State trace analysis: Dissociable processes in a connectionist network?

F. Yeates[1], A.J. Wills[2], F.W. Jones[3] & I.P.L. McLaren[1]

[1]University of Exeter, School of Psychology

[2]Plymouth University, School of Psychology

[3]Canterbury Christ Church University, School of Psychology

Running head: State trace analysis of connectionist networks

Keywords: state trace analysis; multiple systems; model evaluation; connectionist network; dual processes; computer simulation

Address for correspondence:

Fayme Yeates

School of Psychology

University of Exeter

Washington Singer Laboratories

Perry Road

Exeter EX4 4QG

England

E-mail: fy212@exeter.ac.uk

Phone: +44 (0) 1392 724626

Fax: +44 (0) 1392 724623

ABSTRACT

Some argue the common practice of inferring multiple processes or systems from a dissociation is flawed (Dunn, 2003). One proposed solution is state trace analysis (Bamber, 1979), which involves plotting, across two or more conditions of interest, performance measured by either two dependent variables, or two conditions of the same dependent measure. The resulting analysis is considered to provide evidence that either: (1) a single process underlies performance (one function is produced) or (2) there is evidence for more than one process (more than one function is produced). This article reports simulations using the simple recurrent network (SRN, Elman, 1990) in which changes to the learning rate produced state trace plots with multiple functions. We also report simulations using a single-layer error-correcting network that generate plots with a single function. We argue that the presence of different functions on a state trace plot does not necessarily support a dual-system account, at least as typically defined (e.g. two separate autonomous systems competing to control responding); it can also indicate variation in a single parameter within theories generally considered to be single-system accounts.

1. Introduction

The question of how many psychological processes may be contributing to a particular behavior or effect is often central to research in our discipline. Are there two routes to visual processing? Do children acquire language through a single system? Is there a separate mental system for the processing of faces? Are there separate brain regions for semantic and auditory language processes? Does learning occur implicitly as well as explicitly in humans? All these questions converge on the common issue of: "how many functionally distinct psychological processes are we dealing with?"

The result most often employed to support the presence of multiple processes (multiple latent psychological variables) is the behavioral dissociation. The underlying rationale will be familiar to most researchers in two forms: the single dissociation, which occurs when one manipulates a given independent variable that affects one dependent variable and not another; and the double dissociation, which involves two independent variables that produce complementary single dissociations on the same two dependent variables. The demonstration of such dissociations is often taken to provide evidence for a multiple process/systems hypothesis. This inference, however, has been shown to be insecure (see Dunn, 2003 for an analysis). Many have argued that the use of bounded variables, such as accuracy, may result in floor and ceiling effects that can both produce dissociations in the absence of multiple processes, and may overlook multiple processes in the absence of a dissociation (Loftus, 1978), but Dunn (2003) makes a case for there being more fundamental problems with this approach that go beyond artifacts of this kind. He shows that, whilst one can infer that a variable has an effect on performance of a given task, one can never infer that a variable has no effect on the performance of another task.

State trace analysis (Bamber, 1979), sometimes referred to as dimensional analysis (Loftus, Oberg & Dillon, 2004), is one proposed solution to these ambiguities. Instead of considering variables in terms of their main effects and interactions, it plots them against one another and examines the function(s) that the dependent variables follow. If the dependent variables follow one, single monotonic function then we can reject the idea of multiple processes. This result is taken to suggest that a single latent variable underlies performance, providing confirmation of a "simple and elegant" single-function structure (Loftus, Oberg & Dillon, 2004, p. 838). However, if there is no single monotonic function produced, one must reject the single-function account and infer that more than one process underlies performance – where multiple functions are seen on the state trace plot.

Bamber (1979), Dunn and Kirsner (1988) and Loftus (1978) have all contributed to the development of state trace analysis. An exponentially increasing number of researchers have been using state trace analysis in place of the traditional dissociation logic in recent times, and the method has already been employed in a diverse range of research areas, including category learning (Newell, Dunn & Kalish, 2010; Newell, 2012), cognitive development (Mayr, Kleigl & Krampe, 1996), the face inversion effect (Loftus, Oberg & Dillon, 2004; Prince & Heathcote, 2009), remember-know judgments (Dunn, 2008; Heathcote, Bora & Freeman, 2010) and the neuroscience of recognition memory (Staresina, Fell, Dunn, Axmacher & Henson, 2013).

This increase in popularity may in part be due to the simplicity of state trace analysis, which provides a compelling visual representation of dimensionality. Each state trace analysis requires two *dimensions*, representing either one dependent variable measured under two different conditions, or two different dependent variables. As a concrete example, one could plot recognition accuracy for upright and inverted faces on the x and y axes. Performance is plotted across the *trace* of the experiment, i.e. across some continuous measure of time or

number of blocks to produce the function of interest. In our example, this would correspond to plotting the points representing mean recognition accuracy for upright and inverted faces in each block of an experiment run over several blocks. These plots can then be made for two or more independent variables of interest– these are the *states*. Here an example of a state manipulation would be making plots for 1) performance on faces drawn from one very familiar ethnic group and 2) performance on faces from another less familiar ethnic group. The points in the scatter plot are usually given two-dimensional error bars to aid visual assessment of the case for overlap. The analysis consists of determining whether our two plots are best described as part of one continuous function, or require two distinct functions to capture each trace.

Four idealized state trace plots are shown in Fig. 1, which are based on hypothetical data for the purposes of exposition. Fig. 1C illustrates a single function plot and Fig. 1D a multiple function plot, the latter of which implies a multiple process account of whatever task domain is being investigated. The top two graphs (Fig. 1A and 1B) show situations in which state trace analysis cannot be used, because of the assumptions and requirements of the method. State trace analysis assumes that latent psychological variables have a monotonic effect on performance. Thus, a non-monotonic state trace plot (Fig. 1A) cannot be used to infer dimensionality. Further, if both traces are monotonic, they must overlap at some point on the x or y dimension, otherwise one cannot establish whether they follow the same function or not. Therefore, there may be four possible outcomes to your analysis: non monotonic; no overlap; single function; or multiple functions.

----------Insert Figure 1 about here----------

While an increasing number of researchers are discovering state trace analysis and applying its framework to their research questions, what is not clear is what the status of the processes discovered might actually be. What counts as dissociable processes within the framework of state trace analysis? Must they be two functionally separate processing systems? If indeed a single function on a state trace plot suggests a single latent psychological variable underlies performance, does this mean that in perceiving, learning and recalling faces (not to mention the other motor skills involved in such a task) there is only one cognitive or neurological process or set of processes? And are multiple functions produced only when functionally different processes / systems are evident between states? Newell, Dunn and Kalish (2011, p. 198) point out that "The dimensionality of the state-trace plot reveals the number of underlying latent variables but says nothing about their nature". Our intention here is to try cast some light on the possible relationships between the dimensionality of the state-trace plot and the nature of the processes involved by analyzing examples where we are entirely certain of the nature of the system in question – because it is one we have specified.

Thus, to attempt to answer these questions this paper will consider the performance of computational models, whose processes we can both quantify and manipulate. The simple recurrent network (SRN, Elman, 1990) will be used to simulate a two-choice sequence learning task. Learning will be varied by altering a parameter that controls the rate of change of the connection weights between units (the learning rate parameter). This will result in a number of networks that differ only in this parameter, the rationale being that simply speeding up or slowing down learning in the network (as long as we don't move into regions of parameter space where the learning algorithm exhibits pathological behavior) should not alter the basic nature of the network. As such, it should produce simulations that are characteristic of a single system. This is a novel application of computational modeling to this area (though there are parallels in the work of Bullinaria, 2007), and the point of doing this is that our

understanding of state trace logic predicts that running the same model with different values of this one parameter would not be thought to be the sort of manipulation that would produce multiple functions on a state trace plot (e.g. McCarley & Grant, 2008; Reinitz, Séguin, Peria, & Loftus, 2012; Staresina et al., 2013).

2. SRN simulation details

2.1 Model construction

The SRN (Elman, 1990) is a recurrent, feed-forward connectionist network (see Fig. 2a) that starts with an input layer of units that are set to either a value of 0 (off) or 1 (on). When on, these units feed activation forward (using the logistic activation function: Rumelhart, Hinton & Williams, 1986) into a hidden layer, which in turn feeds activation to an output layer. The hidden unit activations are also copied into a set of context units at the input layer, whose activations are then fed back into the hidden layer as input on the next trial. This produces a recurrent loop, feeding the internal representation of the model back into itself and enabling the model to learn contingencies that do not occur on the same trial (e.g. sequences). The model learns through back propagating error correction, comparing output activations to an expected response and updating the weights between all units within the model appropriately. Performance is calculated by comparing the output activations to their expected values, taking the difference, squaring and averaging to give a mean squared error (MSE). Following the human behavioral experiment (Yeates et al., 2013) on which this simulation is based, 128 networks were run for each simulation, 32 networks for each group (as described below).

The model comprised of two input units and two output units, which represented the two ‘stimuli’ that formed the sequence that the model was trained on. The hidden layer comprised of 20 units; and hence 20 context units as input. The initial connection weights were uniformly distributed to random values between -0.5 and 0.5 for each network. The model’s learning rate was the only parameter manipulated – running networks with different values. The learning rate parameters used (0.15 and 0.4) were the values given in previous work by Cleeremans and McClelland (1991) and Jones and McLaren (2009).

----------Insert Figure 2 about here----------

2.2 Sequence learning task

The task was a two-choice serial reaction time (SRT) task whereby one of two locations on either the right or left of the screen flash and this requires a spatially compatible key press response. These flashes follow a sequence – which in the case of this task has a probabilistic structure. Four groups of networks were run to simulate this task – two experimental and two controls. The control groups were trained on blocks that contained 40 subsequence ‘triplets’ of all the eight possible combinations in a two-choice task: XXX, XXY, XYX, XYY, YYY, YYX, YXY, YXX. An equal number (5) of each triplet were randomly ordered and concatenated (e.g. XXYXYYYYYYXX…) within a block so that there was no obvious delineation of the triplets. In the case of control networks no part of the trial order is predictive as any subsequent trial type is equally likely. The two experimental groups were trained on blocks that contained 40 subsequence ‘triplets’ of half of the possible combinations so that they followed a rule: Group Different - first trial in triplet is opposite to last trial, XXY, XYY, YYX, YXX; and Group Same - first trial in triplet is same as last trial, XXX, XYX,

YYY, YXY. An equal number of each (10) were randomly concatenated within a block, and thus when one considers the trial sequence (e.g. XXXXYXYXYYYYXXX…etc) two-thirds of experimental trials are predictive. This is because every third trial is 100% predictable, as the trial that occurred two trials previously signals what the third trial will be for that group in every instance. On every first and second trial it is equally likely that the trial either follows this rule or not, thus the overall probability of any given trial following the rule is two thirds. Networks were trained on 35 blocks (4200 trials) and tested over 5 blocks (600 trials) after training, where all groups received pseudorandom sequences containing all possible triplets. This trial number was chosen to match that used in our previous work in order to ensure that the models learnt the sequences (Yeates et al., 2013). We chose the task we used because we knew the SRN could simulate it well, and as such is (in slightly modified form) our best current model for human performance on this type of sequence learning (Jones & McLaren, 2009; Yeates et al., 2013). As we will see, it also lends itself well to state-trace analysis.

Learning was measured by taking the difference between performance on trials that do not follow the rule (Inconsistent Trials) minus performance on trials that follow the rule (Consistent Trials). As lower MSE represents better performance, higher values of the Inconsistent-minus-Consistent measure denote better learning of the trained sequences. Control networks were not trained to a particular rule, but are assigned one as a dummy variable and the equivalent difference calculated. These control groups are needed to control for sequential effects (see Anastasopoulou & Harvey, 1999; Jones & McLaren, 2009; Yeates et al., 2013) as performance on a particular subsequence may be easier than another, thus our Inconsistent-Consistent measure alone does not adequately index learning, it needs to be evaluated by comparison with the appropriate control differences. A difference between the difference scores for Experimental and Control networks is therefore calculated, and this is

used to demonstrate how much the networks have learned about the sequential structure they have been exposed to.

2.3 Results

An ANOVA was run in order to demonstrate whether learning had occurred, comparing experimental and control groups across training. The training data for Groups Different and Same were analyzed separately, with the factor of condition (experimental versus control) alongside the repeated measure block. The SRN exhibited learning for both experimental groups' sequences at both learning rates as demonstrated by the main effect of condition in all cases (experimental > control). For the SRN with a learning rate of 0.15 a main effect of condition was found for Group Different, $F(1,62) = 237.1$, $p < .001$, and Group Same, $F(1,62) = 217.8$, $p < .001$. Learning was also evident in the SRN with a learning rate of 0.4 in Group Different, $F(1,62) = 354.7$, $p < .001$, and Group Same, $F(1,62) = 537.5$, $p < .001$.

Using the simulation data, a state trace analysis was then conducted. This involved plotting the learning scores of the networks across 7 epochs of training (1 epoch = 5 blocks), containing 600 trials each (the *trace*). Performance on the two sequence learning tasks (Group Different and Group Same) form the two *dimensions* on the x and y axes, respectively. Performance at each learning rate was plotted separately as one of two *states*. Following McCarley and Grant (2008), a visual inspection of the plot was carried out. The state trace plot can be seen in Fig. 3A, which on visual inspection clearly shows two separate functions, rather than one single monotonically increasing function. This suggests that state trace analysis is sensitive to the differences between the two sets of simulations, and therefore that a purely parametric manipulation (speeding up learning) can lead to multiple processes being inferred if one employs the state trace methodology.

The plot (Fig. 3A) could be analyzed in a variety of ways, from visual inspection (McCarley & Grant, 2008), to Spearman's Rho (Loftus, Oberg & Dillon, 2004; Prince & Heathcote, 2009), maximum likelihood estimation (MLE, Newell & Dunn, 2008), hierarchical linear regression (Yeates, Jones, Wills & McLaren, 2012) and Bayesian models (Prince et al., 2012). We settled on a hierarchical linear regression as the preferred method to examine the number of functions within the plots. Group Different scores were used to predict Group Same performance. The learning rate was then added as a predictor and a statistically significant change in R-square taken as evidence for multiple functions. The hierarchical multiple regression demonstrates that the addition of learning rate to the model significantly improves the R^2_{adj} value from 94.6% to 98.1%, ΔR^2: $F(1,11) = 23.6$, $p = .001$. This model, Group Different = 0.79(Group Same) + 0.97(Learning Rate) – 0.007, showed significant fit between model and data, $F(2,11) = 343.1$, $p < .001$. This provides good evidence against the state trace plot being adequately described as one monotonic function.

2.4 Discussion

The state trace plot (Fig. 3A) demonstrates that increasing the learning rate of the SRN networks increases the amount of learning of Group Same relative to Group Different sequences. This suggests that there are multiple processes that underlie the SRN's performance on the two tasks. These simulations demonstrate that state trace analysis is sensitive to the effect that variations in the rate of learning can have on a simple recurrent network. Our result may be analogous to one that could be obtained by assessing task performance as a function of individual differences, or by manipulating differences in attention, context, or indeed any number of exogenous factors. How are we to interpret this result in terms of multiple processes or systems, given that the SRN embodies what would

often be considered to be a single (associative) process account of learning? Obtaining multiple functions on the state trace plot in these circumstances came as a surprise to us, and, we imagine, will surprise many researchers with an interest in this methodology. We predicted that varying the learning rate would simply vary the rate of acquisition of the problems, but that the different plots would nevertheless form a smooth, coherent function. These predictions have been roundly disconfirmed, and now we have to ask ourselves why this is so, and what are the implications for state trace analysis?

3. Single layer network

To enable us to investigate further to what extent the state trace plot is sensitive to differences in model parameters, we chose to simulate the same task on a conceptually simpler model – a single layer error-correcting network (see Fig. 2b). The idea is that this model will act as a "control" for the SRN simulations we have just reported. This model lacks any more complex component (e.g. recurrence, multiple layers of weights) but still learns through error-correction. In this case then, it is hard to see how a state trace plot with multiple functions could occur when one varies the learning rate parameter. If this turns out to be the case, and we obtain a single (uni-dimensional) plot in this case, then we will have evidence that it is the greater complexity of the SRN that led to the multiple function plot in our previous simulations.

3.1 Simulation details

To obtain a single layer network we modified the SRN from the description above so that 1) the context units were always set to zero, eliminating recurrence and 2) each input unit

had just one fixed weight to a corresponding hidden unit, with the weight of all such connections set to a fixed value of 0.5. This effectively reduces the SRN to a single layer, error-correcting network; albeit one that is still using a non-linear activation function and otherwise operates in a similar fashion to the earlier SRN. To enable the network to learn the sequences presented to it, we included two additional input units that provided trial n-1 as input (as well as the existing units already providing trial n as input) to predict trial n+1 as output.

3.2 Sequence learning task and procedure

Both the sequence learning task and procedure followed were the same as described above for the SRN.

3.3 Results

An ANOVA was conducted as before to investigate whether learning had occurred. The single layer networks demonstrated learning (experimental better than control) on both groups of sequences with both learning rates. The single layer network with a learning rate of 0.15 demonstrates a main effect of condition for Group Different, $F(1,62) = 441.3$, $p < .001$, and Group Same, $F(1,62) = 637.6$, $p < .001$. The main effect of condition was also significant in the single layer networks with a learning rate of 0.4 in Group Different, $F(1,62) = 2719.8$, $p < .001$, and Group Same, $F(1,62) = 3285.2$, $p < .001$.

We constructed the equivalent state trace plot to the SRN networks (Fig. 3A) for the single layer networks, this is shown in Fig. 3B. Visual inspection immediately reveals that this time the plots seem to lie on a single function, though changing the learning rate has

obviously had a substantial impact on performance. Analysis of these plots revealed that there was no evidence that adding learning rate as a factor improved the regression ($F(1,11) = 1.13$, $p = .3$ for the change), confirming that a single linear function adequately describes the data from these simulations. This model, Group Different = 1.12(Group Same) – 0.002, demonstrated a significant fit between the model and data, $F(2,11) = 2283$, $p < .001$, and accounted for 99.4% of the variance.

----------Insert Figure 3 about here----------

3.4 Discussion

With a single layer network, relative performance on Group Same to Group Different sequences was consistent, regardless of the learning rate. Thus, with these networks, a single function was visualized on the state trace plot (Fig. 3B) when we varied the learning rate. This is consistent with a single process account for this learning system as we expected. In the single layer network, only one set of weights can change, and the rate of change is influenced by the parameter we varied. In the SRN, however, there are two layers of weights, and in addition there are recurrent connections that, though they are themselves fixed, nevertheless have a strong influence on the learning that takes place in the system by virtue of supplying much of the input that drives that learning. The conclusion we are pushed towards, then, is that the state trace methodology is sensitive to these differences between our two specimen networks, and that it is capable of making process distinctions at a much finer grain than may have hitherto been suspected by researchers employing this methodology.

4. General discussion

When changing the learning rate parameter of the SRN a multiple function state trace plot (Fig. 3A) is produced, suggesting the existence of multiple processes within the model. This result went against our intuitive predictions about state trace analysis, leading us to question the requirements for a multiple function plot. A higher learning rate increases the amount of learning of Group Same sequences relative to those in Group Different for the SRN, but a simple single layer network performs consistently on Group Same relative to Group Different sequences, regardless of the learning rate. Therefore, the multiple functions observed in the SRN simulations are reduced to a single function when the model is altered to a simple single layer network. This suggests that there are not multiple processes at work in this case, even though this network, like the SRN, uses non-linear activation functions and a number of parameters that could be varied to influence learning. Given that when one of these parameters (the learning rate parameter) is varied, the plots obtained indicate that a single latent variable or process is responsible for performance on our task in this case, we have an existence proof that simply adding layers and recurrence to a connectionist network is enough to transform it from a single-process to a multi-process system in state trace terms.

As suggested above, this indicates that state trace analysis is sensitive to the presence of process differences at a much finer level than was perhaps initially realized. One implication of this result is that state trace analysis can reveal multiple processes within what might be considered to be a single system. When we take into account the single function obtained with the single layer network simulations, a corollary is that state trace analysis might not only be capable of distinguishing at a relatively gross level between, for example, an associative system and another system based on a different kind of computation, but could also distinguish between varieties of associative network.

We are not usually in the situation of knowing exactly what the computational specification of the system that we are dealing with is, as was the case here. When we apply state-trace analysis to data derived from humans or other animals, the aim is to tease out the processes involved in task performance so that we are then able to construct better models of human or infra-human learning. Here, we were able to manipulate our models so as to help us interpret the results of our state-trace analysis. What are the implications now for the application of state-trace analysis to experimental data where the underlying processes are unknown?

We believe that our findings compel us to qualify the conclusions that can be drawn from a state trace plot that reveals multiple functions. Clearly, as Newell et al. (2011) acknowledge, one cannot securely infer the presence of two functionally dissociable systems from a two-function state trace plot. We have demonstrated in a concrete way that it could simply reveal that performance is based on a single, multi-process system, if variation in the state variable differentially affected those processes, and altered their relative contributions to performance. This possibility, in turn, makes it somewhat harder to interpret a plot with a single function as well. The reason is that, if multiple functions can be a consequence of parametric variation altering the relative contributions made by different processes, then a single function could be produced by the change in the state variable affecting these processes equally. If their relative contributions are not changed, then we might expect state trace analysis to indicate a single, monotonic function, suggesting that only one process need be invoked. The fact is, however, that this result might be due to a single process, or to a set of (in this case) correlated processes. We find ourselves with the possibility of one state trace analysis suggesting that a multi-process explanation is required for task performance, whereas another on the same system might indicate that a single process would suffice. Given that this could, in principle, be the case, how then are we to proceed?

Our tentative answer to this question is to abandon the one function = single system, multiple functions = multiple system dichotomy, and instead adopt an approach couched in terms of sets of processes that can act like a single system/process in some circumstances, but reveal their multiple process nature in others. If a state trace plot reveals multiple functions – then there are multiple processes involved. If, another analysis using a different state variable but otherwise employing the same paradigms now produces a single function, then this should not be taken to contradict the earlier finding, but simply indicates that in these circumstances the multiple processes are equivalent to one single process because the state variable affects them in a non-differential fashion. We can never be sure that there is only one process in play given a single function on a state trace plot, as on our analysis, the definitive result is always the one with multiple functions. But multiple functions do not necessarily signify functionally separable processes at a gross level (i.e. completely different types of computation). Instead, we can allow that there might be different sub-types of the same computational process as in our SRN example, where recurrence, and learning of the non-linear mappings from the input to the hidden units and the hidden to the output units were the processes differentially affected by changing the learning rate.

To further clarify our new understanding of what we mean by "process", another, illustrative example can be extrapolated from the work of Wills and McLaren (1997) and Jones, Wills and McLaren (1998). Both these papers make the case for a competitive process that translates the categorical outputs of a network into a real-time response using a winner-take-all approach. This could be added to the simple single layer network considered here, and would constitute another process that could be discovered by means of state-trace analysis, without actually being a qualitatively different kind of computation. Hence, one interpretation of a "process" is that it can refer to part of the architecture of a model that performs a certain computation as in this case. Another, equally valid possibility is that it

could be just what it says, a process, that acts within a model architecture but is governed by its own parameters so that it can decouple from other processes that are also at work. For an example of what we mean by this see McLaren and Dickinson's (1990) discussion of how Hebbian and anti-Hebbian processes might interact within a connectionist network.

With these caveats in mind, we conclude that state-trace analysis still has something to offer our discipline. It allows us to test the hypothesis that two functionally separable sets of processes contribute to performance on a given task (analysis must produce a multiple function plot to be consistent with this assumption as long as steps are taken to ensure that these processes do not co-vary). It also enables us to detect multiple processes within single systems, allowing a more detailed analysis of that system's components. Thus, we believe that state trace analysis can still be a valuable methodological tool in the behavioral scientist’s armory.

References

Anastasopoulou, T., & Harvey, N. (1999). Assessing sequential knowledge through performance measures: The influence of short-term sequential effects. *Quarterly Journal of Experimental Psychology, 52A*, 423-448.

Bamber, D. (1979). State-trace analysis: A method of testing simple theories of causation. *Journal of Mathematical Psychology, 19*, 137-181.

Bullinaria, J.A. (2007). Understanding the emergence of modularity in neural systems. *Cognitive Science, 31,* 673-95.

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

Diamond, R., & Carey, S. (1986). Why faces are and are not special: An effect of expertise. *Journal of Experimental Psychology: General, 115,* 107-117.

Dunn, J. C. (2003). The elusive dissociation. *Cortex, 39,* 177-179.

Dunn, J. C. (2008). The dimensionality of the remember-know task: A state-trace analysis. *Psychological Review, 115*, 426-446.

Dunn, J. C., & Kirsner, K. (1988). Discovering functionally independent mental processes: The principle of reversed association. *Psychological Review, 95,* 91-101.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14* (2), 179-211.

Heathcote, A., Bora, B., & Freeman, E. (2010). Recollection and confidence in two-alterative forced choice episodic recognition. *Journal of Memory and Language, 62,* 183-203.

Jones, F. W., & McLaren, I. P. L. (2009). Human sequence learning under incidental and intentional conditions. *Journal of Experimental Psychology: Animal Behavior Processes, 35* (4), 538-553.

Jones, F., Wills, A.J., and McLaren, I.P.L. (1998). Perceptual categorisation: connectionist modelling and decision rules. *Quarterly Journal of Experimental Psychology*, *51* B,

33-58.

Loftus, G. R. (1978). On interpretation of interactions. *Memory & Cognition, 6,* 312-319.

Loftus, G. R., Oberg, M. A., & Dillon, A. M. (2004). Linear theory, dimensional theory, and the face-inversion effect. *Psychological Review, 111,* 835-863.

Mayr, U., Kleigl, R., & Krampe, R. T. (1996). Sequential and coordinative processing dynamics in figural transformations across the life span. *Cognition, 59,* 61-90.

McCarley, J. S., & Grant, C. (2008). State-trace analysis of the effects of a visual illusion on saccade amplitudes and perceptual judgments. *Psychonomic Bulletin & Review, 15* (5), 1008-1014.

McLaren, I.P.L. and Dickinson, A. (1990). The conditioning connection. *Philosophical Transactions of the Royal Society of London, 329* B, 179-186.

Newell, B. R. (2012). Levels of explanation in category learning. *Australian Journal of Psychology, 64,* 46-51.

Newell, B. R., & Dunn, J. C. (2008). Dimensions in data: Testing psychological models using state-trace analysis. *Trends in Cognitive Science, 12,* 285-290.

Newell, B. R., Dunn, J. C., Kalish, M. (2010). The dimensionality of perceptual category learning: A state-trace analysis. *Memory & Cognition, 38,* 563-581.

Newell, B.R., Dunn, J.C., & Kalish, M. (2011). Systems of category learning: Fact or fantasy? In B.H. Ross (Ed) *The Psychology of Learning & Motivation, 54,* 167-215.

Prince, M., Hawkins, G., Love, J., & Heathcote, A. (2012). An R package for state-trace analysis. *Behavioural Research, 44,* 644-655.

Prince, M., & Heathcote, A. (2009). State-trace analysis of the face-inversion effect. In N. A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Reinitz, M. T., Séguin, J. A., Peria, W., & Loftus, G. R. (2012). Confidence-accuracy relations

for faces and scenes: Roles of features and familiarity. *Psychonomic Bulletin and Review, 19,* 1085-1093.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representation by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* (Vol. 1: Foundations.). Cambridge, MA: Bradford Book/MIT Press.

Staresina, B. P., Fell, J., Dunn, J. C., Axmacher, N., & Henson, R. N. (2013). Using state-trace analysis to dissociate the functions of the human hippocampus and perirhinal cortex in recognition memory. *PNAS: Proceedings of the National Academy of Sciences of the United States of America, 110,* 3119-3124.

Wills, A.J., and McLaren, I.P.L. (1997). Generalisation in human category learning: a connectionist explanation of discriminative vs. non-discriminative training gradient differences. *Quarterly Journal of Experimental Psychology*, *50* A, 607-30.

Yeates, F., Wills, A., Jones, F. & McLaren, I.P.L. (2012). State-Trace analysis of sequence learning by simple recurrent networks. In N. Miyake, D. Peebles, & R. P. Cooper (Eds.), *Proceedings of the 34th Annual Conference of the Cognitive Science Society* (pp. 2581-2586). Austin, TX: Cognitive Science Society.

Yeates, F., Jones, F. W., Wills, A. J., McLaren, R. P., & McLaren, I. P. L. (2013). Modeling human sequence learning under incidental conditions. *Journal of Experimental Psychology: Animal Behavior Processes, 39* (2), 166-173.

Acknowledgements

The authors would like to thank Christopher Chatham and the reviewers for their insightful comments.

Figure Captions

Figure 1: Hypothetical state trace plots, showing four possible outcomes of a state trace analysis of Dimension 1 against Dimension 2 for State 1 and State 2. The top two state trace plots demonstrate instances where no conclusions regarding dimensionality may be made, as the states are either non-monotonic (A) or do not overlap (B). The bottom two plots demonstrate hypothetical single function (C) and multiple function (D) outcomes.

Figure 2: Model architectures for both the SRN (top panel, A) and the single layer network (bottom panel, B). Circles represent units within the model with three black dots representing further units not shown. The SRN has two input units, representing the two stimuli that make up the sequence the networks are trained on at time *t*. The single layer network requires these units as well as a further two input units in order to learn these sequences, which provide information about the two stimuli on the previous trial, at time $t - 1$. Both models have two output units and twenty hidden units. The SRN has a further twenty context units, whose activations are constantly set to zero in the single layer network, effectively removing them from the model architecture (shown here for illustrative simplicity). Weighted connections that update through error-correction are shown by dotted lines. Fixed connections, whose weights do not alter, are shown by solid lines.

Figure 3: Top panel (A): state trace plot of mean performance of Group Different against mean performance of Group Same by 128 SRN networks with a learning rate of 0.15 and 128 SRN networks with a learning rate of 0.4 across 7 epochs of training (1 epoch=5 blocks). Error bars give 1 SE. Bottom panel (B): similar plot for single layer networks run with the same learning rate parameters (see text for additional details).

Figures

Figure 1.

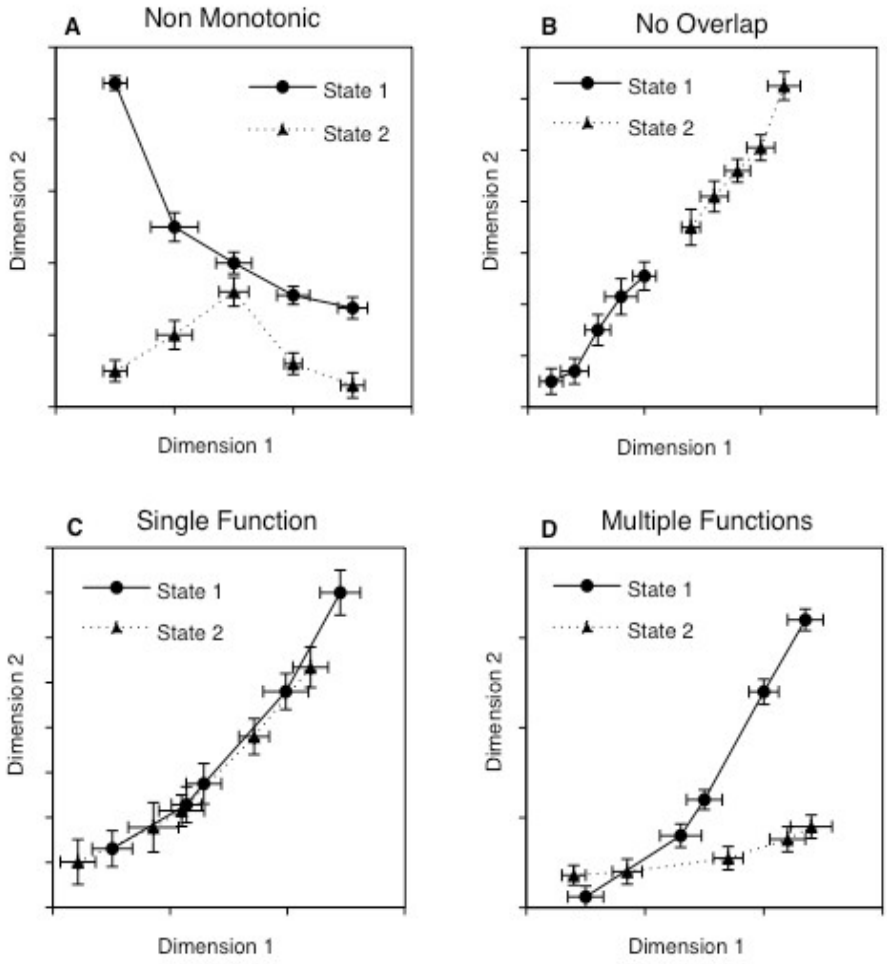
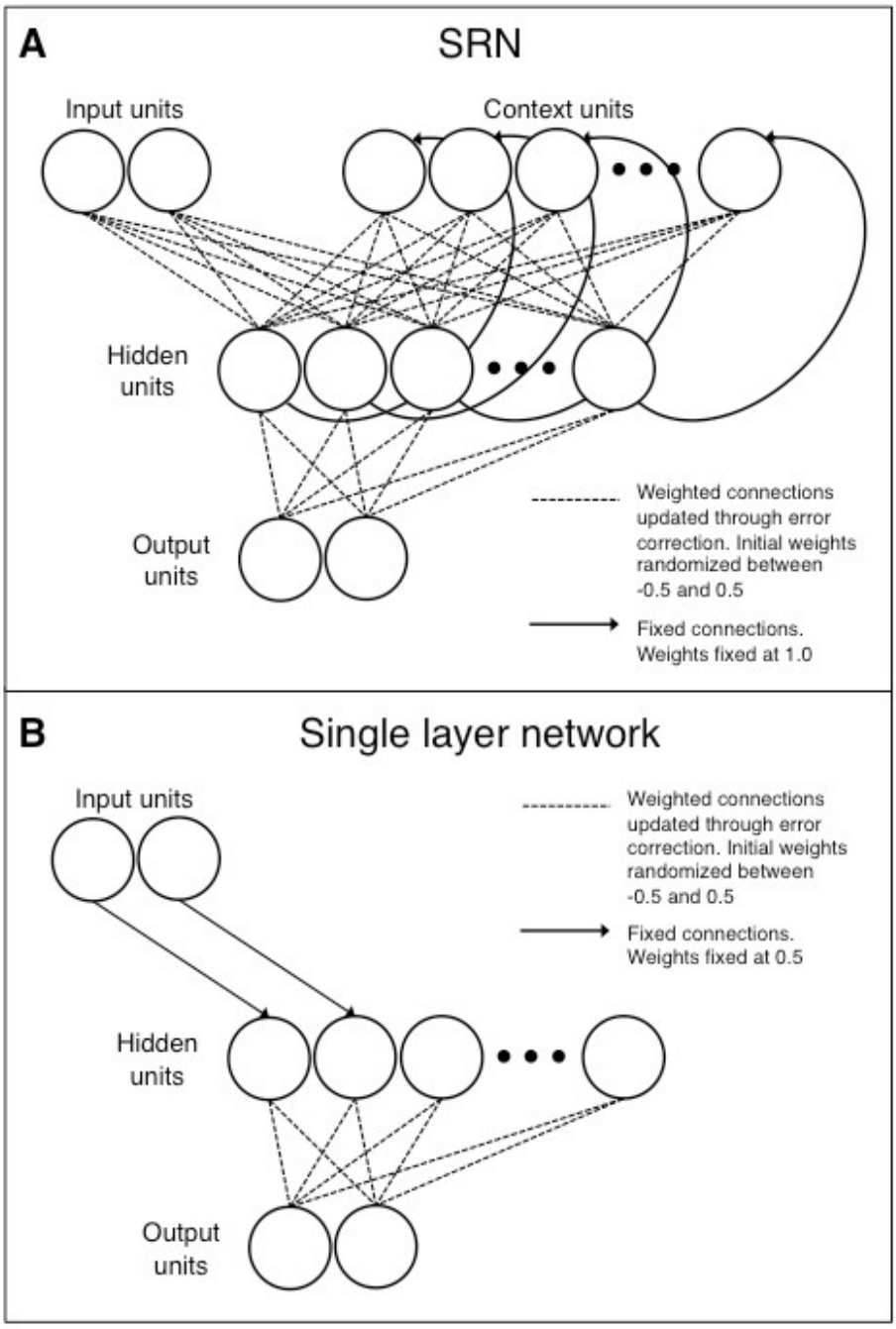

A
Non Monotonic
State 1
State 2
Dimension 2
Dimension 1
B
No Overlap
State 1
State 2
Dimension 2
Dimension 1
C
Single Function
State 1
State 2
Dimension 2
Dimension 1
D
Multiple Functions
State 1
State 2
Dimension 2
Dimension 1

Figure 2.


A
SRN
Input units
Context units
Hidden units
Output units
Weighted connections updated through error correction. Initial weights randomized between -0.5 and 0.5
Fixed connections. Weights fixed at 1.0
B
Single layer network
Input units
Hidden units
Output units
Weighted connections updated through error correction. Initial weights randomized between -0.5 and 0.5
Fixed connections. Weights fixed at 0.5

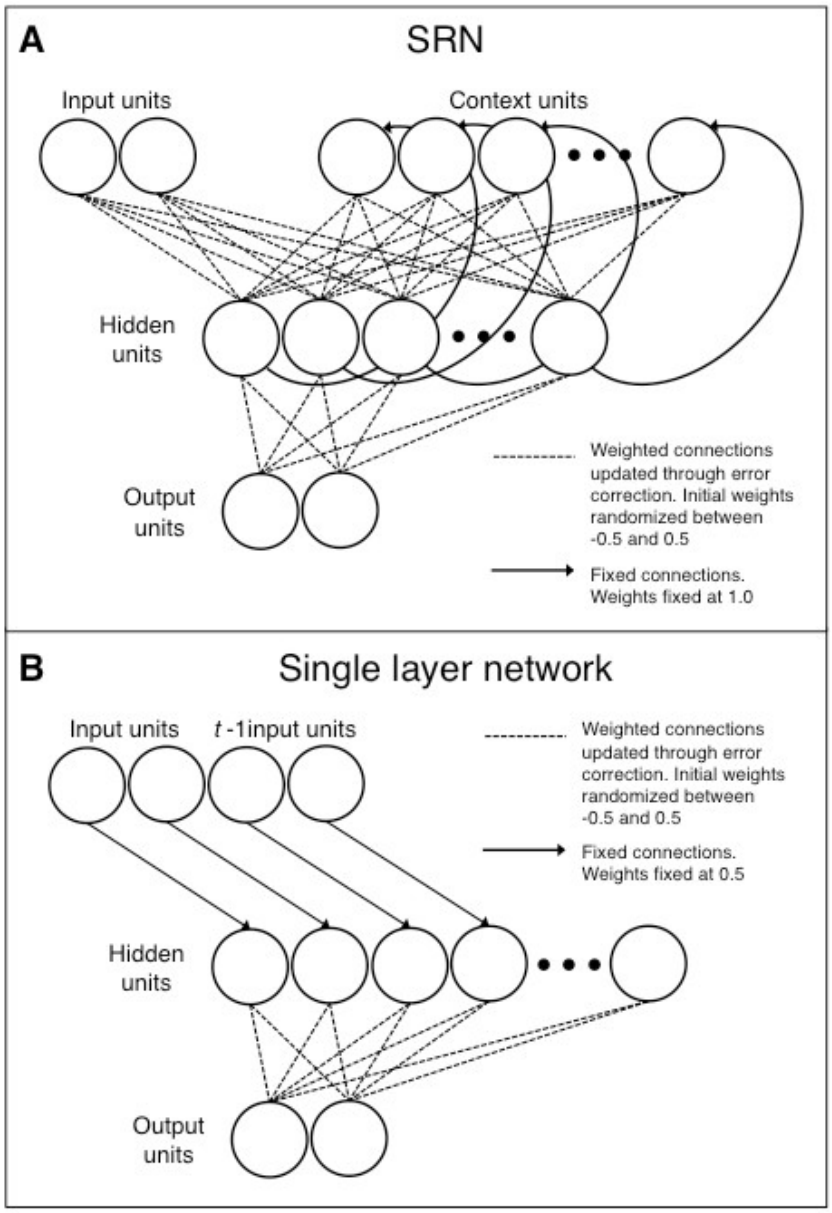
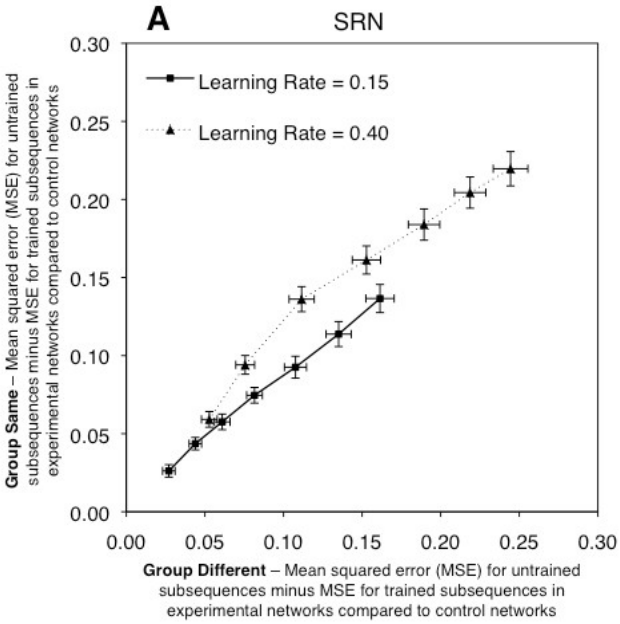
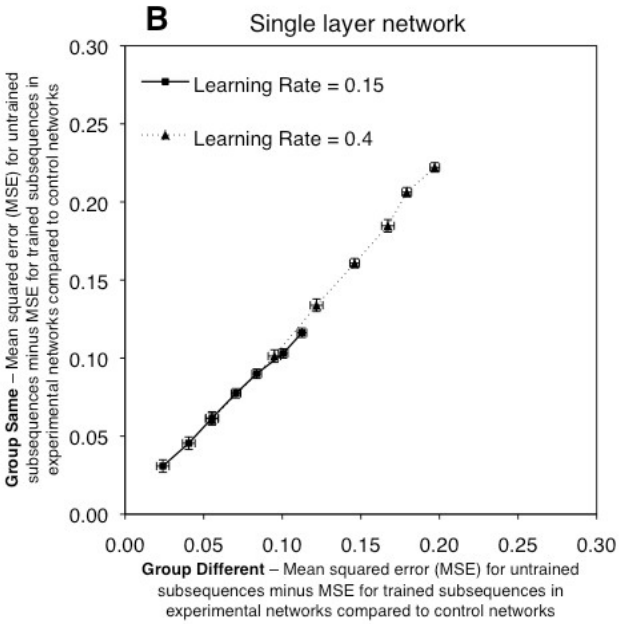

A
SRN
Input units
Context units
Hidden units
Output units
Weighted connections updated through error correction. Initial weights randomized between -0.5 and 0.5
Fixed connections. Weights fixed at 1.0
B
Single layer network
Input units
t -1input units
Hidden units
Output units
Weighted connections updated through error correction. Initial weights randomized between -0.5 and 0.5
Fixed connections. Weights fixed at 0.5

Figure 3.


A
SRN
Learning Rate = 0.15
Learning Rate = 0.40
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Group Same – Mean squared error (MSE) for untrained subsequences minus MSE for trained subsequences in experimental networks compared to control networks
Group Different – Mean squared error (MSE) for untrained subsequences minus MSE for trained subsequences in experimental networks compared to control networks


B
Single layer network
Learning Rate = 0.15
Learning Rate = 0.4
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Group Same – Mean squared error (MSE) for untrained subsequences minus MSE for trained subsequences in experimental networks compared to control networks
Group Different – Mean squared error (MSE) for untrained subsequences minus MSE for trained subsequences in experimental networks compared to control networks