ARTICLE IN PRESS

Available online at www.sciencedirect.com
ScienceDirect

COGNITION

Cognition xxx (2007) xxx–xxx

www.elsevier.com/locate/COGNIT

Learning non-local dependencies

Gustav Kuhn [a,*], Zoltán Dienes [b]

[a] *Department of Psychology, University of Durham, South Road, Durham DH1 3LE, UK*
[b] *Department of Psychology, University of Sussex, Brighton BN1 9QG, UK*

Received 28 June 2006; revised 20 December 2006; accepted 12 January 2007

Abstract

This paper addresses the nature of the temporary storage buffer used in implicit or statistical learning. Kuhn and Dienes [Kuhn, G., & Dienes, Z. (2005). Implicit learning of nonlocal musical rules: implicitly learning more than chunks. *Journal of Experimental Psychology-Learning Memory and Cognition*, *31*(6) 1417–1432] showed that people could implicitly learn a musical rule that was solely based on non-local dependencies. These results seriously challenge models of implicit learning that assume knowledge merely takes the form of linking adjacent elements (chunking). We compare two models that use a buffer to allow learning of long distance dependencies, the Simple Recurrent Network (SRN) and the memory buffer model. We argue that these models – as models of the mind – should not be evaluated simply by fitting them to human data but by determining the characteristic behaviour of each model. Simulations showed for the first time that the SRN *could* rapidly learn non-local dependencies. However, the *characteristic* performance of the memory buffer model rather than SRN more closely matched how people came to like different musical structures. We conclude that the SRN is more powerful than previous demonstrations have shown, but it's flexible learned buffer does not explain people's implicit learning (at least, the affective learning of musical structures) as well as fixed memory buffer models do.

Keywords: Implicit learning; Statistical learning; Artificial grammar learning; Chunks; Non-local dependencies; Simple Recurrent Network; Memory buffer model

* Corresponding author. Tel.: +44 1913343258.
E-mail address: Gustav.kuhn@durham.ac.uk (G. Kuhn).

0010-0277/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cognition.2007.01.003

1. Introduction

Implicit or statistical learning is an incidental learning process in which people become sensitive towards structures and regularities without needing to be aware of the knowledge acquired (Cleeremans, Destrebecqz, & Boyer, 1998). A basic question concerning how such knowledge is learned is whether the learning mechanism uses a temporary storage buffer, and, if so, what the nature of the buffer is. Though fundamental, this question has been scarcely addressed in an explicit way; its answer is intimately related to specifying what contents can be implicitly learned.

Implicit learning historically has been most vigorously investigated using the artificial grammar learning paradigm (Reber, 1989), in which participants are asked to memorize a set of letter strings that have all been generated using a finite state grammar. Following this memorization phase, participants are presented with a new set of test items, half of which obey the rule used to create the training items and the other half of which violate the rule in some way. Even though participants are usually unable to describe the rules used for their decisions, their classification performance is above chance. Similar statistical learning effects have been shown in many other paradigms (see Perruchet & Pacteau, 2006, for a review), but there remains controversy over what participants have learnt, and the computational mechanism responsible for this type of learning.

To date, most results from artificial grammar learning experiments using finite state grammars can be accounted for by postulating people learn chunks of adjacent elements (e.g. Dulany, Carlson, & Dewey, 1984; Perruchet & Pacteau, 1990). This acquisition of chunks is explicitly captured by chunking models of implicit learning (e.g. Knowlton & Squire, 1994, 1996; Perruchet & Vinter, 1998; Servan-Schreiber & Anderson, 1990) and is also predicted by connectionist models of implicit learning (e.g. Cleeremans, 1993). Although several different connectionist architectures have been proposed (cf. Dienes, 1992; Kinder, 2000b), the Simple Recurrent Network (SRN) (Altmann, 2002; Elman, 1990) has become one of the most popular, based both on empirical and theoretical grounds (Kinder, 2000a). We will first present the SRN network and then the buffer memory network, two networks that operate with contrasting types of storage buffers. Then we will introduce materials that cannot be learnt by simple chunking models as they require a buffer.

2. The Simple Recurrent Network

The SRN is a three-layered feed-forward network consisting of an extra set of units (context units) which is a copy of the hidden layer from the previous time step that then feed back into the hidden layer; thus, at time t the activation of the hidden units is influenced by both the input activation and the activation of the hidden units at time $t-1$ (see Fig. 1). During the training phase, the SRN is presented with each element of the sequence and is trained to predict the next element. During this training phase the weights are updated using backpropagation. Once the SRN is trained, it becomes sensitive to the transitional probabilities between the elements of the

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

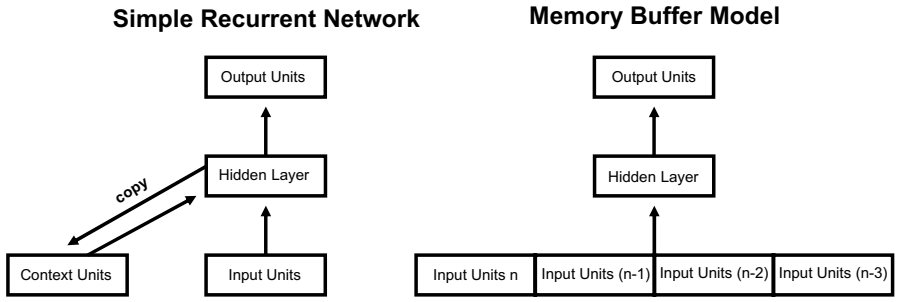


Fig. 1. Schematic diagram of the SRN and the memory buffer models.

sequence, which are often reliable predictors of an item's grammaticality. Each input at the first time step will lead to a different activation of the hidden units (Servan-Schreiber, Cleeremans, & McClelland, 1991). Due to the network's recurrent architecture, at the second time step the activation of the hidden units is then influenced by both the input and the activation of the previous hidden units. At the third time step the activation of the hidden units is influenced by both the activation of the first and the second input. This means that the network stores different representations for each input depending on the context, and therefore can become sensitive to higher order dependencies by storing contextual information over several time steps (Cleeremans & McClelland, 1991; Rodriguez, 2003; Servan-Schreiber et al., 1991).

Although chunking models and the SRN make different predictions about the exact nature in which the chunks are learnt (see Boucher & Dienes, 2003), both chunking models and the SRN are good at learning local dependencies. However, chunking models and the SRN have very different abilities with regards to learning non-local dependencies. Chunking models, as currently instantiated, simply link adjacent elements. The context units of the SRN, however, provide a buffer, in principle enabling the SRN to learn more powerful grammars (e.g. Christiansen & Chater, 1999). The context units allow the SRN to learn (fallibly) how far into the past it needs a memory in order to reduce error.

3. The memory buffer model

The SRN can be contrasted with a fixed memory buffer model, similar to the SRN in operating characteristics, learning rule, etc., except for how time is coded. The architecture of the memory buffer model is similar to the SRN except that it has no context units (see Fig. 1). Rather than storing information about the previous events in the recurrent context units, the input units of the memory buffer model not only encode the input presented at time t, but also at time $t-1$, $t-2$, and $t-3$. The size of the memory buffer is specified by the number of time steps that are encoded. Moreover, the number of time steps that have been encoded will determine definitively the length of the non-local dependency that can be learnt. The simplicity of this means of encoding time (i.e. unfolded in space) has often

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

recommended itself to researchers (e.g. Sejnowski & Rosenberg, 1987; aspects of coding in the Plaut, McClelland, Seidenberg, & Patterson, 1996, model). Cleeremans (1993) fit a memory buffer network, coding four time steps into the past, to the reaction times of people implicitly learning a serial reaction time (SRT) task. He found people became gradually sensitive in their reaction times to information contained up to four time steps into the past, and the memory buffer network could behave in a similar way.

Human learning in general requires a buffer. Aspects of language (Gomez, 2002; Onnis, Monaghan, Christiansen, & Chater, 2004) and music (Dienes & Longuet-Higgins, 2004) that can be learned in the lab rely on non-local dependencies, i.e. dependencies which take the form of two dependent items that are separated by a varying number of embedded items. Several studies have shown that under certain circumstances people can learn non-local dependencies that go beyond the learning of adjacent regularities (Gomez, 2002; Kuhn & Dienes, 2005; Newport & Aslin, 2004).

We will test the SRN and memory buffer models on the materials of Kuhn and Dienes (2005).

3.1. Materials: long distance dependencies in music

Kuhn and Dienes (2005) investigated whether people could implicitly learn a musical rule, a diatonic inversion, that was solely based on non-local dependencies. A diatonic inversion changes the direction of the diatonic intervals without changing the magnitude of the intervals. This inversion can be formed by numbering each of the notes in a tune from 1 to 8. The inversion is then formed by subtracting each pitch number from a constant, in this case 9. This diatonic inversion can be represented in terms of a non-local dependency, or biconditional grammar, in the sense that the first note is linked to the fifth note, the second note is liked to the sixth note, etc. For example if the first note is an F the fifth note will be a G (see Fig. 2).

The training and test material used by Kuhn and Dienes (2005) was designed to allow for the type of knowledge required to distinguish between the grammatical and the ungrammatical tunes to be manipulated. All the training tunes obeyed the diatonic inversion. For the test phase, different sets of test tunes were created for which alternative strategies could be used to distinguish between grammatical and ungram-

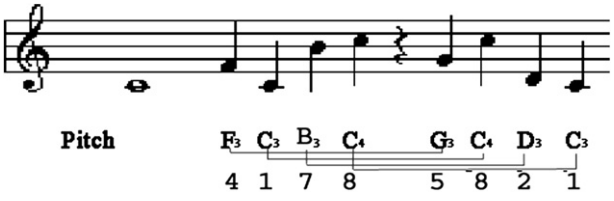


Fig. 2. Example of a grammatical training tune. The lines show the biconditional mapping between the different pitches; e.g. F_3 is linked to G_3.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

matical tunes.[1] In the *Exemplar* set, the grammatical items consisted of items that had occurred in the training set. Furthermore, the grammatical items had a higher ACS (associative chunk strength) than the ungrammatical items. In the *Fragment* set, all test items were novel. However, the grammatical items had a higher ACS than the ungrammatical items. This meant that for the *Exemplar* and the *Fragment* sets, correct classification could be based on either knowledge about chunks, and/or knowledge about the inversion rule, thus these two test sets will be referred to as ACS discriminating test sets. The material used in the *Abstract* set, on the other hand, was created from a set of bigrams that never occurred in the training set. This meant that correct discrimination could no longer be based on knowledge about chunks, and therefore relied on leaning of non-local dependencies, thus an ACS non-discriminating test set.

In the test phase, participants' were asked to give liking ratings to the test items. The results showed that participants' liking ratings for grammatical items increased over that given to ungrammatical items, as a result of having been exposed to the training items. Crucially, participants could distinguish between grammatical and ungrammatical items in their liking ratings regardless of whether grammaticality was associated with differences in ACS or not. Participants' successful discrimination occurred for test items that consisted entirely of bigrams that never occurred in the test phase, demonstrating that participants had acquired knowledge that went beyond the learning of adjacent elements.

In situations where all the test items are created from chunks that never occurred in the training material, which was the case in the *Abstract* set, existing chunking models cannot distinguish between grammatical and ungrammatical items (see Boucher & Dienes, 2003; Perruchet, Tyler, Galland, & Peereman, 2004; Redington & Chater, 1996 for discussion of relevant extensions to chunking models). Kuhn and Dienes (2005) showed that by using liking ratings, participants were able to discriminate between grammatical and ungrammatical tunes to the same extent in the absence as in the presence of chunking cues. Chunking models, as currently instantiated, can therefore be excluded as suitable models for participants' liking responses.

To learn long distance dependencies a buffer is needed, for example as in the SRN and fixed buffer models. While the fixed buffer model can self-evidently learn long distance dependencies by simple associative learning, the extent to which the SRN can is a more interesting question. Timmermans and Cleeremans (2000) carried out a series of simulations investigating whether the SRN could learn a biconditional grammar used by Shanks, Johnstone, and Staggs (1997). Shanks et al. designed material in which grammatical letter strings were generated using a biconditional grammar that determined the relationships between letters in position 1 and 5, 2 and 6, 3 and 7, and 4 and 8. For the test set, letter strings were produced which were balanced in terms of ACS. Grammatical and ungrammatical items therefore could not be

[1] Musical tunes can be represented in several different ways; pitches, chromatic intervals, diatonic intervals and contour. For simplification only the pitch representation will be used in this paper. However, it should be noted that all the above mentioned stimulus dimensions were taken into consideration when balancing the material in Kuhn and Dienes (2005).

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

distinguished from one another by relying on ACS, and required knowledge about the non-local mapping between letters. Timmermans and Cleeremans (2000) showed that after fairly extensive training (50–3000 epochs), the SRN learnt to distinguish between grammatical and ungrammatical letter-strings. Other studies have shown that the SRN is capable of learning non-local dependencies (Cleeremans & McClelland, 1991; Rodriguez, 2001, 2003; Servan-Schreiber et al., 1991). However, in all these studies the SRN was tested on material which contained *n*-grams that occurred in the training set. The material designed by Kuhn and Dienes is different in the sense that the items in the Abstract set were created from bigrams that never occurred in the training set, thus preventing the use of any local transitional probabilities. It is not clear that the SRN can learn non-local dependencies under these conditions. A further difference between the current study and these previous simulations lies in the amount of training involved. The learning of higher order dependencies requires extensive training, which is demonstrated by the use of very large training sets or large numbers of training epochs. For example, the training corpus used by Servan-Schreiber et al., 1991 utilised 70,000 training items. Furthermore, Rodriguez, 2003 showed that the learning of higher order dependencies improved with increasing corpus size[2] and increasing number of training items. In the human experiments presented by Kuhn and Dienes (2005), participants were exposed to the training material, 120 items, only once. If we want to accept the SRN as a suitable model of implicit learning, the SRN should learn the non-local associations in 1 epoch, which is less training than was used in these previous simulations.

The first aim of this paper was to investigate whether the SRN and the memory buffer model could learn the material used by Kuhn and Dienes (2005). As the non-local dependency learning is explicitly implemented in the memory buffer model it is fairly obvious that these types of models should have no problems in learning the non-local dependency. However, due to the reasons stated previously, it is less clear whether the SRN is capable of learning the biconditional grammar.

A common criticism of computational models is that if the parameters of a model that could in principle fit any human data are simply tweaked until a fit is obtained then nothing has been explained (Olsson, Wennerolm, & Lyxzen, 2004). This criticism is applicable to connectionist models, as these models rely on a range of free parameters, which have relatively large effects on the network's performance. Furthermore, as most of these parameters have no obvious psychological or physiological meaning it could be rather tempting to use post hoc justification to select parameters that give the best fit for a given set of human data. In implicit learning most researchers attempt to find a best set of parameters and try to fit simulation performance to a mean performance of human subjects, such as endorsement rates (e.g. Altmann & Dienes, 1999; Dienes, 1993; Dienes, Altmann, & Gao, 1999; Kinder, 2000b; Kinder & Assmann, 2000). The problem of multiple free parameters can be circumvented by running simulations on a large set of parameter values and then evaluating the models in terms of all of these parameters. Boucher and Dienes (2003) evaluated two different types of models by training them

[2] Corpus size varied from 10,000 letters to 100,000 letters.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

over a large region of parameter space. The *characteristic* behaviour of the models could then be determined and compared to human behaviour. This model selection process can also be viewed in maximum likelihood terms, i.e. one selects the model that provides the highest probability density for the behaviour that matches human behaviour (Bishop, 1996). The networks used in the present paper, were trained using a large set of parameter values, and then tested on the different test sets. These results led to a parameter space, rather than a single best fit value, which could be use to evaluate how well the model's characteristic performance matched that of the human participants. This approach allows the model to have explanatory power and goes beyond merely giving an existence proof that a model can fit a given set of data. The second aim of this paper was therefore to evaluate how characteristic the human data was of the memory buffer and the SRN.

4. Simulation study 1

The aims of the first simulations were to investigate whether the SRN and the memory buffer models could learn the material designed by Kuhn and Dienes (2005). The models were trained on the same material used by Kuhn and Dienes (2005), and tested on the two ACS-discrimination sets (*Exemplar* set and the *Fragment* set) and the ACS non-discriminating test set (*Abstract* set). In the *Exemplar* and the *Fragment* set, the grammaticality of the items was correlated with ACS. In the *Abstract* set, all of the items were created from a different set of bigrams, thus discrimination performance cannot be based on ACS. In simulation 1a the tunes were coded locally, whereby each element in the sequence was represented using one unique active unit. Similar to Boucher and Dienes (2003), simulations were run using a large range of different parameter values.

4.1. Method

4.1.1. Material

The grammar used was a diatonic inversion rule. All tunes consisted of 8 notes, which were selected from the C-major scale. These notes can be numbered from 1 to 8; $C_3 = 1$; $D_3 = 2$; $E_3 = 3$; $F_3 = 4$; $G_3 = 5$; $A_3 = 6$; $B_3 = 7$; $C_4 = 8$, where C_3 is middle C. The first four notes formed the prime and were selected semi-randomly, while the last four notes formed the inversion, which was created by subtracting the pitch number from a constant (9). The prime 3 6 4 3 leads to the following inversion 6 3 5 6, and the tune[3] 3643–6356.

One hundred and twenty different grammatical training tunes were constructed. These tunes were created from a unique set of bigrams, ensuring that a new set with different interval bigrams could be designed.

[3] It should be noted that the material was also balanced in terms of diatonic intervals, chromatic intervals, and contour.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

Three different sets of test tunes were created, which differed in the way they were associated to the training set. For the *Exemplar* set, 12 tunes were selected from the training set, which formed the grammatical tunes, and 12 ungrammatical tunes were created which violated the inversion. Furthermore, the ungrammatical items were created from bigrams that occurred rarely in the training set, thus leading to a lower ACS. Care was taken to ensure that grammaticality was not correlated with first order frequencies. For the *Fragment* set, 12 novel grammatical tunes were created with high ACS. The 12 ungrammatical tunes were created using the same rationale as in the *Exemplar* set, and therefore had a lower ACS than the grammatical items. For the *Abstract* set, both grammatical and ungrammatical tunes were created from a novel set of bigrams, which never occurred during the training phase. This meant that none of the tunes had any bigrams in common with the training set, thus leading to zero ACS. A full list of the material can be found on http:www.lifesci.sussex.ac.uk/home/Gustav_Kuhn/ch4/material_Kuhn_Dienes_ch4.htm.

The SRN had the same number of input units as output units and one hidden layer. During training, tunes were presented one element at a time by activating the input units, and the network was trained to predict the next element in the sequence, using backpropagation. Ten input/output units represented the 8 notes, the end and the beginning. The activation of the appropriate unit was then set to 0.9 and all other units were set to 0.1. A complete tune could be represented as a vector by concatenating the vectors specifying the unit activations for each successive note. The error between the target and the output activation was used to update the weights by using backpropagation, and after each completed sequence presentation, the context units were set to zero. For each network, the sequence of tunes was presented in a different random order.

The memory buffer model had 40 input units which could code four sequential elements, one hidden layer, and 10 output units. Test items were presented in the following way. The first element of the tune "1 2 3 4–8 7 6 5" was "* * * 1", and the network was trained to predict the second element, which was "2". On the second time step, the network was presented with the first and the second elements of the tune "* * 1 2 " and it was trained to predict the third element "3", etc. All other parameters were identical to the SRN.

The simulations were carried out using all possible permutations of the parameter values presented in Table 1. The selection of these parameters was based on existing artificial grammar learning simulations. A buffer of four time steps was used because the stimuli played to participants consisted of two perceptually separate melodies each consisting of four notes, naturally suggesting a theme of four notes and its reply.

Table 1
Range of parameter values used in the simulations

Network parameters	
Learning rate	0.1, 0.3, 0.5, 0.7, 0.9
Momentum	0.1, 0.3, 0.5, 0.7, 0.9
Number of hidden units	5, 10, 15, 30, 60, 120
Epochs	1

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

Moreover, with this size of buffer, the network should be able to learn the biconditional grammar. We consider this assumption of buffer size further in Section 8.

Epochs refer to the number of times the network cycles though the training set. In the human experiments, participants were exposed to the training set once. The number of epochs was therefore set to 1. The number of hidden units ranged from 5 to 120. Although 120 hidden units may seem like a very large number, several studies have shown that the learning of higher order dependencies does require rather large numbers of hidden units (Cleeremans & McClelland, 1991; Rodriguez, 2003; Servan-Schreiber et al., 1991). The learning rate is a constant that determines the size of the weight change, which ranged from 0.1 to 0.9 in steps of 0.1. Although 0.9 appears to be a large learning rate, this is commonly used value in artificial grammar learning simulations (e.g. Altmann & Dienes, 1999; Dienes, 1993; Dienes et al., 1999; Kinder, 2000b; Kinder & Assmann, 2000). Finally the momentum, a parameter that specifies the amount a weight keeps changing in the same direction over trials, ranged from 0.1 to 0.9, in steps of 0.1. Incorporating all of the above mentioned parameter combinations resulted in 150 different models. Each of these models was run 25 times, each time using a different set of random starting weights, using the Nguyen-Widrow method (1990), thus leading to 3750 simulations. All networks were simulated using the Matlab 6 Neuralnetwork toolbox.

In the test phase, networks were presented with the test items, and their ability to predict the next tone in the sequence was used as an index of performance. If the network has acquired knowledge about the training items it would be expected to perform better on the grammatical than on the ungrammatical items. Each network was tested independently on the material from the 3 test sets. Performance was assessed by calculating the cosine (COS) of the angle between the target vector **t** and the output vector **o** (e.g. Altmann & Dienes, 1999; Dienes, 1993; Dienes et al., 1999; Kinder, 2000b). A large COS implies a small angle, and thus a small distance between the two vectors. The COS can therefore be used as an index of how well the network performs at predicting the correct output sequence. The larger the COS, the better the performance. In order to compare the networks' discrimination performance to that of the human subjects, *z*-scores were calculated by subtracting the ungrammatical items' mean COS from the grammatical items' mean COS, and dividing this difference by the pooled standard deviation. A positive score represents an ability to discriminate between grammatical and ungrammatical items, and 0 is chance performance. During the test phase, training was left on; presumably, learning devices in the brain do not switch themselves off in test phases. In the human experiments participants' learning was assessed by comparing the experimental group's discrimination performance with that of an untrained control group. Similarly, the networks' learning was evaluated by comparing the trained networks' performance with that of a set of untrained networks (see Christiansen & Chater, 1999).

4.2. Results and discussion

Table 2 shows the mean *z*-scores for each network on each of the test sets. Paired sample *t*-test showed that all trained networks performed significantly better than the untrained networks on all three test sets (all $ps < .0005$). Moreover, both the SRN and

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

Table 2
Mean *z*-scores for trained and untrained networks, and the discrimination performance of the human subjects reported by Kuhn and Dienes (2005) for all three test sets

Test set	SRN				Buffer model				Human data			
	Trained		Untrained		Trained		Untrained		Experimental		Control	
	M	SE	M	SE	M	SE	M	SE	M	SE	M	SE
Exemplar	0.128	0.204	0.012	0.088	0.214	0.011	0.045	0.007	0.17	0.111	−0.03	0.092
Fragment	0.279	0.253	0.039	0.099	0.267	0.013	0.047	0.007	0.263	0.086	−0.09	0.096
Abstract	0.03	0.049	0.005	0.04	0.073	0.007	0.013	0.006	0.144	0.09	−0.13	0.099

the Buffer model performed significantly better on the ASC discriminating test sets *(Exemplar* and *Fragment)* than on the *Abstract* test, (all $ps < .0005$).

The relatively large standard errors demonstrate that the networks' performances were strongly affected by the particular parameter setting, which clearly illustrate one of the major problems with evaluating computational models that involve free parameters. In situations where the network's performance is affected by the parameter it would be rather tempting to select the parameter settings leading to the most desirable results. For example, one of the parameter settings resulted in positive discrimination performance on all three test sets, whilst several other parameter settings only led to correct discrimination performance on the *Exemplar* and the *Fragment* set, but not the *Abstract* test set. However, if the pattern of the models' characteristic behaviour is different from that of people, the model barely provides an explanation of peoples' behaviour, a good fit with a specific set of parameter values not withstanding. It therefore seems most important to indicate what the model's characteristic behaviour is and the degree to which peoples' behaviour is typical of this behaviour. It is possible that a computational model could figure in an explanation of human behaviour even if human behaviour was uncharacteristic of the model if uncharacteristic behaviour was produced by a set of parameter values for which a coherent story could be told. Clearly, in the case of human behaviour being uncharacteristic of a model, this coherent story bears a large burden in making the model explanatory. This problem will be addressed in simulations 3.

The fact that both the SRN and buffer model successfully discriminated between grammatical and ungrammatical items in the absence of chunking cues, suggests that they possibly learnt the biconditional mapping used in the inversion rule. The Buffer model was explicitly designed to learn non-local dependencies. It was therefore expected that this model would learn to discriminate between grammatical and ungrammatical items on the *Abstract* set. However, the fact that the SRN was able to make this discrimination was more surprising. It would be rather tempting to conclude that the SRN did in fact learn the biconditional grammar. However, it is possible that the network learnt certain idiosyncratic statistical regularities in the test and training material that were not intended by the experimenter, but could be used for the successful discrimination between grammatical and ungrammatical items. Before any conclusion about whether the network has acquired knowledge of the biconditional mapping can be reached, we must gain an insight into what the network has actually learnt, which will be the focus of the simulations presented next.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

5. Simulations study 2: non-local dependency learning by the SRN

One way of establishing what the network has learnt is by looking at the network's internal representations. Cleeremans (1993) trained an SRN on material generated using a finite state grammar. Once trained, the network was presented with a set of grammatical items, and the pattern of activation on the hidden units was recorded. The matrix of the Euclidian distance between these vectors was then used as the input for a cluster analysis. This analysis revealed that the activation patterns of the hidden units were grouped according to different nodes in the finite state grammar, thus suggesting that the SRN not only acquired knowledge about local dependencies, but acquired distributed knowledge of the finite state grammar itself.

In investigating hidden unit representations, the general strategy is to determine the proximity of the hidden unit activation patterns to each other for different states of affairs in the world. The functional proximity of the different activation patterns depends on how these patterns are used. We know that in the SRN the output units use the hidden unit activation by transforming the hidden unit activation with a sigmoid squashing function. Therefore the most natural way of learning about the proximity of different hidden unit activation patterns to each other is not to use Euclidean distance but to determine the closeness of the output they produce after the sigmoid squashing. Thus the approach presented here evaluated the relation between states of affairs in the world and output unit activation as a way of determining whether the hidden units had learnt to encode the states of affairs. If the SRN has learnt the non-local mapping, then once the network has been presented with a particular input element, it should be able to correctly predict the associated item in the correct position. For example, if the network is presented with an F in the first position, it should be able to predict a G in the fifth position, as this was one of the mappings the network was trained to learn, regardless of what the intervening material is. If the network fails to predict, whatever the intervening material, we can conclude that the network used other regularities in the training material that could be used to successfully discriminate between grammatical and ungrammatical items. In the simulations presented here, a new set of test material was designed that enabled us to determine whether the SRN was able to predict the correct non-local mappings independently of the separating elements. In order to select the best parameter settings, discrimination performance was averaged across the three test sets, and the parameter values of the network with the highest overall performance were used for the subsequent simulations.

In the Buffer Network, the non-local dependency involves learning simple associations between the item at t_0 and t_{-4}, and we can assume that these associations are learnt easily: the relevant information about the item at t_{-4} is just as readily available as the item at t_{-1} in predicting the item at t_0. Local dependencies and dependencies back to t_{-4} are equivalent for the network. Thus, we did not run any formal analysis on the type of representations that were learnt for the Buffer Network.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

5.1. Method

The SRN containing 120 hidden units, a learning rate of 0.3, and a momentum of 0.3 led to the best overall discrimination performance on all three test sets. These parameter values were therefore used for the subsequent simulations. The training and test procedures were identical to the previous simulations. The models were run 25 times by initiating a different set of random weights each time, using the Nguyen-Widrow method (1990).

A set of test items was designed with the aim of establishing whether the network learnt each of the 8 biconditional mappings in each of the 4 positions. Sequences were created that started with a specified input element, followed by 3 random elements. If the network has learnt the biconditional mapping, and is presented with a specific input element in the first position, it should be able to predict the correct target element in position 5. Similarly, if the network is presented with a specific input element in position 2, it should be able to correctly predict the corresponding element in position 6, independent of the separating elements and so on. The numbers 1–8, which corresponded to the 8 different pitches, were used as specified input numbers, and random numbers between 1 and 8 were used as the embedded elements. Table 3 shows a schematic representation of the material used.

The networks' performance was then measured by calculating the Luce ratio (Luce, 1963) for the target output unit, and the Luce ratios for all the non-target output units. The Luce ratio is calculated by dividing the activation of the target output unit by the sum of the activation of all other output units. If the network has learnt the biconditional mapping we would expect a higher Luce ratio for the target unit, than the non-target units. A total of 8000 test items were created whereby each of the 8 pitch numbers occurred 1000 times as an input. This procedure was repeated for each of the 4 positions. In the same way as the previous simulations, each sequence was preceded by the beginning marker. These simulations were carried out using the 25 trained networks.

Table 3
Schematic diagram of the test items used for each of the 4 biconditional mappings

Biconditional mapping	Element number								
	1	2	3	4	5	6	7	8	9
1–5	Beginning marker	**N**	R	R	R	**P**	R	R	R
2–6	Beginning marker	R	**N**	R	R	R	**P**	R	R
3–7	Beginning marker	R	R	**N**	R	R	R	**P**	R
4–8	Beginning marker	R	R	R	**N**	R	R	R	**P**
	N = {1,2,3,4,5,6,7,8}								
	R = random number {1,2,3,4,5,6,7,8}								
	P = predicted element								

Each sequence contained 9 elements, whereby the first element was always the beginning marker. *N*, input number which took the values 1, 2, 3, 4, 5, 6, 7, and 8. *R*, random numbers between 1 and 8. *P*, predictor element. The bold items highlight the biconditional mapping.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

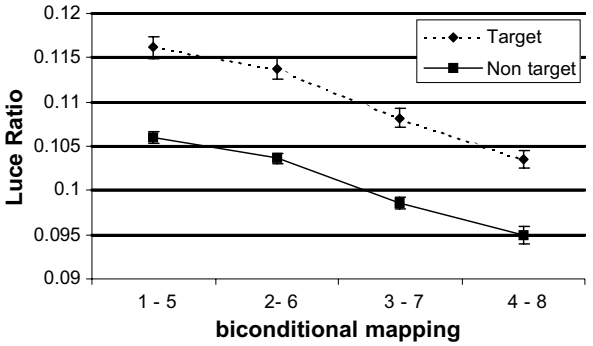


Fig. 3. Mean Luce ratio averaged across specified input and network, for target and non-target elements. Error bars represent standard errors.

5.2. Results and discussion

Fig. 3 shows the mean Luce ratios averaged across networks and specified inputs, for target and non-target elements, for each of the 4 biconditional mappings. *t*-Tests showed that mean Luce ratios of the target elements were significantly greater than those of the non-target items, all $ps < .0005$. That is, for each position the network predicted the correct pitch more strongly than the other pitch. This leaves open the question of whether it predicted this pitch more strongly for the appropriate position rather than the other positions. This question is now addressed.

The next simulations aimed to assess whether the network was sensitive to the positional information inherent in the bi-conditional rule, by looking at whether the network predicts the corresponding note better in its associated position than any others position.[4] For example, an F3 in the first position should predict a G3 in the fifth position. However, it is possible that the network predicts a G3, but without the specific location of the inversion being predicted. In order to investigate whether the network learnt the positional information we calculated the Luce Ratio for the predicted element in its correct location, and compared this with the average Luce Ratios for the same element in the remaining three positions.

The 25 trained networks from the previous simulations were presented with 1000 randomly generated inversion sequences. For each of these sequences we calculated the Luce Ratio for the predicted element in its predicted location and in the Luce Ratios for the three non-predicted locations. From Fig. 4 it can be seen that in locations fife and six the Luce Ratios for the correct locations were significantly higher than for any of the non-predicting locations (all $ps < .0001$). For the location seven, the Luce Ratio for the correct location was significantly higher than the adjacent

[4] We thank Pierre Perrucht for the suggestion of this simulation. We also ran an additional simulation suggested by Perruchet similar to the one in Fig. 4 except that in calculating the Luce Ratio, the unit coding the same note as occurred 4 time steps back was not used. The results were the same as in Fig. 4; i.e. the network is doing more than learning simply not to predict the same element again.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

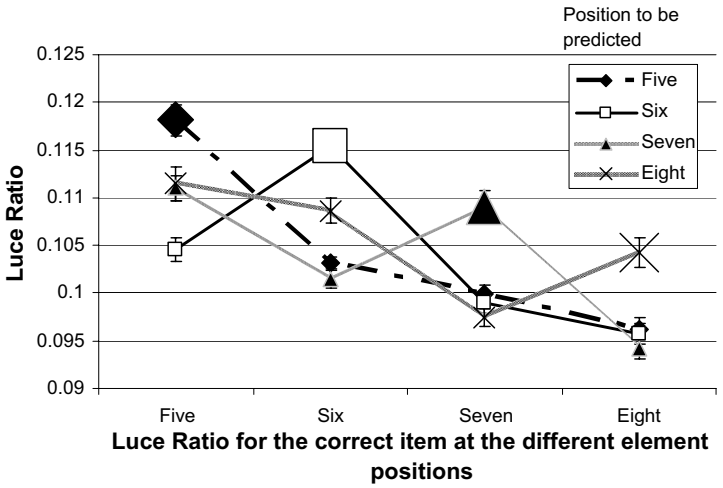


Fig. 4. Mean Luce Ratios for correctly predicting the element at each of the five time steps for each of the to be predicted links. The large data points represent the correct location. Error bars represent standard errors.

incorrect locations, and significantly higher than the average of incorrect locations, but no higher than the element in position five. These results imply that for these positions, the network successfully learnt the positional information. For the element in location eight, the Luce Ratio of the correct location was no higher than the average of the other locations. However, the Luce Ratio for the correct location was higher than the preceding incorrect location, suggesting that the network may have learnt at least some positional information for this location. These results demonstrate that the network was able to predict the correct position, but that this ability decreases further along the sequence. After each time step the Network stores more and more information in its context units. It is therefore likely that this additional information made the task more difficult, which may explain this drop in performance further along the sequence.

The results show that if the SRN is presented with a specific input element followed by four random elements, it produced a significantly higher activation level for the output nodes that were associated with that particular biconditional mapping, than for the remaining nodes. Furthermore, we showed that for positions five to seven the activation level of the output nodes associated with a particular biconditional mapping was higher in the correct location than in the incorrect locations. These results demonstrate that at least for the first three biconditional mappings, the network was able to correctly predict the location of the corresponding element. This method offers a direct way to gain an insight into what the network has actually learnt, and provides strong evidence to suggest that the SRN did in fact learn the biconditional mapping between the pitches in the different positions, and at least to some extent, the positional information that governs the biconditional rule.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

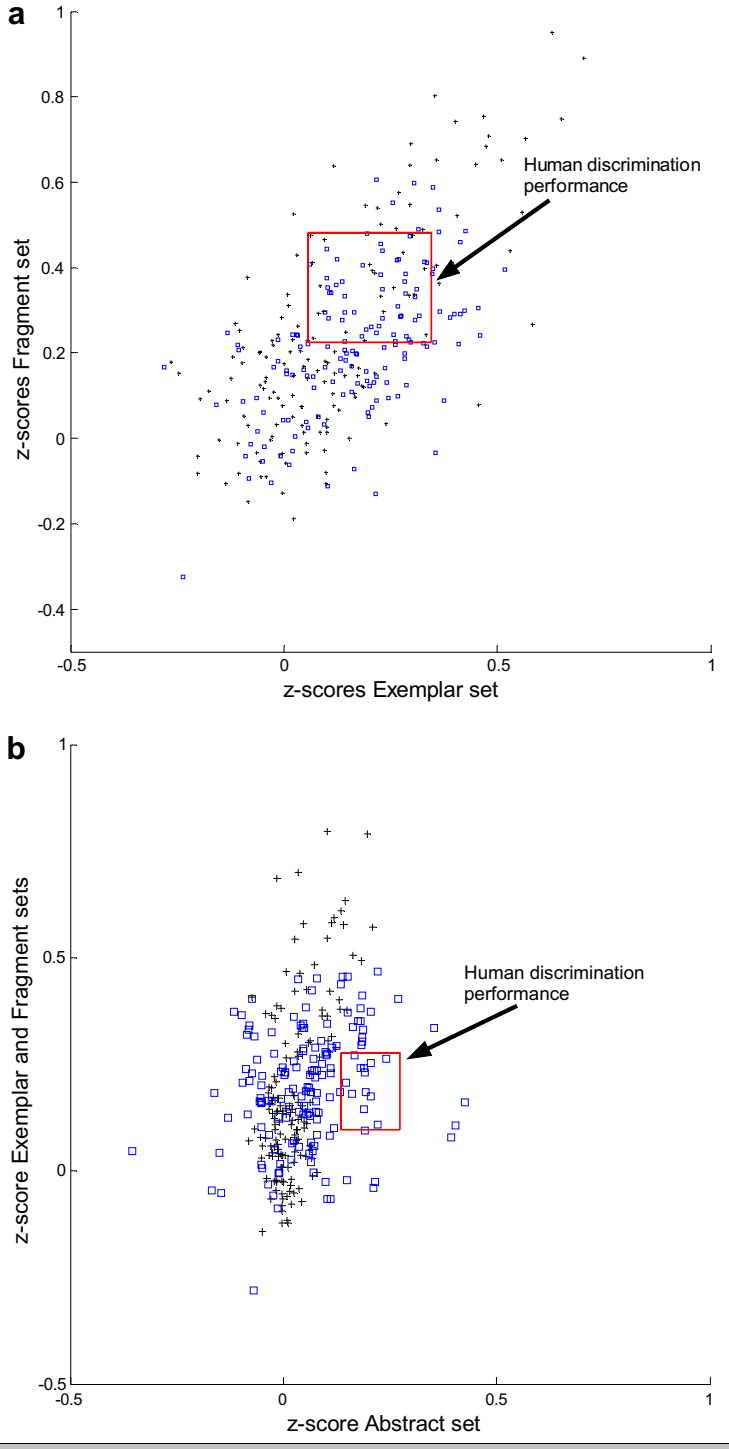
6. Simulations 3a: characteristic behaviour of the networks

The aim of the analyses presented here was to evaluate how the models' typical performance compared to the human data. Table 2 also shows participants' discrimination performance. The results so far suggest that both models learned the material where grammatical items had higher ACS than ungrammatical items more easily than the material where grammatical and ungrammatical items contained novel bigrams. In the analysis reported next we evaluate how typical the performance of the networks are to that compared to the human data, by looking at the relationship between discrimination performance based on chunks and discrimination performance based on non-local associations more closely. The effects of learning were calculated for each network by subtracting the *z*-scores of the untrained networks from the *z*-scores of the trained networks. The differences in *z*-scores on the *Exemplar* and the *Fragment* set were then plotted against each other (Fig. 5a). The area covered by the human subjects (mean ± 1SE) was also plotted in the same graph, which was calculated by subtracting the mean *z*-scores of the control group from that of the experimental group. Fig. 5a shows a positive relationship between the discrimination performance on the *Exemplar* and the *Fragment* set for both types of models. Moreover, the models' characteristic discrimination performance appears to match the human data, which is demonstrated by the fact that a substantial number of the SRNs (21/150) and the memory buffer models (47/150) fell within the area covered by the human data. In order to look at the networks' sensitivity towards chunks we plotted the discrimination performance on material where grammaticality was correlated with ACS (average of the *Exemplar* and *Fragment* set) against the discrimination performance where grammaticality was independent of chunks (*Abstract* set). Although the data from the two models does still overlap, the distributions of the two models look somewhat different. For the SRN there was a very steep slope ($r^2 = 0.51$, Beta = 0.56) which demonstrates that the SRN was particularly sensitive to chunks. For the memory buffer model on the other hand, the data was less correlated ($r^2 = 0.28$, Beta = 0.53), illustrating that the model was less affected by chunks. With regards to the human performance, the graph shows no overlap in performance between the SRN and human's discrimination performance, as none of the models fell within the square defined by human data. However, several of the memory buffer models fell within the human data (12/150), which was significantly more than the SRN ($\chi^2 = 11.76$, $p < .0005$), thus suggesting that it was more characteristic of the human data.

7. Simulation 3b

In the previous simulation the tunes were coded locally and therefore independent of the neighbourhood relations between the notes. Although this type of coding is useful for most computational simulations, it bares limited resemblance to the way in which the human participants represented the material. Participants would have experienced two notes that are located closely to each other on the scale as being

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003



Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

more similar than those separated by several notes. In simulations 3b we incorporated these neighbourhood relationships by using a distributed coarse coding rather than local coding. For example, a C3 was coded with units 1, 2, and 3 being active; a D3 with units 2, 3, and 4 being active, an E3 with units 3, 4, and 5, and so on. Each note shared an active unit with its two neighbouring notes of the diatonic scale. This meant that two notes that were in close proximity on the scale also shared common activation. In all other respects the simulations were identical to the previous simulations.

7.1. Results and discussion

The results of these simulations were very similar to the ones using the simple coding. On all three test sets, the trained networks performed significantly better than the untrained networks, and they performed significantly better on the ACS discriminating test sets than on the Abstract set (all $ps < .0005$). Similar to the previous simulations we evaluated the models' characteristic behaviour, with regards to chunks and non-local dependencies. When the discrimination performances of the *Exemplar* and the *Fragment* sets were plotted against each other, the SRN (17/150) and the memory buffer (18/150) model captured the human data equally well ($\chi^2 = 0.32$, $p = .86$). However, when the discrimination performance on the ACS discriminating test sets were plotted against the discrimination performance on the *Abstract* set, the human data captured significantly more memory buffer models (15/150) than SRNs (4/150), $\chi^2 = 6.80$, $p = .009$. Similar to the previous simulations, the characteristic behaviour of the memory buffer model was more typical of the human behaviour, than that of the SRN.

Finally, we ran simulations in which the melodies were encoded at input and output by their neighbourhood relations in the circle of fifths using coarse coding. Longuet-Higgins (1987) argued that in hearing music tonally, a salient dimension is the number of fifths apart two notes are. The participants in Kuhn and Dienes (2005) may well have coded stimuli along this dimension if they heard the melodies in a musical way. Our coarse coding activated units (say) 1, 2, and 3 to code C, activated units 2, 3, and 4 for G (which is a fifth higher), units 3, 4, and 5 for D (a fifth higher again) and so on. Virtually identical results were obtained as for localist pitch coding

Fig. 5. (a) Scatter plot in which the difference between the trained and the untrained networks' z-scores on the Fragment set was plotted against the z-scores from the Exemplar set. The + signs represent the SRN's performance, and the squares represent the performance of the memory buffer model. The plot also shows the areas covered by the human discrimination performance, which was calculated by subtracting the control group's mean z-scores from the Experimental group's mean score and plotting these mean differences ± 1 standard error for each test sets on the corresponding axis. (b) Scatter plot in which the difference between the trained and the untrained networks' z-scores on the Abstract set was plotted against the average z-scores of the Fragment and the Exemplar sets. The + signs represent the SRN's performance, and the squares represent the performance of the memory buffer model. The plot also shows the areas covered by the human discrimination performance, which was calculated by subtracting the control group's mean z-scores from the Experimental group's mean score and plotting these mean differences ± 1 standard error for each test sets on the corresponding axis.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

and the coarse coding of pitch: Training led to significant improvement of models' discrimination on all tests sets, but significantly and substantially better performance with the ASC discriminating tests sets than the abstract set (all $ps < .0005$). When the discrimination performance of the *Fragment* set was plotted against the discrimination performance of the *Exemplar* set, the human data captured significantly more memory buffer models (38/150) than SRN models (18/150) $\chi^2 = 8.78$, $p = .003$. Furthermore, when the discrimination performance of the ACS discriminating test sets were plotted against the Abstract set, the human data captured significantly more of the memory buffer models (7/150) compared to the SRN models (1/150) $\chi^2 = 4.62$, $p = .032$.

8. General discussion

The simulations presented in this paper used rather novel ways of exploring the types of rules a SRN and a memory buffer model could learn, and the extent to which its characteristic performance matched that of human participants. Computational models, whether connectionist or symbolic, with sufficient free parameters to replicate a large repertoire of behaviours explain nothing by fitting one particular pattern of behaviour. The problem of multiple parameters was circumvented by training the networks on a wide range of parameter settings, and then evaluating the model across this parameter space. The effect of learning was shown by comparing the discrimination performance of the trained networks to a set of untrained networks. This method has several advantages over other methods that simply fit models to data. The idea is to gain an accurate picture of the network's characteristic behaviour. This approach revealed an interesting point that would be ignored using conventional model fitting; namely that when grammaticality was independent of ACS there was only a small subset of models that learnt to discriminate between the grammatical and ungrammatical items.

The results from the previous simulations provided evidence to suggest that the SRN and the memory buffer models could discriminate between grammatical and ungrammatical tunes in which grammaticality was defined by a non-local mapping, and independent of bigrams. The memory buffer model was explicitly designed to learn non-local dependencies. It's architecture meant it could store information from up to four time steps, which allowed it to learn non-local dependencies up to $n = 4$. The fact that the SRN learnt the biconditonal grammar was more interesting. However, the fact that the SRN could discriminate between grammatical and ungrammatical items in the absence of chunks does not yet prove that it has learnt the biconditional grammar. It is possible that the SRN picked up on quirky statistical regularities present in the training and the test set, which could be used to successfully discriminate between grammatical and ungrammatical items. The aim of the final simulations was to establish whether the SRN had in fact learnt a biconditional mapping. This was achieved by demonstrating that the SRN could correctly predict the target elements as defined by the biconditional grammar, independent of the embedded elements. In a similar study, Timmermans and Cleeremans (2000)

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

investigated whether an SRN could learn the biconditional rule used by Shanks et al. (1997). Although Timmermans and Cleeremans showed that the SRN did learn to discriminate between novel grammatical and ungrammatical items generated by the biconditional grammar, these results were possibly due to irregularities in the training and test material, rather than the network having learnt the biconditional mapping. The results presented here on the other hand showed that the SRN could correctly predict the items linked to the biconditional grammar, independently of randomly constructed embedded elements over a large number of simulations, thus providing strong evidence that the SRN can in fact learn the biconditional rule. Although several studies have shown that the SRN is capable of learning non-local dependencies, these have all been based on looking at the extent to which the SRN could learn *n*-grams (e.g. Cleeremans, 1993; Rodriguez, 2003). The results presented in this paper offer more direct evidence showing that the SRN had in fact learnt the non-local dependency. Furthermore, the networks in these previous studies required extensive training, which seems to be rather untypical of the procedure employed in implicit learning paradigms, where participants are usually exposed to the same training items once or twice. The simulations presented here have demonstrated that the SRN could learn the non-local associations using only 1 epoch and 120 training items.

The suitability of each model as a model of participants' discrimination performance was evaluated by comparing the behavioural data with the models' performance across the whole range of parameters. This analysis can be regarded as motivated by likelihood or Bayesian considerations. In likelihood and Bayesian inference, simply finding a good fit model is not sufficient to evaluate the model. In Bayesian inference, one also has to determine how probable the data is, given the model, and the prior probability distribution of the parameter values in order to finally calculate how much one's prior probability of the model being true could be increased. In finding roughly how the characteristic behaviour of the model matches what people do, we have in one sense done a poor man's Bayes. Rather than seeing a poor man's Bayes as a weakness of our approach, we see it as strength. One need not buy into Bayesian approach in toto in order to appreciate the weaker claim that the model's characteristic behaviour should be similar to people's behaviour.

Both the SRN and the memory buffer models performed better on the test sets where grammaticality was associated with chunks, than when the rule was solely defined in terms of the non-local mapping. The way in which chunks influenced the network's discrimination performance was further investigated by plotting the discrimination performance from the test sets in which grammaticality was associated with differences in chunks against the performance where grammaticality was solely determined by the non-local mapping. Similar to Chater and Conkey (1992) we showed that the SRN was mainly sensitive to chunks. Furthermore, the comparison between these results and the results obtained by the human data revealed that peoples' discrimination performance was uncharacteristic of the SRN's behaviour and more characteristic of the memory buffer model. These results suggest that although similar to the memory buffer model, the SRN did learn to discriminate between grammatical and ungrammatical items when the grammar was only defined in terms

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

of non-local dependencies, the memory buffer model provided a better account of the human behaviour.

We have only shown an advantage for one sort of memory buffer model over the SRN for a certain type of data, we can make no claims about an advantage in general (see Dominey, 1998, for another way of associatively learning events at different time delays). Further, although we chose a buffer of four time steps a priori (we did not simulate with any other buffer sizes), it could be argued by using task characteristics to stipulate part of the model architecture, we have unfairly favoured the buffer model over the SRN. Whether and in what conditions task requirements can fix a buffer size is a question that can only be answered with future research. But we have shown that the SRN barely predicts human performance at all, despite its ability to fit such performance, and also that a certain plausible buffer model fares better.

Two possible ways of construing buffer models might account for future data. One is that the buffer size is fixed for all tasks. The second is that task requirements lead to different buffer sizes, perhaps depending on domain (e.g. auditory vs. visual stimuli) or, more interestingly, as a learned adjustment to the statistical properties of the stimuli. In the latter case, learning might be best understood in terms of a pool of buffer models, each with different buffer sizes; the model that performs best gets progressively weighted more strongly so that buffer size adjusts to task requirements (cf. Jacobs, Jordan, & Barto, 1991).

Elman (1990) pointed out the computational inelegance of fixed buffer models: a certain time window needs to be hard-wired in, indeed a window as large as the longest dependency that can and should be learned, which means many pools of units will be irrelevant most of the time. The SRN *learns* to remember over different time spans depending on task. Further, the SRN has been shown to have interesting properties in terms of the hidden unit representations it can learn (e.g. Cleeremans, 1993; Elman, 1993). Despite the attractions of the SRN, there remains the question of what sort of model actually best explains *human* data, which is our interest. Cleeremans (1993, chapter 5) compared the SRN and a buffer model in terms of their ability to simulate how people learn on a serial reaction time (SRT) task. He found that where the buffer model (with a buffer of four time steps) and the SRN made different predictions, and where the data differed significantly in that respect, the buffer model performed better than the SRN. Specifically, both the buffer model and people could learn a certain probabilistic difference over random intervening material, whereas the SRN could not. Future research needs to consider more ways of directly contrasting the models in explaining human learning.

In the buffer model we implemented, all stored elements are accessible at once. It is neither a 'first in-first out' nor a 'last in-first out' buffer. As discussed by Dienes and Longuet-Higgins (2004), whether people use a buffer for learning in which the first or last item is most accessible first has important and testable implications for the types of musical structures that can be most easily learned, whether context free or context sensitive grammars are relative easier to learn.

Although it was shown that the SRN and the memory buffer model did in fact learn the non-local mapping in the form of a *value–value* mapping between elements, it is uncertain whether this is the type of representation used by the human subjects.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

The inversion rule can also be represented in the form of a *variable–variable* mapping, or what Marcus refers to as operations over variables (Marcus, 2001).

These two forms of representations are very different and it is seems rather unlikely that the SRN or memory buffer model could learn a *variable–variable* mapping. Similarly, from the empirical work presented by Kuhn and Dienes (2005) it is impossible to distinguish between these two forms of representation in terms of what people implicitly learn, and until we know how the inversion rule is represented in the human mind, it remains uncertain whether the models can capture all aspects of implicit learning of musical structures. This must remain a question for future research. Moreover, the types of rules that can be learnt largely depend on the domain in which the stimuli are presented. For example, when presented with letter strings, people tend to become more sensitive towards local rather than non-local dependencies. Indeed, in everyday life, what we learn about letters is what letters can go together; what we learn about music includes long distance relations like transposition. Maybe the ability to track long distance relations arose specifically in the auditory domain because of our need to keep track of people attempting to reproduce sounds given that different people have different productive pitch ranges making transpositions inevitable. It therefore remains to be seen whether the data from other domains is relatively best captured by the SRN or the memory buffer model.

In sum, this paper has considered the SRN and memory buffer models as models of human learning. We argue that the SRN *can* rapidly learn long distance dependencies that people can learn, which is important in evaluating the SRN as a model of people. However, the SRN does not characteristically behave in this way and so it does not constitute an explanation of how people learn long distance dependencies. We show the memory buffer model is more strongly supported by the data of Kuhn and Dienes (2005) by virtue of more strongly predicting those data than the SRN does. Future research can pursue some questions we have begged: for example, what in general determines the size of the buffer and in what domains might the SRN be superior to the buffer model as a model of human learning?

Acknowledgment

The work for this paper was conducted in partial fulfillment of a D. Phil thesis by Gustav Kuhn in the Department of Experimental Psychology at the University of Sussex.

References

Altmann, G. T., & Dienes, Z. (1999). Rule learning by seven-month-old infants and neural networks. *Science, 284*, 875.

Altmann, G. T. M. (2002). Learning and development in neural networks – the importance of prior experience. *Cognition, 85*(2), B43–B50.

Bishop, C. M. (1996). *Neural Networks for pattern recognition*. Oxford: Clarendon Press.

Boucher, L., & Dienes, Z. (2003). Two ways of learning associations. *Cognitive Science, 27*(6), 807–842.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

Chater, N., & Conkey, P. (1992). Finding linguistic structure with recurrent neural networks. In *Proceedings of the fourteenth annual meeting of the cognitive science society* (pp. 402–407). Hillsdale, NJ: Lawrence Erlbaum.

Christiansen, M. H., & Chater, N. (1999). Towards a connectionist model of recursion in human linguistic performance. *Cognitive Science, 23*(2), 157–205.

Cleeremans, A. (Ed.). (1993). *Connectionist models of sequence learning*. Cambridge: MIT Press.

Cleeremans, A., Destrebecqz, A., & Boyer, M. (1998). Implicit learning: news from the front. *Trends in Cognitive Sciences, 2*(10), 406–416.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology-General, 120*(3), 235–253.

Dienes, Z. (1992). Connectionist and memory-array models of artificial grammar learning. *Cognitive Science, 16*(1), 41–79.

Dienes, Z. (1993). Computational models of implicit learning. In D. Berry & Z. Dienes (Eds.), *Implicit learning: Theoretical and empirical issues*. Hove: Lawrence Erlbaum Associates.

Dienes, Z., Altmann, G. T. M., & Gao, S. J. (1999). Mapping across domains without feedback: a neural network model of transfer of implicit knowledge. *Cognitive Science, 23*(1), 53–82.

Dienes, Z., & Longuet-Higgins, C. (2004). Can musical transformations be implicitly learnt? *Cognitive Science, 28*, 531–558.

Dominey, P. F. (1998). A shared system for learning serial and temporal structure of sensori-motor sequences? Evidence from simulation and human experiments. *Cognitive Brain Research, 6*, 163–174.

Dulany, D. E., Carlson, R. A., & Dewey, G. I. (1984). A case of syntactical learning and judgment – how conscious and how abstract. *Journal of Experimental Psychology-General, 113*(4), 541–555.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 16*, 41–79.

Elman, J. L. (1993). Learning and development in neural networks – the importance of starting small. *Cognition, 48*(1), 71–99.

Gomez, R. (2002). Variability and detection of invariant structure. *Psychological Science, 13*(5), 431–436.

Jacobs, R. A., Jordan, M. I., & Barto, A. G. (1991). Task decomposition through competition in a modular connectionist architecture – the what and where vision tasks. *Cognitive Science, 15*(2), 219–250.

Kinder, A. (2000a). Can we do without distributed models? Not in artificial grammar learning. *Behavioural and Brain Sciences, 23*(4), 484.

Kinder, A. (2000b). The knowledge acquired during artificial grammar learning: testing the predictions of two connectionist models. *Psychological Research-Psychologische Forschung, 63*(2), 95–105.

Kinder, A., & Assmann, A. (2000). Learning artificial grammars: no evidence for the acquisition of rules. *Memory & Cognition, 28*(8), 1321–1332.

Knowlton, B. J., & Squire, L. R. (1994). The information acquired during artificial grammar learning. *Journal of Experimental Psychology-Learning Memory and Cognition, 20*(1), 79–91.

Knowlton, B. J., & Squire, L. R. (1996). Artificial grammar learning depends on implicit acquisition of both abstract and exemplar-specific information. *Journal of Experimental Psychology-Learning Memory and Cognition, 22*(1), 169–181.

Kuhn, G., & Dienes, Z. (2005). Implicit learning of nonlocal musical rules: implicitly learning more than chunks. *Journal of Experimental Psychology-Learning Memory and Cognition, 31*(6), 1417–1432.

Longuet-Higgins, H. C. (1987). *Mental processes: studies in cognitive science*. Cambridge, MA: MIT Press.

Luce, R. D. (Ed.). (1963). *Detection and recognition*. New York: Wiley.

Marcus, G. (2001). *The algebraic mind*. Cambridge: MIT Press.

Newport, E. L., & Aslin, R. N. (2004). Learning at a distance 1. Statistical learning of non-adjacent dependencies. *Cognitive Psychology, 48*, 127–162.

Nguyen, D., & Widrow, B. (1990). *Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights*. Paper presented at the IEEE Proc. 1st Int. Joint Conf. Neural Networks.

Olsson, H., Wennerolm, P., & Lyxzen, U. (2004). Exemplars, prototypes, and the flexibility of classification models. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 30*(4), 936–941.

Onnis, L., Monaghan, P., Christiansen, M.H., & Chater, N. (2004). *Variability is the spice of learning, and a crucial ingredient for detecting and generalizing in nonadjacent dependencies*. Paper presented at the Proceedings of the 26th Annual Conference of the Cognitive Science Society, Mahwah, NJ.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003

Perruchet, P., & Pacteau, C. (1990). Synthetic grammar learning – implicit rule abstraction or explicit fragmentary knowledge. *Journal of Experimental Psychology-General, 119*(3), 264–275.

Perruchet, P., & Pacteau, C. (2006). Implicit learning and statistical learning: one phenomenon, two approaches. *Trends in Cognitive Science, 10*(5), 233–238.

Perruchet, P., Tyler, M. D., Galland, N., & Peereman, R. (2004). Learning nonadjacent dependencies: no need for algebraic-like computations. *Journal of Experimental Psychology-General, 133*(4), 573–583.

Perruchet, P., & Vinter, A. (1998). PARSER: a model of word segmentation. *Journal of Memory and Language, 39*, 246–263.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological Review, 103*, 56–115.

Reber, A. S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology – General, 118*(3), 219–235.

Redington, M., & Chater, N. (1996). Transfer in artificial grammar learning: a reevaluation. *Journal of Experimental Psychology-General, 125*(2), 123–138.

Rodriguez, P. (2001). Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural Computation, 13*(9), 2093–2118.

Rodriguez, P. (2003). Comparing simple recurrent networks and *n*-grams in a large corpus. *Applied Intelligence, 19*(1-2), 39–50.

Sejnowski, T., & Rosenberg, C. (1987). Parallel networks that learn to pronounce English text. *Complex Systems, 1*, 145–168.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1991). Graded state machines – the representation of temporal contingencies in simple recurrent networks. *Machine Learning, 7*(2-3), 161–193.

Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology-Learning Memory and Cognition, 16*(4), 592–608.

Shanks, D. R., Johnstone, T., & Staggs, L. (1997). Abstraction processes in artificial grammar learning. *Quarterly Journal of Experimental Psychology Section A-Human Experimental Psychology, 50*(1), 216–252.

Timmermans, B., & Cleeremans, A. (2000). Rules vs. statistics in biconditional grammar Learning: A simulation based on Shanks et al. (1997). In *Proceedings of the twenty-second annual conference of cognitive science society* (pp. 947–952). Hillsdale: Lawrence Erlbaum Associates.

Please cite this article in press as: Kuhn, G., & Dienes, Z., Learning non-local dependencies, *Cognition* (2007), doi:10.1016/j.cognition.2007.01.003