

Redefining the rules: providing race models with a connectionist learning rule

Denis Cousineau*, Guy L. Lacroix[†] and Sébastien Hélie[‡]

**Département de Psychologie, Université de Montréal, C. P. 6128, Succ. Centre-ville, Montréal (Québec), H3C 3J7 Canada*

email: denis.cousineau@umontreal.ca

tel: (514) 343-7981;

fax: (514) 343-2285

[†]*Concordia University, Centre for the Study of Learning and Performance, 7141 Sherbrooke Street W., Montreal, Quebec, Canada H4B 1R6*

[‡]*Université du Québec à Montréal, Laboratoire d'Études en Intelligence Naturelle et Artificielle, Département de Psychologie, C.P. 8888 Succ. Centre-Ville, Montréal, Québec, Canada H3C 3P8*

Abstract. In this paper, we describe the Parallel Race Network (PRN), a race model with the ability to learn stimulus–response associations using a formal framework that is very similar to the one used by the traditional connectionist networks. The PRN assumes that the connections represent abstract units of time rather than strengths of association. Consequently, the connections in the network indicate how rapidly the information should be sent to an output unit. The decision is based on a race between the outputs. To make learning functional and autonomous, the Delta rule was modified to fit the time-based assumption of the PRN. Finally, the PRN is used to simulate an identification task and the implications of its mode of representation are discussed.

Keywords: learning, race networks, delta rule, redefined linear algebra

1. Introduction

Connectionist networks have proven to be insightful models of human cognition. This is reflected in a recent survey showing that their progression in the literature has been exponential (Golden 2001). Their popularity stems from their ability to simulate both learning and knowledge representation using a relatively small set of assumptions. These models use very simple processing units embedded within a large network. Information is stored as weighted associations and learning is achieved either by input accommodation (unsupervised learning) or error reduction (supervised learning).

Although it enjoys less popularity than the connectionist network family, there exists a second family of models that can use simple processors to simulate cognition. It is the family of sampling models, which includes accumulator models such as random walk and race models (Luce 1986). They share the common assumption that the senses (or input units) are sampled once or many times to produce a noisy representation of the information obtained from the outside world. This way of representing knowledge makes

sampling models very powerful models to predict response time distributions (Townsend and Ashby 1983). However, they are not used nearly as often as the connectionist network family for one important reason. As yet, there exists no learning rule that allows these models to process information dynamically and autonomously. Thus, sampling theories have been useful when an analytical solution to a given problem is sought (Huber and Cousineau, submitted). However, as far as simulating human cognition goes, they cannot compete with connectionist networks.

The goal of this text is to bridge this gap between connectionist networks and sampling theories. We shall show how a new sampling model, the Parallel Race Network (PRN), can be organized into a network that can simulate brain-like information processing. Our discussion will proceed as follows. First, we shall present a member of the connectionist network family, the perceptron. We are well aware that many readers are familiar with this model already, but in order to lay the foundation for the subsequent discussion of race models, we need first to review the formal aspects of perceptrons: the architecture, the input–output representation, the integration rule and the learning rule. Then, we shall briefly introduce an overview of several sampling theories with the aim of showing clearly these models’ focus on a time-based representation of the information rather than a strength-based one as is the case for connectionist networks. This is a profound change that allows thinking about brain processes from a radically different perspective. This will be followed by the presentation of the PRN. We shall show that the formal tools used to run simulations with connectionist networks also work with the PRN. This breakthrough is linked directly to our reanalysis of the Delta rule. We shall show that this rule can be divided into two constituent operators, the joining operator and the aggregation function, which can be modified to accommodate race models. Finally, we shall present simulations of a simple identification task conducted with the perceptron and the PRN. This will allow us to argue that the PRN is a compelling alternative to connectionist networks.

2. One type of connectionist network: the perceptron

We begin our discussion with a well-known member of the connectionist network family, the perceptron (Widrow and Hoff 1960, Rosenblatt 1961). The limits of this network have been clearly documented. For example, it cannot solve non-linearly separable problems such as the XOR (Minsky and Papert 1969). Although newer network models have been shown to be more powerful due to the addition of different innovations such as hidden layers (McClelland and Rumelhart 1988), more effective learning rules (O’Reilly 1996), recurrent architectures (Anderson 1995), lateral inhibition (Kohonen 1984) and unsupervised learning (Hinton and Sejnowski 1998), the perceptron possesses all the fundamental attributes of these current connectionist networks regardless of its apparent simplicity. Thus, the formal description of the perceptron that we present here will allow us to highlight the commonalities and the differences between connectionist and time-based networks.

2.1. Architecture

The perceptron is a feed-forward network of connections between processing units. Hence, it is a unidirectional network and it does not allow for recursion. Typically, it is built with only input and output units, but more levels of processing units may be added if desired. Figure 1(a) illustrates a simple architecture.

2.2. Input–output representation

The processing units in the perceptron represent input using a strength of activation approach. By convention, a value of ‘0’ shows no activation (the input is said to be *Off*),

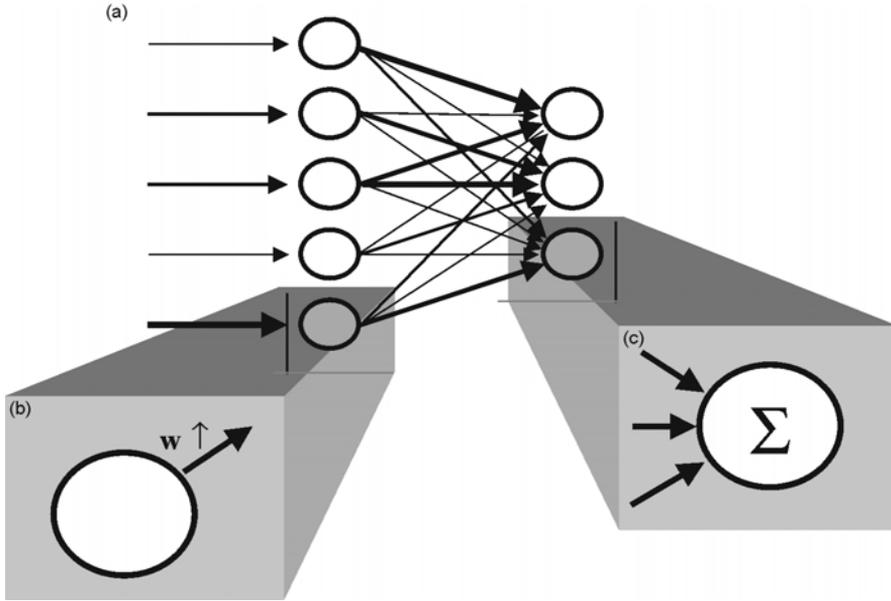


Figure 1. Representation of a simple strength-based network with two layers of units, the perceptron. (a) Architecture and input–output visual representation (b) Weights as an amplitude. (c) The integration rule using a summation.

whereas ‘1’ shows very strong activation (the input is said to be *On*). Strength of activation is a continuous variable. Therefore, any value between 0 and 1 may be registered.

For consistency, the connections between the processing units are also represented as strengths of association. Moreover, inhibitive connections may be represented in the network with the use of a negative value. For instance, ‘ -0.5 ’ shows a moderately strong inhibitive association. Finally, the outputs are also assigned the same numerical scale so that ‘0’ shows no response and ‘1’ shows a strong response.

As all numerical values in these networks represent strengths, connectionist networks can be called strength-based networks. Using the analogy presented in figure 1(b), the connections can also be described as weights showing the amplitude of the associations.

2.3. Integration rule

In a feed-forward architecture, the signal must propagate from the input layer (representing the senses) to the following layer. The equation describing how these signals are transformed is called the integration rule. In the perceptron, the signals are modified by the weight of the connections through which they must travel.

Let \mathbf{W} be a matrix $\{w_{ij}\}$ containing all the weights for the connections linking unit i on the input layer to unit j on the second layer. Furthermore, let \mathbf{A} be the input vector $\{a_i\}$ representing the strength of the activation on the i th input unit. Then, the total activation of a unit on the second layer is assumed to be a sum of its inputs weighted by the connection, as illustrated schematically in figure 1(c). Formally, if we denote the j th unit on the second layer by o_j , we can write:

$$o_j = \sum_{i \in \text{Input}} a_i \times w_{ij}. \quad (1)$$

In order to compute the activation of all the units on that layer, we can generalize equation (1) using the vector and matrix defined above. Therefore, \mathbf{O} , the resultant vector $\{o_j\}$ is given by $\mathbf{O} = \mathbf{A} \cdot \mathbf{W}$, in which the dot represents the standard inner product. It is important to note that the inner product actually represents two operations. First, it joins pairs of values, the inputs and the weights of the connections, by multiplication ($a_i \times w_{ij}$). Second, it aggregates all the received activation at an output unit by summing each connection's level of activation. Thus, we may note the integration rule more explicitly:

$$\mathbf{O} = \mathbf{A}_{\left(\begin{smallmatrix} \times \\ \Sigma \end{smallmatrix}\right)} \mathbf{W}, \quad (2)$$

in which $\left(\begin{smallmatrix} \times \\ \Sigma \end{smallmatrix}\right)$ expands the inner product to show separately the multiplication as the joining operation and the summation (Σ) as the aggregation operator. This notation, although more cumbersome, will be critical to our description of the PRN.

Let us mention that this inner product is compatible with the input–output representation that we have assumed because 1 is neutral with respect to the multiplication and 0 is neutral with respect to the summation. This implies that a matrix with a diagonal composed of ones, while the remaining values are zeros, will be neutral with respect to the inner product. This matrix is well known as the identity matrix \mathbf{I} such that $\mathbf{I} \cdot \mathbf{A} = \mathbf{A} \cdot \mathbf{I} = \mathbf{A}$, for any matrix \mathbf{A} .

Finally, note that strength-based networks, such as the perceptron, do not predict response times, only the strength of the responses. Whether these networks can successfully explain response times has not yet been established (but see Usher and McClelland 2001). However, it is easy to demonstrate that any perceptron with a sufficiently large number of processing units will produce a normal distribution of activation at the output if there is noise in the input and/or the connections. This result is obtained by use of the Central Limit Theorem (Cramér 1946, Feller 1957).

2.4. Learning rule

Traditionally, the Delta rule is used to implement learning in the perceptron (Rumelhart *et al.* 1986). It modifies erroneous outputs by increasing or decreasing the weights of the connections as a function of input strength. Thus, the change in a connection weight, Δw_{ij} , is proportional to the amount of error multiplied by the strength of the input:

$$\Delta w_{ij} \propto a_i \times (e_j - o_j),$$

in which e_i is the expected output on unit j . In matrix form, we may write:

$$\Delta \mathbf{W} \propto \mathbf{A}_{\times} (\mathbf{E} - \mathbf{O}), \quad (3)$$

in which \mathbf{E} is the vector $\{e_i\}$ of the expected output on all the units and \times is the outer product, showing explicitly that pairs of values are joined using multiplication. Weights are updated by integrating the old weights and the corrections with a summation:

$$\mathbf{W} \xleftarrow{\text{update}} \mathbf{W} + \alpha \Delta \mathbf{W}, \quad (4)$$

in which α is a modeller-determined learning rate parameter. This allows for gradual changes in the connection weights and prevents the network from entering a chaotic mode.

Potentially problematic is the fact that equation (3) can produce weights that are larger than the allowable range for the outputs ($w_{ij} < -1$ or $w_{ij} > 1$). For that reason, it is necessary to add an operation that bounds the output. One simple solution is to truncate any illegal output. However, a more elegant solution is to use a filtering function f so that the results are given by $\mathbf{O} \leftarrow f(\mathbf{O})$. Often, f is the sigmoid function, chosen for its mathematical tractability (Hinton 1992).

If the network is multi-layered, the error must be fed backward into the network. First, the weights in the last layer must be corrected. This produces a residual error that is assigned to the previous layer. This procedure is repeated backward until the first layer is reached or when no more residual error remains.

3. Sampling theories

In this section, we briefly present the sampling family of models, or what can be called time-based models. The PRN introduced in the following section is the newest member of this family. This review will allow us to outline these models' common assumptions and their limitations. Figure 2 presents a simplified hierarchy.

At the core of sampling models lies the notion that the senses are sampled and that this process produces a certain activation level. As the sampling process is not perfect, the activation level is a noisy version of the true physical stimulation.

The first sampling model was the Signal Detection Model (SDM, often called the Signal Detection Theory, Green and Swets 1966). It carries the assumption that the senses are sampled only once and the activation level can be either '0' if no stimulation is present or d' if a stimulation is present. The value d' depends on the strength of the physical stimulation and is often called perceptibility. Because of the noise added to the sampling process, there can be overlap between the activation levels for a signal present versus a signal absent. Thus, an optimal decision rule is to use a criterion c to minimize the errors allowed (Dorman and Alf 1969, Coombs *et al.* 1970, Geschelder 1985). Although the quantities d' and c are still often used to describe patterns of errors, the use of SDM as a model of cognition is now marginal. The limitations are that: (i) the SDM samples the senses only once; and (ii) the sampling time is not specified.

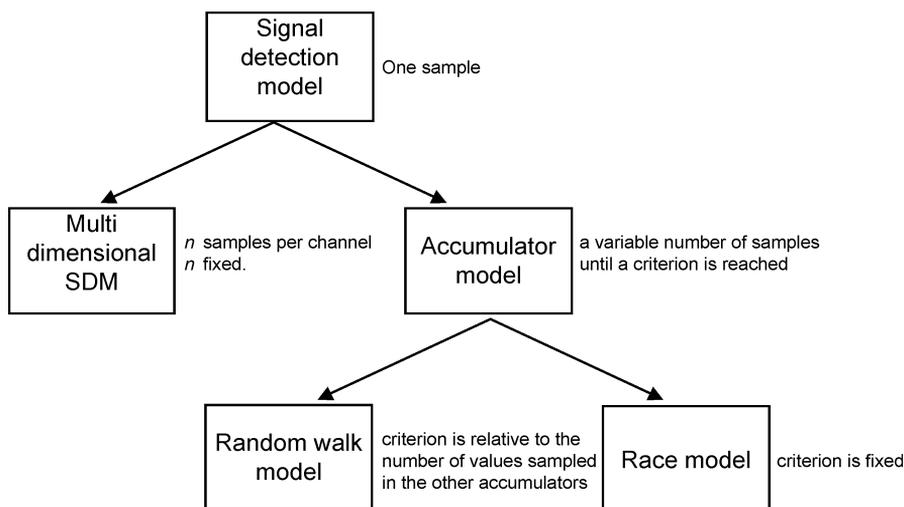


Figure 2. Brief genealogy of the sampling models.

To address this problem, the model was generalized in two ways, yielding what we term the Multidimensional Signal Detection Model (MDSDM). The new core assumption was that n channels could be sampled in parallel and each channel could be sampled m times. The ideas about noisy sampling and decision criteria were preserved. However, the decision was based on m activation levels per channel that had to be compared with a single criterion. Input aggregation was achieved either by calculating the average or by finding the highest level of activation (Zenger and Fahle 1997). These models were useful in describing detection accuracy when the number of items presented increased (Shaw 1978, Eckstein 1998, Eckstein *et al.* 2000), but response time prediction was difficult within this model (Palmer 1998).

The other branch of sampling models, more relevant to our discussion, is grouped under the generic term of accumulator models. These models assume that each input can be sampled many times, the exact number depending on the informativity of the samples. However, they are more complex than the SDM and the MDSDM because there can be noise on the magnitude of the sample, on the time between two samples, or on both. The criterion can be viewed as an objective that states how much activation should be received before an output unit is triggered. Thus, these models are said to accumulate activation and are triggered when an accumulator is full. Because the accumulator will always become full at some point in the presence of noise, two or more accumulators are placed in the network and the first accumulator to be filled makes the decision. Therefore, time is an essential aspect of accumulator models. One important distinction between different varieties of accumulator models is whether the criteria are dependent on other accumulators' level of activation or not. Random walk models (with discrete activation times; see Link 1975, 1992, Ratcliff 1978, Smith 1990) and diffusion models (with continuous activation times; see Diederich 1995, Ratcliff *et al.* 1999) assume that the race is finished if one accumulator exceeds the others by a certain amount. On the other hand, race models assume that the criterion for one accumulator does not depend on the other accumulators' state (see LaBerge 1962, Pike 1973, Meijers and Eijkman 1977, Smith and Vickers 1988, Logan 1988, Van Zandt *et al.* 2000).

One limitation of the accumulator models so far is that they assume a single-channel architecture for each accumulator. If multiple samples are required, they must travel serially (often under the form of spikes of activity). This is a serious shortcoming because it entails either that the responses are based on a disjoint pool of information or that a 'dispatcher' is necessary to select the channels on which information should travel. Clearly, these options are undesirable.

One solution to avoid these limitations is to build a complete network of connections including parallel sources of input that allows the model to select channels autonomously. This is the solution that we wish to present by introducing the PRN.

4. The PRN

The PRN is a new variety of accumulator models within the sampling theory family. Each output unit is postulated to be an accumulator that is triggered when a certain number of inputs is received. For simplicity, the activations are assumed to be discrete. Hence, each activation received is said to fill a slot in the j th accumulator that has a total size of k_j . The time that elapses between two activations is continuous. Processing, as the name of the network implies, involves a competition between the output units: the first accumulator to be filled wins the race and therefore determines the answer.

4.1. Architecture

The architecture of the PRN is very similar to that of the perceptron. There are units connected to the senses on the first layer and one or many layers of units that accumulate the activations from the previous layers. Figure 3(a) illustrates a PRN with two layers of units.

4.2. Input–output representation

Conceptually, the values assigned to the inputs in the PRN differ markedly from those used in the perceptron because they code a different aspect of the input. While the perceptron codes input strength, the PRN codes input arrival times represented in arbitrary units. If the input is immediately active, it is represented by '0'. However, it is possible for an input never to be activated. In this case, it would hypothetically react after an infinite amount of time ' ∞ '. This transition from $\{0, 1\}$ in the perceptron to $\{\infty, 0\}$ in the PRN may initially seem counter-intuitive. Yet this type of representation clearly emphasizes the PRN's focus on the temporal aspect of the activation. This point is illustrated schematically in figure 3(b).

The connections are also considered with respect to time. An input that is strongly related to a certain response may be postulated to fill immediately one slot of the corresponding accumulator. Likewise, it can be supposed that an uninformative input will never reach the accumulator even though it may be active. One way to implement these assumptions is to introduce delays in the connections. A 'slippery' connection is one that does not delay the passage of information. To use the familiar expression, this connection will be said to be '*On*'. Similarly, a connection that offers 'resistance' will delay the passage of information indefinitely and will be said to be '*Off*'.

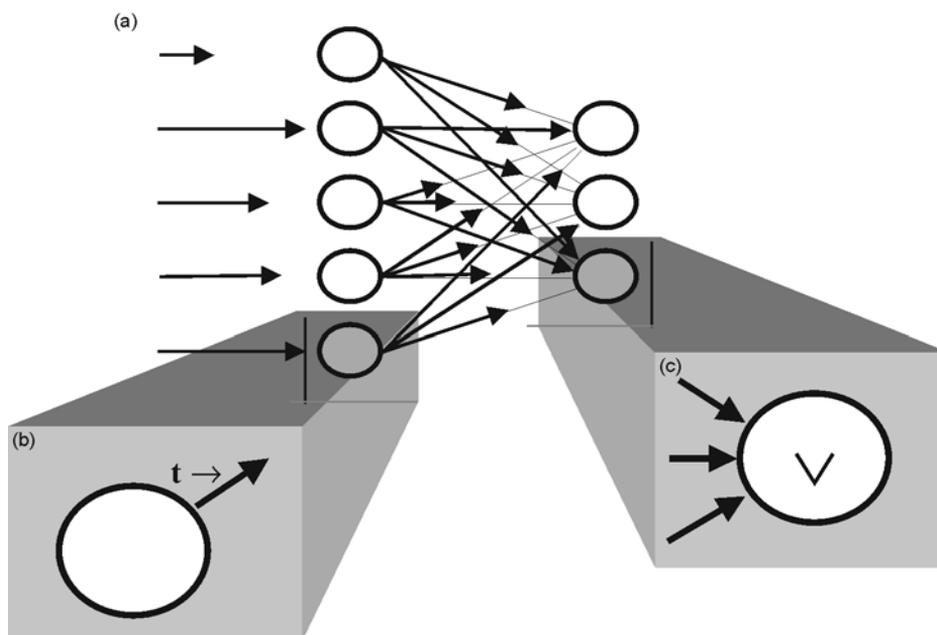


Figure 3. Representation of a simple time-based network, the PRN, with two layers of units (a) Architecture and input–output visual representation. (b) Delays as an amount of time. (c) The integration rule using a minimum.

In terms of passage times, the first case introduces a delay of zero and the second case introduces an infinite delay. Thus, the coding $\{\infty, 0\}$ is also used to express the state of the connections.

4.3. Integration rule

Let d_{ij} be the delay introduced by information travelling from the input i to the output j and $\mathbf{D} = \{d_{ij}\}$ be the full matrix of connections.¹ Further, let $\mathbf{A} = \{a_i\}$ be the moment at which the i th input becomes active. An accumulator with k slots will react when k activations are received. Each input becomes active at a time a_i and is delayed in the connections by a time d_{ij} so that it reaches the j th accumulator after a total time of $a_i + d_{ij}$. All the inputs will reach the j th accumulator after the times given by a list $\{a_i + d_{ij}\}$ for all i . The accumulator will be triggered as soon as the k_j th fastest signal is received. The time for the k_j th fastest is determined by the k_j th smallest time in the list $\{a_i + d_{ij}\}$. Thus, the decision time for the j th output is given by:

$$o_j = \bigvee_{i \in \text{input}}^{k_j} a_i + d_{ij}, \quad (5)$$

in which \bigvee^{k_j} locates the k_j th smallest element of the list. The central role of the minima is illustrated schematically in figure 3(c).

Equation (5) is functionally very similar to equation (1). For each output, couples taken from the inputs and the connections are joined, and the resulting list is aggregated into a single summary value. However, the PRN does not use the same operations as the perceptron to accomplish this goal. First, an addition is used to join pairs of values instead of a multiplication. Second, the minimum is used to execute the aggregation of the list instead of the summation of activation used in connectionist networks. These operations over all the outputs may be summarized using the following matrix notation:

$$\mathbf{O} = \mathbf{A}_{(\bigvee)}^+ \mathbf{D}, \quad (6)$$

in which $(\bigvee)^+$ represents a redefined inner product that shows explicitly the use of an addition as a joining operator and the minimum for the aggregation operation. We shall denote the redefined inner product more compactly with τ .

Here, \mathbf{O} is a vector that contains the times at which each of the outputs fired. Once again, note that only the fastest output matters. Nevertheless, the vector contains the information of all the participating output units. Thus, this representation allows for the evaluation of phenomena such as confidence levels without the addition of extra parameters (see, e.g. Huber and Cousineau, submitted).

It can also be appreciated that the redefined inner product $(\bigvee)^+$ has an elegant relationship with the input–output representation values $\{\infty, 0\}$. Indeed, the value 0 is neutral with respect to addition and ∞ is neutral with respect to minimum. Therefore, a matrix with all values set at ∞ except for the main diagonal set at 0 would yield a neutral matrix with respect to τ . By analogy to work on linear algebra, we call this matrix the redefined identity matrix, which we denote $\tilde{\mathbf{I}}$ so that $\tilde{\mathbf{I}} \tau \mathbf{A} = \mathbf{A} \tau \tilde{\mathbf{I}} = \mathbf{A}$.

One final attractive quality of the PRN related to its integration rule is its capacity to predict response times when noise is present. This model only allows noise that is positive. That is, noise simply creates further delays in the time taken for the activation to reach the accumulators. Using a proof developed by Cousineau *et al.* (2002), which

can be termed the ‘Extreme Limit Theorem’, it is possible to infer the distribution of these finishing times. Under very general conditions (satisfied here), the theorem shows that the distribution of the k_j th fastest activation follows a Weibull distribution. This distribution is generally positively skewed and is congruent with response time data (Luce 1986).

4.4. Learning rule

The PRN requires a supervised network with two learning rules: one to speed up the connections that convey diagnostic information and one to update the sizes of the accumulators. The corrections are based on the error between the actual outputs and the desired output \mathbf{E} . In its simplest form, the desired output consists of a 0 for the output that should fire first and ∞ for the outputs that should not fire at all.

The learning rule that we present here is called, by analogy with equation (3), the redefined Delta rule, or $\tilde{\Delta}$ rule. The delay between the input unit i and the output unit j must change in proportion to the error (defined here in terms of precocious responses) and the times at which the inputs were available:

$$\Delta d_{ij} \propto a_i + (e_j - o_j),$$

in which e_j is the expected response time of unit j . In matrix form, we may write:

$$\Delta \mathbf{D} \propto \mathbf{A})+(\mathbf{E} - \mathbf{O}), \quad (7)$$

in which $)+($ is a redefined outer product showing explicitly the use of addition to join pairs of values. The matrix is updated by determining the shortest delay between the old delay and the corrected delay:

$$\mathbf{D} \xleftarrow{\text{update}} \mathbf{D} \vee \mathbf{D} + \varphi \Delta \mathbf{D}, \quad (8)$$

where φ is the learning rate parameter for the delay. Comparing equations (3) and (4) with equations (7) and (8), respectively, note that wherever a multiplication was used in the previous equations, an addition is used, and wherever a sum was used, a minimum is used. Thus, the changes made in the integration rule, going from a standard inner product (\sum^{\times}) to a redefined one (\vee^+), are mirrored by equivalent changes in the learning rule.

Accumulator sizes must also be changed throughout learning. Let \mathbf{K} be a vector containing all the accumulator sizes $\{k_j\}$ for the j th accumulator. In case of an error, the size of the output that missed the response is updated using this rule:

$$\mathbf{K}_{\text{missed}} \xleftarrow{\text{update}} \mathbf{K}_{\text{missed}} + \omega \text{Sign}(\#\mathbf{A} - \mathbf{K}_{\text{missed}}),$$

where $\text{Sign}(x)$ returns $+1$, -1 or 0 depending on whether $x > 0$, $x < 0$, or $x = 0$, respectively. $\#\mathbf{A}$ returns the number of inputs that were active at the time the error was detected. In other words, the size of an accumulator is increased if its number of slots is smaller than the number of input it received. Finally, ω is the learning rate parameter for the changes in the accumulator sizes.

We believe that our formal description of the PRN is very promising from a modelling point of view as it shows unequivocally that these time-based models of cognition may be implemented as easily as the current strength-based networks while avoiding the

limitations of previous accumulator models. To illustrate this last point, we now present a simulation in which both the perceptron and PRN solve a simple problem.

5. An example

In this section, we use the perceptron and the PRN to simulate human performances in an identification task involving letters (conducted by Larochelle *et al.*, submitted). The stimuli were eight lowercase letters {n, h, b, u, y, q, p, d}. To minimize the number of possible features, the ‘y’ was drawn as a reversed and inverted ‘h’ (see Figure 4). It turned out that the participants found these stimuli difficult to search for, even after extended practise with consistent mapping (Shiffrin and Schneider 1977).

Our goal in presenting the following simulations is not to set up a head-to-head competition between two models. Rather, we wish to show that race models can simulate human cognition as well as connectionist networks and that their time-based representation provides new insights in understanding old problems.

5.1. Learning with the perceptron

A perceptron with two layers of units was trained to identify the stimuli of figure 4. Each column in figure 4 coded one stimulus. For the simulation, the ‘+s’ were replaced by ‘1s’ and the empty locations were filled with ‘0s’ in order to respect the input–output representation of the strength-based network. Before training, the connections were set at random. Uniform random values between -0.5 and $+0.5$ were used. The learning rate parameter was set at $\alpha = 1$. The task of the network was to decide which letter had been presented. The features were presented as input and the network responded with the activation of one of the eight outputs. The architecture of the network was composed of five input units and eight output units. The connections were contained in a 5×8 matrix. The network was trained for 300 epochs. Each stimulus was presented once in a random order within an epoch.

Figure 5(a) shows the errors that the network produced. They were measured using the root mean square deviation (RMSD) between the expected and observed output across all eight outputs. If we assume that a RMSD below 0.3 indicates successful learning, the perceptron took an average of 165 epochs to learn to identify the eight letters when the simulation was replicated 1000 times.

We have illustrated the perceptron’s solution to the identification problem in figure 5(b) using a bubble plot. The bubble sizes are directly proportional to the strength of the associations. This bubble plot was filtered in figure 5(c) to show only the largest weights in

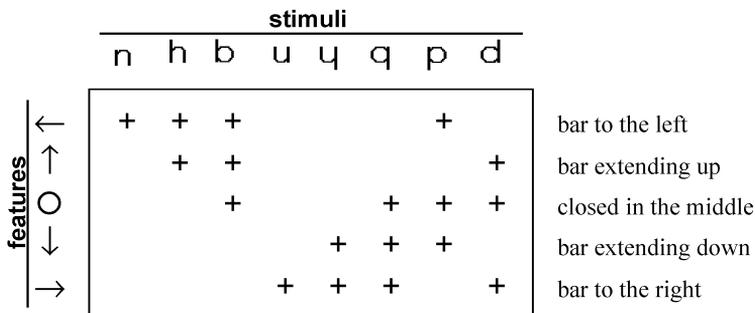


Figure 4. Stimuli used in the example and their featural composition. A ‘+’ indicates the presence of the feature, an empty location its absence.

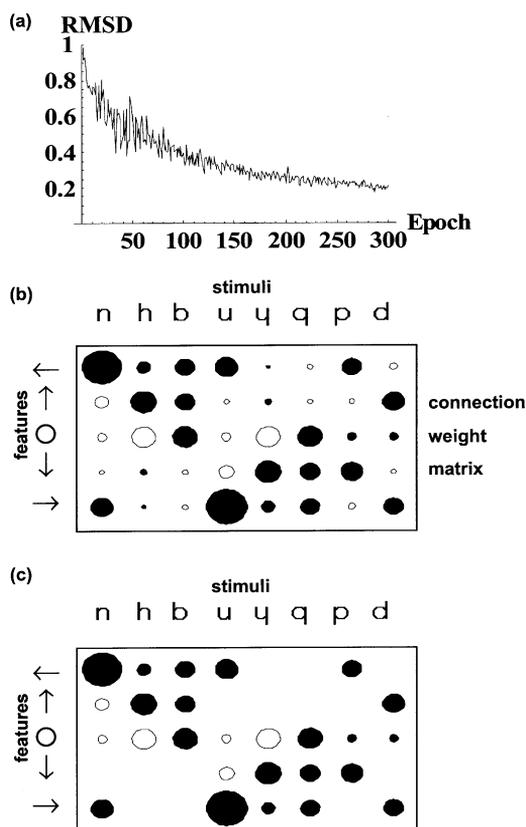


Figure 5. Example of a learning session with the perceptron. (a) Errors made by the network through training epochs measured by the root mean square deviation between the observed output and the desired output. One epoch represents a cycle through the eight stimuli, in a random order. (b) The solution found, in terms of the connection weights in the bubble plot. Large bubbles indicate strongly associated connections whereas small bubbles indicate weakly associated connections. Closed bubbles indicate positive weights and open bubbles, negative weights. (c) Same as previously but filtered to show only the relevant weights.

order to evaluate learning better. We can see from the bubble plot why the stimuli are difficult to discriminate. For example, the letter ‘n’ is totally embedded within the letter ‘h’, which is itself embedded within the letter ‘b’. Because there is no feature to signal the absence of a bar up, the ‘n’ output must be strongly associated with the bar to the left. However, if a bar to the left is presented, it does not necessarily imply the presence of an ‘n’. Thus, the network had to develop inhibitive connections. Finally, as is always the case when dealing with simple connectionist networks, there is no straightforward way of predicting any kind of response time data.

5.2. Learning with the PRN

The PRN was also trained to identify the stimuli in figure 4. All ‘+’s were replaced with ‘0s’ (present) and the other locations were replaced with ‘∞s’ (absent). The architecture of the network was identical to that of the perceptron presented in the previous section. The PRN was initialized with random delays large enough that they could be reduced

through training. Random uniform numbers between 100 and 110 were used to serve as arbitrary units of time. Furthermore, all the thresholds were set to 1, the lowest possible value. We set the learning rate parameters φ to 1.1 and ω to 1. These numbers were arbitrary except that ω had to be smaller or equal to 1 so that all successive threshold sizes could be tested by the learning rule. The network was trained for 300 epochs through all the eight inputs. The stimuli were presented randomly within each cycle.

Figure 6(a) shows the percentage of error throughout learning. It was unnecessary to plot all 300 epochs because the PRN learned the problem quickly. In fact, in over 1000 replications, the networks never took more than eight epochs to identify all eight stimuli without error. Notice that this is much faster than the average 165 epochs that the perceptron took.

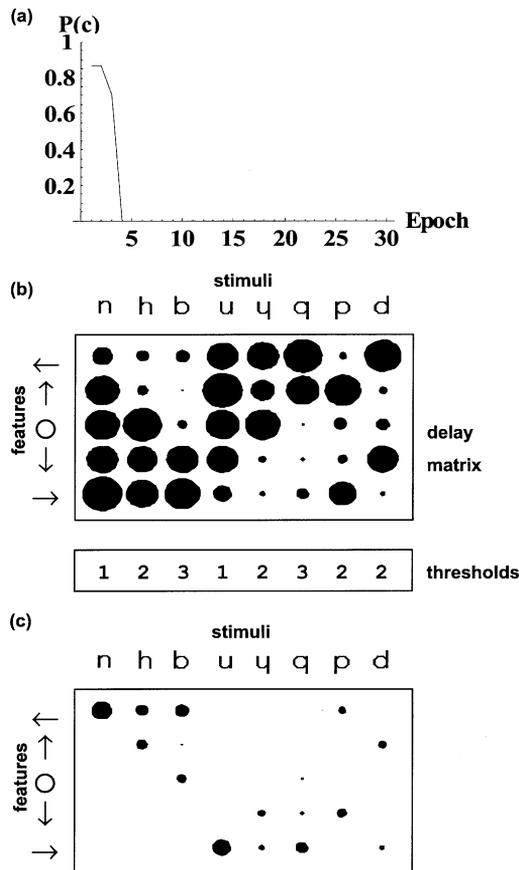


Figure 6. Example of a learning session with the PRN. (a) Errors made by the network through training epochs measured by the percentage of errors. One epoch represents a cycle through the eight stimuli, in a random order. (b) The solution found, in terms of the connection delays in the bubble plot and the corresponding threshold sizes beneath the bubble plot. The smaller bubbles represent the shortest delays (an average of 94 units of time) and the largest, the longer delays (105 units of time on average). (c) Same as previously but filtered to show only the relevant delays.

Figure 6(b) shows the delay matrix of the network illustrated using a bubble plot. The bubble sizes are proportional to the duration of the delay for any given connection. The large bubbles represent units that have no chance of triggering a response because they are the slowest. Consequently, the small bubbles represent units that are highly involved in triggering a response because they are the fastest. Once again, we filtered the bubbles in figure 6(c) to highlight the important connections. As can be observed, the solution is similar to that of the perceptron. However, the PRN did not struggle as much as the perceptron with the embedded letters ‘n’ and ‘h’ because its time-based representations solved the problem by simply waiting. Indeed, the network determined that a given stimulus was an ‘n’ by monitoring whether the ‘h’ and ‘b’ units had answered first. When these units did not answer, the network concluded that an ‘n’ had to have been presented. By contrast, the perceptron had to develop inhibitive associations. Thus, the PRN provides an elegant solution to this seemingly paradoxical situation in which simpler stimuli (in terms of number of features) actually produce longer response latencies. This type of counter-intuitive result has been obtained in several studies, such as the word superiority effect (Rumelhart and Siple 1974), the triple conjunction search (Fournier *et al.* 1998) and the redundant-target procedure (Miller 1982).

Finally, for the letters ‘p’ and ‘d’ which are composed of three features, the accumulator sizes stabilized at two because two features (the left and down features or the right and up features, respectively) are sufficient to identify them unambiguously. The PRN thus reduced the amount of information manipulated, a result also compatible with some empirical findings (Haider and Frensch 1996, 1999). Because there is no noise in this model, the output response times are deterministic and have no variability. Yet, each stimulus settled at different response times, the one containing less information having a lower priority. This allows for an ordering of response times such as $RT_{\text{‘n’}} \approx RT_{\text{‘u’}} \gg RT_{\text{‘h’}} \approx RT_{\text{‘y’}} \gg RT_{\text{‘b’}} \approx RT_{\text{‘q’}}$. Thus, *RT* ordering should be preserved even in the presence of variability and noise (Cousineau *et al.* 2002; Cousineau 2002).

Before concluding, a note on the rate of learning is in order. As we saw, the PRN learned 20 times faster than the perceptron. Yet, we do not believe that learning speed *per se* is a major issue here. Indeed, it is quite conceivable that more sophisticated connectionist networks would have solved our identification task more quickly. The promising observation is that the PRN, equipped with an architecture and a learning rule of comparable complexity to that of the perceptron, found a solution because it formed a localist representation at the output level (as opposed to a distributed representation; Page 2000). Thus, the winner-take-all decision rule adopted by the PRN might produce a more efficient error-driven correction when used in conjunction with a localist representation.

6. Conclusion

An essential feature of the connectionist network family is the use of the standard inner product that integrates the inputs using a weighted sum. This is present in the perceptron and in other models such as Anderson’s autoassociator, the Boltzmann networks and the Hopfield networks (Freeman 1994), one exception being Kohonen’s (1984) self-organizing map. Then, we showed that it was possible to modify this core feature without affecting the network’s ability to be operational. Two important steps had to be taken though. The first and most profound change was a new way to represent the inputs and the connections in the network. Rather than viewing the values in terms of associative strengths, we decided to view the values as units of time. Second, to accommodate our new time-based representation of the network, the learning rule’s constituent operators were modified. The joining operator was changed for an addition and the aggregate

function was changed for a minimum. These changes, while keeping the newly created network in line with previous networks from a mathematical perspective, created a race model that has the ability to simulate some aspects of human cognition. The establishment of this link between the two families of models is our key result.

We believe that calling this new rule the redefined Delta rule is most appropriate because it preserves the spirit of the original Delta rule. That is, it allows the network to solve problems by reducing the importance of connections that contribute most heavily to the error in the output. More importantly, this is achieved without any intervention on the part of the modeller.

From this simple model, future work is possible following different directions. One avenue is to examine whether the time-based network can be extended to multiple-layer architectures. A second avenue is to explore the possibility of using a redefined Hebbian learning rule in the context of an autoassociator race network. Eventually, it will be interesting to determine whether this redefined rule is based on gradient descent. At present, this question cannot be answered, as the redefined inner product is very different from anything that is used in linear algebra. Nevertheless, we believe that this issue is not a pressing one and that, given further analysis, it will be resolved. Our experience with the PRN shows that given a sensible architecture, integration rule and learning rule, many different types of networks can produce fast and reliable learning. In addition to the PRN, we are currently exploring other mutated networks.

One simplifying assumption that was made is that only one input can fill one slot when it reaches the output (discrete evidence). This move is open to criticism if one believes that it lacks biological validity. One possible solution would be to adopt a continuous form of coding (Smith and Vickers 1988). However, this would complicate things a lot because the ‘Extreme Limit Theorem’ would no longer be applicable (Cousineau *et al.* 2002). An indirect solution is to assume a lot of redundancy in the connection paths. If we assume that a single input can travel through hundreds of connections (as is presumably the case in the brain) and if we assume that the accumulator sizes are large, then we would, once again, obtain a quasi-graded representation of the input. This massive-redundancy approach (first proposed in Cousineau 2001) preserves the applicability of the Extreme Limit Theorem and might be biologically plausible. For example, Thorpe and Gautrais (1999) state that it was:

‘... recently demonstrated that the human visual system can process previously unseen natural images under 150 ms... To reach the temporal lobe in this time, information from the retina has to pass through roughly ten processing stages. If one takes into account the surprisingly slow conduction velocities of intracortical axons, it appears that the computation times within any cortical stage will be as little as 5 ms.’ (Thorpe and Gautrais, 1999, p. 1).

Although it may be argued that our time-based approach and the race between signals it favours are both plausible and appealing, we are not suggesting that strength-based approaches should be dismissed at this point as they seem to account naturally for several empirical findings. For instance, it has been shown that the strength of activation of a single neuron diminishes with time (Tsodyks and Markam 1997) and that this refractory period may have an important role in priming studies (Huber and O’Reilly 2003). Yet, connectionist networks also need subsidiary assumptions to account for this type of finding.² In any case, evaluating connectionist models strictly from the perspective of biological plausibility has not been the most productive endeavour as proponents of different models can always show that a given assumption does not clearly map on to our present knowledge about the brain. Here, the most salient example may be the attacks on the original Delta rule (O’Reilly 1996).

Rather, our goal was to show that there is a viable alternative to connectionist networks for those who are in the business of simulating human cognition in order to gain a better understanding of the mind. Furthermore, we believe that the PRN's time-based representations have the potential to generate exciting new explanations for a wide variety of tasks, such as identification. In particular, the parallel race network seems to offer an ideal framework for the simultaneous modelling of error rates and response times. Thus, this is an invitation to explore the possibilities that a shift in perspective can offer.

Acknowledgements

We would like to thank Robert Proulx for his comments on an earlier version of this text and for his very stimulating teaching that made this research possible. We also want to thank two anonymous reviewers for their comments. A web version of the PRN along with source code in Java and in *Mathematica* is available at <http://MAPAGEWEB.UMontreal.CA/cousined/papers/17-PRN>. This research was supported in part by funding from the Fonds de Recherche sur la Nature et les Technologies du Québec and the Conseil de Recherches en Sciences Naturelles et en Génie du Canada.

Notes

1. These connections code the waiting time and so we thought of calling them 'waits'. Although elegant, this notation might be confused with 'weights' in spoken discourse.
2. Note that the refractory period could be modelled by reducing the probability that a redundant unit will fire for a given amount of time once it has fired. This would introduce a longer delay in the transmission of activation and is compatible with our time-based perspective.

References

- Anderson, J. A., 1995, *An Introduction Neural Networks* (Cambridge MA: MIT Press).
- Coombs, C. H., Dawes, R. M., and Tversky, A., 1970, *Mathematical Psychology: An Elementary Introduction* (Englewood Cliffs, NJ: Prentice-Hall).
- Cousineau, D., 2001, Redundancy conjecture and super-capacity rate of increase. *Society for Mathematical Psychology Annual Meeting*, Providence, RI, July.
- Cousineau, D., submitted, Merging race models and adaptive networks: a parallel race network. *Psychonomic Bulletin & Review*.
- Cousineau, D., Goodman, V., and Shiffrin, R. M., 2002, Extending statistics of extremes to distributions varying on position and scale, and implication for race models. *Journal of Mathematical Psychology*, **46**, 431–454.
- Cramér, H., 1946, *Mathematical Methods of Statistics* (Princeton: Princeton University Press).
- Diederich, A., 1995, Intersensory facilitation of reaction time evaluation of counter and diffusion coactivation models. *Journal of Mathematical Psychology*, **39**: 197–215.
- Dorman, D. D., and Alf, E. Jr., 1969, Maximum likelihood estimation of parameters of signal-detection theory and determination of confidence intervals—rating-method data. *Journal of Mathematical Psychology*, **6**: 487–496.
- Eckstein, M. P., 1998, The lower visual search efficiency for conjunctions is due to noise and not serial attentional processing. *Psychological Science*, **9**: 111–118.
- Eckstein, M. P., Thomas, J. P., Palmer, J., and Shimozaki, S. S., 2000, A signal detection model predicts the effects of set size on visual search accuracy for feature, conjunction, triple conjunction, and disjunction displays. *Perception and Psychophysics*, **62**: 425–451.
- Feller, W., 1957, *An Introduction to Probability Theory and Its Application*, Vol. 2, 2nd ed (New York: John Wiley).
- Fournier, L. R., Eriksen, C. W., and Bowd, C., 1998, Multiple-feature discrimination faster than single-feature discrimination within the same object? *Perception and Psychophysics*, **60**: 1384–1405.
- Freeman, J. A., 1994, *Simulating Neural Networks with Mathematica* (Reading MA: Addison-Wesley).

- Geschelder, G. A., 1985, *Psychophysics Method, Theory and Application* (New York: Laurence Erlbaum).
- Golden, R. M., 2001, Statistical hypothesis testing in complex behavioral simulation models. *Society for Mathematical Psychology Annual Meeting*, Providence, RI, July.
- Green, D. M., and Swets, J. A., 1966, *Signal Detection Theory and Psychophysics* (New York: John Wiley).
- Haider, H., and Frensch, P. A., 1996, The role of information reduction in skill acquisition. *Cognitive Psychology*, **30**: 304–337.
- Haider, H., and Frensch, P. A., 1999, Eye-movement during skill acquisition: More evidence for the information-reduction hypothesis. *Journal of Experimental Psychology: Learning, Memory and Cognition*, **25**: 172–190.
- Hinton, G. E., 1992, How neural networks learn from experience. *Scientific American*, September, 145–151.
- Hinton, G. E., and Sejnowski, T. J., 1998, *Unsupervised Learning: Foundations of Neural Computation* (Cambridge MA: Bradford Books).
- Huber, D. E., and Cousineau, D., submitted, A race model of perceptual forced choice reaction time. In *Proceedings of the Cognitive Science Conference*, Boston, MA, June.
- Huber, D. E., and O'Reilly, R. C., in press, Persistence and accommodation in short-term priming and other perceptual paradigms. Temporal segregation through synaptic depression. *Cognitive Science*.
- Kohonen, T., 1984, *Self-organization and Memory*, **27**(3), 2003 (Berlin: Springer).
- LaBerge, D. A., 1962, A recruitment theory of simple behaviour. *Psychometrika*, **27**: 375–396.
- Larochelle, S., Lefebvre, C., and Cousineau, D., submitted, Determinants of automaticity in visual-memory search tasks: pitting target–distractor similarity against stimulus–response mapping. *Perception and Psychophysics*.
- Link, S. W., 1975, The relative judgment theory of two choice response time. *Journal of Mathematical Psychology*, **12**: 114–135.
- Link, S. W., 1992, Imitatio Estes: stimulus sampling origin of Weber's law. In: A. L. Healy, S. M. Kosslyn and R. M. Shiffrin (eds) *From Learning Theory to Connectionist Theory: Essays in Honor of William K. Estes* (Hilldale NJ: Lawrence Erlbaum), pp. 97–113.
- Logan, G. D., 1988, Toward an instance theory of automatization. *Psychological Review*, **95**: 492–527.
- Luce, R. D., 1986, *Response Times, Their Role in Inferring Elementary Mental Organization* (New York: Oxford University Press).
- McClelland, J. L., and Rumelhart, D. B., 1988, *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises* (Cambridge MA: MIT Press).
- Meijers, L. M. M., and Eijkman, E. G. J., 1977, Distributions of simple RT with single and double stimuli. *Perception and Psychophysics*, **22**: 41–48.
- Miller, J., 1982, Divided attention: evidence for coactivation with redundant signals. *Cognitive Psychology*, **14**: 247–279.
- Minsky, R., and Papert, S., 1969, *Perceptrons: An Introduction to Computational Geometry* (Cambridge MA: MIT Press).
- O'Reilly, R. C., 1996, Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm. *Neural Computation*, **8**: 895–938.
- Page, M., 2000, Connectionist modelling in psychology: a localist manifesto. *Behavioral and Brain Sciences*, **23**: 443–512.
- Palmer, J., 1998, Attentional effects in visual search: relating search accuracy and search time. In R. D. Wright (ed.) *Visual Attention* (New York: Oxford University Press), pp. 348–388.
- Pike, R., 1973, Response latency models for signal detection. *Psychological Review*, **80**: 53–68.
- Ratcliff, R., 1978, A theory of memory retrieval. *Psychological Review*, **85**: 59–108.
- Ratcliff, R., Van Zandt, T., and McKoon, G., 1999, Connectionist and diffusion models of reaction time. *Psychological Review*, **106**: 261–300.
- Rosenblatt, F., 1961, *Principles of Neurodynamics: Perceptrons and the Theory of the Brain Mechanisms* (Washington DC: Spartan).

- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning internal representation by error propagation. In D. E. Rumelhart, J. L. McClelland and the PDP Research Group (eds) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Cambridge MA: MIT Press), pp. 318–362.
- Rumelhart, D. E., and Siple, P., 1974, Process of recognizing tachistoscopically presented words. *Psychological Review*, **81**: 99–118.
- Shaw, M. L., 1978, A capacity allocation model for reaction time. *Journal of Experimental Psychology: Human Perception and Performance*, **4**: 586–598.
- Shiffrin, R. M., and Schneider, W., 1977, Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review*, **84**: 127–190.
- Smith, P. L., 1990, A note on the distribution of response times for a random walk with Gaussian increments. *Journal of Mathematical Psychology*, **34**: 445–459.
- Smith, P. L., and Vickers, D., 1988, The accumulator model of two-choice discrimination. *Journal of Mathematical Psychology*, **32**: 135–168.
- Thorpe, S. J., and Gautrais, J., 1999, *Rapid Visual Processing Using Spike Asynchrony* (Toulouse. Centre de Recherche Cerveau & Cognition).
- Townsend, J. T., and Ashby, F. G., 1983, *Stochastic Modelling of Elementary Psychological Processes* (Cambridge: Cambridge University Press).
- Tsodyks, M. V., and Markam, H., 1997, The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Science*, **94**: 719–723.
- Usher, M., and McClelland, M. L., 2000, On the time course of perceptual choice: the leaky competing accumulator model. *Psychological Review*, **108**: 550–592.
- Van Zandt, T., Colonius, H., and Proctor, R. W., 2000, A comparison of two response-time models applied to perceptual matching. *Psychonomic Bulletin & Review*, **7**: 208–256.
- Widrow, B., and Hoff, M. E., 1960, Adaptive switching circuits. *Institute of Radio Engineer, Western Electronic Show and Convention, Convention Record*.
- Zenger, B., and Fahle, M., 1997, Missed targets are more frequent than false alarms: a model for error rates in visual search. *Journal of Experimental Psychology: Human Perception and Performance*, **23**: 1783–1791.