Progressive Lifelong Learning by Sharing Representations for Few Labeled Data

Guoxi Su(✉), Xiangmin Xu, Chaowen Chen, Bolun Cai, and Chunmei Qing

South China University of Techonology,
381 Wushan Road, Guangzhou 510640, China
su.guoxi@mail.scut.edu.cn, {xmxu,qchm}@scut.edu.cn,
czwmtnh@gmail.com, caibolun@gmail.com

Abstract. Lifelong Machine Learning (LML) has been receiving more and more attention in the past few years. It produces systems that are able to learn knowledge from consecutive tasks and refine the learned knowledge for a life time. In the optimization process of classical full-supervised LML systems, sufficient labeled data are required for extracting inter-task relationships before transferring. In order to leverage abundant unlabeled data and reduce the expenditure of labeling data, an progressive lifelong learning algorithm (PLLA) is proposed in this paper with unsupervised pre-training to learn shared representations that are more suitable as input to LML systems than the raw input data. Experiments show that the proposed PLLA is much more effective than many other LML methods when few labeled data is available.

Keywords: Lifelong machine learning · Representation learning
Few labeled data

1 Introduction

Over the last few decades there have been critical progresses in machine learning theory and algorithms which aims to enable machines to learn intelligently like human. However, the capacity of machines for persistent learning has a large gap from that of human. And it is now appropriate to more seriously consider the nature of systems that are capable of learning, retaining and using knowledge over a life time [1]. It has a variety of related applications such as robotic controlling [2], online image retrieval [3,4] and topic modelling [5–7].

Among all the LML algorithms, ELLA (Efficient Lifelong Learning Algorithm) [8] is a representative and effective algorithm which achieves nearly identical performance to batch Multi-task Learning (MTL) [9] with three orders of magnitude speedup in learning time. ELLA develops an efficient procedure of updating shared knowledge between each learned task and improved performance of learned task through reverse transfer. However, ELLA is a supervised learning algorithm so that its training procedure needs plenty of labeled data while labeling data needs the expenditure of much time and work especially

B. Huet et al. (Eds.): ICIMCS 2017, CCIS 819, pp. 411–418, 2018.
https://doi.org/10.1007/978-981-10-8530-7_40

under the current big data environment. The lack of labeled training data may also restrict the model of ELLA to scale up. On the other hand, future machine learning algorithms tend to learn without supervision.

In order to effectively exploit unlabeled data, this paper proposed an progressive lifelong learning algorithm (PLLA) based on ELLA and Deep Belief Network (DBN) [10,11]. This is based on the fact that unsupervised deep learning methods can capture underlying regularities in the data and project all the raw input data to a shared feature representation. When used to learn multifarious and consecutive tasks, experiments show that hierarchically learned features help to capture commonalities between tasks and gets much better performance than ELLA and other LML methods when using less training labeled data.

2 Related Work

In this section, we introduce some LML and online multi-task learning (OMTL) frameworks which is related to our work in sharing representations or integrating hypothesis. Differences between PLLA and other methods are also illustrated.

Inspired by the short-term and long-term learning in psychology, Silver proposed an algorithm of LML based on multi-task learning (MTL) neural network [12]. In this framework, the input layer and hidden layer are shared among tasks to transfer knowledge and the output nodes are task specific. Recently, Lifelong Learning of Discriminative Representations (LLDR) [13,14] extended the MTL neural networks in order to deal with high dimensional problems and large amount of tasks in actual lifelong learning. This framework is similar to our work with shared hierarchical representations and task specific hypothesis. But we have further considerations on the transferring of the hypothesis functions and updating the representations with inherited gradients.

Compared to the above OMTL and LML paradigms where all tasks are in a single group [15–17], learning task grouping may be a better way to transfer knowledge between tasks in LML.

Disjoint grouping MTL (DG-MTL) [9] presented a model that can share representations among tasks in the same group while learning the disjoint grouping simultaneously. More recently, Mishra extended DG-MTL to fit in lifelong learning setting [18] which learned both partition functions and parameters online. These algorithms have different assumptions on task grouping from ours where tasks in different groups are totally untransferable.

Against the disjoint grouping models, the Grouping and Overlapping MTL (GO-MTL) algorithm [19] is a rich model of underlying task structure exploiting a sparsely shared basis. It automatically learns overlapping groups of tasks that allowing two tasks from different groups to share knowledge by one or more basis in common. Efficient Lifelong learning Algorithm (ELLA) [8] is developed employing GO-MTL as its starting point, greatly reducing its running time while retaining nearly identically performance. This work has been extended by the authors in multiple ways as in [2,20]. This efficient LML framework of integrating hypothesis is a fundamental part of our work. We adapt it for shared representations and introduce a new online updating strategy to ensure efficiency.

3 Progressive Lifelong Learning with Shared Representations

Lifelong Machine Learning considers systems that can learn many tasks from one or more domains over its lifetime [1]. We employ a lifelong learning framework in which the agent faces a series of supervised learning tasks $\mathcal{Z}^{(1)}$, $\mathcal{Z}^{(2)}$,...,$\mathcal{Z}^{(T_{max})}$. Each learning task $\mathcal{Z}^{(t)} = (\hat{f}^{(t)}, \mathbf{X}^{(t)}, \mathbf{y}^{(t)})$ is defined by a hidden function $\hat{f}^{(t)}{:}\mathcal{X}^{(t)} \rightarrow \mathcal{Y}^{(t)}$ from an instance space $\mathcal{X}^{(t)} \subseteq \mathbb{R}^d$ to a set of labels $\mathcal{Y}^{(t)}$ where $t = 1, 2, ..., T_{max}$. To learn $\hat{f}^{(t)}$, the agent is given n_t training instances $\mathbf{X}^{(t)} \in \mathbb{R}^{d \times n_t}$ with corresponding labels $\mathbf{y}^{(t)} \in \mathcal{Y}^{(t)^{n_t}}$ given by $\hat{f}^{(t)}$. Its goal is to construct task-specific hypothesis function $f^{(t)}$ for each task t to ensure the accuracy of labeling new data.

To model the relationships between tasks, it is assumed that the parameter vectors $\boldsymbol{\theta}^{(t)}$ can be represented using a linear combination of k shared latent model components from $\mathbf{L} \in \mathbb{R}^{d \times k}$ by computing $\boldsymbol{\theta}^{(t)} = \mathbf{L}\mathbf{s}^{(t)}$ where the weight vector $\mathbf{s}^{(t)} \in \mathbb{R}^k$ is encouraged to be sparse.

PLLA is formed by two layers. The upper layer is the inferring layer based on features extracted from the lower one. The lower layer is the shared hierarchical feature model initialized by unsupervised pre-training. The structure of PLLA is illustrated in Fig. 1.

Since we have integrated the shared feature representations and the potential knowledge basis into a new model, the objective function is changed from that of ELLA: (assume that the representations have only one layer of hidden units for brevity)

$$
\begin{aligned}
e_T(\mathbf{L}, \mathbf{W}) = \frac{1}{T} \sum_{t=1}^{T} \min_{\mathbf{s}^{(t)}} \Bigg\{ \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}(f(sigmoid(\mathbf{W}^\top \mathbf{x}_i^{(t)});& \\
\mathbf{L}\mathbf{s}^{(t)}), y_i^{(t)}) + \mu \|\mathbf{s}^{(t)}\|_1 \Bigg\} + \lambda \|\mathbf{L}\|_2^F.&
\end{aligned}
\tag{1}
$$

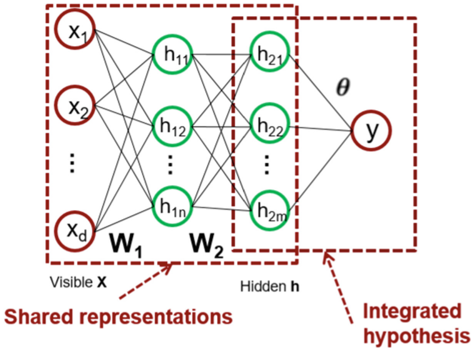

Fig. 1. The framework of PLLA with DBN representations and integrated hypothesis.

Algorithm 1. PLLA

```
Require: labeled data set ($\mathbf{X}_l^{(t)}$,$\mathbf{y}^{(t)}$) of current task $t$
Ensure: the library of learned knowledge $\mathbf{L}$
    and the hypothesis function $f^{(t)}$ with its parameters $\boldsymbol{\theta}^{(t)}$
initialize $\mathbf{W}_l$ of each layer $l$ with $\bigcup_{t=1}^{T_c} \mathbf{X}^{(t)}$ by CD-k algorithm, ($T_c$ is the number of candidate tasks)
while isMoreTaskToLearn() do
    $(\mathbf{X}_l^{(t)}, \mathbf{y}^{(t)}, t) \leftarrow$ getTrainningDataSet()
    apply visible to hidden algorithms to get $\mathbf{H}^t$
    $\boldsymbol{\theta}^{(t)} = argmin_{\boldsymbol{\theta}} \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}(f(\mathbf{h}_i^{(t)}; \boldsymbol{\theta}^{(t)}), y_i^{(t)})$
    $\mathbf{s}^{(t)} = argmin_{\mathbf{s}^{(t)}} f_l(\mathbf{L}, \mathbf{s}^{(t)}, \boldsymbol{\theta}^{(t)}, \mathbf{D}^{(t)})$, where $\mathbf{D}^{(t)}$ is the Hessian matrix of the loss function on task t
    compute $\gamma_t = (cos < \mathbf{s}^{(t)}, \mathbf{s}^{(t-1)} > +1)/2$
    for $i = 1$ to maxepoch do
        apply visible to hidden algorithms to get $\mathbf{H}^t$
        $\boldsymbol{\theta}^{(t)} = argmin_{\boldsymbol{\theta}} \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}(f(\mathbf{h}_i^{(t)}; \boldsymbol{\theta}^{(t)}), y_i^{(t)})$
        for $l = n_l - 1$ downto 0 do
            if $(i == 1) and (t \geq 1)$
            $g\mathbf{W}_l^t = \gamma_t g\mathbf{W}_l^{t-1} + \nabla_{\mathbf{W}_l} \mathcal{L}(f(\mathbf{H}^t; \boldsymbol{\theta}^{(t)}), y_i^{(t)})$
            else
            $g\mathbf{W}_l^t = \gamma_c g\mathbf{W}_l^t + \nabla_{\mathbf{W}_l} \mathcal{L}(f(\mathbf{H}^t; \boldsymbol{\theta}^{(t)}), y_i^{(t)})$
            end if
            $\mathbf{W}_l = \mathbf{W}_l - \alpha g\mathbf{W}_l^t$
        end for
    end for
    get the new $\boldsymbol{\theta}^{(t)}$ by single task learners from $(\mathbf{H}^t, \mathbf{y}^t)$
    $\mathbf{L} \leftarrow argmin_{\mathbf{L}}(\lambda \|\mathbf{L}\|_2^F + \frac{1}{T} \sum_{t=1}^{T} f_l(\mathbf{L}, \mathbf{s}^{(t)}, \boldsymbol{\theta}^{(t)}, \mathbf{D}^{(t)}))$
end while
```

However, Eq. 1 is not jointly convex in $\mathbf{L}$ and $\mathbf{W}$, it is difficult and inefficient to optimize them simultaneously. As described in the block of Algorithm 1, we make an online strategy that optimizing the feature representation matric $\mathbf{W}$ firstly by minimizing the lost function on the labeled data of current task. The gradients for updating each weighting matric at the first iteration is inherited from the last one by a coefficient γ which capturing the relatedness between tasks:

$$\begin{aligned} \gamma_t &= dist(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}) \\ &= dist(\mathbf{L}\mathbf{s}^{(t)}, \mathbf{L}\mathbf{s}^{(t-1)})) \\ &= cos < \mathbf{s}^{(t)}, \mathbf{s}^{(t-1)} >, \end{aligned} \tag{2}$$

where $dist()$ is the cosine distance of parameters between task t and task $t-1$. We also rescale γ_t to the range [0, 1].

The intuition is if current task and learned task are closely related, the angle between their selecting vectors should be nearly the same. During the following iterations, gradients are maintained with a hyper parameter γ_c.

4 Experiments

In this section, we evaluate our proposed PLLA model against five baselines: Single task learning (STL), Disjoint grouping MTL (DG-MTL)[9], LLDR [13], ELLA [21] and Online multi-task boosting (OMB)[22]. In experiments, we reduced the amount of labeled data of each task to [10%, 20%, ..., 100%]. We ensure that all models exploit the same amount labeled data in the experiments. We evaluate the performance on prediction over two databases: the Land Mine Data set and the London School Data set.

4.1 Parameter Settings

All the comparing models have some hyper-parameters need to be confirmed by the user. We also use gridsearch procedure if the algorithm has multiple hyper-parameters which need to be selected.

In the STL mehtod, the regularization coefficient of regression is picked in $\{exp(-5), exp(-4), ..., exp(5)\}$.

In DG-MTL, the number of groups are chosen from a pool of $\{2, 3, 4, 5\}$ and the values of regularization parameters are picked from $\{0.001, 0.01, 0.1, 1, 10, 100\}$.

In ELLA, The parameter values of k and λ are selected independently for each algorithm and data set using a gridsearch over values of k (the number of hidden basis) from 2 to 10 and values of the ridge term for single task learner, λ_1 from the set $\{exp(-5), exp(-4), ..., exp(5)\}$. We also pick the regularization parameters for the basis and the sparsity constraint from $\{exp(-10), exp(-5), exp(-2), exp(1), exp(4)\}$. Other parameter settings follow the default settings in the code that provided by the authors.

In PLLA and LLDR, the number of hidden layers n_l is chosen over values $n_l \in \{1, 2\}$. Since the size of the database is not huge, it is unnecessary to build deeper architectures. Hyper parameter γ_c in PLLA is picked in $\{0.1, 0.2, ..., 0.9\}$ by grid-search and the learning rate α is chosen in $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$. The regularization coefficient for single task learning was selected in $\{exp(-5), exp(-4), ..., exp(5)\}$.

In OMB, the number of base learners is chosen from 5 to 20 and we exploit Naive Bayesian classfiers as the base learners.

4.2 Land Mine Detection

In the Land Mine data set [23], the goal is to detect whether or not a land mine is present in an area based on RADAR images. The 10 input features (plus a bias term) are extracted from radar data. The data set consists of 14,820 instances in total, divided into 29 different geographical regions. We treat each region as a different task.

In this database, one hidden layer is formed in PLLA $n_l = 1$ and the number of hidden variables m is 10. In this database, the number of total labeled data for each task varies from 449 to 690.

Since the land mine data is real-valued, we use Gaussian-Binary DBN in our model instead of binary DBN.

4.3 London School Data

The London School data set consists of examination scores from 15,362 students in 139 schools from the Inner London Education Authority. We treat the data from each school as a separate task. The goal is to predict the examination score of each student with 27 different features.

In this database, one hidden layer is formed in PLLA $n_l = 1$ and the number of hidden variables m is 30. We also set a decreasing learning rate since this dataset has much more tasks than others.

4.4 Results

Figure 2 left shows the results of the performance on prediction (AUC) with few labeled data over the Land Mine Data set. The given average and the standard deviation results are computed over running 100 times. It can be observed that our proposed PLLA gets about 8% improvement than ELLA when only 10% labeled data is leveraged.

Figure 2 right depicts the similar regularity on the London School Data set. PLLA is always more efficient than ELLA when few labeled data are exploited. On the other hand, the result with only 30% of labeled data in PLLA is nearly identical to the result of ELLA with 100% labeled data. DG-MTL does not fit for this situation again with so many tasks to be learned. Its efficiency and performance are both the worst.

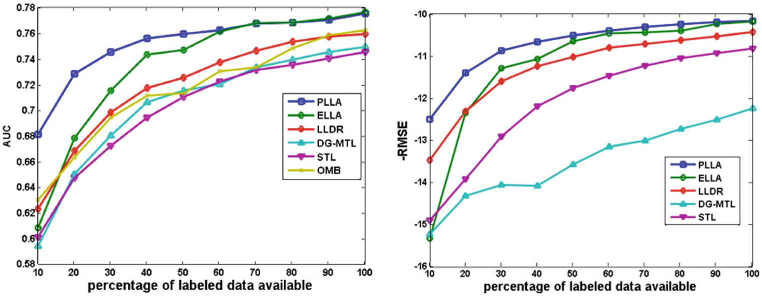

Fig. 2. The results of training with different amount of labeled data on left: land mind detection; right: London school

5 Conclusion

In order to reduce human effort in labeling data for supervised lifelong machine learning, it is motivated to consider how to improve the performance on prediction when few labeled data is available. In this paper, we proposed an effective algorithm called PLLA. It can discover and leverage the hidden structures in the unlabeled data to enhance the performance on prediction of supervised learning especially when few label data is available.

Acknowledgements. This work is supported in part by the National Natural Science Founding of China (61171142, 61401163, U1636218), Science and Technology Planning Project of Guangdong Province of China (2014B010111003, 2014B010111006), the Fundamental Research Funds for the Central Universities (2017MS045), and Guangzhou Key Lab of Body Data Science (201605030011).

References

1. Silver, D.L., Yang, Q., Li, L.: Lifelong machine learning systems: beyond learning algorithms. In: AAAI Spring Symposium: Lifelong Machine Learning (2013)
2. Ammar, H.B., Eaton, E., Ruvolo, P., Taylor, M.: Online multi-task learning for policy gradient methods. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1206–1214 (2014)
3. Banko, M., Etzioni, O.: Strategies for lifelong knowledge extraction from the web. In: Proceedings of the 4th International Conference on Knowledge Capture, pp. 95–102 (2007)
4. Hong, R., Yang, Y., Wang, M., Hua, X.S.: Learning visual semantic relationships for efficient visual retrieval. IEEE Trans. Big Data **1**, 152–161 (2017)
5. Chen, Z., Liu, B.: Topic modeling using topics from many domains, lifelong learning and big data. In: International Conference on International Conference on Machine Learning, pp. 703–711 (2014)
6. Hong, R., Hu, Z., Wang, R., Wang, M., Tao, D.: Multi-view object retrieval via multi-scale topic models. IEEE Trans. Image Process. **25**, 5814–5827 (2016)
7. Hong, R., Zhang, L., Zhang, C., Zimmermann, R.: Flickr circles: aesthetic tendency discovery by multi-view regularized topic modeling. IEEE Trans. Multimedia **18**, 1555–1567 (2016)
8. Ruvolo, P., Eaton, E.: Ella: an efficient lifelong learning algorithm. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), June 2013
9. Kang, Z., Grauman, K., Sha, F.: Learning with whom to share in multi-task feature learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 521–528 (2011)
10. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
11. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**, 504–507 (2006)
12. Silver, D.L., Mercer, R.E.: The task rehearsal method of life-long learning: overcoming impoverished data. In: Cohen, R., Spencer, B. (eds.) AI 2002. LNCS (LNAI), vol. 2338, pp. 90–101. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47922-8_8
13. Alsharif, O., Bachman, P., Pineau, J.: Lifelong learning of discriminative representations. CoRR (2014)
14. Alsharif, O., Bachman, P., Pineau, J.: Representation as a service. arXiv preprint arXiv:1404.4108 (2014)
15. Archambeau, C., Guo, S., Zoeter, O.: Sparse bayesian multi-task learning. In: Advances in Neural Information Processing Systems, pp. 1755–1763 (2011)
16. Evgeniou, A., Pontil, M.: Multi-task feature learning. Adv. Neural Inf. Process. Syst. **19**, 41 (2007)
17. Bakker, B., Heskes, T.: Task clustering and gating for bayesian multitask learning. J. Mach. Learn. Res. **4**, 83–99 (2003)

18. Mishra, M., Huan, J.: Learning task grouping using supervised task space partitioning in lifelong multitask learning. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1091–1100. ACM (2015)
19. Kumar, A., Daume III, H.: Learning task grouping and overlap in multi-task learning. ArXiv e-prints (2012)
20. Ruvolo, P., Eaton, E.: Online multi-task learning via sparse dictionary optimization. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14) (2014)
21. Ruvolo, P., Eaton, E.: Active task selection for lifelong machine learning. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI-13) (2013)
22. Wang, B., Pineau, J.: Online boosting algorithms for anytime transfer and multitask learning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
23. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with dirichlet process priors. J. Mach. Learn. Res. **8**, 35–63 (2007)