Avoiding Cross-Talk Interference during the Sequential Learning of Multiple Skills

Ashish Gupta

Paper for the Written Component of the Vanderbilt Graduate Computer Science Program Preliminary Examination in Intelligent Systems

Humans can learn multiple skills sequentially. We can also apply the knowledge of previously learned skills to novel domains. In contrast, most machine learning techniques fail to do so. Traditional artificial neural networks have been extensively used to model human behavior as well as for many engineering applications. However, they too fail to learn sequentially. Numerous ideas have been proposed to overcome this problem with artificial neural networks. This paper discusses the applicability of these ideas to machine learning and also discusses their biological plausibility. The shortcomings in these ideas are discussed and areas that need further research have been identified.

1. Introduction

Most machine learning techniques assume that all the domain information is available a priori. The learning agent extracts useful knowledge from the existing domain information and encodes it to solve a particular problem. Most practical applications that require a learning component, however, are very different. In most cases, it is too costly to gather all the domain information beforehand. Also, there are cases when the domain information changes throughout agent's lifetime. It is also possible that over its lifetime, the agent has to work in multiple domains. Traditional machine learning systems perform poorly in such cases, because they can't learn sequentially. Once trained, these systems have no mechanism of learning something new, while still holding on to the previously acquired knowledge. Hence, coming up with systems that can learn sequentially is an important challenge for the machine learning community.

One good source to explore for ideas and inspiration is the human brain. Humans and non-human animals interact with multiple domains over their lifetime. We are not only able to learn sequentially in each new domain, but we can also apply the previously acquired knowledge in novel ways in new domains. It could be interesting to explore the mechanisms that enable us to learn sequentially, with the hope of applying them to machine learning.

Human brain is a network of neurons. Biologically inspired mathematical models of neural networks have been used for successful explanation of many cognitive and psychological phenomena. Artificial neural networks have been employed in numerous engineering applications as well. However, traditional artificial neural network models also fail to learn sequentially. Artificial neural networks use a set of weights as their memory. These weights, along with powerful learning algorithms like error back-propagation give them the capability of arbitrarily approximating almost any function. However, this is the root cause of the problem as well, since it makes the network susceptible to forget all old information as soon as the weights change. Hence, these networks exhibit '*catastrophic interference*', where all the knowledge of an initially acquired skill is lost soon after the training on a second skill starts **[M. McCloskey, 1989]**.

There have been numerous attempts at solving the problem of catastrophic interference. One simple approach is to retain all old training data and retrain the network each time new data gets added. A better approach is to train the network on '*pseudo-patterns*' **[B. Ans, 2000; B. Ans, 2002; B. Ans, 2004a; B. Ans, 2004b; K. A. Norman, 2005]**, where the network learns the underlying distribution of the input data and then generates the training patterns by sampling from that distribution. Another solution is to use a new network for each new data set. A more refined approach is the modular network architecture **[D. M. Wolpert, 1998; T. Brashers-Krug, 1995]**, where multiple modules are used, each specializing for a subset of the data. Outputs of different modules are gated and combined depending on the context, giving more weightage to the modules that are applicable in the current context. Another related approach is to generate sparse internal representations, which could lead to orthogonal encoding of different unrelated skills, thereby reducing the amount of cross-talk **[A. Gupta, 2005a; A. Gupta, 2005b; A. Gupta, 2006; R. C. O'Reilly, 2001a; R. C. O'Reilly, 2001b; R. M. French, 1999; R. M. French, 2004]**. If the previously acquired skills are related to the new skill, then it is possible that the previous knowledge facilitates the acquisition of the new skill. There have been attempts to explore mechanisms for the transfer of related knowledge during sequential learning **[B. P. Goettl, 1996; D. M. Clawson, 2001; S. Thrun, 1995]**.

In the next section, we will review the above mentioned approaches for solving the sequential learning problem and explore their applicability machine learning. We will also discuss the biological plausibility of these methods. In the third section, we will compare these approaches with our own research work. We will summarize our conclusions in section 4.

2. Techniques to Avoid Cross-Talk Interference

2.1 Retraining

One straightforward approach to avoid cross-talk interference is to retain the previous training data and retrain the network each time new data is added. A drawback of this approach is that a lot of space could be required for retaining all old data. Another drawback is that training the network each time new data arrives could be time consuming. Finally, this mechanism is unlikely to be used by the human brain to overcome interference.

[R. M. French, 1999] proposed a way to overcome the problem of space complexity. Their approach feeds randomly generated input patterns through the previously trained network. The network produces corresponding output patterns. These randomly generated input-output patterns are mixed with the newly arrived training data, and the network is trained on this expanded set. This approach does not overcome the problem of time complexity. Another problem is that since the input patterns are generated randomly, the probability of reproducing patterns similar to those in the initial training set is very small, in fact, zero if the patterns are real valued.

A possible enhancement is to learn the underlying distribution of the input data and then sample from that distribution to generate inputs. This approach should significantly improve the chances of generating patterns similar to those in the previous training set. This is the approach proposed by **[B.**

Ans, 2000; B. Ans, 2002; B. Ans, 2004a; B. Ans, 2004b], where they used a pair of '*reverberating feed-forward networks*' for learning multiple sets of patterns one after the other. A typical feed-forward network is modified by adding auto-associative units to the output layer along with the target units. These auto-associative units are trained to reproduce the same pattern that is given to the network as input. These units can then be used to produce attractor patterns of input vectors via a reverberation mechanism. In this mechanism, the input layer of the trained network is given a randomly generated input pattern. The input pattern's activity is propagated forward through the network. The activity produced in the auto-associative units is then used as the input to the network for the next iteration. This process is continued until the auto-associative units settle into an attractor state. Starting with different random initial patterns would result in different attractor states, and each of these attractor states would closely match one of the input patterns that the network was previously trained on. Hence, this pseudo-pattern, comprising of the attractor state input activity and the corresponding target activity can be used for training the network. Two trained networks need to be maintained at any time, one for generating the pseudo-patterns while the other is being training on the combined set of pseudo-patterns and new patterns. **[B. Ans, 2004b]** have extended this technique for simple recurrent networks. A pair of '*reverberating simple recurrent networks*' can learn multiple temporal sequences one after the other. A simple recurrent network is trained on a temporal sequence of patterns. When a new sequence arrives, the network is trained on this new sequence along with the pseudo-patterns generated using the backup network, as explained earlier.

The approach of interleaved training is inspired by the complementary learning systems theory of hippocampus and neocortex **[J. L. McClelland, 1995]** in the human brain. According to this theory, catastrophic interference is alleviated through the use of a fast hippocampal learning system that uses sparse representations. While neocortical systems are assumed to use less sparse representations, making them more vulnerable to interference, problems are avoided through a hippocampally mediated process of consolidation, where neocortical networks receive interleaved '*virtual*' practice in multiple skills. Retention of knowledge through pseudo-pattern training is definitely a strong hypothesis. However, there is no evidence that pseudo-patterns are generate in our brain in the same way as proposed by the authors. Moreover, the hypothesis that the brain is continuously generating pseudo-patterns for all previously acquired knowledge, even while something new is being learned, seems difficult to prove. Further, not all kinds of learning require the involvement of the hippocampus. **[I. H. Jenkins, 1998]** have shown that animals with hippocampal lesions can continue to learn new motor skills.

This technique could be useful for engineering applications. However, the burden on pseudo-pattern generator would keep increasing as the volume of existing knowledge increases. Secondly, it is possible that the system needs to learn something new that is contradictory to its previous learning. Hence, some way of recognizing conflicting patterns needs to be incorporated into this mechanism. One naïve and time consuming approach could be to compare each pseudo-pattern with the new patterns and ignore conflicting pseudo-patterns.

Recently, **[K. A. Norman, 2005]** have proposed an oscillating algorithm that oscillates the strength of inhibitory input to a layer of excitatory neurons. There are inhibitory neurons pervasive throughout the human brain. These neurons interact with the excitatory neurons and act in a regulatory role. The inhibition provided through these neurons keeps the overall activity in any region of the brain constant. Hence, at any point in time, only a subset of the excitatory neurons becomes active. A direct implication of this constraint is that only a subset of the neurons participates in learning any task. In this algorithm, oscillating strength of inhibition changes the number of active excitatory neurons. Learning happens as a result of this change. This mechanism selectively strengthens the weak parts of the target memories and selectively punishes strong competitors. If some noise creeps into the old memories due to subsequent training, then this algorithm can restore the old representations. This is a biologically plausible theory that could be significant for machine learning applications as well.

2.2 Modular Architecture

One extreme solution for the problem of cross-talk interference is to train a new network for each new task. With this approach, we will end up with a group of trained networks. To use these networks, we will need some switching mechanism that can identify the network suitable for solving a given task. This is very similar to the approach taken by modular neural network architectures.

[D. M. Wolpert, 1998] have proposed a modular architecture that is inspired by the behavioral findings from the domain of human motor learning. Human motor system has to deal with multitude of objects and act in many contexts. They proposed that the human motor system comprises of multiple controllers, each suitable for a subset of the contexts. Each controller in their model comprises of a pair of forward and inverse models. The forward model takes the current state and motor command as input and predicts the next state. Multiple forward models learn to predict using a competitive self supervised learning method. Each forward model's output is weighted by a responsibility signal, which is the probability that the current context is represented by that forward model. The final state prediction is a weighted sum of forward model outputs, where each output is weighted by the corresponding responsibility signal. The job of the inverse models is to produce a motor command based on the current state of the system and the desired next state. The final motor command is also a weighted sum of the inverse model outputs.

This model is well suited for engineering tasks. It can generalize to novel domains, if the tasks in the new domain could be represented as a combination of the existing module outputs. It can learn sequentially, without forgetting the previously learned tasks. In fact, the more distinct the context of the new task, the lesser will be the cross-talk, since a very small responsibility signal will be generated for the previously trained modules.

[D. M. Wolpert, 1998] present behavioral findings that suggest that humans are able to learn multiple controllers and switch between them, based on the context. In these studies, human subjects learned to adapt their motor behavior to compensate for some perturbations in the environment. Once they had learned to adapt, these perturbations were removed and it was seen that de-adaptation to the default motor behavior happened very fast. Furthermore, when the perturbations were re-established, the re-adaptation process was faster than it was the first time. In some cases, such switching happened simply when the subjects were moved to a new context. Having distinct modules for different contexts can easily explain this behavior. However, we will explore in section 3 that such behavior could also naturally emerge due to the inhibitory interactions between the neurons in the brain.

While such modular models can exhibit very robust savings, the biological plausibility of a reserve of untrained neural modules awaiting assignment while a new task comes is questionable. Even if we assume the existence of unused modules, question still remains about their granularity – do we need a new module for each new task? It also poses a question concerning the total number of available modules. Even though biologically questionable, such architectures will continue to be of importance for machine learning systems.

2.3 Sparse Representation

The problem of cross-talk interference can be avoided, if we somehow prevent the reuse of neurons for successive tasks. Modular architecture is one extreme way of achieving this neural segregation. Another similar approach is to somehow force neural units within a single module to form a sparse representation, such that only a subset of the units is used to learn any task. If the percentage of units used for learning a particular task is small, the chance of reusing the same units for two different tasks will also be small. There have been attempts to generate sparse representation by manipulating the network parameters. One approach is to initialize the network weights with a large variance so that different hidden units are selectively sensitive to a subset of the input units. This is similar to the approach proposed by **[K. McRae, 1993]**. They pre-trained their network on a large number of related items to build this variance into the weights. However, this technique does not yield robust results.

[R. M. French, 1994] proposed a '*node sharpening*' algorithm. In each iteration of this algorithm, the connection weights between the active input units and the most active hidden layer unit are increased by a small amount while the weights to the rest of the hidden layer units are decreased. If the network weights are randomly initialized, the chances are that the most active hidden layer unit would be different for different input patterns and this difference would be further enhanced through training, thereby reducing interference. Using a single unit to represent each pattern restricts the learning capacity of neural networks, with the maximum number of patterns that can be learned being equal to the number of units in the hidden layer. **[R. M. French, 1994]** modified his node sharpening algorithm to activate a subset of the units, rather than a single unit, to form '*sparse-distributed*' representation. With this approach, the network retained the capacity to learn a large number of patterns while still reducing cross-talk interference to a significant extent.

Could a similar mechanism be working in human brain? There are a large number of neurons in any region of the brain. It is possible that at any given time, only a subset of these neurons become active. In fact, as discussed previously, due to the lateral inhibition pervasive throughout the cortex, this is exactly what happens. **[R. C. O'Reilly, 2001a; R. C. O'Reilly, 2001b]** have proposed the Leabra modeling framework that implements inhibitory interactions using k-Winners-Take-All (kWTA) inhibition function. This function quickly modulates the amount of pooled inhibition presented to a layer of simulated cortical neural units based on the layer's level of input activity. This results in a roughly constant (k) number of units surpassing their firing threshold. A layer of neural units with a small value for the k parameter will produce sparse representations, with few units being active at once. **[R. C. O'Reilly, 2001b]** have shown improved performance during sequential list learning tasks using this mechanism. We have explored the role of such sparse representation in savings during the sequential learning of multiple motor trajectories. We will discuss this in more detail in section 3.

2.4 Transfer of Related Knowledge

If the new task shares some common structure with the previous tasks learned by the network, then there is an avenue for using the existing knowledge to facilitate the learning of the new task. Such facilitation will be seen in cases where due to prior trainings, the network starts at a point in the weight space that is close to the desired point for the new task. Such a network could converge at a faster rate than a randomly initialized network. However, previously acquired knowledge could be used in a more efficient way than just for weight initialization.

[S. Thrun, 1995] have proposed '*explanation based neural network*' learning algorithm for robots. A robot, with a given set of sensors and actuators, might work in an environment to solve numerous different tasks through its lifetime. Such a robot would need to learn a model of the environment, the actor model, which takes the current state and action as input and predicts the next state. Depending on the task at hand, certain states will be beneficial and certain others will be harmful. If the robot is given a reinforcement signal, it can learn a Q function that predicts the reward associated with each state-action pair. This Q function would subsequently direct the robot's behavior.

If the robot is later employed for a different task in the same environment, the actor model remains the same but a new Q function needs to be learned. Explanation based neural network learning algorithm uses the existing actor model knowledge to bias the learning of the Q function. In this algorithm, the robot explores the environment to gather an episode of training examples that map state-action pairs to rewards. Then, using the actor model function, the slope of the desired Q function for each state in the training episode is computed. This additional slope information along with the collected training examples helps in learning a more accurate Q function quickly.

The authors approximate the actor model function using an artificial neural network. Hence, it is differentiable. However, the slope of the reward function with respect to the final state in the training episode is assumed to be available. Hence, this algorithm won't work if the reward function derivate is not available. Moreover, if the robot does not have an accurate actor model, then it will result in incorrect slope information which could potentially lead to an inaccurate Q function. The authors have proposed to check the accuracy of the actor model by comparing its predicted states with the actual states encountered in the training episode.

It is possible that the robot needs to work in many different environments through its lifetime. Even in this case, the sensors and actuators remain the same. Moreover, there could be some similarities between the different environments. The robot could learn to use its sensor information or its actuator capabilities to take advantage of those similarities. **[S. Thrun, 1995]** performed experiments in which a robot had to navigate through different rooms in various buildings. The robot learned a model that took its sensor readings as input and predicted the chances of collision with an obstacle. This learned behavior of collision avoidance was transferred when the robot worked in new environments.

Transfer of learned behaviors to new environments is extremely useful. However, a lot of domain information needs to be provided to the robot by pre-processing the inputs or through the external reinforcement signal. For example, in the above case, the sensor input to the robot was its distance from various obstacles and the robot was penalized each time it hit something. Hence, the inputs and the reward were carrying the information that obstacles need to be avoided in an extremely obvious way. Further, since letting the robot bump around the walls could have damaged it, the authors trained it in a simulated environment. In such a case, is it even valid to call it learning? Unless the robots are able to discover interesting behaviors while actually exploring the environment, their utility in most real world domains will remain very limited.

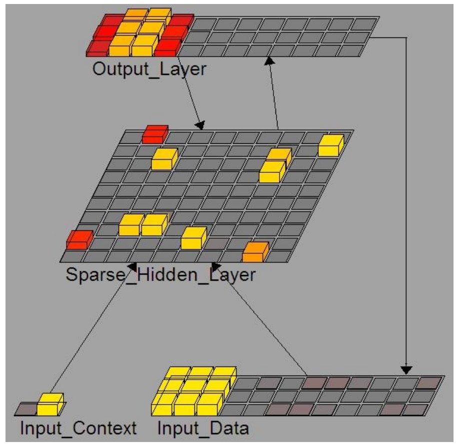

Figure 1: The Leabra network used for learning the motion trajectories of a three joint planar arm. Each rectangular grid represents a layer of neurons. Arrows show the directions of connections. The Input_Data layer has 3 rows of units, one row for representing the current angle of each of the joints. The Output_Layer has three rows of units for generating the three joint angles for the next state. Contextual cues are generated by the Input_Context layer, which has two units. One of the units is switched on during the base task training and the other unit is switched on during the interfering task training. These two units are randomly connected to the Sparse_Hidden_Layer units with an 80% probability of any connection. The Sparse_Hidden_Layer uses the k-Winners-Take-All (kWTA) lateral inhibition function to generate sparse internal representations for the tasks. For learning the motion trajectories of the planar arm, the output for a particular time step is fed as input for the next time step. This is represented by the arrow going from the Output_Layer to the Input_Data layer

If we need to train a robot on a complex task, we could possibly start by training it on low complexity tasks that share some critical features with the complex task. This prior training could speedup the learning of the goal task and also improve the overall performance of the robot. Using this idea, mobile robots were able to solve a complex navigation task when they were pre-training on simpler related tasks **[S. Thrun, 1995]**. Transfer of knowledge through part-task training is important in the domain of human learning as well **[B. P. Goettl, 1996; D. M. Clawson, 2001].** However, research in this direction has shown limited success so far. One of the most difficult challenges in part-task training is to come up with a domain independent set of guidelines for designing the simpler tasks, which could possibly provide training on the critical aspects of the more complex task.

3. Avoiding Interference Using Sparse Representations

In this section, our research on sequential learning is compared with the work described in the previous section. The focus of our research has been on exploring biologically plausible mechanisms for sequential learning in humans and other non-human animals. In **[A. Gupta, 2005a],** the benefits of Leabra's **[R. C. O'Reilly, 2001b]** kWTA inhibition during the sequential learning of motor sequences have been explored (Refer *Figure 1* and the associated caption for a brief overview of the model). Motion trajectories of a three joint planar arm were used to measure savings shown by a Leabra network. When the network was trained on multiple motion trajectories sequentially, interference was found to decrease substantially for small values of k.

In most real-world situations, there are distinct sensory or internal control cues associated with different tasks. As explained earlier, **[D. M. Wolpert, 1998]** have used these cues to generate responsibility signal, which helps in the selection of modules applicable for the current problem. We have explored the role of such contextual cues in shaping the internal representations. Contextual cues were introduced in our model by adding an input layer with two units (*Figure 1*). One unit in this layer was active for each of the two tasks that were learned. These two units were randomly connected to the units in the hidden layer, with an 80% probability of any particular connection being formed. Contextual input along with a small k (kWTA) value for the hidden layer significantly improved savings by segregating the representations even further. This model provides an alternative explanation to the behavioral results described in **[D. M. Wolpert, 1998]**, where they have used distinct modules to learn motor trajectories in different contexts.

Humans show increased savings in motor skills when the initial skill is over-learned. We increased the training time for the base task in our simulations. Due to the use of Hebbian learning component along with the error driven learning, the weights in the Leabra network continue to change even after the task has been learnt to the criterion. Use of Hebbian learning along with the kWTA mechanism results in continued sharpening of the hidden layer representation. Typically, with lesser training, more than k hidden layer units become active at moderate levels for any input pattern. However, with continued training, a subset of these units strengthens its weights and thereby suppresses the less active units. This sharper and stronger representation is difficult to perturb when an interfering task is learned.

We have explored the transfer of knowledge shown by our model during the sequential learning of multiple tasks **[A. Gupta, 2005b]**. Interfering tasks with a subset of patterns common to those in the base task were used for this exploration. It was desirable that for the common patterns, the network would continue to use the same hidden layer representation that had emerged during the training on the base task. It was found that a fine balance between the extent of sparsity in the hidden layer and the strength of contextual cues makes this happen. With the right balance, the contextual cues are just strong enough to force un-common patterns to use previously unused hidden units. When a common pattern is presented, previous hidden units start with a bias which helps them win the inhibitory competition. If the contextual cues are made stronger, the network fails to generalize. If contextual cues are made weaker, the network suffers interference. As expected, the network also show a speedup in training times if the interfering task shares patterns with the base task.

[R. C. O'Reilly, 2001a] have explored transfer of knowledge in another way. They tested Leabra's performance on novel tasks that were formed by combining the features from different tasks that the network was previously trained on. It was found that the biological principles of Hebbian learning and lateral inhibition enable the network to generalize well. We plan to explore if such generalization would occur when novel tasks are formed by combining features from sequentially learned previous tasks.

It is possible that a previously acquired skill has become obsolete and has to be replaced by some contradictory skill. How should the network behave in this case? How do humans and other non-human animals behave? This scenario commonly arises in the domain of animal conditioning where an animal is given extinction training after initial acquisition training. During the acquisition training, the animal learns to produce a particular response to a stimulus due to the association of some reward. During extinction training, that response is suppressed by the removal of the reward. It is believed that acquisition training happens due to the change in synaptic connections between certain neurons in the brain. For a long time, it was believed that during extinction training, the strength of these connections is reversed such that the animal stops producing the response. However this theory fails to explain numerous new behavioral findings. These behavioral findings suggest that extinction training involves the formation of a separate memory that suppresses the acquisition related learning.

We **[A. Gupta, 2005c; A. Gupta, 2006]** have explored a simple Leabra model of conditioning in which acquisition and extinction learning involve the formation of separate patterns of activity that compete via the lateral inhibitory mechanism. Acquisition training involves the strengthening of acquisition related pathway. Extinction training involved the strengthening of extinction related pathway. It also leads to the simultaneous weakening of acquisition related pathway. However, as soon as the extinction neurons win the lateral inhibitory competition over the acquisition neurons, the acquisition neurons' activity is suppressed to a very small value. At this point, the acquisition neurons stop participating in the training process, and do not show any decrease in connection strengths. Hence, re-acquisition training happens at a faster rate than the initial acquisition training. Similarly, second extinction training also requires lesser time. We have been able to fit this model to many more behavioral results. Even though biologically inspired, the above model shows a characteristic that could be desirable in machine learning systems as well. If some piece of knowledge that was once useful subsequently becomes obsolete, it might still be in the best interest of the agents to not forget that knowledge completely, in case they need to use it again in future.

Our work so far has explored biologically plausible hypotheses for sequential learning and related problems. We are planning to explore the engineering applicability of our results by using more extensive training sets with larger number of patterns and by training the network on more than just two tasks sequentially.

4. Conclusions

Sequential learning is an important challenge for the machine learning community. It is also an important piece of puzzle for understanding how the human brain works. In this paper, we have discussed the different ideas proposed as a solution to this problem. These ideas can be broadly classified as retraining using pseudo-patterns, modular network architectures, sparse internal representation and transfer of previous knowledge. All of these ideas are inspired by the behavioral or neuroscientific findings of the human brain. Out of these, modular architectures and sparse representation are inspired by the same underlying principles. The rest are very different from each other. This difference could be useful since it gives us the opportunity to combine these techniques together. Such a collaborative solution should give more robust results than any of the individual approaches. Moreover, it seems plausible that human brain also uses a combination of all of these techniques in some way for solving the sequential learning problem.

References

[A. Gupta, 2005a] A. Gupta and D. C. Noelle (2005), The role of neurocomputational principles in skill savings, in the proceedings of the 27th annual conference of the Cognitive Science Society, pp. 863--868

[A. Gupta, 2005b] A. Gupta and D. C. Noelle (2005), Neurocomputational mechanisms for generalization during the sequential learning of multiple skills, in the proceedings of the Indian International Conference on Artificial Intelligence, pp. 2699—2711

[A. Gupta, 2005c] A. Gupta and D. C. Noelle (2005), Savings in conditioning via asymmetric synaptic plasticity, poster presented at Dynamical Neuroscience XIII: Computational Cognitive Neuroscience conference

[A. Gupta, 2006] A. Gupta and D. C. Noelle (2006), Lateral inhibition explains savings in animal conditioning, submitted to the 28th annual conference of the Cognitive Science Society

[B. Ans, 2000] B. Ans and S. Rousset (2000), Neural networks with a self refreshing memory: knowledge transfer in sequential learning tasks without catastrophic forgetting, Connection Science, Vol. 12, No. 1, pp. 1--19

[B. Ans, 2002] B. Ans, S. Rousset, R. M. French and S. Musca (2002), Preventing catastrophic interference in multiple-sequence learning using coupled reverberating Elman networks, in proceedings of the 24th annual conference of the Cognitive Science Society, pp. 71--76

[B. Ans, 2004a] B. Ans (2004), Sequential learning in distributed neural networks without catastrophic forgetting: A single and realistic self-refreshing memory can do it, Neural Information Processing – Letter and Reviews, Vol. 4, No. 2, pp. 27—32

[B. Ans, 2004b] B. Ans, S. Rousset, R. M. French and S. Musca (2004), Self-refreshing memory in artificial neural networks: learning temporal sequences without catastrophic forgetting, Connection Science, Vol. 16, No. 2, pp. 71—99

[B. P. Goettl, 1996] B. P. Goettl and V. J. Shute (1996), Analysis of part-task training using the backward-transfer technique, Journal of Experimental Psychology: Applied, Vol. 2, No. 3, pp. 227--249

[D. M. Clawson, 2001] D. M. Clawson, A. F. Healy, K. A. Ericsson and L. E. Bourne Jr. (2001), Retention and transfer of Morse code reception skill by novices: part-whole training, Journal of Experimental Psychology: Applied, Vol. 7, No. 2, pp. 129—142

[D. M. Wolpert, 1998] D. M. Wolpert and M. Kawato (1998), Multiple paired forward and inverse models for motor control, Neural Networks, Vol. 11, pp. 1317—1329

[I. H. Jenkins, 1998] I. H. Jenkins, D. J. Brooks, P. D. Nixon, R. S. J. Frackoviak and R. E. Passingham (1994), Motor sequence learning: A study with positron emission tomography, The Journal of Neuroscience, Vol. 14, No. 6, pp. 3775--3790

[J. L. McClelland, 1995] J. L. McClelland, B. L. McNaughton and R. C. O’Reilly (1995), Why there are complementary learning systems in the hippocampus and neocortex: Insights from the success and failures of connectionist models for learning and memory, Psychological Review, Vol. 102, No. 3, pp. 419--457

[K. A. Norman, 2005] K. A. Norman, E. L. Newman and A. L. Perotte (2005), Methods for reducing interference in the complementary learning systems model: Oscillating inhibition and autonomous memory rehearsal, Neural Networks, Vol. 18, pp. 1212--1228

[K. McRae, 1993] K. McRae and P. A. Hetherington (1993), Catastrophic interference is eliminated in pretrained networks, in proceedings of the 15th annual conference of the Cognitive Science Society, pp. 723—728

[M. McCloskey, 1989] M. McCloskey and N. J. Cohen (1989), Catastrophic interference in connectionist networks: The sequential learning problem, in G. H. Bower (Ed.) The Psychology of Learning and Motivation, Vol. 24, New York Academic Press, pp. 109--165

[R. C. O'Reilly, 2001a] R. C. O'Reilly (2001), Generalization in interactive networks: The benefits of inhibitory competition and hebbian learning, Neural Computation, Vol. 13, pp. 1199--1241

[R. C. O'Reilly, 2001b] R. C. O'Reilly and Y. Munakata (2001), Computational exploration in cognitive neuroscience, MIT Press

[R. M. French, 1994] R. M. French (1994), Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference, in proceedings of the 16th annual conference of the Cognitive Science Society, pp. 335—340

[R. M. French, 1999] R. M. French (1999), Catastrophic forgetting in connectionist networks, Trends in Cognitive Sciences, Vol. 3, pp. 128--135

[S. Thrun, 1995] S. Thrun and T. M. Mitchell (1995), Lifelong robot learning, Robotics and Autonomous Systems, Vol. 15, pp. 25—46

[T. Brashers-Krug, 1995] T. Brashers-Krug, R. Shadmehr and E. Todorov (1995), Catastrophic interference in human motor learning, Advances in Neural Information Processing Systems, Vol. 7