Lifelong Machine Learning guided by a Task-Relatedness Hierarchy

Neil Ashtekar
PhD student in CSE
The Pennsylvania State University
nca5096@psu.edu

Introduction

Lifelong machine learning (LML) is one of the most important open problems in artificial intelligence (AI). The goal of LML is to develop algorithms capable of learning a series of tasks sequentially such that performance on previous tasks does not degrade as new tasks are learned. This should be done efficiently, both with respect to computation and memory. LML attempts to build models which adapt to new information over time, similar to the way humans learn.

Most LML approaches learn from training data alone, and do not incorporate any form of expert or domain-specific knowledge to aid learning. This is a very difficult problem, as illustrated by the difference between LML and human learning. Humans leverage a vast collection of previous experience accumulated over many years when learning new tasks, while current LML methods are much more limited by hardware constraints. Even the largest ML models are many orders of magnitude smaller than the estimated capacity of the human brain (Wang, Liu, and Wang 2003). Until hardware catches up or entirely new paradigms of ML emerge, useful intermediate solutions are necessary.

I propose a relaxation of the LML problem in which training data is supplemented by a *task-relatedness hierarchy* indicating task similarity. This hierarchy can be used to guide the model, indicating when to make use of existing knowledge and when to learn from scratch.

Related Work

There has been relatively little previous work combining LML with expert knowledge. This is likely because LML is still in its infancy and as has yet to be widely adopted in real-world applications, while systems supplemented with expert knowledge are generally more useful in applied settings. A handful of review-style articles have explored preliminary ideas merging LML and expert knowledge.

(Hong et al. 2021) discusses human-AI collaboration in the context of LML. It mainly focuses on explainability and ways in which humans can learn from AI systems (i.e. AlphaGo), though it briefly touches on active learning with human experts used to guide LML systems. (Silver 2013) examines similarities between neurosymbolic integration and LML. It explores high-level ideas relevant in my proposed method (i.e. inductive bias, efficient knowledge transfer, expert knowledge) but does not introduce specific neurosymbolic LML algorithms. Finally, (Veron et al. 2020) introduces a novel LML problem statement in the context of natural language processing. Specifically, it focuses on the problem of question-answering using a knowledge graph, with the goal of learning over time from interactions with users.

Another set of related work focuses on hierarchical LML. (Zhang et al. 2019) proposes the hierarchical lifelong learning algorithm (HLLA) consisting of a bottom-level shared shared representation knowledge layer with top-level hypothesis testing functions. (Sun, Cong, and Xu 2018) and (Sun et al. 2021) propose "watchdog" algorithms for LML. Theses approaches consider a variation of the LML problem in which a set of candidate tasks is provided and the model is able to choose the order in which to learn them. They address this setting by choosing the task with the most information to learn, framing the problem as outlier detection (hence the name "watchdog"). The resulting models are then used to create a hierarchical knowledge library which encodes learned tasks. Methods in this category of related work either use hierarchical models or attempt to learn a task hierarchy, while my proposed approach leverages a given task-relatedness hierarchy encoding expert knowledge.

Finally, describing my proposed method requires a general understanding of LML approaches. There are three main categories of approaches to LML (Liu and Ke 2022):

- **Replay-based**: When a new task arrives, retrain the model using a combination of new data and old data (Lopez-Paz and Ranzato 2017). For memory-efficiency, store only a subset of important exemplars or use a generative model to reconstruct old data (Sun, Ho, and Lee 2019).
- **Regularization-based**: Freeze the parameters which are important to old tasks and only allow a subset of the model to learn new tasks (Kirkpatrick et al. 2017). Alternatively, use some form of knowledge distillation to efficiently store information for performing old tasks (Buzzega et al. 2020).
- **Architecture-based**: Dedicate specific submodels for each task. These submodels may be subnetworks of a neural network and may be overlapping to allow knowledge transfer (Serra et al. 2018; Wortsman et al. 2020).

Methods

At a high level, my proposed method uses expert knowledge to dynamically decide when and how to transfer previous task knowledge to new tasks. Specifically, this expert knowledge is provided in the form of a task-relatedness hierarchy. When new tasks arrive, this hierarchy can be used to determine LML hyperparameters, or at a more meta-level, to choose among lifelong learning strategies, potentially resulting in a combination of approaches (i.e. combine replay-based and architecture-based methods to retrain only a relevant portions of a network).

To make this more concrete, consider the following example. A computer vision model is to be trained in the LML setting to classify various animal species. The task-relatedness hierarchy is given as an animal taxonomy chart, listing each animal's phylum, class, order, family, genus, and species (see Figure 1). If two tasks are very similar as indicated by the taxonomy chart (i.e. first classifying brown bears vs black bears, then classifying polar bears vs pandas), then there is likely a high potential for knowledge transfer, and it would make sense to use an approach like replay. However, if two tasks are very different (i.e. first classifying brown bears vs black bears, then classifying cobras vs pythons), then there will likely be little potential for transfer and a new submodel should be learned. In this example, the similar tasks all involve animals in the same taxonomy family (bear), while the dissimilar tasks involve animals in entirely different taxonomy classes (mammal vs reptile). Note that this approach works in both the class-incremental and task-incremental LML settings, and does not require within-task data to have similar labels (i.e. one task could be classifying brown bears vs cobras – the previous examples were given for simplicity).

To realize this proposed method, several key steps must be completed:

1. Select specific techniques from the general LML categories (replay, regularization, architecture) to be used in the proposed model's implementation
2. Determine how to combine and swap between the selected LML techniques in a way that makes sense and is efficient with respect to both computation and memory
3. Formalize what it means for tasks to be similar/dissimilar, how this information should be extracted from the hierarchy, and how it should be used to dynamically choose LML techniques during training
4. Decide which LML hyperparameters should be tuned by the model based on the task-hierarchy and which hyperparameters should be tuned by the programmer
5. Implement the proposed model in a package like PyTorch and conduct experiments

Milestones

The main question to be answered through experimentation is as follows: does the use of a task-relatedness hierarchy in addition to training data improve LML model performance when compared to existing methods which only use training data? More generally, can expert knowledge be used to guide LML models?

First, a relevant application or set of applications must be chosen on which to run experiments. The previously mentioned animal classification example with taxonomy information could be used as a starting point, though there are many other interesting applications with relevant knowledge bases that could be explored (healthcare, e-commerce, etc). Note that the knowledge base could be any type of graph structure linking tasks or classes based on similarity, and is not restricted only to hierarchies. [1]

Next, a set of experiments should be run evaluating the proposed model's performance against existing models when (1) the amount of training data is fixed and (2) the amount of training data varies. This would indicate (1) if the addition of expert knowledge improves performance and (2) if the addition of expert knowledge matches performance given less training data. Additional experiments could be run to evaluate the performance of the proposed model versus existing methods with various task orderings. Understanding the effect of task order on model performance is a big open question in LML. Perhaps the use of a task-relatedness hierarchy would result in more consistent performance across various orderings.

All evaluations of the proposed method should be compared against current state-of-the-art LML methods. Typical evaluation metrics for LML should be used: performance on both new and old tasks at each step, forward and backward transfer rates, forgetting rates, learning curves, etc. Classification accuracy is not the only relevant metric – runtime and memory use should also be evaluated for all methods, both during training and at inference.

Conclusion

If this project is successful, it will serve as a promising intermediate step in LML. Success would indicate that supplementary expert knowledge is useful for LML – this is notably since prior work rarely incorporates expert knowledge. In addition, the proposed method applies the idea of ensembles to LML at a more meta-level than previous approaches, combining very different techniques dynamically during training based on task similarity. This is an interesting idea that could be applicable to general LML even without expert knowledge.

An extension of this project could focus on knowledge extraction from other data sources to create an expert knowledge base. Instead of assuming that a task-relatedness hierarchy is given, such a hierarchy could be learned from auxiliary data, further automating the training process.

[1] Hierarchies seemed to be the most obvious fit for most applications – animal taxonomy, International Classification of Diseases (ICD) codes, etc – hence the title and focus of the project

Figure 1: Subset of an animal taxonomy chart from kingdom (top) to species (bottom) (Dorling and Kindersley 2022)

References

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems* 33:15920–15930.

Dorling, and Kindersley. 2022. Classifying animals. Website.

Hong, X.; Guan, S.-U.; Wong, P. W.; Xue, N.; Man, K. L.; and Liu, D. 2021. Can ai teach humans? humans ai collaboration for lifelong machine learning. In *2021 4th International Conference on Data Science and Information Technology*, 427–432.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114(13):3521–3526.

Liu, B., and Ke, Z. 2022. Lifelong and continual learning. Course Lecture.

Lopez-Paz, D., and Ranzato, M. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30.

Serra, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, 4548–4557. PMLR.

Silver, D. L. 2013. On common ground: Neural-symbolic integration and lifelong machine learning. In *Proceedings of the 9th International Workshop on Neural-Symbolic Learning and Reasoning NeSy13*, 41–46.

Sun, G.; Cong, Y.; Gu, C.; Xu, T.; Ding, Z.; and Yu, H. 2021. Hierarchical lifelong machine learning with watchdog. *IEEE Transactions on Big Data*.

Sun, G.; Cong, Y.; and Xu, X. 2018. Active lifelong learning with “watchdog”. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Sun, F.-K.; Ho, C.-H.; and Lee, H.-Y. 2019. Lamol: Language modeling for lifelong language learning. *arXiv preprint arXiv:1909.03329*.

Veron, M.; Peñas, A.; Echegoyen, G.; Banerjee, S.; Ghannay, S.; and Rosset, S. 2020. A cooking knowledge graph and benchmark for question answering evaluation in lifelong learning scenarios. In *International Conference on Applications of Natural Language to Information Systems*, 94–101. Springer.

Wang, Y.; Liu, D.; and Wang, Y. 2003. Discovering the capacity of human memory. *Brain and Mind* 4(2):189–198.

Wortsman, M.; Ramanujan, V.; Liu, R.; Kembhavi, A.; Rastegari, M.; Yosinski, J.; and Farhadi, A. 2020. Supermasks in superposition. *Advances in Neural Information Processing Systems* 33:15173–15184.

Zhang, T.; Su, G.; Qing, C.; Xu, X.; Cai, B.; and Xing, X. 2019. Hierarchical lifelong learning by sharing representations and integrating hypothesis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51(2):1004–1014.