2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Alleviating Catastrophic Forgetting via Multi-Objective Learning

Yaochu Jin, *Senior Member, IEEE*, and Bernhard Sendhoff, *Senior Member, IEEE*

***Abstract*— Handling catastrophic forgetting is an interesting and challenging topic in modeling the memory mechanisms of the human brain using machine learning models. From a more general point of view, catastrophic forgetting reflects the stability-plasticity dilemma, which is one of the several dilemmas to be addressed in learning systems: to retain the stored memory while learning new information. Different to the existing approaches, we introduce a Pareto-optimality based multi-objective learning framework for alleviating catastrophic learning. Compared to the single-objective learning methods, multi-objective evolutionary learning with the help of pseudo-rehearsal is shown to be more promising in dealing with the stability-plasticity dilemma.**

I. Introduction

Learning in the human brain is inherently a multi-objective process [9], [27]. One well-known issue is the stability-plasticity dilemma [3], which means that the learning system should be able to learn new information efficiently without completely forgetting what has been learned previously. The stability versus plasticity dilemma is often known as catastrophic forgetting in neural network based machine learning [28].

Existing techniques for alleviating catastrophic forgetting can largely be divided into three categories [16]. The methods developed in the first category are mainly based on the idea that catastrophic interferences in learning are caused by distributed representation of the previously learning patterns (referred to *based patterns* hereafter) and the *new patterns* to be learned. Thus, to avoid catastrophic forgetting, semi-distributed [14] or sparse representations instead of fully distributed representations are used.

In the second category, all or part of the base patterns are interleaved with the new patterns during learning of the new patterns, which is known as *direct rehearsal*. A more technically sound and biologically plausible variant of the direct rehearsal method is the *pseudo-rehearsal* technique [32]. By pseudo-rehearsal, it is meant that the base patterns are not directly re-learned together with the new patterns. By contrast, random inputs are generated and fanned into the trained neural network to get the corresponding outputs. These patterns (termed *pseudo-patterns*), are then mixed with the new patterns and are re-learned by the network. This is biologically more plausible due to its similarity to the memory consolidation mechanism in human brain [17]. A problem that arises in pseudo-rehearsal is the *runaway effect*, which means that it is possible that one or a few of the base patterns monopolize the rehearsal process and the rest base patterns are forgotten [29]. Several techniques have been suggested to solve the runaway effect to a certain degree [29].

The authors are with the Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany (*email: yaochu.jin@honda-ri.de*)

A further step to go from the semi-distributed representations is to adopt a dual-network structure [15], [7] or a complementary learning systems [9], which belong to the third category. In these methods, one sub-structure is responsible for learning new patterns, and the other for consolidating the previously learned patterns. These dual-network learning models are similar to the two separate areas in the brain, namely, the hippocampus and the neocortex. In some of the methods in the third category, pseudo-patterns are also generated from one sub-structure to simulate the interactions between the hippocampus and the neocortex in the brain during memory consolidation.

One recent paper [34] employs a single-objective evolutionary algorithm to evolve minimal catastrophic forgetting neural systems by counting the number of remembered base patterns when learning new patterns.

In all existing methods, catastrophic forgetting, which reflects the trade-off between learning new patterns and remembering base patterns, is addressed using learning methods by minimizing one single cost function. Since learning the pseudo-patterns and learning the new patterns are very likely competitive, it is natural to deal with the conflicting objectives using the Pareto-based multi-objective learning, which has received increasing attention in machine learning over the past few years [25]. The Pareto-based approach to machine learning has been shown to be advantageous over the traditional learning algorithms in the following aspects. First, the performance of learning algorithms can be improved, probably due to the new error surface introduced by multi-objective optimization [1]. Second, it is possible to simultaneously generate multiple learning models that account for different learning goals, e.g., accuracy and complexity [20], [22], multiple error measures [12], interpretability and accuracy [24]. Third, the multiple learning models produced using multi-objective optimization are well suited for constructing learning ensembles [2], [10], [22]. And finally, more information can be gained by analyzing the Pareto front obtained in multi-objective machine learning. For example, the number of optimal clusters can be obtained by analyzing the Pareto front in multi-objective clustering [18]. It is also shown in [26] that by taking a closer look at the Pareto front trading off between accuracy on the training data and the complexity of the neural networks, we are able to identify the neural networks on the Pareto front that are most likely to generalize well on unseen data.

A related work is the learning with minimal degradation (LMD) suggested in [5]. In the LMD, sequential learning of n patterns is treated as the minimization of the error over the $n-1$ previously learned patterns subject to the perfect encoding of the n-th pattern. The LMD has been extended

0-7803-9490-9/06/$20.00/©2006 IEEE

in [6], where *a priori* interference prevention is introduced in learning the $n-1$ patterns in addition to the *a posteriori* interference minimization. However, as indicated in [5], the success of the LMD is very limited.

This paper presents our preliminary results on alleviating catastrophic forgetting using evolutionary multi-objective learning. In the next section, a brief introduction to Pareto-based multi-objective evolutionary optimization is provided. Section III describes how avoiding catastrophic forgetting can be formulated as an evolutionary multi-objective problem. Section IV describes the evolutionary multi-objective algorithm for optimizing the parameters and structure of feed-forward neural networks. It is shown that the multi-objective learning framework is more elegant in addressing the runaway effect in pseudo-rehearsal and more efficient in tackling catastrophic forgetting. A summary of the paper is provided in Section VI.

II. Pareto-based Evolutionary Multi-objective Optimization

Without loss of generality, we discuss minimization problems. A multi-objective problem can be formulated as:

$$\min F(X) = (f_1(X), f_2(X), ..., f_m(X)), \quad (1)$$

$$\text{s.t. } g_j(X) \leq 0,\ j = 1, ..., K. \quad (2)$$

In the equations, $f_i(X)$ are the objectives, $g_j(X)$ are the constraints, and X is the n-dimensional decision variable. Usually, there is no single ideal solution X^0 that minimizes all objectives simultaneously. Instead, a finite or infinite number of Pareto-optimal solutions can be obtained for the multi-objective optimization problems. A solution X is said to be Pareto-optimal if and only if there is no X' in the whole search space such that for all $i = 1, 2..., m$, $f_i(X') \leq f_i(X)$. In other words, there does not exist X' that dominates X.

Traditional multi-objective optimization algorithms combines multiple objectives into a scalar objective function as follows:

$$F = \sum_{i=1}^{m} w_i f_i, \quad (3)$$

where $w_i \geq$ is the weight for the i-th objective.

In the recent years, a number of multi-objective evolutionary algorithms (MOEAs) have been proposed by incorporating the concept of Pareto-optimality [11]. The main advantage of MOEAs is that they are able to achieve a set of Pareto-optimal solutions in one single run. One major development in MOEA research is the introduction of the elitism strategy, which can be realized by maintaining a second population (or archive) [31], or by combining parent and offspring populations before selection [11]. The inclusion of local search in multi-objective evolutionary algorithm has also proved to be able to improve the performance effectively, which can largely be attributed to the fact that local search implicitly takes advantage of the regularity in the distribution of the Pareto-optimal solutions [21]. It is advocated in [23] that a more explicit way of exploiting the regularity in the distribution of Pareto-optimal solutions is to build a model that captures the regularity, and then use this model to guide the search. This idea has proved to be successful [37], where a model composed of a deterministic part capturing the regularity and a probabilistic part describing the local dynamics has been suggested to guide the evolutionary search. In this work, we adopt a variant of the NSGA-II [11] for solving the multi-objective machine learning problems.

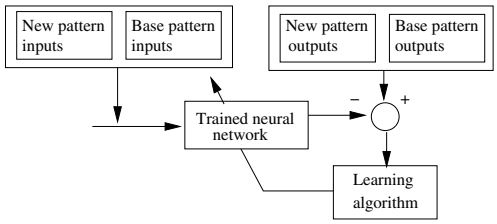


Fig. 1. Diagram of rehearsal.

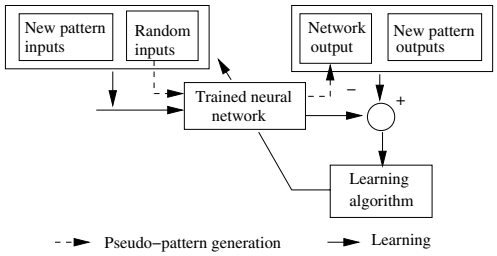


Fig. 2. Diagram of pseudo-rehearsal.

III. Multi-objective Formulation of Pseudo-rehearsal

As discussed in the Introduction, rehearsal and pseudo-rehearsal are two effective approaches to avoid catastrophic forgetting, refer to Fig. 1 and Fig. 2, respectively, for an illustration of rehearsal and pseudo-rehearsal.

One problem that arises in pseudo-rehearsal is the runaway effect. We argue that the runaway effect can mainly be attributed to two reasons. First, it is assumed that the pseudo-patterns should be able to embody the main features of the base patterns. However, when the inputs of the pseudo-patterns are generated randomly, this assumption is valid only if the learning model (e.g., a neural network) is able to generalize perfectly, which is very unlikely for high-dimensional problems. Second, conventional learning is mostly single-objective. Thus, learning the new patterns and learning the pseudo-patterns simultaneously may be two conflicting targets. In other words, the learning of new patterns might have negative influence on the learning of the pseudo-patterns, and vice versa, as demonstrated in [29].

To address these problems, we have taken two measures. The first measure we take is to check the similarity of the random input patterns to the base patterns. We will show in Section IV that pseudo-rehearsal works only if the input of the pseudo-patterns are sufficiently similar to the base patterns. In some of the experiments, we even assume that the input of the base patterns is known, though the output must be generated by the trained model. In addition, the multi-

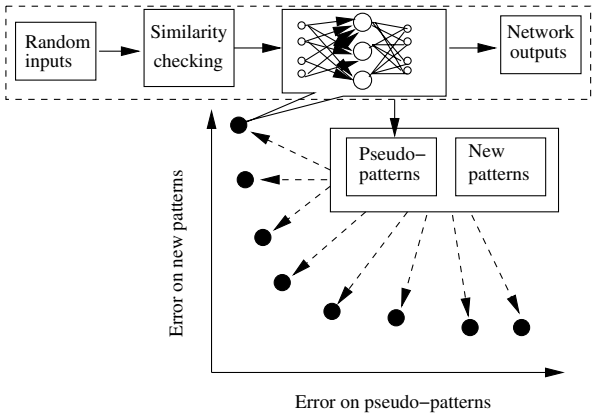


Fig. 3. Pseudo-pattern generation through multi-objective optimization.

objective learning approach is adopted where the error on the new patterns and the error on the pseudo-patterns are used as two objectives. The pseudo-patterns are generated from the neural network with the minimal error on the base-patterns. Since the multi-objective approach is adopted, the neural network with the minimal error on the base patterns will survive through the generations, see Fig. 3. In this way, the runaway effect can be avoided.

IV. Evolutionary Single-objective and Multi-objective Neural Network Learning

In this work, single-objective (SO) learning and the multi-objective (MO) learning are achieved with the help of an evolutionary algorithm. Both the parameters (weights) and the structure (connections) of the neural network are encoded in the chromosome and are evolved during the learning. In the SO learning, the mean square error (MSE) on the patterns is used as the fitness function, while in the MO learning, the MSE on base and new patterns are used as two separate objectives.

A. Representation of the Neural Networks

A connection matrix and a weight matrix are employed to describe the structure and the weights of the neural networks. The connection matrix specifies the structure of the network, whereas the weight matrix determines the strength of each connection. Assume that a neural network consists of M neurons in total, including the input and output neurons, then the size of the connection matrix is $M \times (M+1)$, where an element in the last column indicates whether a neuron is connected to a bias value. In the matrix, if element $c_{ij}, i = 1, ..., M, j = 1, ..., M$ equals 1, it means that there is a connection between the i-th and j-th neuron and the signal flows from neuron j to neuron i. If $j = M+1$, it indicates that there is a bias in the i-th neuron. Fig. 4 illustrates a connection matrix and the corresponding network structure. It can be seen from the figure that the network has two input neurons, two hidden neurons, and one output neuron. In addition, both hidden neurons have a bias.

The strength (weight) of the connections is defined in the weight matrix. Accordingly, if c_{ij} in the connection matrix

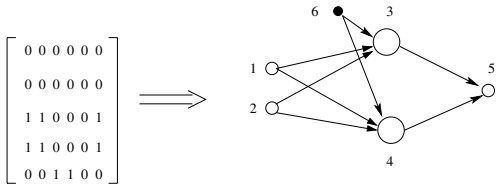


Fig. 4. A connection matrix and the corresponding network structure.

equals zero, the corresponding element in the weight matrix (w_{ij}) must be zero too.

B. Genetic Operators

A genetic algorithm is used for optimizing the structure and weights of the neural networks. Binary coding is adopted representing the neural network structure and real-valued coding for encoding the weights. Five genetic operations have been introduced in the evolution of the neural networks, four of which mutate the connection matrix (neural network structure) and one of which mutates the weights. The four mutation operators are the insertion of a hidden neuron, deletion of a hidden neuron, insertion of a connection, and deletion of a connection. The probability of deleting a connection between an input node I_i and a hidden node (H_j) is

$$p_{c_{ij}} = \frac{1}{1 + w_{ij}}, \tag{4}$$

and the probability of deleting a hidden node (H_j) is roughly inversely proportional to the root of squared sum of the weights fanning into the nodes:

$$p_{H_j} = \frac{1}{1 + \sqrt{\sum_{i=1}^{n} w_{ij}^2}}, \tag{5}$$

where n is the number of inputs. A Gaussian-type mutation is applied to mutate the weight matrix:

$$w_{ij}(t+1) = w_{ij}(t) + N(0, \sigma^2), \tag{6}$$

where σ is the standard deviation of the Gaussian noise.

C. Life-time learning

After mutation, an improved version of the Rprop algorithm, Rprop$^+$ [19] is employed to train the weights. This can be seen as a kind of life-time learning within a generation. Notice that in sequential learning, only the base patterns are learned in the first phase. In the second phase, both the pseud-patterns and the new patterns are learned. We will show that the way to combine the pseudo-patterns and the new patterns can have great influence on the results of the evolution. In the following, we describe briefly the Rprop+ learning algorithm used in this work.

Let w_{ij} denote the weight connecting neuron j and neuron i, then the change of the weight (Δw_{ij}) in each iteration is as follows:

$$\Delta w_{ij}^{(t)} = -\text{sign}\left(\frac{\partial E^{(t)}}{\partial w_{ij}}\right) \cdot \Delta_{ij}^{(t)}, \tag{7}$$

where $sign(\cdot)$ is the sign function, $\Delta_{ij}^{(t)} \geq 0$ is the step-size, which is initialized to Δ_0 for all weights. The step-size for each weight is adjusted as follows:

$$\Delta_{ij}^{(t)} = \begin{cases} \xi^+ \cdot \Delta_{ij}^{(t-1)} & , \quad \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \xi^- \cdot \Delta_{ij}^{(t-1)} & , \quad \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & , \quad \text{otherwise} \end{cases}, \quad (8)$$

where $0 < \xi^- < 1 < \xi^+$. To prevent the step-sizes from becoming too large or too small, they are bounded by $\Delta_{\text{min}} \leq \Delta_{ij} \leq \Delta_{\text{max}}$.

One exception must be considered. After the weights are updated, it is necessary to check if the partial derivative changes sign, which indicates that the previous step might be too large and thus a minimum has been missed. In this case, the previous weight change should be retracted:

$$\Delta w^{(t)} = -\Delta_{ij}^{(t-1)}, \text{ if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} < 0. \quad (9)$$

Recall that if the weight change is retracted in the t-th iteration, the $\partial E^{(t)}/\partial w_{ij}$ should be set to 0.

In reference [19], it is argued that the condition for weight retraction in equation (9) is not always reasonable. The weight change should be retracted only if the partial derivative changes sign and if the approximation error increases. Thus, the weight retraction condition in equation (9) is modified as follows:

$$\Delta w^{(t)} = -\Delta_{ij}^{(t-1)}, \quad \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} < 0, \text{ and } E^{(t)} > E^{(t-1)}. \quad (10)$$

It has been shown on several benchmark problems in [19] that the modified Rprop (termed as Rprop$^+$ in [19]) exhibits consistent better performance than the Rprop algorithm.

D. Selection

In SO learning, a tournament selection with a tournament size of 4 is employed. The tournament selection is carried out as follows. Four individuals are randomly chosen from the offspring and then the best one among the 4 individuals is chosen as the parent for the next generation. By best, we mean in this work the solution with the minimal MSE on the training samples. This process is repeated for P times, where P is the population size.

A major difference between SO learning and MO learning algorithms is the selection strategy. In the MO learning, the parent and offspring individuals are combined and all individuals are assigned a rank (r_i) and a crowding distance (d_i) according to the non-dominated sorting and the crowded distance sorting suggested in NSGA-II [11]. After sorting, the crowded tournament selection is applied. In the crowded tournament selection, two individuals are picked out randomly, the one that wins the tournament is passed to the next generation. A solution wins the tournament either if it has a better rank, or if it has the same rank but a better crowding distance. In this context, a lower rank is better and a larger crowding distance is better. The readers are referred to [11] for further details.

V. Simulation Studies

In this section, we compare the conventional single-objective (SO) approach to pseudo-rehearsal and the proposed multi-objective (MO) approach to alleviating catastrophic forgetting in neural network learning. Similar to the experimental setups in the literature [33], the neural network is required to memorize two sets of binary patterns. Both base patterns and new patterns consist of 25 pairs of random patterns with 10 inputs and 10 outputs.

The population size of the evolutionary algorithm is 100. Between each generation, 50 iterations of learning using the Rprop$^+$ algorithm are performed. In the SO approach, the cost function of the life-time learning is the MSE on the patterns to be learned. For the Rprop$^+$ algorithm, the step-sizes are initialized to 0.01 and bounded between $[0, 50]$ during the adaptation, and $\xi^- = 0.5$, $\xi^+ = 1.2$. The maximal number of hidden nodes is set to 7. In the MO approach, we minimize either the MSE on the union of the base and the new patterns in case of direct rehearsal, or the MSE on the union of the pseudo-patterns and the new patterns in pseudo-rehearsal. We will discuss this setup afterward since this setup turns out to be not ideal for the MO learning approach.

In the evolution, an equal probability of 0.25 is implemented for the five genetic operators, namely, node insertion, node deletion, connection insertion and connection deletion, and weight jogging. When new weights are inserted, they are initialized randomly and uniformly in the interval of $[-0.2, 0.2]$. Within the first 100 generations, the neural network attempts to learn the base patterns. From the 101-th to the 400-th generations, the new patterns are presented to the neural network. The target is that the neural network is able to remember the new patterns without forgetting the base patterns. In our simulations, we use a tolerance level of 0.3 to judge if a pattern is memorized. That is to say, if a desired output is 0, then the output of the neuron should be smaller than 0.3. If the desired output is 1.0, then the neural output should be larger than 0.7. If all 10 output neurons satisfy the tolerance, we say that the pattern is memorized correctly.

The first experiment we do is to show that catastrophic forgetting does exist in evolutionary sequential learning, see Fig. 5. Different to conventional learning methods, we notice that the base patterns are completely forgotten within one generation. This is due to the fact that within one generation, the structure has been changed and 50 iterations of learning are conducted. An interesting point is that it takes only 6 generations to memorize 21 of the 25 new patterns, while it has taken 63 generations to learn the 25 base patterns. We can assume that the neural network structure evolved for the base patterns is also sub-optimal for the new patterns due to the similarity in nature between the new and base patterns. The network fails to memorize 4 of the new patterns, probably due to the fact that there is inconsistency among the new patterns, recalling that all the associative patterns are generated randomly.

In a second experiment, we perform direct rehearsal using

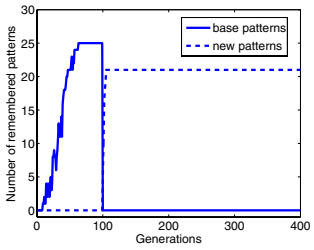
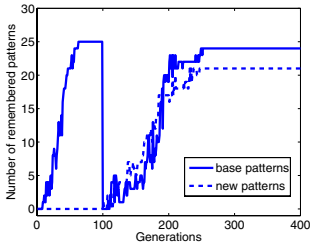
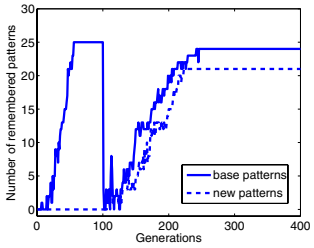


Fig. 5. Catastrophic forgetting in evolutionary learning of random patterns.



(a)



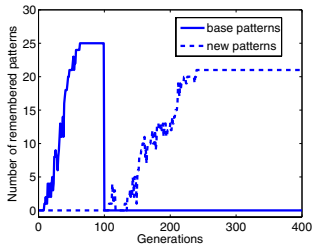
(b)

Fig. 6. Direct rehearsal. (a) SO approach, and (b) MO approach.

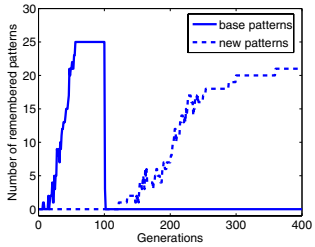
the SO approach and the MO approach. In this case, we assume that the base patterns are still available for training when learning the new patterns. Therefore, the base patterns are simply combined with the new patterns for the neural network to learn from generations 101 to generations 400. The results are shown in Fig. 6. It can be seen that in the SO approach, the network memorized 24 of the 25 base patterns, and 21 of the 25 new patterns within approximately 250 generations. Similar results are observed in the MO approach. On the other hand, we find that the learning speed is slowed down greatly when the base patterns are learned together with the new patterns. Recall, however, that direct rehearsal is not always practical in machine learning and biologically implausible.

Now let us investigate if pseudo-rehearsal is able to avoid catastrophic forgetting. According to [32], we generate 25 pseudo-patterns by creating random inputs. From Fig. 7, we see that neither the SO approach, nor the MO approach is able to avoid catastrophic forgetting, since in both cases, the base patterns are forgotten completely.

To improve the chance of a successful avoidance or alleviation of catastrophic forgetting, we impose similarity checking of the generated random inputs to those of the base patterns. This operation is supported by the findings in



(a)

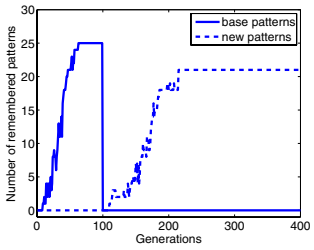


(b)

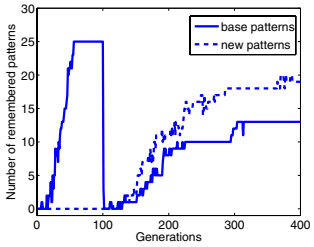
Fig. 7. Pseudo-rehearsal using random inputs. (a) SO approach, and (b) MO approach.

biology that in the brain, only relevant cells in the neocortex are continuously activated by the hippocampal-neocortical connections so that the connections of these activated cells in separate regions of the neocortex are strengthened [4], [36]. In this work, we assume that the inputs of the base patterns are available for similarity checking. The similarity checking here is functionally similar to the reverberating neural networks suggested in [7], where the random inputs are fed into an auto-associative sub-network so that the random inputs will converge to the most similar base pattern input.

The issue now is that how large the similarity should be to ensure successful pseudo-rehearsal. In the simulations, we find that for this experimental setup, a successful pseudo-rehearsal can be observed in the MO approach only if the input similarity between the pseudo-pattern inputs and the base pattern inputs are larger than 0.8. Unfortunately, catastrophic forgetting persists in the SO approach even if the similarity is 1, which means that the inputs of the pseudo-patterns are the same as the base pattern inputs. This is actually obvious if we take a look at the learning procedure in Fig. 8 (a), where the similarity is 0.8. It can be seen that all base patterns are forgotten within one generation. In this case, it is unrealistic to expect that pseudo-patterns generated from the network can be of any help for learning the base patterns. It should also be pointed out that an elitist selection does not help in the SO approach. In the MO case, the situation is quite different. Thanks to the multi-objective selection criterion, the network with the minimal error on the base patterns are maintained. Thus, the information on the base patterns are transferred to other neural networks gradually as the learning proceeds, refer to Fig. 8 (b), where 13 out of 25 base patterns, and 20 out of 25 new patterns are memorized. The number of forgotten base patterns is



(a)



(b)

Fig. 8. Pseudo-rehearsal using inputs similar to the base patterns. The required similarity is equal to or larger than 0.8. (a) SO approach, and (b) MO approach.

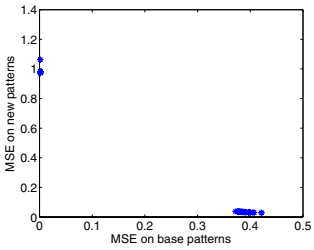


Fig. 9. Pareto-optimal solutions from the MO approach. The life-time learning minimizes the union of the pseudo-patterns and the new patterns at a probability of 0.5.

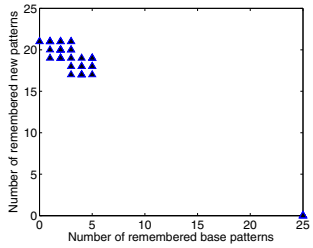


Fig. 10. Pareto-optimal solutions w.r.t. the number of remembered patterns. The life-time learning minimizes the union of the pseudo-patterns and the new patterns at a probability of 0.5.

still quite high though, however, it should be noticed that the number of new patterns need to be learned is also quite high compared to the experimental setups in the literature.

The above observations are true under the assumption that the inputs of the pseudo-patterns are generated once and the outputs of the pseudo-patterns need to be re-calculated in every generation using the neural network having the minimal error on the base patterns.

Although the MO approach is able to memorize part of the base patterns, the final generation does not provide us a variety of neural networks with different errors on base and new patterns. Rather, the entire population have converged to one solution.

Taking a closer look at the MO approach, we find that the life-time learning based on the Rprop$^+$ is still single-objective (learning the union of the pseudo-patterns and the new patterns), though the evolutionary algorithm tries to optimize two objectives simultaneously. We therefore make a small modification to the simulation setup. Instead of always learning the union of the pseudo-patterns and the new patters, the network learnings the union of the pseudo-patterns and the new patterns at a probability of 0.5 The non-dominated solutions of the obtained neural networks are shown in Fig. 9 and Fig. 10 with respect to the MSE and the number of remembered patterns, respectively. We see that we do achieved 19 different solutions, the number of remembered base patterns are relatively low.

In a further experiment, we change the probability of learning the union of the pseudo-patterns and new patters to 2/3 and that of learning the new patterns to 1/3. The results show that we can generate neural networks that memorize as many as 11 base patterns while learning 21 of 25 new patterns, however, the diversity is greatly reduced again, refer to Fig. 11.

It turns out that life-time learning minimizing the union of the pseudo-patterns and the new patterns tends to reduce the diversity of the population. To rectify this weakness, we modified the setup further so that either the pseudo-patterns or the new patterns are learned randomly. The neural network learns the pseudo-patterns at a probability of 1/3 and the new patterns at a probability of 2/3. The non-dominated solutions w.r.t. the MSE on the pseudo-patterns and the new patterns are shown in Fig. 12, denoted by stars. To verify how much the pseudo-patterns are reflecting the base patterns, we also plot the same solutions w.r.t. the MSE on the base patterns and the new patterns, denoted by circles. It can be seen that the errors on the pseudo-patterns and on the base patterns are quite different.

The solutions measured by the number of remembered base patterns and the number of remembered new patterns are plotted in Fig. 13. It can be seen that except for one solution that remembers 1 base pattern and 1 new pattern simultaneously, no neural network is able to remember both base patterns and new patterns, which is somewhat surprising in contrast to to the tradeoff on the MSE. We have also performed additional simulations with a different probability for learning different patterns during life-time learning, but no significantly better results have been achieved.

Through the above simulations, we find that if we use a combination of the pseudo-patterns and the new patterns during life time learning, the population will converge to few solutions that can memorize both base and new patterns, and the diversity of the population is lost. In contrast, if we separate pseudo-patterns and new patterns during the

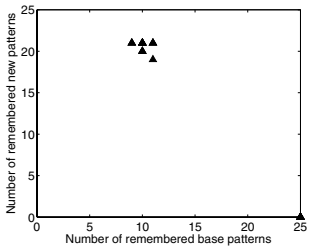


Fig. 11. Pareto-optimal solutions w.r.t. the number of remembered patterns. The life-time learning minimizes the union of the pseudo-patterns at a probability of 2/3 and the new patterns at a probability of 1/3.

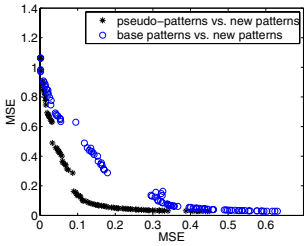


Fig. 12. Pareto-optimal solutions from the MO approach. The life-time learning minimizes the MSE on the pseudo-patterns at a probability of 1/3 and that on the new patterns at a probability of 2/3.

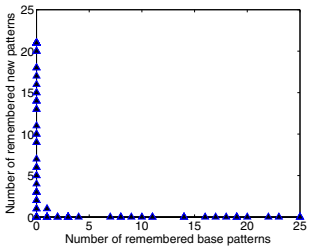


Fig. 13. Pareto-optimal solutions w.r.t. the number of remembered patterns.

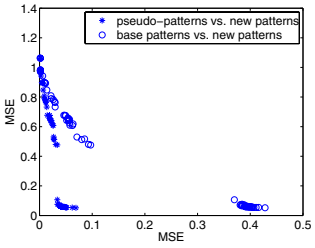


Fig. 14. Pareto-optimal solutions w.r.t. the MSE on base, new and pseudo patterns.

life-time learning, a large number of neural networks that trade off between the error on the base patterns and the new patterns can be obtained. Unfortunately, these networks can memorize either base patterns or new patterns only.

Finally, we combine the above setups during the learning. In other words, three different sub-tasks are learned at random, namely, the pseudo-patterns, the union of the pseud-patterns and new patterns, and the new patterns. The results are shown in Figs. 14 and 15, where the former shows the tradeoff between the MSE on the pseudo-patterns and that on the new patterns, and the latter shows the number of the base and new patterns remembered by each non-dominated solution. We see that the diversity has been improved somehow, but again, the MSE on the pseudo-patterns is quite different from that of the base patterns. Besides, neural networks that memorize the base patterns well behaves poorly on the new patterns.

Interestingly, if we assume the input of the pseudo-patterns are known in generating pseudo-patterns, then quite different results are obtained. In this case, the MSE on the pseudo-patterns is much closer to the MSE on the base patterns, and a few neural networks that perform well on both base and new patterns have been obtained, refer to Figs. 16 and 17.

VI. Conclusions

Avoiding catastrophic forgetting is an important issue when connectionist networks are used to simulate the memory mechanisms of the brain. This paper suggests a method for alleviating catastrophic forgetting using multi-objective pseudo-rehearsal. The advantage of the MO approach is that the network that has the minimal error on the base patterns can be maintained, which makes it possible to avoid the runaway effect.

Several questions remain open concerning the multi-objective pseudo-rehearsal method suggested in this paper. First, it is unclear how pseudo-patterns can be generated in a biologically plausible way. Though the auto-associative reverbrating network in [7] is a possible approach, it is still impossible to guarantee that pseudo-pattern inputs are sufficiently similar to the base patterns. Second, life-time learning poses a new challenge in multi-objective learning. At the first glance, life-time learning is similar to local search in evolutionary multi-objective optimization. However, the genetic operators in multi-objective neural network learning mainly change the structure of the networks. Thus, the genetic search is quite rough and the life-time learning plays an essential role in finding neural networks with acceptable performance on accuracy. It should be pointed out that the multi-objective learning in this work is quite different to the existing research on multi-objective learning where the main goal is to improve the generalization performance, and therefore the second objective introduced is usually related to the generalization ability, e.g., minimizing complexity, maximizing diversity, or minimizing the error on a test data set, where life-time learning does not cause a big problem. Third, it is found that randomly generated pseudo-patterns makes it quite difficult for the networks to learn the new patterns too. How to resolve this difficulty should also be investigated.

Acknowledgment

The authors would like to thank E. Körner for inspiring discussions on the memory mechanisms in human brain.

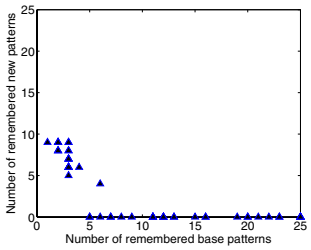


Fig. 15. Pareto-optimal solutions w.r.t. the number of remembered base and new patterns.

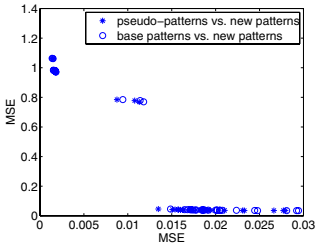


Fig. 16. Pareto-optimal solutions w.r.t. the MSE on base and new patterns.

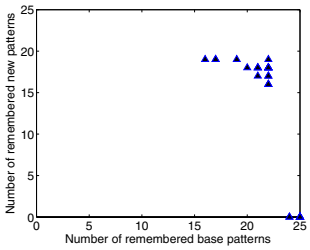


Fig. 17. Pareto-optimal solutions w.r.t. the number of remembered base and new patterns.

REFERENCES

[1] H.A. Abbass. Speeding up back-propagation using multi-objective evolutionary algorithms. *Neural Computation*, 15(11):2705–2726, 2003

[2] H.A. Abbass. Pareto neuro-evolution: Constructing ensembles of neural networks using multi-objective optimization. In: *Congress on Evolutionary Computation*. pp.2074-2080, 2003

[3] W.C. Abraham, A. Robins. Memory retention - the synaptic stability versus plasticity dilemma. *Trends in Neuroscience*, 28(2), 73–78, 2005

[4] P. Alvarez, L.R. Squire. Memory consolidation and the medial temporal lobe: a simple network model. *Proc. of National Academic Science of USA*, 91, 7041–7045, 1994

[5] V.R. de Angulo, C. Torras. On-line learning with minimal degradation in Feedforward networks. *IEEE Transactions on Neural Networks*, 6(3): 657–668, 1995

[6] V.R. de Angulo, C. Torras. A framework to deal with interference in connectionist systems. In: *International Conference on Neural Networks*, LNCS 2415, pp.1339–1344, 2002

[7] B. Ans and S. Rousset. Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Life Sciences*, 320:989–997, 1997

[8] B. Ans. Sequential learning in distributed neural networks without catastrophic forgetting: A single and realistic self-refreshing memory can do it. *Neural Information Processing*, 4(2): 27–32, 2004

[9] E.S. Boyden et al. Cerebellum-dependent learning: The role of multiple plasticity mechanisms. *Annual Review of Neuroscience*, 27:581–609, 2004

[10] A. Charandra and X. Yao. DIVACE: Diverse and accurate ensemble learning algorithm. In: *IDEAL'04*, pp.619-625. LNCS 3177, 2004

[11] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, 2001

[12] J. Fieldsend and S. Singh. Pareto evolutionary neural networks. *IEEE Transactions on Neural Networks*, 16(2):338–354, 2005

[13] M. Frean and A. Robins. Catastrophic forgetting in simple networks: An analysis of the pseudo-rehearsal solution. *Network: Computation Neural Systems*, 10:227–236, 1999

[14] R.M. French. Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. In: *Proc. of the 13th Annual Conf. of the Cognitive Science Society*, pp.173–178, 1991

[15] R.M. French. Pseudo-recurrent connectionist networks: An approach to the "sensitivity-stability" dilemma. *Connection Science*, 9:353–397, 1991

[16] R.M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999

[17] M.A. Gluck and C.E. Meyers. *Gateway to Memory*. A Bradford Book, Cambridge, MA, 2001

[18] J. Handl and J. Knowles. Exploiting the trade-off - The benefits of multiple objectives in data clustering. *Evolutionary Multi-Criterion Optimization*, LNCS 3410, pages 547–560, 2005

[19] C. Igel and M. Hüsken. Improving the Rprop learning algorithm. In *Proc. of the 2nd ICSC Int. Symposium on Neural Computation*, pages 115–121, 2000

[20] C. Igel. Multi-objective model selection for support vector machines. *Evolutionary Multi-Criterion Optimization*, LNCS 3410, pages 534–546, 2005

[21] Y. Jin, B. Sendhoff. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. *Congress on Evolutionary Computation*, 1910–1917, 2003

[22] Y. Jin, T. Okabe, B. Sendhoff. Neural network regularization and ensembling using multi-objective evolutionary algorithms. In: *Congress on Evolutionary Computation*, pages 1–8. IEEE, 2004

[23] Y. Jin. Rethinking multi-objective evolutionary algorithms. *Invited Talk*, Dagstuhl Seminar on Theory of Evolutionary Algorithms, Schloss Dagstuhl, February 2004

[24] Y. Jin, B. Sendhoff, E. Körner. Evolutionary multi-objective optimization for simultaneous generation of signal-type and symbol-type representations. *Evolutionary Multi-Criterion Optimization*, LNCS 3410, pages 752–766, 2005

[25] Y. Jin (editor). Multi-objective Machine Learning. Springer, Berlin, 2006

[26] Y. Jin, Simultaneous generation of accurate and interpretable neural network classifiers. In: *Multi-Objective Machine Learning*, Y. Jin(ed.), pp.291-312, 2006

[27] S.B. Klein et al. Decision and evolution of memory: Multiple systems, multiple functions. *Psychological Review*, 109(2):306–329, 2002

[28] M. McCloskey and N.J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:109–165, 1989

[29] M. Meeter. Control of consolidation in neural networks: avoiding runaway effects. *Connection Science*, 15(1):45–61, 2003

[30] K.A. Norman and R.C. O'Reilly. Modeling hippocampal and neocortical contributions to recognition memory: A complementary-learning-system approach. *Psychological Review*, 110(4):611–646, 2003

[31] S. Obayashy, S. Takahashi, Y. Takeguchi. Niching and elitist models for MOGAs. *Parallel Problem Solving from Nature*, LNCS 1498, 260–269, 1998

[32] A. Robins. Catastrophic forgetting, rehearsal, and pseudo-rehearsal. *Connection Science*, 7, 123–146, 1995

[33] T.T. Rogers and J.L. McClelland. Semantic Cognition – A Parallel Distributed Processing Approach. MIT, 2004

[34] T. Seipone, J. A. Bullinaria. The evolution of minimal catastrophic forgetting in neural systems. In: Proc. of the 27th Annual Conf. of Cognitive Science Society, pages 1991–1996, 2005

[35] R. Ratcliff. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, 97:285–308, 1990

[36] G.M. Wittenberg, M.R. Sullivian, J.Z. Tsien. Synaptic reentry reinforcement based network model for long-term memory consolidation. *Hippocampus*, 12, 637–647, 2002

[37] A. Zhou, Q. Zhang, Y. Jin, B. Sendhoff, E. Tsang. A model-based evolutionary algorithm for bi-objective optimization. *Congress on Evolutionary Computation*, 2569–2575, IEEE, 2005