A Delta-Rule Model of Numerical and Non-Numerical Order Processing
Tom Verguts and Filip Van Opstal
Online First Publication, January 6, 2014. doi: 10.1037/a0035114

CITATION
A Delta-Rule Model of Numerical and Non-Numerical Order Processing

Tom Verguts and Filip Van Opstal
Ghent University

Numerical and non-numerical order processing share empirical characteristics (distance effect and semantic congruity), but there are also important differences (in size effect and end effect). At the same time, models and theories of numerical and non-numerical order processing developed largely separately. Currently, we combine insights from 2 earlier models to integrate them in a common framework. We argue that the same learning principle underlies numerical and non-numerical orders, but that environmental features determine the empirical differences. Implications for current theories on order processing are pointed out.

Keywords: numerical cognition, non-numerical orders, computational modeling, learning

Supplemental materials: http://dx.doi.org/10.1037/a0035114.supp

In terms of military rank, a soldier is below a lieutenant, who is below a general; a general in turn receives commands from the president herself. In the army, as well as in daily life more broadly, order processing is an important cognitive ability. Indeed, one particular order, namely the number sequence (1, 2, 3, . . . ), is key for our understanding of the physical world (Wigner, 1960). Moreover, not only primates (including humans) but also some nonprimates, such as rodents, are capable of order processing (e.g., Dusek & Eichenbaum, 1997; Van der Jeugd et al., 2007), suggesting a rich phylogenetic history.

In experimental psychology, order processing is often studied by presenting a pair of stimuli on every trial, requiring participants to identify the larger/smaller/most rightward/most leftward of the two stimuli. For example, in numerical orders, stimuli could come from the range 1 to 9, and eligible stimulus pairs would be 1 2, 5 9, 9 3, and so on. Also, an arbitrary order of elements may be constructed, in which case participants have to extract the relevant order by trial and error and/or prior instruction (e.g., A < B < C < D < E).

However, the cognitive operations underlying order processing remain poorly understood. This is not because of lack of data: Indeed, a number of well-replicated phenomena characterize order processing across species. Different theories and models have been proposed, but consensus has not been reached. In addition, theories on numerical and non-numerical orders have largely developed in separate streams. In the current article, we follow up on an earlier approach in modeling order processing as emergent from simple network properties (Leth-Steenstrup & Marley, 2000; Verguts, Fias, & Stevens, 2005). In Verguts et al. (2005), we attempted to understand various and seemingly contradictory findings in numerical cognition as emergent from the interaction between environmental properties and decision making in neural networks. As noted by Botvinick and Cohen (in press), such interaction is a fundamental assumption of the neural network approach. In our view, this has lead to considerable success in understanding old and predicting novel data in a variety of domains (e.g., Botvinick, Braver, Barch, Carter, & Cohen, 2001; Frank, Seeberger, & O’Reilly, 2004). One prominent feature of many neural networks in general, and the Verguts et al. model in particular, is the use of a delta rule for training the network. “Delta” refers to the difference between desired activation (teacher signal) and predicted activation at response level (δ = desired – predicted). Cognitive (neuro)science is rapidly (re)discovering the importance of prediction and prediction error (e.g., Alexander & Brown, 2011), but it has long been an inherent part of neural networks (Rescorla & Wagner, 1972). Although models trained by a delta rule can vary across many architectural properties, we will use the term delta-rule model in the following because it allows for convenient labeling, and because it is an important feature. Here, we extend the earlier delta-rule model of numerical cognition to obtain a general framework on numerical and non-numerical order processing. We train a one-layer neural network using the delta rule (Rescorla & Wagner, 1972; Widrow & Hoff, 1960) on an order. When we sample elements from a frequency distribution typical for non-numerical orders (i.e., uniform distribution), we obtain an end effect. On the other hand, if we sample elements from a frequency distribution that mimics training with numerical orders (strongly biased toward smaller numbers; Dehaene & Mehler, 1992), we obtain a size effect (as in Verguts et al., 2005). This suggests that the frequency distribution at least partially determines the observed differences between numerical and non-numerical order processing characteristics.

The remainder of this article is structured as follows. First, we list the relevant empirical phenomena, followed by models of order
processing. Then, we describe our new model, followed by two simulation sections. We close with a General Discussion.

Order Processing: Data

Consider a six-element order $A < B < C < D < E < F$. These elements may refer to numbers ($1 < 2 < \ldots < 6$) or to some other order (e.g., soldier ranks below sergeant ranks below $\ldots$ ranks below president). When a novel order is used, the experiment typically contains a training stage. In some experiments, all pairs are presented during training. Often, however, only the adjacent pairs are presented (e.g., Tzelgov, Yehene, Kotler, & Alon, 2000; von Fersen, Wynn, Delius, & Staddon, 1991). In both cases, accuracy during the test phase is high (at least above the chance level, but typically much higher) not only for trained pairs, but also for the untrained pairs (in the latter case). This demonstrates that participants are able to generalize (e.g., Van Elzakker, O’Reilly, & Rudy, 2003). We will summarize high accuracy on trained and untrained pairs by the term \textit{high accuracy}. Although this property seems elementary, some theories can be ruled out by this requirement (see below).

Second, a distance effect is obtained, meaning that pairs with small distances (e.g., pair AB) are more difficult to solve (e.g., slower response times [RT]s than large distances, such as pair AD). For example, a classic finding in numerical cognition is that identifying the larger of 1 and 2 is more difficult (longer response time [RT]) than identifying the larger of 1 and 6 (Moyer & Landauer, 1967). This holds even when small-distance pairs are more frequent (e.g., when only adjacent pairs are trained). The distance effect has been reported in both humans (e.g., Leth-Steensen & Marley, 2000; Moyer & Landauer, 1967) and animals (e.g., Rilling & McDiarmid, 1965; for review, see Ansari, 2008; Dehaene, Dehaene-Lambertz, & Cohen, 1998). It is robustly obtained in both numerical and non-numerical orders.

The third empirical finding is the end effect, which is specific to non-numerical orders. It refers to the fact that pairs with end elements (in the example, A and F) are easier than pairs with middle elements. In the military example, comparing a president with a general (both elements near the end of the sequence) would be easier than a lieutenant to a sergeant (both elements near the middle of the sequence). This is again reported both in humans (Leth-Steensen & Marley, 2000) and in animals (Van Elzakker et al., 2003). In numerical orders, the end effect is subsumed by the more salient size effect, whereby pairs containing larger elements are more difficult (Banks, Fujii, & Kayra-Stuart, 1976; Moyer & Landauer, 1967; Schwarz & Stein, 1998).

The fourth empirical finding is the semantic congruity effect: When the instruction is to choose the larger of two elements, the task is relatively easier for large than for small stimuli. Vice versa, if the instruction is to choose the smaller of two elements, this is relatively easier for small elements. For example, identifying the larger of 8 and 9 is easier than identifying the smaller of 8 and 9 (in comparison to the same instructions for pair 1 2, e.g.). This has been found numerous times in humans (e.g., Banks, 1977; Cech, Shoben, & Love, 1990) and more recently in animals (Cantlon & Brannon, 2005). Furthermore, the semantic congruity effect is context-dependent, meaning that a pair (CD, say), is easier with the instruction “choose larger” than “choose smaller” in a small range (e.g., in the range A to D, where CD is the largest pair); but the same pair (CD) is easier with the instruction “choose smaller” than “choose larger” in a large range (e.g., in the range C to F, where CD is the smallest pair). Again, this is observed both in humans (Cech & Shoben, 1985) and in animals (Jones, Cantlon, Merritt, & Brannon, 2010).

Order Processing: Theories

These findings, obtained across various experimental designs and animal species, have generated a host of theories about order processing. We summarize them here.

Continuum Models

Continuum explanations are perhaps the oldest account of order processing. To interpret the origin of the distance effect in Arabic numbers, Moyer and Landauer (1967) proposed an internal continuum on which Arabic numbers are placed for purposes of comparison. To account for the size effect in number processing, they proposed that the continuum is compressed so that processing larger numbers is more difficult (e.g., Dehaene, 1992). To account for semantic congruity and the end effect in non-numerical orders, later authors added the assumption that comparison is not performed between the two to-be-compared elements. Instead, each element is separately compared with a reference point (Holyoak, 1978; Jamieson & Petrusic, 1975). Typically, the reference point is taken to be the smallest (A) or largest (F) element in the sequence. For example, if elements B and C have to be compared, and the reference point is F, then distances F-B and F-C are computed. In a subsequent step, these two distances (F-B and F-C) are compared with one another to determine which one is closer to the reference point, thus determining which element is smaller or larger. If performance is determined by the ratio between these two distances, then this principle can account for the distance and semantic congruity effects. However, it can account for the end effect only in pairs that are congruent with the instructions. Nevertheless, the continuum model (but usually without the reference) became the standard model in numerical cognition (e.g., Brannon, 2006; Dehaene, 1989; Walsh, 2003; Zorzi, Priftis, & Umilta, 2002).

Semantic Coding

An alternative to the continuum model was proposed by Banks and colleagues (e.g., Banks et al., 1976). Here, an instruction to compare two elements generates two linguistic size codes (e.g., “large” and “small”). When the size codes are the same (“small” and “small”), new codes have to be constructed (e.g., “very small” and “small”). When elements are close, the linguistic codes tend to be similar, so new codes have to be constructed. Thus, a distance effect is obtained. When the instruction is to find the larger of two elements, the “very small” and “small” codes of the example have to be reversed in polarity (e.g., to “not large at all” and “not so large”, respectively) thus to determine that the second element is larger than the first element. This generates a semantic congruity effect.

Reinforcement-Learning (RL) Theories

RL models have been applied extensively to decision making (e.g., Silvetti, Seurinck, & Verguts, 2011; Sutton & Barto, 1998).
Less frequently, they have been applied to order processing, in particular to account for transitive inference in (nonhuman) animals (Couvillon & Bitterman, 1992; Dehaene & Changeux, 1993; Frank, Rudy, & O’Reilly, 2003; von Fersen et al., 1991). Some of these models cannot account for high accuracy in human order processing. For example, the learning rule for changing weights between stimuli (indexed \(i\)) and responses (indexed \(j\)) used by Dehaene and Changeux is as follows:

\[
\Delta w = (\beta - 1)w + \gamma \cdot B \cdot R \cdot x_i \cdot (2x_j - 1)
\]

Here, \(\gamma\) and \(\beta\) are constants, the variable \(R\) indicates whether there was reward on that trial or not (1 or 0), and \(x_i\) and \(x_j\) are the activation of stimulus and response units, respectively. When training occurs on adjacent pairs only, each stimulus (except the end elements) is reinforced in 50% of all trials (e.g., B is reinforced when combined with C, but not when combined with A). Application of this rule leads to the same weight for each of the nonend elements. As a consequence, the model cannot distinguish between the elements. So, a model based on reinforcing individual elements, cannot account for even the most basic data pattern, high accuracy. Some RL-based models of order processing attempt to deal with this issue (e.g., Frank et al., 2003; von Fersen et al., 1991), but even when allowing extra degrees of freedom, such models have not been shown to be able to account for high accuracy. For example, on pair DE, model performance was below 60% in Couvillon and Bitterman’s model. For modeling the animal learning literature, this is not problematic, because here accuracy is typically modest (e.g., on pair DE, between 60 and 70% in the rodent subjects of von Fersen et al., 1991). However, in human subjects accuracy is typically much higher (e.g., consistently above 90% in Leth-Steensen & Marley, 2000).

**Delta-Rule Models**

Leth-Steensen and Marley (2000) proposed a process model to account for the full range of data in human transitive inference tasks (cf. also Page et al., 2004; Shultz & Vogel, 2004). Their model has at its core a single layer of weights between input elements and responses, trained by the delta rule. The delta-rule model of Leth-Steensen and Marley learned, using trial-to-trial feedback, an ordering of the elements A to F. During training, when the task was “choose larger,” it constructed connection weights from input (elements A to F) to response units, with monotonic ordering of the weights. For one input set, it had larger weights for larger elements; for the other input set, it had larger weights for smaller elements. As noted above, a similar approach was followed in a delta-rule model for numerical cognition (Van Opstal, Gevers, De Moor, & Verguts, 2008; Verguts et al., 2005).

Besides the core delta-rule model, Leth-Steensen and Marley (2000) also added some other components to fit their empirical data. One component was added to account for the end effect. In particular, extra boundary detection units computed the distance of each element separately to the two end points (e.g., A and F). Smaller distances led to extra input to the correct response units. Another component was added to account for the semantic congruity effect. In particular, when the task requirement was “choose larger,” larger elements decreased the competition between the two possible instructions (“choose larger” and “choose smaller”), thus facilitating the instruction “choose larger” specifically for larger elements. A similar but reverse effect occurred for the instruction “choose smaller.”

Each of these earlier models can account for (a subset of) findings in either the numerical or non-numerical domain. However, no model has been applied across domains, despite their obvious commonalities both empirically and theoretically. We confront this issue in the current article. For this purpose, we follow up on the earlier delta-rule models. Extending the strategy of Verguts et al. (2005), we exploit their elementary mathematical properties in interaction with the frequency distribution that elements are sampled from. This leads to alternative conceptualizations of the effects described above. Perhaps most important, it unites numerical and non-numerical order processing, demonstrating that a single computational process may underlie both.

**Data**

Before embarking onto the modeling section, we briefly summarize two prototypical data sets that will be used as our target for non-numerical (Leth-Steensen & Marley, 2000) and numerical (Banks et al., 1976) orders, respectively. Further details can be found in the original papers.

Leth-Steensen and Marley (2000) trained their subjects on a novel sequence of six arbitrary names. Both instructions (“choose smaller” and “choose larger”) were presented randomly mixed. Data are shown in Figure 1a (distances 1 to 3, to be comparable to the numerical data). Consistent with the discussion above, there is a distance effect (large distances lead to faster RTs); there is an end effect (RTs are faster for elements at either end of the sequence); and there is a semantic congruity effect (RTs are relatively faster for congruent instructions).

Banks et al. (1976) administered all number pairs consisting of numbers 1 to 9, with distances 1 to 3. Both the “choose smaller” and “choose larger” instructions were administered (randomly mixed). Figure 1b reports the effects for numbers 1 to 6 (in order to be comparable to the Leth-Steensen and Marley data; but see Supplementary Material, Simulation S4). First, there is a distance effect. Second, there is a size effect (RTs are slower for bigger numbers). Third, there is a semantic congruity effect.

**Model**

An outline is shown in Figure 2. The model was trained on a six-element sequence A to F. Input units code for the elements (A–F), and output units code for the relevant response. Input coding was localist, with activation \(x_i\) for unit \(I\), coding for element \(I\) (\(I = 1\) to 6 for left element; \(I = 7\) to 12 for right element). There are no hidden units. The model is trained on the instruction “choose the larger of the two elements” (Figure 2a), with response options “Left element larger” or “Right element larger”. Alternatively, the model is trained on the instruction “choose the smaller of the two elements” (Figure 2b), with response options “Left element smaller” or “Right element smaller.” In actual experimental settings, the stimuli may be presented sequentially or in a vertical arrangement (up – down rather than left – right). The model can easily be relabeled at both input and response level according to such alternative settings. The two response units are activated according to the standard sigmoid (logistic) function.
y_j = \frac{1}{1 + \exp(-\sum_i w_{ij} x_i)} \quad (1)

for j = 1, 2. The value w_{ij} is the weight between input unit I and response unit j. These weights are learned using the delta rule (Rescorla & Wagner, 1972; Widrow & Hoff, 1960):

\Delta w_{ij} = \lambda (t_j - y_j) x_i y_j (1 - y_j). \quad (2)

The parameter \lambda is the learning rate. This is arbitrarily set to 0.1 for direct connections (full lines in Figure 2). Direct connections connect input elements to their corresponding response. For example, the connection between input units 1 to 6 and response “Left element larger” is direct, because input units 1 to 6 code for the left stimulus. The learning rate is reduced to 60% of its direct value (i.e., to 0.06) for the cross-connections (dashed lines in Figure 2). Cross-connections connect elements to the opposite response (e.g., input units 1 to 6 to response “Right element larger”; or input units 7 to 12 to response “Left element larger”). This difference in learning rates implements the idea that each element contributes mostly to its corresponding response. For example, the left element contributes mostly to the response “Left element larger” when the task is “choose larger” (Figure 2a); and mostly to the response “Left element smaller” when the task is “choose smaller” (Figure 2b). As such, any value \lambda < 0.1 for cross-connections would be acceptable and, indeed, other values than 0.06 (but <0.1) lead to qualitatively similar results. The term \( t_j - y_j \) in Equation (2) is the deviance between what is desired (teacher signal \( t_j \)) and the obtained response (\( y_j \)). This deviance and the following terms \( x_i y_j (1-y_j) \) derive from the mathematical motivation of the delta rule, which is that it performs steepest descent in weight space to minimize error (e.g., Widrow & Hoff, 1960). The detailed training settings are described in the Simulation sections.

After training, all possible pairs are tested. In this test phase, elements are presented at input, and a race between the two responses is initiated. In particular, response activation of each response option evolves according to

\[ y_j(t) = \gamma y_j(t-1) + (1 - \tau) \sum_i w_{ij} x_i. \quad (3) \]

When one of the two responses \( (j = 1, 2) \) reaches a fixed threshold (arbitrarily chosen to be 0.4), the response and RT (number of time steps \( t \) required to reach the threshold) are

\[ y_j(t) = \begin{cases} 1 & \text{if } y_j(t) > 0.4 \\ 0 & \text{otherwise} \end{cases} \]

Figure 1. (a) Response times (RTs) from Leth-Steenensen and Marley (2000). The x-axis represents the smallest element of each pair, separately for each distance 1–3. Each curve represents a different instruction (“choose larger” or “choose smaller”). (b) RTs from Banks et al. (1976).

Figure 2. (a) Model for “choose larger” instruction. (b) Model for “choose smaller” instruction.
recorded. Clipping activation to be non-negative is not necessary because the delta rule assures that activation does not become negative. No noise term is added (see General Discussion), so for a fixed set of parameters (e.g., weights), the RT will be the same. However, different simulations may end up with different weights after training, so RTs can still differ across simulations. The integration constant \( \tau \) was set to 0.99. Note that the model dynamics are slightly different in the training and test phases for practical purposes only. Indeed, the activation rule from the test phase (Equation (3)) leads asymptotically to the activation \( y_j = \sum_i w_{ij} x_i \). Hence, if the nonlinearity in Equation (1) is applied after response competition, then the model dynamics (activation and learning) are exactly the same in the training and test phase. All results were robust to changes in the parameter settings.

**Simulation 1: Equal Frequencies**

**Method**

The model was trained for 10,000 trials on each of the two instructions (“choose larger” and “choose smaller”). For each instruction, this procedure was replicated 100 times; results are averaged across the 100 replications. Before each replication, weights were set to random numbers from a uniform distribution between 0 and 1. As in typical experimental paradigms, training occurred on adjacent pairs only (e.g., AB, BC). Each training pair was equally likely; in other words, elements were sampled from a uniform distribution. Testing was done on all pairs. This was done just once because, as mentioned above, for a fixed set of parameters, the simulated RT is always the same.

After obtaining simulated RTs, we linearly regressed mean RTs (across replications) onto the data of both Leth-Steensen and Marley (2000; Figure 1a) and of Banks et al. (1976; Figure 1b) (for a similar approach, see Chen & Verguts, 2010). For Banks et al., only distances 1 to 3 were available, so for consistency regression was performed on the same distances for the Leth-Steensen and Marley data too. Hence, each model fit entails two estimated parameters (slope and intercept), for the Leth-Steensen and Marley data too. Hence, each model fit.

Accuracy is high because the weights are monotonically increasing. In particular, a larger left element activates the response “Left element larger” more strongly than a smaller left element. There is a distance effect because of the monotonicity of the weights (Figure 4a). This causes the activation of the correct response unit for two close elements (e.g., A and B, presented as left and right elements, respectively) to be smaller

**Results and Discussion**

After training, accuracy was very high (99.7%). The fit to the Leth-Steensen and Marley data (\( R^2 = .56 \)) was higher than to the Banks et al. data (\( R^2 = .28 \)). In Figure 3a, we plot the simulated data (linearly regressed to the empirical data in Figure 1a). As in the empirical data, simulated RTs are slower for adjacent (distance = 1) pairs than for nonadjacent pairs (distance >1). In general, RTs are faster when the distance between elements is larger (distance effect). Furthermore, end elements are faster than middle elements (end effect). Finally, there is a semantic congruity effect. For small elements (A, B), the instruction “choose smaller” is easier; for large elements (E, F) the instruction “choose larger” is easier.

To understand why the model captures this data pattern, consider Figure 4a. We plot the average weights (across 100 replications) for the instruction “choose larger.” Weights of individual simulations are not plotted, but are similar to the average weights. In fact, the standard error across replications is smaller than the marker used to plot the dots in Figure 4a. We plot the weights from the left element to response “Left element larger” (direct connections, left part of Figure 4a); and the weights from the right element to response “Right element larger” (direct connections, right part of Figure 4a). The weights to the other response (cross-connections, e.g., left element to response “Right element larger”) are not plotted, but in general, they are decreasing rather than increasing, and (in absolute value) smaller than the direct (plotted) weights. This is because of their smaller learning rate. Table 1 reports the average weights for Simulation 1 (including cross-connections).

Accuracy is high because the weights are monotonically increasing. In particular, a larger left element activates the response “Left element larger” more strongly than a smaller left element. There is a distance effect because of the monotonicity of the weights (Figure 4a). This causes the activation of the correct response unit for two close elements (e.g., A and B, presented as left and right elements, respectively) to be smaller.
than if two more distant elements are shown (e.g., A and D, presented left and right, respectively). For example, assume for simplicity that left elements $i_L$ have weight equal to $i_L$ to the response “Left element larger” (i.e., linearly increasing function), and right elements $i_R$ have weight $10 - i_R$ to the same response “Left element larger” (i.e., linearly decreasing function). If the left number is larger, then the “Left element larger” response is correct, and it receives net input $\sum_i w_{ij} x_j$ which in this example reduces to $10 + i_L - i_R$. Hence, if the distance between the elements is large, so will be the net input into the correct response unit, which will lead to fast RTs (cf. Equation [3]). See Van Opstal et al. (2008) for a detailed exposition of this argument (cf. also Cohen Kadosh, Brodsky, Levin, & Henik, 2008). Note that this mechanism also allows generalization to untrained pairs. For example, the model receives training on pairs DE and EF, not on DF. With the instruction “choose larger,” the weights of F to “right larger” will be larger than the weight of E (to solve the pair EF). Likewise, the weight of E will be larger than the weight of D (to solve the pair DE). Hence, the weight of F must necessarily also be larger than the weight of D, allowing solution of DF.

The end effect derives from the nonlinearity in Equation (1). This nonlinearity (logistic function) is schematically depicted in Figure 5. Because of the delta rule (Equation [2]), the model attempts to space the elements equally in terms of activation at response level. Hence, the model must space the net input more strongly for end elements. Indeed, in a logistic curve, distances must be stretched more when they are closer to the end points, for equal differences in function value (cf. Figure 5). As a consequence, the weight curve (Figure 4a) develops a sigmoid (nonlinear) shape, where the end points are more stretched out than the middle ones. Note that this stretching is not an added assumption of the model, but follows from the model formulation in Equations (1) through (3). Also, it does not derive specifically from this logistic curve; it is obtained with any bounded and monotonic transformation function (Figure 5).

The semantic congruity derives from the fact that, with the “choose larger” instruction, large elements build up activation more strongly than small elements (Figure 4a). This happens as a consequence of the cross-connections being smaller (in absolute value) than the direct connections. See Supplementary Material Simulation S1 for further exploration of this effect. The reverse argument holds for the “choose smaller” instruction. The context-dependency of semantic congruity (e.g., Cech & Shoben, 1985; Jones et al., 2010) is consistent with the model if the two separate contexts each develop their own weights. For example, elements C and D would receive large weights in a small-element context (when only elements A to D are trained); but would receive small weights in a large-element context (when only elements C to F are trained).

Finally, we note that the results are very similar (not explicitly reported) when training occurred on all pairs, rather than just the adjacent ones. The reason for this insensitivity to the training set, is that the best weight solution for adjacent pairs (e.g., $w_A < w_B < w_C < w_D < w_E < w_F$) is also the best solution for nonadjacent pairs because both adhere to the same unidimensional structure.

### Table 1

<table>
<thead>
<tr>
<th></th>
<th>Simulation 1</th>
<th>Simulation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left larger</td>
<td>Right larger</td>
</tr>
<tr>
<td>Left element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-3.84</td>
<td>3.15</td>
</tr>
<tr>
<td>2</td>
<td>-1.58</td>
<td>1.86</td>
</tr>
<tr>
<td>3</td>
<td>-0.55</td>
<td>0.27</td>
</tr>
<tr>
<td>4</td>
<td>0.55</td>
<td>-0.31</td>
</tr>
<tr>
<td>5</td>
<td>1.53</td>
<td>-1.88</td>
</tr>
<tr>
<td>6</td>
<td>3.80</td>
<td>-3.19</td>
</tr>
<tr>
<td>Right element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.95</td>
<td>-3.52</td>
</tr>
<tr>
<td>2</td>
<td>1.71</td>
<td>-1.30</td>
</tr>
<tr>
<td>3</td>
<td>0.44</td>
<td>-0.73</td>
</tr>
<tr>
<td>4</td>
<td>-0.39</td>
<td>0.76</td>
</tr>
<tr>
<td>5</td>
<td>-1.68</td>
<td>1.33</td>
</tr>
<tr>
<td>6</td>
<td>-2.92</td>
<td>3.55</td>
</tr>
</tbody>
</table>

Note. Nonitalicized weights are plotted in Figure 4a (Simulation 1) and Figure 4b (Simulation 2). Numerical values for other simulations can be obtained from the first author.

![Figure 4](image-url)

**Figure 4.** (a) Average direct connection weights (across 100 replications) in Simulation 1. (b) Average direct connection weights (across 100 replications) in Simulation 2.
Simulation 2: Skewed Frequencies

In Simulation 1, elements were sampled from a uniform distribution. This corresponds to the typical practice in order learning experiments. However, symbolic number processing is usually (i.e., in daily life) trained with smaller numbers presented more frequently (Dehaene & Mehler, 1992). To mimic that situation, in Simulation 2 elements are sampled from a skewed distribution in which smaller elements appear more frequently (error-corrected exponential distribution, see below). Because of this change, we now interpret the elements of the order as numbers and use the terms interchangeably.

Method

Numbers (elements) $i$ were sampled during training from an exponential distribution (frequency proportional to $\exp(-\alpha \times i)$). However, if recent accuracy on a given number was below a fixed threshold, it received a small boost so it would be sampled more often. This represents more faithfully the educational practice where problems that children have difficulties with, are emphasized (i.e., their frequency increased) on an individual basis. Formally, elements were sampled from an error-corrected exponential distribution,

$$p(i) \propto \exp(-\alpha \cdot i) + \varepsilon_i \quad (4)$$

where $\alpha$ is the exponential rate parameter, currently set at 1. Furthermore, $\varepsilon_i$ is a variable indicating (0 or 1) whether the previous error score at the response level for element (number) $i$ was above an arbitrarily chosen threshold of 0.1. Note that with sufficient practice, the error-corrected exponential reduces to an exponential distribution. Simulation S2 in Supplementary Material reports the case when the error-correction mechanism is removed.

Results and Discussion

Mean accuracy was 99.9%. In contrast to Simulation 1, the fit to the Banks et al. data ($R^2 = .74$) was now higher than to the Leth-Steensen and Marley data ($R^2 = .05$). Figure 3b shows the simulated RTs (cf. data in Figure 2b). There is again a distance and semantic congruity effect, for the same reasons as in Simulation 1. However, the end effect has changed to a size effect. As a consequence, the semantic congruity effect takes the form of a funnel; RT curves increase for both instructions, but more steeply for the “choose smaller” instruction.

In Figure 4b, we plot the weights in the same format as in Simulation 1. Table 1 again reports the full set of average weights. The weights now exhibit a compressed pattern in the sense that weights are spaced more closely together for larger numbers. This generates the size effect, as explained in detail in Verguts et al. (2005). Basically, because weights for large numbers are close together (compressive weight pattern), net inputs to the response units are reduced, which slows down RT. Finally, in the Supplementary Material we report a Simulation (S4) where the number of elements was nine instead of six.

General Discussion

We presented a model of order processing where empirical phenomena followed from elementary neural network properties in combination with environmental properties. This unites the domain of order processing (numerical and non-numerical). This approach, where numbers are connected to other symbolic sequences is in line with recent developmental observations (Bertelletti, Lucangeli, & Zorzi, 2012), and complements earlier approaches where numbers are connected to visuospatial processing (Grossberg & Repin, 2003; Roggeman, Fias, & Verguts, 2010; Stoianov & Zorzi, 2012). Furthermore, it connects order processing to a broader tradition in psychological modeling where ele-
mentary network properties shed light on cognition (e.g., Pouget & Sejnowski, 2001; Seidenberg & McClelland, 1989).

One crucial network feature was the use of a delta rule. Indeed, with modulated Hebbian learning (e.g., the reinforcement learning models mentioned above), it is not even possible to learn an order where adjacent elements only are presented. A second crucial property was the nonlinearity in response units. This is biologically plausible, as neurons can only produce a limited number of spikes per time unit because of their refractory period. Moreover, this nonlinearity has been used to explain various phenomena ranging from larger incongruency than congruency effects in the Stroop task (Cohen, Dunbar, & McClelland, 1990) to the age-of-acquisition effect (Ellis & Lambon Ralph, 2000).

Yet another tradition our work connects to is exploiting environmental properties to understand cognition (e.g., Anderson, 1991; Griffiths & Tenenbaum, 2006; Seidenberg & McClelland, 1989). Besides the element frequency distribution, another possible difference (but currently not investigated) is the fact that numerical orders have a visuospatial meaning related to collections of objects, which may partially drive relevant effects when number is presented as an object collection (Verguts & Fias, 2004; Zorzi, Priftis, Meneghello, Marenzi, & Umiltà, 2006). In particular, the size effect in nonsymbolic number processing could partially derive from other factors than those proposed here. For example, if every element has a constant noise factor, noise will scale up with larger numbers even when frequency is constant. Also, numbers (more than other sequences) are often represented in a left-to-right setting in Western schools, which may also drive at least some relevant effects (Chen & Verguts, 2010; Zorzi et al., 2002).

Our model also contained some simplifications. First, at least for non-numerical orders that are sometimes learned in the context of a few hours time, the number of training trials was quite high. It should be kept in mind however, that a “trial” in a simulation does not necessarily correspond to a trial in an experiment. Perhaps a process similar to what is assumed here occurs several times in an experimental trial. Also, implementing more trials than actually presented to subjects is common practice in neural network modeling (but see Leth-Steensen & Marley, 2000).

A second simplification was the absence of a noise term in the model. If we add a Gaussian noise term (extra term on right-hand side of equation (3)), the model becomes formally identical to an Ornstein-Uhlenbeck model if the time steps are made infinitesimally short (Busemeyer & Townsend, 1993). This and related models (Bogacz, Brown, Moehlis, Holmes, & Cohen, 2006; Ratcliff, 1978) have been used extensively to account for RT distributions. Because of this formal similarity, it follows immediately that our model generates plausible RT distributions (i.e., with positive skew) with such a noise term.

Third, we indicated above that RL-based models have problems accounting for the high accuracies in the human learning literature. Correspondingly, our own model does not address the more modest accuracies from the animal learning literature. As such, the current model integrates the human (numerical and non-numerical) data, but not the animal data. Integrating these two literatures remains an issue for further research.

A related issue of interpretation concerns the status of the “weights” in the model. Although neural network weights are often interpreted as long-term memory weights, this is not necessary. In fact, similar biological processes may underpin short-term and long-term memory synaptic changes (Lisman, Grace, & Duzel, 2011). Given the (developmental) time scale at which numbers are learned, we interpret the numerical order weights as long-term weights. In contrast, in many non-numerical orders, in particular those learned at a time scale of at most a few hours, the weights more plausibly reflect short-term synaptic changes (and thus constitute working memory; Lyons & Beilock, 2009; Van Dijck & Fias, 2011). However, different orders may operate at different time scales, and still exhibit similar regularities.

Another issue is that we used just one input coding type, namely localist coding. In particular, each stimulus was represented by exactly one unit. This was done in continuity with earlier work (Verguts et al., 2005), but later work should explore more general coding types.

As a final limitation, we discuss the finding that in some studies differential empirical effects are observed when a task is formulated to the participant as either ordering elements (order judgment) or choosing the larger element (quantity judgment). For example, Turconi, Campbell, and Seron (2005) obtained distance effects for quantity judgments but reversed distance effects for an order judgment task, at least for small distances (cf. also Franklin & Jonides, 2009; Franklin, Jonides, & Smith, 2009). At a general level, the model is not inconsistent with such dissociations. Indeed, even though the two tasks are formally equivalent, each task instruction may have received differential training across developmental history. However, we did not explicitly predict such dissociations. One possibility is that a scanning mechanism (which leads to reverse distance effects; see Franklin & Jonides, 2009; Franklin et al., 2009) is used in some contexts. However, this interpretation remains to be fully specified and tested.

Whereas numbers were identified explicitly, all other orders are aggregated as “non-numerical”. As a consequence, it is difficult to make precise predictions about each possible order because its statistics are typically unknown. However, the model does produce a framework to generate predictions for different types of orders. In particular, when there is no frequency bias toward small (or large) numbers, a bow-shaped curve will be obtained. This bow will be transformed into a size effect if frequencies are sufficiently skewed. If frequencies are skewed, but not as strong as for numbers, a combination between a bow-shaped and increasing curve is obtained. This mixed size/bow effect is indeed observed when the comparison task is performed using letters A to I (Jou, 2003).

Another prediction concerns the overall architecture of numerical and non-numerical orders. Whereas the input units for numerical and non-numerical orders are separated, their decision units (“smaller”, “larger”) can in principle be shared. So, from the present perspective, numerical and non-numerical orders should not be clearly shared or separate; the decision part can be shared, and the representational part not. There is indeed preliminary evidence for a combined association/dissociation (Fias, Lamme, Caensens, & Orban, 2007; Nieder, Diester, & Tudusciuc, 2006; Thioux, Pesenti, Costes, De Volder, & Seron, 2005; Zorzi, Di Bonito, & Fias, 2011), but this could be tested more directly using fMRI adaptation. Such a test would be similar to how Roggeman, Santens, Fias, and Verguts (2011) identified separate stages in nonsymbolic number processing. Relatively, the model predicts that the two task instructions (“choose smaller” and “choose larger”) should be at least partially separable. We don’t expect standard fMRI contrasts to have the required spatial reso-
solution to distinguish between such instruction areas, but this test should be possible using multivoxel pattern analysis (Pereira, Mitchell, & Botvinick, 2009). A final prediction, but more difficult to test or even formalize precisely, is the existence of “order” representations from this perspective. In the model, order is represented only implicitly, via a gradient of weights connected to a decision system. We showed that this gradient is sufficient not only for solving the (comparison) task, but also for accounting for empirical patterns reported in the literature. However, there are data patterns that the model cannot account for. For example, Nieder et al. (2006) showed in a temporal version of a comparison task that some cells respond specifically to a given position in the sequence. The point of our model is not so much that this is the final word on order representation; instead, in line with a neural network approach, we argue that representation and decision processes develop depending on the task that the organism is confronted with. Hence, different (order) representations and order-relevant decision processes can be expected for organisms that are trained using different tasks.

In the remainder of the General Discussion, we revisit the four broad model classes that were identified in the Introduction and compare it with the current model.

One similarity between our own and the continuum model is that relevant order is contained in the model structure. It is represented along a continuum in the former, and in the connection weight pattern in the latter. However, an important difference is that our model provides a computational rationale for both how and why such a pattern develops. Moreover, it predicts that this pattern of weights should be specifically used in a comparison task, not in other tasks, a prediction we recently verified empirically (Van Opstal & Verguts, 2011; Verguts & Van Opstal, 2005).

Most importantly, the continuum model cannot explain the qualitatively different data across numerical and non-numerical orders.

In the continuum model, the origin of the distance effect is the continuum on which numbers (elements) are represented. In contrast, the distance effects in our model, like those in the semantic coding model (Banks et al., 1976) are not continuum based. However, our model considers semantic congruity to be related to the decision level, whereas according to the semantic coding model semantic congruity is a postdecisional linguistic effect. Both aspects of the latter explanation have been criticized. First, Petrusis (1992) and Shaki and Algom (2000) demonstrated that semantic congruity is related to the decision level. In particular, when the decision process becomes more difficult, the semantic congruity effect increases. This finding suggests a gradual evidence accrual process in favor of one of the two responses, where each evidence accrual step is sensitive to semantic congruity (see Petrusis, 1992, for details). Second, Cantlon and Brannon (2005) demonstrated that the effect is not linguistic, by showing that the effect is also robustly observed in monkeys (also Jones et al., 2010). This latter aspect is consistent with our model because it contains only input elements and their corresponding responses, neither of which has to be linguistic.

In RL-based models, values are assigned to the different elements (A-F), and these values are used to solve the task. Despite our criticism of RL-based models, it is important to note that these models fail for a very specific reason, which is the absence of a delta-signal that combines the input from the two stimuli. In the RL models described in the introduction, there is no delta signal in the sense we defined it here, integrating information from the two input elements (as in Rescorla & Wagner, 1972). However, this assumption is not mandatory (see Sutton & Barto, 1998, for an overview of RL-models with a delta term). For example, a recent RL-based model (AGREL, Roelfsema & van Ooyen, 2003) uses a delta term similar to the Rescorla-Wagner rule that we have used. Hence, we also implemented AGREL and applied it to the current experimental paradigms (not shown). We observed that its results were very similar to those discussed for the model of the current paper. Hence, what matters for order processing is to have a prediction error (delta) signal as used in the delta rule. Whether this model is then formulated as RL-based or not, is not relevant.

The fourth class of models was labeled as delta-rule models because they use some variant of the Widrow-Hoff (delta) rule to obtain appropriate weights. The Leth-Steenen and Marley (2000) model generated a detailed fit to non-numerical order data of human subjects. Moreover, it showed small distance effects for end pairs (consistent with data in Figure 2a). However, this model was applied to non-numerical orders only. Similarly, the Verguts et al. (2005) model was fitted to numerical order data only. There are detailed differences between these two models, but their core architecture (one-layer network trained by the delta rule) is similar. Currently, we again applied a one-layer delta-rule model, and used the frequency distributions of Leth-Steenen and Marley (uniform) or Verguts et al. (skewed). Our work thus points toward strong similarities between numerical and non-numerical order processing, and shows how architectural-environmental interactions are able to provide order.

References


Received May 22, 2013
Revision received October 15, 2013
Accepted October 17, 2013