COGNITIVE SCIENCE Vol 24 (1) 2000, pp 93–122 ISSN 0364-0213

Computational Evidence for the Subitizing Phenomenon as an Emergent Property of the Human Cognitive Architecture

SCOTT A. PETERSON
Cameron University

TONY J. SIMON
Georgia Institute of Technology

A computational modeling approach was used to test one possible explanation for the limited capacity of the subitizing phenomenon. Most existing models of this phenomenon associate the subitizing span with an assumed structural limitation of the human information processing system. In contrast, we show how this limit might emerge as the combinatorics of the space of enumeration problems interacts with the human cognitive architecture in the context of an enumeration task. Subitizing-like behavior was generated in two different models of enumeration, one based on the ACT-R cognitive architecture and the other based on the principles of parallel distributed processing (PDP). Our results provide good qualitative fits to results obtained in a variety of empirical studies.

I. INTRODUCTION

In the typical visual enumeration experiment, human participants are instructed to quantify collections of visually-presented objects as quickly and accurately as possible. Reaction times tend to increase with numerosity, although not in a strictly linear fashion as one might first expect (Figure 1a). Instead, aggregated reaction time data usually exhibit a marked discontinuity at about $N = 4$. The prototypical finding (Figure 1b) is a bilinear function with a shallow slope of about 50 ms/object for small collections (up to three or sometimes four objects) and a much steeper slope in the range 250 to 300 ms/object for

Direct all correspondence to: Scott Peterson, Department of Psychology and Human Ecology, Cameron University, 2800 West Gore Blvd., Lawton, OK 73505; E-Mail: scottp@cameron.edu

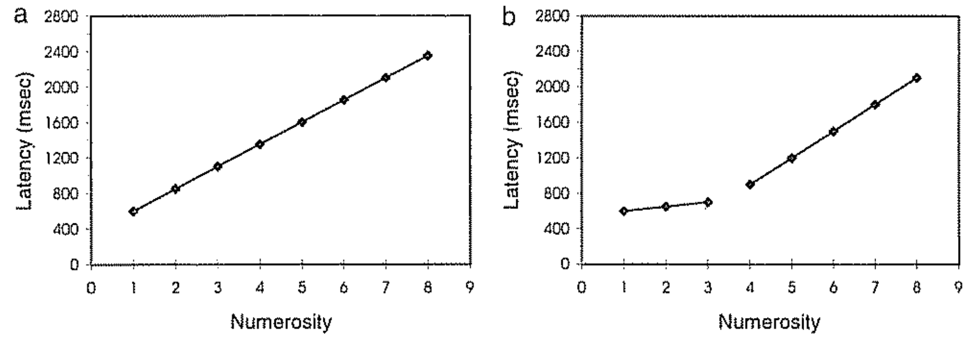

Figure 1 Theoretical latency profiles for visual enumeration task: linear model (a) and bilinear model (b)

larger collections (e.g., four or more objects) This discontinuity in slope is usually taken to reflect deployment of different enumeration processes in the two numerosity ranges. The process associated with enumeration of small collections is commonly referred to as subitizing (Kaufmann, Lord, Reese, & Volkman, 1949). Researchers studying the phenomenon of subitizing are faced with the task of explaining its two defining characteristics: 1) the limited range of numerosities on which it operates (subitizing limit), and 2) the relatively small reaction time increase from one numerosity to the next within this range (subitizing slope). Although its investigation has continued for well over a century (e.g., Jevons, 1871), a definitive explanation of subitizing has not yet emerged.

The central aim of the present study was to investigate an alternative to the dominant explanation for the subitizing limit. Most previous accounts of subitizing explain its limit in terms of some structural limitation of the human information processing system. For example, Trick and Pylyshyn (1993, 1994) suggest that the subitizing limit is related to a fixed number of attentional tags (which they call FINSTs) that can be assigned to discrete entities in the visual field. Assigned after preattentive feature registration and grouping operations, these tags mark items for subsequent attentional processing. According to Trick and Pylyshyn, the subitizing limit is four because human adults only have four FINSTs. Similarly, computational models of subitizing (e.g., Anderson, Matessa, & Lebiere, 1997; Klahr & Wallace, 1976) have tended to operate with precoded recognition rules for just the first three or four items. Because there does not seem to be any independent evidence of this sort of structural limitation, in the present study we examined whether the subitizing phenomenon could be explained without relying on the existence of such a structural limit.

The question of interest can be stated as follows: Why does the human cognitive architecture produce the profile in Figure 1b rather than the profile in Figure 1a when presented with an enumeration task? This question is not easily answered through empirical observation alone because the mechanisms responsible for performance can only be inferred from the data. An alternative research strategy, employed here, is to utilize a computational model of human cognitive processing that can be studied in a more direct manner. Models of the human cognitive architecture such as Soar (Newell, 1990) and ACT (Anderson, 1983, 1993; Anderson & Lebiere, 1998) attempt to encapsulate the general principles and mechanisms of human cognition in a way that is open to detailed inspection and analysis. Our general approach was to use such an architecture as our theoretical basis, augment it with the capabilities necessary to perform the enumeration task, and then attempt to determine why the model behaves as it does in the context of a given set of enumeration problems. By adopting an architecture whose theory is inconsistent with a structural limit explanation of subitizing, we can examine the necessity, or otherwise, of such an account to explain the phenomenon. Thus, this study was concerned less with how subitizing works than in explaining why it is limited to the enumeration of so few items.

II. SUBIT-R: AN ACT-R MODEL OF SUBITIZING

As a starting point, we selected the ACT-R cognitive architecture (Anderson,1993) and used this environment to develop a production system model of enumeration, called SUBIT-R.[1] To enable our model to carry out the enumeration task, we implemented two simple enumeration methods: counting and recognition. Counting relies on the existence of counting knowledge, represented in the form of an ordered sequence of number facts between one and ten, and a counting procedure. The counting procedure[2] is implemented as a set of production rules that uses the number facts to successively assign numbers to objects in a display. Recognition is implemented as a simple pattern-matching procedure that can match a given configuration of objects to a remembered configuration with known numerosity. In this case, the numerosity of the current display is directly retrieved from memory via the remembered instance.

For any given enumeration problem, SUBIT-R must select and use one of these two enumeration procedures. The counting procedure can be used for any configuration, so is always an available option. The recognition procedure, on the other hand, can be used only when the given configuration is familiar enough (i.e., has sufficiently strong memory trace) that it can be retrieved from memory. For highly familiar patterns, then, both the counting and recognition procedures are available options. In these situations, the enumeration procedure to be used is determined by ACT-R's conflict resolution mechanism (see Anderson, 1993, chap. 5), which decides among multiple competing production rules. This decision is made on the basis of expected values, computed as expected gain (or expected value of the goal) minus expected cost (of achieving the goal by this means). SUBIT-R incorporates a bias for using the recognition procedure whenever possible (i.e., a higher expected gain for recognition than for counting), consistent with the idea that recognition generally leads to a faster and more accurate response. Thus, the recognition procedure is selected whenever there is a sufficiently strong memory trace for the given configuration, and the counting procedure is selected otherwise.

As described by Anderson (1993, chap. 4), the strength of a trace in human memory seems to depend on both the number of past references and the elapsed time between them, increasing according to a power law of practice and decreasing according to a power law of delay. These "behavioral laws" are implemented by ACT-R's base level learning mechanism, which provides the learning substrate for SUBIT-R.[3] When a particular configuration of objects, or pattern, is enumerated for the first time (using the counting procedure), a declarative memory element representing the pattern and its associated numerosity is created. A subsequent exposure to the same pattern results in an activation increase (i.e., a stronger memory trace). Conversely, a decay, or decrease in activation occurs during cycles when the pattern is not referenced. Successful retrieval of a declarative memory element, that is, operation of the recognition procedure, is possible only when the element's activation level is above a specified retrieval threshold, a parameter of the ACT-R system. When recognition is not possible, the counting procedure can be used, as mentioned previously. Thus, selection of enumeration procedure is mediated by activation levels of patterns in declarative memory; counting is used for patterns whose activations are below the retrieval threshold and recognition is used when the retrieval threshold is exceeded. The behavior that emerges from this processing is a general progression from counting to recognition with experience as patterns increase in activation.

The importance of this sort of progression from algorithmic solution to memory retrieval has been recognized by researchers interested in explaining different forms of skill acquisition. For example, in his instance theory of automaticity, Logan (1988) claims that development of a wide variety of competencies can be explained in terms of a gradual shift from use of an algorithm to retrieval of specific instances from memory. Siegler (e.g., Siegler & Robinson, 1982; Siegler & Shrager, 1984; Siegler & Shipley, 1995) has simulated this sort of strategy usage shift in the domain of arithmetic fact learning. We reasoned that the same type of learning might also be operating in the domain of object enumeration, and furthermore, may underlie the emergence of the subitizing limit. This is because an important aspect of the enumeration domain is the combinatorics of the space of enumeration problems. The finite set of possible configurations of small collections of objects is relatively small whereas the number of possible configurations of larger collections increases rapidly with the number of objects. Thus, an agent operating in this domain is likely to be exposed frequently to each of the small numerosity patterns but relatively infrequently to the larger ones. This should differentially strengthen memory traces for the small numerosity patterns to the extent that their numerosities can be directly retrieved whereas larger numerosity patterns must continue to be enumerated by the counting procedure (cf., Mandler & Shebo, 1982; Wolters, van Kempen, & Wijlhuizen, 1987).

Recognition of Subpatterns

As Frick (1987) has pointed out, empirical data for enumeration of numerosities in the counting range does not seem consistent with strict usage of an item-by-item counting procedure. Strict item-by-item counting should result in a positive intercept for the computed regression line, representing for each trial a fixed "overhead" including things such as time to initiate the counting procedure and time to deliver the response. However, extension of the counting slope usually produces a negative intercept (see Figure 1b).

Frick concluded that larger numerosity collections are enumerated by "direct apprehension" (recognition) of the first four objects, and then counting of the rest. This sort of combined usage of the two enumeration procedures is also allowed in SUBIT-R. When recognition of the current collection of objects as a whole is not possible, either because the pattern has not previously been enumerated or because its activation level is below the retrieval threshold, SUBIT-R attempts to recognize a familiar (highly-activated) subset of the objects. The numerosity of this subpattern is retrieved and then the remaining objects are enumerated via the counting procedure. Thus, operation of this subpattern recognition results in further activation increases for small numerosity patterns even during enumeration of large collections of objects. When multiple subpatterns can be recognized in a particular configuration, SUBIT-R selects the largest one. This priority for larger subpatterns over smaller ones is consistent with data showing that human participants prefer to form initial chunks of 3 or 4 items (as opposed to smaller chunks) when enumerating random dot patterns (Shrager, Klahr, & Chase, 1982).

Latency Computation

For each enumeration performed by SUBIT-R, a latency value, analogous to reaction time in an experimental trial, is automatically computed by the ACT-R system. In ACT-R, the amount of time it takes a production rule to match is affected by the activation levels of declarative memory elements tested by that rule. Shorter matching times arise from testing more highly-activated items (see Anderson, 1993, chap. 3). Matching latency is added to the time to execute the production rule to produce a latency value for each production. Total latency for the enumeration task is computed as the sum of the latencies for the productions that are executed.

In SUBIT-R, recognition is handled by a single production that matches the currently presented pattern to one stored in declarative memory. Latency for the recognition procedure, then, is largely determined by the time required for a single production to match to the stored pattern. Therefore, overall latency is short and individual pattern activation has minimal effect. In contrast, the counting procedure generates relatively long latencies because it involves multiple production firings, one for each object to be counted. As a result, variability in pattern activation has the potential to create relatively large cumulative differences.

III. EXPERIMENT 1

The purpose of Experiment 1 was to determine whether presenting SUBIT-R with a standard set of enumeration problems would result in an enumeration profile consistent with the subitizing phenomenon; that is, a profile characterized by recognition of the small numerosity patterns and counting of the larger ones. In terms of latencies, subitizing-like performance should produce a bilinear latency profile similar to that shown in Figure 1b, which is typical of empirical studies.

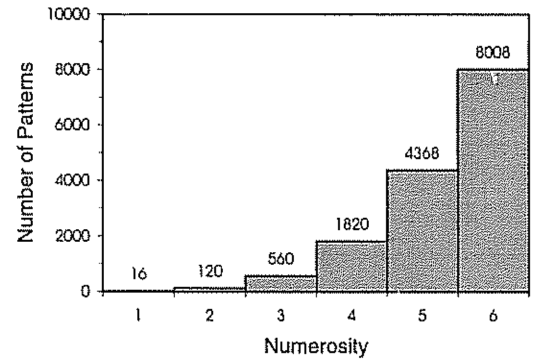

Figure 2 Number of unique patterns for each numerosity, assuming a 4 × 4 grid

IV. METHODS

Patterns

Each pattern represented one possible configuration of up to six objects on a hypothetical 4 × 4 grid of locations.[4] Generation of each pattern involved random selection of a numerosity between one and six and then random assignment of the objects to locations on the 4 × 4 grid The size of the grid constrained the number of possible configurations that could be generated using this methodology. The distribution of possible configurations across the numerosities is shown in Figure 2 As the chart indicates, there is a combinatoric increase in the number of possible patterns with increasing numerosity For example, although there are only 16 possible patterns for Numerosity 1, there are over 8000 unique patterns for Numerosity 6

Procedure

A series of simulation runs was conducted, each run consisting of a training stage and a test stage. The length of the training stage was varied across the runs, from 1000 to 25,000 trials, so that stored pattern activations and distribution of enumeration procedure usage could be examined as a function of the amount of training The test stage always consisted of 150 trials, with each of the six numerosities represented in 25 trials During each trial (both training and testing), SUBIT-R used either the recognition procedure, the counting procedure, or a combination of the two procedures (subpattern recognition, then counting) to enumerate the pattern. Latency for each trial was automatically computed by ACT-R as described above. During the test stage, enumeration procedure and latency were recorded for each trial. On completion of the test stage, activations of the patterns currently stored in declarative memory were averaged for each numerosity, providing a more general measure of memory trace strength by numerosity.

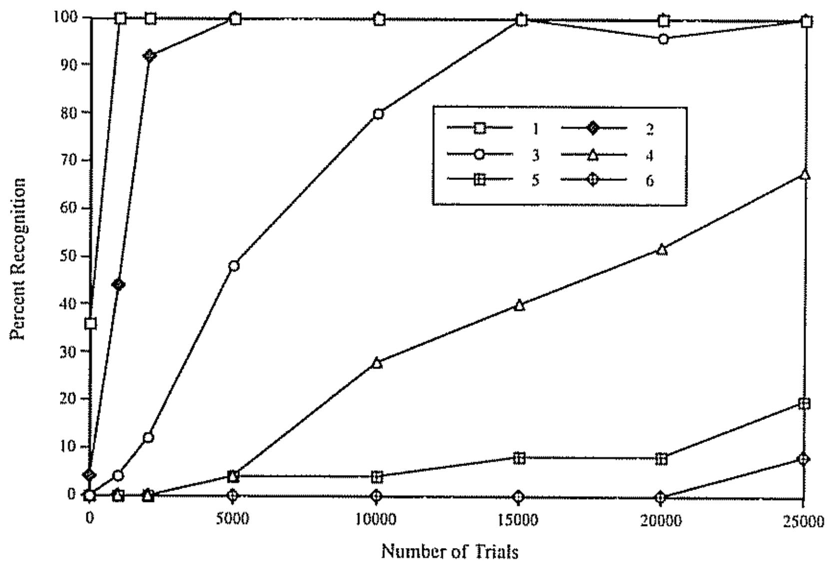

Figure 3. Percent use of recognition procedure for each numerosity as a function of the number of training trials

V. RESULTS

The effect of training on enumeration procedure usage is shown in Figure 3. In this graph, we have plotted the percentage of test trials for each numerosity in which the entire pattern (i.e., not a subpattern) was recognized, as a function of the number of training trials. Although the counting method is exclusively used early in the training stage for all numerosities, SUBIT-R seems to have learned to recognize all 16 patterns for Numerosity 1 well within the first 1000 trials. Patterns for Numerosities 2 and 3 require somewhat longer training periods, but are also enumerated nearly exclusively by the recognition procedure with relatively little training, as evidenced by the steep slopes in the graph for those numerosities. In contrast, the graph shows very little recognition of patterns for the larger numerosities, 5 and 6, indicating that they continue to be enumerated primarily by the counting procedure even after extensive training. Interestingly, the recognition rate for Numerosity 4 is consistently about midway between the two extremes. After 25,000 training trials, SUBIT-R has not yet learned to conclusively use one or the other enumeration procedure for this numerosity, as it did for the others. Rather, it persists in using a mixture of the two procedures, recognizing about 68% of the patterns and counting the remainder. These results for Numerosity 4 are reminiscent of empirical findings such as those reported by Svenson and Sjoberg (1983) who produced regression lines in the aggregated data for $N = 1$ through 3 and $N = 5$ through 8 but were unsure with which range to associate $N = 4$. There is also a notable similarity in appearance between the data plotted in Figure 3 and the time-accuracy functions (TAFs) computed by Simon, Cabrera, and Kliegl (1993), which describe accuracy as

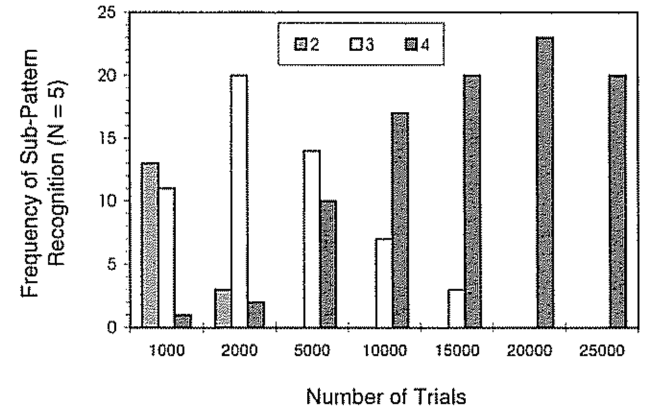

Figure 4. Frequency of subpattern recognition for test patterns with five objects as a function of the number of training trials. Each bar represents the number of times that the recognition procedure (for $N = 2$, 3, or 4) was used as a component operation of the counting procedure during the course of 25 test patterns.

a function of presentation time for individual participants. Although based on different measures, TAFs exhibit a similar sort of qualitative difference in performance between small and large numerosities.

As expected, one contributing factor to the rapid learning of small numerosity patterns was the increased activation resulting from recognition of small numerosity subpatterns appearing within patterns for the larger numerosities. The model's usage of subpattern recognition for collections of five objects is illustrated in Figure 4. As the graph indicates, almost all of the 25 enumerations in each test involve initial recognition of a smaller subpattern. Subpatterns of size two and three are frequently recognized early in training (less than 5000 trials) whereas frequent recognition of subpatterns of size four emerges with moderate training (about 10,000 trials) and increases thereafter with the amount of training. By 25,000 trials, virtually all patterns for Numerosity 5 were enumerated by first recognizing a subpattern of size four and then counting the remaining object. Similar results were obtained for Numerosity 6 and, to a lesser extent, for the smaller numerosities (a complete tabulation of recognition procedure usage by numerosity is provided in the Appendix). This pattern of results is consistent with data reported for human participants enumerating random dot patterns (Shrager et al., 1982). Both sets of data show a strong preference for the formation of chunks containing 3 to 4 items.

Because choice of enumeration procedure is largely determined by stored pattern activations, a better understanding of SUBIT-R's performance can be gained by examining how those activations vary over time, and by numerosity. The effect of training on average activation levels of the stored patterns is shown in Figure 5. Early in training, activations for Numerosities 1 through 3 increase rapidly, reaching levels well above the retrieval threshold by 5000 trials. This indicates that these patterns are being referenced often enough to compensate for the power function decay in activation occurring during

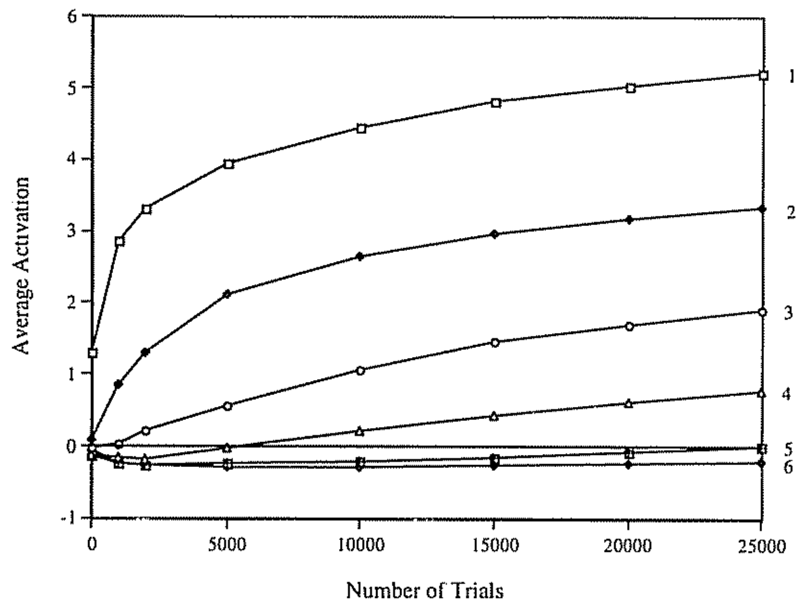

Figure 5 Average stored pattern activation for each numerosity as a function of the number of training trials

cycles when they are not referenced. Thus, recognition quickly becomes the primary enumeration procedure for these numerosities. Average activations for Numerosities 4 through 6 are affected noticeably by decay (resulting in negative activation values) and increase much more slowly, leading to the observed prolonged usage of the counting procedure. The data for these numerosities reflect the relatively infrequent repetition of each of the individual patterns. About 5000 training trials are required before the average activation for Numerosity 4 exceeds the threshold, and then activations continue to increase thereafter. These data for Numerosity 4 fit nicely with the increase in recognition of Numerosity 4 patterns and subpatterns that also occurs at about 5000 trials as can be seen in Figures 3 and 4. Average activation for Numerosity 5 achieves the threshold only after 25,000 trials whereas average activation for Numerosity 6 remains well below threshold even at that point, indicating very little opportunity for recognition of patterns for those numerosities.

Figure 6 shows the average latencies computed for a set of test trials (25 random patterns for each numerosity) presented to SUBIT-R before and after 25,000 training trials. Before training, the counting strategy is used predominantly, resulting in a roughly linear increase in latency from one numerosity to the next throughout the entire range of numerosities (as in Figure 1a). After 25,000 trials, we see a profile more consistent with the prototypical bilinear enumeration profile (Figure 1b), reflecting almost exclusive use of the recognition procedure for Numerosities 1 through 3 and more substantial use of the counting procedure for larger numerosities.

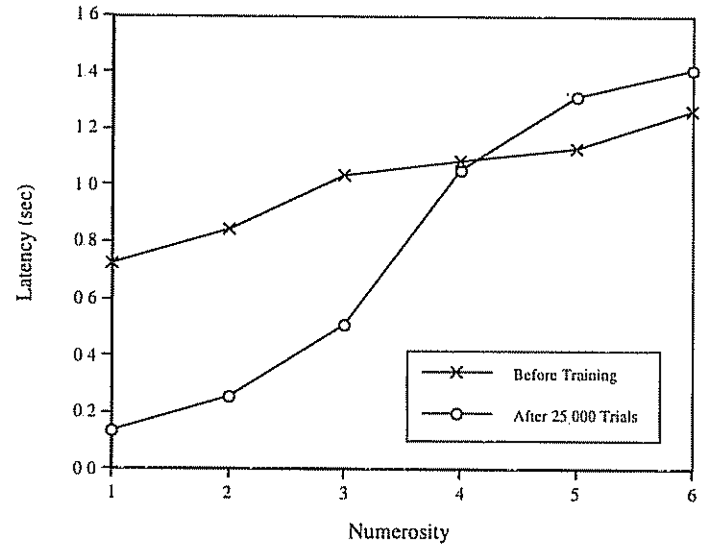

Figure 6 Enumeration latencies before and after 25,000 training trials

VI. DISCUSSION

Previous accounts of the subitizing phenomenon have tended to be descriptive in nature (e.g., Chi & Klahr, 1975, Svenson & Sjoberg, 1983), and previous models have typically relied on assumptions setting the subitizing limit at three or four objects (e.g., Anderson, Matessa, & Lebiere, 1997; Klahr & Wallace, 1976; Trick & Pylyshyn, 1993, 1994). In contrast, we have attempted to provide an *explanation* for the subitizing limit in the form of a computer simulation that generates rather than structurally embodies the subitizing limit. SUBIT-R demonstrates how this limit might emerge as a function of the combinatorics of the space of patterns interacting with the learning characteristics of the cognitive architecture. For small numerosities, where the number of possible patterns is sufficiently small, all patterns are referenced frequently enough that they become highly activated in memory. For larger numerosities, where the number of possible patterns is much greater, and thus each individual pattern is seen much less frequently, the power function of delay operates to suppress activations of the stored patterns such that direct retrieval is difficult. Thus, SUBIT-R partitions the space of numerically-varying patterns into those it can recognize (small numerosities) and those it must count (larger numerosities). As a result, the phenomenon of subitizing emerges through experience in this domain rather than being the result of a hardwired structural limit on the representational capacities of the architecture. SUBIT-R thus suggests that this subitizing limit is likely to vary by individual, being dependent on both a retrieval (or confidence) threshold and the individual's own characteristics such as learning history and possibly processing speed, working memory capacity, or other similar parameters.

In addition to demonstrating this computational explanation for the subitizing limit, SUBIT-R suggests a potential explanation for the shallow subitizing slope. As previously mentioned, recognition latency is largely determined by the time to match the current display to a pattern previously stored in memory. As matching latency increases, so does overall latency of the recognition procedure, which corresponds to response time in humans. Further, computation of matching time is based on the activation of the pattern; the higher the activation, the shorter the matching latency. So patterns with high activations will match faster than patterns with low activations, and therefore will result in shorter response latencies. Thus, the subitizing slope may arise from differences in the average level of activation across the numerosities in the subitizing range. As can be seen in Figure 5, at any given level of training, there is a general increase in activation level as numerosity decreases. Higher activations are associated with smaller numerosities. This pattern of activations across the numerosities translates into a shallow, positive slope in the subitizing range because it is in this range that use of the recognition procedure dominates and thus latencies are determined by stored pattern activations.

Although SUBIT-R produced a bilinear latency profile consistent with the subitizing phenomenon, the actual latency values and slopes that were generated do not match empirical data well because we have not attempted to closely model the details of the enumeration processes. Probably the most notable shortcoming is that we did not obtain the relatively large slope in the counting range as compared to the subitizing range. A more plausible model of counting processes would include additional execution costs for counting operations and thus lead to a more consistent profile. The larger slope in the counting range would then be explained as resulting from an increasing number of required counting operations with increasing numerosity, whereas the relatively small slope in the subitizing range is explained by an average increase in matching time for a single recognition operation.

When faced with multiple alternative courses of action, the human cognitive architecture apparently favors the easiest and most efficient response mode (Anderson, 1990; Siegler, 1986). For enumeration of small collections of objects, the method of choice seems to be immediate recognition of the configuration. For larger collections, on the other hand, SUBIT-R shows how strengthening of a memory trace (required for recognition) becomes intractable due to the spacing effect of the patterns in the environment. Thus, we have shown that the dynamic deployment of some key functional characteristics of a theory of the human cognitive architecture (Anderson, 1993) is sufficient to produce key characteristics of the subitizing phenomenon without the need for a limited set of special purpose constructs such as FINSTs (Trick & Pylyshyn, 1993, 1994). Of course, an immediate question that must be raised concerns the generality of this account of subitizing. Is it highly dependent on the specific characteristics of ACT-R, or is it general enough that it would also emerge from rather different computational instantiations of the human cognitive architecture? This issue was examined in Experiment 2.

VI. EXPERIMENT 2

Parallel distributed processing (PDP) systems, sometimes referred to as neural networks, are becoming increasingly popular as models of various aspects of the human cognitive architecture. Compared to production system architectures such as ACT-R (Anderson, 1993; Anderson & Lebiere, 1998), these systems make radically different representation and processing assumptions in the way that they simulate human cognitive processing. Therefore, we reasoned that implementing the key aspects of the SUBIT model in such a system would provide a very strong test of our claim that the subitizing limit is an emergent property of the interaction between relevant aspects of the human cognitive architecture and the task domain. Some form of replication would greatly increase the generality of our findings, indicating that they are not dependent on the ACT-R system. Thus, in Experiment 2, we developed a parallel distributed processing model of enumeration, called SUBIT-PDP, and proceeded to train and test it on the same set of patterns used in Experiment 1.

VII. SUBIT-PDP: A PARALLEL DISTRIBUTED PROCESSING MODEL OF SUBITIZING

SUBIT-PDP was implemented as a feedforward, fully-connected, three-layer neural network. Learning function was provided by the general-purpose backpropagation learning algorithm (Rumelhart, Hinton, & Williams, 1986) which has been successfully used to model a wide variety of psychological phenomena (e.g., McClelland & Rumelhart, 1986; Zipser & Anderson, 1988). Learning in backpropagation networks occurs as follows. First, a data pattern is presented to the input layer of the network. The inputs are propagated through the network until the signals reach the units in the output layer. At this point, the network outputs are subtracted from the desired output values (provided as part of each training pattern) to produce an error signal. The error signal is then "backpropagated" through the network; that is, it is used as a basis for adjusting the connection weights in such a way that the computed outputs more closely match the desired outputs for the training pattern. The goal of backpropagation training is to converge on a set of connection weights that correctly classifies each of the input patterns. Furthermore, the nature of the learned mapping allows input patterns not occurring in the training data to also be correctly classified on the basis of their similarity to the patterns on which the network was trained. In other words, the network can generalize to novel input patterns.

There are necessarily some differences between SUBIT-PDP and the previously described SUBIT-R. Major differences are the following: 1) because it functions as a classifier, SUBIT-PDP can easily model the recognition procedure but not the more algorithmic counting procedure; and 2) SUBIT-PDP does not provide any direct measure of latency for enumeration of each pattern. Thus, it is difficult to compare performance of this model with performance of SUBIT-R on the basis of percentage usage of the recognition procedure (Figure 3) or enumeration latencies (Figure 6). However, at least a qualitative comparison to the SUBIT-R results can be made by examining activations of the network output units. In fact, we can obtain an activation profile similar to the one obtained in the previous experiment (Figure 5) by averaging the activation of the correct output unit across a set of test patterns for each numerosity. This activation profile should reveal the performance discontinuity between small and large numerosities that signals the subitizing phenomenon, should this characteristic emerge.

We attempted to closely follow the training and testing procedures used in the previous experiment. All patterns were randomly generated and each of the six numerosities was equally likely on every trial. As in the previous experiment, the combinatorics of this problem space ensured more frequent repetition of individual pattern instances for small numerosities than for large numerosities. In the neural network framework it is not so clear, however, that frequency of repetition should be directly reflected in the amounts of training required to correctly classify the numerosities. In fact, one basic tenet of neural network modeling holds that generalization is better achieved through a wide variety of different examples than through repetition of specific instances. This suggests that generalization may develop as quickly for large numerosities as for small, in which case a discontinuity in performance would not emerge in the activation profile. However, complexity and discriminability of the output categories are also likely to be major determinants of training requirements. Large numerosity patterns are certainly more complex and generally harder to distinguish from one another than small numerosity patterns. The increase in complexity and decrease in discriminability with size, although much more difficult to quantify, may in fact follow a combinatoric explosion similar to the one we have documented for the distribution of unique patterns in the problem space (shown in Figure 2). If this is the case, then a discontinuity in performance for small and large numerosities may also emerge for SUBIT-PDP.

VIII. METHOD

Patterns

Each input pattern was represented by a 17-element integer vector. The first 16 positions in this vector correspond to the positions on our hypothetical 4 × 4 grid of locations. Presence or absence of an object in each position is denoted by a 1 or 0, respectively, in the corresponding vector location. The final element of the vector is used to indicate the number of objects in the pattern (i.e., the desired output of the network). The vector shown below is an example of a training pattern for Numerosity 5 and constitutes one line in the training data file.

0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 5

So that each of the six numerosities could be associated with a particular unit in the network's output layer, a six-item one-of-N code was chosen to represent the desired output pattern. In this representation, each item in the code represents a distinct numerosity. A one appears in the position corresponding to the desired numerosity with zeros in all other positions. For the vector shown above, the integer five is encoded as

0 0 0 0 1 0

resulting in the following 22-element vector that gets presented to the network:

0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0

Two pattern data files were created, one for training the network and the other for testing. As in the previous experiment, each pattern was created by randomly selecting a

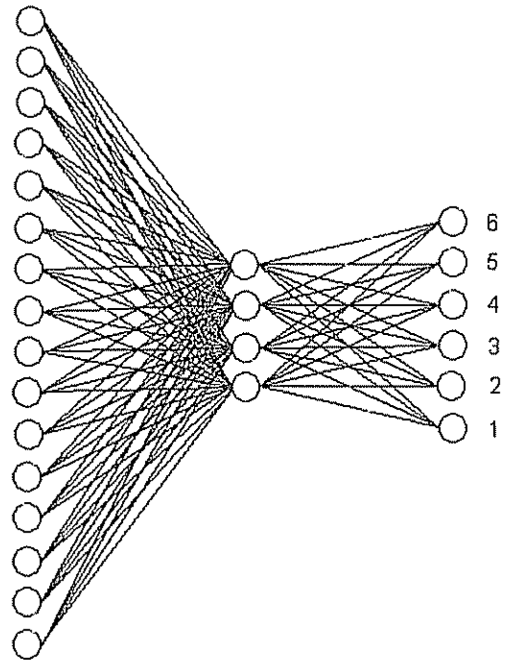

Figure 7 Neural network architecture for Experiment 2 This figure depicts a fully connected network with 16 input units, four hidden units, and six output units

numerosity and then randomly assigning objects to the 16 possible locations. The training data file contained 50,000 random patterns. The test data file contained 150 random patterns, 25 for each numerosity.

Network

We used the IBM Neural Network Utility software to create a backpropagation neural network consisting of three fully-connected layers with 16 units in the input layer, four units in the hidden layer, and six units in the output layer. Each of the 16 input units was associated with a location in our hypothetical grid. Each of the six output units was associated with one of the six possible numerosity categories. The network architecture is depicted in Figure 7. All connection weights were initialized to random values before the start of training.

For each input pattern, the network produces a set of activations (values ranging from 0 to 1.0) on the six output units. Because each output unit represents one of the six possible numerosities, correct categorization of the input pattern is indicated by high (close to 1.0) activation of the unit corresponding to the number of objects in the input pattern and low (close to 0) activations in the other five units. In other words, correct categorization should produce a set of activations that closely approximates the six-

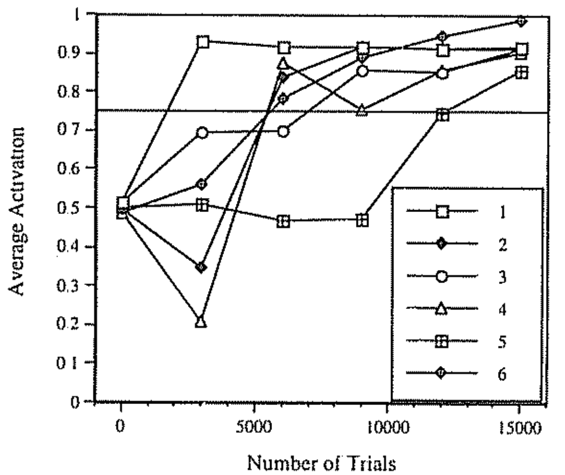

Figure 8 Average activation of correct output unit for network with 16 input units, four hidden units, and six outputs

element one-of-N code for the desired output. For the purpose of evaluating the performance of this model, we made the assumption that a pattern is correctly categorized by the network if the activation of the correct output unit exceeds a .75 threshold level. Incorrect classification can be assumed to indicate the use of a more effortful counting procedure, possibly of the type implemented in SUBIT-R.

Procedure

The network was trained by iterating through the training data file and presenting each training pattern to the network in sequence. After every 1000 training patterns, the network was temporarily "locked" to prevent any changes to the connection weights, and the performance of the network was evaluated on the set of 150 patterns in the test data set. After testing, the network was "unlocked" and the next set of training patterns was presented. This process continued until 50,000 training patterns had been processed. For each test pattern, the activation level (always in the range 0.0 to 1.0) of the correct output was recorded.

IX. RESULTS

Average activation of the correct output unit is plotted as a function of the number of training trials in Figure 8. Although 50,000 trials were performed, most of the learning occurred very early in training and thus only the data for the first 15,000 trials are shown here. No significant changes occurred after 15,000 trials. As the graph indicates, the network learned to recognize Numerosities 1 and 2 by about 5000 trials, and Numerosities 3 and 4 by about 10,000 trials. The profile for Numerosity 5 is very different, indicating very little learning within the first 10,000 trials. By 15,000 trials, the network as a whole seems to have achieved convergence, correctly classifying test patterns for all numerosities

The noticeable performance discontinuity between Numerosities 4 and 5 signals a capacity limitation of the network that can perhaps best be quantified by evaluating performance relative to threshold within the first 10,000 trials. Thus, we will assume the subitizing range to consist of those numerosities whose activation level has exceeded the .75 threshold by 10,000 trials. One potential problem with this criterion is that the network learned to recognize Numerosity 6 relatively early in training, in fact within the first 10,000 trials, well before Numerosity 5 was learned. This would seem to place Numerosity 6 within the subitizing range along with Numerosities 1 through 4. However, we interpret these data for Numerosity 6 as reflecting a sort of "end effect" (see discussion below) and therefore do not consider Numerosity 6 to fall within the subitizing range.

X. DISCUSSION

Results obtained for Numerosities 1 through 3 are very similar to the results obtained for Numerosities 1 through 3 in the SUBIT-R simulations. Both models were able to learn these small numerosities within the first 10,000 training trials. The data produced by SUBIT-PDP for Numerosity 4 exhibit some instability, again consistent with the SUBIT-R results, but by 12,000 trials seem to have settled into a stable profile. In the previous experiment, it was not so clear whether Numerosity 4 belonged within or outside the subitizing range because the model persisted in using a mixture of recognition and item-by-item counting even after large amounts of training. Here too, there seems to be some uncertainty for Numerosity 4, but in this case, a much more obvious distinction emerges between Numerosities 4 and 5. Applying our criterion of exceeding the .75 threshold by 10,000 trials we obtain a subitizing limit of four for SUBIT-PDP as compared to three for SUBIT-R. The discrepancy in subitizing spans found in Experiments 1 and 2 is interesting because it parallels results of empirical studies, some experiments producing a span of three (e.g., Chi & Klahr, 1975) and some producing a span of four (e.g., Frick, 1987) in the aggregated data. Similarly, when data are analyzed on an individual basis, some participants exhibit a span of three whereas others exhibit a span of four (e.g., Simon et al., 1993; Trick & Pylyshyn, 1993).

The finding that SUBIT-PDP learned to recognize Numerosity 6 early in training can be explained as an end effect for the largest numerosity. If this experiment were performed with human participants we might expect many of them to begin to guess the highest possible numerosity (six in this case) for displays containing "a lot" of items, resulting in reduced latencies for that numerosity. Indeed, exactly this sort of effect has been documented by a number of enumeration researchers (e.g., Simon, Peterson, Patel, & Sathian, 1998; van Oeffelen & Vos, 1982; Wolters, van Kempen, & Wijlhuizen, 1987). SUBIT-PDP apparently learns to recognize this numerosity more quickly because of the reduced discrimination requirements. Numerosity 5 must be distinguished both from Numerosity 4 and Numerosity 6 whereas Numerosity 6 need only be distinguished from

Numerosity 5. Thus, the network suggests a potential explanation for human behavior: Participants learn patterns for the highest numerosity faster than other numerosities in the counting range because of their relatively high discriminability. High discriminability might also underlie the empirical finding that human participants can readily learn to recognize familiar (canonical) patterns such as those appearing on the faces of a die (Mandler & Shebo, 1982).

Although clearly evident in Figure 8, the discontinuity in performance between the small and large numerosity ranges is certainly not as sharp as that observed in the performance of SUBIT-R. One potential reason is the lack of a decay mechanism in the modeling framework used here. In SUBIT-R, the decay mechanism had greater effect on the larger numerosities due to their infrequent occurrence, tending to force activations of those patterns below the retrieval threshold. It is not clear exactly how to implement this sort of decay mechanism in a neural network model like SUBIT-PDP, but its presence could operate to sharpen the observed discontinuity.

Another factor that seems to be moderating the observed effect is the generalization capability of the network, which results from the distributed nature of the numerosity representations that develop during training. This is evident in the sizable reduction in the amount of training required to recognize the larger numerosities. For example, Numerosity 5 patterns are recognized at a high confidence level after 15,000 trials in SUBIT-PDP whereas the recognition rate was still extremely low for SUBIT-R after 25,000 trials. To demonstrate that generalization is indeed occurring in SUBIT-PDP, a simple test was performed. The network was reset (weights randomized) and then SUBIT-PDP was retrained on the first 20,000 patterns in the training data file. At this point, the network was locked and a collection of 25 novel patterns for Numerosity 5 was presented. The activation of the correct output unit exceeded the .75 threshold for each pattern, indicating that SUBIT-PDP was able to recognize every one of the 25 novel patterns even though none had appeared during training. This is clear evidence that SUBIT-PDP has developed an abstract, distributed representation of Numerosity 5 and no doubt similar representations for the other numerosities as well.

Because it relies on memory of specific instances rather than distributed numerosity representations, it is likely that SUBIT-R overestimates the amount of training required for recognition to occur. In contrast, SUBIT-PDP probably underestimates the training requirements for larger numerosities because it does not account for the decay in activation resulting from relatively infrequent experience with these numerosities. Notwithstanding these issues, the similarity in the results obtained using two very different cognitive architectures is striking. The subitizing limit emerged at about the same point in both models, suggesting that the explanation lies not in a specific instance of a simulated cognitive architecture but in the general case of the human cognitive architecture's attempt to deal with the structure in the environment.

However, one question does remain over the interpretation of the findings from Experiment 2. It should be noted that the subitizing limit that was found (four) was the same as the number of hidden units in the neural network. Could this in fact be evidence in favor of the structural limitation view? Perhaps there is some correspondence between hidden units, the representational mediators of a network, and FINSTs, the purported limiting factor on object individuation. The next experiment set out to investigate that possibility.

XI. EXPERIMENT 2A

Experiment 2a was designed to test the sensitivity of our SUBIT-PDP results to the number of units in the neural network hidden layer. One possible explanation for the observed subitizing limit of four is that it somehow results from using four units in the hidden layer. If only three hidden units were used, would the subitizing limit be reduced to three? To examine the relationship of subitizing limit and number of hidden units, we repeated the experiment with networks containing two, three, and five units in the hidden layer.

XII. METHOD

Patterns

The same training and test data files used in Experiment 2 were used in the present experiment.

Network

Three additional neural networks were created, containing two, three, and five hidden units. Number of inputs (16) and outputs (six) remained the same as in the previous experiment. Connection weights in all three networks were initialized to random values.

Procedure

Each of the three networks was trained and tested in the same manner as described above for Experiment 2.

XIII. RESULTS AND DISCUSSION

In general, learning capability of the model increased with the number of hidden units. However, we did not find a direct relationship between the number of hidden units and the emergent subitizing limit. Results obtained using the two-hidden unit network (not shown) indicated very little learning of the numerosities and no clear emergence of a subitizing limit, even with extensive training. These results are probably due to insufficient representational capacity in the network's hidden layer. The two-hidden unit network learned to recognize Numerosities 1 and 3 by 10,000 trials but was not able to learn the patterns for Numerosity 2, or the patterns for any of the larger numerosities, even by the end of the simulation run.

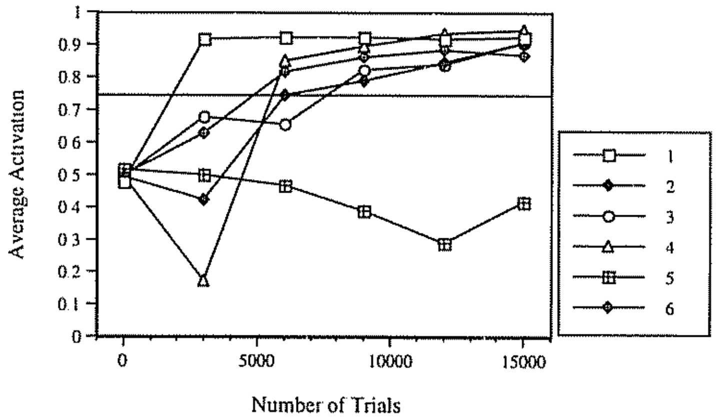

Figure 9 Average activation of correct output unit for network with 16 input units, three hidden units, and six outputs

Results for the three-hidden unit network, shown in Figure 9, are very similar to the results obtained with the four-hidden unit network described in the previous experiment. The network learned to recognize Numerosities 1 through 4 and 6 within about the first 10,000 trials whereas the profile for Numerosity 5 indicated no learning within this range. Interestingly, even after 50,000 trials, the three-hidden unit network had not learned to recognize Numerosity 5 patterns. Thus, although the subitizing limit for this network was also determined to be four (using the same criterion described earlier), performance data indicate that the network has somewhat less representational power than the four-hidden unit network.

Results for the five-hidden unit network, shown in Figure 10, indicate that learning occurred more rapidly in this network for all numerosities, seemingly due to the increased representational capacity. Activations exceeded the threshold for Numerosities 1 through 3 and 6 by 5000 trials and for Numerosities 4 and 5 by 10,000 trials. These data are consistent with a subitizing limit of either five or six, depending on whether the rapid learning of Numerosity 6 patterns is considered to be due to an end effect.

This pattern of results, particularly those obtained for the three- and four-hidden unit networks, does not seem to support the hypothesis that the subitizing limit is determined simply by the number of hidden units in the network. Our results do, however, point out the importance of representational capacity for the emergence of this limit. With insufficient capacity (two hidden units), very little convergence occurred and the model produced an indeterminate subitizing limit. On the other hand, excess capacity (five hidden units) seemed to facilitate learning to the point that all numerosities were learned relatively quickly, again resulting in a subitizing limit whose exact value was not easy to discern. It was only at moderately-constrained capacity levels (three and four hidden units) that the apparent true nature of model was revealed, indicating a rather clear subitizing limit of four objects in both cases.

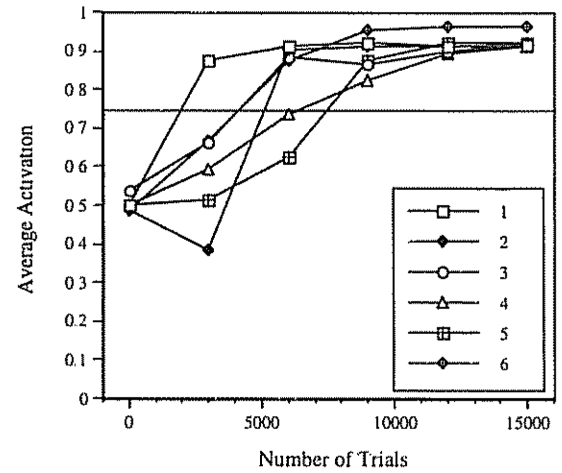

Figure 10 Average activation of correct output unit for network with 16 input units, five hidden units, and six outputs

One additional question that emerges here is whether this "optimal" level of representational capacity (about four hidden units) is related in any way to the rather constrained combinatoric space in which we have been working. If the size of the combinatoric space were increased, would we need a corresponding increase in representational capacity to produce the same results, or would learning just take longer given a larger space to explore? The following experiment was designed to investigate this issue.

XIV. EXPERIMENT 3

In Experiment 3 we tested the scalability of our representational claim to a larger problem space. The size of the input vector was extended from 16 to 36 elements (representing a 6 × 6 grid) and the number of possible objects increased from six to eight. The number of hidden units in the network was set to four. Of primary interest was whether the same qualitative behavior pattern would emerge during training, that is, recognition of small numerosities early in training and large numerosities much later, or not at all. If so, this would show that our explanation of the subitizing limit is not fragile with respect to the combinatoric space from which the enumeration patterns are selected. Also of interest was whether an end effect would occur as it did in the previous simulation. In this case, the end effect should be associated with Numerosity 8.

XV. METHOD

Patterns

Each input pattern was represented by a 37-element integer vector. The first 36 positions in this vector correspond to the positions in our hypothetical 6 × 6 grid. Presence or absence of an object in each position was denoted by a one or zero, respectively, in the corresponding vector location. The final element of the vector was used to indicate the number of objects in the pattern (i.e., the desired output of the network). An eight-item one-of-N code was used to represent the desired output pattern so that each of the eight numerosities could be associated with a particular unit in the network's output layer.

Again, two pattern data files were created, one for training the network and the other for testing. As in the previous experiments, each pattern was created by randomly selecting a numerosity and then randomly assigning objects to the 36 possible locations. The training data file contained 50,000 random patterns and the test data file contained 200 random patterns, 25 for each numerosity.

Network

The backpropagation neural network for this experiment consisted of three fully-connected layers with 36 units in the input layer, four units in the hidden layer, and eight units in the output layer. Each of the 36 input units was associated with a location in our hypothetical grid of locations. Each of the eight output units was associated with one of the 8 possible numerosity categories. All connection weights were initialized to random values before the start of training. As in the previous experiments, each input pattern resulted in a set of activations on the eight output units, correct categorization being denoted by high activation of the unit corresponding to the number of objects in the input pattern and low activations in the other seven units.

Procedure

Training and testing proceeded in the same fashion as described for Experiment 2. The training data file was opened and patterns were sequentially presented to the network. After every 1000 trials, the network was locked (preventing connection weight adjustments) and tested by cycling through the 200 patterns in the test data file. Again, the activation level of the correct output unit was recorded for each training pattern.

XVI. RESULTS

Simulation results for this network are shown in Figure 11a (N = 1–4) and 11b (N = 5–8). For each numerosity, average activation of the correct output unit is plotted as a function of the number of training trials. Note that these graphs include data for the entire simulation run (50,000 trials). These data again indicate relatively rapid learning of Numerosities 1 through 4. Activation levels for Numerosities 1 through 3 exceed the .75 threshold by 10,000 trials and for Numerosity 4 by about 15,000 trials. In contrast, the profiles for the larger numerosities indicate much different learning processes. Activation levels for Numerosities 5, 6, and 8 generally remain below threshold until very late in the simulation run, indicating that the network had much greater difficulty in learning these patterns. Note that learning would be even slower for these large numerosities if operated

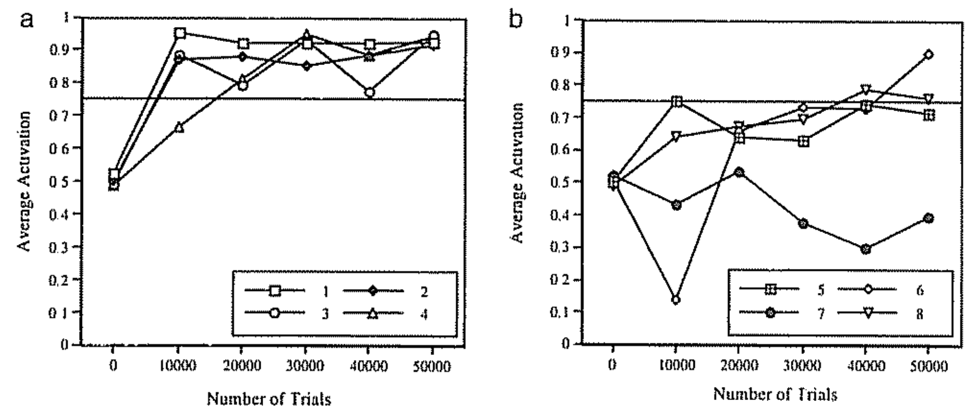

Figure 11. Average activation of correct output unit for network with 36 input units, four hidden units, and eight outputs. Numerosities 1 through 4 are shown in Figure 11a and Numerosities 5 through 8 are shown in Figure 11b.

on by some form of activation decay over time as in SUBIT-R. The profile for Numerosity 7 indicates no learning for this numerosity within the first 50,000 trials. Using the criterion for subitizing described earlier (activation above .75 by 10,000 trials), we obtain a subitizing limit of three for the present experiment. Interestingly, if we were to relax our criteria by allowing a slightly longer training cutoff (e.g., 15,000 trials), then Numerosity 4 might also be included in the subitizing range, as it was in the previous experiment.

XVII. DISCUSSION

The large (6 × 6) network used here produced roughly the same behavior as the smaller one (4 × 4) in Experiment 2. SUBIT-PDP learned to recognize Numerosities 1 through 3 within 10,000 trials and required only slightly longer to learn the Numerosity 4 patterns. This replication of the previous results is quite surprising given the large increase in complexity of the problem space (see below). The present findings are also very similar to the SUBIT-R simulation results obtained in Experiment 1. Those data indicated that small (N = 1–3) but not large numerosities (N = 5–6) were recognized by the end of the simulation run (25,000 trials) whereas Numerosity 4 patterns were sometimes recognized and sometimes not. Thus, the present results suggest that our previous findings (Experiments 1, 2, and 2a) do indeed scale well to larger combinatoric spaces. Emergence of a clear subitizing limit did not require an increase in representational capacity, nor did it necessarily require an increase in training time. We see no reason that similar results would not be obtained for even larger grid sizes and numbers of objects.

Probably the most notable difference in performance for this version of SUBIT-PDP as compared to the previous (i.e., the one for the 4 × 4 problem space) is the large increase in amount of training required to achieve asymptotic performance levels for the larger numerosities (Figure 11b). Much of this can be attributed to the increased complexity of

TABLE 1
Distribution of Unique Patterns by Numerosity for the 4×4 and 6×6 Problem Spaces

N	4×4 grid	6×6 grid
1	16	36
2	120	630
3	560	7140
4	1820	58,905
5	4368	376,992
6	8008	1,947,792
7		8,347,680
8		30,260,340
Total	14,892	40,999,515

the 6×6 problem space used here. One indicator of this complexity is the number of different patterns to be learned. Considering just Numerosities 1 through 6, the total number of unique patterns on a 4×4 grid is just under 15,000 (see Figure 2) whereas the number of unique patterns on a 6×6 grid is well over 2 million! When Numerosities 7 and 8 are included, the total number of patterns that can be represented in this problem space rises to nearly 41 million! Table 1 provides a complete account of the combinatoric increase by numerosity for each of the two grid sizes used in this study. Given such a dramatic increase in complexity, it is quite amazing that our Experiment 3 data show any evidence of learning at all, especially for the larger numerosities where only a very small subset of the possible patterns were actually presented to the network. Learning in this version of SUBIT-PDP clearly depends on the network's ability to develop abstract representations and then generalize to novel patterns. A model such as SUBIT-R, which does not have the ability to generalize but rather depends on the repetition of specific instances, would likely require a much longer training period to achieve similar levels of performance.

Interestingly, the end effect (for Numerosity 8) was not as strong in this version of SUBIT-PDP, probably due to the increased number of large numerosity patterns in the training data. Apparently it was much easier to discriminate Numerosity 6 from Numerosities 1 through 5 (Experiment 2) than it was to discriminate Numerosity 8 from Numerosities 1 through 7 (Experiment 3). This finding suggests that emergence of an observable end effect in empirical data are not a certainty, but rather depends on the characteristics of the entire problem set being used, and most likely other factors as well such as previous enumeration experience and retrieval thresholds of individual participants.

XVIII. EXPERIMENT 3A

Again we decided to explore the sensitivity of our findings to the number of hidden units, or representational capacity, in the neural network. In this case we repeated Experiment 3 using a five-hidden unit network as opposed to the four-hidden unit network used in the original experiment. As discussed earlier, results obtained with a five-hidden unit network in Experiment 2a were unclear. In that experiment, we found a subitizing limit of either five or six, which could be viewed as consistent with the idea that the subitizing limit is determined by the number of hidden units. Yet we argued that those results were due to excess representational capacity in the network, which allowed patterns for all numerosities to be learned relatively quickly. Thus, excess capacity effectively eliminated the subitizing phenomenon. For the much more complex 6×6 problem space, it seems unlikely that five hidden units will provide sufficient capacity to eliminate emergence of the subitizing limit altogether, and thus we expect some sort of limit to emerge. If a subitizing limit of five is obtained here, our claim that the subitizing limit is not determined by the number of hidden units would be seriously challenged. Alternatively, a limit of three or four would provide additional evidence favoring the generality of our representational claim.

XIX. METHOD

Patterns

The same training and test data files used in Experiment 3 were also used here.

Network

One additional neural network was created, containing five hidden units. Number of inputs (36) and outputs (eight) remained the same as in the previous experiment. Connection weights were initialized to random values.

Procedure

The network was trained and tested in the same manner as described for the previous experiment.

XX. RESULTS AND DISCUSSION

Data obtained for this simulation run are shown in Figure 12a (N = 1–4) and Figure 12b (N = 5–8). Results for Numerosities 1 through 3 are almost identical to those shown in Figure 11a for the four-hidden unit network. These numerosities are again learned within the first 10,000 trials with very stable performance from that point onward. The network also learns to recognize Numerosity 4 by 15,000 trials, but stable performance for this numerosity is not achieved until about 40,000 trials. In comparing Figures 11b and 12b, it can be seen that activation profiles for Numerosities 5, 6, and 8 reach higher asymptotic levels than in the previous experiment, most likely a result of the increased representational power of the network. However, there is still no evidence of learning for Numerosity 7. Applying our subitizing criterion to these data, we again obtain a subitizing limit

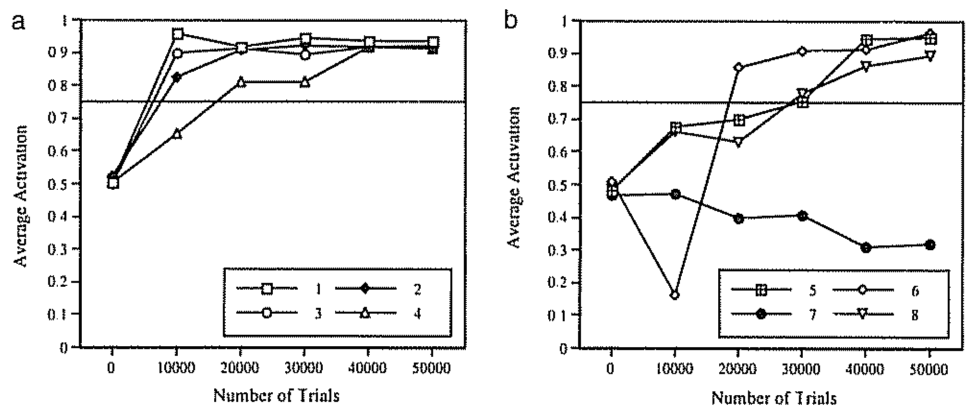

Figure 12 Average activation of correct output unit for network with 36 input units, five hidden units, and eight outputs Numerosities 1 through 4 are shown in Figure 12a and Numerosities 5 through 8 are shown in Figure 12b

of 3. The end effect for Numerosity 8 is slightly stronger, probably also related to the increase in representational power.

In Experiment 2 and 2a, we obtained a subitizing limit of four for networks containing three- and four-hidden units. In Experiments 3 and 3a, using the same criterion, we obtained a subitizing limit of three (or possibly four with relaxed criteria) for networks with four- and five-hidden units. Taken together, these results provide strong evidence that the subitizing limits produced by SUBIT-PDP are not a simple reflection of the number of hidden units in the underlying neural network.

XXI. GENERAL DISCUSSION

These experiments provide converging evidence that the subitizing limit is an emergent property of the human cognitive architecture and its interaction with the structured domain of enumeration problems. Without any prespecified structural limit such as that proposed by Trick and Pylyshyn (1994), we repeatedly showed that the space of numerically-varying patterns was partitioned into two regions. One contained numerosities up to three or four, whereas the other contained the larger sets. This behavior was first demonstrated in Experiment 1, where SUBIT-R was presented with randomly-generated patterns based on a hypothetical 4 × 4 grid of locations. Although each numerosity was enumerated roughly the same number of times, there was much greater repetition of small numerosity patterns than large numerosity patterns due to the combinatorics of the problem space. We demonstrated how the numerosity distribution of those patterns, when operated on by a simple set of enumeration procedures, interacted with the learning mechanisms in the ACT-R system to produce the subitizing phenomenon.

In Experiment 2, we demonstrated that this finding is not entirely dependent on the particular theoretical assumptions built in to the ACT-R cognitive architecture by also generating the effect within the PDP framework. When trained on a similar series of enumeration problems, SUBIT-PDP also exhibited an emergent limited capacity of numerosity recognition consistent with the subitizing phenomenon. It is worth noting, however, that SUBIT-PDP, at least as currently implemented, lacks certain features that seem to be important for a full account of the subitizing phenomenon. For example, it does not provide a mechanism, such as that provided by the ACT-R architecture, for the decay of pattern activations over time. Neither does it incorporate a counting procedure which, as in SUBIT-R, involves as a component operation the recognition of small numerosity subpatterns, thus further increasing activations for small numerosities. Nonetheless, the subitizing limit clearly emerged in the SUBIT-PDP simulations, suggesting that pattern discriminability also plays a critical role. As numerosity increases, unique patterns become increasingly difficult to discriminate from one another (see van Oeffelen & Vos, 1982 for a similar conclusion), resulting in greater difficulty in learning the correct response.

In Experiment 3, we demonstrated that the effect was not dependent on our choice of grid size (4 × 4) or number of possible objects (six). In this experiment, we adopted a larger problem space based on a 6 × 6 grid of locations with 8 possible objects in each pattern. Despite an enormous increase in complexity of the problem space resulting from this change, SUBIT-PDP again exhibited an emergent limited recognition capacity consistent with the subitizing limit. Surprisingly, the emergence of this limit did not require any additional representational capacity in the neural network hidden layer, nor did it seem to be associated with additional training time.

The manipulations of the number of hidden units in Experiments 2a and 3a showed that increased representational capacity facilitates network convergence, and that the emergent limits in our data were not simply determined by the number of hidden units. These experiments also pointed out the possible importance of amount of representational capacity for the emergence of the subitizing limit. For example, in Experiments 2 and 2a, we found clear evidence of a subitizing limit of four when the network contained either three- or four-hidden units but less evidence of a limit when the number of hidden units was two (insufficient capacity) or five (excess capacity). This is consistent with the general notion that the subitizing phenomenon is in some way a function of a limited representational capacity (e.g., Trick & Pylyshyn, 1993, 1994). However, it is not the case that a fixed representational capacity determines a given subitizing limit. Rather it is the dynamic interaction of this limited representational capacity with the combinatorics of stimulus distribution in the environment that determines the limit. Indeed, a similar subitizing limit was generated in Experiment 1 by SUBIT-R, a model whose architecture contains no a priori limit on representational capacity. In general, our results seem to indicate that there is something optimal about the ability of the human perceptual system to process small collections of three or four items. Consistent with this notion is the finding that automatic capture of visual attention by multiple abrupt onsets also has an upper limit of about four (Yantis, 1996).

An interesting aspect of our SUBIT-R and SUBIT-PDP models is that they both seem to provide plausible, yet different, explanations for the type of representation that might support the subitizing phenomenon. SUBIT-R utilizes an instance-based representation (cf., Logan, 1988) in which the activation level of each instance reflects its frequency of occurrence. The subitizing limit emerges as activation levels for small numerosity instances become much greater than those for larger numerosities. In contrast, SUBIT-PDP relies on distributed patterns of activation across a neural network. In each of the SUBIT-PDP simulations, what developed over time was a limited capacity, numerosity recognition network. The subitizing limit apparently emerges in this type of network when it is trained to the point where abstract representations for small but not large numerosity patterns have developed (e.g., 10,000 trials). Thus, small numerosity patterns can be easily classified (or recognized) whereas large numerosity patterns cannot.

The type of distributed numerosity representation employed by SUBIT-PDP may be able to explain the findings of subitizing-like behavior in infants, who apparently can discriminate small numerosities (two dots from three dots) but not larger ones (four dots from six dots) (Antell & Keating, 1983; Starkey & Cooper, 1980). All that is required to explain these findings is the assumption that abstract representations for small numerosities can develop rather quickly, possibly even over the course of the habituation trials in an experiment, whereas representations for larger numerosities require significantly more time to develop, if they develop at all.

In fact, it would be straightforward to replicate the results of numerical discrimination experiments such as these using a neural network model like SUBIT-PDP. For example, suppose a network is trained exclusively on patterns for Numerosity 2 to the point where it can generalize to all patterns containing two objects. Then, if tested with patterns containing both two and three objects, the network would effectively discriminate two objects from three objects because it can reliably recognize Numerosity 2 but not Numerosity 3. The same number of training patterns would likely not be sufficient to develop abstract representations of larger numerosities such as four or six and thus the network would not be able to discriminate these larger numerosities from one another. With sufficiently large amounts of training, however, we would expect the network to be able to differentiate such large numerosity patterns. In most of our simulations, abstract numerosity representations did indeed emerge for these numerosities with sufficient training.

The experiments presented here show that the subitizing phenomenon can be explained without assuming any particular limit on the representational capacity of human cognition. Previous models of the phenomenon have generated the standard profile by incorporating a predefined subitizing limit. Our account demonstrates instead how the interaction of cognitive processes with the structure of the enumeration problem space is sufficient to produce a learning and performance profile in which small numerosity patterns come to be immediately recognized whereas large numerosity patterns must continue to be counted. Moreover, differing activation levels of the patterns stored in memory translate into a small increase in recognition latencies from one numerosity to the next, thereby offering a possible explanation of the subitizing slope. Thus, the subitizing phenomenon, as characterized by a limit of 4 ± 1 objects and a shallow reaction time slope, seems to be an emergent property of human cognitive processing and its interaction with the distributional properties of the enumeration domain.

Acknowledgments: We would like to thank Kimberly Morton and Noel Rappin for their contributions to earlier versions of the SUBIT-R model. We would also like to acknowledge Christian Lebiere and the rest of the ACT-R group at Carnegie Mellon University for their ongoing support of this project. Details of the SUBIT-R model can be obtained through the ACT-R Web site (http://act.psy.cmu.edu/ACT/ftp/contributions). Finally, we are grateful to John Anderson and Graeme Halford for helpful comments on an earlier version of this manuscript.

NOTES

1 SUBIT-R is actually our second version of this model. The first version was described in Peterson, Morton, and Simon (1997).

2 This counting procedure is not intended as a detailed model of the information processing that takes place during such enumeration. Instead it is basically an implementation of the *counting principles* described by Gelman and Gallistel (1978), which seem to be understood by most three- and four-year olds. The counting principles are: 1) *one-to-one principle*: one and only one number should be assigned to each object, 2) *stable order principle*: numbers should always be assigned in the same order, 3) *cardinal principle*: the last number assigned indicates the total number of objects, 4) *abstraction principle*: these principles apply to any set of objects, and 5) *order irrelevance principle*: objects can be counted in any order.

3 The operation of ACT-R's base level learning mechanism is governed by two parameters, base level learning (bll) rate and retrieval threshold (rt). For all SUBIT-R simulations reported in this study, we used the settings bll = .05 and rt = 0.

4 The choice of a 4 × 4 grid size was not theoretically motivated, but rather reflects current limitations in computing resources.

REFERENCES

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum.
Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum.
Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.
Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher-level cognition and its relation to visual attention. *Human Computer Interaction, 12*, 439–462.
Antell, S. E., & Keating, D. P. (1983). Perception of numerical invariance in neonates. *Child Development, 54*, 695–701.
Chi, M., & Klahr, D. (1975). Span and rate of apprehension in children & adults. *Journal of Experimental Child Psychology, 19*, 434–439.
Frick, R. W. (1987). The homogeneity effect in counting. *Perception & Psychophysics, 41*, 8–16.
Gelman, R., & Gallistel, C. R. (1978). *The child's understanding of number*. Cambridge, MA: Harvard University Press.
Jevons, W. S. (1871). The power of numerical discrimination. *Nature, 3*, 44–64.
Kaufman, E., Lord, M., Reese, T., & Volkmann, J. (1949). The discrimination of visual number. *American Journal of Psychology, 62*, 498–525.
Klahr, D., & Wallace, J. G. (1976). *Cognitive Development: An Information Processing View*. Hillsdale, NJ: Lawrence Erlbaum.
Logan, G.D. (1988). Toward an instance theory of automatization. *Psychological Review, 95*, 492–527.

Mandler, G., & Shebo, B (1982) Subitizing: An analysis of its component processes *Journal of Experimental Psychology General, 111*, 1–22
McClelland, J L, & Rumelhart, D E (1986) *Parallel distributed processing: explorations in the microstructure of cognition* (Vol 2) Cambridge, MA: MIT Press
Newell, A (1990) *Unified theories of cognition* Cambridge, MA: Harvard University Press
Peterson, S, Morton, K, & Simon, T J (1997) Modeling a functional explanation of the subitizing limit In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* Hillsdale, NJ: Lawrence Erlbaum
Rumelhart, D E, Hinton, G E, & Williams, R J (1986) Learning internal representations by error propagation In D E Rumelhart, & J L McClelland (Eds.), *Parallel distributed processing explorations in the microstructure of cognition* (Vol 1, pp 318–362) Cambridge, MA: MIT Press
Shrager, J, Klahr, D, & Chase, W (1982) Segmentation and quantification of random dot patterns Paper presented at the 23rd annual meeting of the Psychonomics Society
Siegler, R S (1986) Unities in strategy choices across domains In M Perlmutter (Ed), *Minnesota symposium on child development* (Vol 19) Hillsdale, NJ: Lawrence Erlbaum
Siegler, R S, & Robinson, M (1982) The development of numerical understandings In H W Reese, & L P Lipsitt (Eds.), *Advances in child development and behavior* New York: Academic Press
Siegler, R S, & Shrager, J (1984) Strategy choices in addition and subtraction: How do children know what to do? In C Sophian (Ed), *Origins of cognitive skills* Hillsdale, NJ: Lawrence Erlbaum
Siegler, R S, & Shipley, C (1995) Variation, selection, and cognitive change In T J Simon & G S Halford (Eds), *Developing cognitive competence: New approaches to process modeling* Hillsdale, NJ: Lawrence Erlbaum
Simon, T, Cabrera, A, & Kliegl, R (1993) A new approach to the study of subitizing as distinct enumeration processing *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society* Hillsdale, NJ: Lawrence Erlbaum
Simon, T J, Peterson, S, Patel, G, & Sathian, K (1998) Do the magnocellular and parvocellular visual pathways contribute differentially to subitizing and counting? *Perception & Psychophysics, 60*, 451–464
Starkey, P, & Cooper, R S (1980) Perception of numbers by human infants *Science, 210*, 1033–1035
Svenson, O, & Sjoberg, K (1983) Speeds of subitizing and counting processes in different age groups *Journal of Genetic Psychology, 142*, 203–211
Trick, L, & Pylyshyn, Z (1993) What enumeration studies can show us about spatial attention: Evidence for limited capacity preattentive processing *Journal of Experimental Psychology Human Perception and Performance, 19*, 331–351
Trick, L, & Pylyshyn, Z (1994) Why are small and large numbers enumerated differently? A limited-capacity preattentive stage in vision *Psychological Review, 101*, 80–102
van Oeffelen, M P, & Vos, P G (1982) Configural effects on the enumeration of dots: Counting by groups *Memory and Cognition, 10*, 396–404
Wolters, G, van Kempen, H, & Wijlhuizen, G (1987) Quantification of small numbers of dots: Subitizing or pattern recognition? *American Journal of Psychology, 100*, 225–237
Yantis, S (1996) Attentional capture in vision In A F Kramer, M G H Coles, & G D Logan (Eds), *Converging operations in the study of visual selective attention* Washington, DC: American Psychological Association
Zipser, D, & Anderson, R A (1988) A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons *Nature, 331*, 679–684

APPENDIX

Numerosity	Amount recognized 1	2	3	4	5	6
1000 trials						
1	100					
2	56	44				
3	12	84	4			
4	4	84	12	0		
5	0	52	44	4	0	
6	0	24	68	12	0	0
2000 trials						
1	100					
2	8	92				
3	0	88	12			
4	0	44	56	0		
5	0	12	80	8	0	
6	0	0	72	24	4	0
5000 trials						
1	100					
2	0	100				
3	0	52	48			
4	0	12	84	4		
5	0	0	56	40	4	
6	0	0	32	52	16	0
10,000 trials						
1	100					
2	0	100				
3	0	20	80			
4	0	0	72	28		
5	0	0	28	68	4	
6	0	0	0	92	8	0
15,000 trials						
1	100					
2	0	100				
3	0	0	100			
4	0	0	60	40		
5	0	0	12	80	8	
6	0	0	0	80	20	0
20,000 trials						
1	100					
2	0	100				
3	0	4	96			
4	0	0	48	52		
5	0	0	0	92	8	
6	0	0	0	72	28	0
25,000 trials						
1	100					
2	0	100				
3	0	0	100			
4	0	0	32	68		
5	0	0	0	80	20	
6	0	0	0	40	52	8

Rate of full recognition and subpattern recognition for each simulation run Each cell represents percentage of trials for a given numerosity (row) in which a pattern or subpattern of a particular size (column) was recognized