

# Modeling Grouping with Recursive Auto-Associative Memory

**Andreas Hansson** ([andreas.hansson@ida.his.se](mailto:andreas.hansson@ida.his.se))

Department of Computer Science, University of Skövde, Box 408  
541 28 Skövde, Sweden

**Lars F. Niklasson** ([lars.niklasson@ida.his.se](mailto:lars.niklasson@ida.his.se))

Department of Computer Science, University of Skövde, Box 408  
541 28 Skövde, Sweden

## Abstract

Sometimes humans have a need for storing long sequences of information in memory. Several experiments show that grouping the items in the sequence helps storing the sequence in auditory short-term memory. One architecture used by connectionist cognitive researchers when representing and processing sequences is Recursive Auto-Associative Memory. One of the aspects of it is that its capacity for storing sequences is limited, leading to that the longer the sequence the less likely it is that the entire sequence can be recalled; the deepest parts of the sequence are forgotten. Two experiments are performed to test if grouping affects storage in Recursive Auto-Associative Memories. We conclude that grouping affects the ability for storing sequences in Recursive Auto-Associative Memories much in the same way as it affects the human auditory short-term memory, i.e., using grouping increase the probability of that the sequence can be recalled correctly.

## Introduction

One technique for studying how memory is constructed is serial recall (see for example Baddeley, 1999 and Bridges and Jones, 1996; Pickering, Gathercole, Hall and Lloyd, 2001). A sequence of items (for example objects, digits, letters, etc.) is presented to the research subjects. The task for the subjects is to recall all items in the sequence in the same order as they were presented.

One of the aspects studied using serial recall tasks is how to increase the ability of the subjects to recall longer sequences.

Martin and Fernberger (1929) performed such a study regarding the improvement of auditory short-term memory. They concluded that the ability to remember sequences increased, if the objects in the sequence were organized in groups. Wickelgren (1964) and Pollack, Johnson and Knaff (1959) among others later confirmed this.

Grouping entails that the sequence is divided into shorter sub-sequences. The actual grouping has in previous experiments been done in two ways. Either the size of the groups is decided by the experiment leader (cf. Wickelgren (1964); Pollack, Johnson and Knaff (1959)) or the size of the groups is selected by the subjects and can vary within a sequence (cf. Martin and Fernberger (1929); Baumann and Trouvain (2001)).

Inspired by the findings of Martin and Fernberger (1929); Pollack, Johnson and Knaff (1959) and Wickelgren (1964), that grouping can increase the ability to recall longer sequences, the interest here is to test how grouping affect the recall of sequences in Pollack's (1990) Recursive Auto-Associative Networks (RAAMs). A RAAM is a type of artificial neural network used for representing sequences and structures of unknown or dynamical size. It has previously been used in experiments involving manipulation of structured objects (e.g. sentence transformation (Chalmers, 1990), language translation (Chrisman, 1991)). These experiments showed that RAAM uses the sequential order of the presented objects to develop highly structured (spatial) internal representations. In the experiments presented in the following a variant of these networks, called Extended Recursive Auto-associative Networks (ERAAM), originally suggested by Niklasson and Sharkey (1992), will be used.

## Grouping experiments

The exact definition of what a *group* is varies. According to Wickelgren (1964), Fraisse (1945) defined a group as a cluster of correct items separated by one or more errors. Wickelgren noted that this way of defining groups assumes that the items in a group are rarely forgotten and that the subjects practically never can remember two groups in succession. Others, for example Baumann and Trouvain (2001), used rhythm of speech and intonations for finding groups when the subjects recalled the sequences. Wickelgren defined a *grouping method* as a method for *rehearsal*, i.e. grouping in twos means rehearsing the items in twos; grouping in threes means rehearsing the items in threes; etc.

A large number experiments on how grouping affects the ability to recall longer sequences have been conducted. Most of these experiments confirm each other in that they show similar results.

Martin and Fernberger (1929) trained subjects to remember long sequences. This was achieved by the subjects using increasingly larger groups (first the subjects used groups of size two, after a while groups of size three and so on). Martin and Fernberger noted that the performance increased up to groups of size 5, after that the performance decreased.

Pollack, Johnson and Knaff (1959) confirmed these results using temporal grouping (i.e. making a short pause between the groups when presenting the sequence for the subjects) for items presented in 1's, 2's, 3's, 4's and 6's. According to their experiment, using groups of size four resulted in the best overall performance. Note however, that they did not test groups of size five, which Martin and Fernberger (1929) concluded was better than other group sizes.

Wickelgren (1964) also conducted experiments on how different groupings affected recall. According to that study groupings in threes were the best, closely followed by grouping in fours. The worst was grouping in twos, closely followed by grouping in fives. Groups of one item resulted in a performance quite close to the five-grouping, see figure 1.

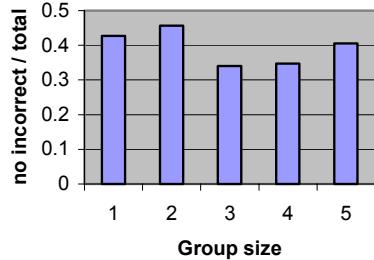


Figure 1: Results from Wickelgren's (1964) experiments for grouping in human short-term memory. Error rate is used to measure the subjects' performance.

In his experiments, Wickelgren used *error rate* to compare performance using different groupings. The error rate for a group is calculated by dividing the number of incorrectly recalled sequences (all items must be recalled in the correct order for the sequence to be regarded as correctly recalled) with the total number of sequences.

## ERAAM

As its name suggests, the Extended Recursive Auto-Associative Memory (ERAAM), originally suggested by Niklasson and Sharkey (1992), is an extension of Pollack's (1990) Recursive Auto-Associative Memory (RAAM), which is a connectionist architecture able to represent dynamically large structures (for example sequences) in a fixed-sized artificial neural network. Due to several of its features it has been used by several cognitive researchers who want to investigate theories for sequence recall (see for example Adamson and Damper (1999); Blank, Meeden and Marshall (1992)). One of the features is that the network is trainable, i.e. that it can learn to represent, for example, a specific type of sequence. Another feature is that RAAM during recall is more likely to produce decoding errors the more complex the sequence is. The errors usually appear towards the end of the sequence, quite like humans (cf. Henson (1996), Baddeley, (1999) and Adamson and Damper (1999)).

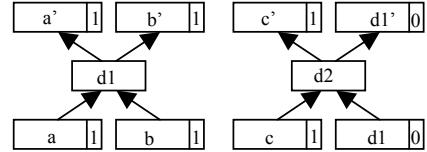


Figure 2: The structure of ERAAM. The extra bit in the input and output layers is used to classify if the representation is a terminal or a composite structure.  $d1$  is the compressed representation of  $a$  and  $b$ .  $d2$  is the compressed representation of  $c$  and  $d1$ . Due to decompressing errors we do not get *exactly* the same representation as we put in. To illustrate this,  $a'$  and  $b'$  are the representations from the decompressed  $d1$  and  $c'$  and  $d1'$  results from the decompressed  $d2$ . Note that the RAAM requires fixed valence on the structures represented.

In RAAMs two networks are involved: a compressor and a decompressor. The compressor consists of an input layer where an item in the sequence is presented together with a compressed representation of the previously presented items of the sequence and an output layer that contains the compressed representation of new item in combination with the previous (in figure 2 'd1' is a compressed representation of the combination of the terminals 'a' and 'b' and 'd2' is a compressed representation of the terminal 'c' and the non-terminal (i.e. previously compressed) 'd1'). The decompressor consists of an input layer containing the compressed representation and an output layer containing the (partially) decompressed representation. The compressor and decompressor networks respectively are used recursively to do further compressions or decompressions of sequences.

When decompressing, a terminal test is normally performed on the decompressed representations to see if any of them is a terminal and therefore should not be further decompressed. The terminals contain the representation of the items in the sequence. Different alternative methods for this terminal test have been proposed (see for example Chalmers (1990); Pollack (1990)). ERAAM is another suggestion for how to interpret if a decoded representation is a terminal or non-terminal. In ERAAM an extra bit of information has been added to each part of the sequence. If the representation is a terminal this bit is set to 1 and otherwise 0, see figure 2. This extra bit is compressed and decompressed along with the other information. When, during decoding, a 1 is encountered as the last bit in the representation, this is interpreted as the representation being a terminal. The representation is then compared to the representations for the known terminals and the terminal that is closest in Euclidean space is the one said to be present in the output.

## Experiment 1

The purpose of this experiment is to see how grouping affect the ability to recall longer sequences in ERAAMs. In the experiments Wickelgren's (1964) definition of grouping as a method for rehearsal is used, i.e. the ERAAMs rehearse the groups in the sequence as well as the complete sequence. The group sizes vary from 1 to 6, but the same group size is used throughout the sequence. According to Baddeley (1999) grouping can be used as long as the subjects can notice the presence of the groups. In order to mark the boundaries of groups a 'nil' character is used marking the beginning of a new group.

### Method

**Sequences** The sequences are entirely made up by digits between 0 and 9 and 'nil' representing spacing between groups. The digits and the 'nil' character have orthogonal representations so as not to give any benefit of grouping several representationally similar digits together. The sequences used are produced by picking a random digit for each position in the sequence. However, extra constraints are used. Each digit may only appear once during the first ten positions in the sequence, only once between the eleventh and twentieth position and only once between the twenty-first and thirtieth position, this to prevent that the probability for a specific digit to appear is larger than for the others thereby making the sequence a bit easier to learn. Furthermore, for groups of size two a specific digit may not occur in the same position *within* a group twice for the first twenty positions in the sequence. For groups of size three no digit may appear in the same position in a group during the first thirty items, and so on. This is done to ensure that the probability for a group to contain a specific digit in a specific position should be similar to the probability of other digits thereby giving as little advantage as possible to any digit combination in a group and as little advantage as possible to any specific group size.

Three sequences are generated following the above constraints:

- 97250831645982164073516,
- 58472096133509847261915,
- 25713680491368049257804

From each of these sequences twelve subsequences of are created varying in lengths from 12 to 23 digits. The subsequences of length 12 are constructed from the twelve first digits in the above sequences, the subsequences of length 13 are constructed from the thirteenth first digits and so on.

In order to separate the groups that the subsequences are divided into a 'nil' character was inserted marking the beginning of each group. This is done after the groups are constructed (see figure 3).

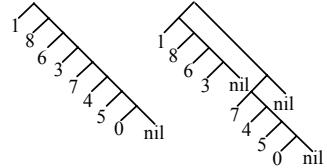


Figure 3: example of how the sequences are organized. To the left an ungrouped sequence, to the right a sequence with group size of four. Nil is used as an end marker.

**ERAAMs** 30 Extended Recursive Auto-associative memories are used having  $2 \times 11$  sigmoidal input and output nodes and 10 sigmoidal hidden nodes. Each ERAAM has a unique and random initialization so as to minimize the probability that the experiments, by coincidence, give bias to some specific group size.

**Procedure** Each subsequence is trained on all the thirty ERAAM networks. Each network is trained using backpropagation on the given subsequence for 200,000 iterations. This number of iterations is established by several dry runs, where almost no changes in the learning of the network are detected after 50,000 iterations and no changes are detected after 150,000 iterations. The limit is set to 200,000 iterations to give the ERAAMs plenty of time to learn the subsequences, but still provide an upper limit to the training time.

The ERAAMs receive the subsequence one item at a time, but each group in the subsequence is trained to be reproduced separately.

The ERAAMs are tested after 200,000 iterations on whether or not they can compress and decompress the entire subsequence they have been training on.

The ERAAM is considered successful in this experiment if the entire recalled subsequence is compressed and decompressed correctly, i.e. there must not be a single error anywhere in the decompressed subsequence.

### Results and Discussion

The proportion of errors in relation to number of trials in each group (i.e. the same measure as used by Wickelgren, 1964) can be seen in figure 4. For group size 1 the proportion of errors for sequences of length 12 to 23 is .748. For groups of size 2 it is .820. For groups of size 3 it is .768. For size 4 it is .745. For size 5 it is .729 and finally for groups of size 6 it is .769. It seems as if, analogous to Wickelgren's (1964) results, using a group size of 2 yields the worst result. We can also see that when using larger groups the error rate decreases until the use of group size 6. However, the performance is never much better than that achieved using an ungrouped sequence.

In the experiments the performance declines in a sigmoidal fashion from almost correct recall of ungrouped sequences of length 12 to none correctly recalled at sequence lengths of 20 or more.

The results acquired indicate that grouping *has* an effect on the ability to recall sequences in ERAAMs. It is better to

group a sequence into fours and fives than into twos with respect to the probability of recalling the sequence correctly. However, the ungrouped sequence is about as likely to be recalled correctly as using groups of size four and five. There are two possible reasons for this.

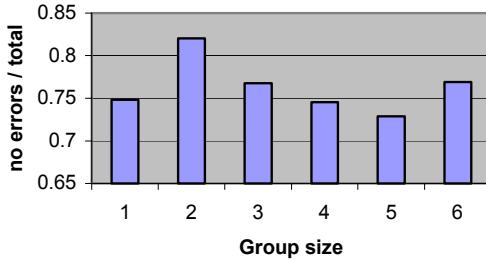


Figure 4: summary of results for experiment 1. As we can see, in almost 75% of the trials the ERAAMs fail to correctly encode and decode the entire sequence. We can also see that grouping for the most sizes makes this worse. However when using groups of sizes four and five there are fewer errors made than in the ungrouped sequences.

The first one is that grouping has no advantage in sequence recall compared to the ungrouped case.

The other reason is that in the grouped sequences there are extra characters inserted: the ‘nil’ characters. Since they are items as well as the digits this has the effect that the sequences get longer; more items need to be correctly recalled in the grouped sequences than in the ungrouped. This makes the task harder. If this is indeed the case, then it can be noted that the recall of sequences divided into groups of size four and five have the same performance as the ungrouped sequence, actually indicating that grouping has influence. To test whether this is the case, another experiment was performed where the ‘nil’ characters was removed.

## Experiment 2

Whereas the performance in experiment 1 never gets much better for a grouped sequence than for an ungrouped sequence, an interesting effect can be seen in that the use of group size 2 is the worst, but, as the size of the groups increase, the error rate decreases leaving at the group size of 5 a result about the same as an ungrouped sequence before getting worse again with groups of size 6. What differs between the grouped sequences, other than the group size, is the number of ‘nil’s used, being the most in groups of size two and then gradually decreasing with larger group sizes. Since ‘nil’ is an extra character, in practice, the sequence gets as much longer as the number of groups used than if it had been ungrouped.

To test whether or not the ‘nil’ character, since it makes the sequences longer, makes it more difficult for the ERAAMs to learn the sequences another round of

experiments are performed, this time without the ‘nil’ marking the boundaries of the groups.

## Method

**Sequences** The same subsequences as the ones used in experiment 1 are also used here. The difference between the subsequences in this experiment and the previous is the grouping method. In this experiment ‘nil’s are not used to mark group boundaries, see figure 5. This leads to that the subsequences contain the same number of items regardless of the group size used, whereas in the previous experiment the ‘nil’s constitute extra items making the subsequences longer.

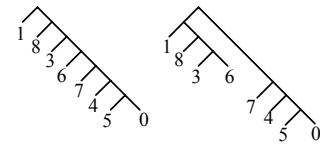


Figure 5: example of how the sequences are organized. To the left an ungrouped sequence, to the right a sequence with group size of four.

As in experiment 1 the size of the subsequences varies between 12 and 23.

**ERAAMs** The same ERAAMs as in experiment 1 are used; 30 differently initialized ERAAMs of size 2\*11 sigmoidal input and output nodes and 10 sigmoidal hidden nodes.

**Procedure** As in experiment 1 the ERAAMs are trained to compress and decompress the given sequence for 200,000 iterations using backpropagation. The ERAAMs are reset to their initial configuration between each sequence so that the training of one subsequence does not affect the next.

After training, the ERAAM is tested whether or not it can compress and decompress the entire subsequence. As before, the entire subsequence must be correctly reproduced for successful result, otherwise the ERAAM is considered to have failed in representing the subsequence.

## Results and Discussion

The error rates for different groupings can be seen in figure 6. For groups of size 1 the overall error rate (the number of correctly compressed and decompressed sequences divided by the total number of sequences used) is .577, for size 2 it is .482, for size 3 it is .402, for size 4 it is .358, for size 5 it is .398 and for groups of size 6 it is .370. It seems as if grouping the sequence leads to better performance, continuously improving until group size 4 when the performance cease to improve any further, see figure 6.

There is now a clear benefit shown of using grouping when trying to recall longer sequences. Using the group size of four yields a 41% better result (the difference between the number of incorrectly recalled sequences for an ungrouped sequence and the number of incorrectly recalled sequences using group size four divided by the number of

incorrect ungrouped sequences) than trying to represent an ungrouped sequence.

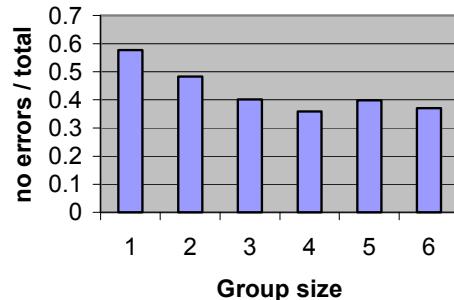


Figure 6: summary of results experiment 2. As we can see, not grouping the sequence here leads to that fewer sequences are recalled than in the grouped cases. We see that using groups of sizes four, five and six results in that more sequences are correctly recalled.

We can also see that this way of grouping the sequences results in that more sequences are correctly recalled than when using 'nil's as group separators.

## General Discussion

Sometimes there is a need to store dynamically long sequences in a memory. One of the techniques used by connectionist cognitive researchers, is to compress the sequences using recursive auto-associative memories (RAAMs). However, as in human memory, an aspect of the RAAM architecture is that the longer the stored sequence is the more probable it is that the deepest parts of the sequences cannot be recalled correctly, due to some compressing and decompressing errors. Since grouping a sequence increases the probability that it is correctly recalled in human memory we set out to test if grouping has a similar effect when recalling sequences in Extended Recursive Auto-Associative Memories (ERAAMs).

The experiments reported here show that grouping has a large impact on the probability that the sequences are correctly recalled. The probability that a sequence between the size of 12 and 23 is recalled correctly is on average 41% better using the group size of four instead of not using grouping at all.

The problem when recalling long sequences is that the deeper into the sequence, the more likely it is to miss-recall an item due to cumulating compressing and decompressing errors. We believe that grouping, dividing the sequence into smaller sub-sequences that are linked, works since it decreases the depth that needs to be recalled. The larger the groups the more the depth is decreased. After a while, however, the sub-sequences get so large that they also start to suffer from errors. This means that, as in humans, using increasingly large groupings the performance starts to decrease again.

In many situations there is a need to store long sequences. It is well known that grouping the sequence makes it easier to recall in auditory short-term memory. We see that this is also true for ERAAMs; grouping the sequences that are to be stored leads to an increased probability that they are correctly recalled.

What is suggested here is that RAAM like architectures can be used to model the human ability to store and recall sequences. However, many questions demand answers before networks of this kind can be said to model all the aspects of human memory. This includes questions concerning biological plausibility, a performance more closely matching the human, especially when using meaningful subsequences, etc. The results presented here show that RAAM like architectures indeed have a promising potential for supplying the answers to these questions.

## References

Adamson, M. J. & Damper, R. I. (1999). B-RAAM: A Connectionist Model which Develops Holistic Internal Representations of Symbolic Structures. *Connection Science*, 11(1), 41-71.

Baddeley, A. D. (1999). *Essentials of Human Memory*. Psychology Press Ltd.

Baumann, S. & Trouvain, J. (2001). On the Prosody of German Telephone Numbers. *Proceedings Eurospeech 2001 Scandinavia*, 557-560.

Blank, D.S., Meeden, L.A., and Marshall, J. (1992). Exploring the Symbolic/Subsymbolic Continuum: A case study of RAAM. In The Symbolic and Connectionist Paradigms: Closing the Gap.

Bodén, M. (1994), On Biased Learning for Generalisation, in *Proceedings of the International Conference on Neural Information Processing*, Seoul.

Bridges, A.M., & Jones, D.M. (1996). Word-dose in the disruption of serial recall by irrelevant speech. *Quarterly Journal of Experimental Psychology*, 49A, 919-939.

Chalmers, D. J. (1990). Syntactic transformations on distributed representations, *Connection Science*, 2, 53-62.

Chrisman, L. (1991). Learning Recursive Distributed Representations for Holistic Computation. *Connection Science* 3, 345-366.

Fraisse, P. (1945). L'influence de la vitesse de présentation et de la place des éléments. La nature du présent psychologique. *Année psychologique*, 45, 29-42.

Henson, R. N. A. (1996). Short-term memory for serial order. Unpublished doctoral thesis, University of Cambridge.

Martin, P. R. & Fernberger, S. W. (1929). Improvement in memory span. *American Journal of Psychology*, 41, 91-94.

Niklasson, L. F. (1999). Extended encoding/decoding of embedded structures using connectionist networks. In *Proceedings of the 9<sup>th</sup> International Conference on Artificial Neural Networks (ICANN99)*, IEE, 886-891.

Niklasson, L. F. & van Gelder, T. (1994). Can Connectionist Models Exhibit Non-Classical Structure

Sensitivity?, Proceedings of the Sixteenth Annual Conference of the Cognitive Society -94, Atlanta, Lawrence Erlbaum Associates, Hillsdale, NJ, 664-669.

Pollack, J. B. (1990). Recursive Distributed Representations. *Artificial Intelligence* 46(1), pp 77-105.

Pollack, I., Johnson, L. B. & Knaff, P. R. (1959). Running memory span. *Journal of Experimental Psychology*, 57, 137-146.

Pickering, S. J., Gathercole, S. E., Hall, M., & Lloyd, S. A. (2001). Development of memory for pattern and path: Further evidence for the fractionation of visual and spatial short-term memory. *Quarterly Journal of Experimental Psychology*, 54A, 397-420.

Wickelgren, W. A. (1964). Size of rehearsal group and short-term memory. *Journal of Experimental Psychology*, 68, 413-419.