© Académie des sciences / Elsevier, Paris
Neurosciences / *Neurosciences*

Avoiding catastrophic forgetting by coupling two reverberating neural networks

L'oubli catastrophique évité par couplage de deux réseaux neuronaux réverbérants

BERNARD ANS*, STÉPHANE ROUSSET

Laboratoire de psychologie expérimentale (CNRS EP 617), université Pierre-Mendès-France, BP 47, 38040 Grenoble cedex 9, France

RÉSUMÉ

Les procédures d'apprentissage de descente de gradient sont le plus souvent utilisées dans la modélisation de réseaux neuronaux. Lorsque ces algorithmes (e.g., la rétropropagation) sont appliqués à des tâches d'apprentissage séquentiel, un défaut majeur, appelé oubli catastrophique (ou interférence catastrophique), survient généralement : lorsqu'un réseau ayant déjà appris un premier ensemble d'items est ensuite entraîné sur un deuxième ensemble d'items, alors l'information nouvellement apprise peut complètement détruire celle antérieurement apprise. Pour éviter ce défaut peu plausible, il est ici proposé une architecture à deux réseaux neuronaux où les nouveaux items sont appris par un premier réseau conjointement avec des pseudo-items internes provenant d'un second réseau. Comme il est démontré que ces pseudo-items reflètent la structure des items antérieurement appris par le premier réseau, le modèle implémente ainsi un processus de rafraîchissement par l'ancienne information. Le point crucial est que ce mécanisme de rafraîchissement repose sur des réseaux neuronaux réverbérants qui n'ont besoin que de stimulations aléatoires pour opérer. Le modèle fournit ainsi un moyen de réduire d'une manière spectaculaire l'interférence rétroactive tout en conservant la nature essentiellement distribuée de l'information et propose une hypothèse originale et plausible sur la manière de « copier et coller » la mémoire distribuée au sein d'une structure cérébrale vers une autre.

***Mots clés :** modèles connectionnistes de l'apprentissage et de la mémoire, oubli catastrophique, réseaux neuronaux réverbérants, processus de ré-injection, pseudo-rafraîchissement*

ABSTRACT

Gradient descent learning procedures are most often used in neural network modeling. When these algorithms (e.g., backpropagation) are applied to sequential learning tasks a major drawback, termed catastrophic forgetting (or catastrophic interference), generally arises: when a network having already learned a first set of items is next trained on a second set of items, the newly learned information may completely destroy the information previously learned. To avoid this implausible failure, we propose a two-network architecture in which new items are learned by a first network concurrently with internal pseudo-items originating from a second network. As it is demonstrated that these pseudo-items reflect the structure of items previously learned by the first network, the model thus implements a refreshing mechanism using the old information. The crucial point is that this refreshing mechanism is based on reverberating neural networks which need only random stimulations to operate. The model thus

Note présentée par Michel Imbert
Note remise le 25 juillet 1997, acceptée après révision le 3 novembre 1997

*Correspondence and reprints

provides a means to dramatically reduce retroactive interference while conserving the essentially distributed nature of information and proposes an original but plausible means to 'copy and paste' a distributed memory from one place in the brain to another.

Key words: connectionist models of learning and memory, catastrophic forgetting, reverberating neural networks, re-injection process, pseudo-rehearsal.

VERSION ABRÉGÉE

Les procédures d'apprentissage de *descente de gradient* sont couramment utilisées dans la modélisation des réseaux neuronaux. Lorsque ces algorithmes sont appliqués à des tâches d'apprentissage séquentiel, un défaut majeur appelé *oubli catastrophique* (ou interférence catastrophique) survient généralement : lorsqu'un réseau connexionniste ayant déjà appris un premier ensemble d'items est ensuite entraîné sur un deuxième ensemble d'items, alors l'information nouvellement apprise peut complètement détruire celle antérieurement apprise. Parmi les différentes voies qui ont été explorées pour éviter ce défaut psychologiquement peu plausible, celle récemment proposée par Robins apparaît la plus prometteuse. En effet, cette approche permet de réduire l'ampleur de l'interférence catastrophique tout en conservant la nature distribuée de la représentation de l'information au sein des réseaux, propriété essentielle requise pour conserver leur aptitude à généraliser.

Le principe de base de la méthode, appelée par son auteur *mécanisme de pseudo-répétition à balayage*, est le suivant. Une fois qu'un réseau a complètement appris un premier ensemble d'items, il est ensuite stimulé par une activité aléatoire, ce qui produit des pseudo-items dont un certain nombre sont stockés dans une pseudo-base. Lorsque le réseau est ensuite entraîné sur un nouvel ensemble d'items, il est parallèlement rafraîchi par les pseudo-items de la pseudo-base, pseudo-items qui sont censés refléter la structure de l'information antérieurement apprise.

Cependant, deux critiques peuvent être formulées à l'égard de la précédente approche. La première est que le mécanisme de rafraîchissement n'est pas réalisé en termes connexionnistes car la pseudo-base est en fait constituée par une simple liste de vecteurs stockés localement à différentes adresses-mémoire du calculateur utilisé pour les simulations. La deuxième critique est que la réduction attendue de l'interférence rétroactive est jugée à notre sens insuffisante dans les simulations de tâches d'apprentissage séquentiel où les associations à apprendre sont arbitraires. La présente note répond donc à un double objectif. i) Construire une architecture connexionniste permettant de réaliser le mécanisme de pseudo-rafraîchissement exclusivement en termes de « neurones » et de « poids synaptiques ». ii) Proposer un processus de traitement de l'information neurale permettant de capturer d'une manière optimale la structure profonde de l'information distribuée au sein des réseaux neuronaux, ceci afin de réellement minimiser l'interférence rétroactive.

Concernant le premier point, nous proposons une architecture constituée de deux réseaux neuronaux à couche cachée. Ces réseaux, notés NET1 et NET2, peuvent dans un premier temps être considérés comme de classiques réseaux unidirectionnels, où chaque unité d'entrée est connectée à toutes les unités cachées, elles-mêmes connectées à toutes les unités de sortie. Le réseau NET1, seul réceptif aux items externes, est celui qui doit apprendre séquentiellement une première base d'associations (Base *A*) *puis* une deuxième base (Base *B*), les associations étant constituées par des couples stimulus/réponse. Une base est considérée comme apprise lorsque chacun des stimuli qui la composent, présenté en entrée du réseau, génère sur la couche de sortie la réponse désirée (ou cible). Pour expliquer simplement le fonctionnement du système, on se place dans une situation initiale où NET1 a déjà appris complètement la Base *A* et où NET2 est encore « vide » (i.e., avec des poids de connexion aléatoires). Entre alors en action un premier processus, Processus (I), dans lequel NET1 ne peut être réceptif aux activations externes mais où sa couche d'entrée est constamment activée par une stimulation aléatoire interne. La succession des patterns aléatoires d'entrée et des sorties induites dans NET1 est continuellement transmise au réseau NET2 qui est entraîné sur ces pseudo-associations, les sorties générées par NET1 jouant le rôle de pseudo-cibles pour NET2. L'apprentissage au sein de NET2 s'arrête lorsque entre en jeu un deuxième processus, Processus (II), dans lequel NET1 devient à nouveau réceptif aux activations externes, en l'occurrence aux items de la Base *B*. C'est maintenant NET2 qui est continûment activé par une stimulation aléatoire induisant une succession de pseudo-associations qui sont transmises à l'autre réseau. NET1 est donc simultanément entraîné sur les associations de la Base *B* et sur les pseudo-associations provenant continuellement de NET2, pseudo-associations qui devraient refléter la Base *A* antérieurement apprise par NET1. Le système réalise ainsi, mais cette fois d'une manière exclusivement connexionniste, le mécanisme de pseudo-rafraîchissement formulé par Robins.

Quant au second objectif de cette note, nous pensons qu'un simple passage d'une activité aléatoire dans un réseau neuronal unidirectionnel est largement insuffisant pour capturer d'une manière optimale l'information distribuée au sein des poids de connexion. Nous proposons de substituer à la structure de NET1 et NET2 une structure de type *réverbérante*, où, par rapport à celle précédemment décrite, la couche cachée est de plus en connexion totale avec l'entrée. Une première différence est que pendant les phases d'apprentissage les réseaux apprennent non seulement des hétéro-associations (stimulus/réponse ou pseudo-stimulus/pseudo-réponse) mais aussi des auto-associations (stimulus/stimulus ou pseudo-stimulus/pseudo-stimulus). Une deuxième différence, cruciale, réside dans la manière de générer des pseudo-associations.

Lorsque l'un ou l'autre des réseaux reçoit en entrée une activation aléatoire, l'activité induite doit préalablement se réverbérer un certain nombre de fois, entre les couches cachée et d'entrée, avant que le couple entrée–sortie résultant soit transmis à l'autre réseau pour apprentissage. Il est ainsi attendu que ce processus de *ré-injection* multiple converge vers les attracteurs du réseau, ce qui doit permettre une capture optimale de la structure profonde inscrite dans sa connectivité, et donc conduire à un processus de pseudo-rafraîchissement beaucoup plus efficace.

Le comportement du système connexionniste proposé a été simulé à partir d'une tâche d'apprentissage séquentiel classiquement utilisée pour mettre en évidence l'oubli catastrophique dans le cadre de la rétropropagation. Les résultats obtenus en utilisant le processus de réverbération montrent clairement que l'interférence rétroactive peut être pratiquement éliminée. En revanche, lorsque le processus de réverbération n'est pas utilisé, l'interférence rétroactive reste élevée et les apprentissages se révèlent laborieux, voire impossibles.

Introduction

Learning and memory processes in the field of neural network modeling (or connectionism) are most often achieved through one or another of several gradient descent adaptive algorithms, of which the most popular and widely used is the iterative and supervised learning procedure called backpropagation [1]. Such algorithms are generally used to associate input patterns to specified target output patterns. Typically, a set of input–target pairs (the training base) is repeatedly presented to a connectionist network and at each presentation (or iteration) of a pair, the connection weights within the network are adjusted so as to minimize the error between the output actually computed by the network (in response to the input) and the desired output pattern (the target provided in each pair).

It is well-known that when *gradient descent* learning procedures are applied to sequential learning tasks a major drawback, termed *catastrophic forgetting* (or catastrophic retroactive interference), generally arises: when a network having already learned a first set of items is then retrained on a second set of items, the newly learned information may completely destroy the information previously learned about the first set [2, 3]. Since this psychologically implausible behavior is unacceptable for models of human learning and memory, a number of authors have explored ways of reducing the retroactive interference in sequential learning tasks [2–15]. Solving this problem is rather difficult because the distributed character of represented information, essentially required within networks to achieve good generalization, seems incompatible with a weak interference level. In highly distributed systems, knowledge representations about different learned items largely share the same connection weights. When a new set of items is learned, the same connection weights, which were already adjusted for previously learned items, will be modified again. This may completely abolish memory of old information, giving rise to the classical 'stability–plasticity dilemma' [15].

In the present note, we propose a neural network model whose aim is to suppress catastrophic forgetting and dramatically reduce retroactive interference in gradient descent algorithms. The backpropagation learning procedure is used since it is extensively explored in the connectionist literature. However, our proposal also works as well for other gradient descent learning algorithms as the model is related to the distributed nature of representations and not to a specific learning algorithm.

Catastrophic forgetting in a sequential learning task

To illustrate catastrophic forgetting and demonstrate how this implausible failure can be suppressed, the same theoretical example will be taken throughout this study. This example is adapted from a sequential learning task classically used to highlight retroactive interference in connectionist systems [2]. A first training base (base *A*) is made up of 20 stimulus–response pairs, denoted (X_p, Y_p), with $p = 1, 2, \ldots 20$. A second training base (base *B*) is also a list of 20 stimulus–response pairs, denoted (X_p, Z_p), in which each pair *p* is composed of the same stimulus X_p as that of the corresponding *p*-pair in base *A*, but this time associated with a new desired response $Z_p \neq Y_p$. Items *X*, *Y* and *Z* are represented by distinct binary-valued pattern vectors with 32 components chosen at random (with values 0 or 1 being equally likely). The three-layer neural network model, shown in *figure 1*, has to sequentially learn first the set of *X*–*Y* associations (base *A*) and next the set of *X*–*Z* associations (base *B*). The network input layer is in fact composed of two subsets of units, the first representing stimuli X_p (32 input units with activation values of 0 or 1) and a second subset coding for an arbitrary contextual pattern, denoted *C*. This context pattern serves to indicate to the network (in learning and testing phases) whether processing is related to the *X*–*Y* or to the *X*–*Z* list. The context subset has five units: when the network is supposed to work on base *A*, the contextual pattern vector, chosen arbitrarily as $C = C_A = [10110]$, is sustained throughout processing associations of type $(X_p, C_A) \rightarrow Y_p$, and when processing is related to base *B* a second context pattern, $C = C_B = [10101]$, is maintained for referring to associations of type $(X_p, C_B) \rightarrow Z_p$. Each stimulus and context input unit is connected to all units of a hidden layer (50 hidden units), which are themselves connected to all of the 32 output units coding for response patterns Y_p or Z_p.

Learning within the network uses the standard backpropagation. Each presentation of an input pattern (stim-

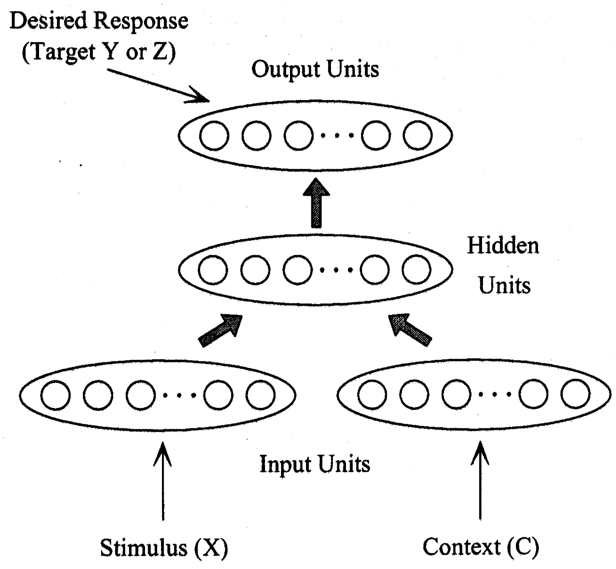

Figure 1. The classical network architecture used in the sequential learning task.

Each input unit is connected to all of the hidden units, which are themselves connected to each output unit (gray arrows).

ulus X and context C) gives rise to activation, which is propagated in the network, and next the connection weights are modified so that the output actually calculated by the network is as close as possible to the desired response pattern Y or Z. The learning rule governing weight modification tends to minimize a given error function, which is classically based on the sum of squared difference between the computed output unit activity and the corresponding elementary component of the desired target vector. However, there are many possible choices for the error function and throughout the present study the 'cross-entropy' function [16, 17] will be preferred to the quadratic function. In all the following simulations the classical logistic unit activation (with positive values and a bias) will be used and the usual learning rule parameters will take the following values: 0.01 for the learning rate and 0.5 for the momentum term.

The network, where all the connection weights are initially chosen randomly (between –0.5 and 0.5 uniformly), is first trained on base A associations ($[X_p, C_A] \rightarrow Y_p$), which are repeatedly presented at random (the contextual pattern C_A being maintained). Learning is considered as completed when the following criterion is reached: the absolute value of the difference between the activity of each output unit and the corresponding elementary component of the desired response pattern Y has to be less than or equal to 0.1 for all the base A associations. Once this learning criterion is reached on the first base, the base B associations ($[X_p, C_B] \rightarrow Z_p$) are subsequently learned using the same criterion.

During the test phases, the behavior of the output units is observed when an input pattern of type $[X_p, C_A]$ or $[X_p, C_B]$ is presented. In order to evaluate the network's ability to correctly reproduce the appropriate outputs (i.e., the targets) for a given set of inputs, a performance measure, called *goodness* [3], was adopted. Formally, if s_i denotes the activity generated over the output unit i and t_i the corresponding component of the desired response pattern, then the goodness, denoted g, of a single association being tested is defined as:

$$g = \frac{1}{L}\sum_{i=1}^{i=L}(2t_i - 1)(2s_i - 1)$$

where L is the number of output units. A goodness value of 1 indicates a perfect match between the calculated output and the desired response and a value of 0 indicates chance performance. The mean goodness G of a base of associations is simply the average of the individual goodness g related to the associations belonging to the base.

To show catastrophic forgetting, the network's ability to correctly reproduce the appropriate outputs for base A inputs (the 'old' base already learned) is estimated at different points in the course of learning the 'new' base B. Across all ten training cycles on the new base (one cycle corresponds to the exposure of all the 20 stimulus–response pairs of the base) the mean goodness G is calculated for the old and the new base (20 individual goodness values g averaged in each case). The results in *figure 2* clearly show a severe destruction of the old base A associations by the new ones being learned. This catastrophic retroactive interference is even more manifest when the network's performance is estimated according to a more behaviorally relevant, but less fine-grained, correctness measure. A response pattern generated by the network will be considered as correct if each output unit activity is 'on the right side' of 0.5, otherwise the response is considered as incorrect. That is, for any target component equal to 1, the corresponding output unit activity has to be greater than 0.5, and for any target component equal to 0 the related output unit activity has to be less than 0.5. This contrasted correctness measure is generally considered as suitable because it is easy to conceive a post-processing of the output layer activity (such as simple threshold units), which would give only the binary values 0 and 1 in response to the network's output activities respectively less than and greater than 0.5. Using this contrasted correctness measure, catastrophic forgetting is well-highlighted: tests performed on the old base give 0 % correct responses from cycle 8 up to the end of learning the new base, when within the same time the correct response rate for the new base being learned is 0 % from cycle 0 to cycle 22. It has to be noticed that catastrophic forgetting cannot be attributed to limited network resources since we intentionally took a high number of hidden units with respect to the number and size of item pairs to be processed.

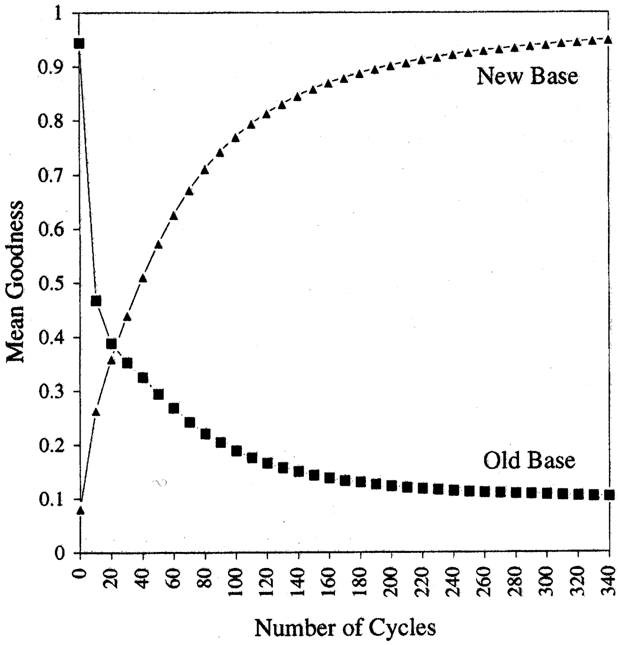

Figure 2. Catastrophic forgetting in a sequential learning task.

The lower graph shows a dramatic decrease in the mean goodness of a set of items previously learned by the network (old base) during training (all ten cycles) on a new set of items (new base). The increase in goodness for the new base in training is shown on the upper graph.

A dual neural network for learning that minimizes retroactive interference

Catastrophic interference can be eliminated by using a rehearsal mechanism: the old information previously learned by a network has to be continually refreshed (i.e., retrained) during the learning of new information. This trivial solution, requiring a permanent access to all events on which the network was trained during its history, is unacceptable when it is seen as the only solution for the human brain. Indeed, humans in general have the ability to learn new events without the complete abolition of memory for old events, which however do not occur again systematically for their consolidation. Nevertheless, this potential solution is useful since it leads to the interesting notion of *pseudo-rehearsal* recently proposed by Robins [12, 13]. Keeping the same example as above, the basic principle of the most efficient pseudo-rehearsal mechanism, called by the author *sweep pseudo-rehearsal*, is the following one. Once learning of base *A* is completed and before base *B* learning begins, the network is activated by a number of random input patterns, each giving rise to a corresponding output pattern. These input–output pairs are successively stored in a *pseudo-base,* which is then considered as having captured something reflecting the base *A* associative structure implicitly represented within the network. When training on base *B* occurs, the network is also concurrently trained on the input–output pairs previously stored in the pseudo-base: such a pair is seen as a pseudo-association, reflecting the old base, in which the output term serves as a pseudo desired target in learning. As usual, the real input–target pairs are chosen at random in base *B* and at the same time the pseudo-input–target pairs are also randomly chosen in the pseudo-base. In this mechanism, the determinant parameters are the size of the pseudo-base and the ratio between the numbers of real and pseudo-associations conjointly trained (in the cited papers, these parameters were usually set at 128 and 1/3, respectively). Learning the second set is considered to be complete when the learning criterion is reached for all the base *B* item pairs (the pseudo-item pairs not being subject to a learning criterion). If a third set of associations should be learned, then the same process would be applied again: before learning the new set, a pseudo-base has to be first built up, hence capturing some representation of the *A–B* structure, and then the new set is trained in conjunction with the *A–B* pseudo-information thus refreshed, and so on. The sweep pseudo-rehearsal mechanism was applied to several sequential tasks [12, 13] in the framework of standard backpropagation and the obtained results were rather encouraging since they show a significant decrease in catastrophic forgetting.

However, one question immediately arises with this hybrid algorithm where the current pseudo-base consists in fact of a set of distinct pattern vectors simply stored as a list of computer memory addresses: how the pseudo-base notion could be *neurally* implemented in the framework of a pure connectionist architecture. A second crucial question is concerned with how the deep structure of the distributed information represented in the connection weights of a neural system can be *optimally* captured.

The model

The connectionist architecture we propose to address these two questions is outlined in *figure 3*. Although our proposal would work on any sequential task, the same example as before will be used to present the model. In the figure, NET1 is a network that is trained on the external input–target patterns from bases *A* and *B*, and NET2 is a similar network but one that does not receive external activations. Consider now, for the sake of simplicity, an initial state of the whole architecture in which NET1 has already learned base *A* and NET2 is still 'empty' (i.e., with random connection weights). Assume that the neural system then enters in a first processing procedure, denoted process (I) in *figure 3*, in which NET1 cannot be receptive to external activations but is continuously receiving over its input layer a random activation from an internal noise generator. The successive NET1 inputs and induced outputs are both constantly sent to NET2, which is continually trained on these pseudo-associations, NET1 inputs

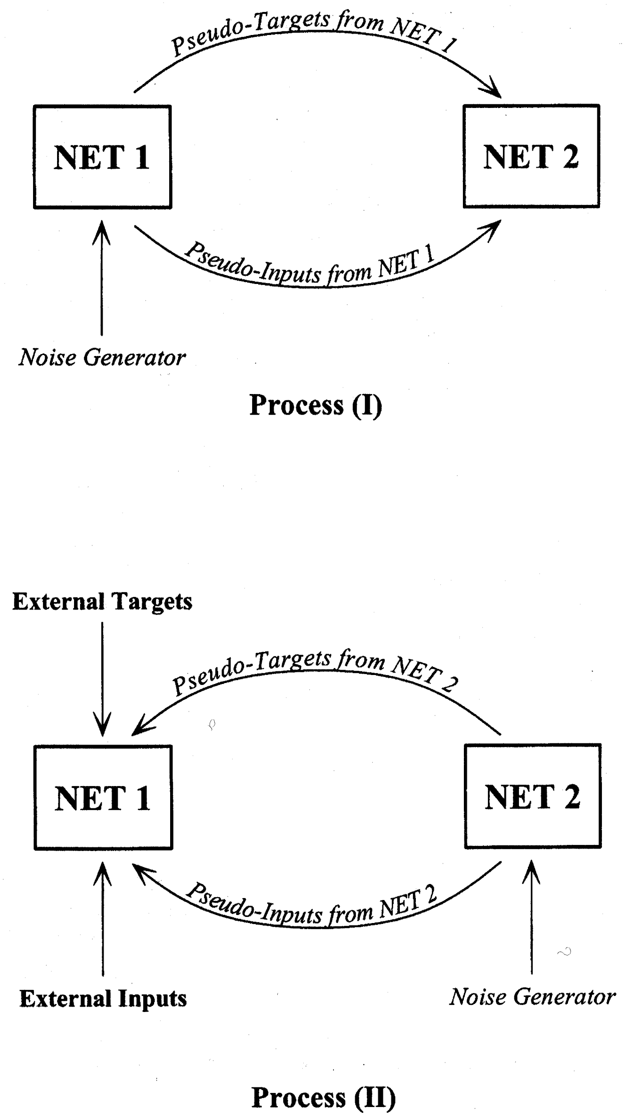

Figure 3. Functional diagram of the neural network architecture for learning new events without forgetting old ones.

Process (I)*: a noise generator continually stimulates the NET1 network in which information previously learned is continuously extracted and learned by a second network NET2.*

Process (II)*: when NET1 is presented with new external events, it is concurrently refreshed by internal information from NET2 reflecting the old events previously learned by NET1. Information capture from NET2 is performed as in process (I) solely on the basis of random stimulations.*

and outputs playing respectively the role of pseudo-inputs and pseudo-targets for NET2. Learning within NET2 stops when a second processing procedure, denoted process (II), occurs, in which NET1 becomes receptive to external activations and where NET2 is now continuously activated by the internal noise generator inducing a corresponding output activity. As in process (I), but this time in the opposite direction, NET1 is trained on the pseudo-associations originating continuously from NET2 while NET1 is also concurrently trained on external activations (i.e., the input–target pairs originating from base *B*). At this stage, it is expected that NET2 can play the role of the Robin's pseudo-base: during process (I), NET2 should capture some structure of the information previously stored within NET1 (base *A*) and then, during process (II), NET1, which is now learning new information (base *B*), should be concurrently refreshed on something reflecting the old information already stored (base *A*), thus avoiding catastrophic forgetting in this sequential *A–B* learning task. For several sequential tasks, as in the Robin's method, processes (I) and (II) should continue to work alternatively. The fundamental difference between the Robin's method and our proposal is that the pseudo-rehearsal mechanism can now be implemented according to a whole neural network system.

With respect to the second point above, we think that only a single pass of each random input pattern through a feedforward network is largely insufficient to correctly extract the information structure from its connection weights. To address this point we propose to use, for both NET1 and NET2, the network architecture shown in *figure 4*. In comparison to the classical network in *figure 1*, connections have been added from the hidden layer to the input layer: as previously, each hidden unit is connected to all output units, but is now also connected to all input units.

When one network is learning, the error function to minimize will now not only be based on the error between the calculated output and the output target but also on the error between the computed activation from

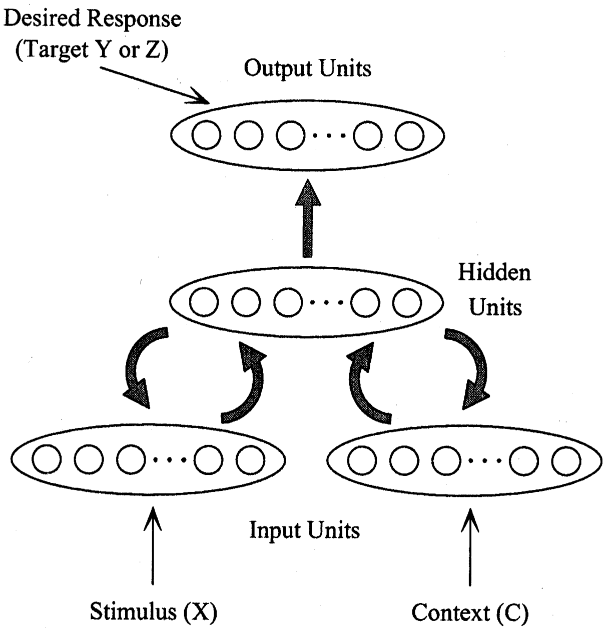

Figure 4. Architecture of a reverberating network.

With respect to the network in figure 1*, which only implements hetero-associations, the hidden layer is furthermore fully connected to the input layer, thus implementing also auto-associations.*

hidden units to input units and the current input pattern also playing the role of a desired target. In short, connections from hidden units to output units implement hetero-associations (external inputs–external targets or pseudo-inputs–pseudo-targets) and those from hidden to input units implement auto-associations (external inputs–external inputs or pseudo-inputs–pseudo-inputs). It should be noted that when NET1 or NET2 is in a learning situation only a single pass of activity determines weight modification. During this single step, the activity returned from the hidden units back to the input layer is never re-injected toward the hidden layer. In other words, learning is still feedforward with no recurrence.

When we consider the situation where one non-learning network is transferring information toward the other for training, the implementation of auto-association makes it possible to propose a new hypothesis on the way one network can generate pseudo-associations. Each time the random generator initializes the input layer of the non-learning network, the activity returned from its hidden units back to its input layer creates a new input unit activity, as an echo, which in turn is re-injected within the hidden layer, which in turn recreates a new echo over the input units, and so on. After a number of *re-injections* (which is a simulation parameter) during which activity flows back and forth between the hidden and the input layers of the network, the last echo generated on the input units and the last resulting output pattern are both sent to the other network for training. Thus, the basic functioning principle of the neural dual-process, implementing learning with pseudo-rehearsal, is essentially the same as that already described in *figure 3*. However, the crucial difference is that, for the network being currently emitting pseudo-associations toward the other, the activity generated in response to each random pattern has to first *reverberate* inside the network before transmission. It can be readily reasoned that the above re-injection process from a random seed, which is expected to converge close to network attractors, is much more suitable for capturing the deep structure implicitly contained within a distributed memory than a single feedforward pass of activity.

Simulations

The *A*–*B* sequential learning task was simulated in the framework of the proposed dual-process system with two reverberating networks. For all the following simulations, we always started from an initial state of the system where base *A* was already learned by the NET1 network (hetero-associations and auto-associations trained up to the learning criterion equal to 0.1) and where NET2 is still empty (random connection weights). In order to compare the simulation results with those classically obtained in the literature on feedforward networks, re-injections of input activity were not used during testing phases. Only output patterns resulting from one single pass of activity through the network were considered for calculating results, and data were only computed for hetero-associations.

Initially the mean goodness of base *A* calculated on the NET1 output units was equal to 0.948. When process (I) occurs, the noise generator activating NET1 was simulated by a sequence of binary-valued pattern vectors with a size corresponding to that of the input layer. Thus, these patterns have 37 components (with a value of 0 or 1) chosen at random uniformly (graded values would work as well). Each input from the random sequence reverberates within NET1 and after a number of re-injections, denoted *R*, the resulting input–output pattern is transmitted to NET2 as a pseudo-input–pseudo-target pair. Following this, the connection weights were modified using the same learning procedure and parameters as those in NET1. Within NET2, learning of the continual flow of pseudo-associations originating from NET1 proceeded until process (I) ended. In the simulations, process (I) was maintained until the mean goodness computed on NET2, with the real items already learned by NET1, was close to that calculated on NET1. In fact, in simulations with $R = 10$ re-injections, the mean goodness of base *A* in NET2 reached exactly the same value (0.948) as that in NET1, which means that information previously learned in the first network can be very well captured and transferred to the other.

During process (II), which occurs only in the presence of external stimulation, the continual generation of pseudo-associations by NET2 in response to the action of the random generator was exactly the same as that in NET1 during process (I). The external input–target pairs from base *B* were concurrently trained with the internal pseudo-input–pseudo-target pairs originating from NET2: for each base *B* association, *N* pseudo-associations from NET2 were conjointly trained. The upper graph in *figure 5* shows, with $R = 10$ re-injections and $N = 4$ pseudo-items pairs, the variation of the mean goodness related to the hetero-associations from base *A* (old base) tested on NET1 during learning of the new base *B* by the same network. For comparison, the lower graph shows the corresponding old base goodness when no pseudo-rehearsal is used, in fact for $N = 0$. Note that this curve, showing catastrophic forgetting, although related to an auto-hetero associative network, is quite similar to that in *figure 2*, which, however, relates to a simple hetero-associative network. Graphs in *figure 5* stop when the usual 0.1 learning criterion is reached on the new base *B*. As can easily be seen on the upper graph, the retroactive interference is dramatically reduced (catastrophic forgetting is completely suppressed) since the final mean goodness is close to its initial level. The initial descent of the old base goodness corresponds to a restructuring of the NET1 connection weights, which have to adapt to both the new base and the old structure. The number of cycles needed to learn the new base may appear relatively important. However, if the contrasted correctness measure is used, tests

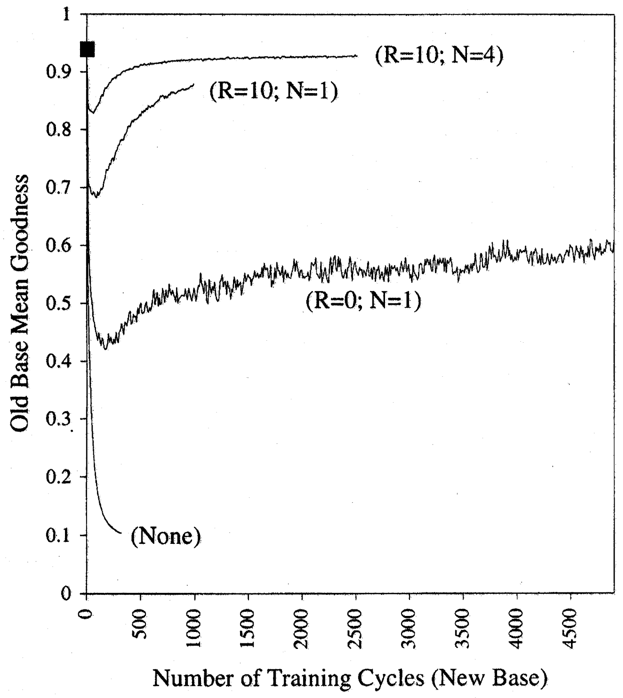

Figure 5. Mean goodness of the old base as a function of the number of training cycles of the new base.

Graphs (plotted for each set of ten cycles) stop when the new base learning is completed and the full square refers to the initial goodness of the old base before the new base training starts.

***Upper graph**: goodness variation with R = 10 re-injections of input activity in the two networks and N = 4 pseudo associations per one real external association concurrently trained in NET1. Note that retroactive interference is dramatically reduced since the final mean goodness of the old base is close to its initial level.*

***Middle graphs**: note the important enhancement of the old base goodness between simulations performed without (R = 0) and with (R = 10) reverberating networks, for the same N parameter value (N = 1).*

***Lower graph**: catastrophic forgetting with no refreshing process.*

performed on the new base give 100 % correct responses from cycle 210.

In order to show the effect of changing the *R* and *N* parameters, two other simulations were performed, the results also being represented in *figure 5*. The graph noted (R = 0; N = 1) represents the old base goodness when the two networks are not allowed to reverberate (i.e., zero re-injection) and where there is only one pseudo-input–pseudo-target pair originating from NET2 per one external input–target pair during the new base training in NET1. The main point to note is that the retroactive interference is relatively high. Using the contrasted correctness measure, the rate of correct responses for the old base at the end of learning the new base reaches only 5 % (i.e., one correct response among 20). A second point is that learning the new base proves rather difficult since the final number of cycles needed to reach the criterion is high. Moreover, other simulations with R = 0 showed that the new base cannot be learned when $N > 1$.

It is worth mentioning that any simulation performed with zero re-injection in the present model can be expected to give results similar to those obtained with the Robin's method. When we applied the Robin's procedure (results not shown), we had to take a stored pseudo-base of size 1 000 and one pseudo item pair per one real item pair (N = 1) to obtain a goodness variation almost identical to that noted (R = 0; N = 1) in *figure 5*. This performance is the best result possible when the Robins' procedure was applied with other pseudo-base sizes and other values for the *N* parameter. It is noticeable that this best result, obtained using the cross-entropy error function, has not been reached with the classical quadratic error function normally used in the Robins' method, in fact the new base could not be learned. More generally, it appeared from pilot simulations performed on our example of sequential learning task that the new base could not be learned when using the quadratic error function whatever the refreshing method was.

To highlight the effect of re-injections on system performance, a fourth simulation was run taking R = 10 and N = 1. As can be seen in *figure 5*, in comparing the case (R = 0; N = 1) with the case (R = 10; N = 1), where only re-injections are added, the old base goodness is in this last case much enhanced and learning of the new base is faster. This comparison demonstrates clearly the crucial role of neuron-like processes, reverberating from random stimulations, in discovering the deep structure of information distributively represented within network connectivity.

Conclusion

The reverberating process assumed to work within neural networks is at the root of the efficiency of the proposed pseudo-rehearsal mechanism minimizing retroactive interference. The need for reverberating networks, to greatly reduce forgetting of old hetero-associations when new ones are learned, highlights a basic principle postulating that hetero-associations have to be learned conjointly with auto-associations. This assumption, which simply means that any input stimulus is also learned in itself, seems likely from an ecological point of view. In the model the re-injection process, which operates only on auto-associative parts, permits the production of pseudo-items reflecting also the deep structure of hetero-associations. The usefulness of such re-injection processes in revealing the structure of learned information has already been shown [18] in the field of connectionist models of identification. It should be noted that the proposed reverberating process should not be confounded with recurrent time-delay networks [19–21] where no auto-associations are implemented and where recurrent activities are mainly used in order to implement time delays for learning and reproducing temporal sequences.

The dual-system approach, which is implemented here according to a pure connectionist architecture, goes in the same direction as a recent paper [11] where it is claimed, on the basis of a detailed study on the behavior of connectionist models and neurophysiological–psychological data, that two complementary learning systems are necessary (in fact, in the hippocampus and neocortex) for consolidation without catastrophic forgetting. However, the proposed processes were described in general terms but not in the framework of a neural network implementation.

Dual neural network processing, which in the present note is achieved mainly to suppress retroactive interference in sequential learning tasks, leads to an important corollary. Indeed, this type of processing provides an original but plausible means to 'copy and paste' a distributed memory from one place in the brain to another, this information transfer being achieved solely on the basis of random stimulations. For simplification, the two networks NET1 and NET2, between which information is transferred, were in fact considered as having the same structure. However, it is worth mentioning that this assumption is in no way required for the reliability of the information transfer. Two networks with different numbers of layers and different hidden layer sizes would work as well, which is essential for the generality of transport processes between neural structures. The only restriction may arise when one network does not have enough processing capacities to encode all the information contained within the other. In this case, the resulting system behavior could be psychologically relevant. Indeed, in distributed networks, two different resource capacities give rise to two different generalization abilities, which allow us to conceive of an architecture composed of one network being able to retain specific information while the other has a high ability to generalize.

Finally, two points concerning the plausibility of the model should be noted. The first concerns the flexibility of the model's main parameters, that is the number (R) of reinjections within networks and the number (N) of pseudo items needed during pseudo-rehearsal, which can be taken in large ranges without being harmful to the basic functioning of the system (with, of course, variations in performance). The second point concerns the speed of learning, which would seem rather slow. However, it was demonstrated [21] that learning could be made much faster in connectionist models when input–output layers were composed of sets of winner-take-all clusters (WTA clusters) coding for stimulus–responses pairs. In this case, a precise computation of output unit activities is not required during learning and testing, since it is sufficient that the expected output will be the higher within each WTA cluster, leading to a high learning speed.

References

1. Rumelhart D.E., Hinton G.E., Williams R.J. 1986. Learning internal representations by error propagation. In: *Parallel Distributed Processing: Explorations In The Microstructure Of Cognition* (Rumelhart D.E. et al., eds.), Vol I, MIT Press, Cambridge, MA, 318-362

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

3. Ratcliff R. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychol. Rev.* 97, 285-308

4. Kortge C.A. 1990. Episodic memory in connectionist networks. In: *Proceedings of the 12th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, Hillsdale, NJ, 764-771

5. French R.M. 1992. Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Sci.* 4, 365-377

6. French R.M. 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*. Lawrence Erlbaum, Hillsdale, NJ, 335-340

7. Murre J.M.J. 1992. The effects of pattern presentation on interference in backpropagation networks. In: *Proceedings of the 14th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, Hillsdale, NJ, 54-59

8. McRae K., Hetherington P.A. 1993. Catastrophic interference is eliminated in pretrained networks. In: *Proceedings of the 15th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, Hillsdale, NJ, 723-728

9. Lewandowsky S. 1994. On the relation between catastrophic interference and generalization in connectionist networks. *J. Biol. Systems* 2, 307-333

10. Lewandowsky S., Li S.C. 1995. Catastrophic interference in neural networks. Causes, solutions, and data. In: *New Perspectives On Interference And Inhibition In Cognition* (Dempster F.N., Brainerd C., eds.), Academic Press, New York, NY, 329-361

11. McClelland J.L., McNaughton B.L., O'Reilly R.C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the sucesses and failures of connectionist models of learning and memory. *Psychol. Rev.*102, 419-457

12. Robins A. 1995. Catastrophic forgetting, rehearsal and pseudo-rehearsal. *Connection Sci.* 7, 123-146

13. Robins A. 1996. Consolidation in neural networks and in the sleeping brain. *Connection Sci.* 8, 259-275

14. Sharkey N.E., Sharkey A.J.C. 1995. An analysis of catastrophic interference. *Connection Sci.* 7, 301-329

15. Grossberg S. 1987. Competitive learning: from interactive activation to adaptive resonance. *Cognitive Sci.* 11, 23-63

16. Hinton G.E. 1989. Connectionist learning procedures. *Artificial Intelligence* 40, 185-234

17. Plaut D.C., McClelland J.L., Seidenberg M.S., Patterson K. 1996. Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychol. Rev.* 103, 56-115

18. Wang S., Schreiber A.C., Rousset S. 1989. Connectionist modelling of a cognitive model of face identification. Simulation of context effects. In: *Proceeding of the 1st International Joint Conference on Neural Networks, Washington*. Vol II. San Diego, USA, IEEE & INNS, 549-556

19. Jordan, M.I. 1986. Serial order: A parallel distributed processing approach. Tech. Rep. ICS-8604, University of California at San Diego, La Jolla, CA

20. Elman, J. L. 1990. Finding structure in time. *Cognitive Sci.* 14, 179-211

21. Ans B., Coiton Y., Gilhodes J.C., Velay J.L. 1994. A neural network model for temporal sequence learning and motor programming. *Neural Networks* 7, 1461-1476