*cs*MTL: a Context Sensitive Lifelong Learning System

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

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

Abstract

*cs*MTL, or *context-sensitive* Multiple Task Learning, is presented as a method of inductive transfer that uses a single output neural network and additional contextual inputs for learning multiple tasks. The *cs*MTL approach is demonstrated to produce hypotheses that are equivalent to or better than standard MTL hypotheses when learning a primary task in the presence of related and unrelated tasks. The paper also describes a machine lifelong learning system based on *cs*MTL for sequentially learning multiple tasks. The approach satisfies a number of important requirements for knowledge retention and inductive transfer; taking advantage of representational transfer for rapid short-term learning and functional transfer for long-term consolidation.

1 Introduction

Multiple task learning (MTL) neural networks are one of the better documented methods of inductive transfer of task knowledge [3, 8]. An MTL network is a feed-forward multi-layer network with an output for each task that is to be learned. The standard back-propagation of error learning algorithm is used to train all tasks in parallel. Consequently, MTL training examples are composed of a set of input attributes and a target output for each task. Figure 1 shows a simple MTL network containing a hidden layer of nodes that are common to all tasks. The sharing of internal representation is the method by which inductive bias occurs within an MTL network [2]. The more that tasks are related, the more they will share representation and create positive inductive bias.

We have investigated the use of MTL networks as a basis for developing a machine lifelong learning (ML3) system [9, 10, 6] and have found them to have several limitations related to the multiple outputs of the network. First, and foremost, is the problem of measuring task relatedness. Previous work on MTL, including our own, is based on the premiss that knowledge is shared at the task level and that optimal inductive transfer occurs between related tasks. This task-level perspective does not consider the sharing of knowledge and inductive transfer at the example level. Consider two concept tasks where only half of the MTL training examples have the same target class value. From a task-level perspective

*http://plato.acadiau.ca/courses/comp/dsilver/

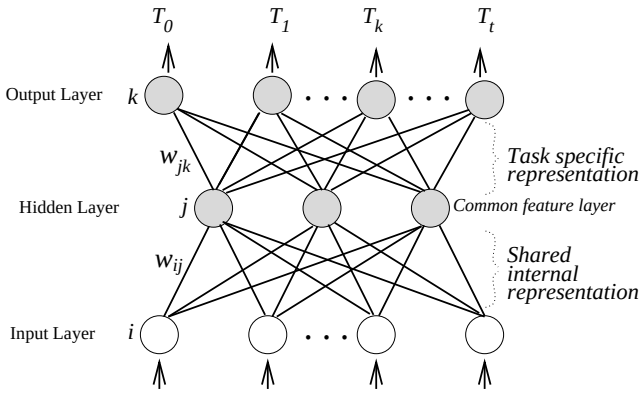

Figure 1: A multiple task learning (MTL) network with an output node for each task being learned in parallel.

the two tasks would be unrelated by most statistical and information theoretic measures. However, from an example-level perspective, the tasks are partially related in that half of their examples are identical. Perhaps, relatedness should be judged at this finer level of detail. A second problem in using MTL for lifelong learning is that it can generate redundant outputs and representation of the same task that must be explicitly managed. This is a particular problem when one considers that a lifelong learning system should be capable of practising a task (acquiring new examples of the same task over time). This build up of redundant outputs and representation increases the problem of indexing into prior knowledge for inductive transfer. Lastly, there is the practical problem of how a lifelong learning agent would know to add a new task output to an MTL based system. Clearly, the learning environment should provide the contextual queues, however this suggests additional inputs and not outputs.

In response to these problems, this paper introduces *context sensitive* MTL, or *cs*MTL, as a method of inductive transfer. *cs*MTL is based on standard MTL with two major differences; only one output is used for all tasks and additional inputs are used to indicate the example *context*, such as the task to which it is associated. The following section describes the *cs*MTL network. Section 3 presents a ML3 system based on a *cs*MTL network.

2 *cs*MTL

Figure 2 presents the *cs*MTL network. It is a feed-forward network architecture of input, hidden and output nodes that uses the back-propagation of error training algorithm. The *cs*MTL network requires only one output node O for learning concept tasks (more outputs could be used for predicting a vector of values). Similar to standard MTL neural networks, there is one or more layers of hidden nodes that act as feature detectors. The input layer can be divided into two parts: nodes $I_0 ... I_n$ correspond to *primary* input variables for the tasks; nodes $T_0 .. T_k$ provide the network with the *context* of each training example. The context inputs can simply be a set of task identifiers that associate each training example to a particular task. Alternatively, they can offer more specific environmental information (such as location and light level) and in this way index over a continuous domain of tasks.

A training example for a *cs*MTL network is the form (T, I, O); where I is the vector of primary input values, T is the vector of context values and O is the desired class label. The values for the vector T are set to associate each example with a particular task. Therefore, a training set for multiple task learning is a concatenation of standard training examples for

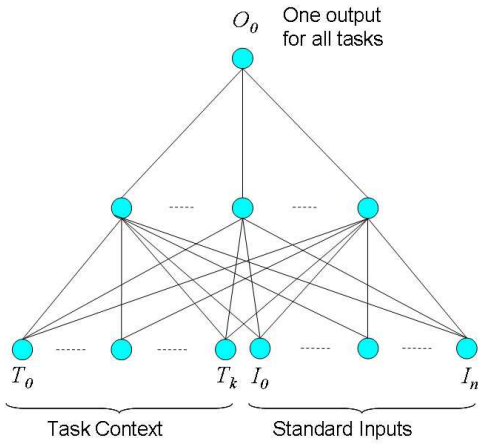

Figure 2: Proposed system: *cs*MTL

each task that contain the additional context inputs. When training the *cs*MTL network, a tuning set of examples for all tasks must be used to prevent over-fitting.

With *cs*MTL the focus shifts from learning tasks of a domain to learning a domain of tasks. The entire representation of the network is used in common to develop hypotheses for all tasks. This presents a more continuous sense of domain knowledge and the objective becomes that of learning internal representations that are helpful to related tasks and index by a combination of the primary and context inputs. We are currently examining the theory of Hints [1] for direction on formalizing the notion that each separate task can be seen as a Hint that reduces the VC dimension for learning the internal representation of related tasks within the domain. Related work on context-sensitive machine learning can be found in [12] and other papers from the ICML'96 Workshop on *Learning in Context-Sensitive Domains*.

3 A Machine Lifelong Learning System based on *cs*MTL

Figure 3 shows the proposed *cs*MTL ML3 system. It has two components; a temporary *short-term learning network* and a permanent *long-term consolidation cs*MTL *network*. The long-term *cs*MTL network is the location in which domain knowledge is retained over the lifetime of the learning system. The weights of this network are updated only after a new task has been trained to an acceptable level of accuracy in the short-term learning network. The short-term network can be considered a temporary extension of the long-term network that adds representation (several hidden nodes and a output node). At the start of short-term learning the weights associated with these temporary nodes are initialized to small random weights while the weights of the long-term network are frozen. This allows representational knowledge to be rapidly transferred from related tasks existing in the long-term network without fear of loosing prior task accuracies.

Once the new task has been learned, the temporary short-term network is used to consolidate knowledge of the task into the permanent long-term network. This is accomplished by using a form of functional transfer call *task rehearsal* [9]. The method uses the short-term network to generate *virtual examples* for the new tasks so as to slowly integrate (via back-propagation) the task's knowledge into the long-term network. Additional, virtual examples for the prior tasks are used during consolidation to maintain the existing knowledge of the long-term network. Note that it is the functional knowledge of the prior tasks that

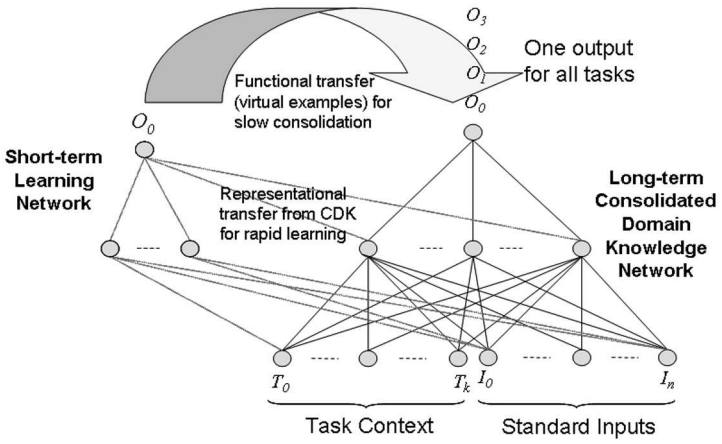

Figure 3: Proposed system: *cs*MTL

must be retained and not their representation; the internal representation of the long-term network will necessarily be updated as the new task in integrated.

Algorithm for short-term learning:

- Fix the representation of the long-term *cs*MTL network;
- Add one or more temporary hidden nodes and an output node (fully feed-forward connected);
- Initialize associated connection weights for the new nodes to small random values;
- Train and test short-term network using available data; and
- If generalization accuracy is sufficient, consolidate into long-term network.

Algorithm for long-term consolidation:

- Generate virtual examples for new and prior tasks using existing representation of the short-term and long-term networks;
- Remove temporary short-term network components;
- Unfix the representation of the long-term network; and
- Train the long-term network using the virtual examples.

4 Meeting the Requirements of a Lifelong Learning System

We have recently outlined a set of requirements for a ML3 system [11]. The following uses this list of requirements as a basis for discussing the benefits and limitations of the proposed *cs*MTL method.

4.1 Requirements for Long-term Retention of Learned Knowledge.

Effective and efficient retention. Knowledge retention in the *cs*MTL system is the result of consolidation of new and prior task knowledge in the long-term network using a form of functional transfer called task rehearsal [9]. Task rehearsal overcomes the stability-plasticity problem originally posed by [5] taken to the level of learning sets of tasks as

opposed to learning sets of examples [7, 4]. Consolidation of new task knowledge without loss of existing task knowledge is possible given sufficient number of training examples, sufficient internal representation for all tasks, slow training using a small learning rate and a method of early stopping to prevent over-fitting and therefore the growth of high magnitude weights[10]. In the long-term *cs*MTL network there will be an effective and efficient sharing of internal representation between related tasks, as in the case of an MTL network, without the disadvantage of having duplicate representation of identical or near identical task outputs.

The *cs*MTL approach suffers from the scaling problems of all back-propagation neural networks. The computational complexity of the standard back-propagation algorithm is $O(W^3)$, where W is the number of weights in the network. Long-term consolidation will be computationally more expensive than standard MTL because the additional contextual inputs will increase the number of weights in the network at the same rate as MTL and it may be necessary to add an additional layer of hidden nodes for certain task domains. The rehearsal of each of the existing domain knowledge tasks requires the creation and training of $m \cdot k$ virtual examples, where m is the number of virtual training examples per task and k is the number of tasks. An important benefit from consolidation is an increase in the accuracy of related hypotheses existing in the *cs*MTL network as a new task is integrated.

Accumulation of practice. Because the long-term network has only one permanent output there are no redundant representations for the same task. Over time, more detailed practice sessions for the same task will contribute to the development of a more accurate long-term hypothesis. Learning closely related tasks will fill in useful knowledge of the domain. In fact, the *cs*MTL network can represent a fluid domain of tasks where subtle differences between tasks can be represent by small changes in the context inputs.

Effective and efficient indexing. Our conjecture is that *cs*MTL does not require an explicit method of indexing into domain knowledge for related tasks. Instead, the internal representation of all tasks saved in the long-term network is used (held fixed) as a portion of the new hypothesis. Indexing occurs as the connection weights between the long-term network and the temporary short-term network are trained.

Meta-knowledge of the task domain. Meta-knowledge of the task domain is required by *cs*MTL during long-term consolidation. The virtual examples used for task rehearsal of the new and prior tasks should be generated based on the estimated probability distribution of training examples over the input space.

4.2 Requirements for Short-term Learning with Inductive Transfer

Effective and efficient learning. The *cs*MTL system uses a temporary short-term learning network with representational transfer from the long-term consolidation network. This form of transfer will be very efficient and effective. If the current task has been previously learned and retained, then the weights between the long-term network and the short-term network will train quickly to produce the desired output. If the new task is different but related to a prior task, the long-term to short-term network weights will select the most appropriate features of domain knowledge and the supplemental hidden nodes of the short-term network will play only a partial role in the hypothesis. If the new task is unrelated to any prior learning, the supplemental internal representation of the short-term network will play the major role in the new hypothesis.

Although the computational cost of long-term consolidation is high, the benefit is that a hypothesis for a new but related task can be quickly developed in the short-term network. The reasons for this are: the existing internal representation of the long-term network can be used to develop the hypothesis, only the new task training examples are required, and there are relatively few weights in the temporary short-term network to be trained.

Transfer versus training examples. An inductive transfer system should produce a hypothesis for the primary task that meets or exceeds the generalization performance of hypotheses developed strictly from the training examples. The experiments reported in this paper use the standard back-propagation algorithm for training the short-term learning network. We have found that this algorithm will automatically take advantage of related features in the long-term consolidated network, if they exist, prior to developing new internal features for a task. An additional experiment is planned to test the benefits of learning the hidden-to-output weights from the long-term consolidated network first and then learning the remaining short-term weights to reduce the residual error.

5 Experimentation

This section reports on a set of initial experiments that compares the ability of *cs*MTL to transfer knowledge with MTL and ηMTL, a variant of MTL that selects the most related task knowledge based on correlation of the target outputs. All experiments use a *cs*MTL a network as described in section 2. A second set of experiments that use a lifelong learning system with short-term and long-term components, as described in section 3, will be reported in a future article.

Three domains have been studied using *cs*MTL. The *Band domain*, described in [9], consists of seven synthetic tasks. Each task has a band of positive examples across a 2-dimensional input space. The tasks were synthesized so that the primary task T_0 would vary in its relatedness to the other tasks based on the band orientation. The *Logic domain* consists of six synthetic tasks. Each positive example is defined by a logical combination of 4 of the 10 real-valued inputs of the form, $T_0 : (I_0 > 0.5 \vee I_1 > 0.5) \wedge (I_2 > 0.5 \vee I_3 > 0.5)$. The tasks of this domain are more or less related in that they share zero, one or two features such as $(I_0 > 0.5 \vee I_1 > 0.5)$ with the other tasks. The Band and Logic domains have been designed so that all tasks are non-linearly separable; each task requires the use of at least two hidden nodes of a neural network to form an accurate hypothesis. The *fMRI domain* challenges the learning systems to develop models that can classify 24 features extracted from fMRI images as a subject reading a sentence or viewing a picture[1]. Inductive transfer between two subject models is examined; from subject T_0 for which good models could be developed to a second subject T_1 for which only poor models could be developed.

5.1 Method

A *cs*MTL network was configured for each domain with one output node, a layer of hidden nodes (30 for the Band, 20 for the Logic and 10 for the fMRI domain) and a layer of input nodes. The Band domain has 9 inputs, 2 represent the coordinates of the 2-dimensional input space and the remaining 7 provide the *context*, that is, they indicate the task to which each example belongs. The Logic domain has 16 inputs, 10 represent the primary values for logical expression and the remaining 6 provide the task *context*. The fMRI domain has 26 inputs, where 24 are used to represent the activity level of a region of interest in the subjects brain and the remaining 2 select subject A or B. For all domains

For all three domains, the objective is to learn task T_0 using an impoverished training set of examples (10 for the Band, 20 for the Logic and 48 for the fMRI domain) for which single task learning (STL) does poorly. Each of the other tasks of the domain have 48 or more training examples that have been demonstrated to develop models with accuracies greater than .75 using a STL network. A tuning set of examples is used to prevent over-fitting on each domain. An independent test set (200 for the Band, 500 for the Logic and 24 for the fMRI domain) was used to determine hypothesis performance. The mean accuracies

[1]Courtesy of the Brain Image Analysis Research Group, CALD, Carnegie Mellon University.)

reported below are from repeated studies (10 for the Band, 15 for the Logic domain and 5 for the fMRI domain).

5.2 Results

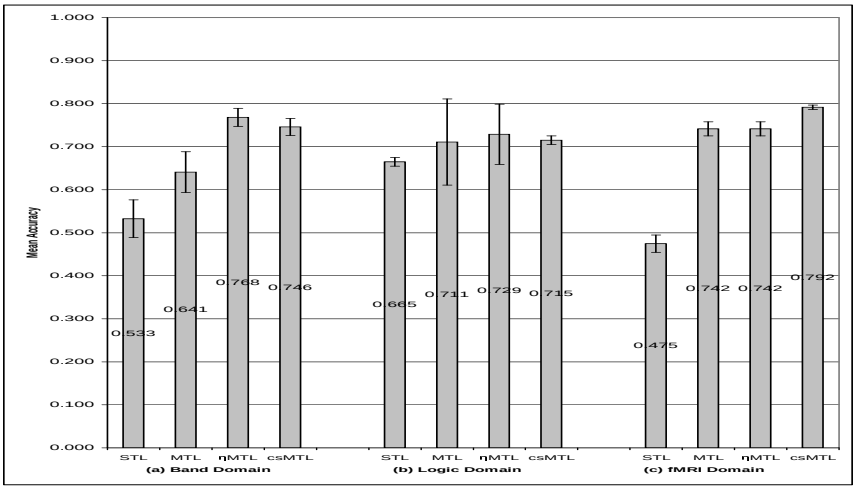

Figure 4: *cs*MTL compared to STL and previous MTL methods. Shown is the mean test set accuracy for T_0 hypotheses for the three domains of tasks.

Figure 4 shows the results for the three domains. It compares the mean accuracy of the hypotheses developed for the T_0 tasks with *cs*MTL to no inductive transfer under STL, transfer with standard MTL, and selective transfer with ηMTL. The training examples for T_0 in each domain are insufficient to develop hypotheses with accuracies above .65. The MTL and ηMTL results demonstrate the advantage of knowledge transfer with mean accuracies that are significantly better in the case of the Band and fMRI domains. *cs*MTL does significantly better than STL on all domains, significantly better than MTL on the Band and fMRI domains and equal in performance to ηMTL on all domains. The results indicate that *cs*MTL is able to selectively transfer knowledge from the shared internal representation of related tasks to a new task when training on examples of *all* prior tasks.

6 Conclusion

This paper has presented *cs*MTL as a method of inductive transfer that uses a single output neural network and additional *context* inputs for learning multiple tasks. The method was developed in response to problems we had encounter in using MTL networks for developing machine lifelong learning systems. The question of how an example is associated with one task versus another is solved by the context inputs. The operator (or the environment) can provide these contextual queues with each example. The method eliminates the build-up of redundant task representation that can frustrate the search for related prior knowledge. Lastly, and perhaps most importantly, the *cs*MTL approach shifts the focus from learning tasks of a domain to learning a domain of tasks where the context inputs can be seen as indexing over that domain at the example level as opposed to the task level. Our conjecture is that this approach avoids the issue of having to measure the relatedness

between tasks in order to ensure a positive inductive bias. Similar examples of the primary task will collaborate with similar examples of related tasks to build mutually beneficial internal representation. Dissimilar examples will work to develop unique representations that capture the subtleties of the individual tasks. Experimentation on three different domains of tasks has demonstrated that *cs*MTL can produce hypotheses that are equivalent to or better than standard MTL hypotheses when learning a primary task in the presence of related and unrelated tasks.

The paper also describes a machine lifelong learning (ML3) system based on *cs*MTL that is capable sequential knowledge retention and inductive transfer. The system is meant to satisfy a number of ML3 requirements including the effective consolidation of task knowledge into a long-term network using task rehearsal, the accumulation of task knowledge from practice sessions, effective and efficient inductive transfer during new learning, and the tradeoff between inductive transfer and training examples during new learning. In future work, we plan to formalize the *cs*MTL approach in terms of previous work on *Hints* and *context-sensitive* machine learning. Our intention is to construct a *cs*MTL ML3 system and conduct experiments on other informative synthetic and real-world domains so as to more fully explore the approach in light of the formal theory.

References

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

[2] Jonathan Baxter. Learning model bias. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 169–175. The MIT Press, 1996.

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

[4] Robert M. French. Pseudo-recurrent connectionist networks: An approach to the sensitivity-stability dilemma. *Connection Science*, 9(4):353–379, 1997.

[5] Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–64, 1987.

[6] Robert O'Quinn, Daniel L. Silver, and Ryan Poirier. Continued practice and consolidation of a learning task. In *Proceedings of the Meta-Learning Workshop, 22nd International Conference on Machine Learning (ICML 2005)*, Bonn, Germany, 2005.

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

[8] Daniel L. Silver and Robert E. Mercer. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. *Learning to Learn*, pages 213–233, 1997.

[9] 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.

[10] Daniel L. Silver and Ryan Poirier. Sequential consolidation of learned task knowledge. *Lecture Notes in Artificial Intelligence, 17th Conference of the Canadian Society for Computational Studies of Intelligence (AI'2004)*, pages 217–232, 2004.

[11] Daniel L. Silver and Ryan Poirier. Requirements for machine lifelong learning. *Jodrey School of Computer Science, TR-2005-009*, November 2005.

[12] Peter D. Turney. The identification of context-sensitive features: A formal definition of context for concept learning. In NRC 39222, editor, *13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains*, volume NRC 39222, pages 53–59, Bari, Italy, 2005. NRC 39222.