Dynamic Multi-Task Learning with Convolutional Neural Network

Yuchun Fang[1], Zhengyan Ma[1], Zhaoxiang Zhang[2,3,4,5*], Xu-Yao Zhang[3], Xiang Bai[6]
[1]School of Computer Engineering and Science, Shanghai University
[2]Research Center for Brain-inspired Intelligence, CASIA
[3]National Laboratory of Pattern Recognition, CASIA
[4]CAS Center for Excellence in Brain Science and Intelligence Technology
[5]University of Chinese Academy of Sciences
[6]School of Electronic Information and Communication, Huazhong University of Science and Technology

Abstract

Multi-task learning and deep convolutional neural network (CNN) have been successfully used in various fields. This paper considers the integration of CNN and multi-task learning in a novel way to further improve the performance of multiple related tasks. Existing multi-task CNN models usually empirically combine different tasks into a group which is then trained jointly with a strong assumption of model commonality. Furthermore, traditional approaches usually only consider small number of tasks with rigid structure, which is not suitable for large-scale applications. In light of this, we propose a dynamic multi-task CNN model to handle these problems. The proposed model directly learns the task relations from data instead of subjective task grouping. Due to its flexible structure, it supports task-wise incremental training, which is useful for efficient training of massive tasks. Specifically, we add a new task transfer connection (TTC) between the layers of each task. The learned TTC is able to reflect the correlation among different tasks guiding the model dynamically adjusting the multiplexing of the information among different tasks. With the help of TTC, multiple related tasks can further boost the whole performance for each other. Experiments demonstrate that the proposed dynamic multi-task CNN model outperforms traditional approaches.

1 Introduction

Traditional computer vision is based on low-level and handcraft features such as HOG, LBP and Haar, which do not take use of the task-specific supervision information. Higher-level features are usually more specific and more efficient to semantic-level tasks. The design of high-level feature extraction algorithm requires problem-specific knowledge. However, every problem has its own characteristic. Analyzing them with different features is strenuous work. In worse condition, relevant prior knowledge may not be adequate to handle the problem. In that condition, the advantage of representation learning [Bengio *et al.*, 2013] is evident. It is able to automatically learn expressive features, which are more likely to capture the key information of a problem. In recent years, as a widely used efficient representation learning algorithm, deep CNN [LeCun *et al.*, 1989], has been successfully applied in various fields, such as object detection [Girshick *et al.*, 2014; Ren *et al.*, 2015; Redmon *et al.*, 2016], image classification [Krizhevsky *et al.*, 2012; Szegedy *et al.*, 2015; He *et al.*, 2016] and face related problems [Sun *et al.*, 2014; Taigman *et al.*, 2014; Schroff *et al.*, 2015; Yang *et al.*, 2015]. With the enormous amount of parameters in each layer, the cascade structure helps CNN model to extract more abstract features, which are usually more effective for semantic-level tasks.

Multi-Task Learning (MTL) [Caruana, 1998] is a general approach to learning related tasks using shared representation, with the aim of improving the performance. It is quite useful for multiple tasks, which have inherent relations to each other. The superior performance of MTL has been demonstrated in many fields, such as natural language process [He *et al.*, 2009] and computer vision [Quattoni *et al.*, 2008]. Therefore, it is natural to combine CNN and multi-task learning together for proper tasks to get a superior model.

Some previous work has shown that multi-task CNN model is helpful to improve the performance of the main task. The experiments in [Zhang *et al.*, 2014] showed that robust landmark detection can be achieved better through joint learning with heterogeneous but subtly correlated tasks. [Devries *et al.*, 2014] demonstrated that learning representations to predict the position and shape of facial landmarks could improve expression recognition. The work in [Zhang and Zhang, 2014] uses multi-task CNN to build a post filter improving the accuracy of multi-view face detection. The multi-task CNN models mentioned above have similar structures, namely, some layers are artificially set to be shared for all the tasks. In [Misra *et al.*, 2016], cross-stich unit is proposed, which is a principled approach to learning shared representations for multi-task CNN. However, the amount of tasks involved in existing models is very small (around two or three). Even worse, some models differentiate between main task and assistant tasks, which means only main task can benefit from multi-task learning. Current multi-task CNN models empirically select tasks into a group, assuming the tasks are trained in harmony with each other. However, in real applica-

*Corresponding author: zhaoxiang.zhang@ia.ac.cn

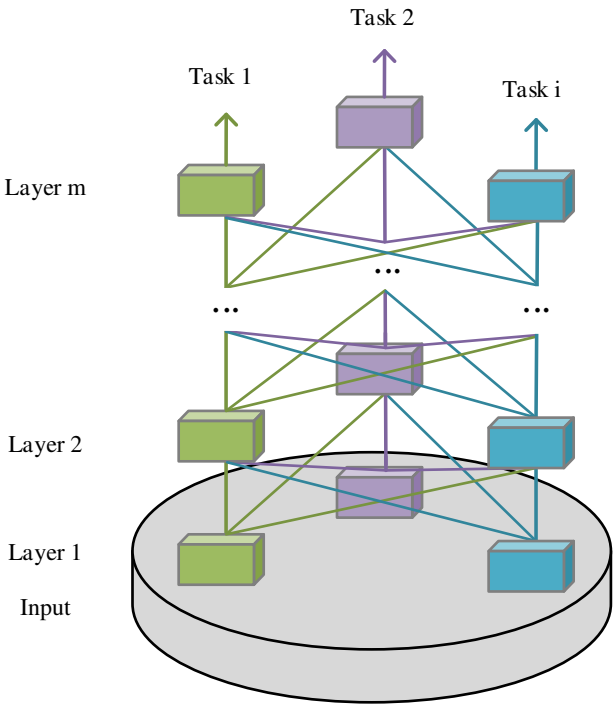


Figure 1: Illustration of dynamic multi-task CNN model. The lines linking different tasks represent task transfer connections.

tions, the relations among tasks are complicated. Rigid information sharing scheme may force tasks to transfer the learned between each other, despite whether it is helpful for the tasks.

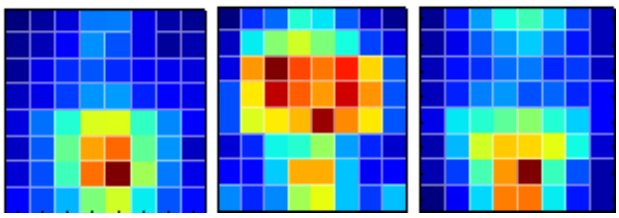
To this end, we propose dynamic multi-task CNN (DMT CNN) model, which is more suitable for multi-task problems. The model is presented in Figure 1. In the enhanced multi-task CNN model, each task holds a subnet. Information sharing within a task group is through task transfer connections (TTC) in corresponding layers. With the dynamic adjustment of TTC, the model is able to make better use of relations among tasks. Single task CNN model only copes with one task; traditional multi-task CNN models focus on whether the information among tasks are shared. Compared with them, our model handles the comprehensive information more elasticly. In training progress of our model, tasks form into weak groups spontaneously. The information sharing among tasks is no longer measure as binary form, but with the degree of relevance. Tasks make use of the information from other tasks in a moderate way. When the information from other tasks is ignored, each subnet can work completely independently as single task CNN models; for any task, if the information from others is equally adopted, the model will be equivalent to the traditional multi-task CNN model.

Current multi-task CNN models usually consider only a small number of tasks. In that situation, it is possible to design the model structure manually for better performance. However, when task amount is huge, the task relations will be intricacy. Therefore it is hard to build a model adaptive for all the tasks. Our model consists of simple subnets, and automatically establishes TTC among them to construct a favorable multi-task structure. This structure is adaptable to different amount of tasks. In the model, all task branches are parallel. Every task branch maintains its completeness and independence. Besides unlimited task amount, the structure is capable of adding new task dynamically in an incremental way. The experimental results show that our model has better performance with different amount of tasks. And with the help of transfer connections, the task-wise incremental training strategy is more efficient than training from scratch.

2 Motivation

In multi-task learning, diverse task groups lead to different results. Multi-task model performs well when related tasks are learned jointly. However, it is an open question that what related tasks are. There is no adequate definition of task relatedness or the guidance of selecting related tasks. In preliminary experiments, we find that the multi-task CNN model performs better, when tasks are grouped according to CNN response maps for binary classification problem.

To verify the effect of feature response grouping method, we take the experiment on celebA [Liu *et al.*, 2015]. CelebA is a data set with massive face attribute recognition tasks. We compare the mean feature maps of positive samples and negative samples of various face attributes. A response region with high contrast between positive and negative indicates that the region maintains key signals for recognition since the positive and negative samples have different response there. We form the tasks that have similar key response regions into a group. The response contrast maps of several attributes are shown as examples in Figure 2.



(a) Goatee (b) HeavyMakeup (c) Smiling

Figure 2: Examples of response contrast map.

In order to measure the correlation of two attributes, we compute the L1 distance between the corresponding response contrast maps. Attributes with shorter distance mean they have more overlapped interest spots, which are more likely to benefit from learning jointly. We select multiple attributes which have the shortest distances with each other within a group. Then they are trained in multi-task CNN model. Such as Sideburns, Goatee, and NoBeard. The performances of different task combinations are shown in Table 1. Multi-task models (MT CNN) show more superior results than single task models (ST CNN). [Liu *et al.*, 2015] proposes a face attribute grouping method by co-occurrence. We select the attributes with high frequency of co-occurrence. The performance is shown in Table 2. The results show that task grouping is important for multi-task model and the response map grouping method is able to find out related attributes that are more likely to benefit from multi-task learning.

Attributes	ST CNN	MT CNN
Sideburns	91.8%	**92.5%**
Goatee	91.7%	**92.6%**
NoBeard	**92.0%**	**92.0%**
H.Makeup	89.1%	**89.4%**
Wear.Lipstick	92.5%	**92.6%**
Smiling	91.8%	**91.9%**
H.Cheekbones	86.0%	**86.2%**

Table 1: Performance of multi-task cnn grouped by response maps.

Attributes	ST CNN	MT CNN
NoBeard	**92.0%**	91.6%
Smiling	91.8%	**91.9%**
NoBeard	92.0%	**92.2%**
Wear.Lipstick	**92.5%**	92.4%
NoBeard	**92.0%**	91.4%
Young	81.1%	**81.7%**

Table 2: Performance of multi-task CNN grouped by co-occurrence frequency.

3 Dynamic Multi-Task CNN Model

As shown in Section 2, multi-task performs well when tasks are well grouped. However, task grouping is not provided in advance in most real applications. The feature-based grouping method can only measure the correlation of two tasks. It is a more challenging probem when multiple tasks need to be grouped. A more general method is required. Moreover, for multi-task learning, more attention is paid to whether features are learned jointly, instead of measuring the degree of task sharing. Supervised learning aims to capture and represent the relevant information in the input variable with respect to the output. The presentation ability of a model is finite [Tishby and Zaslavsky, 2015]. So the model may be stretched thin when the jointly learned tasks are quite distinctive. Hence, the degree of information sharing among tasks should be flexible, especially when the relations of grouped tasks are not entirely clear.

To overcome the obstacles of multi-task learning problems, we propose the dynamic multi-task CNN model. Compared with conventional CNN model, the major change is the dynamic connections amount subnets of different tasks. Supervisory signals from the higher layers are shared in the lower layers within tasks group. Task transfer connections (TTC), which control the impact of a supervisory signal at a certain lower layer, are dynamically learned by Gradient Descent algorithm. Via this method, tasks are automatically arranged in weak groups during training, which means relations among tasks are no longer measured in binary case as related or not related, but as the degree of relevance. When the TTC factors $\alpha^{ij}(i, j \in taskset)$ are the same, the proposed model becomes conventional multi-task CNN model; when the TTCs within a same subnet $\alpha^{ij}(i = j)$ are 1, and those across tasks are 0, the model degenerates into multiple single-task learning models.

Traditional CNN model can be represented as a function $f(.)$ of input X and weights W:

$$f(X, W). \tag{1}$$

The target is minimizing the discrepancy between ground truth Y and the output F of $f(X, W)$:

$$min(C). \tag{2}$$

where:

$$C = -[Yln(F) + (1 - Y)ln(1 - F)]. \tag{3}$$

When there are n tasks, dynamic multi-task CNN model is represented as function of X,W, and TTC α:

$$F(X_1, X_2, ..., X_n, W_1, W_2, ..., W_n, \alpha). \tag{4}$$

And the target of multi-task model is minimizing the cost of all the tasks:

$$\min(\sum_{i=1}^{n}(C_i)). \tag{5}$$

where:

$$C_i = -[Y_i ln(F_i) + (1 - Y_i)ln(1 - F_i)], \tag{6}$$

$$F(X_i, W_1, W_2, ..., W_n, \alpha). \tag{7}$$

Considering the meaning of TTC factor, α is expected in the range of 0 to 1. Therefore, we introduce auxiliary variable β where:

$$\alpha = sigmoid(\beta), \beta \in (-\infty, +\infty). \tag{8}$$

As to the training time, the major difference of weight updating between our multi-task model and conventional CNN model is that each subnet needs to take supervisor signals from other subnets into account besides its own. The weight updating detail of various optimization algorithms (such as Stochastic Gradient Descent, Adaptive Gradient, and RMSprop) is different. Hence, we only show the general forms. In conventional CNN model, weight updating can be represented as:

$$W_{t+1} = W_t + V_{t+1}. \tag{9}$$

In our model, to obtain weights of n tasks, $W^1, W^2, ..., W^n$ and all the TTC factors α^{ij}, the weight updating procedure is carried out as algorithm 1.

In algorithm 1, W^i_{t+1} is the updated weights at iteration $t + 1$ of task i, W^i_t is weights at iteration t; α^{ij}_t is the corresponding TTC factor of W^i_t to task j, which controls impact of supervisory signal from task j on task i; β^{ij}_t is the auxiliary variable; $V^{W^{ij}}_{t+1}$ and $V^{\beta^{ij}}_{t+1}$ are the update values of W^i and β^{ij} for computing the cost of task j by a specific optimization algorithm:

$$V^{W^{ij}}_{t+1} = \frac{\partial c_j}{\partial W^j} \cdot \frac{\partial W^j}{\partial W^i}, \tag{10}$$

Algorithm 1 Parameter updating

Input: Labeled data of task set T.

Initialize W_0^i and $\alpha_0^{ij}(i, j \in T)$.
while not converged **do**
  \\ Update the weights of each CNN subnet:
  for $i \in T$ **do**
    $W_{t+1}^i = W_t^i + \sum_{i=1}^n \alpha_t^{ij} V_{t+1}^{W^{ij}}$
  end for
  \\Update the TTC factors:
  for $i \in T$ **do**
    for $j \in T$ **do**
      $\beta_{t+1}^{ij} = \beta_t^{ij} + V_{t+1}^{\beta^{ij}}$
      $\alpha_{t+1}^{ij} = sigmoid(\beta_{t+1}^{ij})$
    end for
  end for
end while

$$V_{t+1}^{\beta^{ij}} = \frac{\partial c_j}{\partial \alpha^{ij}} \cdot \frac{\partial \alpha^{ij}}{\partial \beta^{ij}}. \tag{11}$$

Notably, to keep the dominant position of the information from a subnet itself, the supervisory signals from other subnets will be weaker. Therefore, α_t^{ij} is set to 1 when $i = j$. To relieve the range fluctuation of feature values, batch normalization is applied.

Because our model maintains completeness for each task branch, it is able to add new tasks by incremental approach, which is superior to the traditional training strategy. In training progress, the subnets of the new tasks establish connections with the trained tasks and meanwhile learn the parameters of their own; the already learned tasks keep the parameters of corresponding subnets fixed. Figure 3 illustrates the situation where two tasks are trained and one new task is merged into the model.

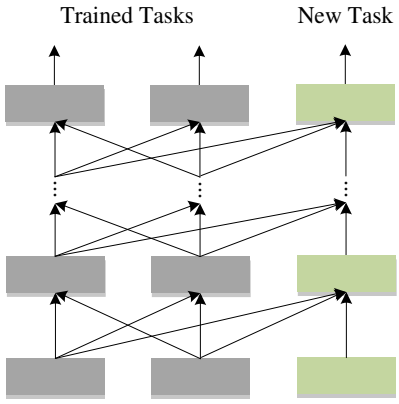


Figure 3: Incremental task transfer learning, in the case of two trained tasks and one new task.

4 Experiments and Analysis

As a typical computer vision problem, face attributes recognition is very challenging. One face image usually contains many face attributes, and among the massive tasks there are complex relations. Therefore, our experiments are mainly conducted on face attributes dataset. CelebA is one of the largest face attribute dataset, which contains two hundred thousand images. Each image has 40 attributes labeled with binary classes: positive or negative, meaning with or without the corresponding attribute. The sample distributions across the attributes are highly imbalanced.

Attribute		Bald	Wear.Hat	Mustache	Goatee
Minority Proportion		0.02	0.05	0.04	0.06
Trained On Original Unbalanced Dataset	ST	56.4%	82.2%	58.6%	73.3%
	MT	79.4%	89.1%	60.3%	78.7%
		Learned jointly		Learned jointly	
	DMT	**80.1%**	**91.2%**	**62.0%**	**79.3%**
		Learned jointly		Learned jointly	
Trained After Resampling Minority Samples	ST	95.5%	95.3%	90.9%	93.7%
	MT	**96.2%**	95.5%	91.2%	**94.1%**
		Learned jointly		Learned jointly	
	DMT	**96.2%**	**96.0%**	**91.9%**	**94.1%**
		Learned jointly		Learned jointly	

Table 3: Performance of different models on two tasks.

The training set has significant impact on the performance. Thus, to avoid the influence of extraneous variables, all the experiments only take use of the original training set without data augmentation or pre-training. Before experiments, we preprocess all the images to face aligned and histogram equalized grayscale images.

4.1 Multi-task Performance on Two Tasks

To test the performance of our model, we take experiments on celebA and choose four attributes with the most imbalanced sample distributions. The tasks are arranged into two groups. Single task CNN model (ST CNN), traditional multi-task CNN model (MT CNN) and dynamic multi-task CNN (DMT CNN) is compared under the same conditions. Each task uses the same model structure, which has five convolutional layers and two fully connected layers. The parameters of 5 convolutional layers are (40, 5x5) (kernel amount, height x width), (60, 5x5), (80, 3x3), (100, 3x3), (140, 2x2). The first fully connected layer has 360 neurons and the second one is the output layer. Between each two convolutional layers, there are non-overlapped max-pooling layers. The activation function is ReLU.

Sample imbalance might hinder the model performance [Japkowicz and Stephen, 2002]. Even worse, sample imbalance plus multi-label tasks can make the learning harder to perform [Boutell *et al.*, 2004; Fang *et al.*, 2014]. Since each sample may be positive samples of some attributes and negative samples of the other attributes, simply resampling to

Attribute	Goatee	HeavyMakeup	H.Cheekbones	NoBeard	Sideburns	Smiling	Wear.Lipstick
[Liu *et al.*,2015] LNets+ANet(w/o)	92.0%	85.0%	84.0%	92.0%	91.0%	88.0%	90.0%
MT	93.9%	89.7%	86.0%	91.9%	93.1%	91.8%	92.7%
DMT	**94.9%**	**89.8%**	**86.5%**	**92.6%**	**93.9%**	**92.1%**	**93.2%**
DMT(tasks removal)	93.1%	89.1%	85.8%	91.2%	/	91.7%	/

Table 4: Performance of different models with more tasks.

weaken the influence of imbalance for one attribute, may lead to more severe imbalance problem for others. To handle the multi-label imbalance problem, we resample the data of minor class for each task respectively, and the tasks take in turns to train the model with its own resampled data pool. The recognition results of four attributes with most imbalanced sample distribution are given in Table 3. The minority proportion of each attribute on the training set is also listed. The model trained on an imbalanced set tends to identify all the test samples to the major class, which severely decrease the robustness and effectiveness of the model. In consideration of sample distribution, we test the positive sample and negative sample respectively and get the corresponding accuracy acc_p and acc_n. The final accuracy shown in tables are the mean of acc_p and acc_n. The first part of Table 3 is the test results on the balanced training set. The second part of Table 3 shows the performances of three models on the original imbalanced training set. The results show that DMT CNN performs better than MT CNN and ST CNN.

4.2 Tasks Capacity

The former experiments show the performances when multi-task model with two tasks. To verify the task capacity of our model, we take the experiment with more tasks. On CelebA, we select seven face attributes from Table 1. The training sets of all the tasks are resampled to make sure each batch contains equal amount of negative and positive samples. We take the single task CNN performance without pretraining from [Liu *et al.*, 2015] as a baseline. For multi-task CNN model, all of these attribute recognition tasks are learned jointly. As presented in Table 4, traditional multi-task CNN (MT CNN) and the proposed dynamic multi-task CNN (DMT CNN) both perform better than single task CNN. And DMT CNN is the most superior.

Besides face dataset, we also test our model on cifar-10. We split cifar-10 into five tasks: airplane_ship, automobile_truck, bird_frog, cat_dog and deer_horse. Each task contains two classes of objects. And our model shows competive performance as shown in Table 5.

4.3 Task Transfer Connection Analysis

To explore the relations among tasks, we analyse the tasks in Table 4. Figure 4 and Figure 5 shows the influence of other tasks on a specific task. The task grouping result is consistent with the former discovery in Section 2. The tasks within a group in Table 1 are shown to have stronger connections. For example, the face attribute NoBear shares more information with Sideburns and Goatee, while it has slight connection

Tasks	ST	MT	DMT
cat_dog	73.8%	**73.9%**	**73.9%**
deer_horse	88.8%	89.1%	**89.4%**
bird_frog	89.2%	89.6%	**90.7%**
airplane_ship	89.9%	90.4%	**91.2%**
automobile_truck	88.4%	**90.0%**	**90.0%**

Table 5: Performance of different models on cifar-10.

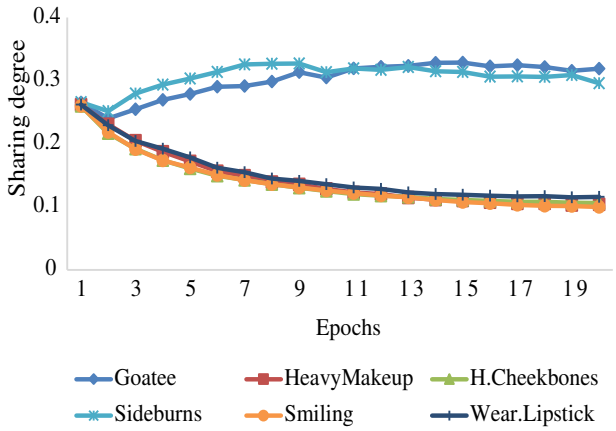


Figure 4: Relative magnitudes of influence of other tasks on NoBeard task in training progress. Take layer 6 in subnet as an example.

with other tasks. The result is in accordence with common sense. People with heavy makeup usually wear lipsticks, and when to smile, the cheekbones are more obvious. Similarly, in the test result of cifar-10, tasks about animals have closer correlation with each other; transportations do the same.

The schema of information sharing among tasks is important. The TTC at higher layers shows obvious connections with related tasks. This phenomenon indicates that what the high level layers learn is closer to semantic information. And for multi-task learning, the high level layers may still share some useful information among tasks. Thus, completely insulating tasks in high level layers in a multi-task CNN is probablely not the optimal architecture.

To verify the effectiveness of the task transfer connection (TTC), we remove some tasks from the trained dynamic

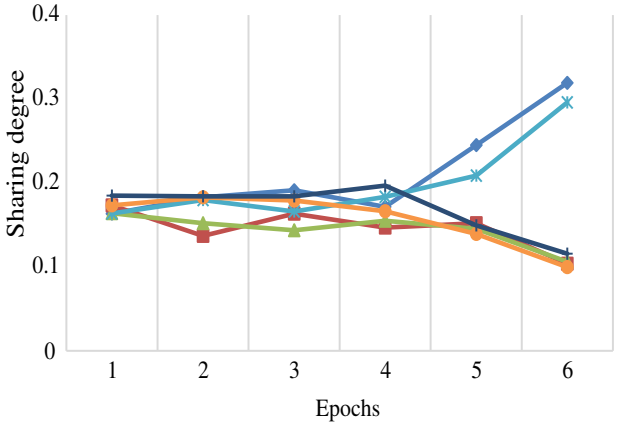


Figure 5: Sharing degree of other tasks on NoBeard task in different layers. The legend is the same as Figure 4.

From \ To	Goatee	H.Makeup	H.Cheek.	NoBeard	Smiling
Goatee	+6%	+10%	+4%	+7%	+3%
H.Makeup	+11%	+15%	+3%	+3%	+8%
H.Cheek.	+3%	0%	+9%	+1%	+1%
NoBeard	-2%	+7%	+3%	+9%	+3%
Smiling	+1%	+2%	-2%	0%	-2%

Table 6: Changes of TTC weights from layer 6 to layer 7.

multi-task model. Specifically, we cut off some subnets from the trained model while frozen the others. Then we re-train the TTC to reconnect the incomplete model. For example, we remove Sideburns and Wear.Lipstick from the tasks in Table 4. After reconnection, the information route changes. The changes of TTC weight from layer 6 to layer 7 are enumerated in Table 6. The former TTC weights of removed tasks are partitioned by reserved tasks. Task removal results in the optimal structure broken, decreasing model performance. As shown in the last row of Table 4, all the recognition accuracy is influenced. This phenomenon implys tasks take advantage of properly learned TTC better than casual information sharing structure.

4.4 Task-wise Incremental Transfer Learning

Most models work normal when the requirements are stable. However, when task amount is changed especially new task need to be added to the model, traditional multi-task CNN need to be re-trained entirely. Due to the flexibility of our model, the learned parameters for other tasks are able to be utilized by the new task with the help of TTC. By task-wise incremental transfer learning, new tasks can be added to the already trained model, meanwhile the training time is significantly reduced. For example, we train the attribute Smiling alone, and then the error rate in training progress is compared with that of the incremental method. The result in Figure 6 shows that our model with incremental transfer learning is more efficient. It is about 4 times faster than training a single-task CNN model to reach the error rate of 0.1.

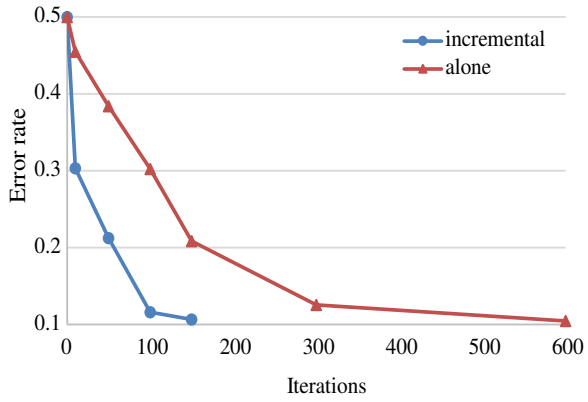


Figure 6: Error rates in training progress. Task-wise incremental transfer learning is more efficient.

5 Conclusion

In this paper, a novel multi-task CNN model is proposed. Task transfer connections (TTCs) are established during training, which automatically partition tasks into different weak groups. The relations among tasks are reasonably measured by learned softly relevant degrees. Experimental results demonstrate that the proposed dynamic multi-task CNN model performs better than the traditional multi-task CNN structure. Because the structure is flexible, our model is capable of adapting to large quantity of tasks. With the help of TTC, it is able to conduct incremental task transfer learning, which can further boost the training efficiency.

Acknowledgments

The work is funded by National Nature Science Foundation of China (No. 61170155, 61375036, 61511130079) and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR). Zhaoxiang Zhang is the corresponding author of this paper.

References

[Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[Boutell *et al.*, 2004] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[Caruana, 1998] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[Devries *et al.*, 2014] Terrance Devries, Kumar Biswaranjan, and Graham W. Taylor. Multi-task learning of facial

landmarks and expression. In *Computer and Robot Vision*, pages 98–103, 2014.

[Fang *et al.*, 2014] Ming Fang, Yuqi Xiao, Chongjun Wang, and Junyuan Xie. Multi-label classification: Dealing with imbalance by combining labels. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pages 233–237. IEEE, 2014.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[He *et al.*, 2009] Saike He, Taozheng Zhang, Xue Bai, Xiaojie Wang, and Yuan Dong. Incorporating multi-task learning in conditional random fields for chunking in semantic role labeling. In *Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on*, pages 1–5. IEEE, 2009.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[Japkowicz and Stephen, 2002] Nathalie Japkowicz and Shaju Stephen. *The class imbalance problem: A systematic study*. IOS Press, 2002.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LeCun *et al.*, 1989] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[Misra *et al.*, 2016] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[Quattoni *et al.*, 2008] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Transfer learning for image classification with sparse prototype representations. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Schroff *et al.*, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[Sun *et al.*, 2014] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[Taigman *et al.*, 2014] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[Tishby and Zaslavsky, 2015] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE, 2015.

[Yang *et al.*, 2015] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015.

[Zhang and Zhang, 2014] Cha Zhang and Zhengyou Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 1036–1041. IEEE, 2014.

[Zhang *et al.*, 2014] Zhanpeng Zhang, Ping Luo, Change Loy Chen, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108, 2014.