MTL	100%	0.896 ± 0.002	0.700 ± 0.009	0.602 ± 0.004	0.639 ± 0.003	7.25
		SE	-	0.899 ± 0.002	0.702 ± 0.003	0.626 ± 0.005	0.659 ± 0.002	4
		TR	90%	0.901 ± 0.002	0.718 ± 0.012	0.611 ± 0.009	0.651 ± 0.004	4
		MR	90%	0.899 ± 0.002	0.704 ± 0.009	0.622 ± 0.003	0.654 ± 0.001	4.25
		RS	90%	0.904 ± 0.001	0.728 ± 0.004	0.613 ± 0.002	0.654 ± 0.001	2.75
		FS	**10%**	**0.910 ± 0.0003**	**0.753 ± 0.002**	**0.637 ± 0.003**	**0.681 ± 0.0004**	1

Table 1: Comparative study of related algorithms on DKL-mnist and CelebA dataset



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 1 (e) Task 2 (f) Task 3

Figure 2: Comparison of the accuracy and F-score of all algorithms across each task of DKL-mnist dataset. (a)-(c) and (d)-(f) present a comparison of six algorithms' Accuracy and F-SCORE across every task, respectively.

performs the comparative algorithms on all tasks, especially for the more challenging optimization tasks where FS method demonstrated a more evident superiority. For example, task 3 on the DKL-mnist dataset and task 5 on the CelebA dataset.

(3) **The FS uses fewer parameters and obtains better performance compared with related methods.** Table 1 and 2 presents the optimum p values obtained from the results. The TR, MR, and RS methods show optimal performance at $p = 90\%$, whereas the FS method shows good performance for CelebA at $p = 10\%$, utilizing parameters of only 10%. Although the FS achieves optimal performance at $p = 90\%$ for the DKL-mnist, it also yields an improvement of 30% in F-score. Experiments on Cityscapes also demonstrate the significant advantage of the proposed method.

(4) **The FS has more obvious advantages in homogeneous feature tasks.** In comparison, the FS method demonstrates significantly better performance on the homogeneous feature multi-task learning datasets CelebA and Cityscapes, compared to the heterogeneous feature multi-task dataset DKL-minst. This result suggests that the proposed method is more prone to interference in heterogeneous tasks owing to differences in tasks' data distributions.

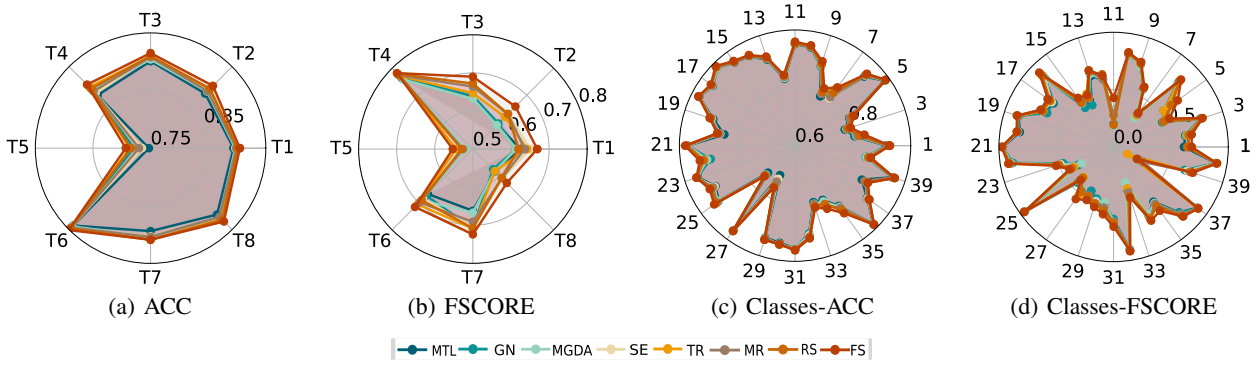
(5) **The performance of the simple FS method closely that of the EFS method.** We also compared the performance of the FS and EFS methods on both DKL-mnist and CelebA datasets. The results are presented in Fig. 4. The results show that the FS and EFS are consistent on the same dataset with the increase of "keep ratio". Sometimes, the performance of the FS method is even better than that of the EFS method, such as when the keep ratio is 0.1 on CelebA.

Conclusion

To overcome the negative knowledge transfer existing in multi-task learning, this paper proposes a sparse sharing method based on the Fisher information of neural network

Datasets	$\#N$	Model	p	Segmentation↑		Depth estimation↓		Avg. Rank
				mIoU	Pix. Acc.	Abs. Err.	Rel. Err.	
Cityscape	8	STL		58.57 ± 0.49	97.46 ± 0.03	0.0141 ± 0.0002	22.59 ± 1.15	-
		MTL	100%	56.57 ± 0.22	97.36 ± 0.02	0.0170 ± 0.0006	43.99 ± 5.53	6.75
		GradNorm	100%	56.77 ± 0.08	97.37 ± 0.02	0.0199 ± 0.0004	68.13 ± 4.48	5.875
		MGDA-UB	100%	56.19 ± 0.24	97.33 ± 0.01	0.0130 ± 0.0001	**25.09 ± 0.28**	3.75
		SE	-	55.45 ± 1.03	97.24 ± 0.10	0.0160 ± 0.0006	35.72 ± 1.62	6.625
		TR	60%	56.52 ± 0.41	97.24 ± 0.04	0.0155 ± 0.0003	31.47 ± 0.55	5.625
		MR	60%	57.93 ± 0.20	97.37 ± 0.02	0.0143 ± 0.0001	29.38 ± 1.66	3.5
		RS	30%	60.50 ± 0.19	97.53 ± 0.05	0.0143 ± 0.0003	25.87 ± 0.58	2.375
		FS	**30%**	**66.24 ± 0.34**	**98.05 ± 0.02**	**0.0125 ± 0.0001**	25.86 ± 0.647	1.25

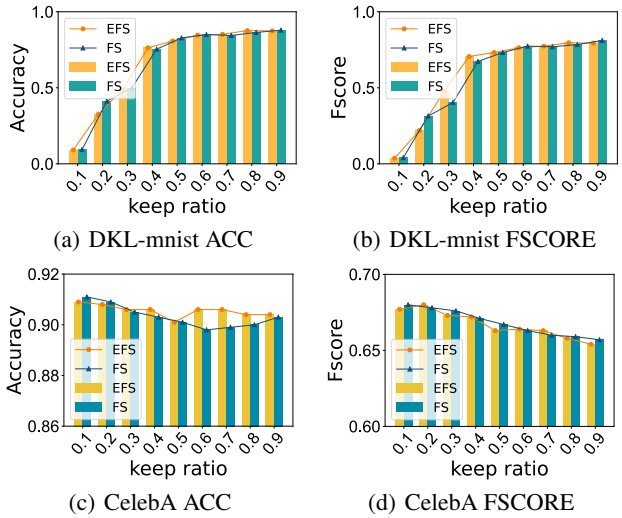
Table 2: Comparative study of related algorithms on Cityscape dataset



(a) ACC (b) FSCORE (c) Classes-ACC (d) Classes-FSCORE

(e) legend

Figure 3: Comparison of the accuracy and F-scores of all algorithms across each task and each class of CelebA dataset. Panels (a) and (b) show the comparison results of each task. Panels (c) and (d) show the comparison results of each classes.



(a) DKL-mnist ACC (b) DKL-mnist FSCORE

(c) CelebA ACC (d) CelebA FSCORE

Figure 4: Comparison of the performance of FS and EFS metod on DK-mnist and CelebA datasets.

parameters. It assumes that tasks share knowledge in a sparse subspace rather than the full feature space. Specifically, we introduce the sparse variable S in the traditional deep learning model. The sparse variable gives the optimization path of the task, and it determines which parameters should be shared between tasks. By alternately optimizing neural network parameters and sparse variables, sparse sharing representation is finally optimized for each task.

Experiments conducted on both homogeneous and heterogeneous multi-task datasets have unequivocally demonstrated the effective alleviation of negative knowledge transfer between tasks by the proposed method. Furthermore, a comparison with related methods has revealed the clear advantages of the proposed approach. It has not only improved the average task performance but has also enhanced the performance on each task with greater sparsity.

Multi-task sparse modeling is a method to alleviate task conflicts, which aims to resolve which knowledge should be shared between tasks in multi-task learning and attain smaller inference model. This method possesses both scientific and practical significance. Moving forward, we will further develop this approach to foster systematic research and promote its comprehensive exploration.

Acknowledgments

This work was supported by National Key Research and Development Program of China (No. 2021ZD0112400), National Natural Science Foundation of China (Nos. 62136005, 62306171), the Science and Technology Major Project of Shanxi (No. 202201020101006), the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant CityU-11215622, the Key Basic Research Foundation of Shenzhen under Grant JCYJ20220818100005011

References

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Bragman, F. J.; Tanno, R.; Ourselin, S.; Alexander, D. C.; and Cardoso, J. 2019. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV-19)*, 1385–1394.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Chen, Z.; Badrinarayanan, V.; Lee, C.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning(ICML-18)*, 794–803. PMLR.

Chen, Z.; Ngiam, J.; Huang, Y.; Luong, T.; Kretzschmar, H.; Chai, Y.; and Anguelov, D. 2020. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems(NeurIPS-20)*, 33: 2039–2050.

Crawshaw, M. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Fernando, H. D.; Shen, H.; Liu, M.; Chaudhury, S.; Murugesan, K.; and Chen, T. 2023. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *The Eleventh International Conference on Learning Representations(ICLR-23)*.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR-18)*, 7482–7491.

Liang, J.; Meyerson, E.; and Miikkulainen, R. 2018. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 466–473.

Lin, B.; Ye, F.; Zhang, Y.; and Tsang, I. W.-H. 2021. Reasonable Effectiveness of Random Weighting: A Litmus Test for Multi-Task Learning. *Trans. Mach. Learn. Res.*, 2022.

Lin, X.; Yang, Z.; Zhang, Q.; and Kwong, S. 2020. Controllable Pareto multi-task learning. *arXiv preprint arXiv:2010.06313*.

Lin, X.; Zhen, H.; Li, Z.; Zhang, Q.; and Kwong, S. 2019. Pareto multi-task learning. *Advances in Neural Information Processing Systems(NeurIPS-19)*, 32: 12060–12070.

Liu, B.; Liu, X.; Jin, X.; Stone, P.; and Liu, Q. 2021a. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems(NeurIPS-21)*, 34: 18878–18890.

Liu, L.; Li, Y.; Kuang, Z.; Xue, J.; Chen, Y.; Yang, W.; Liao, Q.; and Zhang, W. 2021b. Towards impartial multi-task learning. In Proceedings of the Ninth International Conference on Learning Representations(ICLR-21).

Liu, S.; Johns, E.; and Davison, A. J. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR-19)*, 1871–1880.

Ma, P.; Du, T.; and Matusik, W. 2020. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning(ICML-20)*, 6522–6531. PMLR.

Mahapatra, D.; and Rajan, V. 2020. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning(ICML-20)*, 6597–6607. PMLR.

Maninis, K.; Radosavovic, I.; and Kokkinos, I. 2019. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1851–1860.

Martens, J. 2020. New insights and perspectives on the natural gradient method. *The Journal of Machine Learning Research*, 21(1): 5776–5851.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR-16)*, 3994–4003.

Navon, A.; Shamsian, A.; Achituve, I.; Maron, H.; Kawaguchi, K.; Chechik, G.; and Fetaya, E. 2022. Multi-Task Learning as a Bargaining Game. In *International Conference on Machine Learning*, 16428–16446. PMLR.

Pascal, L.; Michiardi, P.; Bost, X.; Huet, B.; and Zuluaga, M. 2021. Maximum Roaming Multi-Task Learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence(AAAI-21)*, volume 35, 9331–9341.

Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. *arXiv preprint arXiv:1810.04650*.

Strezoski, G.; Noord, N. v.; and Worring, M. 2019. Many task learning with task routing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV-19)*, 1375–1384.

Sun, T.; Shao, Y.; Li, X.; Liu, P.; Yan, H.; Qiu, X.; and Huang, X. 2020a. Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence(AAAI-20)*, volume 34, 8936–8943.

Sun, X.; Panda, R.; Feris, R.; and Saenko, K. 2020b. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems(NeurIPS-20)*, 33.

Sung, Y.-L.; Nair, V.; and Raffel, C. A. 2021. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems(NeurIPS-21)*, 34: 24193–24205.

Wang, J.; Li, F.; Li, J.; Hou, C.; Qian, Y.; and Liang, J. 2023. RSS-Bagging: Improving Generalization Through the Fisher Information of Training Data. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Weiss, K.; Khoshgoftaar, T. M.; and Wang, D. 2016. A survey of transfer learning. *Journal of Big data*, 3(1): 1–40.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems(NeurIPS-20)*, 33: 5824–5836.

Zhang, Y.; Qian, Y.; Ma, G.; Liang, X.; Liu, G.; Zhang, Q.; and Tang, K. 2023. ESSR: Evolving Sparse Sharing Representation for Multi-task Learning. *IEEE Transactions on Evolutionary Computation*.