Requirements for Machine Lifelong Learning

Daniel L. Silver*
Jodrey School of Computer Science
Acadia University
Wolfville, NS, Canada B4P 2R6
danny.silver@acadiau.ca

Ryan Poirier
Jodrey School of Computer Science
Acadia University
Wolfville, NS, Canada B4P 2R6
045434p@acadiau.ca

Abstract

A significant advance in inductive modelling are systems that retain learned knowledge and selectively transfer portions of that knowledge as a source of inductive bias. We define such to be machine lifelong learning (ML3) systems. This paper makes an initial effort at specifying the scope of ML3 systems and their functional requirements.

1 Introduction

Over the last ten years progress has been made in machine learning and statistical modelling that exhibit aspects of knowledge retention and inductive transfer. These represent advances in inductive modelling that move beyond *tabula rasa* learning and toward machines capable of lifelong learning [15]. Henceforth, this article will refer to such as machine lifelong learning (ML3) systems. Despite the progress that has been made, there is need for a clear definition of the knowledge retention and inductive transfer problem. Toward that end, this paper makes an initial effort at specifying the scope of ML3 systems and their functional requirements.

2 Scope of ML3 Systems

In [12, 13] *knowledge-based inductive learning* is defined as an ML3 approach that uses knowledge of the task domain as a source of inductive bias. As with a standard inductive learner, training examples are used to develop a hypothesis of a classification task. However, unlike a standard learning system, knowledge from each hypothesis is saved in a long-term memory structure called domain knowledge. When learning a new task, aspects of domain knowledge are selected to provide a positive inductive bias to the learning system. The result is a more accurate hypothesis developed in a shorter period of time. The method relies on the transfer of knowledge from one or more prior secondary tasks, stored in domain knowledge, to the hypothesis for a new primary task. The problem of selecting an appropriate bias becomes one of selecting the most related task knowledge for transfer.

An ML3 system is typically composed of short-term and long-term components and/or exhibits short-term and long-term processes. Although two phases of learning may not be necessary, it is frequently required so as to ensure that long-term domain knowledge is not corrupted by inaccurate short-term learning. The following three sections outline general requirements for ML3 systems and specific requirements for long-term retention of learned knowledge and short-term learning with inductive transfer.

*http://birdcage.acadiau.ca:8080/ml3

3 General Requirements

3.1 Form of knowledge retention

Learned knowledge can be stored in functional or representational form within a ML3 [12]. The simplest method of retaining task knowledge in functional form is to save the respective training examples. Other methods of retaining functional knowledge involve the storage or modelling of search parameters such as the learning rate in neural networks. An advantage of retaining functional knowledge, particularly the retention of the actual training examples, is the accuracy and purity of the knowledge. Disadvantages of retaining functional knowledge are the large amount of storage space that it requires and difficulties in using such knowledge during future learning.

Alternatively, a description of an accurate hypothesis developed from the training examples can be retrained. We define this to be a representational form of knowledge retention. The description of a decision tree or a neural network are examples of representations. The advantages of retaining representational knowledge is its compact size relative to the space required for the original training examples and its ability to generalize beyond those examples. The disadvantage of retaining representational knowledge is the potential loss of accuracy from the original training examples.

3.2 Form of knowledge transfer

The form in which task knowledge is retained can be separated from the form in which it is transferred. For example, the retained hypothesis representation for a learned task can be used to generate functional knowledge in the form of training examples [8, 13].

Representational transfer involves the direct or indirect assignment of known task representation to the model of a new target (or primary) task [12]. In this way the learning system is initialized in favour of a particular region of hypothesis space of the modeling system [7, 9, 14]. Representational transfer often results in substantially reduced training time with no loss in the generalization performance of the resulting hypotheses.

In contrast to representational transfer, functional transfer employs the use of implicit pressures from training examples of related tasks [1], the parallel learning of related tasks constrained to use a common internal representation [2, 3], or the use of historical training information from related tasks [15, 4]. These pressures reduce the effective hypothesis space in which the learning system performs its search. This form of transfer has its greatest value in terms of increased generalization performance from the resulting hypotheses.

3.3 Input and output type, complexity and cardinality

The output representation of a system capable of retaining and transferring knowledge should not be constrained to a particular data type. A ML3 system should be capable of predicting class categories and real-value outputs including scalar values as well as vectors.

An ML3 should be capable of dealing with its environment over a lifetime with a fixed number of inputs and outputs for the task domain(s) under study. Certain inputs or outputs might go unused for many tasks of a domain early in the learning system's lifetime only to be used quite frequently later in life. The rationale for this requirement is not to constrain an ML3 system to a fixed amount of internal representation (this could change over time) but to ensure a consistent interface with the environment and with other entities such as a software agent, a application program or a human user.

3.4 Scalability

A ML3 system must be capable of scaling up to large numbers of inputs, outputs, training examples and learning tasks. Preferably, both the space and time complexity of the learning system grows polynomially in all three of these factors.

3.5 Accumulation of Practice

A ML3 system should facilitate the practice of a task. The system's normal methods should retain and transfer knowledge from one learning episode of a task to another such that the generalization accuracy of the long-term hypothesis for the task increases. But, how can a ML3 system determine from the training examples that it is practicing a task it has previously learned versus learning a new but closely related task [5, 10]. We have come to the conclusion that a ML3 system should not have to be explicit in this determination. Rather, the similarity, or relatedness, of a set of training examples to that of prior domain knowledge should be implicit; each training example should be able to draw upon those aspects of domain knowledge that are most related. This suggests that domain knowledge should be seen as continuum as apposed to a set of disjoint tasks.

4 Requirements for Long-term Retention of Learned Knowledge

4.1 Effective retention

A ML3 system should resist the introduction and accumulation of domain knowledge error. Only hypotheses with an acceptable level of generalization accuracy should be retained else, once saved in long-term memory, the error from a hypothesis may be transferred to future hypotheses. A ML3 system must be concerned with this systemic growth in error over its lifetime. Similarly, The process of retaining a new hypothesis should not reduced its accuracy or that of prior hypotheses existing in long-term memory. In fact, the integration or consolidation of new task knowledge should increase the accuracy of related prior knowledge.

4.2 Efficient retention

A ML3 system should be efficient in its use of long-term memory (efficient in space). In particular, the system should make use of memory resources such that the duplication of information is minimized. A representational form of task knowledge will be more space efficient than a functional form because of the reasons cited in Section 3.1. A ML3 system should also be computationally efficient (efficient in time) when storing learned knowledge in long-term memory. Ideally, retention should occur during short-term learning, however, in order to ensure effective retention (reduction of error) this is rarely possible.

4.3 Effective indexing

A ML3 must be capable of selecting the appropriate prior knowledge for inductive transfer during short-term learning. This requires that a ML3 be capable of indexing into long-term memory for task knowledge that is most related to the primary task. Typically, primary task knowledge will arrive in the form of training examples (functional knowledge) and no representational knowledge will be provided. This requires design choices in the construction of the ML3 system. The system must either use functional examples to select related domain knowledge or generate a hypothesis representation for the primary task to estimate its similarity to existing domain knowledge representation.

4.4 Efficient indexing

A ML3 system must make the selection of related knowledge as rapid as possible. Preferably, the computational time for indexing into domain knowledge should be no worse than polynomial in the number of tasks having been stored. Experimentation has shown that a representational form of retained knowledge (*e.g.*weights of a neural network) can be more efficiently indexed than a functional form (*e.g.*examples used to train the network) [6].

4.5 Meta-knowledge of the task domain

In most cases, it will be necessary for a ML3 system to determine and retain meta-knowledge of the task domain. For example, it may be necessary to estimate the probability distribution over the input space so as to manufacture appropriate functional examples from retained task representation [13]. Alternatively, it may be necessary to retain characteristics of the learning process (learning curve, error rate) for each task.

5 Requirements for Short-term Learning with Inductive Transfer

5.1 Effective learning

The inductive transfer (bias) from long-term memory should never decrease the generalization performance of a hypothesis developed by a ML3 system. A ML3 system should produce a hypothesis for the primary task that meets or exceeds the generalization performance of that developed strictly from the training examples. There is evidence that the functional form of knowledge transfer somewhat surpasses that of representation transfer in its ability to produce more accurate hypotheses [3, 11]. Starting from a prior representation can limit the development of novel representation required by the hypothesis for the primary task. In terms of neural networks this representational barrier manifests itself in terms of local minimum.

5.2 Efficient learning

Inductive transfer from long-term memory should not increase the computational time for developing a hypothesis for the primary task as compared to using only the training examples. In fact, inductive transfer should reduce training time. In practice this reduction is rarely observed because of the computation required to index into prior domain knowledge. In terms of memory (space), there will typically be an increase in complexity as prior domain knowledge must be used during the learning of the new task. Our research has shown that a representational form of knowledge transfer will be more efficient than a functional form (supplemental training examples) [11].

Sections 4.3, 4.4, 5.1 and 5.2 indicate an interesting dichotomy between effective and efficient inductive transfer. Effective learning requires functional transfer whereas efficient learning requires representation transfer.

5.3 Transfer versus training examples

A ML3 must take into consideration the estimated sample complexity and number of available examples for the primary task and the generalization accuracy and relatedness of retained knowledge in long-term memory. During the process of inductive transfer a ML3 must weigh the relevance and accuracy of retained knowledge along side that of the information resident in the training examples.

6 Conclusion

This paper has outlined the scope and functional requirements for a ML3 system. A ML3 system can retain and transfer knowledge in either representational or functional form. A ML3 system should have no bounds on input and output variable type and complexity and it should be scalable in terms of number of inputs, outputs, number of training examples and learning tasks. A ML3 should facilitate the practice of a task and treat domain knowledge as a continuum of tasks rather than a set of disjoint tasks.

Efficient long-term retention of learned knowledge should cause no loss of prior task knowledge, no loss of new task knowledge, and an increase in the accuracy of old tasks if the new task being retained is related. A ML3 must be capable of efficiently selecting the most effective prior knowledge for inductive transfer during short-term learning.

Efficient short-term learning with inductive transfer should produce a hypothesis for a primary task that meets or exceeds the generalization performance of a hypothesis developed from only the training examples. Experimental results indicate that effective learning excels under functional transfer whereas efficient learning requires representation transfer. Lastly, we point out that a ML3 must weigh the relevance and accuracy of retained knowledge along side that of the available training examples for the primary task.

References

[1] Yaser S. Abu-Mostafa. Hints. *Neural Computation*, 7:639–671, 1995.

[2] Jonathan Baxter. Learning internal representations. *Proceedings of the Eighth International Conference on Computational Learning Theory*, 1995.

[3] Richard A. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[4] D.K. Naik and Richard J. Mammone. Learning by learning in neural networks. *Artificial Neural Networks for Speech and Vision*, 1993.

[5] Robert O'Quinn. *Knowledge Transfer in Artificial Neural Networks*. Honours Thesis, Jodrey School of Computer Science, Acadia University, Wolfville, NS, 2005.

[6] Ryan Poirier and Daniel L. Silver. Effect of curriculum on the consolidation of neural network task knowledge. *Proc. of IEEE International Joint Conf. on Neural Networks (IJCNN 05)*, 2005.

[7] Mark Ring. Learning sequential tasks by incrementally adding higher orders. *Advances in Neural Information Processing Systems 5*, 5:155–122, 1993. ed. C. L. Giles and S. J. Hanson and J.D. Cowan.

[8] Anthony V. Robins. Catastrophic forgetting, rehearsal, and pseudorehearsal. *Connection Science*, 7:123–146, 1995.

[9] Jude W. Shavlik and Thomas G. Dietterich. *Readings in Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[10] Daniel L. Silver and Ricchard Alisch. A measure of relatedness for selecting consolidated task knowledge. *Proceedings of the 18th Florida Artificial Intelligence Research Society Conference (FLAIRS05)*, pages 399–404, 2005.

[11] Daniel L. Silver and Peter McCracken. Selective transfer of task knowledge using stochastic noise. In Yang Xiang and Brahim Chaib-draa, editors, *Advances in Artificial Intelligence, 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI'2003)*, pages 190–205. Springer-Verlag, 2003.

[12] Daniel L. Silver and Robert E. Mercer. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. *Connection Science Special Issue: Transfer in Inductive Systems*, 8(2):277–294, 1996.

[13] Daniel L. Silver and Robert E. Mercer. The task rehearsal method of life-long learning: Overcoming impoverished data. *Advances in Artificial Intelligence, 15th Conference of the Canadian Society for Computational Studies of Intelligence (AI'2002)*, pages 90–101, 2002.

[14] Satinder P. Singh. Transfer of learning by composing solutions for elemental sequential tasks. *Machine Learning*, 1992.

[15] Sebastian Thrun. Lifelong learning algorithms. *Learning to Learn*, pages 181–209.1 Kluwer Academic Publisher, 1997.