THE CONSOLIDATION OF NEURAL NETWORK TASK KNOWLEDGE

Daniel L Silver and Peter McCracken
Acadia University
Wolfville, Nova Scotia, Canada B4P 2R6
danny.silver@acadiau.ca

Abstract: A fundamental question of any life-long learning system is addressed: How can task knowledge be consolidated within a long-term domain knowledge structure for efficient storage and for more efficient and effective transfer of that knowledge when learning a new task. A review of relevant background material on knowledge based inductive learning and the sequential transfer of task knowledge using multiple task learning (MTL) neural networks is presented. A theory of task knowledge consolidation is proposed that uses a large MTL network as the domain knowledge structure and task rehearsal as a method of overcoming the catastrophic forgetting problem. The theory is tested on a synthetic domain of seven tasks and it is shown that task knowledge can be sequentially consolidated within a domain knowledge MTL network both effectively and efficiently.

Keywords: knowledge consolidation, neural networks, sequential learning, life-long learning

1 INTRODUCTION

The majority of machine learning research has focused on the single task learning approach where an hypothesis for a single task is induced from a set of training examples with no regard to previous learning or to the retention of task knowledge for future learning. In contrast, humans take advantage of previous learning by retaining task knowledge and transferring this knowledge when learning a new and related task. Life-long learning is a relatively new area of machine learning research concerned with the persistent and cumulative nature of learning [1]. Life-long learning considers situations in which a learner faces a series of different tasks and develops methods of retaining and using task knowledge to improve the effectiveness (more accurate hypotheses) and efficiency (shorter training times) of learning. Our research investigates methods of knowledge retention and transfer within the context of artificial neural networks and applies these methods to life-long learning problems, such as learning accurate medical diagnostic models from small samples of a patient population [2].

One of the fundamental problems in developing a life-long learning system is devising a method of retaining task knowledge in an efficient and effective manner such that it can be later used when learning a new task. This requires the integration or consolidation of new task knowledge with previously learned task knowledge within a domain knowledge structure. This paper presents a theory of task knowledge consolidation within the context of multiple task learning (MTL) neural networks and tests that theory on a synthetic domain of tasks. The results indicate that it is possible to sequentially consolidate task knowledge provided there are sufficient numbers of training examples and that task rehearsal is used to overcome the stability-plasticity problem of neural networks that leads to catastrophic forgetting of previous task knowledge.

2 BACKGROUND

2.1 Knowledge Based Inductive Learning. The constraint on a learning system's hypothesis space, beyond the criterion of consistency with the training examples, is called inductive bias [3]. For example, Occam's Razor suggests a bias for simple over more complex hypotheses. Inductive bias is essential for the development of an hypothesis with good generalization from a practical number of examples. Ideally, a life-long learning system can select its inductive bias to tailor the preference for hypotheses according to the task being learned. One type of inductive bias is prior knowledge of the task domain [4]. The retention and use of task domain knowledge as a source of inductive bias remains an open problem in machine learning [1,5].

We define knowledge-based inductive learning (KBIL) as a learning method 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 an hypothesis for classifying future examples. Information from previously learned tasks is accumulated within a domain knowledge store. The intent is that aspects of domain knowledge can be selected to provide a positive inductive bias to the inductive 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 secondary tasks, stored in domain knowledge, to a new primary task. Therefore, the problem of selecting an appropriate bias becomes one of selecting the appropriate task knowledge for transfer. Much of our prior work has focused on knowledge transfer and the measurement of task relatedness [6,7,8].

Following the successful learning of a new task, information regarding that task is retained and consolidated within domain knowledge. Two major questions concerning knowledge-based inductive learning can be asked: (1) In what form should
previously learned task knowledge be retained within domain knowledge? and (2) In what form should task knowledge from domain knowledge be transferred? The following reviews these questions.

2.2 Knowledge Retention. The simplest method of retaining task knowledge is to save all the training examples for the task. In [6] we define the training examples to be a functional form of task knowledge. Other methods of retaining functional knowledge of a task involve the storage or modeling 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. A disadvantage of retaining functional knowledge is the large amount of storage space that it requires.

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. The representation of a hypothesis involves a description of the representational language (architecture of the neural network) and the values of the free parameters used by that representation (the weights of the connections between neurons). 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 loss of purity and potential loss of accuracy from the original training examples.

2.3 The Need for Consolidated Domain Knowledge. Knowledge retention is necessary for a knowledge-based inductive learning system, however, retention is not sufficient for a life-long learning agent. Domain knowledge must be integrated or consolidated in a systematic fashion for the purposes of efficient and effective retention and for more efficient and effective transfer during future learning. For this reason we focus on the retention of representational knowledge.

Over a long period of time it would be inefficient to retain multiple copies of knowledge for the same or similar tasks. To some extent the knowledge acquired at different times will overlap and support each other. A life-long learning system must have a mechanism for integrating the task knowledge it acquires within a domain knowledge database of finite size. Previous research by [4, 5] has also shown that the accuracy of several related tasks of a domain can in fact be increased by learning these tasks simultaneously within a shared representation. This would increase the overall effectiveness of domain knowledge.

The consolidation of task knowledge is equally important for the transfer of domain knowledge to a new target task. Knowledge integration requires a method of indexing into domain knowledge that reduces search time and increases the probability of selecting the most appropriate knowledge for transfer. Our previous research into knowledge transfer [2] has shown that the most effective method of indexing into domain knowledge is through structural measures of task relatedness that required common internal representation.

2.4 Knowledge Transfer. The form in which task knowledge is retained can be separated from the form in which it is transferred. In [6] we define the difference between representational and functional transfer. Representational transfer involves the direct or indirect assignment of known task representation (weight values) to the model of a new task. In this way the learning system is initialized in favour of a particular region of hypothesis space within the modeling system. We consider this to be an explicit form of knowledge transfer from a source task to a target task. Since 1990 numerous authors have discussed methods of representational transfer [9,10,11,12]. 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 does not involve the explicit assignment of prior task representation when learning a new task; rather, it employs the use of implicit pressures from training examples of related tasks [13], the parallel learning of related tasks constrained to use a common internal representation [4,5], or the use of historical training information from related tasks [1,14,15]. These pressures serve to 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.

Multiple task learning (MTL) neural networks are one of the better documented methods of functional transfer [4]. 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 shared by all tasks. The sharing of this...
internal representation is the method by which inductive bias occurs within an MTL network. MTL is a powerful method of knowledge transfer because it allows two or more tasks to share all or part of internal representation to the extent to which it is mutually beneficial.

In [7] the task rehearsal method (TRM) was introduced as a method of retention and recall of learned task knowledge. Building on the theory of pseudo-rehearsal [16], previously learned but unconsolidated task representations are used to generate virtual examples as a source of functional knowledge. Task rehearsal uses an MTL network (initialized with random weight values) to relearn, or rehearse, these secondary tasks in parallel with the learning of a new task. It is through the rehearsal of previously learned tasks that knowledge is transferred to the new task.

In [8] we turned our attention to the representational form of knowledge transfer in the hopes of overcoming a fundamental problem with functional transfer based on task rehearsal and MTL; relearning secondary tasks within an MTL network starting from initial random connection weights is not an efficient use of domain knowledge. The paper presents a theory of task knowledge transfer that is based on the existence of an MTL network that contains all previously learned task knowledge in a consolidated representational form. This consolidated representation is used as the starting point for learning a new task. Task rehearsal is used to ensure the stability of related secondary task knowledge within the MTL network and stochastic noise is used to create plasticity in the network so as to allow a new task to be learned from an impoverished set of training examples. The transfer of knowledge under the method is therefore both representational and functional. Experiments have shown that the method quickly produces accurate hypotheses for tasks of a synthetic domain.

Note that the above method depends on the existence of a consolidated MTL network containing all previously learned task knowledge. The question of how new task knowledge can be sequentially consolidated within a neural network is interesting and challenging. In fact, it is the stability-plasticity problem originally posed by [17] taken to the level of learning sets of tasks as opposed to learning sets of examples. The stability-plasticity problem points out the difficulty in trying to learn a new task within a neural network while at the same time trying to maintain knowledge of previously learned tasks. The loss of the previously learned task knowledge has been referred to as catastrophic forgetting [18]. The next section presents our approach to overcoming this problem so as to sequentially consolidate knowledge.

3 CONSOLIDATION THROUGH MTL AND TASK REHEARSAL

The consolidation of various task knowledge using a connectionist network is proposed in [19]. The report suggests a method by which the neocortex of the mammalian brain consolidates new knowledge without loss of previous knowledge. Consolidation occurs through a slow process of interleaved learning of new and old knowledge within a long-term memory structure of sparse representation. MTL networks would seem to be a natural choice for consolidated domain knowledge. They have the ability to integrate knowledge from several tasks of a domain; tasks can share portions of a network’s internal representation and the degree to which sharing occurs can be used as a powerful measure of task relatedness [7]. However, there are at least three significant challenges that must be overcome if MTL networks are to be used to as a substrate in which to sequentially consolidate domain knowledge: (1) preventing the catastrophic forgetting of previously learned tasks already existing within the MTL network, (2) learning related and unrelated tasks within the same MTL network, and (3) escaping from pre-existing high-magnitude weight representations.

The problem of catastrophic forgetting can be overcome by using task rehearsal. As shown in [8], training examples for a new task can be used to develop available representation within a large MTL network while the virtual training examples for each of the old tasks are used to maintain accurate representations of those tasks. However, there are two sub-problems in this area. To ensure the accuracy of both the old and the new tasks a large number of training examples are required. These examples should cover the input space to the breadth and depth required by each task; that is the number and selection of examples should reflect the probability distribution over the input space. Only in this way can the prior knowledge of the MTL network be maintained while at the same time integrating knowledge from the new task. This suggests that training times will be very long during the consolidation of new task knowledge. A second problem concerns the storage of large numbers of virtual examples for the purposes of rehearsal. Our prototype software generates all virtual examples at one time and uses them during batch learning. Although this problem is outside of the scope of the current paper, we suspect this space complexity problem can be overcome by producing virtual examples on-line during learning such that their storage is not required. Some related work by [16] on pseudo-rehearsal has indicated that this may be possible.

The problem of learning related and unrelated tasks in the same MTL network is associated with the previous problem. Consolidation is the process of creating a shared use of the internal representation (hidden nodes) within an MTL network. Related sets of tasks will share common features within the internal representation whereas unrelated tasks will not. To the degree to which a new task shares the common features with one or more older tasks, there will be consolidation. To the degree to which a new task cannot share the existing features there will be interference and eventually the network will have to create new features. This is...
accomplished by using portions of the internal representation that are redundant or unimportant to the current mix of domain knowledge tasks. Therefore, in the worst case, there must be sufficient internal representation within the MTL network for learning each task independent of all others. In addition, there must be sufficient training examples so as to ensure that new features are created as necessary.

The final problem, identified by [20], concerns the escape from high-magnitude connection weights within a previously trained network. This issue had to be addressed in previous work [8] in order to use a consolidated MTL network as the starting point for learning a new task from an impoverished set of training examples. In this case a validation or tuning set of data is used to prevent over-fit of the network to the training data and therefore to promote generalization. The solution was to use stochastic noise to provide a degree of plasticity in the network so as to escape from the local minimum created by the high-magnitude weights. Initial experimentation using consolidated MTL networks has shown that with large and rich sets of training examples for both the primary and secondary tasks there is no need for stochastic noise. The large training set for the primary task will ensure the development of an effective hypothesis.

4 EXPERIMENTS

To test our theory of sequential consolidation using an MTL network we conduct a series of three experiments on a single domain. Our objective is to show that given sufficient training examples for each new task that it is possible to sequentially consolidate the representations of those tasks within a single MTL network with little or no loss in accuracy to any of the tasks. The first experiment shows that it is possible to consolidate an increasing number of tasks within a large MTL network each time starting from random initial representations. The second experiment examines the benefit of learning each task of the sequence starting from previously consolidated MTL network representations but without the use of task rehearsal. The final experiment explores learning each task of the sequence with the benefit of both previously consolidated MTL network representations and task rehearsal.

4.1 Test Domain. The seven tasks of the Band domain are characterized in Figure 2. Each is a band of positive examples (the shaded area) across a 2-dimensional input space. All tasks are non-linearly separable requiring two hidden nodes to form a proper internal representation. A visual inspection of the domain suggests that each task varies in its relatedness to the other tasks according to the similarity of the orientation of the band of positive examples. From an inductive learning perspective, those tasks that share common features within internal representation, in this case discriminate boundaries, will be most highly related.

A set of 200 examples and their corresponding classification values for each task was generated as the training set. Another set of 200 examples was generated as an independent test set. No validation (tuning) is needed as the set of 200 training examples is sufficient for developing hypotheses for all tasks with 99% classification accuracy on the test set.

![Figure 2. The band domain. Each task is a 2-variable input space consisting of a band of positive examples](image)

4.2 General Method. The MTL neural networks used in the following experiments have an input layer of 2 nodes, one hidden layer (common feature layer) of 28 nodes, and an output layer of 7 nodes, one for each task. The number of hidden nodes is more than is required for the standard MTL method, because a maximum of two hidden nodes are needed to create the internal representation for each of the band tasks. In all experiments, the mean squared error cost function is minimized by the back-propagation algorithm that uses a momentum term. The base learning rate is 0.1 and the momentum term is 0.9. For all runs that do not use representational transfer from consolidated domain knowledge, random initial weight values are selected in the range -0.1 to 0.1.

The hypotheses are developed using the 200 training examples and then tested against the test set. Training proceeds for 20,000 iterations through the training set or until the mean squared error (on the training set) averaged over all tasks reaches a tolerance value of 0.01. The network representation is saved at the point of minimum mean training set error. Each experiment reports the results of 10 repetitions using different random initial weights within the network.

Performance of the methods is compared in terms of the effectiveness of maintaining the accuracy of the consolidated domain knowledge tasks and the efficiency of consolidating each new task into the MTL network. Effectiveness is measured as the mean percentage of correct classifications, over 10 repetitions, made by the hypotheses against a test set. Efficiency is measured as the mean number of iterations to reach the mean error tolerance value of 0.01. Difference of means hypothesis tests (2-tailed, paired) based on a t-distribution will determine the significance of the difference between the statistics.
4.3 Experiment 1: Consolidation within an MTL network starting from random initial representation.

Method. This experiment examines consolidation within an MTL network as the number of tasks is increased. The tasks are learned in reverse order, from T7 through T1. Each time a task is learned all previous tasks of the sequence are also learned within the network (T6 will be learned in parallel with T7, T5 will be learned in parallel with T6 and T7, etc). Before training begins, the MTL network has its representation (network weights) set to small random values. Thus previously consolidated domain knowledge is not used. As the number of tasks being learned increases there will be an increasing demand for varied use of internal representation so training time is expected to increase.

Results and Discussion. Figure 3 shows the results in the order in which the tasks were learned. The first graph indicates that it is generally possible to develop excellent models for all tasks of the domain within a large MTL network. There was a difficulty in developing accurate models for Task T7 and the combination of tasks T6 and T7 on a couple of the trials. This is due to the large hypothesis space of the MTL network that must be searched within 20,000 iterations. Once the number of tasks being learned in parallel within the MTL network exceeded two, then very accurate hypotheses are consistently developed for all tasks. The competition for internal representation actually works to the benefit of all hypotheses. This is an example of how learning multiple tasks can generally be helpful. The mean test set accuracy over all runs for all tasks was 95.0%.

The second graph shows the mean number of iterations before reaching the desired level of error tolerance across all hypotheses. The second graph shows the mean number of iterations before reaching the desired training error tolerance level. Notice how the time required to train the network generally increases as the number of tasks increase. Because the MTL networks start from random initial weights each time, the time to develop good hypotheses grows as a function of the number of tasks. The mean number of training iterations over all runs for all tasks was 11,580 (95% confidence interval of 5,810).

4.4 Experiment 2: Consolidating a new task within an MTL network starting from previously consolidated task representation but not using task rehearsal.

Method. This experiment examines the benefit of learning of each task of the sequence starting from previously consolidated MTL network representations but without the use of task rehearsal. As in Experiment 1, the tasks are learned in reverse order, from T7 through T1. Each time a task is learned a consolidated MTL network containing accurate representations of the previously learned tasks is used as the initial representation (T6 will be learned starting from the MTL network representation for T7, T5 will be learned starting with the consolidated MTL network representation for T6 and T7, etc). This required the creation of six consolidated MTL networks prior to running the experiment. For each run, weights between the hidden node and the task output node were initialized to small random values. Training proceeds with the learning rate for all tasks except the new task set to zero. This means that the domain knowledge tasks are not rehearsed. This should make it easier for the new task to learn however the internal representations of the previously consolidated tasks will be degraded and therefore the overall accuracy of the tasks should decrease.

Results and Discussion. It was possible to learn highly accurate models for each of task of the sequence except for a couple runs of the first task, T7. As explained in Experiment 1 this is because of the large hypothesis space of the MTL network for a single task. The results demonstrate that with sufficient training examples it is possible to overcome the high-magnitude weights of a previously trained network in order to produce accurate hypotheses for a new task. Furthermore, the number of iterations required to develop these hypotheses generally decreases as knowledge of the domain increases. This can be seen in the second graph of Figure 4. By the time that T1 is learned, the domain knowledge (common features of the internal representation) within the consolidated MTL network provides an excellent starting point for learning a new task of the domain. This reduces the mean number of training iterations over all runs for all tasks to 1,063 (95% confidence interval of 286).

However, there is price that is paid for this rapid learning. The first graph of Figure 4 shows quite clearly that the overall accuracy of previously consolidated domain knowledge tasks decreases as their representation is replaced with representation for the new task. Without parallel rehearsal of the domain knowledge tasks while learning the new task that domain knowledge will be lost. Consequently, the mean test set accuracy over all runs for all tasks is down to 85.6%.
network for learning a new task. It is possible to create a consolidated representation within the MTL network, it uses task rehearsal to reinforce the existing results.

4.5 Experiment 3: Consolidating a new task within an MTL network starting from previously consolidated task representation and using task rehearsal.

Method. The final experiment explores learning each task of the sequence with the benefit of both previously consolidated MTL network representations and task rehearsal. As before, the tasks are learned in sequence from T7 through T1. After each new task is learned the consolidated MTL network representation is saved. Before training begins on the next task, this consolidated MTL network is used as the initial representation. Only the weights between the hidden nodes and the new task output node are initialized to small random values. All previously learned tasks of the sequence are rehearsed within the network when learning a new task. The current consolidated MTL network is used as the source of the virtual examples for this rehearsal.

When training begins the error on the previously learned tasks is very small. Only the new task shows significant error. This guides the back-propagation algorithm of the MTL network to find and/or create the necessary internal representation for the new task. This process will interfere with the rehearsal of the previously consolidated tasks and drive their error rates upward temporarily. Over several thousand iterations however, sufficient internal representation should be found for all tasks and the mean error rate should drop below the tolerance level. In this way task rehearsal is used to maintain the accuracy of prior task knowledge while the new task is consolidated into the MTL network.

Results and Discussion. As with the previous experiments it was possible to learn highly accurate hypotheses for all tasks beyond the first task, T7. The results shown in Figure 5 demonstrate that despite the use of task rehearsal to reinforce the existing consolidated representation within the MTL network, it is possible to create additional features within the network for learning a new task. The mean test set accuracy over all runs for all tasks was 95.7% which is equivalent to that of MTL learning without the benefit of consolidated domain knowledge.

The mean number of training iterations over all runs for all tasks was 2,137 (95% confidence interval of 946). This is statistically a greater number of iterations (p=0.00096) than learning without task rehearsal, however it is substantially less than that of standard MTL learning from random initial weights. Also notice that as the number of tasks within domain knowledge increases, the number of required iterations at first increases and then decreases. We suspect that this is because of a complex interaction between the number of tasks consolidated with the MTL network and the relatedness between these tasks. As the number of tasks within the network increases, the difficulty in generating new features with the internal representation increases. However, after a critical number of common features have been developed with the consolidated MTL network they start to be reused by new tasks. This leads to a reduction in training time. Our intention is to investigate this more thoroughly in the future.

5 CONCLUSION

In this paper we have addressed a fundamental question of a life-long machine learning system: How can task knowledge be consolidated within a long-term domain knowledge structure for the benefit of future learning? A theory of task knowledge consolidation was proposed that uses a large MTL network as the domain knowledge structure and task rehearsal as a method of overcoming the problem of catastrophic forgetting. Experiments have been conducted on a synthetic domain of seven tasks using software that was developed in accord with the theory. The results indicate that, given an abundance of training examples for each new task, the method is capable of sequentially consolidating task knowledge within a domain knowledge MTL network in an efficient and effective manner; far fewer training iterations are...
required to develop accurate models than if the network was starting from initial random weights.

These results are important because having a consolidated source of domain knowledge in the representational form of an MTL net has been shown to provide a basis for more efficient and effective transfer of knowledge when learning a new task from sets of impoverished data [8]. It provides the basis for indexing into domain knowledge using deep structural measures of task relatedness and it can speed up learning through the direct use of prior representation.

There are some limitations of the method that need to be addressed. First of all there is the issue of the inaccurate models created for the first task of the sequence. This can be eliminated by growing the size of the internal representation (number of hidden nodes) with each new task. Therefore when the first task is learned a relatively small hypothesis space would be present. A second potential problem is one of scalability. Notice that the mean accuracy of all models decreases slightly as the number of domain knowledge tasks that are being rehearsed increases. This may indicate a compounding loss of prior task knowledge due to increased competition for existing representation. Once again, increasing the amount of internal representation may help or alternatively just starting with a larger MTL network may be sufficient. Both of these areas will be explored in future research.

ACKNOWLEDGEMENT

This research has been funded by the Government of Canada through NSERC grants.

REFERENCES


