MERGING RACE MODELS AND ADAPTIVE NETWORKS:

A PARALLEL RACE NETWORK

Denis Cousineau

Université de Montréal

Running head: A parallel race network

<<Submitted manuscript; please do not circulate. Version 6.1 of March 27th, 2002>>

For correspondence:
Denis Cousineau
Département de psychologie
Université de Montréal
C. P. 6128, succ. Centre-ville
Montréal (Québec) H3C 3J7 CANADA

E-mail:denis.cousineau@umontreal.ca

ABSTRACT

This paper presents a generalization of the race models involving parallel channels. Previous versions of this class of models involved only serial channels, either dependent or independent. Further, concrete applications of these models did not involve variability in the accumulator size (Logan, 1988) or were based on a specific distribution (Bundesen, 1990). We show that the distributions of response times predicted by the parallel race model can be solved analytically to yield a Weibull distribution using asymptotic (large *n*) theories. The model can be manipulated to predict the effects of reward on ROC curves and speed-accuracy decomposition. However, the major contribution of this paper is the implementation of a learning rule that enables networks based on a parallel race model to learn stimulus-response associations. We call this model a parallel race network. Surprisingly, with this architecture, a parallel race network learns a XOR without the benefits of hidden units. The model described here can be seen as a reduction of information system (Haider and Frensch, 1996) and is compatible with a priority learning system. An emergent property of this model is seriality: in some conditions, responses are performed with a fixed order although the system is parallel. Finally, the mere existence of this supervised network demonstrates that networks can perform cognitive processes without the weighted sum metric that characterizes connectionist models.

Introduction

The aim of this text is to expand our knowledge of the race models by showing that there exist similarities between the race models and the connectionist networks. The major objective of this paper is to introduce a learning rule for race models. With it, race models and connectionist models can be compared on similar grounds. We will not undertake this comparison in this paper, its purpose being just to describe the parallel race network. To this end, we will first cast the race model into the form of a network of connections. A second objective of this paper is to describe the response times (RT) predicted by race models. Many predictions will be done in terms of the distribution of response times (averages can easily be derived from them). Finally, networks differ on what is being transported by the connections (sometimes called the channels hereafter). We show that race models don't transport strength of activation. One important feature that we will address is redundancy of the channels. Redundancy turned out to be necessary to simplify the mathematics of the parallel race model.

Overview of the race models

Accumulator models and random walk models are generalization of the Signal Detection Theory (SDT) because they share the same coding in the form of samples coming from the senses. However, instead of taking a single sample from the stimuli, these two classes of models can take an arbitrary number of samples, this number changing from trial to trial.[1] In these models, the samples can be evidence for one response or an alternative response. In some variants, the samples can also be evidence for both responses, or neither. Following Smith and Vickers (1988), the distinction between random walks and race models pertains on the evidence

accumulation process. For the former, the accumulation process is not independent because an evidence for one response implies a reduction of the amount of evidence for the alternative response. For the latter, the evidence accumulations are done in independent accumulators. However, the first accumulator filled triggers a response. Thus, the name of race model.[2]

The class of accumulator models can be further broke down by whether the evidence collected are discrete, called simple accumulator models by Luce (1986) or continuous, called strength accumulator model. Examples of simple accumulator models are given by Audley and Pike (1965) where the sampling process takes a fixed amount of time, and by Pike (1973), where the time between two samples is continuous. An example of strength accumulator is given by Smith and Vickers (1988) where the time between two samples is fixed.

Another important distinction between the race models is whether the channels bringing evidence to the accumulator are dependent or independent (Meijers and Eijkman, 1977). We describe one accumulator model, the Poisson race model because it recapitulates in a nutshell the following sections of this paper.

The Serial Poisson Race Model

One of the first quantitative description of the Poisson race models was described by Pike (1973, 1965; also see Van Zandt, Colonius and Proctor, 2000).

Architecture: This model is essentially composed of a single channel that encodes information (under the form of spikes of activation) that is brought to a unit accumulating the spikes (an accumulator) containing K slots (see Figure 1, panel a). As soon as the accumulator is filled, a response is emitted. This model is serial (a sequential sampling model) because the spikes travel one at a time. It is also based on a race model since a decision criterion can be formulated by a

rule akin to "**As soon as** you have *K* evidences, respond". In other word, the accumulator is a hard threshold unit.

Insert Figure 1 about here

Response times: From this, it follows that the response times of the output **o** is given by:

$$\boldsymbol{o} = \sum_{k} \boldsymbol{t}_i \ .$$

where $\boldsymbol{t}_i$ is the time separating spike $i - 1$ from spike i.

Distributions: In order to derive the distribution of response times for this model, it is necessary to add one more assumption: how much time elapses between two spikes of activation. A simple assumption is to postulate that a spike results from a Poisson process so that the time $\boldsymbol{t}$ between two spikes are distributed with an Exponential distribution function $\mathcal{T}$ with a certain rate θ.

The corresponding distribution of output response times is given by:

$$\boldsymbol{O} = \underset{K}{*}\, \mathcal{T}$$

that is, a convolution of *K* Exponential distributions. This problem is easily solved and $\boldsymbol{O}$ turns out to be a Gamma distribution with rate parameter θ and shape parameter *K* (Townsend and Ashby, 1983, Luce, 1986).

Sequential vs. parallel sampling models

Due to this serial architecture, the accumulators are located after a bottleneck. All the evidence must travel through the same channel, thus involving cumulative delays. Yet, as noted by Thorpe and Gautrais (1999), there might not be enough time for information jam: "We recently demonstrated that the human visual system can process previously unseen natural

images in 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 time within any cortical stage will be as little as 5 ms." (p. 1). Introducing a parallel architecture eliminates the bottleneck. This was one of the motivations of the model described in this paper since it is a fully parallel race model.

Classic connectionist networks

Parallel networks, in their simplest form, are composed of a network of fully connected input and output units. In some models, extra units are added in what is called a hidden layer (McClelland and Rumelhart, 1988) whereas in other variants, the input units are the same as the output units (Caudill and Butler, 1992). The output units must make decisions on the basis of the activation at the input level using some integration rule specifying how to ponder them. Thus, specifying an integration rule implicitly specifies the nature of the information flowing through the channels.

In the early connectionist networks (Widrow-Hoff, 1960, Rosenblatt, 1962), the inputs were assumed to reflect the strength of activation. The channels reflected, loosely speaking, the relevance of the input with respect to the output. The decision is thus a sum of the input's activation weighted by its relevance. The connections came to be labeled the weights. Panel b of Figure 1 illustrates this case. Past the weighted sum, some researchers added a threshold to the output. They were either soft, using a sigmoid function (McClelland and Rumelhart, 1986) or hard, with an absolute threshold (Arguin and Bub, 1995, Page, 1999). Learning in such network was thus to reduce the importance of non-diagnostic inputs.

The Perceptron

The Perceptron is a classic example of a feed-forward supervised learning network.

Architecture: The network is composed of two layers of units, the input units and the output units, which are fully connected.

Response pattern: In this model, the outputs result from a weighted average of the input. The state of the output units can be given by a vector $\boldsymbol{O}$ whose relation to the input is given by

$$\boldsymbol{O} = \boldsymbol{I} \,.\, \boldsymbol{W}$$

where the dot represents the inner product. Contrary to accumulator model, this is not a race model since the overall pattern of output is the response. This is labeled a "distributed representation network" (Page, 1999).

Learning rule: The learning rule consists in reducing the error between the network's pattern and the desired pattern. This is done in Perceptrons using a simple rule correcting mostly the weights that were the most responsible for an error by a decrement Δw using the formula:

$$\Delta w_{ij} = \beta (d_j - o_j) I_i$$

where Δw_{ij} is the strength of the connection relating input i to output j, I_i is the i^{th} input, and d_j is the desired output on unit j. β is the learning rate, a free parameter.

One utility of integrating a learning mechanism in a network model is to predict learning curves over time (Logan, 1988, Indow, 1993). More importantly, it can be used to reduce the number of free parameters. For example, consider a random walk model defined by boundaries and drift rates (e.g. Ward and McClelland, 1996, Ratcliff, Van Zandt and McKoon, 1999). If these parameters could be learned by exposure to stimuli, they would no longer be free to vary.

The only free parameters remaining would correspond to a description of the random fluctuations in stimulus sampling (i. e., variability).[3]

Response times distributions in classic neural networks

Analyses of response time (RT) distributions in speeded tasks show that they are lower bounded and always positively skewed. This general and very persistent result is rarely accounted for by modelers, and never used as a criterion for testing models. One method to implement RT prediction is to use a performance criterion (such as the squared error). This approach, used by Seidenberg and McClelland (1990), is based on the implicit assumption that there is a correspondence between the criterion and the RTs. Another solution is to use a dynamical multi-layered network. During processing, the activation increases until a given threshold is reached (see for example Cohen, Dunbar and McClelland (1990) where they propose to add to each unit a feedback loop similar to the cascade model of McClelland, 1979). Hinton and Shallice (1991) and Lacouture (1989) proposed a hybrid architecture where the output of the multi layered network is then decoded by a decision module equipped with a feedback loop. Thus, although a few researchers have looked into this issue (e. g. Adams and Myerson, 1999, Seidenberg and McClelland, 1990, Lacouture, 1989), it is difficult for most classic connectionist networks to show that they can reproduce this general shape. In this respect, asymptotic theories (assuming large n) can greatly simplify the investigation. Indeed, the RT distribution of the parallel race model will be analyzed using asymptotic arguments.

Some applications of the race models

The instance based theory of automaticity

Logan (1988, 1992) was among the first to use a parallel race model. He postulated in his model that there is a competition across memorized instances of the presented stimuli, each one racing to fill the accumulator (thus the word racers often used to refer to these past instances). He assumed for mathematical convenience that the accumulator size was one, so that any single racer could trigger a response. He also assumed that racers are independent and identically distributed random variables. It turned out that these three assumptions are not necessary. First, as shown by Gumbel (1958, also see Leadbetter, Lindgren and Rootzén, 1983), any small accumulator size (in absolute, not relative to the number of input channels) would leave Logan's prediction unchanged. Second, as seen in Galambos (1978), racers can be dependent, as long as dependency diminishes with any abstract measure of distance between the racers. This mathematical result is compatible with the idea that inhibition is a signal traveling from one unit to other units. If this signal has a finite velocity, then distant units will receive the inhibition after longer delays. Thus, the signal may be too late to inhibit the winner. The results of Galambos show that in these conditions, inhibition has only a marginal impact, at least in the asymptotic case where the number of racers is large. Third, Cousineau, Goodman and Shiffrin (in press) showed that the racers do not have to be identically distributed. In fact, the parameters describing the distribution of the racers can themselves be based on random variables with no disruption in Logan's predictions.

A theory of visual attention

Bundesen's theory of visual attention (TVA, 1990) is a very elaborated model of vision, attention and identification. TVA assumes a competition between the various responses. This is thus a race model. Further, the samples are collected independently for each response. By using the Exponential distribution to model the variability in the sampling times, Bundesen showed that TVA is formally identical to Luce's Choice model.

The most relevant aspect of TVA for the present is that the decision is assumed to be based on the hazard function (Luce, 1986, Burbeck and Luce, 1982). It is well known that the hazard function of the Exponential distribution is flat, reflecting the fact that it has no memory. This means that TVA in its current implementation does not accumulate evidences. Hence, TVA is a race model but not an accumulator model.

Overview of the paper

Although the weighted sum approach had a profound impact in many fields, other approaches are possible. The one developed in this text is to assume that the network of channels is not described by how much evidence it brings, but rather by when it is received.

In what follows, we will describe in depth the parallel race model. This model is general because it is not based on restrictive assumptions, such as a Poisson process in the case of the serial version. In fact, the parallel race model can be solved analytically for a broad range of distributions including the Poisson process. Because there is no serial architecture, the bottleneck is also removed. Yet, this model displays some serial effects as a response to noise. Further, we show that this model can learn to solve non-linearly separable problems (the XOR problem) without hidden units.

1- A deterministic Parallel Race Network

The purpose of this section is to introduce all the key elements of the parallel race network (PRN). This section alone is sufficient to obtain a good intuitive grasp on the model and in fact, we strongly suggest the reader to take a pause before going to the following sections to tinker the concepts developed here. The reader should also try to solve the toy problems presented in Appendix A.

The point of this section is to cast the parallel race model into a network of connections identical to those used in classic (feed-forward) connectionist networks (see Rosenblatt, 1961, McClelland and Rumelhart, 1988). The only difference is that PRN does not consider the weighted average of the connection strength but only the fastest connections.[4] All the inputs in this section are assumed binary (present or absent). However, they are considered with respect to the time dimension. We want to know when the input is activated (how strongly being irrelevant here). As such, a present input will activate its connections early, in fact, at time $t = 0$ if there is no randomness in the system. On the other hand, an absent input will never activate its connections. We can assign a time $t = \infty$ for an absence of input. This is a major change in input representation. In PRN, zero and infinity means present and absent respectively whereas in classic connectionist networks (CCN), one and zero have the opposite meaning. This change often resulted in much confusion when I presented this work in the past.

The output layers of PRN are composed of accumulator units.[5] Their purpose is to get filled, at which time they fire. Every accumulator has a certain number of slots, represented by the accumulator size K; each incoming input fills one slot.[6] Thus, the K^{th} fastest input unit will trigger a response from that accumulator.

In-between the inputs and the outputs, there are connections, also called channels. As for the input, the single relevant property of the connections in represented in the time domain: Some connections can be very efficient and introduce almost no delay for the information in a certain input to reach the accumulator of a certain output. Other connections may be less efficient (e.g. oppose a stronger resistance) to the transmission of information, so that delays will result. A connection with an infinite delay is simply a broken connection. Hence, connections with delays of zero are very effective channels whereas connections with infinite delays are off, again in opposition with the notion of weight in classic neural network.

Delays and hard threshold are the key ingredients of a race model. Because evidence can arrive on parallel channels, we use the general label “parallel race networks”.

These concepts will be discussed in more details in this section, along with two other fundamental aspects of PRN, namely, redundancy and time-out units. What we will not cover in this section are: (i) A fully elaborated mathematical notation. It will be described in Section 2. This notation will reveal important parallels between PRN and CCN. (ii) The introduction of noise (e.g. imperfect information transmission) in the inputs and the connections. With noise, it is possible to predict the shape of reaction time data produced by this model. Section 3 will solve the distributions of the PRN quite easily using asymptotic theories. (iii) The description of a learning rule. Assuming that the resistance to transmit information can be altered by exposure to stimulus-response pairs (supervised learning), a very simple learning rule can be proposed that will be described in Section 4.

Description of the PRN

The PRN assumes that information (or activation, essentially the same thing in the parallel race model) travels along channels with a finite velocity. As such, delays are the *modus operandi* of this model. What is crucial is the moment when the inputs become activated and reach the accumulators. We show here how a parallel race network can integrate confusable inputs into distinguishable patterns of response by manipulating delays.

Input-Output coding: In its simplest form, a parallel race model is composed of inputs and outputs, as seen in Figure 2(a). The purpose of the input units is to transmit the signals coming from the stimulus to the output. Each output is an accumulator (a threshold unit) with only integer-value sizes.

As mentioned earlier, the inputs are not described by how strong the signal is but by when the signal is available. In this paper, we assume that a trial begins at time $t = 0$, although this choice is arbitrary.

Insert Figure 2 about here

Connections. As is generally the case in CNN, all the inputs have a connection to all the outputs. All these channels transmit information (often termed evidence in the following). In PRN, all the channels are described by how well the information flows through them. A very efficient channel will transmit the information with but a minimal delay. Here, we assume arbitrarily that the smallest delay can be zero, representing a highly efficient channels. At the other extreme, an off connection will be labeled with an infinite delay.

The delays are neither arbitrary nor constant. They are set specifically to solve the problem, that is, to answer to the stimulus with the correct response in a given context. For

example, if an evidence is highly diagnostic for a response, it should reach the relevant accumulator with no delay (see the end of this section for alternative ways to interpret the delays). One purpose of the learning rule described in section 4 is to change the delays following exposures to the stimulus-response pairs.

Redundancy: Because it is unlikely in a large system that information travels following only one path, we introduce redundancy in the inputs. Redundancy is obtained by simply duplicating each input by a factor ρ, hence, the total number of input is actually multiplied by ρ.

Redundancy is not a necessary ingredient of PRN. This model can respond accurately to input in most problems without redundant input (one exception is the XOR problem, see Appendix A). Redundant inputs will however be useful in Section 3 where noise is introduced, allowing the use of asymptotic theories (when ρ is large).

Thresholds and decision: In PRN, each output is an accumulator with K slots. An accumulator makes a response when all its slots are filled. This happens when the K^{th} fastest input reaches the accumulator. It is assumed that K is a finite integer number, small relative to the number of redundant inputs, and independent of it.

All the accumulators can differ on there size K. This value can alternatively be called a threshold (or a boundary, by analogy with diffusion models). The thresholds and the delays in the connections are the two fundamental aspect of PRN. The second objective of the learning rule (Section 4) is to adjust the size of the accumulators when exposed to stimulus-response pairs so as to collect just the right amount of evidence.

Clocks: Discrimination is difficult when the system has to make an answer for "nothing presented" (an accumulator responding to zero evidence is meaningless). Hence, the system must

be able to decide that after a certain amount of time, if the other accumulators have not reacted, then maybe it is time for a "nothing presented" guess. This implies that time is a source of evidence in itself.

Perception of time can be achieved simply by the presence of clocks whose sole purpose is to tick at regular intervals. There exist empirical evidence for the existence of such clocks in brains. Oscillatory circuits (sometimes called pacemaker) are identified in the nervous system of many animals (including mammals; Chaplain, 1979, Levitan, Hamar and Adams, 1979, Changeux, 1983). Further, human behavioral data in favor of internal clocks were reported by Rousseau and Rousseau (1995) and Hopkins and Kristofferson (1980). In the last study, subjects had to listen to two taps and press a button when the third tap should occur. The results are exceptional (standard deviation in response times of 7 ms and symmetrical distributions) and they strongly favors "internally timed delays which can be inserted in the S-R chain" of processing (p. 241).

Because some decisions are default responses when the network perceives nothing, extra temporal inputs are needed. We call them clock units or time-out units, noted with the symbol 🕐. Clocks do not have a special status in PRN; they are extra inputs in addition to those related to the stimulus. However, the activity of the clocks does not depend on the actual pattern of input presented. They are always on. Therefore, the time-out units are activated on every trial. The number of clocks is arbitrary (and can even be large), but for many problems, one clock is sufficient. Figure 2(a) shows a PRN with one clock.

Alternative representation (A): Priority learner

The basic idea of the network is to let the information flows through it as fast as possible while avoiding conflicting responses. If responses A and B were to be made at the same time, the winning response would be random, which is of no use to the system. PRN thus let the response with the largest amount of evidence (say A) have priority while refraining the other responses for a little extra time. If after that extra time, response A is not emitted, then it is further evidence that B must be the correct response. The whole point of the parallel race network is thus to delay the movement of information.

The easiest alternative representation is to express the parallel race model as a set of rules reflecting the priority of each response. For example,

Rule R_1: "Say response A **as soon as** you receive K_A evidences".

When two responses are possible, the second must avoid interfering with the first. Even if both responses are based on some common information (embedded patterns), it does not mean that both answers should be emitted. It all depends on the presence or absence of complementary information.

In the model, the absence of a signal is not coded in any way. The only way to know that an information is missing is to make sure it had plenty of time to arrive by now. The rule for the second response could be expressed by:

Rule R_2: "After a while, say response B is you received K_B evidences".

Discrimination thus becomes the problem of finding how long is a while. Figure 3(a) represents such rules using a decision tree. The decision tree representation is misleading for one reason: it suggests that the rules are applied serially. This is not the case in PRN.

Insert Figure 3 about here

Because this model is expected to react as soon as possible, it is assumed that the accumulator sizes are the shortest possible (given constraints on accuracy). With redundancy in the inputs, this model therefore disregard unneeded information, which makes it compatible with reduction of information theories. Haider and Frensch (1996, 1999) postulated such a model where only the smallest subset of diagnostic information is used to base a decision. Race models could be used to implement such a system.

2- Mathematical notations

The Parallel race model can be represented by drawing units connected to accumulators (as in Figure 2a). However, a vector-and-matrix representation is much more convenient to compute the output of the network. Such an approach is widely spread when studying CNN. For example, in a two-layer network, the input and the connection weights are often represented by a vector I and a matrix W. The output is then simply given by a standard inner product $O = I \,.\, W$ (here, the dot is used to show explicitly the presence of an inner product). The inner product performs a weighted sum of the input. However, the minima operator does complicate things since the inner product aggregates by summing columns. In the following, we propose a simple generalization of the inner product that overcomes this problem by introducing minima into vector operations.

Let's assume that the input is composed of n_I dimensions. Suppose further that we add $n_{\clock}$ time out units. We can define a vector I that contains all the information from the input and the clocks. Formally, let $I_i > 0$ denotes the moment when the i^{th} input gets activated and let I denote the input vector $\{I_i\}$. I is of size $\rho \times (n_I + n_{\clock})$ where ρ is the redundancy factor. Because the clocks are always on, we can see I as the input vector augmented by the time at which the clocks turn on. In general, I_i is the time at which dimension I becomes activated whereas $I_{\clock}$ is the time at which the clocks gets activated.

All the inputs and the clocks have connections to the accumulators, represented by $D_{ij} > 0$, the delay between the activation of the input i and the moment when one slot of accumulator j is filled (the D_{ij} can be seen as conduction times). In order to connect all the inputs i to all the output j, the vector D is a two-dimensional matrix $\{D_{ij}\}$. Further, let D the delay matrix $\{D_{ij}\}$.

There is no restrictions on the connections: some I_i my bring evidence in favor of one response, some I_i to the other, and some to both. There may even be evidence that will contribute to no accumulator. For example, if $D_{i\mathrm{A}}$ is zero, this means that as soon as the input i is activated, this activation immediately fills one slot of the accumulator A (no delay). If, on the other hand, $D_{i\mathrm{A}}$ is infinite, this piece of evidence will never reach the accumulator. In other words, the input i is not an evidence in favor of response A. Finally, if $D_{i\mathrm{A}} = D_{i\mathrm{B}}$, then input i is equally important to decide whether the response is A or B (with respect to response C, for example). D can also be seen as a matrix of connections for the input augmented by a matrix of connections for the clocks.

Let K_{A} denotes the size of the first accumulator associated with response A, and in general, K_j the size of the j^{th} accumulator. Further, let K denotes all the accumulator sizes $\{K_j\}$.

Figure 2(b) illustrates the three components using boxes: I (changing with the input), and D and K.

For modeling purpose, we need to decide at what time the clocks become active ($I_{\textcircled{c}}$). In the following examples, we set this time at zero: the clocks are immediately active. This is unrealistic but it is not important since the delays in connections $\boldsymbol{D}_{\textcircled{c}j} > 0$ can compensate for fast clocks, delaying this temporal evidence for some time.

The output time of the i^{th} input to accumulator j is given by the sum $I_i + D_{ij}$, so that the first slot of the accumulator j will be occupied with the fastest of all the $\{I_i + D_{ij}\}$, at a time given by $Min\{I_i + D_{ij}\}$. The second slot will be filled when the second fastest input reaches the

accumulator, etc, until K_j slots are filled. The output of this accumulator occurs at time O_j when the K_j^{th} fastest input reaches the accumulator.

As an example, suppose the following PRN having 2 inputs and 1 clock (no redundancy, $\rho = 1$) defined by the following:

$$D=\begin{pmatrix} 4 & 5 \\ 1 & 3 \\ 7 & 5 \end{pmatrix} \qquad K=(2\ \ 1)$$

When presented with the input $I = (0\ 5\ 2)$, the accumulator A will have its first slot filled at time 4 (by the first input) and the second slot at time 6 (by the second input, arriving at 5, but delayed by 1 in the matrix *D*). The accumulator B will be full at time 5 (by the first accumulator). The first accumulator to fire is thus accumulator B at time $t = 5$, ending the trial.

To summarize all these steps, we can't use the standard inner product (unlike with CNN) because it is a combination of product and summation and PRN requires that the fastest time be kept. To that end, we propose a non-standard inner product that we will note $\sim$. See Appendix B for a full description of this redefined operation.

With that operation, the response times of each accumulator given a pattern of input *I* are contained in a vector $O = \{O_j\}$ given by:

$$O = I \sim D . \qquad (1.1)$$

This notation has the same level of compactness as in CNN. However, it is not explicit in this notation that the accumulator sizes *K* must be provided. Appendix B also proposes a more formal notation, although more cumbersome. Of course, only one response can be made, given by the first filled accumulator *j*, at time $t = Min\{O_j\}$, the other responses being too late.

In our previous example with `K=(2 1)`, $I \tilde{-} D$ yield `(0 5 2)` $\tilde{-} \begin{pmatrix} 4 & 5 \\ 1 & 3 \\ 7 & 5 \end{pmatrix} =$ `(6 5)`. The response is thus given by the accumulator 2 at time $t = 5$.

Despite its awkward look, remember that $\tilde{-}$ is simply a (redefined) inner product between a vector and a matrix. Whereas the standard inner product is at the core of CCN, the redefined operation is at the core of race models. Although this $\tilde{-}$ operator does not seem to facilitate the comprehension of the model, the following section will show that there exists a whole literature in statistics of extremes that connects to this operator.

Alternative representation (B): A random walk representation

The behavior of the parallel race model can be represented using a graph similar to those used with random walk models (see Figure 3(b) and Luce, 1986, Ratcliff, 1978, Ratcliff and Murdock, 1976).

In random walk models, three parameters are of interest. First, there is the base accumulation rate, sometimes called drift and denoted δ. One conceptual difficulty in the parallel race model is that the fastest delay path (closest to zero) will fill the accumulator faster. It therefore corresponds to the highest accumulation rate. At a given trial, we can obtain a rough estimate of the accumulation rate by using the reciprocal of the average delay in $I + D$.[7]

The second parameter describes the amount of variability, denoted η, affecting the accumulation rate and its distribution function (the amount of noise is set to zero so far). In addition, the distribution of noise, although often perceived as a qualitative parameter, has to be considered a free parameter. Indeed, in terms of fit, some distribution functions may provide

better fits than others, resulting in an informal (and often not reported) search over the distribution function space.

For a given amount of redundancy, we could equate $\delta = 1 / E(I + D)$ and $\eta = \mathrm{Var}(I + D)$. However, a model with larger ρ will have faster accumulation rate δ because the fastest units are likely to be much faster when sampled from a larger pool of inputs. Therefore, the parameter ρ should also enter into the description of δ. Section 5 will consider the impact of ρ.

Third, there are the boundaries. Boundaries have a direct representation in PRN: They are obtained from the accumulator size vector K.

Figure 3(b) recapitulates the parameters, and shows the network in a form closely mimicking a random walk. Despite the above similarities, the parallel race model cannot be considered as a random walk model for at least three reasons. First, evidence for one response is not necessarily evidence against another response. Each accumulator collects its evidence independently. Second, there is no within-trial variability; all variability occurs between trials by computing different sets of noisy inputs (as seen in Section 3). Third, this model does not predict a Wald distribution. Random walk models with small steps (Luce, 1986), generally called Diffusion models, are based on assumptions that are opposite to race models assumptions, and it is instructive to see in what sense. Wald (1947), who laid down the assumptions behind diffusion model analyses, wanted to make rapid decisions based on observations. He assumed that each observation was costly and so he wanted to minimize the number of **started** observations. In PRN, the exact opposite is true: the observations (the inputs) are free. There is no cost in using an extra racer (increasing ρ). The cost comes from the total decision time, so it is important to minimize the number of **completed** observations that will be used (accumulator size K). In the

race model, since each observation is free, we can start as many of them as we want, but we will not wait for all of them before answering. Given this low cost, observations are best started in parallel, while in the Wald model, one is better use them in a sequential manner.

3- Introducing variability

The version of PRN discussed in Sections 1 and 2 was deterministic. The next step is to introduce variability. Random fluctuation can be introduce in two different locations in the model: i) in the signal incoming time ***I***, or ii) in the delays ***D***. Randomness in ***I*** can be seen as the result of imperfect transmission of information. For example, if information is an impulse traveling on a neuron membrane, then the distance traveled or impurities in its composition can introduce small variability in the transmission rate. Delays *D* could be variable because of the availability of chemical neurotransmitters and their release that requires a certain amount of energy.

In addition, we show here how PRN can adapt to noise. In the following, we use the parameter η to describe the amount of noise present on the input channels. More generally, we should have noise in the input and noise in the connections. We should thus use η_I and η_D. However, because of the additivity between *I* and *D*, this generalization is trivial and will not be detailed much.

At this point, the redundancy factor is crucial because the larger redundancy is, the less variable the fastest units are. Thus, we intuitively see that increasing redundancy reduces the impact of noise.

The objectives of this Section are to derive the distribution of times to fill one accumulator and the distribution of response times across accumulators.

Because we handle various types of variables in this section, a short note on the notations used is required. We use a bold letter to denote a random variable and a handwriting font to

denote distribution functions. For example, t is one possible value of the random variable $\boldsymbol{t}$ with distribution function $\mathcal{T}$.

Variability in the input

When I and D are deterministic (not random) variables, the redefined dot product provides a convenient way of computing the response times. However, with random variables, individual response times are not as informative as the whole distribution of response time, out of which mean and standard deviation can be deduced easily. Before analyzing the distribution of the output **O**, we first define what we mean here by noise and then solve the distributions.

Given a pattern of input with its ideal activation times I, we can define the actual, noisy activation times by:

$$\boldsymbol{I} := I + noise$$

Here, we chose to use additive noise for both simplicity and plausibility. The noise must also be in the time domain so that the average noise is the average amount of delay before the i^{th} input I_i reaches the network. With additive noise, the extra delays are independent of the ideal activation times and independent of the other channels.

Similarly, we can define:

$$\boldsymbol{D} := D + noise$$

The noise can be described by a distribution function. The most important restriction is that it is never negative. Indeed, noise cannot make a signal available earlier. As such, the Gaussian distribution (among other) cannot be used in conjunction with PRN. In this paper, we suggest to restrict ourselves to the family of power distributions, of which the Exponential, the

Gamma, and the Weibull distributions are members (but not the Lognormal, the Gaussian and the Ex-Gaussian distributions). This choice is justified for mathematical tractability only.

Distribution of response times when variability is introduced

The first step is to infer the distribution of times before one accumulator is filled. This amount of time depends on the actual pattern of input and the size of the accumulator. If the inputs are highly dynamic, that is, various dimensions are accessible at different times, we would have to compute the output distributions for each output and each input pattern. This would be the case for speech recognition where various sounds are available at various moments depending on the word. For example, the sound 'a' in "alibi" and "dimensionality" occurs at very different moments. For visual stimulation inside the fovea however, it is more reasonable to think that the dimensions presented are ideally (before noise) available at stimulus onset ($t = 0$) in which case the accumulator size is the only relevant information. With this caveat in mind, we will not subscript the output distribution functions with the input pattern used.

In order to find the distribution function of each accumulator, we assume ample redundancy so that asymptotic theories can be used. The problem is to find the distribution of the K_j^{th} fastest incoming signals. Statistics of extremes (minima or maxima) studied such problems and Appendix C presents a brief overview of the results.

The problem is to find the distribution of the response times of each accumulator. Let $\mathbf{O} = \{ \mathbf{O}_j \}$ be a vector containing the response times of each accumulator (of which only the fastest is observable).

$$\boldsymbol{O} = \boldsymbol{I} \tilde{-} \boldsymbol{D} \qquad (3.1)$$

where, as usual, $\tilde{-}$ finds the K_j^{th} fastest combination of input plus delays $\boldsymbol{I}_i + \boldsymbol{D}_{ij}$.

By analogy to Eq. 3.1, we can define $\boldsymbol{O}$ as a vector of distribution functions given by:

$$\boldsymbol{O} = \{\boldsymbol{O}_i\} = \{L_{\sim}(\boldsymbol{I}, \boldsymbol{\mathcal{D}})\}$$

where L returns the asymptotic distribution of $\boldsymbol{I}$ and $\boldsymbol{\mathcal{D}}$ with respect to $\sim$. Now, the first step is to join, that is, to sum the delays in the input and the delay matrix so that the convolution describes the combination of $\boldsymbol{I}$ and $\boldsymbol{\mathcal{D}}$:

$$\boldsymbol{O} = \{L_{Min}(\boldsymbol{I} * \boldsymbol{\mathcal{D}})\}$$

At this point, solving the network's distribution is simply to find the asymptotic distribution with respect to minima, a well-known problem is statistics of extreme (Gumbel, 1958, Galambos, 1978), yielding Weibull distributions for the accumulators:

$$\boldsymbol{O} = \{W_{\alpha,\beta,\gamma}\} \tag{3.2}$$

where W is the Weibull distribution whose functions are given by

$$CDF(W_{\alpha,\beta,\gamma})(t) \equiv F(t) = 1 - e^{-\left(\frac{x-\alpha}{\beta}\right)^{\gamma}}$$

$$PDF(W_{\alpha,\beta,\gamma})(t) \equiv f(t) = \beta^{-\gamma}\gamma\,(t-\alpha)^{(\gamma-1)}\,e^{-\frac{(t-\alpha)^{\gamma}}{\beta}}$$

in which α, β and γ are parameters depending on the accumulator sizes and the number of activated input, thus related to the input pattern. Apparently, γ, the shape parameter is related to the physical aspect of the network and could be a constant across the accumulators (Cousineau and al., in press).[8]

It is important to see that $\boldsymbol{O}$ is not directly observable because all the accumulators are racing: the fastest one wins. For example, a slow response to a certain input pattern may be

censored because another accumulator produced a faster (incorrect) response. The observed distributions will therefore be distorted in some predictable way by the other responses' distributions. The observed distributions of response times $\{\boldsymbol{\mathcal{RT}}_j\}$ are given by:

$$\boldsymbol{\mathcal{RT}} = \{\boldsymbol{\mathcal{RT}}_j\} = \{\frac{1}{P_j} f_j(t) \prod_{k \neq j} [1 - F_k(t)], \quad j = A...\} \qquad (3.3)$$

where $P_j = \int_{t=0}^{\infty} f_j(u) \prod_{k \neq j} [1 - F_k(u)] du$ is a normalizing value to make sure the density function has an area of 1 (Van Zandt, Colonius and Proctor, 2000). For example, in a Yes-No experiment, by combining which type of input was entered and which accumulator responded, we could derive four RT distributions for the hits, misses, false alarms, and correct rejections. Assuming a constant γ, the resulting $\boldsymbol{\mathcal{RT}}$ distributions would be based on four parameters, α_y β_y for a "Yes" input and α_n β_n for a "No" input. Note that the corresponding $\boldsymbol{\mathcal{RT}}$ distributions are not Weibulls; in fact, the Eq. 3.3 does not have a closed form solution. Some informal simulations (reported in section 5) indicate that the theoretical $\boldsymbol{\mathcal{RT}}$ distributions are close to Weibull. Further works along the lines of Eq. 3.3 will be done in the future.

Integration and discrimination in the PRN

The capabilities of PRN can be loosely dichotomized into integration and discrimination. Integration is the ability of a system to select from a rich environment the relevant information. In PRN, this is accomplished by setting $\boldsymbol{D}_{ij}$ to the shortest possible delay for relevant information and to a very long (ultimately infinite) delay for irrelevant information.

Discrimination is the ability to select the right response given a certain pattern of input. In presence of noise, this can be very difficult to achieve. If two distinct responses are to be made using, as source of evidence, the same inputs (the worst case being an embedded pattern; see later), they may well fire at approximately the same moment. This is a problem if, for example, the response is a yes/no response. Any slight variation in the incoming information time resulting from noise will determine which response will be first, independent of the actual input. The second response should be suppressed. Perceptrons do not have this problem because high activation for one response should at the same time result in low activation for the opposite response. Similarly, a race network would have to make sure that a fast response for one alternative is accompanied by a very slow (and ultimately a never coming) response for the other alternative.

As an example, suppose the following case where response A is to be triggered by the pattern $\boldsymbol{I}^{A} = \{0, \infty, \infty, \ldots \infty\}_n$ where 0 denotes an input activated by the stimulus and ∞ an absence of input. Response B results from the complementary pattern $\boldsymbol{I}^{B} = \{\infty, 0, 0, \ldots 0\}_n$. Obviously, the thresholds are 1 and n - 1 for response A and B respectively. This should work except that in presence of noise, it is always possible that the first input becomes activated from spurious activation (noise). Since collecting the n - 1 evidences for B will take some time, it is likely (depending on the amount of noise) that false alarms will occur in the meantime.

One way to keep discrimination reliable is not to make hasty decisions (after all, race models are all about time!). Suppose it takes an average of 100 ms to collect n - 1 evidences when in presence of pattern $\boldsymbol{I}^{B}$, why not delay activation of the input $\boldsymbol{I}_1$ (real or spurious) for 120 ms? In other words, collecting n - 1 evidences is in itself good evidence that we are in presence

of the pattern $\boldsymbol{I}^{B}$ whereas collecting one evidence is not really convincing. The delays therefore should reflect the odds of a particular input in presence of noise.

Another case where discrimination is difficult is in the case of embedded patterns. Let us suppose that the pattern of input for the first response $\boldsymbol{I}^{A}$ is {1, 0, 0} and for the second $\boldsymbol{I}^{B}$ is {1, 1, 0}. Activation for $\boldsymbol{I}^{A}$ is included in the activation pattern $\boldsymbol{I}^{B}$ and in order to be sure that A is the right choice, the system must be sure that $\boldsymbol{I}_2$ will not become activated in the next few moments. In presence of unbounded variability, it is impossible to have a perfect confidence that $\boldsymbol{I}_i$ will not fire soon. One solution is to proceed with an on-average reasoning: if, on average, $\boldsymbol{I}_2$ is activated after 200 ms ± 20 ms standard deviation given pattern $\boldsymbol{I}^{B}$ was presented, then after 260 ms (three standard deviations), the system can guess confidently that $\boldsymbol{I}_2$ was not presented. Therefore, the first input should be delayed 260 ms before reaching the accumulator A ($\boldsymbol{D}_{1A}$).

Alternative representation (C): A reversed Signal Detection Theory analogy

This section draws a few parallel between PRN and SDT. For simplicity, we assume that the network is geared to solve a detection problem (with one accumulator) and that the time-out nodes are responsible for the "No" answers (see Appendix A for a complete solution). This is arbitrary, other input units could be responsible of the "No", but the clocks are ideal for our illustration since they must be delayed to be informative.

The model described so far can produce misses when some $\boldsymbol{I}_i$ are activated too late, as a result of variability in the total time to fill the A accumulator. Consequently, the clock units (variable since clocks are also subject to noise) fills the accumulator B, resulting in an erroneous B response.

Before we look at how false alarms can be introduced in the parallel race model, we will briefly look at the signal detection theory framework for a similar reasoning (see Figure 4(a)). Signal detection theory assumes that the critical value is not time but the strength of activation (more in line with connectionist networks). According to signal detection theory, an incoming signal ***S*** is defined by a distribution of activation (often a normal distribution). Noise ***N*** also has a distribution of activation, generally with smaller values. Because the system does not know whether input is present of not, it uses a criterion *c* that can be optimal or not (Zenger and Fahle, 1996). Most important is that for signal detection theory, *c* is a constant value. Hits are given by Pr(HIT) = Pr(***S*** > *c*). Independently, false alarms can occur if noise is too high and confused with the presence of a non-existent signal, with probability Pr(FA) = Pr(***N*** > *c*). This framework predicts specific curves for the Receiver-Operating Characteristics (ROC) graph (Green and Swets, 1966) as the criterion *c* is moved by putting emphasis on hits to the detriment of false alarms (or vice-versa).

Insert Figure 4 about here

In the Parallel race model, there is no false alarm so far because there is no spurious activation in the absence of inputs. Time-out units are the only way to conclude a trial when no input is present. In what follow, we assume that there is a spurious amount of activation time ***Sp*** that can fill the accumulator A (with distribution function $\overline{O_A}$). ***Sp*** could occur very early in the trial (thus being confused with a signal), although it is unlikely. Figure 4(b) shows Weibull distributions for signal present ***S*** and spurious activities ***Sp***. Weibull were used because asymptotically (if ρ is large), this is the shape they will assume.

In this version, a miss occurs when a signal is slow to activate the input $\boldsymbol{I}_i$. However, $\boldsymbol{I}_i$ has to be larger than the time-out units ***C***, which fill the B accumulator in a time whose distribution function is given by $\boldsymbol{O}_B$. Thus, in terms of signal detection theory, the criterion is not a constant but changes from trial to trial and Pr(HIT) = Pr(***S*** < ***C***). Similarly, a false alarm occurs when ***Sp*** occurs before the clock, Pr(FA) = Pr(***Sp*** < ***C***). Given that ***S***, ***C***, and ***Sp*** define the values sampled from $\boldsymbol{O}_A$, $\boldsymbol{O}_B$, and $\overline{\boldsymbol{O}_A}$ respectively, it is possible to compute these two probabilities and have one point on the ROC curve. By manipulating emphasis on signal-present or signal-absent responses, we obtain other points along the curve. The following illustrates what is meant by manipulating emphasis in PRN.

Given that redundancy is present, one solution to be sure that a signal is present is to wait for independent confirmations. This is very simply done in the parallel race model by increasing the threshold K_{A}. Similarly, to avoid false alarms, increasing K_{B} is a good solution. Formally, the probabilities are given by:

$$\Pr(HIT) = \Pr(\mathbf{S} < \mathbf{C} \mid K) = \int_0^\infty f_{\mathbf{S}}(z)[1 - F_{\mathbf{C}}(z)]dz = \int_0^\infty F_{\mathbf{S}}(z) f_{\mathbf{C}}(z)dz \qquad \text{when } K_{\mathrm{A}} = K_{\mathrm{B}} = 1$$

$$= \int_0^\infty F_{\mathbf{S}}(z)^{K_A} f_{\mathbf{C}}(z)dz \qquad \text{when } K_{\mathrm{A}} \geq 1$$

$$= \int_0^\infty f_{\mathbf{S}}(z)[1 - F_{\mathbf{C}}(z)^{K_B}]dz \qquad \text{when } K_{\mathrm{B}} \geq 1$$

where *f* returns the pdf and *F*, the cdf, ***S*** is distributed as $\boldsymbol{O}_A$, and ***C*** is distributed as $\boldsymbol{O}_B$.

Replacing ***S*** for ***Sp*** gives the equivalent results for Pr(FA), where ***Sp*** is distributed as $\overline{\boldsymbol{O}_A}$. As an

illustration, we used in Figure 4(c) Weibull distributions with standard deviation of 1 for $\boldsymbol{O}_A$ and 5 for $\overline{\boldsymbol{O}_A}$ and manipulated *K* within each curve. What is generally interpreted as a change in criterion *c* is obtained by increasing thresholds in the Parallel race model. In other words, criterion shift does not result from a change in a physical criterion but in the number of evidence collected without changing the time-out distribution. In the Figure, the three distinct curves are obtained by manipulating the position of $\boldsymbol{O}_B$ of the time-out distribution, that is, how long the clock units are delayed (noise notwithstanding). This is the same as changing perceptibility (usually measured by *d*'). Overall, the curves seems plausible (but see Yonelinas, Kroll, Dobbins and Soltani, 1998) although it is not the aim of this alternative representation.

Conclusion

We added noise to the PRN using very general distribution functions $\mathcal{I}$ and $\mathcal{D}$. Despite the generality of these assumed distribution and using asymptotic theories of extremes, we solved the distribution times to fill every accumulator. Of course, since only the fastest accumulator wins, the Weibull distributions are somehow distorted using Eq.3.3. Nevertheless, the important point is that the presence of noise changes nothing with respect to how the network operates. It is always a question of delaying les diagnostic information and waiting for a critical number of them. Noise, introducing a larger uncertainty on how long to wait, simply increases the delays. The system has to wait longer if the inputs are often delayed. This is a natural extension of the underlying assumptions of the PRN.

Meyer, Irwin, Osman and Kounios (1988) suggested a method to address the question of internal noise using speed-accuracy decomposition techniques. As seen in Appendix D, the model exhibits typical patterns of speed-accuracy trade-off (SATO) when time constraints are imposed. In this case, a third response (C: “Respond now”) is added to the system. Its connections, $D_{\textcircled{t}C}$ are activated by an external signal rather than by an internal clock. SATO plots can be obtained with one extra hypothesis related to the guessing content.

4- A parallel race model with a learning rule: A parallel race network

So far, the PRN turns out to be a very flexible model. A compact mathematical notation stem from it (Section 2) and it supports the presence of noise quite naturally (Section 3). It also offers alternative representations (Random walk, Priority learner, reverse SDT) that help understanding its behavior.[9] However, the parallel race model cannot describe the dynamic of changes in *D* and *K* when the external conditions change. For example, suppose a task where, at some point, some distractors are no longer shown so that formerly irrelevant features become diagnostic. It is not clear how the parallel race model would adapt to changes because so far, it lacks learning mechanisms.

This is in opposition with the parallel distributed processing (PDP) models (McClelland and Rumelhart, 1988) where the learning mechanism is fully described. However, there is no clear consensus on what kind of RT distributions PDP models predict. Thus, the predictive strength of the parallel race model (the RTs) is the weakness of PDP models, and vice versa for learning.

In this section, we fill this gap by providing the parallel race model a simple learning rule. The objective is that the network finds itself the right delays *D* and the appropriate thresholds *K*. The learning rule described is simple and when we tested the resulting network with the problems described in Appendix A, it always converged toward the theoretical solutions.

Description of the Parallel Race Network

Architecture

The physical implementation of the network is the same as before (refer to Figure 2). As usual, inputs are considered to be signals coming from the stimulus. Because signals take time to travel, inputs are defined by the moment their activation reaches the input unit, not by their strength. As such, signals are stochastic: either a signal has arrived, or it has not yet. However, if there is no signal, there is still the possibility that the input unit receives a spurious signal from random activation. We assume that spurious signals whose distribution function is given by $\bar{I}$ will arrive on average after a larger amount of time (the variability of $\bar{I}$ is larger than the variability of I). As before, the connections D between the inputs and the accumulators are not weights but delays in the transfer of information.

The accumulators' role is to integrate the number of evidence coming from the inputs. Accumulators are filled by incoming signals; each activated signal takes 1 slot. Accumulators are in effect a thresholded unit because they trigger a response only when enough evidence is received.

Decision

Response times are computed in exactly the same way as before. However, as seen next, the learning rule allows thresholds to increase by non-integer values. Because only integers can be used for the accumulator sizes, we use the integer parts of the thresholds $\lfloor K \rfloor$ in the decision rule and write:

$$\boldsymbol{O} = \boldsymbol{I} \tilde{-} \boldsymbol{D}$$

which is, in the more elaborated notation of Appendix A, noted:

$$\boldsymbol{O} = \boldsymbol{I}_{\left(\underset{Min_{\lfloor K \rfloor}}{+}\right)} \boldsymbol{D} .$$

As usual, only the fastest accumulator makes a response j such that $\boldsymbol{O}_j = Min\{\boldsymbol{O}\}$.

Learning

Learning in the parallel race network takes place when an error occurs. There are two types of errors:

i) For the accumulator j which has produced a false alarm (it has fire too soon while it should have remained silent), one of the delay D_{ij} responsible for that error (i. e. one of the K_j^{th} fastest delays $\{I_i + D_{ij}\}$) is increased by an arbitrary value φ. In other words, because that accumulator was filled too rapidly, it slows one of its inputs to avoid this error again (through the connection D_{ij}).

ii) For the accumulator that missed (it remained silent while if should have fired), the threshold K_j is modified. If the total number of actual input activated on that trial is larger than the value of K_j, the size of that accumulator is increased by an arbitrary value ε (or vice-versa). As long as errors occur, the accumulator size will vary to reflect the number of input.

At the beginning of a simulation, all thresholds K_j are set to 1, and delays D_{ij} are set to random uniform numbers between 1.0 and 1.1 in arbitrary units of time. We used random initial delays to avoid tied decisions. Otherwise, it would make the error attribution difficult to assess. In the simulations of Section 5, the changes in delay φ were set to a small value, 0.5, and changes in threshold ε were set to 0.2. The choices for ε and φ were arbitrary and we tested

different values with no qualitative change on the learning behavior of the network. The only restriction is that ε be smaller or equal to 1, otherwise $\lfloor K_j \rfloor$ might not assume all the successive integer. For example, with ε = 1.5, on the second miss, the effective threshold would jump from 2 ($\lfloor 2.5 \rfloor$) to 4 ($\lfloor 4 \rfloor$), skipping the value 3.

Overall, this network embodies two principles: i) the response is a competition among accumulators, each waiting for K_j supportive inputs before it fires. It is a winner-takes-all process because one or a limited number of inputs will end a trial. ii) The learning is a cooperative process. If the accumulator connected to response *j* fires too rapidly, it will be a false alarm. The system needs to slow these accumulators by increasing the delays.

Alternative representation (D): How long to wait for friends.[10]

You are at the corner of a street, waiting for a friend who is late. How long are you going to wait? In this situation, there are no false alarms (unless you are prosopagnosic), but there can be misses if you leave too early. Of course, your friend might never come too, in which case, the earlier you give up (correct rejection), the better it is for you. The best course of action is to give up reasonably fast for a first meeting, but if you later discovered that just missed your friend, then, you are going to increase your tolerance, how long you wait. In effect, you delay the time-out (give-up) signal to accommodate that friend. Depending on the punctuality of your various friends, you end up with different delays for each. These delays are learned by a trial-and-miss procedure. This is exactly the procedure implemented in PRN.

5- Simulations and experiment

Section 4 presented a learning rule that seems both simple and intuitive. Application of this rule to deterministic input (no noise) is straightforward, and the reader is invited to apply it to the problems presented in Appendix A or check the web site for the results. However, despites all the mathematical and statistical niceties that we developed in Sections 2 and 3, we have no certainty that this learning rule can learn and settle in a stable configuration of delays and thresholds when the inputs are noisy, redundant, or both.[11]

In this section, we explore the capabilities of PRN to learn a difficult problem, the XOR problem. We first show that this network can learn such a non-linearly separable problem even though it does not have hidden layer units (the XOR problem is described in Appendix A; Minsky and Papert, 1969). Second, we explore the impact of noise on the learned delays. Using a SDT analogy, we already know that the more noise is present in the input, the slower the time-out units should be to avoid misses. Indeed, the learning rule adapts the delays in such a manner. Third, we look at the simulated response times and look at how similar they are to known distributions. Finally, we present a small experiment with human subjects aimed at testing a simple prediction of a time-based network.

Review of the XOR problem and overview of the simulations

A two-dimensional XOR problem is one where both inputs present and none present elicit the same response (say response A). When there is no redundancy ($\rho = 1$), if two inputs are present, an A response can be made immediately. This suggests that the delays between the inputs and the accumulator A (call them t_0) should be very short or even zero and the

accumulator size should be 2. When redundancy is present ($\rho > 1$) however, accumulator A receiving 2 inputs does not mean that the conjunction is present since both activation could results from the same redundant input. In fact, if one dimension can send ρ redundant activation, an accumulator size of $\rho + 1$ or greater is the only one that guaranties that 2 inputs are present. On the other hand, receiving a single evidence is enough for the alternative decision, say B. Yet, this decision must not be hasty since other evidences might be on their way. Thus, the delay between any input and accumulator B (call it t_1) should be large, al least larger than the average time to receive $\rho + 1$ evidence on accumulator A.

In the same vein, the clocks connected to the A response should be even more delayed (call this delay t_2), at least more than the average delay to say B which is itself larger than the average time to detect the presence of the conjunction.

Figure 5, panel a, illustrates the theoretical solution to a XOR problem inside PRN.

Insert Figure 5 about here.

To check the ability of the PRN to learn, we ran simulations where both $\boldsymbol{I}$ and $\boldsymbol{D}$ were noisy and redundancy manipulated. The delay matrix D was modified on every trial by adding uniform random delays in the range [0..0.1] in arbitrary unit of time ($\boldsymbol{D}_{ij} \sim \text{Uniform}[0, 0.1]$).[12] This represents a small amount of noise, but since it is additive with noise in the input, the overall amount of noise used will be more important in some simulations. The inputs were either 0 if the dimension is on, or 10 if it was off (spurious activation). To this, we added exponentially distributed noise. We chose to describe the magnitude of noise by the mean delay η added to the inputs ($\boldsymbol{\mathcal{I}} \sim \text{Exponential}[\eta]$). Thus, η is the average increase in $\boldsymbol{I}_i$. The choice of the Uniform

and the Exponential distributions is arbitrary. However, both satisfy the criteria C_1 and C_2 presented in Appendix C. On some trials, the noise in one of the two inputs may exceed $\boldsymbol{D}_{1B}$, resulting in information on that relevant dimension coming too late (response B occurred). Because the Exponential is unbounded, there will always be cases where the diagnostic information will arrive too late ($\Pr(\boldsymbol{I}_2 > \boldsymbol{I}_1 + \boldsymbol{D}_{1B}) > 0$). Spurious activation for the absence of signals was also noisy, the increment in delays being on average 10 times slower ($\overline{\boldsymbol{I}} \sim$ Exponential $[10 \times \eta\ ]$).

Figures 5(b) and 5(c) show the two distributions of noise that we added to the delay matrix and to the input signals. We used $\eta = 0.5$ and $\eta = 2$ for small and large amount of noise respectively, in arbitrary units of time. $\eta = 0$ means that no noise at all was added.

We also manipulated redundancy ρ by providing ρ paths where information on dimensions 1 and 2 and the clocks can travel (in this case for a total of 2 ρ inputs). Each path resulted in inputs with its own noise. Accordingly, the rows in the matrix $\boldsymbol{D}$ were duplicated so that each input i have its own connection to the accumulators.

All the simulations were done by first initializing a random network as described in Section 4. Two time-out units were added to all inputs. Training was performed by selecting an input-output pair randomly. For example, $\{0,0\} \rightarrow \{0,\infty\}$ means that when both inputs are on $\{0,0\}$, the first accumulator should respond first. The input was duplicated ρ times and then noise was added to each channel. The delay matrix also received noise on every trial. Based on the thresholds, the time to fill each accumulator O_j was obtained by finding the K_j^{th} fastest input

and the fastest accumulator made a response. An epoch of training consisted of 10 trials. The number of epoch varied in the following simulations.

Overall, the parameters η and ρ are manipulated and a matrix *D* and a vector *K* is learned by PRN. Sample programs to simulate PRN are available for Mathematica on the author's web site at http://ocean.sim.umontreal.ca/cousined/papers/07-PRM/.

Learning a XOR problem.

This first set of simulations explored the learning capabilities of the PRN in presence of noise and redundancy.

We ran PRN on the XOR problem for 200 epochs (2000 trials). Figure 6 shows the percent of errors (P(e)) made by the network averaged over epochs. In Figure 6, there is no redundancy (ρ = 1). As seen in panel (a), learning a XOR problem is rapid (115 trials) when no noise is added (η = 0.0). Such a fast learning is surprising since CCN typically requires hundreds or thousands of trials to do the same (O'Reilly, 1996).[13] Right part of the Figure shows the solution adopted by the network (compare with the matrix solution in Figure 5(a). t_3 is represented by the largest delays (8.04 arbitrary units of time). t_0 corresponds to an average base delay of 4.05. The increase in t_0 occurred while the thresholds were stabilizing at the correct values. It took about 110 trials to adjust them.

Insert Figure 6 about here

Figure 6(b) and (c) show two learning sequences for the XOR problem with two level of noise in the signal (η = 0.5 and η = 2.0), but again, no extra redundancy (ρ = 1). As seen, after 2000 trials, the network still produces a few errors, but they are well spread out, and result only

from noise. Four patterns of error are in fact possible. On a {0,0} input, one of the channel may be slow to be activated, resulting in a false B response. On a {0,∞} or {∞,0} input, the activated dimension could be slow or spurious activation could occur early on the inactivated channel, resulting in either a false A or false A' response respectively. Finally, when noting is presented {∞,∞}, spurious activation could happen earlier than the clocks, resulting in a false B response. As the number of epoch is increased, errors become more and more sparse.[14] For $\eta = 2.0$, the network produced 2% of errors in the last 100 epochs. To avoid misses, the system adapted to the variability in the input by increasing the average delay between t_0 and t_1.

The delay matrix can also be seen with a Random Walk representation. Informally, we can set drift rates as the reciprocal of the delays, shorter delays meaning faster accrual of information. When a small amount of noise is used ($\eta = 0.5$), drift rates for response B are about 1.5 times smaller than the drift rate for response A, and drift rates for response A', about 2.0 times smaller than the drift rate for A. With more noise ($\eta = 2.0$), these figures increased to about 4 times and 7 times smaller, respectively. Clearly, delays are affected by noise. Thresholds, on the other hand, are not affected by noise. This is expected since so far there is no redundant signals and thus no room for larger thresholds.

Apart from noise, we also manipulated redundancy ρ. It is possible, when ρ is larger than 1, that the input can access the system by more than one path. The following set of simulations tries to assess the impact of redundancy on learning. Because we showed in Cousineau and al. (in press) that asymptotic distributions converge rapidly, we explored only values of ρ between 1 and 16.

Figure 7 shows the impact of redundancy for two levels of noise ($\eta = 0.5$ and $\eta = 2.0$). As seen, learning is barely slowed down by the redundant sources of incoming signals, errors reaching a lower asymptote in less than 1100 trials for $\rho = 16$. This means that the extra inputs learn rapidly to coordinate their action (through the delays $\boldsymbol{D}$). Further, adding noise in conjunction with redundancy did not hinder learning, the network finding a stable configuration with only occasional errors.

Insert Figure 7 about here

In sum, PRN can learn very rapidly a difficult problem. Further, noise and redundancy have merely no effect on learning. We see next that redundancy, though having only a limited effect on the amount of trials to reach close to perfect performance has a profound, opposite effect to noise on response times.

Relation of redundancy and noise on delays and thresholds

The purpose of this section is to evaluate how the noise and redundancy factors affect the internal delays D and the thresholds K. For example, how long should be t_1 so that response B is discriminated efficiently from response A, allowing for 3% of errors, and similarly, how long should be t_2 for response A'. We first derive an informal theoretical relationship to gain some preview of the results. In this preview, we will not aim at perfect performance but rather at performance with 3% of errors. We then present actual simulations to check whether the learning rule does perform as expected.

Theoretical delays as a function of η and ρ

To have an intuition of the following arguments, the best analogy is the reverse SDT analogy. In this view, the B response should be delayed, compensating for uncertainty in the input. This is the role of the t_1 delay. Thus, the more noise, the larger t_1 should be. On the other hand, if we wait for the K_A fastest ones, variability will decrease as redundancy increases. This results in a narrower distribution for the A responses and thus the B responses can be closer to zero. Redundancy undoes what noise accomplishes.

The objective of this subsection is to formalize more precisely this relationship. In the following, for simplicity's sake, we ignore the A' responses, concentrating on the A and B responses and on the false B errors to $I^A = \{0,0\}$ input. Errors are a critical aspect because to totally avoid errors, the distribution of the B responses would need to be infinitely distant from the distribution of the A responses.

An error occurs when signals come too late and the alternative response fires. This happens when the signal is detected after the response B is emitted, $E(\boldsymbol{I}) + t_0$. We can imagine that

$$t_1 = \theta \; E(\boldsymbol{I}^A) + t_0 \qquad (5.1)$$

that is, the delay t_1 is θ times above the average time to receive all the dimensions relevant to response A. t_1 is like a criterion in the reverse SDT analogy and θ can be seen as the tolerance parameter.

Thus, the probability of an error is given by:

$$\begin{aligned}\Pr(error) &= \Pr(\boldsymbol{I}^{A} > t_1 + noise) \\ &= \Pr(t_0 + noise > t_1 + noise) \\ &= \Pr(t_0 + noise > \theta\, \mathrm{E}(\boldsymbol{I}^{A}) + t_0 + noise) \\ &= \Pr(noise > \theta\, \mathrm{E}(\boldsymbol{I}^{A}) + noise)\end{aligned}$$

This last equation can be approximated using on-average reasoning:

$$\Pr(error) \approx \Pr(noise > \theta\, \mathrm{E}(\boldsymbol{I}^{A}) + E(\boldsymbol{I}^{A}))$$

Setting Pr(*error*) at 3%, a pre specified level of errors, and knowing that 97% of the Exponential distribution is located below three means, we can solve: $\theta = 2$

The factor $\mathrm{E}(\boldsymbol{I}^{A})$ in Eq. 5.1 is still unknown. It would be tempting to equate $\mathrm{E}(\boldsymbol{I}^{A})$ with the average delay in the signal on a signal channel, η in our case. However, this is erroneous. This is a parallel system and the pattern $\boldsymbol{I}^{A}$ can be recognized based on the fastest channels, not the average channels.

A formula derived in Cousineau and al. (in press, section 3.2) suggests that $\mathrm{E}(\boldsymbol{I}^{A}) = \frac{\eta}{\sqrt[\gamma]{\rho}}$

so that Eq. 5.1 becomes:

$$t_1 \approx \theta \frac{\eta}{\sqrt[\gamma]{\rho}} + t_0 \qquad (5.2)$$

where $\theta = 2$ is the tolerance parameter described previously, γ is the shape of the asymptotic distribution of RT and $t_0$is zero in the solution of the XOR problem. This estimation is based on the postulate that the redundancy factor is large. For small ρ, this is incorrect in the general case but nevertheless produced valid approximations when ρ was greater or equal to 8.

Figure 8 shows the theoretical surface representing the needed amount of time t_1. With a small amount of noise ($\eta = 0.5$ arbitrary unit of time) and no redundancy ($\rho = 1$), the

accumulator B has to wait about 1 arbitrary unit of time before being confident that dimension 2 is indeed absent. This value increases to 4 units of time when the average noise delay η is 2 units. Redundancy reverses this trend. With small noisy delays but high redundancy (η = 0.5 and ρ = 8), a very short delay (t_1 = 0.25) is enough to be sure no inputs are present on dimension 2.

Insert Figure 8 about here

Overall, this predicts that increasing noise should have a linear effect on the B response times but that the effect of redundancy should be a power curve. Further, the curvature of the ρ function, γ, should be the reciprocal of the RT distribution shape.[15] According to this view, the reduction in scale in turn predicts a reduction in mean response time and standard deviation. The form of the reductions should be power functions, an observation that seems reliable (see Newell and Rosenbloom, 1981, Logan, 1988, Anderson and Tweeny, 1997, Cousineau and Larochelle, submitted and Cousineau and Lefebvre, submitted, for more on power curves; but see Rickard, 1997, Heathcote and Mewhort, 1995, Heathcote, Brown and Mewhort, 1998, Myung, 2000).[16]

The main point is that in normal situation, the starting point of the alternative response t_1 is a function of the triplet {η, ρ, θ}. η and ρ describe the variability of the signal and θ determines how many errors are tolerated, i.e., how much overlap are allowed between the A and the B responses.

To further test this relationship, and because the model is much more complex, owing to the learning rule, we ran simulations using a factorial design involving η and ρ simultaneously. We continue to use the XOR problem, because there is no counter indication that forbids it: it is a plain regular problem for parallel race models.

We ran the simulations for 4000 learning trials in which we varied ρ (1, 2, and 4) and η (0.5, 1.0, 1.5, and 2.0). Each combination of η × ρ was replicated ten times. For each, we extracted two summary values described below (Δ***K*** and log Δ***D***) obtained at the end of training, and submitted these results to two distinct ANOVAs.

To accomplish these analyses efficiently, we needed to reduce the matrix ***D*** and the vector *K* to a few summary values. We concentrated on the following summary values. First, we summarized the ***D*** matrix by considering only the two responses "both input present" (response A) and "one input present" (response B). For these, we took the average delays in each section of the matrix t_1 and t_2, called $\overline{t_1}$ and $\overline{t_2}$. These averages excluded the time-out units t_3 since these are used for the third response "nothing present" (response A'). Finally, we computed the difference $\Delta\boldsymbol{D} = \overline{t_2} - \overline{t_1}$. Based on the estimate postulated in Eq. 5.2, Δ***D*** should be proportional to $\theta \frac{\eta}{\sqrt[\gamma]{\rho}}$ and because of this multiplicative relation, we considered log Δ***D*** in relation with log η and log ρ.[17] The second summary value is the difference between the two thresholds $\Delta K = K_B - K_A$.

Learned delays vs. optimal delays

The first ANOVA on Log Δ***D*** (Table 1, top part) reveals that both log noise and log redundancy affect log Δ***D*** significantly. The results confirm what we expected, and the weight for the log ρ factor is close to -1/γ (in the asymptotic case, γ should be 2 because both the Uniform noise in ***D*** and the Exponential variability in ***I*** add 1 to the shape of the simulated response times distribution). The intercept (5.2) reflects the value of log θ . It means θ is very

large, suggesting very few errors remain after 4000 trials. Indeed, the overall error rate was less than 1%.

Insert Table 1 about here

In terms of a random walk representation, we see that the accumulation rates depend on both the amount of noise and redundancy. The model adapts to noise by increasing delays, resulting in a slower rate of accumulation and benefits from redundancy by reducing delays.

Learned threshold vs. optimal thresholds.

At this time, it is not clear if, in presence of redundancy, the system will learn to use the smallest possible thresholds or not. We had the feeling in Figure 6 that they were not affected by noise, but since ρ was constant to one, this remained to be confirmed.

As seen in the bottom part of Table 1, the difference in threshold ΔK is significantly affected by redundancy only. When redundancy increased by one, the lag between thresholds increased by 0.4. In one sense, the increase in ΔK is not as fast as it could be (it should be 1.0 ρ). This suggests that the network does not use all the available information, thus making erroneous decisions. On the other hand, this increase is not a surprise: Collecting K_A evidences when twice as many channels are on is faster. Thus, if by time t_1, the accumulator A is filled, it indicates a high accrual rate. In terms of a random walk representation, the boundaries are not increased as rapidly as redundancy. The larger ρ is, the more stable the winners are, and thus, the criterion t_1 can be set closer to t_0.

Distribution analysis

We are also interested in the distributions of the simulated response times (SRT). Because variability in $\mathcal{D}$ is uniform and variability in $\mathcal{I}$ is exponential, the results from statistics of extremes (Section 3) predict that the SRT distribution for one accumulator will be exactly Weibull with shape γ of 2. However, the response distribution may only look approximately Weibull since two accumulators are in competition. This section explores this issue.

As in previously, the network was trained on a XOR problem for 4000 iterations. Among the last 1000 SRT, we kept the response times to the input pattern $\boldsymbol{I}^{\Lambda} = \{0, 0\}$. The SRT thus collected constituted one sample. We repeated this procedure 100 times for each combination of $\eta \times \rho$. Each sample was subjected to distribution analysis (Cousineau and Larochelle, 1997) to test which distribution fitted best (comparing the Weibull, LogNormal, and ExGaussian distributions).

As can be seen in Table 2 (top part), the SRT distributions were mostly Weibull (all percent best fit larger than 40%). Surprisingly, with an intermediate amount of redundancy ($\rho = 4$ and $\rho = 8$), the index of fit in favor of the Weibull was at its lowest, favoring almost equally the LogNormal (Ulrich and Miller, 1993). This factor stabilizes as ρ increases to larger values. As seen in the second part of Table 2, noise did not have any impact on percent best fit, being in 80% of the case better fitted by a Weibull distribution.

Insert Table 2 about here

In addition, we analyzed the estimated parameters α, β and γ of the best-fitting Weibull distribution to the SRT. Figure 9 shows the average parameters. As seen, the shape parameters $\hat{\gamma}$

slowly converge toward 2, the expected shape for O. This parameter is slow to converge and we had to explore larger amount of redundancy (up to $\rho = 32$). For small ρ, the asymptotic argument is only an approximation and $\hat{\gamma}$ is generally smaller than 2. This confirms that the shape γ does not depend on the amount of noise used in the simulations. In contrast, we see that noise and redundancy both affect the position $\hat{\alpha}$ and the scale $\hat{\beta}$ with roughly the same multiplicative pattern described in the previous section (and illustrated in Figure 8). Hence, a preliminary conclusion is that if we could manipulate redundancy and noise by manipulating the stimuli, we might be able to obtain distinct values for estimated $\hat{\gamma}$ on one side and $\hat{\alpha}$ and $\hat{\beta}$ on the other. However, it is not clear at this point how redundancy can be manipulated empirically (Miller, 1982, Colonius, 1988, 1990, Diederich, 1992, Townsend and Nozawa, 1995, Cousineau, 2001).

Insert Figure 9 about here

Experiment with a XOR task

In the following, we contrasted some predictions of the PRN on the XOR task with empirical data. The most important prediction, one that has been disregarded by many CCN concerns the absence of input. PRN specifically predicts that absent input will be detected more slowly than present input. On logical grounds, this prediction is easy to understand since we are always more readily confident that something is present if we see it than that nothing is there if we don't see anything. However, PRN further supported this from approximations to ΔD (Eq. 5.2). Simulations also supported this claim since $t_2 - t_1$ and $t_1 - t_0$ are the same in Figure 6 for a given amount of noise.

This predicts the following mean RT ordering: Response A (both input present) should be faster than response B (only one input present) itself faster than response A' (no input present). Further, because RT should reflect ΔD, it predicts that the mean RT difference between response B and A should be identical to the difference between the mean RT between response A' and B.

Another question is how the variability in response times changes as we go from responses with short delays to responses with long delays. It is often the case in psychology that longer response times are also more variable. This possibility is not a logical consequence of PRN. Thus, following the reverse SDT analogy, overlap between competing response's distribution (and error rate) is set by shifting the second response distribution toward longer base delay. However, if the second response is more spread out, the shift does not have to be as large to achieve an equivalent amount of errors. The following experiment will explore this issue as well.

Experiment

The experiment is a simple XOR task were subjects had to detect two signals or no signal with one response and a single signal with a second response.

Subjects: Two students from Université de Montréal, one undergraduate and one graduate, one male and one female, participated to the experiment for money. Both were right-handed and had normal or corrected to normal vision.

Stimuli: The stimuli were low-luminance dots 1 pixel in size. The luminance was 0.35 cd/m^2 on a background luminance of 0.003 cd/m^2. There could be either zero, one or two dots presented in one of two possible locations in the center of the computer display. The two locations were horizontal, separated by 5.5 ° and viewed at a distance of 50 cm, using a chinrest.

Procedure: The subjects were first dark-adapted for a period of 10 minutes. Followed 720 trials with no break. Each trial consisted of a 1 s tone (400 MHz) followed by either zero, one or two dots. The subjects' instructions were to response with the key "/" using the right hand if only one dot was present or with the "z" using the left hand otherwise. The test display stayed for 2 s or until a response, measured with ms accuracy. 500 ms separated each trial. The number of dot conditions were equally likely, resulting in the response "z" being twice more frequent than the "/" response. When one dot was visible, left and right locations were used equally often.

Results

Figure 10 shows the mean RT and the percent of errors as a function of the number of dots. The position (left/right) of the dots had no impact on the mean RT ($F(2, 24) < 1$) but had one on the errors ($F(2, 24) = 38.2$, $p < .05$).

Insert Figure 10 about here

As seen in Table 3, the change in mean RT ($\overline{RT}$) is linear with a slope of 17 ms per dot absent ($r = 0.114$, $p < .001$). Also seen is the fact that standard deviation in RT ($\overleftrightarrow{RT}$) is increasing with increasing uncertainty. This suggests that the longer the delays, the more variability there is. Further, we see that the change in standard deviation is not perfectly linear, being much larger when no physical evidence sustains the response. This suggests, in terms of PRN, that the internal clocks are more variable than external signals (Rousseau and Rousseau, 1996, Hopkins and Kristofferson, 1980).

Insert Table 3 about here

Figure 11 shows the RT distributions of the subjects and Table 3, the best-fitting Weibull parameters obtained with PASTIS (Cousineau and Larochelle, 1997). The Weibull parameters are used only as an approximation since, because of response competition, the observed responses are not exactly Weibull.

As seen, the shape parameter $\hat{\gamma}$ is constant at near 1.90. This estimate's reliability should be varying with redundancy (for example, increasing the number of redundant photons reaching the retina), but since we did not manipulated ρ, we can't conclude anything on this value. Further, the change in mean RT is mostly accounted for by an increase in the scale of the distribution, measured by $\hat{\beta}$. The position of the distributions $\hat{\alpha}$ seems to indicate a small trend to be increasing with uncertainty, but this trend is itself uncertain. PRN does predict an increase in $\hat{\alpha}$ as the delays are increased, which should be the case for a XOR problem.

Overall, the pattern of results suggests one last analogy. Clearly, the observed RT distributions overlap amply, which should, according to a reverse SDT analogy, predicts many errors despite the observed data. I believe we are trying to study rocks falling in the middle of a lake from the waves that reach the bank. Any slight variation in their falls will be amplified by the propagation of the wave. However, if the delays between the impacts are small, the difference in the waves will remain small all along, soon becoming not significant. The same may be true if the PRN is organized in layers. Each winner of a given layer will send redundant signals of its success, thus allowing the race to propagate to other layers. Indeed, we may have some indications of this since we have a very small effect of $\hat{\alpha}$ but a huge effect of $\hat{\beta}$. This issue is left for further research.

Finally, researchers analyzing PRN responses would have to explain why conjunctions are answered faster than disjunctions. Such results are sometimes found in empirical studies such as in the present experiment or for example, Trabasso, Rollins and Shaughnessy (1971). Similarly, Fournier, Eriksen and Boyd (1998) found faster RTs for a triple conjunction than to double conjunctions in the same task. Robertson (2000) also has data in a cued search where detecting the conjunction is faster than detecting objects with only one of the two features, itself faster than detecting objects with none of the features. These effects are mirrored by the word superiority effect (Reicher, 1969, Rumelhart and Siple, 1974). In a typical task, subjects are presented strings of letters. The results show that letters are typically reported more accurately if they are part of a known word. The logic of the parallel race model suggests that words will be accessed rapidly if present in the lexicon (therefore preventing decay from iconic memory) and that all the other responses will be on stand-by in the meantime. By contrast, the logic that prevailed in the past suggested that faster RTs resulted from easier stimuli. This lead Smith and Havilland (1972) and others (Schyns and Rodet, 1995, Goldstone, 1998, Travers, 1973) to postulate the existence of a unitization process where distinct parts could be unified (or chunked, Newell and Rosenbloom, 1981) so that complex inputs could be processed as if they were a simple feature. This hypothesis was the result of an unconscious "serial" tradition prevalent at that time. Further, how such unitization was accomplished was not explained, but simply labeled "perceptual learning". This term just pushed the responsibility of the explanation to others. In fact, we think that perceptual learning is the cornerstone of cognition, not perception. In this respect, the parallel race model provides a tentative explanation that breaks with the serial tradition.

General Discussion

The parallel race model is very general because it not only predicts error rates, but also makes clear predictions on mean RT, on RT variance, on the whole shape of the RT distribution, and on phenomena such as the effect of rewards on the receiver-operating characteristics (ROC) curve and on speed-accuracy trade-off (SATO).

It is tempting to call PRN a "race-race" model. Indeed, it is a race model at two distinct levels. First, and following the usual meaning of the expression "race model", this model has accumulators and each one is racing to be the first to trigger a response. At this level, this is a between-accumulator race model. Because there are generally few accumulators, we can't use asymptotic theories to infer the distribution of RT, and thus need Eq. 3.3. Second, when a single accumulator is considered, we have redundant channels in competition to be the first to fill the accumulator. Thus, we have a race going on within accumulator. Assuming large redundancy (and the more technical assumptions C_1 and C_2 of Appendix 3), this within-accumulator race can be solved analytically, thus serving as input to Eq. 3.3.

More importantly however is the fact that PRM is a gate. It is a gate first because it shows how supervised connectionist networks and accumulator models can be related and unified under a unique network architecture. The single fundamental difference between the two is the use of summation or minimum to ponder the incoming evidence. In the first case, the evidence are meaningful if thought as strength of the input and strength of connections whereas in the second case, the evidence are seen in the time-domain as moments when the input becomes available and how determinant they are with delays in the connections.

Second, PRM is also a gate because to my knowledge, it is the first time that distributions of response times are so tightly coupled to the mathematical operation used to compute the output of a network (relating $\boldsymbol{I}_{\left(\begin{smallmatrix}+\\ Min_K\end{smallmatrix}\right)}\boldsymbol{D}$ with $L_{Min_K}(\mathcal{I} * \mathcal{D})$). This coupling is possible here owing to the flexibility of asymptotic theories. This makes quantitative predictions much easier and often, it is not necessary to simulate the model to gain knowledge of its behavior.

Finally, PRN is a gate toward a possibly new field of mathematics, non-linear linear algebra. Indeed, redefining the inner product opens a totally new area for matrix operations. In this new area, the identity matrix is particular since the main diagonal is composed of zeros and the remaining locations are occupied with ∞. The zeros are neutral with respect to the joining operator whereas ∞ is always rejected by the aggregate operator, minimum.[18] If we can make a formal proof that PRN can learn, it will be a demonstration that this new area can support complex reasoning.

Summary of PRM

In one sense, PRM is a race model because of the competition between accumulators. In another sense, it is a connectionist network by its architecture, identical to two-layer Perceptrons, and by its ability to learn (McClelland and Rumelhart, 1988). However, its learning ability is not limited by linearly separable problems (Minsky and Papert, 1969, Medin, Wattenmaker, Michalski, 1987). PRN classified rapidly all the 2D problems used in this text with or without noise and with or without redundancy. Its integration rules also make it similar to Random Walk models (Ratcliff, 1978, Ward and McClelland, 1989). Finally, this model makes predictions

sustained by simple mathematical relationships, about two aspects of the behavior, speed-accuracy trade-off and the receiver-operating characteristics.

This model is tailored to predict response times because the notion of time is the primitive of this system. When statistics of extremes are incorporated, most of its mathematical properties are easy to infer. At the same time, once the concepts of associative strength and weigthed sums are put aside, this model becomes quite intuitive.

Further, we saw in section 3 that the model makes very specific predictions for RT distributions. It predicts that each accumulator is Weibull distributed. Hence, having two or more accumulators introduces deviations that can be predicted. We saw (section 5) that the impact of adding noise did increase slightly the probability of observing a LogNormal distribution (Ulrich and Miller, 1993). Yet, the Weibull distribution predominated in a majority of the simulations, a result strikingly similar to the human RT distributions found using a method of contrasts (Cousineau, in preparation).

Because the system needs to know that time is going on even when nothing else is happening, we added time-out units to the system. Although they add dimensions to the inputs (whose sole presence are irrelevant since clocks are always on), they do not make the problems harder to solve. On the contrary, because they represent chronometers that tick the seconds away, they provide positive information that can be used in rules such as “If by now nothing else has happened, you might consider this”. Thus, this system learns to prioritize based on the received evidence, and one of its options is “let’s wait a little more”. When we assumed that clocks could be pressured by speeded external signals, we obtained results typical of speed-accuracy trade-off studies. The model is also consistent with the methods of speed-accuracy decomposition

introduced by Meyer and his colleagues (1988). Therefore, manipulating time constraints allows evaluating empirically some portion of the ***D*** matrix. Conversely, constraints on accuracy of hits and correct rejections resulted in ROC curves that are congruent with empirical ones and in turn gave some clues on the relative size of the thresholds. It is our belief that to uncover the mechanisms of decision, time constraints and accuracy constraints should be studied at the same time, in the same experiment, with the same subjects. SATO and ROC phenomena are but two faces of the same decision process.

Some properties of the parallel race model

The parallel race model has two properties that are very important for psychological plausibility, namely reduction of variability and resistance to partial destruction.

The reduction of variability is a useful property for any psychological model. Suppose that the inputs (or equivalently the units) have a slow conductance rate and a significant amount of variability resulting from noise η (as is the case for neurons, for example). A serial system composed of m stages would end up with a tremendous amount of variability $m \times \eta$. By adding a second unit in parallel with the same purpose as the first one, and considering only the fastest, variability decreases considerably. In some cases, with only two redundant units ($\rho = 2$), variability of the output is reduced to half the variability of the incoming signal; with 3, it is reduced to one third, etc. In general, the resulting variance from ρ redundant inputs is proportional to $\eta / \sqrt[\gamma]{\rho}$, where γ depends on the shape of the variability of each unit (Cousineau et al., in press). This is probably the easiest way to reduce variability (noise) that we know. No special filters or transformations are needed. In fact, it is possible to construct a serial system

composed of *m* stages where each stage is composed of units racing in parallel with variability at the output that is comparable in magnitude with the variability of the inputs.[19] Such a system is said to be in control of the noise and can be extended to a larger number of stages without much decrease in performance resulting from noise.

Resistance to partial destruction also results from the assumption of redundancy: with a large number of redundant channels, a few lose will barely affect the system. Perhaps the only visible manifestation of partial destruction would be a small rise in variability since fewer units are now competing. This property doesn't depend on the nature of the units (neurons or others). They are emergent properties of large collections of random elements where only the fastest are crucial at any given moment.

Serial effects

From a neuroanatomically point of view, the weight of evidences clearly suggests that the brain have a massive parallel architecture. Yet, in cognitive psychology, there are countless box-and-arrow models where certain operations must be performed in a given sequence. Some of these models implicitly assume a central executive who knows when to perform the operation, but this only relegates the notion of time one notch deeper (Van Gelder, 1995). One elegant aspect of the parallel race model is that it does exactly the opposite: time is the front-end of this parallel architecture. Seriality, as we describe next, is one by-product of PRN to accommodate imperfect inputs.

PRN allows faster responses to more complex patterns since it needs to gather a larger amount of evidence. Because the absence of information has no representation in this network,

the various responses in competition must agree to defer their responses a little longer when they might be in presence of embedded patterns.

This results in the most interesting property of all: seriality out of a parallel architecture. The serial effects obtained cannot be schematized appropriately by a box-and-arrow diagram since the B response will occur if response A was not triggered, not **after** response A was discarded. The distinction is subtle but very important since in the latter, response B has no real knowledge whether processing for an A response was completed or not. Further, only training can adjust the delays for the B responses. Finally, this fake seriality may pose problems for those who analyze their data in the context of the additive-factor method (Sternberg, 1969) and the descendant of this method, the mean interaction contrast (Thomas, 2000, Townsend and Nozawa, 1995). Further studies are needed to see whether simple statistics such as the mean interaction contrast and simple properties such as the Miller inequality still hold in the parallel race model (Ulrich and Giray, 1986, Cousineau, 2001). Nevertheless, this model provides a convenient framework, which is at the same time intuitive and highly constrained by mathematical relations.

Strange, did you say?

In retrospect, the parallel race model accomplishments are achieved with only three free parameters. Nevertheless, two of them are quite unusual in psychological models, so they deserve a few comments.

The first parameter is $\mathcal{I} * \mathcal{D}$. It is not common to have a function be treated as a free parameter. The fact that it has no numerical value of its own and is not located on a continuum of function does not interfere with the fact that distribution functions are chosen by modelers (often

with best-fit objectives in mind). The same holds for the parallel race model. Although for large redundancies, the parallel race model predicts that accumulators are Weibull, it does not predict the exact shape γ of these distributions. More information on $\mathcal{I} * \mathcal{D}$ (or taken individually, $\mathcal{I}$ and $\mathcal{D}$) are needed to reduce the number of assumptions. Whether it can be studied empirically is an open question.

The second parameter is ρ. Considering the gigantic number of connections in the brain, it is likely that the same activation will arrive to a neuron through various paths. This intuition is incorporated in the parallel race model. One reason to do this is to increase the number of races while keeping the problem the same. When this is done, the statistics of extreme theorems nicely fit into the picture, providing simple predictions. Yet, ρ also proved interesting in another way because the resistance to partial destruction and the reduction of noise properties are direct consequences of this parameter. Further, Anderson (1994) showed that increasing the number of paths in simple networks can explain the RT-IQ correlation. Indeed, the parallel race model does fit well the empirical data reported when only ρ is manipulated. Hence, more is better. However, a massive dose of redundant inputs right from the start increases slightly the amount of training needed before the network stabilizes. A possible solution, suggested in a different context by Logan, is to suppose that redundancy is not present from the start but rather that extra inputs are recruited as practice on a task goes on. This interesting conjecture makes connection between the learning curves of both mean and standard deviation and the shape of RT distribution (as shown originally by Logan, 1988, 1992; also see Cousineau and Lefebvre, in preparation, Takane, Oshima-Takane and Shultz, 1999).

Finally, the noise parameter η might in contrast seem more conventional. It is seen from times to times in various models. However, it is important to see that the noise as we define it in this paper does not affect the magnitude of numerical values (such as strength or weight) but always introduces delays in the acquisition of information. It is not a "How much" but rather a "When then". It is always an increase because information can only be delayed, not speeded by noise (congruent with entropy).

Taken together, η and ρ more or less describe the residual noise at the output of the system. As such, they should always be studied in conjunction. However, this is not an easy task. For example, reducing luminance contrast is generally modeled by an increase in noise (Palmer, 1994, 1998). However, since it reduces the amount of photons reaching the eye, should not it be seen as a reduction in redundancy? A more formal definition of these factors are therefore needed.

Thorpe's paradox and the learning rule

In light of Thorpe's paradox mentioned in the introduction, an interesting question emerges: if only the first few milliseconds of the neural activities is used, why does the activation last so long? In the parallel race model, only the moment when the activation turns on is relevant. It could shut down immediately after. Therefore, it is not relevant for decision-making, but it might be relevant for learning.

The learning rule discussed here is not of the delta rule type for two reasons. First, there are no hidden layers and the attribution of errors is easy: the fastest are the guilty. Second, a difficult information to assess (and a much more elusive question) is how to know if an error

occurred or not. The solution in supervised network was to assume that a teacher was present. Nevertheless, what can a teacher do? It is doubtful that it can reach the specific faulty unit. It is more reasonable to suppose that it can stop the incorrect behavior. Thus, it seems that the teacher can stop a behavior and interrupt immediately what would otherwise be an excessively long period of high activation. In retrospect, this long period of high activation might serve as a confirmation: If it was not interrupted prematurely, then it was not an erroneous behavior. In other words, neuron activities may have two dimensions: the moment it went on, and the duration of the activation. These two dimensions in turn may be responsible for two distinct behaviors: decision and learning. This view could solve some of the controversy surrounding the notion of teachers in supervised learning.

In a different vein, extended periods of activation could also be relevant to reinforcement. For example, keeping a neuron activated for some time might be used to detect whether other neurons can be activated in the same way and thus joined to the system. Equivalently, the conjoint activation of the input and the output might facilitate the creation of new connections between the two. In either case, they are other forms of the redundancy conjecture.

New mathematics, old asymptotic theories

In order to develop the mathematics of race models, we redefined the dot product, based on the joining function and aggregate functions we chose. Although there is no precedent to this in the literature to my knowledge, it leads to a major simplification for predicting RT distributions since it has direct connections with asymptotic theories. I will show some examples here using three distinct asymptotic theories.

The simplest case is related to the Perceptrons. Let $\boldsymbol{I}$ and $\boldsymbol{W}$ be the input vector and the weight matrix describing a given network. The output of such a network is simply $\mathbf{O} = \mathbf{I}_{\left(\begin{smallmatrix}\times\\ \Sigma\end{smallmatrix}\right)}\mathbf{W}$ where the standard dot product is used. Let further assume that $\boldsymbol{I}$, $\boldsymbol{W}$, or both can be noisy, such that all $\boldsymbol{I_i}$ and $\mathbf{W}_{ij}$ are independent and identically distributed with distribution functions $\mathcal{I}$ and $\mathcal{W}$ respectively. If the mean $m := E(\boldsymbol{I}^P \times \boldsymbol{W})$ and variance $s^2 := Var(\boldsymbol{I}^P \times \boldsymbol{W})$ exist for a given pattern $\boldsymbol{I}^P$, then according to the law of large number (Feller, 1957), the asymptotic distribution of sums $L_\Sigma(\mathcal{I} \hat{} \mathcal{W}_j)$ is normally distributed with mean *m* and standard deviation *s*. Of course, this result is based on the extra condition that the number of inputs is large (larger than 30) because the law of large number is an asymptotic theory too[20]. In general, models based on the standard dot product are called additive models (or linear models, in relation with linear algebra).

A second class of non-linear models is the multiplicative class of models (West and Schlesinger, 1990). Decisions are obtained by chaining processes. Therefore, they are well suited for explanation involving processing steps such as Encoding-Decision-Motor response (but in order for asymptotic theories to apply, the number of steps has to be much larger than three). Following the presentation of Ulrich and Miller (1993), if we let $\boldsymbol{I}$ be a random vector of inputs $\{\boldsymbol{I_i}\}$ and $\boldsymbol{G}$ be a matrix $\{\boldsymbol{g_i}\}$ composed of power functions of the type t^{b_i} , then the response of this system is given when $\boldsymbol{O}^P = \boldsymbol{I}^P{}_{\left(\begin{smallmatrix}\times\\ \Pi\end{smallmatrix}\right)}\boldsymbol{G}$ exceeds a criterion χ. This is a cascade models (McClelland, 1979) and, as shown by Ulrich and Miller, under this formulation, the RT distribution is LogNormal.

A third class of models is the class of competitive models, based on the minimum operator such as the parallel race model. The Weibull is the asymptotic distribution for these models. In the above, we concentrated only on an additive version $\{\boldsymbol{I_i} + \boldsymbol{D_{ij}}\}$. However, we also explored a multiplicative version of this race model, where the output was given by $\boldsymbol{O}^P = \boldsymbol{I}^P \underset{\left(\underset{Min}{\times}\right)}{} \boldsymbol{D}$. As shown in Cousineau et al., there is no difference in the RT distribution predicted, which leads to the temporary conclusion that the joining operator used may not have much impact on learning.[21]

Overall, we conjecture that there may exist three distinct classes of learning networks, the additive (or linear) models, the multiplicative (or cascade) models, and the competitive (or race) models. These classes are easily identified by the type of dot product used in the integration rule, $\left(\underset{\Sigma}{-}\right)$, $\left(\underset{\Pi}{-}\right)$, and $\left(\underset{Min}{-}\right)$ respectively. We put an underscore since it may be that the joining operator is irrelevant. These classes of models have well-known asymptotic behavior since under some general assumptions, $\left(\underset{\Sigma}{-}\right)$ is normal, $\left(\underset{\Pi}{-}\right)$ is LogNormal, and $\left(\underset{Min}{-}\right)$ is Weibull (at least when the underscore represents convolution, production, and convolution or production respectively). Future work is needed to see if very general predictions made by these models can be contrasted using general properties (such as the Miller inequality, Ulrich and Giray, 1986, Cousineau, 2001).

The word of the end

Ractliff, Van Zandt, and McKoon (1999), who explored extensively RT distributions and phenomena such as SATO, remarked: “The findings […] show that the long tradition of

reaction-time research and theory is a fertile domain for development and testing […] how decisions are generated over time." (p. 261). I would like to stress this point even more: maybe time is the fundamental unit of cognition. At least, I showed in this text that, in principle, it could be so. Whether human cognition is based on this class of model remains an open question, but finding which metric underlies cognitive processes would help to develop tests that are more sensitive.

AUTHOR NOTES

We would like to thank M. Arguin, A. Criss, G. Lacroix, C. Lefebvre, G. D. Logan, R. M. Nosofsky, D. Saumier, R. M. Shiffrin, J. Townsend and an anonymous reviewer for their useful comments on earlier versions of this paper. We also want to thank Y. Lacouture, P. Murray, and M. Wenger for suggesting applications for the race model. Parts of this work were presented to the Conférences du Groupe de Recherche en Neuropsychologie. Université de Montréal, February 2001 and to the Thirty-third Annual Meeting of the Society for Mathematical Psychology, Kingston, August 2000. Sample programs can be found at http://Prelude.PSY.UMontreal.CA/~cousined/papers/07-PRM/ . This research was supported by the Fonds pour la formation de chercheurs et l'aide à la recherche, Établissement de nouveaux chercheurs, 2002-NC-72532.

Request for reprint should be addressed to Denis Cousineau, Département de psychologie, Université de Montréal, C. P. 6128, succ. Centre-ville, Montréal (Québec) H3C 3J7, CANADA, or using e-mail at denis.denis.cousineau@umontreal.ca.

FOOTNOTES

[1] Another class of model that generalizes the SDT is the multidimensional SDT, sometimes called the noise models (Palmer, 1994, Eckstein, Thomas, Palmer, Shimozaki, 2000, Maddox, Ashby, 1993). They assume that a fixed number of samples are taken from the stimuli.

[2] A classic example of a race model with a single accumulator is given by a chain (Weibull, 1951) where the overall strength of the chain lies in the strength of its weakest link. However, this example is confusing because the chain is organized serially, one link after the other. It is important to understand that it is a parallel phenomenon since the chain will break as soon as one of its links will. In this situation, the distribution of the weakest link accurately describes the distribution of the tensions at which the chains break.

[3] Because the weights are partly determined by training, we could call them "partly free parameters". To test distinct models, it is necessary to know the number of free parameters using for example the Likelihood ratio test (Myung, 2000, Grünwald, 2000). At this time, no one knows to how many "fully free parameters" corresponds, say, a hundred "partially free parameters" This question is open to further research.

[4] If you need to think in terms of activation strength, then consider that all the connections have an equal strength of one.

[5] Hidden layers will not be studied in this paper.

[6] A variation where one input can fill more or less than one slot will not be considered here. See Laberge, 1962.

---

[7] The average is not the best measure in a parallel model (if $\rho > 1$) and a more sensible value would be the characteristic smallest value (Gumbel, 1958), but our purpose here is to obtain a value that reflects the manipulation of noise and redundancy for graphical purpose only.

[8] To be consistent with the "Serial Poisson race model" label, we should call PRN the "Parallel anything race model". Indeed, the above assumptions and the following results are not based on a specific distribution function. Any functions that satisfy the criterionC_1 and C_2 presented in Appendix C predicts a Weibull distribution. In that sense, the PRN is a more general model.

[9] And these equivalences might explain why Signal Detection theory (Palmer, 1998, Eckstein, 1998), Reduction of information theory (Haider and Frensch, 1996, 1999) and Diffusion models (Ratcliff, Van Zandt., McKoon, 1999) are so successful.

[10] I would like to thank some of my friends for suggesting this representation.

[11] At this time, I couldn't derive a proof of convergence toward the optimal solution, contrary to many neural networks where such a proof exists (Kohonen, 1984). One reason is that the non linear algebra operator $\sim$ is new and not related to derivatives. A gradient descent proof is thus impossible for PRN.

[12] In the following, we use the tilde to mean "is distributed as".

[13] Although not shown, it does not take longer for the parallel race model to learn an AND problem than a XOR problem.

[14] We note that this network has no tolerance to errors: learning occurs on every error. Ultimately, after an infinite amount of training, it would make infinitely few mistakes.

[15] Logan (1992) tested this same prediction involving RT distribution and RT learning curve in the context of his Instance-Based theory of automaticity.

[16] The trouble with this prediction on noise η is that physical measures of noise are not likely to be linearly related to psychological measures of noise (Lu and Dosher, 1998, West and Shlesinger, 1990, Link, 1992).

[17] This model does not have a specific mechanism for tolerance to errors. It therefore aimed at zero errors (expressed by an infinite θ) which wil never be attained. However, we trained the simulations long enough that θ should be reasonably small.

[18] Interestingly, that is exactly the changes of value we had to perform in Section 1 when going from an On-Off representation (1 or 0) to a time-domain representation (0 or ∞).

[19] The question here is to find a solution to

$$\eta_m \approx m\left(\frac{\eta_0}{\sqrt[\gamma]{\rho}}\right) \overset{?}{\leq} \eta_0$$

where η_m is the noise at stage m. In the case where $\gamma = 1$, one obvious solution is $m > \rho$. Therefore, with a system whose "width" is larger than its "depth", a race model with m stages has a stable amount of variability at the output.

[20] By analogy with Appendix C, we can call the two above conditions criterion C_1^{Σ} and C_2^{Σ}. It seems that many asymptotic theories can be satisfied with as few as two criteria.

---

[21] Since we are talking about variations, let me mention one more. We tested a different learning rule where, contrary to the present version in which delays were increased if a unit false-alarmed, delays were decreased if a unit missed. Again, this variation did not affect learning.

Appendix A:

Deterministic solution to some simple problems

The purpose of this appendix is to propose simple problems, that is, situations where the PRM must make correct output to a given input. Although PRM is relatively simple, it is our experience that people acquainted with classic connectionist network have difficulties understanding PRM because some of the basic concepts, although similar looking, have in fact the opposite meanings.

Five different problems are explored: the Detection problem, the 1D problem, the Identification problem, the AND problem and finally, the XOR problem. Figure A.1, left column, illustrates the problem spaces. The dimensions represent the value of a certain attribute of the stimuli. Here, attributes are all binary, which can be implemented as present or absent.

Insert Figure A.1 about here

When working with binary features, it is convenient to think in terms of zeros and ones, and indeed, the problem spaces in Figure A.1 are organized in that way. However, when presenting a stimulus to the network, what is crucial is whether the information on the dimension has arrived (i.e. is present) or has not arrived yet (i.e. is not present). As such, activated units receive an input right away (at time $t = 0$) whereas inactivated units never receive an input (will be activated after an infinite amount of time, $t = \infty$). This distinction is very important since what is fed into the network are times at which the activation occurs, not its strength. Therefore, instead of thinking in terms of 0 and 1 (absent, present), one should think in terms of ∞ and 0 (never vs. sudden).

All the toy problems presented here require at least one time-out unit. For simplicity's sake, we assume that these units are always immediately on, so that there activation times are all zero.

We deliberately ignored variability in the moments of activation. It is an exercise of logic to find the optimal solutions. The learning rule is also not considered in this appendix.

Example 1: The Detection problem

We first study a simple problem that we call the Detection problem. This problem is composed of 1-dimensional inputs, and the task of the system is to decide whether this dimension is on by making an A response or off with a B response. Figure A.1(a), left part, shows the problem space.

In terms of connections, if the input gets activated, it should be conveyed to the output A with no delays. This connection should be immediate and thus receive a value of zero (the smallest delay possible on such a channel). In addition, this single piece of information is sufficient for an A response, so the size of the accumulator A must be 1. Because this same information is totally irrelevant to response B, it should never reach the B accumulator, so there should be no connection between the input and the response B. Instead, the accumulator B should rely on a time-out unit. However, the delay between the activation of the clock and the moment it reaches the B accumulator cannot be zero. If such was the case, in the presence of a signal on the input, both accumulator A and B would be triggered at the same time, and the tie would make the response ambiguous. The proper solution is to defer the clock signal a little, in case an input signal is present. Thus, there should be a small delay $t > 0$ between the clock signal and the response B.

Figure A.1(a), right column illustrate the solution using a matrix of delay and a vector of accumulator sizes.

Example 2: The 1D problem

In this problem, inputs are composed of two dimensions (e. g. features) which can be either present or absent. The correct response is A if dimension two is present. This problem is called a one-dimensional (1D) problem because only one dimension is relevant. The presence or absence of the other dimension is totally uninformative. Figure A.1(b), left column, shows the problem space.

To address a specific connection, we will use the D_{ij} notation, where D is the matrix of connections, representing the delays between an input and an accumulator, i is the input number (1, 2, or 🕑 for the clock unit) and j is the response output, either the accumulator A or B in this example. Thus, $D_{🕑A}$ is the delay in the connection between the clock and the first accumulator. Further, let Kj represents the size of the j^{th} accumulator.

Because the input on dimension 2 is fully diagnostic of response A, it should be conveyed immediately to the accumulator A, and a response can occur immediately. Thus, $D_{2A} = 0$. In addition, this information is sufficient so that the accumulator for response A is 1 ($K_A = 1$). The model should answer response B if and only if nothing is on the second channel, that is, if no response has been made at time 0. Therefore, D_{1B} must be larger than D_{2A}. Call this delay $t > 0$. Finally, if nothing is presented, the clock should also be able to generate a B response. Thus, $D_{🕑B} = t$ as well. The other connections are irrelevant and can be disconnected (∞). Also note that

the presence of a signal on input 1 being uninformative, we could rely on the clock only and set D_{1B} to ∞.

Right column of Figure A.1(b) shows the solution in terms of delays and thresholds.

Note that if an input is present on the first channel, it can either be response A (pattern {1, 1}) or response B (pattern {1, 0}). This is an example of an embedded pattern. Embedded patterns will be discussed in the text.

The solution shows that response B will be performed if and only if response A was not done previously. In a certain sense, B must agree not to compete with A in case where the input on dimension 2 is activated. This form of collaboration is in direct opposition with classic neural networks where the inputs are in competition.

Example 3: The Identification problem

The identification problem is an example where the system must answer A if dimension 1 is on and B if dimension 2 is on (the case where both are on is assumed non existent). When neither dimension 1 nor dimension 2 is present, the system must make a third response, called C. Figure A.1(c), left part, shows the problem space.

Apart from the third accumulator, this example is simple to solve. As soon as something is perceived on dimension 1 or 2, a response A or B, respectively, must be made. Thus, the delays D_{1A} and D_{2B} are set to zero. However, the clock is responsible of the response C and to avoid ambiguity with A and B, it should be delayed a little. $D_{\text{Clock C}} = t > 0$.

Figure A.1(c), right part, shows the solution matrix.

Example 4: The AND problem

The AND problem requires that both inputs be present for the system to respond A. All other patterns of input must result in response B. See Figure A.1(d) for the problem space.

The correct solution requires that the inputs be transmitted rapidly to the accumulator A. However, the two must be present. This requires that the threshold K_A be 2. If the conjunction is not present, any single input will not fill the accumulator. However, it can trigger a B response and should, a little while latter to avoid tie. A time-out unit can also trigger a response B later. Figure A.1(d), right part, shows the solution matrix.

Note that since the correct response is the same whether there is one input or none (response B), we can use only the clock to respond B and shut down the other two connections (D_{1B} and D_{2B} to ∞) without any change in performance or accuracy.

Example 5: The XOR problem

The XOR problem is a more complex problem with an important history. Indeed, Minsky and Papert (1969) showed that the XOR problem is a linearly non separable problem (there exist no single boundary between the As and the Bs in the problem space) and could not be solved by two-layered connectionist network. As this example shows, a hidden layer is not necessary inside a PRN architecture.

The XOR rule is to respond A if both or none of the input are present and B otherwise. Figure A.1(e), left part, shows the problem space in the usual format.

The correct solution is to react rapidly if two inputs are present (D_{1A} and D_{2A} equal zero), and to be cautious if there seems to be no input (delays t_2 for time-out units). However, when only one input is present, accumulator B should react faster than the clock and slower than

accumulator A, to avoid simultaneous responses. So delays should be set at t_1, where $t_1 < t_2$. In this last case, since there is one actual piece of evidence collected so far, the system is rapidly confident that it is not a situation where no input are present. However, the system is still uncertain about the presence or absence of the second input. The purpose of the time-out unit is to wait, but not to wait too long. Also note that the threshold for a response A is 2 since a conjunction is required. Therefore, to make a response A when nothing is present, two activations are required, leading to the consequence that two clocks are absolutely needed. This is an example where redundancy is necessary.

In the priority learner representation, the rules R_1 to R_3 could be:

R_1: Say "A" as soon as there is two evidences;

R_2: Or else, say "B" later (at time t_1) if one evidence;

R_3: Or finally, say «A» later (at time t_2).

In R_1, evidence will come from the inputs whereas for R_3, evidence are likely to come from the clocks. Because in the XOR problem, R_1 and R_3 addresses the same output A, so must the accumulator size be identical. This is why this particular problem requires two clocks. The previous problems could all be learned with only one clock although more than one clock does not hinder performance (the extra clocks might make learning slower).

As seen, because the system can order the responses temporally, there is no need for hidden units. Note that adding clocks does not change the problem space to a linearly separable one since the clocks are always on.

Appendix B:

Redefining the dot product

The whole point of race models is to find which channel transmitted the information to the output first. Noise across the layers is assumed additive. Thus, a combination of sum and minima is required. It turns out that a simple generalization of the inner product allows accommodating the need of the race model. Further, by adopting this inner product representation, it allows for a very compact notation that naturally extends to vectors of random values (as seen in Appendix B) and also allows for comparisons with classic connectionist networks that use the standard inner product in their transmission rule.

Redefining the dot product on vectors

The standard inner product is defined on one-dimensional vectors U and V by:

$$U \bullet V := \sum_{i=1}^{n} U_i \times V_i$$

where n is the size of both vectors. One can note in the above equation that there are two operators at work: a joining operator ($\times$) and an aggregate operator (Σ). The role of the joining operator is to relate a couple $\{U_i, V_i\}$, defined in a domain $\Re^2$ (we will assume the real domain in this text) with a single value in the $\Re$ domain. The product $\times$ does constitute an appropriate joining operator in standard situations.

Having joined all the couples from the two vectors, we obtain a single list of values. This list needs to be reduced to a summary value and that is the role of the aggregate function (defined on the domain $\Re^n \rightarrow \Re$).

A simple example illustrates the role of the two operators in turn. Let

$$U = \begin{pmatrix} 3 \\ 5 \\ 6 \\ 8 \\ 9 \\ 1 \\ 8 \\ 1 \end{pmatrix} \text{ and } V = \begin{pmatrix} 6 \\ 5 \\ 30 \\ 40 \\ 27 \\ 2 \\ 8 \\ 5 \end{pmatrix}$$

First, the join operator is applied on the two vectors, yielding:

$$U \times V = \begin{pmatrix} 3 \\ 5 \\ 6 \\ 8 \\ 9 \\ 1 \\ 8 \\ 1 \end{pmatrix} \times \begin{pmatrix} 6 \\ 5 \\ 30 \\ 40 \\ 27 \\ 2 \\ 8 \\ 5 \end{pmatrix} = \begin{pmatrix} 18 \\ 25 \\ 180 \\ 320 \\ 243 \\ 2 \\ 64 \\ 5 \end{pmatrix}$$

and then, the aggregate operator follows:

$$\Sigma \begin{pmatrix} 18 \\ 25 \\ 180 \\ 320 \\ 243 \\ 2 \\ 64 \\ 5 \end{pmatrix} = 857$$

To avoid confusion, we can use a more explicit notation $U_{\binom{\times}{\Sigma}}V$ to represent the standard inner product $U \bullet V$, based on the product for the join operator and the summation for the aggregate operator. The non-standard dot product used for race models is thus in this representation denoted by $U_{\binom{+}{Min}}V$, where **Min** is the minimum operator. It is defined by:

$$U_{\left(\begin{smallmatrix}+\\ Min\end{smallmatrix}\right)}V := \underset{i=1}{\overset{n}{Min}} U_i + V_i \qquad \text{(B.1)}$$

Here's an example with the same vectors as above, but using the $U_{\left(\begin{smallmatrix}+\\ Min\end{smallmatrix}\right)}V$ operation:

First, the joining operation using sum:

$$U + V = \begin{pmatrix}3\\5\\6\\8\\9\\1\\8\\1\end{pmatrix} + \begin{pmatrix}6\\5\\30\\40\\27\\2\\8\\5\end{pmatrix} = \begin{pmatrix}9\\10\\36\\48\\36\\3\\16\\6\end{pmatrix}$$

followed by aggregation done by locating the smallest of the list:

$$Min\begin{pmatrix}9\\10\\36\\48\\36\\3\\16\\6\end{pmatrix} = 3$$

As can be seen, the results differ markedly.

On some occasion, it is not the first minima that is required, but the second or third, etc. Let us define the K^{th} minimum operator Min_K as a function that returns the K^{th} smallest value of a vector-list. If $K = 1$ then this operation reduces to the standard minimum operator Min. If $K > 1$ and K smaller than the length of the list, Min_K is defined recursively as $Min_K\{list\} := Min_{K-1}\{list - Min\{list\}\}$, where the subtraction removes the smallest element from the vector-list.

In the above example, $U_{\left(\begin{smallmatrix}+\\ Min_2\end{smallmatrix}\right)}V$ would yield 6.

Inner product of a vector and a matrix

In the race model, we use a vector to represent the input. The connections relating all the n inputs to the m outpt are stored in an $n \times m$ matrix. Because each output is an accumulator waiting for the K^{th} fastest channels, we also store in a vector of size m called K the accumulator sizes.

We can redefine the inner product as we did for the dot product. Let $U_{\left(\begin{smallmatrix}+\\ Min_K\end{smallmatrix}\right)} A$ be defined on a vector U_n and a matrix $A_{n \times m}$ using sum as a joining operator and minima as an aggregate operator. As usual, the result will be a vector of size m. The operation is given by:

$$U_{\left(\begin{smallmatrix}+\\ Min_K\end{smallmatrix}\right)} A := \{U_{\left(\begin{smallmatrix}+\\ Min_{K_j}\end{smallmatrix}\right)} A_{\bullet j} \ , \ \ j = 1..m\} \qquad \text{(B.2)}$$

That is, for each column of A, we perform an inner product with U (as defined in Eq. B.1), using the K_J^{th} accumulator size.

As an example, given

$$U = \begin{pmatrix} 1 \\ 4 \\ 3 \end{pmatrix}, A = \begin{pmatrix} 2 & 1 \\ 5 & 4 \\ 7 & 3 \end{pmatrix} \text{ and } K = \begin{pmatrix} 1 & 2 \end{pmatrix},$$

The result of $U_{\left(\begin{smallmatrix}+\\ Min_K\end{smallmatrix}\right)} A$ is given by first applying the joining operation column wise:

$$U + A = \begin{pmatrix} 3 & 2 \\ 9 & 8 \\ 10 & 6 \end{pmatrix}$$

then aggregating by finding the Kth minimum, performed on each of the m column:

$$Min_K \begin{pmatrix} 3 & 2 \\ 9 & 8 \\ 10 & 6 \end{pmatrix} = \left(Min_{K1} \begin{pmatrix} 3 \\ 9 \\ 10 \end{pmatrix}, Min_{K2} \begin{pmatrix} 2 \\ 8 \\ 6 \end{pmatrix} \right)$$

$$= \left\{ Min_1 \begin{pmatrix} 3 \\ 9 \\ 10 \end{pmatrix}, Min_2 \begin{pmatrix} 2 \\ 8 \\ 6 \end{pmatrix} \right\}$$

$$= (\,3, 6\,)$$

When the context is not misleading, we will use the shorter notation $\sim$ to stand for the redefined dot product $U_{\left(\substack{+ \\ Min_2}\right)}V$ or the redefined inner product $U_{\left(\substack{+ \\ Min_K}\right)}A$.

Appendix C:

Details from the statistics of extremes

The simplest race model consists in a single accumulator connected to inputs, a large number of them (say ρ) being redundant. A response is given when the accumulator is filled. Thus, the unit that fills the last slot determines the response time ***t***. This is usually conveyed in the equation $\boldsymbol{RT} = Min\{\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_\rho\}$ where the $\boldsymbol{t}_i$ are random variables distributed with distribution function $\mathcal{T}$. It is shown that if:

C_1: $\mathcal{T}$ has a lower bound (at a point, say $\alpha(\mathcal{T})$), and

C_2: the left-end tail of $\mathcal{T}$ approximates a power curve (i. e., satisfying a criteria of this type: $\lim_{h\downarrow 0} \frac{\mathcal{T}(hx + \alpha(\mathcal{T}))}{\mathcal{T}(h + \alpha(\mathcal{T}))} = x^\gamma$, $\gamma \geq 0$, Gnedenko, 1943[22]),

then the distribution of minima has an asymptotic stable shape (Fisher and Tippet, 1928, Gumbel, 1958, Galambos, 1978). Further, this shape is given by a Weibull distribution (Cousineau and Larochelle, 1997).

This reasoning depends on ρ increasing to ∞, but Cousineau, Goodman and Shiffrin (in press) showed that ρ could be relatively small (smaller than 100, and in some cases, smaller than 20).

The two criteria above are satisfied by many known distributions including the Exponential and its related distributions and by the Uniform distribution. One important exception however is the Normal distribution, which does not have a lower bound (thus violating

C_1). In addition, it is shown that this argument is still true if, instead of waiting for the fastest, we wait for the K^{th} fastest before triggering a response, as long as K is finite, small and independent of ρ (Galambos, 1978, Leadbetter, Lindgren and Rootzén, 1983).

In the parallel race model, we have an input vector, denoted ***I*** and a connection matrix, denoted ***D***.

Let $\mathcal{I}$ and $\mathcal{D}$ be the distribution functions for each element composing the random vectors ***I*** and ***D***, respectively. Using a similar notation to the one used to compute the output **O** in Eq. (1.2), we define $\mathcal{O}$ as the response times distribution produced by the model. We assume that the effects of ***I*** and ***D*** are additive (although it could as well be multiplicative; see the general discussion). The relation between the distribution function of the inputs ***I***, the delay ***D***, and the output **O** is thus given by:

$$\begin{aligned} \mathcal{O} &= L(\mathcal{I}_{\left(\underset{Min_K}{*}\right)}\mathcal{D}) \\ &= L_{Min_K}(\mathcal{I} * \mathcal{D}) \end{aligned} \qquad \text{(C.1)}$$

where * is the convolution operation implementing the effect of adding random variables (Cramer, 1946), and L is defined as the asymptotic distribution. Because the aggregate operator Min_K defines in what respect L is asymptotic, we can migrate Min_K outside the parenthesis. In our case, L_{Min_K} is the asymptotic distribution with respect to the K^{th} minimum. Stated differently, this equation returns the asymptotic distribution of the convolution $\mathcal{I} * \mathcal{D}$ with respect to minima.

In order to solve O (Eq. 1.3), we need to assume some characteristic for the distribution functions $\mathcal{I}$ and $\mathcal{D}$. A first obvious constraint is that a signal cannot arrive before it is emitted. This signifies that the criterion C_1 is satisfied ($\alpha(\mathcal{I}) \geq 0$ and $\alpha(\mathcal{D}) \geq 0$). This also eliminates the Normal distribution; it is an unwise choice for modeling race models. The second assumption concerns the rate of increase of the distribution functions. Although many possibilities exist, we will adopt distributions with a power curve increase in probability because i) they satisfy the criterion C_2, making the solution of O possible; ii) many common and well-known distributions are members of this family, such as the Exponential and the Uniform distributions.

We proved in Cousineau and al. (in press) that convolving two distribution functions satisfying C_1 and C_2 results in a distribution function that also satisfies these criteria. Therefore, Eq. C.1 has a solution. One accumulator race network's distribution is the Weibull distribution.

We believe the above notation is useful since it can easily be extended to other asymptotic distributions. For example, when using the standard dot product (noted $U_{\left(\begin{smallmatrix}\times\\ \Sigma\end{smallmatrix}\right)}V$), the asymptotic distribution, assuming large random vectors with distributions $\mathcal{U}$ and $\mathcal{V}$ is given by:

$$\begin{aligned} O &= L(\mathcal{U}_{\left(\begin{smallmatrix}\times\\ \Sigma\end{smallmatrix}\right)}\mathcal{V}) \\ &= L_{\Sigma}\ (\mathcal{U}\hat{}\mathcal{V}) \end{aligned}$$

where ^ is the production operation, meaning that each pairs of random values in $\boldsymbol{U}$ and $\boldsymbol{V}$ are multiplied. L_{Σ} thus stand for the asymptotic distribution with respect to summation. Now, the

result is well known to be a Normal distribution by appeal to the *Central Limit Theorem* (Cramér, 1947).

Appendix D:

Imposing deadlines and speed-accuracy decomposition

In this section, we introduce sharp time constraints. Thus, the response C will in fact corresponds to a "guess" response because the clock interrupts the trial too soon. We assume that the environment imposes the clock. Experiments imposing deadlines are called Signal-to-Response (StoR) experiments. They have a peremptory signal that mandate a response as soon as possible (Reed, 1976).

Following Meyer, Irwin, Osman and Kounios (1988), we note that correct response can result either from an adequate processing of the input (when the signal **S** is faster than the clock ***C***) or a correct guess (when **S** > ***C***). If no signal-to-response occurs in a trial, there is no guess. Therefore, let us write $\mathbf{T}_n$ the response time obtained on no-StoR, $\mathbf{T}_s$ when a StoR occurs with a certain SOA, and $\mathbf{T}_{gs}$ the response time of a guessing. Both $\mathbf{T}_n$ and $\mathbf{T}_s$ are observable but $\mathbf{T}_s$ represents a compound of $\mathbf{T}_n$ and $\mathbf{T}_{gs}$ where $\mathbf{T}_{gs}$ is not observable and correspond to the distribution of the clock $\boldsymbol{D}_{\textcircled{t}\mathrm{C}}$ (presumably, $\mathbf{T}_{gs} = \mathbf{G} + \mathrm{SOA}$). Meyer et al. suggested a decomposition method so that the distribution of the clock can be obtained. In the following, we apply this method to parallel race model and show that it correctly recovers the clock distribution.

First, note that

$$P(\mathbf{T}_n > c) = P(\mathbf{S} > c) \text{ and } P(\mathbf{T}_{gs} > c) = P(\mathbf{C} > c)$$

where, for simplicity, **S** represents the processing time of the signal $\boldsymbol{I}_1 + \boldsymbol{D}_{1A}$, and $\boldsymbol{C}$ denotes the processing time of the clock $\boldsymbol{I}_{\textcircled{\scriptsize t}} + \boldsymbol{D}_{\textcircled{\scriptsize t}C}$. During a normal trial, the clock is not involved since there is no StoR ($\boldsymbol{D}_{\textcircled{\scriptsize t}C} = \infty$). Under the parallel race model,

$$P(\mathbf{T}_s > c) = P(\mathbf{T}_n > c \cap \mathbf{T}_{gs} > c) = P(\mathbf{T}_n > c)P(\mathbf{T}_{gs} > c)$$

Because all the inputs and the clocks are independent, we can use a product of two probabilities. It follows that

$$P(\mathbf{T}_{gs} > c) = \frac{P(\mathbf{T}_s > c)}{P(\mathbf{T}_n > c)} \qquad \text{(D.1)}$$

That is, pure guess distribution is deduced from a ratio of two observed distributions and yields exactly the clock distribution. Figure D.1(a) illustrates the relation between $\mathbf{T}_n$ and $\mathbf{T}_{gs}$, where SOA occurred at 2 arbitrary units of time. The sooner the distribution of the clock starts, the steeper the distribution of $\mathbf{T}_s$ becomes. In Figure D.1(b), we imposed three StoR SOA to the system by setting $\boldsymbol{D}_{\textcircled{\scriptsize t}C}$ positions. We used the values 1, 2, and 3 in arbitrary units of times. Using Eq. D.1, we recover perfectly the distribution of the clocks. This graph is strikingly similar to the one obtained by Meyer and al. on human subjects, lending support to the parallel race model.

Insert Figure D.1 about here

The final question concerns which response is guessed. According to Meyer et al., some form of sophisticated guessing can occur. In the parallel race model, this is achieved by assuming that partial information can be used even if the accumulator is not full. In the above example, it is of course impossible since the accumulator size is 1: either it is already full or no information at all exists. However, if threshold is much higher than 1 (given that ρ is also larger than 1), then partial information can be available. If guessing is based on the proportion of slots

filled at a certain time, we can then make the following plot of $F_n(\ \mathrm{E}(\mathbf{T}_s \mid \boldsymbol{D}_{\text{⏱C}})\ )$ as a function of $\mathrm{E}(\mathbf{T}_s \mid \boldsymbol{D}_{\text{⏱C}})$ where the first term gives the pdf of the time to fill the accumulator under a normal trial computed when the average time of a StoR trial is attained. Multiple points are obtained by varying SOA ($\boldsymbol{D}_{\text{⏱C}}$). Figure D.1(c) gives the result, again compatible with the Meyer et al. data (their Figure 21).

REFERENCES

Adams, D. R., Myerson, J. (1999, November). A multilayered connectionist model that predicts general properties of speeded performance. 40th annual meeting of the Psychonomic society, Los Angeles.

Anderson, B. (1994). Speed of neuron conduction is not the basis of the IQ-RT correlation: Results from a simple neural model. Intelligence, 19, 317-323.

Anderson, R. B., Tweney, R. D. (1997). Artifactual power curve in forgetting. Memory & Cognition, 25, 724-730.

Arguin, M., Bub, D. (1995). Priming and response selection processes in letter classification and identification tasks. Journal of Experimental Psychology: Human Perception and Performance, 21, 1199-1219.

Audley, R.J., Pike, A.R. (1065). Some alternative stochastic models of choice. British Journal of Mathematical and Statistical Psychology, 18, 207-225.

Bundesen, C. (1990). A theory of visual attention. Psychological Review, 97, 523-547.

Burbeck, S. L., Luce, R. D. (1982). Evidence from auditory simple reaction times for both change and level detectors. Perception and Psychophysics, 32, 117-133.

Caudill, M., Butler, C. T. (1992). Understanding neural networks: Computer explorations. Cambridge, Mass.: MIT press.

Changeux, J.-P. (1983). L'homme neuronal. Paris: Fayard.

Chaplain, R. A. (1979). Metabolic control of neuronal pacemaker activity and the rhythmic organization of central nervous functions. Journal of Experimental Biology, 81, 113-130.

Cohen, J. D., Dunbar, K., McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the stroop effect. Psychological Review, 97, 332-361.

Colonius, H. (1988). Modeling the redundant signals effect by specifying the hazard function. Perception and Psychophysics, 43, 604-606.

Colonius, H. (1990). Possibly dependent probability summation of reaction time. Journal of Mathematical Psychology, 34, 253-275.

Cousineau, D. (2001, July). Redundancy conjecture and super-capacity rate of increase. Society for Mathematical

Psychology Annual Meeting, Providence.

Cousineau, D., Goodman, V., Shiffrin, R. M. (in press). Extending statistics of extremes to distributions varying on position and scale, and implication for race models. Journal of Mathematical Psychology.

Cousineau, D., Larochelle, S. (1997). PASTIS: A Program for Curve and Distribution Analyses. Behavior Research Methods, Instruments, & Computers, 29, 542-548.

Cramér, H. (1946). Mathematical methods of statistics. Princeton: Princeton University Press.

Diederich, A. (1992). Probability inequalities for testing separate activation models of divided attention. Perception and Psychophysics, 52, 714-716.

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., 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, Volume I. New York: John Wiley and son (2nd edition).

Fisher, R. A., Tippett, L. H. C. (1928). Limiting forms of the frequency distribution of the largest or smallest member of a sample. Proceedings of the Cambridge Philosophical Society, 24, 180-190.

Fournier, L. R., Eriksen, C. W., Bowd, C. (1998). Multiple-feature discrimination faster than single-feature discrimination within the same object?. Perception and Psychophysics, 60, 1384-1405.

Galambos, J. (1978). The Asymptotic Theory of Extreme Order Statistics. New York: John Wiley and Sons.

Gnedenko, B. (1943). Sur la distribution limite du terme maximum d'une série aléatoire. Annals of Mathematics, 44, 423-453.

Goldstone, R. L. (1998). Perceptual Learning. Annual Review of Psychology, 49, 585-612.

Green, D. M., Swets, J. A. (1966). Signal Detection Theory and Psychophysics. New York: John Wiley and Sons.

Grünwald, P. (2000). Model Selection based on Minimum Description Lengt. Journal of Mathematical Psychology, 44, 133-152.

Gumbel, E. J. (1958). The Statistics of Extremes. New York: Columbia University Press.

Haider, H., Frensch, P. A. (1996). The role of information reduction in skill acquisition. Cognitive Psychology, 30, 304-337.

Haider, H., 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.

Heathcote, A., Brown, S., Mewhort, D. J. K. (2000). The power law repealled: The case for an exponential law of practice. Psychonomic Bulletin & Review, 7, 185-207.

Heathcote, A., Mewhort, D. J. K. (1995, November). The law of practice. 36th annual meeting of the psychonomics society, Pittsburgh.

Hinton, G. E.; Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. Psychological Review, 98, 74-95.

Hopkins, G. W., Kristofferson, A. B. (1980). Ultrastable stimulus-reponse latencies: Acquisition and stimulus control. Perception and Psychophysics, 27, 241-250.

Indow, T. (1993). Analyses of events counted on time-dimension: A soft model based on extreme statistics. Behaviormetrika, 20, 109-124.

Kohonen, T. (1984). Self-organization and associative memory. Berlin: Springer-Verlag.

LaBerge, D. A. (1962). A recruitment theory of simple behavior. Psychometrika, 27, 375-396.

Lacouture, Y. (1989). From mean square error to reaction time: A connectionist model of word recognition, in Touretzky, David (Ed) (eds.). Proceedings of the 1988 Connectionist Models Summer Schoo (pp. 371-378). San Mateo, CA: Morgan Kaufmann, Inc..

Leadbetter, M. R., Lindgren, G., Rootzén, H. (1983). Extremes and related properties of random sequences and processes. New York: springer-Verlag.

Levitan, I. B., Hamar, A. J., Adams, W. B. (1979). synaptic and hormonal modulation of a neuronal oscillator: A search for molecular mechanisms. Journal of Experimental Biology, 81, 131-151.

Link, S. W. (1992). Imitatio Estes: Stimulus sampling origin of Weber's law, in Healy, A., L., Kosslyn, S., M., Shiffrin, R., M. (eds.). From learning theory to connectionist theory: Essays in honor of William K. Estes (pp. 97-113). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Logan, G. D. (1988). Toward an instance theory of automatization. Psychological Review, 95, 492-527.

Logan, G. D. (1992). Shapes of reaction-time distributions and shapes of learning curves: a test of the instance theory of automaticity. Journal of Experimental Psychology: Learning, Memory and Cognition, 18, 883-914.

Lu, Z.-L., Dosher, B. A. (1998). External noise distinguises attention mechanisms. Vision Research, 38, 1183-1198.

Luce, R. D. (1986). Response times, their role in inferring elementary mental organization. New York: Oxford University Press.

McClelland, J. L. (1979). On the time relations of mental processes: A framework for analyzing processes in cascade. Psychological Review, 86, 287-330.

McClelland, J. L., Rumelhart, D. E. (1988). Explorations in parallel distributed processing. Cambridge (Mass): Bradford Book.

Medin, D. L., Wattenmaker, W. D., Michalski, R. S. (1987). Constraints and preferences in inductive learning: an experimental study of human and machine performance. Cognitive Science, 11, 299-339.

Meijers, L.M.M., Eijkman, E.G.J. (1977). Distributions of simple RT with single and double stimuli. Perception and Psychophysics, 22, 41-48.

Meyer, D. E., Irwin, D. E., Osman, A. M., Kounios, J. (1988). the dynamics of cognition and action: mental processes inferrde from speed-accuracy decomposition. Psychological Review, 95, 183-237.

Miller, J. (1982). Divided attention: Evidence for coactivation with redundant signals. Cognitive Psychology, 14, 247-279.

Minsky, R., Papert, S. (1969). Perceptrons: an introduction to computational geometry. Cambridge, Mass: MIT Press.

Myung, I. J. (2000). The importance of complexity in model selection. Journal of Mathematical Psychology, 44, 190-204.

Newell, A. Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice, in Anderson, J. R. (eds.). Cognitive skills and their acquisition (pp. 1-55). Hillsdale, NJ: Laurence Erlbaum Associates.

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. (1994). Set-size effects in visual search: The effect of attention is independent of the stimulus for simple tasks. Vision Research, 34, 1703-1721.

Palmer, J. (1998). Attentional effects in visual search: relating search accuracy and search time, in Richard D. Wright (eds.). Visual attention (pp. 348-388). New York: Oxford University Press.

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., Murdock, B. B. (1976). Retrieval processes in recognition memory. Psychological Review, 86, 190-214.

Ratcliff, R., Van Zandt, T., McKoon, G. (1999). Connectionist and diffusion models of reaction time. Psychological Review, 106, 261-300.

Reed, A. V. (1976). List length and the time course of recognition in immediate memory. Memory & Cognition, 4, 16-30.

Reicher, G. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. Journal of Experimental Psychology, 81, 275-280.

Rickard, T. C. (1997). Bending the power law: a CMPL theory of strategy shitfs and the automatization of cognitive skills. Journal of Experimental Psychology: General, 126, 288-311.

Roberston, L. C. (2000, November). Brain, space, and feature binding. 41st Annual Meeting of the Psychonomic Society, New Orleans.

Rosenblatt, F. (1961). Principles of neurodynamics: Perceptrons and the theory of the brain mechanisms. Washington, DC: Spartan.

Rousseau, L., Rousseau, R. (1996). Stop-reaction time and the internal clock. Perception and Psychophysics, 58, 434-448.

Rumelhard, D. E. Siple, P. (1974). Process of recognizing tachitoscopically presented words. Psychological Review, 81, 99-118.

Schyns, P. G., Rodet, L. (1995). Concept learning in connectionnist networks, in Arbib, M. (eds.). The handbook of brain theory and neural networks (pp. 1-21). Cambridge, Mass.: MIT Press.

Seidenberg, M. S.; McClelland, J. L. (1990). More words but still no lexicon: Reply to Besner et al. (1990). Psychological Review, 97, 447-452.

Smith, E. E., Haviland, S. E. (1972). Why words are perceived more accurately than nonwords: inference versus unitization. Journal of Experimental Psychology, 92, 59-64.

Smith, P. L., Vickers, D. (1988). The accumulator model of two-choice discrimination. Journal of Mathematical Psychology, 32, 135-168.

Sternberg, S. (1969). The discovery of processing stages: Extensions of Donders' method. Acta Psychologica, 30, 276-315.

Takane, Y., Oshima-Takane, Y., Shultz, T. R. (1999). Analysis of knowledge representations in cascade correlation networks. Behaviormetrika, 26, 5-28.

Thomas, R. D. (2000, August). Analysis of factorial response time patterns predicted by current models of perception. 33rd annual meeting of the Society for Mathematical Psychology, Kingston.

Thorpe, S. J., Gautrais, J. (1999). Rapid visual processing using spike asynchrony. Toulouse: Centre de recherche Cerveau & Cognition.

Townsend, J. T., Ashby, F. G. (1983). Stochastic modeling of elementary psychological processes. Cambridge, England: Cambridge University Press.

Townsend, J. T., Nozawa, G. (1995). Spatio-temporal properties of elementary perception: An investigation of parallel, serial, and coactive theories. Journal of Mathematical Psychology, 39, 321-359.

Trabasso, T., Rollins, H., Shaughnessy, E. (1971). Storage and verification stages in processing concepts. Cognitive Psychology, 2, 239-289.

Travers, J. R. (1973). The effects of forced serial processing on identification of words and random letter strings. Cognitive Psychology, 5, 109-137.

Ulrich, R., Giray, M. (1986). Separate-activation models with variable base times: Testability and checking of cross-channel dependency. Perception and Psychophysics, 39, 248-254.

Ulrich, R., Miller, J. (1993). Information processing models generating lognormally distributed reaction times. Journal of Mathematical Psychology, 37, 513-525.

Van Gelder, T. (1995). What might cognition be, if not computation?. Journal of Philosophy, XCII, 345-381.

Van Zandt, T., Colonius, H., Proctor, R. W. (in press). A comparison of two response-time models applied to perceptual matching. Psychonomic Bulletin & Review.

Wald, A. (1947). Sequential analysis. New York: John Wiley and sons.

Ward, R., McClelland, J. L. (1989). Conjunctive search for one and two identical targets. Journal of Experimental Psychology: Human Perception and Performance, 15, 664-672.

Weibull, W. (1951). A statistical distribution function of wide applicability. Journal of Applied Mechanic, 18, 292-297.

West, J., Shlesinger, M. (1990). The noise in natural phenomena. American Scientist, 78, 40-45.

Widrow, B., Hoff, M. E. (1960,). Adaptive switching circuits. Institute of radio engineer, western electronic show and convention, Convention record.

Yonelinas, A. P., Kroll, N. E. A., Dobbins, I. G., Soltani, M. (1999). Recognition memory for faces: Can familiarity support associative recognition judgments?. Psychonomic Bulletin & Review, 6, 654-661.

Zenger, B., 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.

FIGURE CAPTION

Figure 1. a) The representation of a serial Poisson race model. Signals are received as spikes of activation through a unique channel in a serial mode. Information is accumulated until K_A of them fill the accumulator. b) The representation of a Perceptron. Signals are received as strength of activation through multiple channels in parallel.

Figure 2. A parallel race model with two accumulators and one clock unit. a) Signals are received at various moments $\boldsymbol{I}_i$ through independent channels and stored in the accumulator after going through the connections $\mathbf{D}_{ij}$. The symbol 🕐 denotes the clock (time-out) unit. b) The same model seen as a matrix of delays and vectors for inputs, thresholds and outputs. The dash line indicates the delimitation between inputs from the stimulus and inputs from the time-out units.

Figure 3. Alternative representation of the PRN. a) PRN illustrated using a decision tree that minimizes the information processed. b) Random walk representation where the drift rates δ_i are the reciprocal of the average delays $\boldsymbol{D}_{ij}$ and the variability η_i depends on the noise applied to $\mathcal{I} * \mathcal{D}$.

Figure 4. Building a ROC curve from a Parallel race model. a) Signal detection theory assumes that signal **S** and noise **N** are random activation and that activation higher than a constant *c* results in a signal present response. b) Parallel race model assumes that signal (**S**) is characterized by the moment when the input becomes active. Because of variability, this moment is a random variable. If no signal is present, spurious activation (***Sp***) will occur, but it is likely to occur later (thus, the scale of ***Sp*** is much larger than the scale of **S**). If the signal arrives too late, the clock (**C**) will emit a default response. However, the clock is also a random variable with

position ***t***. c) Some ROC curves generated from the Parallel race model assuming that variability in **S** and ***Sp*** are Weibull. Scale of ***Sp*** is 5 times larger than scale of **S**.

Figure 5. The XOR problem. a) The optimal solution when redundancy is present. b) Illustration of the amount of noise added to the input. Two levels are shown, $\eta = 0.5$ and $\eta = 2.0$. c) Illustration of the amount of noise added to the delays.

Figure 6. Learning curves of the parallel race network presented with the XOR problem. Left column shows the percent of errors (P(e)) as a function of epoch (1 epoch represents 10 trials) when the amount of noise η varies from 0.0 to 2.0 (ρ was constant at 1). As seen, learning is fast, but in presence of noise, the system still do scattered errors. Right column shows the solution achieved by the network. As seen, the drift rates become increasingly shallow when noise increases. Thresholds are unaffected by noise.

Figure 7. Illustration of learning curves for the parallel race network when both redundancy ρ and noise η are manipulated. Columns shows two level of noise ($\eta = 0.5$ and 2.0), rows are for five level of redundancy ($\rho = 1$, 2, 4, 8 and 16). Even with very large amount of redundancy and noise, learning occurs rapidly.

Figure 8. a) Appropriate values for t_1 when noise η (0.5 or 2.0) and redundancy ρ (1 or 4) are varied. b) Estimated surface for t_1 values when noise η (0.5 or 2) and redundancy ρ (1 or 4) are introduced. The reciprocal of these values gives an estimator of the drift rates in a random walk representation.

Figure 9. Recovered parameters from the best-fitting Weibull to the simulated response times to response A (both input present) in a XOR task, as a function of redundancy ρ and noise η in arbitrary units of time. The parameters are the position $\hat{\alpha}$, the scale $\hat{\beta}$, and the shape $\hat{\gamma}$.

Figure 10. a) Mean correct RT and b) Percent correct in the XOR experiment.

Figure 11. Correct RT distributions for the 2 subjects in the XOR experiment as a function of the number of dots presented.

Figure A.1. The problems used in the text. The left column shows the problem space with the associated response. Colors indicate the correct solution to the stimuli used in these problems. Note that a value of zero on any dimension indicates that the corresponding feature was absent. The right column shows the matrix solution for a parallel race model. a) The Detect problem. The system has to say A if something happens. The general solution to this problem is to say "A" as soon as dimension 1 is activated ($\boldsymbol{D}_{1A} = 0$) or else say "B" after a while ($t = \boldsymbol{D}_{⏲B} > 0$). b) The 1D problem and its solution. The correct response is A if and only if dimension 2 is presented. Input for dimension 2 has the highest priority for response A ($\boldsymbol{D}_{2A} = 0$) followed by input for dimension 1 with respect to response B ($t = \boldsymbol{D}_{1B} > 0$). Thresholds are 1. c) The Identification problem. Correct responses are A or B depending on which dimension is on. There is no case where both dimensions are on at the same time. The clock is used to put time pressure on the system (at time $t = \boldsymbol{D}_{⏲C}$) and the corresponding response C is a sophisticated guess. d) The AND problem and its solution. The correct response is A if and only if the two dimensions are presented. Input from both dimensions have the highest priority ($\boldsymbol{D}_{iA} = 0$) but both are needed ($K_A = 2$). Any input or clock can later provoke a response B ($t = \boldsymbol{D}_{iB} > 0$). The clock is essential in case no input is presented. e) The XOR problem. Response A is correct if both or none of the

inputs is on. Response B is made in case where only one of the two inputs is on ($0 < \boldsymbol{t_1} = \boldsymbol{D}_{iB} < \boldsymbol{t_2} = \boldsymbol{D}_{\textcircled{?}C}$). This is a non-linearly separable problem. The solution requires two clocks when no input is present because the threshold K_A must be two for the case where both inputs are on.

Figure D.1. Speed-accuracy decomposition rational. a) The observed guessing times $\boldsymbol{t}_s$ distribution is a compound of a normal processing time ($\boldsymbol{t}_n$ observable when there is no time constraint) and a guessing time ($\boldsymbol{t}_{gs}$ not observable). However, $\boldsymbol{t}_{gs}$ can be deduced from the data. b) Deduced guessing time distributions $\boldsymbol{t}_{gs}$ when three time constraints are used to set $\boldsymbol{D}_{\textcircled{?}C}$. c) Speed-accuracy trade-off plot of Percent correct as a function of time (in arbitrary units).

---

[22] A general power curve function is given by a function $f(x) = b(x - \alpha(f))^{\gamma}$. The purpose of the Gnedenko formulation is to get rib of both the subtractive parameter (through addition) and the multiplicative parameter (through the use of a ratio) so that we retain only the core of a power function: its exponential parameter γ.

Table 1.
Analyses of Variance performed on the summary values ΔK and ΔD obtained after training on a XOR problem

	ANOVA			
Factor	SS	dl	MS	F
log D ***D***				
anova				
log Redundancy (log r)	961.3	3	320.4	73.8 **
log Noise (log h)	1553.4	4	388.3	86.4 **
log r X log h	519.0	12	43.2	9.96 **
error term	260.5	60	4.3	
Best fitting equation				
log ΔD	= 1.1 log h - 0.6 log r + 5.2			R^2 = .75
D ***K***				
anova				
Redundancy (r)	1192.0	3	397.3	142.8 **
Noise (h)	3.6	4	0.9	<1
r X h	19.6	12	1.6	<1
error term	166.9	60	2.8	
Best fitting equation				
ΔK	= 0.41 r - 0.6			R^2 = .77

** : $p < .01$

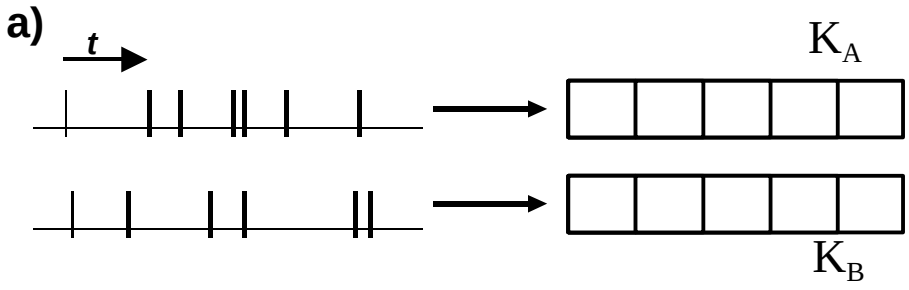
Table 2.

Proportion best fitted for three distribution functions to the response times to patterns A, when noise η and redundancy ρ are varied independantly.

	Distribution		
redundancy (η = 0.5)	Weibull	LogNormal	Ex-Gaussian
1	88	8	4
2	78	15	7
4	48	31	21
8	41	39	20
16	56	23	21
32	67	5	28
noise (ρ = 2)			
0.5	62	23	15
1.0	90	2	8
1.5	84	2	14
2.0	80	3	17

Table 3.
Moments and estimated parameters from the XOR experiment

Number of	Moments		Distribution		
dots	$\overline{RT}$	$\overleftrightarrow{RT}$	$\hat{a}$	$\hat{\hat{a}}$	$\tilde{\hat{a}}$
0	464	158	258	206	1.9
1	454	105	278	188	1.91
2	432	85	277	155	1.88
differences					
0-1	10	53	-20	18	-0.01
1-2	22	20	1	33	0.03



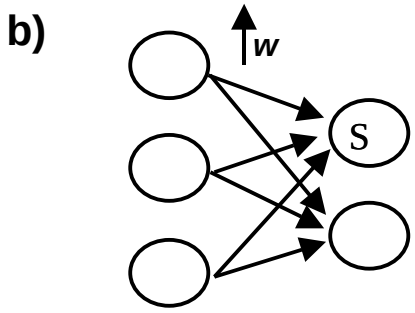


Figure 1

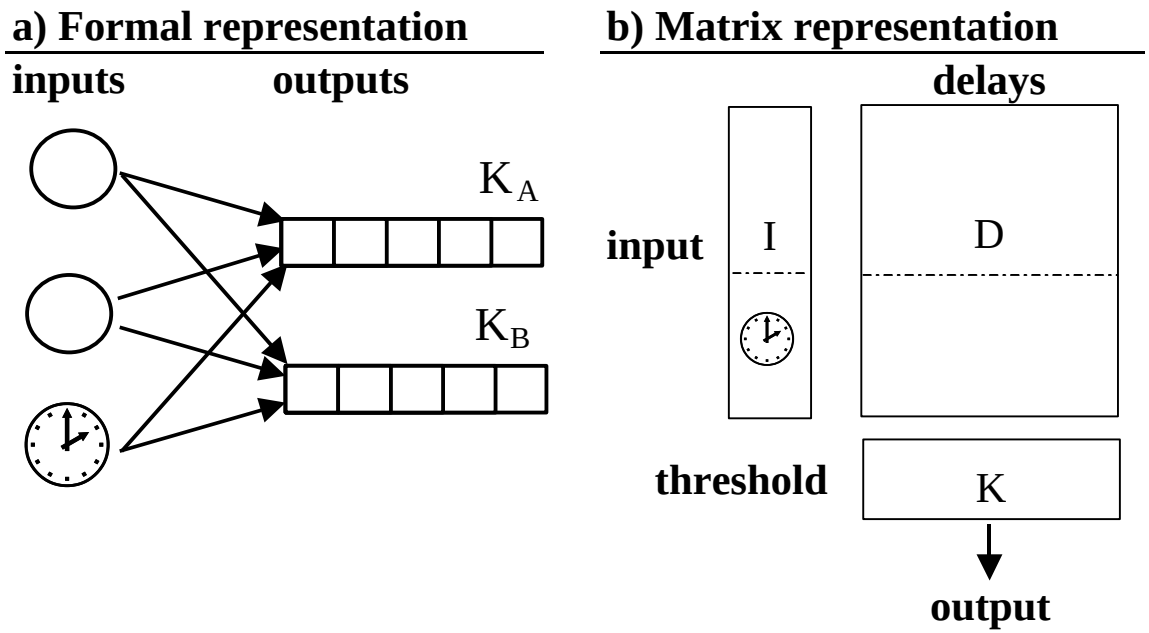
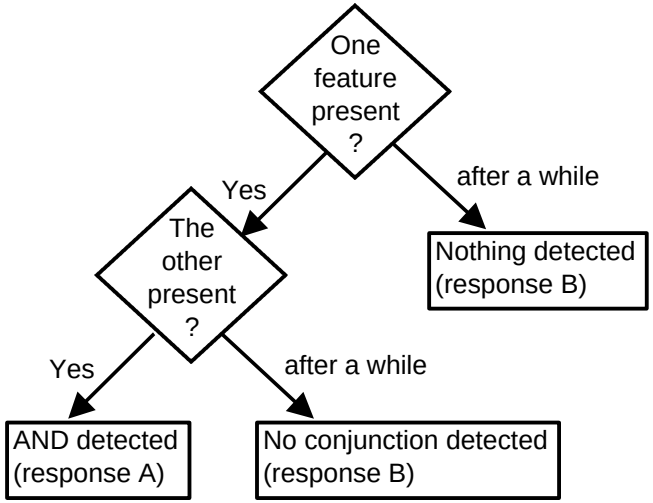
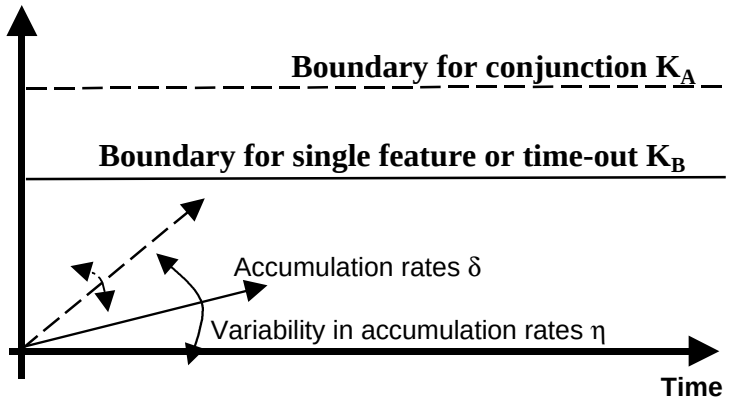

a) Formal representation
inputs
outputs
K_A
K_B
b) Matrix representation
delays
input
I
D
threshold
K
output

Figure 2

a) Priority learner representation **b) Random walk representation**



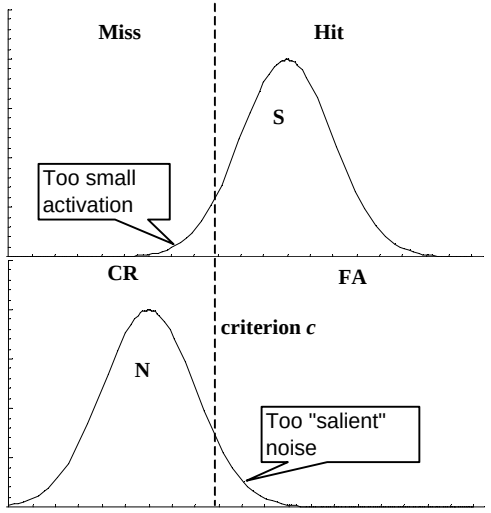


Accumulation rate for the conjunction

Accumulation rate for any single feature and the time-out unit

Figure 3

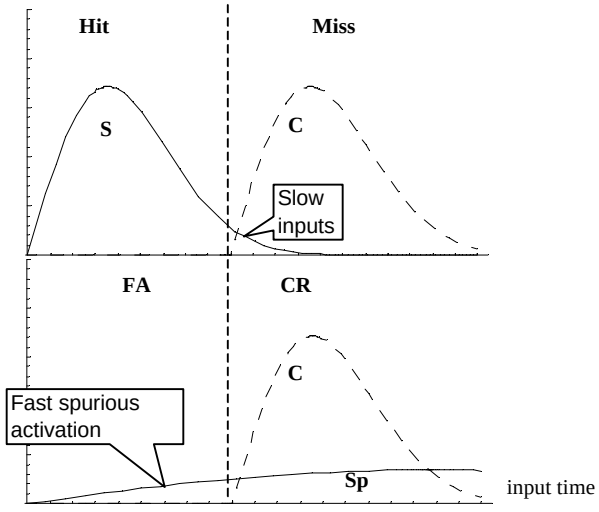
a) Signal detection theory



$$Pr(HIT) = Pr(\mathbf{S} > c)$$

$$Pr(FA) = Pr(\mathbf{N} > c)$$

b) Parallel race model



$$Pr(HIT) = Pr(\mathbf{S} < \mathbf{C})$$

$$Pr(FA) = Pr(\mathbf{Sp} < \mathbf{C})$$

c) ROC curves from b)

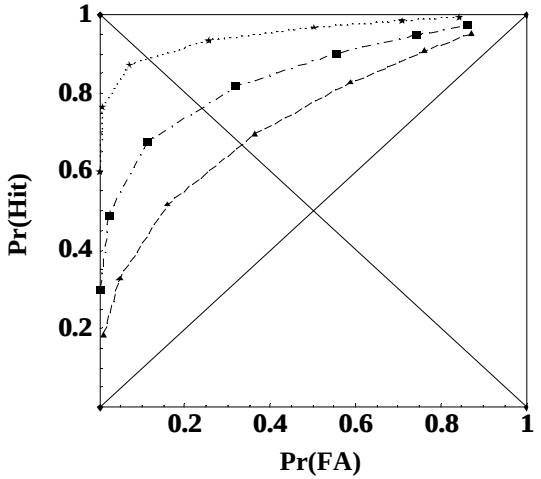


Figure 4

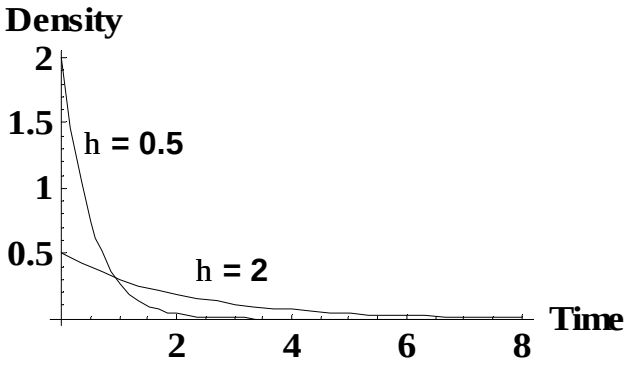
a) Optimal solution

$$D = \begin{array}{|cc|}\hline t_0 & t_1 \\ t_0 & t_1 \\ \hdashline t_2 & t_3 \\ t_2 & t_3 \\ \hline \end{array}$$

$$K = \boxed{r+1 \quad 1}$$

$0 \leq t_0 \ll t_1 \ll t_2 \ll t_3 \leq \infty$

b) Noise functions $\mathcal{I}$



c) Noise function $\mathcal{D}$

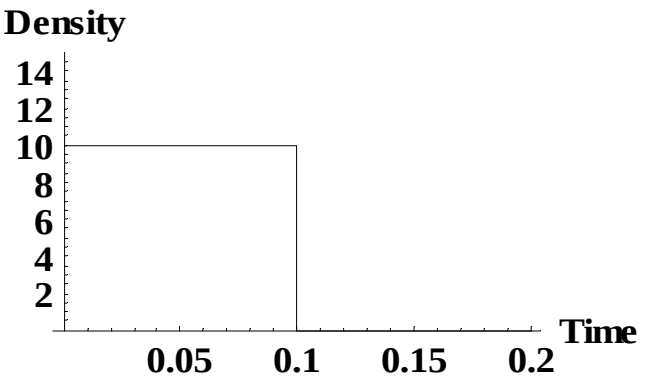
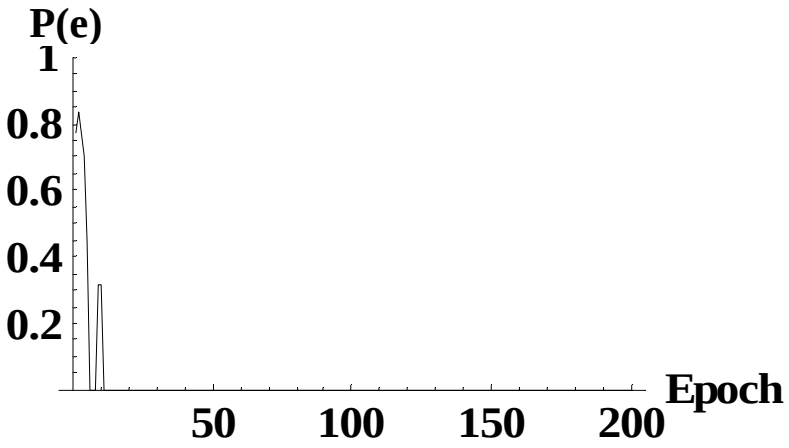


Figure5

h = **0.0**



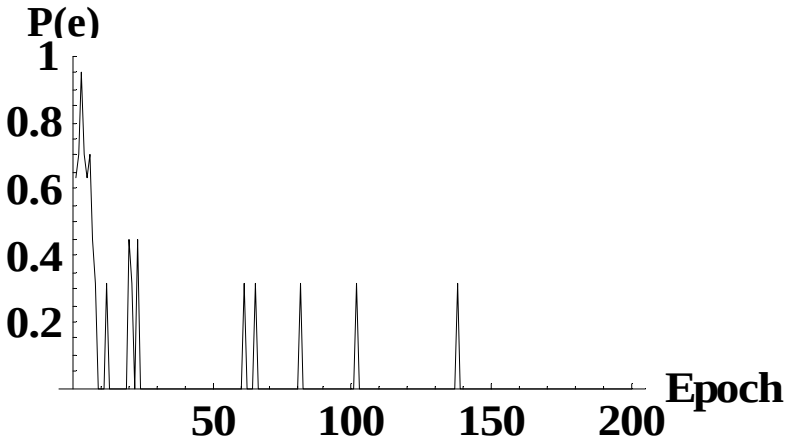
D_{ij} =

4.04	6.01
4.06	6.02
8.08	10.02
8.00	10.07

K =

2.00	1.00

h = **0.5**



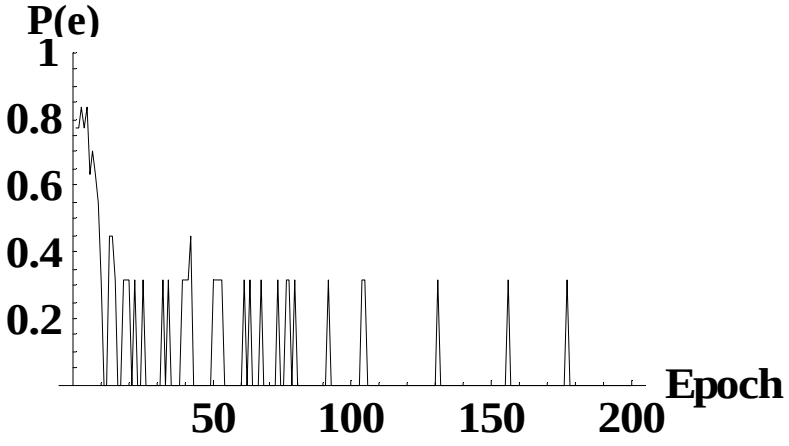
D_{ij} =

4.09	10.09
6.05	8.07
14.00	18.10
14.05	18.06

K =

2.00	1.00

h = **2.0**



D_{ij} =

4.05	16.08
6.06	16.10
28.07	34.07
26.04	34.09

K =

2.00	1.00

Figure 6

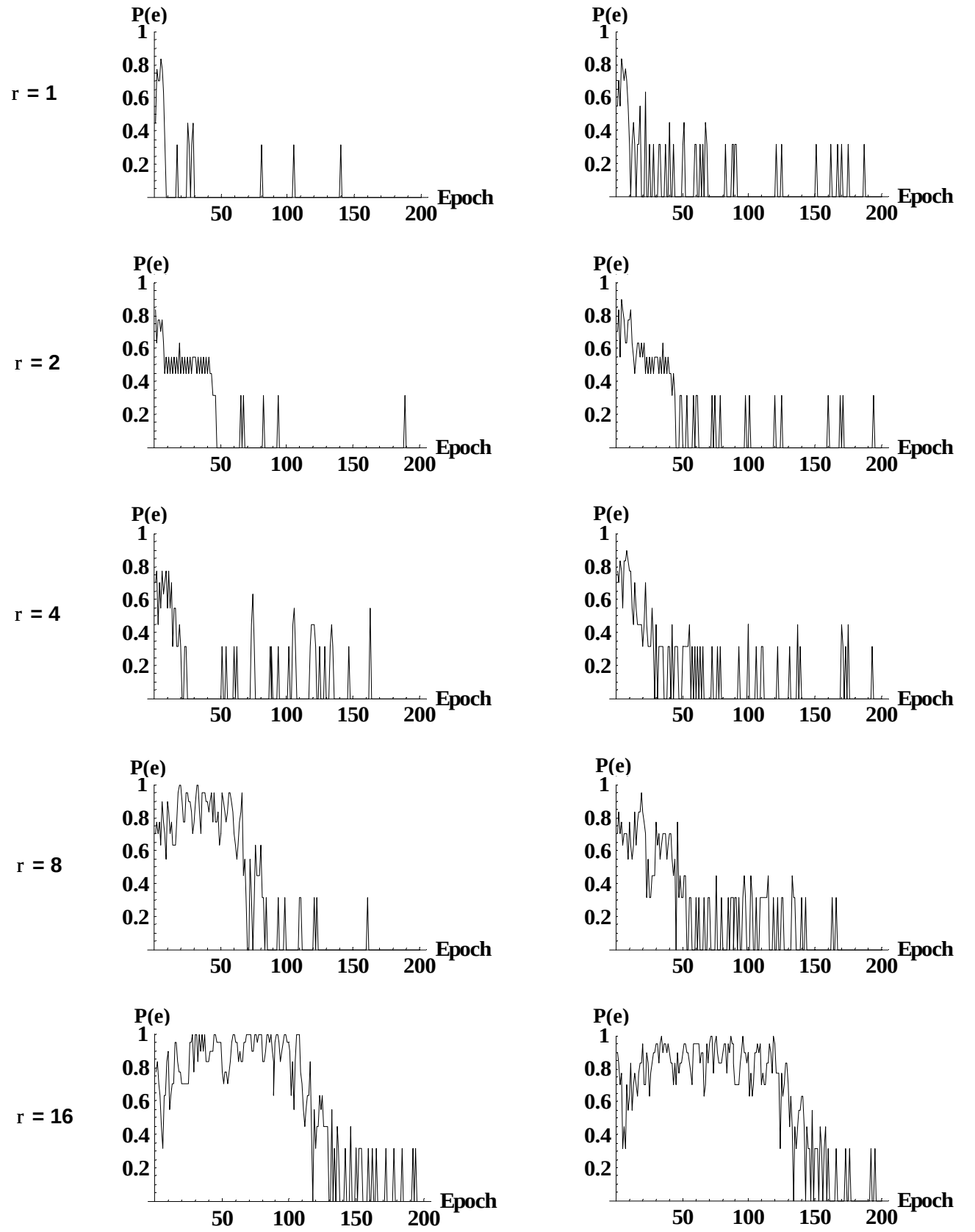

Redundancy
Noise h = 0.5
Noise h = 2.0
r = 1
r = 2
r = 4
r = 8
r = 16
P(e)
1
0.8
0.6
0.4
0.2
50
100
150
200
Epoch

Figure 7

a) Delay D_{1B} as a function of exponential noise *h* and redundancy *r*

	η	
ρ	0.5	2.0
1	1.0	4.0
4	0.2	1.0

b)

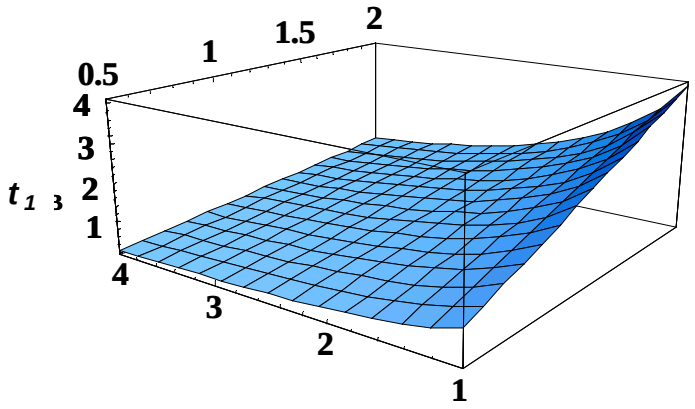
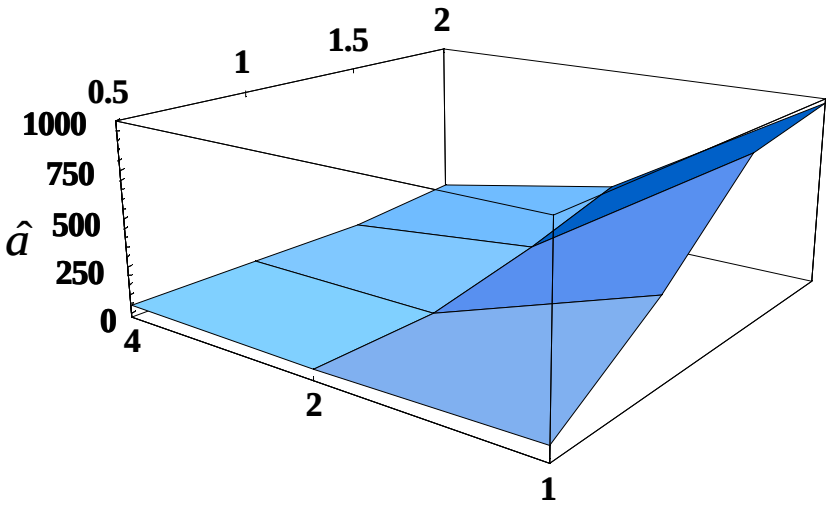
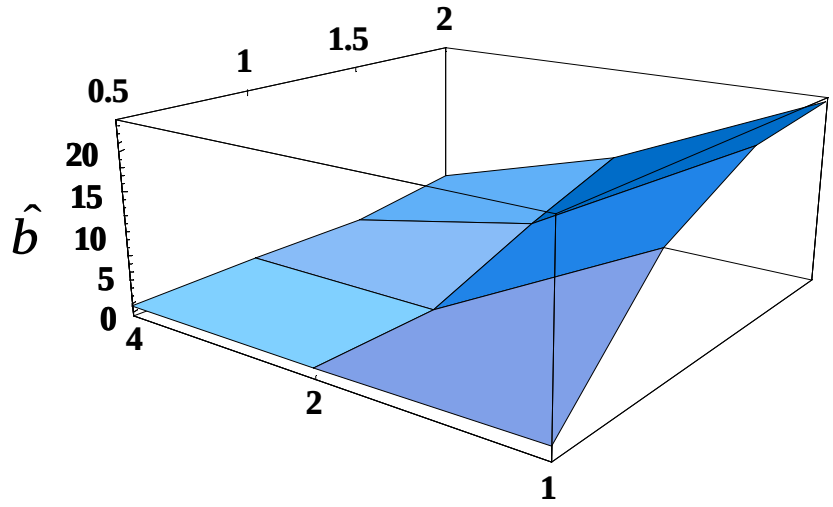


Figure 8


0.5
1
1.5
2
1000
750
500
250
0
$\hat{a}$
4
2
1


0.5
1
1.5
2
20
15
10
5
0
$\hat{b}$
4
2
1

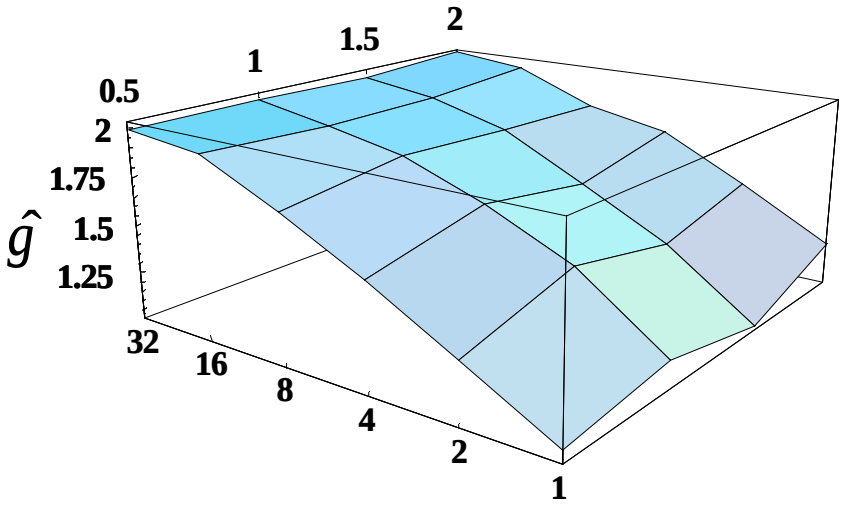

0.5
1
1.5
2
2
1.75
1.5
1.25
$\hat{g}$
32
16
8
4
2
1

Figure 9

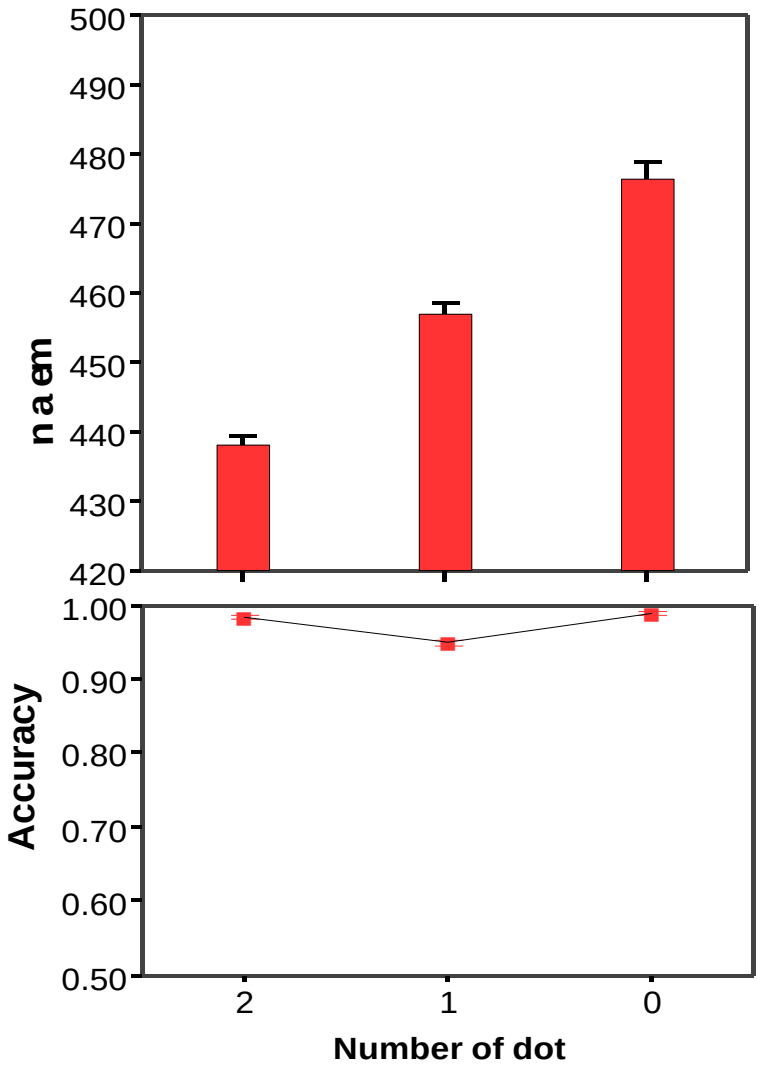

500
490
480
470
460
450
440
430
420
n a em
1.00
0.90
0.80
0.70
0.60
0.50
Accuracy
2
1
0
Number of dot

Figure 10

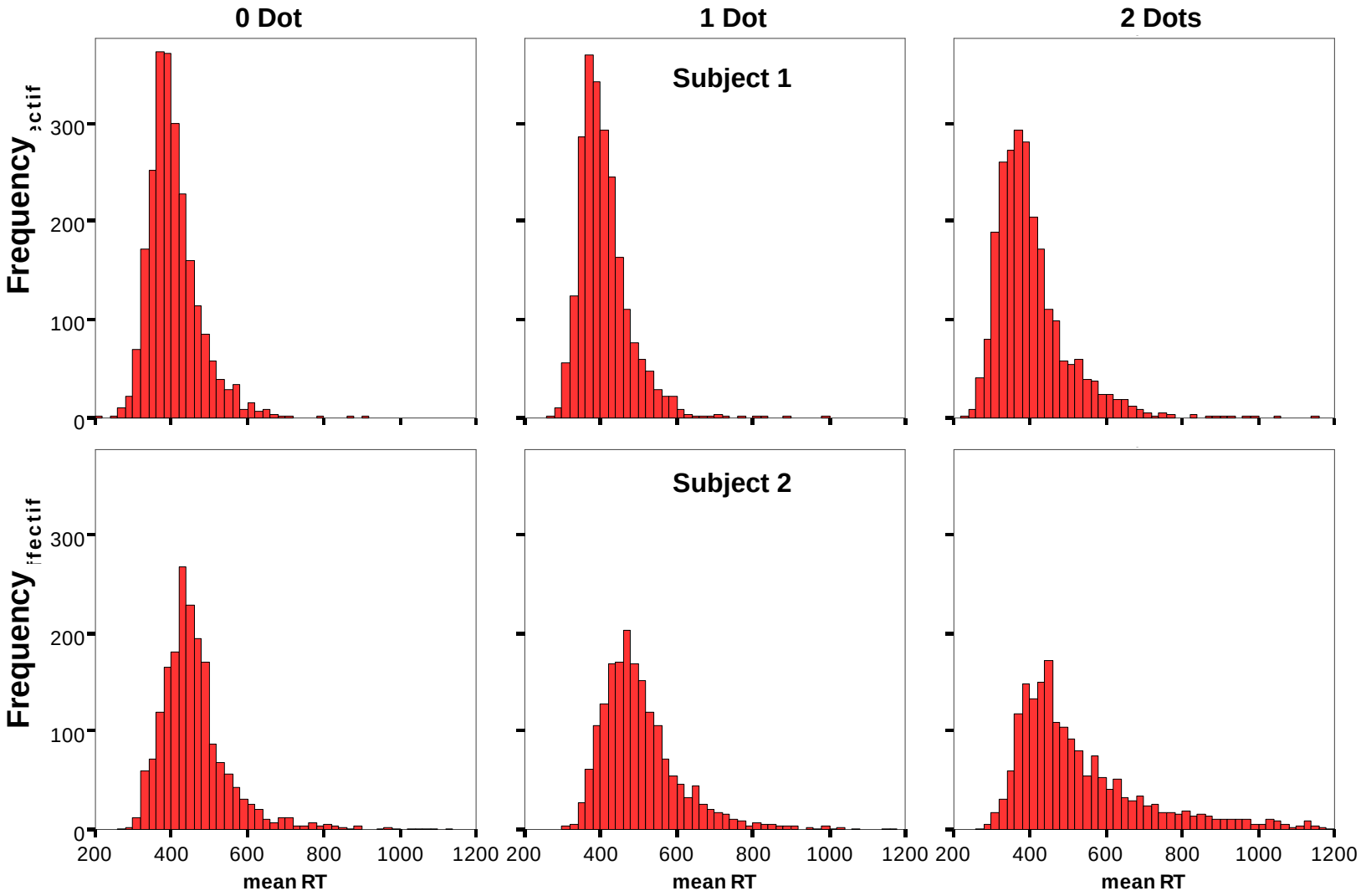
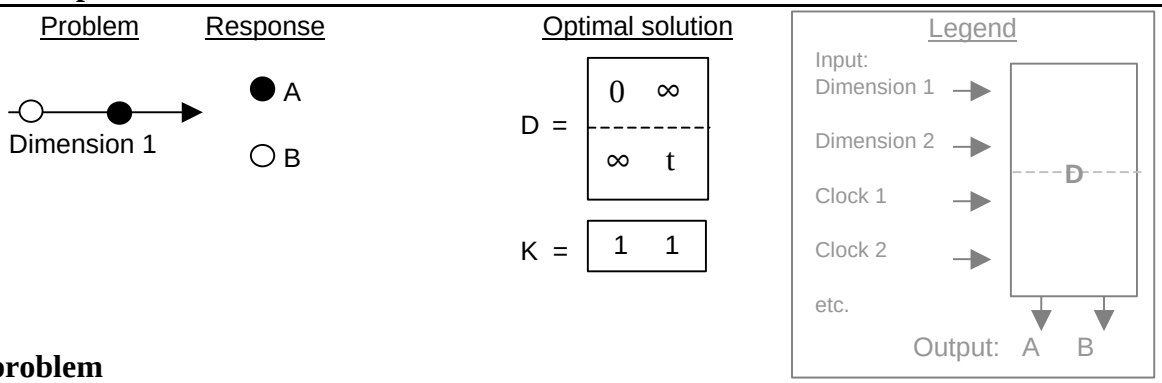
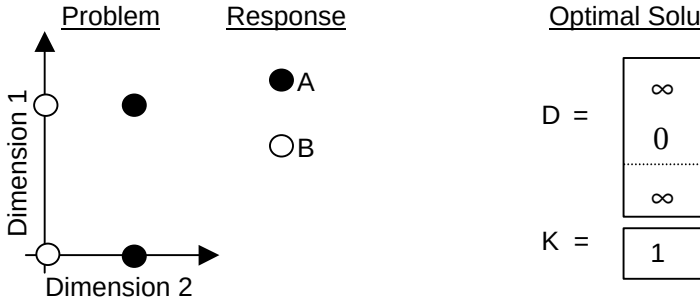

0 Dot
1 Dot
2 Dots
Subject 1
Subject 2
Frequency
0
100
200
300
200
400
600
800
1000
1200
mean RT

Figure 11

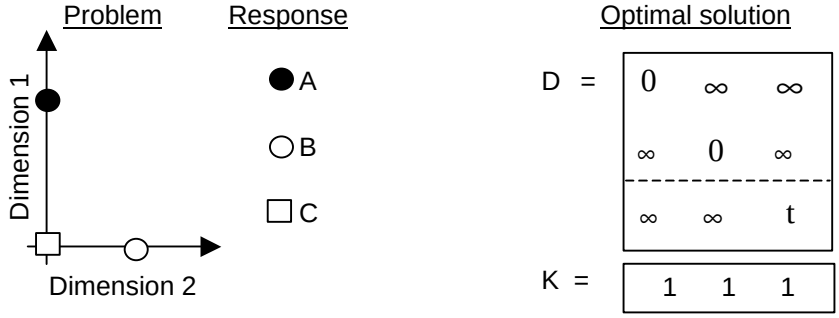
a) Detection problem



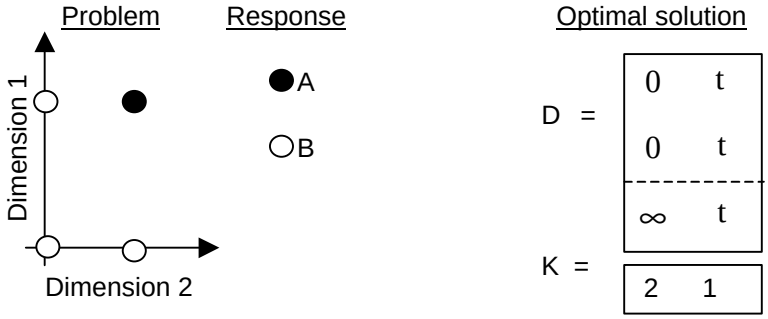
b) 1D problem



c) Identification problem



d) AND problem



e) XOR problem

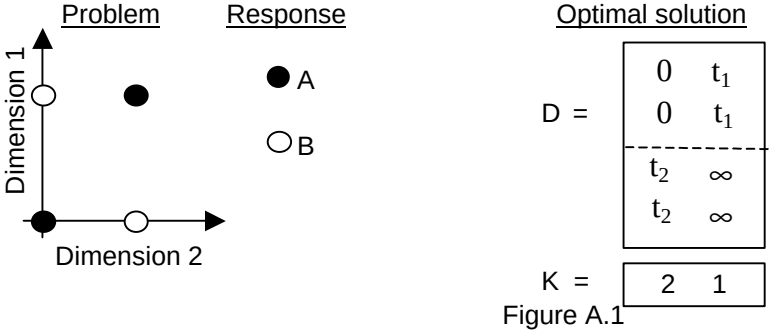
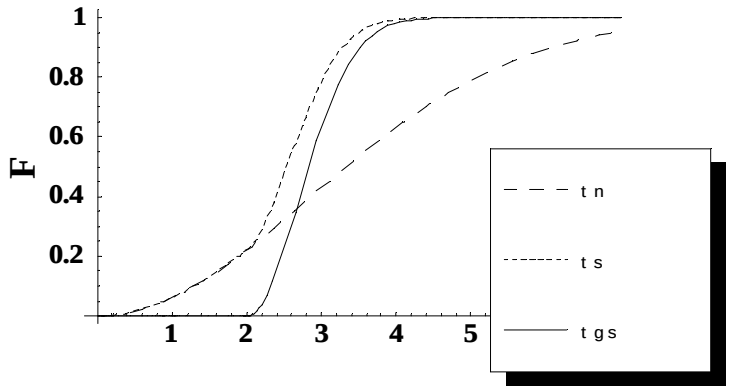
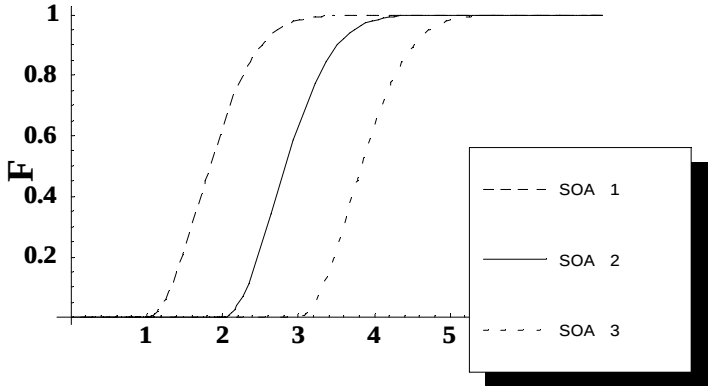


Figure A.1

a) Plot of normal (t_n) and StoR (t_s) distributions along with the guessing time (t_{gs}) distribution



b) Distributions for three different guess SOA



c) Speed-accuracy trade-off plot

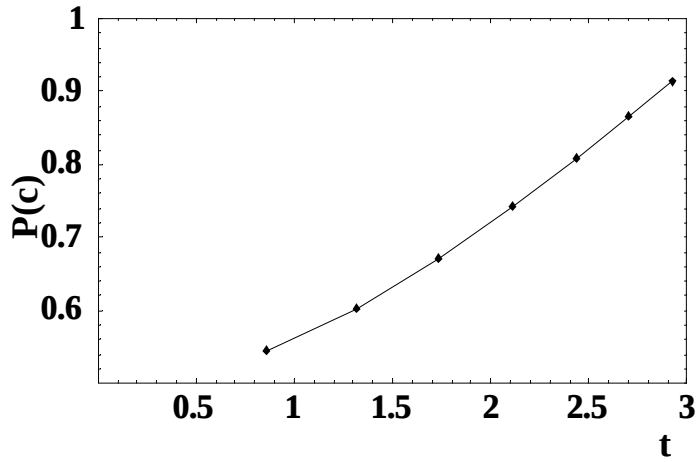


Figure D.1