CHAPTER

6

Development of Counting Ability: An Evolutionary Computation Point of View

*Gali Barabash Katz**, *Amit Benbassat***, *Moshe Sipper***

**Department of Cognitive Sciences, Ben-Gurion University of the Negev, Beer-Sheva, Israel; **Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

OUTLINE

Continuous Issues in Numerical Cognition. http://dx.doi.org/10.1016/B978-0-12-801637-4.00006-8

6.1 INTRODUCTION

Size perception (estimating an area of a blob without counting) is shared by humans (Brannon, Lutz, & Cordes, 2006) and animals (Cantlon & Brannon, 2007). Counting (to report the exact number of items in an array), on the other hand, is a more complex ability, which is currently known to be specific to humans. Although these two systems are different, current literature finds interesting overlaps, which might shed light on the development of counting ability through evolution.

First, there is early evidence of shared functional information between different systems through evolution in the work of Piaget (1955) and Flavell (1963), where they suggested that cognitive structures are content independent and domain general. Kashtan and Alon (2005) demonstrated a similar idea by simulating evolution of artificial neural networks (ANNs) in an environment with modularly varying goals, and showed that the final networks had more reusable building blocks than the control networks that evolved in an environment with a single fixed goal.

When it comes to numerical abilities, it is commonly suggested that the basic numerical intuitions that are shared between humans and animals are supported by an evolutionarily ancient approximate number system (ANS) where the number of discrete objects are represented as a continuous mental magnitude (Cantlon, Platt, & Brannon, 2009; Dehaene, 2001). This core system enables the representation of the approximate number of items in visual or auditory arrays without verbally counting (Halberda, Mazzocco, & Feigenson, 2008) and might be the root for high-level human numerical abilities such as arithmetic (Dehaene, 2001).

Second, early imaging studies that were conducted in order to test the hypothesis of a common substrate for processing symbolic and nonsymbolic stimuli revealed a site in the left intraparietal sulcus (IPS) that showed greater activation during numerical compared to nonnumerical judgments (Fias, Lammertyn, Reynvoet, Dupont, & Orban, 2003).

Subsequent studies compared stimuli for size, luminance, or number and found that the right hIPS (horizontal segment of the IPS) is not devoted exclusively to number processing but is engaged whenever subjects attend to the dimension of size—whether numerical or physical (Pinel, Piazza, Le Bihan, & Dehaene, 2004). Moreover, in one of the studies (Cohen Kadosh et al., 2005), a largely overlapping network of frontal, parietal, and occipitotemporal areas of both hemispheres was found, thus confirming the view that many of the neural resources used for number comparison are shared by other comparison tasks as well.

Today, it seems that the left IPS mainly responds to numerical stimuli in a format-specific manner—specifically, to symbolic stimuli. The right IPS, on the other hand, is activated regardless of stimuli format and is also activated by nonnumerical magnitudes. These findings led researchers to suggest that the right IPS might support a general magnitude system, used to process both numerical and nonnumerical magnitudes, rather than an abstract approximate number system (Chapter 15).

Third, there is also evidence that nonsymbolic number and cumulative area representation contribute shared and unique variance to mathematical competence (Lourenco, Bonny, Ferandez, & Rao, 2012). College students were asked to estimate which array was greater in number or cumulative area, after which they completed a battery of standardized math tests. The authors found that individual differences in both number and cumulative area precision were correlated with interindividual differences in arithmetic and geometry. Moreover, whereas number precision contributed unique variance to advanced arithmetic, cumulative area precision contributed unique variance to geometry. Based on their results, Lourenco et al. (2012) suggested that uniquely human branches of mathematics interface with an evolutionarily primitive general magnitude system, which includes partially overlapping representations of numerical and nonnumerical magnitude.

When thinking about the evolutionary approach, we can identify similar patterns between primate approximate counting and human exact counting. Recently, Cantlon, Piantadosi, Ferrigno, Hughes, and Barnard (2015) reported that nonhuman primates exhibit a cognitive ability that is algorithmically and logically similar to human counting. In their experiment, monkeys were given the task of choosing between two food caches. First, they saw one cache baited with peanuts, one at a time. Then, the second cache was baited, one peanut at a time. At the point when the second cache was approximately equal to the first one, the monkeys spontaneously moved to choose the second cache even before that cache was completely baited. By using a novel Bayesian analysis, the authors showed that monkeys use an approximate counting algorithm for comparing quantities in sequence that is incremental, iterative, and condition controlled, similar to formal counting in humans.

Developmental dyscalculia [DD; a severe difficulty in learning and making simple mathematical calculations (Kosc, 1974), with an estimated prevalence of about 5–7% (Shalev, 2007)] might also serve as a link between the ANS and symbolic arithmetic. Today, there is growing evidence that the relationship between numerical distance and IPS activity is disrupted in children with developmental dyscalculia (Fias et al., 2003), and many known effects are compromised in DD (eg, size congruity effect: Rubinsten & Henik, 2005, 2006; distance effect: Price, Holloway, Räsänen, Vesterinen, & Ansari, 2007). The cause of the deficit might be a network failure between continuous magnitude processing (ie, the size perception system) and discrete magnitude processing (ie, the counting system). Children as well as adults who suffer from these specific learning disabilities are at a disadvantage in both academic and everyday life situations, especially when it comes to handling money (Henik, Rubinsten, & Ashkenazi, 2014). Thus, a better evolutionary understanding of the development of counting in relation to size perception can help researchers think of novel ways to improve day-to-day life of people with arithmetic disabilities.

Currently, the suggested computerized models explaining how the counting system functions use artificial neural networks along with different unsupervised learning techniques (ie, teaching a predefined network to successfully perform a required task such as counting). Verguts and Fias (2004) used an unsupervised learning technique for teaching ANNs to process nonsymbolic and symbolic inputs. First, they presented the ANNs a nonsymbolic input (eg, a collection of dots) in a comparison task and showed the distance and size effects. Then, they presented symbolic and nonsymbolic inputs simultaneously and found that the ANNs used the number-selective neurons that were already available, when processing symbolic stimuli. Their findings present a possible linkage between higher-order numerical cognition and more primitive numerical abilities.

Stoianov and Zorzi (2012) presented binary images (sets of black shapes on a white background) to ANNs in a comparison task (ie, "Which is larger?") against a given reference number, and after an unsupervised learning process, the ANNs excelled in the task. The given stimuli differed by the number of objects presented and their cumulative area (the authors controlled both properties). They suggested a "deep network model" (which developed through unsupervised learning without using evolutionary techniques), containing one visual input layer and two hidden layers, in order to explain the numerosity estimation process. According to their model, there are two types of neurons in hidden layer 1: neurons that deal with visual processing (center surround neurons) and neurons that deal with cumulative area, and hidden layer 2 neurons deal with numerosity processing. In addition, hidden layer 2 neurons' activity can be approximated by a linear combination of the activity of both types of hidden layer 1 neurons. The hierarchical model generated the same results

in a numerosity comparison task as did human adults, thus can serve as the key to understanding the neural mechanism underlying numerosity perception.

Based on the researches described earlier, we consider the following two potential hypotheses regarding the development of counting. One is that the counting system evolved on the basis of a primitive system designed to perceive size and evaluate amount of substance (Cantlon et al., 2009; Henik, Leibovich, Naparstek, Diesendruck, & Rubinsten, 2012; Lourenco et al., 2012). The other is that both systems evolved separately, in different periods of time.

In addition, as previously mentioned, exact counting is a complex task. Kaufman, Lord, Reese, and Volkmann (1949) suggested that the entire enumeration process is carried out first by subitizing, (ie, the phenomenon of giving a rapid, confident and accurate report of the amount of up to four presented items) and then by counting (from five items and above). Thus, subitizing might be a stepping-stone in the development of the exact counting system—a theory we examine in the current research.

Herein we present a novel approach of computerized simulation with ANNs for examining the development of counting ability, in a process known as evolutionary computation.

6.2 EVOLUTIONARY COMPUTATION

Evolutionary computation (EC) is a subfield in computer science that is inspired by biology. An evolutionary algorithm (EA) enables solving complex problems by using an evolutionary process metaphor.

An evolutionary process usually includes:

- An environment with limited resources, which is translated to the problem one wants to solve.
- Individuals, which are translated into candidate solutions.
- A concept of fitness (ie, the compatibility of the individual to survive and reproduce in the environment), which is translated to a probability of generating new solutions.

In nature, the competition for limited resources causes a selection of those who fit better to the environment (ie, natural selection), and as a result, the fitness of the population improves over time. The same happens in EAs, and as in nature, the fit individuals create a new generation (ie, new candidate solutions) by using parameters of recombination and mutation (see Box 6.1 for an example of a typical EA).

Genetic algorithms (GAs, Goldberg, 1989) are a simple variant of evolutionary algorithms that use a simple alphabet (eg, a binary alphabet “0,” “1”) and can solve complex problems with a simple model representation.

BOX 6.1

EXAMPLE OF A TYPICAL EA:

Begin

1. *Initialize* the population with random candidate solutions;
2. *Evaluate* each candidate;
3. Repeat until (*termination condition* is satisfied)
   - **3.1** *Select* parents
   - **3.2** *Recombine* pairs of parents
   - **3.3** *Mutate* the resulting offspring
   - **3.4** *Evaluate* new candidates
   - **3.5** *Select* individuals for the next generation

End

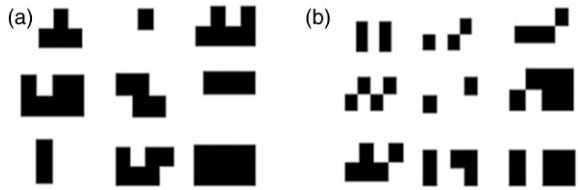


FIGURE 6.1 **Examples of continuous and discrete 2 × 4 binary arrays.** (a) Continuous stimuli and (b) discrete stimuli. We define a stimulus as continuous if there is a path from each visible cell in the array to every other visible cell that passes through visible cells (in single up/down/left/right steps; Katz et al., 2013).

It should be noted that even though GAs use binary discrete inputs, they can represent size, which is continuous, as can be seen in Fig. 6.1a.

Evolutionary algorithms are stochastic, meaning that even for identical parameter setups, results may vary, but they usually share the same trend. Genetic operators (eg, mutation and recombination) can lead to new, unexpected, and creative solutions to complex problems.

6.3 CURRENT STUDY (OR HOW CAN EVOLUTIONARY ALGORITHMS HELP IN UNDERSTANDING THE DEVELOPMENT OF THE COUNTING SYSTEM)?

In this study we computationally examined whether individuals that evolved a proficiency in size perception (ie, evolved to solve a classification problem of BIG vs. SMALL; Table 6.1) have an advantage in learning

TABLE 6.1 Size Perception, Counting, and Control Groups' Genetic Algorithms With Their Fitness Functions and Outputs

#	Evolution	Fitness and output
1	SP	A given input is classified as BIG if its number of ones ≥ array. length/2, and otherwise as SMALL. The same logic is applied to the output. BIG is an output with a number of ones that is ≥ [array.length/2]. SMALL is an output of a number of ones that is < [array.length/2].
2	C	For a given input, the exact number of ones given is expected to be in the output (without order considerations).
3	SP-C	The size perception fitness function is switched to the counting fitness function mid-run after 25 generations (the entire size perception run took an average of 48.5 generations).
4	C1	This is a random classification task. The set of inputs was divided randomly into two groups: one group expected a BIG output and the other a SMALL one, similar to SP (see table item 1).
5	C1-C	The control 1 fitness function switched to the counting fitness function after 25 generations. We expected an output of the exact number of ones, similar to C (see table item 3).

Note: SP, evolved to perceive size; C, evolved to count; SP-C, evolved first to perceive size and then to count; C1, evolved to excel in control 1 task; and C1-C, evolved first to excel in control 1 task and then to count (Katz et al., 2013).

to count. In order to test our hypothesis, we chose genetic algorithms to develop artificial neural networks through an evolutionary process. We used this technique to evolve ANNs that could excel in a size perception task and then further evolve the same networks to count. Our main goal was to examine whether these ANNs have an advantage in evolving the ability to count over new learners of counting (ie, ANNs that did not first evolve to perceive size).

6.4 NEUROEVOLUTION OF AUGMENTING TOPOLOGIES (NEAT)

NEAT is a method for evolving artificial neural networks using genetic algorithms, developed by Stanley and Miikkulainen (2002). It simulates evolution by starting with small, simple networks that become increasingly complex through evolution. Just as organisms in nature increased in their complexity through evolution, so do neural networks in NEAT. This process of continual elaboration allows finding highly sophisticated and complex neural networks. NEAT evolves both the connection weights and architecture (by adding and removing connections and nodes) of the

neural networks. The networks start with minimal topologies (this is analogous to natural evolution that begins from simple forms) and gradually become more complex (Stanley & Miikkulainen, 2002).

We chose to use NEAT[a] in order to evolve ANNs to perceive size and to count. We analyzed the complexities of the final networks created by evolution by examining the number of inner nodes added to the networks as the task became more complex.

6.5 METHODS

6.5.1 Stimuli

In order to represent the visual input, we used a 2 by 4 two-dimensional binary array. A total of 256 (2^8) possible stimuli could be produced by this array; some of these are discrete and some are continuous. We defined a stimulus as continuous if there was a path from each visible cell in the array to every other visible cell that passed through visible cells (in single up/down/left/right steps). According to this definition, of the 256 possible stimuli, 147 are discrete and 109 are continuous (Katz, Benbassat, Diesendruck, Sipper, & Henik, 2013; Fig. 6.1).

We divided the stimuli into three sets of stimuli:

1. The "continuous" set.
2. The "discrete" set.
3. Both continuous and discrete, which was called the "all" set.

Sets (1) and (2) were composed of 108 stimuli randomly divided into training and test sets of 54 stimuli[b] each (no repetitions). Set (3) was composed of a training set of 128 (256/2) stimuli and a test set, which included the remaining 128 stimuli.

Prior to being inserted into the NEAT system, the stimulus was flattened into a one-dimensional array of 0s and 1s (Fig. 6.2).

All the evolutionary simulations that we performed included a comparison to a reference number. In each trial the given blob (ie, binary string) was compared to the half size of the string array (in all three simulations it was $8/2 = 4$ for a Boolean array of 2×4), meaning: "Is the number of 'ones' in the current blob larger than 4?"

[a]Specifically, the NEAT4J Java implementation.

[b]This value was chosen because there were 147 discrete and 109 continuous stimuli out of 256, and we wanted to keep an equal number of stimuli in the training and test groups; thus, we chose the minimal group (i.e., 109) and divided it into two equal groups of 54 stimuli (one for training and one for testing).

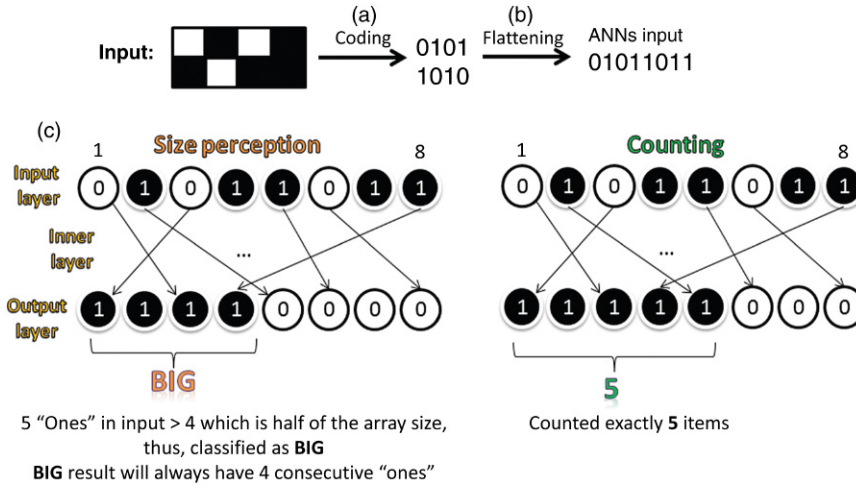


FIGURE 6.2 **Example of encoding a stimulus into a binary string.** (a) Encoding a single discrete stimulus of 2×4 to a binary representation. (b) Flattening the two-dimensional array into a one-dimensional array. (c) A demonstration of how an ANN can perform tasks of size perception and counting on the same input (Katz et al., 2013).

6.5.2 Procedure

Several different types of evolutionary procedures were tested during the study. Every *evolutionary run* was performed 30 times using each of the 3 stimuli sets (*continuous*, *discrete*, and *all*), resulting in a total of 90 runs per type. The procedure was similar for all run types. Every run began with a training stage, where the population was trained to succeed in a certain task (ie, classify BIG/SMALL blobs or counting the exact number of squares of the given blob) until its average fitness score (ie, the score of the function that should be optimized in order to continue to the next generation in evolution; see Fig. 6.3 for fitness calculation formula) exceeded

$$\text{Fitness} = \sum_{i=1}^{\text{Inputs}} \text{Score}, \quad \text{Max} = \text{Output} \cdot \text{Length}$$

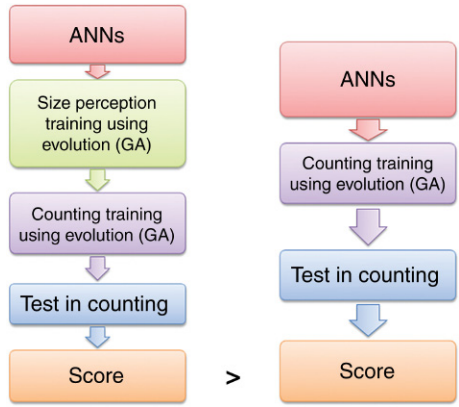
$$\text{Score} = \frac{\text{Max} - \sum_{i=0}^{\text{Max}-1} |\text{Expected}_{\text{Sorted}}[i] - \text{Observed}_{\text{Sorted}}[i]|}{\text{Max}}$$

FIGURE 6.3 **Fitness calculation.** The *fitness* is the summation of all scores of the inputs (ie, the stimuli presented to the current individual). The *score* is calculated according to the distance of each digit from the expected array to the observed array (after sorting both arrays).

10%, and more specifically, the value 0.999.[c] If this condition was not met, the algorithm halted after 3000 generations. Next, an evaluation of the population took place. During this test stage, each individual was evaluated in relation to each of the tasks relevant to the fitness functions used; an additional test on counting accuracy was performed in all runs. For example, the "size perception" population trained to perceive size (ie, classify BIG/SMALL blobs) was tested on size perception (ie, again asked to classify BIG vs. SMALL blobs) but also tested on counting (ie, asked to count the number of squares of the given blob, see Box 6.2 for the procedure schema).

BOX 6.2

FROM SIZE PERCEPTION TO COUNTING SIMULATION SCHEMA



By using evolutionary computation techniques, we generated artificial neural networks that excelled in size perception and presented a significant advantage in evolving the ability to count over those that evolved this ability from scratch.

[c]The value 0.999 in size perception and in counting, and 0.998 in subitizing, as this was found to be sufficient in order to excel in this task in the testing stage.

6.5.3 Genetic Algorithm Parameters

We used the same evolutionary parameters in all the simulations that will be described here:

- Population size of 100 individuals
- Termination condition: fitness of 0.999 or 0.998^c or 3000 generations if previous condition was not met
- Mutation probability $P_m = 0.25$ (ie, when creating a new generation of ANNs, this is the probability of adding/removing connections and nodes in the new ANN offspring)
- Crossover probability $P_c = 1$ (ie, the probability that the new ANN offspring will end up having half the genes from one parent and half from the other)

6.5.4 Calculation of Fitness Function

The *fitness* was the summation of all scores of the inputs (ie, the stimuli presented to the current individual). The *score* was calculated by the summation of the differences between the expected array index and the observed array index, and it was normalized by *max*, which is the output length—in our case, 8 (Katz et al., 2013).

Each evolutionary run defined the expected output it got from NEAT. When a result from NEAT was received, the fitness function checked if the expected output was equal to the observed one. If so, it assigned a 100% score, otherwise it calculated a score according to the distance of each digit from the expected array to the observed one (after sorting both arrays) as follows:

6.6 SIMULATIONS

The following two Simulations #1 and #2 were first presented briefly in Katz et al. (2013) and are being described here with great detail in order to set the context for the *new* Simulation #3.

6.6.1 Simulation 1: From Size Perception to Counting (Katz et al., 2013)

We created five evolutionary runs in order to test our hypothesis. The goal was to train a set of ANNs in a certain task and then switch to a different task by changing the fitness function mid-run. The algorithm performed the switch from one fitness function to the other after a predefined number of generations, which was chosen after several trial-and-error

runs. We opted for this approach instead of waiting for the networks to excel in the tasks, in order to avoid the "overfitting" phenomenon in the consecutive runs (ie, bloated networks with too many inner nodes that excelled specifically on the training inputs rather than solving the more general problem). Thus, we switched from the size classification tasks to the counting task after 25 generations (Table 6.1).

6.6.2 Results

We conducted 3 different two-way ANOVA (analysis of variance) tests, for 3 different dependent variables: counting score, number of generations, and number of inner nodes in an ANN.

The 3 (stimuli type) × 5 (evolutionary runs) design was between subjects, since each evolutionary run produced a different population of ANNs. The following 5 evolutionary runs were performed:

1. Size perception (SP)—the ANNs evolved to perceive size.
2. Counting (C)—the ANNs evolved to count.
3. Size perception and then counting (SP-C)—the ANNs evolved first to perceive size and then to count.
4. Control group (C1)—the ANNs evolved to excel in a control 1 task (a classification task; Table 6.1).
5. Control and then counting (C1-C)—the ANNs evolved first to excel in the control 1 task and then to count.

Each of the 5 evolutionary runs (see previous list) ran 3 times—each time with a new stimuli type:

1. Continuous stimuli only
2. Discrete stimuli only
3. Both continuous and discrete stimuli

In the following analyses, we present the results that are relevant to our theoretical considerations, with significance level of $p < 0.05$. As previously mentioned, these results were first reported in Katz et al. (2013) and are being explained here with great detail to set the context for the new results of Simulation 3).

6.6.2.1 Counting Score (Score of the Final Counting Test)

The counting score (based on the *fitness calculation*, see Fig. 6.3) was higher in networks that first evolved to perceive size or to perform other classification tasks than networks that evolved to count independently (eg, *SP-C vs. C*: $F(1, 435) = 58.49$, $MSE = 0.0062$, $p < 0.01$, $\eta_p^2 = 0.118$, and also *C1-C vs. C*: $F(1, 435) = 385.49$, $MSE = 0.0062$, $p < 0.01$, $\eta_p^2 = 0.47$).

In addition, when the tasks were easy (ie, the networks evolved to excel in a single task, eg, SP, C1, and not two tasks as in SP-C or C1-C), the score for discrete stimuli was higher than for continuous stimuli [eg, *continuous*

vs. discrete in SP: F (1, 435) = 84.47, MSE = 0.0062, $p < 0.01$, $\eta_p^2 = 0.162$, and also *continuous vs. discrete in C1*: F (1, 435) = 215.89, MSE = 0.0062, $p < 0.01$, $\eta_p^2 = 0.33$]. However, when the tasks became more complex, the opposite pattern was observed and the ANNs with continuous stimuli had better scores at counting [*continuous vs. discrete in C1-C*: F (1, 435) = 6.146, MSE = 0.0062, $p < 0.05$, $\eta_p^2 = 0.0139$; in *C* and *SP-C* the differences were not significant, see Fig. 6.4].

6.6.2.2 Generations

SP and *C1* runs were faster (ie, had a smaller number of generations) than all other tasks (*SP*: M = 48.5, SD = 9.21; *C1*: M = 44.9, SD = 5.15; *C*: M = 308.22, SD = 4.32; *SP-C*: M = 330.7, SD = 16.33; *C1-C*: M = 334.7, SD = 24.33). More generations were required to evolve the networks to count after evolving to perform any other task [*C vs. C1-C*: F (1, 435) = 183.49, MSE = 172.51, $p < 0.01$, $\eta_p^2 = 0.296$].

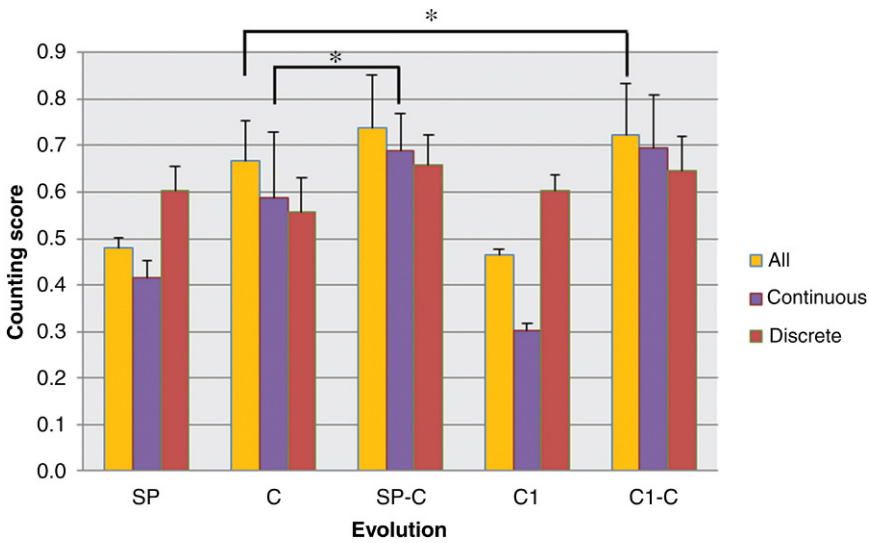


FIGURE 6.4 **Counting scores results of Simulation 1—from size perception to counting.** The counting score of the ANNs that first evolved to perceive size and then to count (SP-C) is higher than the counting score of ANNs that evolved to count (C) from scratch (Table 6.1; Katz et al., 2013). *C1*, control group with a random classification task; *yellow*, ANNs that evolved and were tested with both types of stimuli (continuous and discrete); *purple*, ANNs that evolved and were tested with continuous stimuli; and *red*, ANNs that evolved and were tested with discrete stimuli. In all evolutionary runs, the actual stimuli were different between the training (by evolution) and testing sets. The "*" represents significant differences: the counting score when evolving first to perceive size and then to count is significantly higher than the counting score received when evolving to count from scratch, with continuous stimuli; and the counting score when evolving to count after first learning a random classification task is significantly higher than the score when evolving to count from scratch, with both stimuli types.

6.6.2.3 *Inner Nodes (A Measurement for Network Complexity)*

Complex runs (ie, runs that had a complex task to learn such as counting) generated ANNs with more inner nodes than the simpler ones (complex runs: *C*: $M = 11.81$, $SD = 4.71$; *SP-C*: $M = 12.39$, $SD = 4.6$; and *C1-C*: $M = 12.05$, $SD = 5.04$ and simpler runs: *SP*: $M = 3.64$, $SD = 1.64$; *C1*: $M = 3.96$, $SD = 1.37$).

In addition, during the complex runs the ANNs that *evolved and were tested* on discrete stimuli contained *more inner nodes* than the ones evolved and tested with continuous stimuli [*discrete vs. continuous in C1-C*: F (1, 435) = 25.25, $MSE = 13.87$, $p < 0.01$, $\eta_p^2 = 0.05$, and *discrete vs. continuous in SP-C*: F (1, 435) = 5.55, $MSE = 13.87$, $p < 0.05$, $\eta_p^2 = 0.002$, but *discrete vs. continuous in C* was not significant].

6.6.3 Simulation 2: Continuous Versus Discrete (Katz et al., 2013)

In the evolutionary runs discussed earlier, we trained and tested ANNs on a certain stimuli type (eg, evolved with continuous and tested with continuous), the results of which seemed to imply that continuous stimuli were more suitable to the task of evolving counting ability. In order to examine if this was so, or if the high score for continuous stimuli in the counting test was just due to continuous stimuli being less complex to process than discrete stimuli, we proceeded to train on continuous stimuli and test on discrete stimuli and vice versa.

6.6.4 Results

We conducted an ANOVA with an array of 2 (evolutionary runs) × 4 (stimuli type). The following 2 evolutionary runs were performed:

1. Size perception (SP)—the ANNs evolved to perceive size.
2. Counting (C)—the ANNs evolved to count.

Both evolutionary runs had a final test in Counting. Each evolutionary run was performed 4 times, each time with a new stimuli type:

1. Evolved with continuous stimuli and tested with discrete stimuli
2. Evolved with discrete stimuli and tested with continuous stimuli
3. Evolved with continuous stimuli and tested with continuous stimuli
4. Evolved with discrete stimuli and tested with discrete stimuli

Similar to previous analyses, we present the results relevant to our theory, with a significance level of $p < 0.05$. As previously mentioned, these results were first reported in Katz et al. (2013) and are being explained here with great detail to set the context for the new results of Simulation 3.

6.6.4.1 Counting Score

The interaction between stimuli and evolutionary run was significant, F (3, 232) = 15.16, MSE = 0.0084, $p < 0.01$, $\eta_p^2 = 0.16$. In both SP and C runs, the counting score when *evolved with continuous but tested with discrete* was *higher* than when evolved and tested with continuous stimuli, F (1, 232) = 26.6, MSE = 0.0084, $p < 0.01$, $\eta_p^2 = 0.09$).

Different results were observed with discrete stimuli[d]: When *evolved with discrete and tested with continuous,* the counting score was *lower* than the counting score of the control groups (ie, evolving and testing with discrete stimuli, F (1, 232) = 108.87, MSE = 0.0084, $p < 0.01$, $\eta_p^2 = 0.32$). In addition, the ANNs that *evolved to count with discrete* stimuli were *worse* in counting than those that *evolved to perceive size with discrete stimuli* (but were tested in counting): F (1, 232) = 4.14, MSE = 0.0084, $p < 0.05$, $\eta_p^2 = 0.017$. Another interesting result was that in SP runs, when *evolving ANNs with continuous stimuli and testing with discrete stimuli*, the counting score was *higher* than the other way around (ie, evolving ANNs with discrete stimuli and testing with continuous stimuli): *SP*: F (1, 232) = 41.09, MSE = 0.0084, $p < 0.01$, $\eta_p^2 = 0.15$, and *counting*: F (1, 232) = 70.97, MSE = 0.0084, $p < 0.01$, $\eta_p^2 = 0.23$.

Finally, the results when evolving with discrete stimuli and testing with continuous stimuli were the worst among all other combinations in both size perception and counting runs, F (1, 232) = 37.89, MSE = 0.0084, $p < 0.01$, $\eta_p^2 = 0.14$, and the counting score in this case was the same for size perception and counting, F (1, 232) = 3.65, MSE = 0.0084, $p = ns$, $\eta_p^2 = 0.02$ (Fig. 6.5).

6.6.4.2 Generations

Training ANNs on continuous stimuli and testing on discrete ones required the same number of generations as training ANNs on discrete stimuli and testing on continuous ones. In addition, this number of generations was equal to the number of generations of the control groups (for all four SP runs it took about 46 generations: M = 46.13, SD = 7.41, and for all four counting runs it took about 308 generations: M = 308.45, SD = 4.09).

6.6.4.3 Inner Nodes

The interaction between stimuli and evolutionary run was significant, F (3, 232) = 5.321, MSE = 13.32, $p < 0.01$, $\eta_p^2 = 0.064$. In the *counting* runs, more inner nodes were produced when the ANNs evolved with discrete stimuli (M = 15.66, SD = 5.99) than when evolved with continuous stimuli (M = 11.46, SD = 3.702, F (1, 232) = 10.25, MSE = 13.32, $p < 0.05$, $\eta_p^2 = 0.04$). Moreover, the number of inner nodes in *counting* runs in ANNs that evolved with discrete stimuli and tested with continuous stimuli was

[d]There might be a different explanation for the result differences between discrete and continuous stimuli because there were no discrete 0s, 1s, 7s, or 8s, and a minority of the 6s were discrete, which might have caused a bias toward the subitizing range numbers.

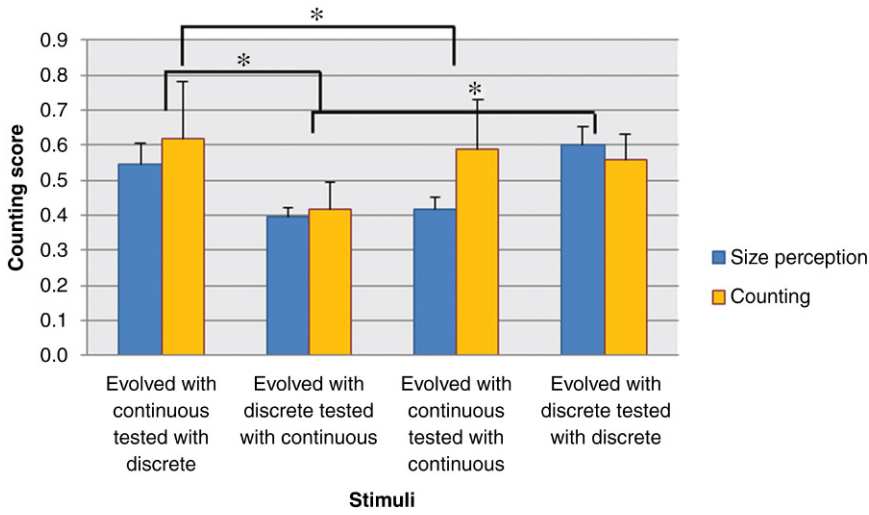


FIGURE 6.5 **Counting scores results of Simulation 2—continuous versus discrete.** The ANNs that were evolved to count with continuous stimuli and were tested with discrete stimuli presented better counting skills than the other group, which evolved to count with discrete stimuli but was tested with continuous stimuli. The "*" represents significant differences: the counting score in both SP and C tasks when evolved with continuous and tested with discrete is significantly higher than the counting score received when evolved on discrete and tested on continuous, and is also significantly higher than the score of the control group (ie, evolved and tested on continuous stimuli); but when evolving on discrete stimuli and tested on continuous, the counting score is significantly lower than counting score of the control group (ie, evolved on discrete and tested on discrete stimuli). (Katz et al., 2013).

higher than all other stimuli combinations, $F(1, 232) = 4.21$, $MSE = 13.32$, $p < 0.05$, $\eta_p^2 = 0.02$. In *SP* runs, the number of inner nodes stayed the same for all stimuli combinations ($M = 3.43$, $SD = 1.72$).

6.6.5 Simulation 3a: Adding a Subitizing Task

In order to improve the counting score (that stood at approximately 60% accuracy in the counting tests in the previous simulations), six additional evolutionary run types containing subitizing tasks were created (Table 6.2).

6.6.6 Results

An ANOVA with an array of 3 (stimuli types) × 6 (evolutionary runs) was conducted. The following 6 evolutionary runs were performed:

1. Size perception (SP)—the ANNs evolved to perceive size.
2. Counting (C)—the ANNs evolved to count.
3. Subitizing (Sub)—the ANNs evolved to subitize.
4. SP-Sub—the ANNs evolved first to perceive size and then to subitize.

TABLE 6.2 Subitizing-Related Genetic Algorithms and Their Fitness Function

#	Evolution	Fitness
1	Sub	For a given input with 1–4 ones, the expected output contains the exact same number of ones. For inputs with $N > 4$ ones, an output of $N - 1$ or $N + 1$ ones is also acceptable.
2	SP-Sub	The size perception fitness function switched to the subitizing fitness function mid-run after 25 generations.
3	Sub-C	The subitizing fitness function switched to the counting fitness function mid-run after 200 generations (number of generations before switch chosen experimentally).
4	C	For a given input, the exact number of ones given is expected to be in the output (without order considerations).
5	SP-Sub-C	The size perception fitness function switched to the subitizing fitness function after 25 generations, and after 200 additional generations there was another switch to the counting fitness function.
6	C1-Sub-C	The control 1 (random classification task—see Table 6.1 item 4) fitness function switched to the subitizing fitness function after 25 generations, and after 200 additional generations there was another switch to the counting fitness function.

Note: SP, evolved to perceive size; C, evolved to count; Sub, evolved to excel in the subitizing task; SP-Sub, evolved first to perceive size and then to excel in subitizing; SP-Sub-C, evolved first to perceive size, then evolved to excel in subitizing and finally evolved to count; and C1-Sub-C, evolved first to excel in control 1 task, then evolved to excel in subitizing and finally evolved to count.

5. SP-Sub-C—the ANNs evolved first to perceive size, then evolved to subitize, and finally evolved to count.
6. C1-Sub-C—the ANNs evolved first to excel in the control 1 task, then evolved to subitize and finally evolved to count.

Each of the 6 evolutionary runs ran 3 times—each time with a new stimuli type:

1. Continuous stimuli only
2. Discrete stimuli only
3. Both continuous and discrete stimuli

6.6.6.1 Counting Score

The interaction between stimuli and evolution was significant, F (10, 522) = 2.68, MSE = 0.01, $p < 0.05$, $\eta_p^2 = 0.05$. All the combinations of *subitizing with counting* led to *better counting scores* than counting independently, $F(1, 522) = 162.54$, $MSE = 0.01$, $p < 0.01$, $\eta_p^2 = 0.24$. Most interesting were the following significant comparisons: *SP-Sub-C vs. C*: F (1, 522) = 106.04, MSE = 0.01, $p < 0.01$, $\eta_p^2 = 0.16$, and *C1-Sub-C vs. C*: F (1, 522) = 108.62, MSE = 0.01, $p < 0.01$, $\eta_p^2 = 0.17$, which is the classification control group.

In the majority of the runs, ANNs that evolved with continuous stimuli attained better counting scores than ANNs that evolved with discrete stimuli, F (1, 522) = 39.75, MSE = 0.01, $p < 0.01$, $\eta_p^2 = 0.07$ (Fig. 6.6).

6.6.6.2 Generations

A significantly *higher number of generations* were required to achieve peak performance in the counting task with a subitizing stage than without it [ie, *Sub-C* (M = 518.4, SD = 36.44) *vs. C* (M = 308.22, SD = 4.32)].

6.6.6.3 Inner Nodes

Discrete stimuli ANNs were composed of more inner nodes than continuous stimuli ANNs, meaning that the networks specialized in discrete stimuli were larger and more complex than the ones specialized in continuous stimuli, F (1, 522) = 52.88, MSE = 30.098, $p < 0.01$, $\eta_p^2 = 0.09$.

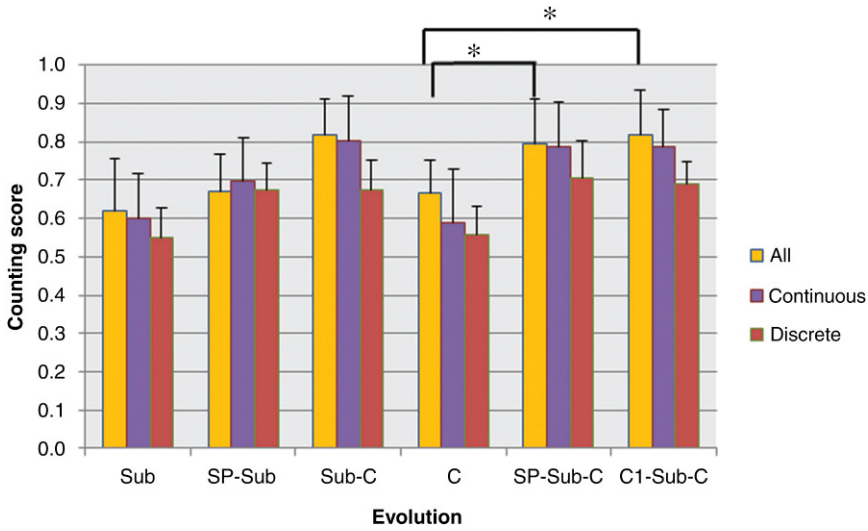


FIGURE 6.6 Counting scores result of Simulation 3a—adding a subitizing task. All the combinations of *subitizing with counting* led to better counting scores than counting independently, and in the majority of the runs, ANNs that evolved with continuous stimuli attained better counting scores than when evolved with discrete stimuli. *Sub*, subitizing; *SP*, size perception; *C*, counting; *C1*, control group with random classification task; *yellow*, ANNs that evolved and were tested with both types of stimuli (continuous and discrete); *purple*, ANNs that evolved and were tested with continuous stimuli; and *red*, ANNs that evolved and were tested with discrete stimuli. In all evolutionary runs, the actual stimuli were different between the training (by evolution) and testing sets. The "*" represents significant differences: the counting score in both SP-Sub-C and C1-Sub-C runs were significantly higher than the counting score recived in the C runs.

6.6.7 Simulation 3b: Continuous Versus Discrete With Subitizing

As in Simulation 2, in order to make sure that the high score for continuous stimuli in the counting test was not just due to continuous stimuli being less complex to process than discrete stimuli were, we performed another set of evolutionary runs, now with a subitizing task, and evolved the ANNs on continuous stimuli and tested them on discrete stimuli and vice versa.

6.6.8 Results

6.6.8.1 Counting Score

An ANOVA with an array of 4 (stimuli types) × 4 (evolutionary runs) was conducted. The following 4 evolutionary runs were performed:

1. Size perception (SP)—the ANNs evolved to perceive size.
2. Subitizing (Sub)—the ANNs evolved to subitize.
3. Counting (C)—the ANNs evolved to count.
4. SP-Sub-C—the ANNs evolved first to perceive size, then evolved to subitize, and finally evolved to count.

Each evolutionary run was performed 4 times, each time with a new stimuli type, as follows:

1. Evolved with *continuous* stimuli and tested with *discrete* stimuli
2. Evolved with *discrete* stimuli and tested with *continuous* stimuli
3. Evolved with *continuous* stimuli and tested with *continuous* stimuli
4. Evolved with *discrete* stimuli and tested with *discrete* stimuli

As in Simulation 2, we noted *significantly higher* counting scores when evolving ANNs with continuous stimuli and testing with discrete stimuli than the other way around, F (1, 464) = 341.68, MSE = 0.009, $p < 0.01$, $\eta_p^2 = 0.42$. In addition, the ANNs that evolved with continuous stimuli but tested with discrete stimuli were *better* than their control group (ie, that were evolved and tested on continuous stimuli), F (1, 464) = 11.29, MSE = 0.009, $p < 0.01$, $\eta_p^2 = 0.03$.

When evolved with discrete stimuli, the ANNs in the control group (that evolved and were tested with discrete stimuli) were *better in counting* than those who evolved with discrete stimuli and were tested with continuous stimuli, F (1, 464) = 244.31, MSE = 0.009, $p < 0.01$, $\eta_p^2 = 0.34$.

Finally, when comparing the ANNs that evolved with continuous stimuli (regardless of the type of stimuli in the counting tests) to the ones that evolved with discrete stimuli, we can see that SP-Sub-C was *significantly better in counting* than all other evolutionary runs, F (1, 464) = 72.25, MSE = 0.009, $p < 0.01$, $\eta_p^2 = 0.13$ (Fig. 6.7).

Note that the difference in counting score between the groups that evolved with continuous stimuli in the SP-Sub-C run was not significant,

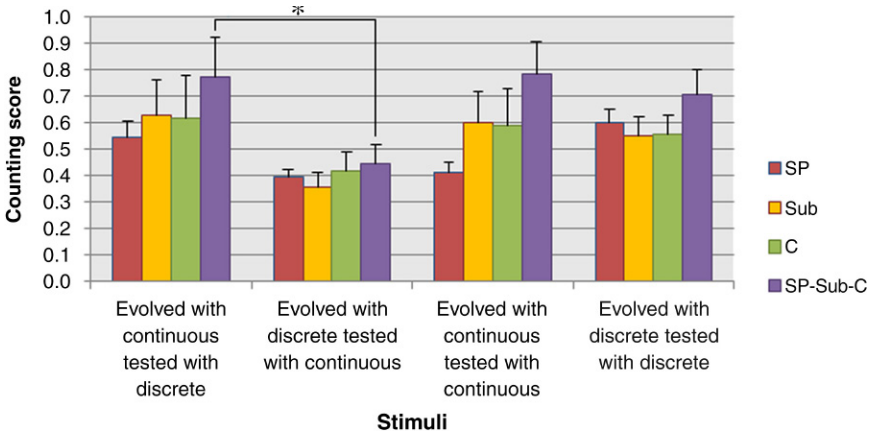


FIGURE 6.7 Counting scores of Simulation 3b—continuous versus discrete with subitizing. *SP*, ANNs that evolved to excel in size perception; *Sub*, ANNs that evolved to excel in subitizing; *C*, ANNs that evolved to excel in counting; and *SP-Sub-C*, ANNs that evolved to excel in all three tasks (size perception, subitizing, and counting). The "*" represents significant differences: the counting score in SP-Sub-C run was significantly higher when evolved with continuous and tested with discrete stimuli than the counting score received when evolved with discrete and tested with continuous stimuli.

F (1, 464) = 0.23, MSE = 0.009, p = 0.63, η_p^2 = 0.0005. The average counting score in the counting run when evolved with continuous stimuli and tested with discrete stimuli was M = 0.61, SD = 0.16. The improved counting score after adding the subitizing task between the size perception and the counting tasks was M = 0.77, SD = 0.15.

6.7 SUMMARY OF MAIN RESULTS

The results of Simulation 1—*From Size Perception to Counting*—indicated that the counting skills can be improved [ie, higher counting scores, less complex structure of the net (less inner nodes and less links), but it took more generations] if ANNs were first evolved to perform another, simpler, classification task (eg, size perception or some other classification task) and then evolved further to count (Katz et al., 2013).

In Simulation 2—*Continuous Versus Discrete*—we found that training with continuous stimuli resulted in significantly better counting skills than training with discrete stimuli, despite the reasonable assumption that discrete stimuli would lend themselves better to the counting task.[d]

In addition, evolving with discrete stimuli resulted in larger and more complex networks (ie, more inner nodes and links) than when evolving with continuous stimuli. Moreover, it seems that a certain division between continuous and discrete stimuli appears to be useful when training ANNs to improve their counting skills (Katz et al., 2013).

Finally, our results in Simulations 3a and 3b—*Adding a Subitizing Task and Continuous Versus Discrete With Subitizing*—indicated that subitizing was indeed a key stage in the evolution of counting systems.

It is important to mention that the division we did between discrete and continuous stimuli might not reflect nature. As mentioned in the Stimuli section (under Methods), we defined a stimulus as continuous if there was a path from each visible cell in the array to every other visible cell that passed through visible cells (in single up/down/left/right steps). Nevertheless, the division might have implications for the development of rehabilitation methods. For example, it is known that dyscalculic people have poor counting skills, but based on their compromised size effects (eg, size congruity effect: Rubinsten & Henik, 2005, 2006), they might also have deficits in the ANS or in the networks interfacing ANS with the exact counting system. Thus, training in size perception tasks with continuous stimuli might result in improving their counting skills.

6.8 DISCUSSION

In the current research, we examined the following two hypotheses: (1) the counting system developed through evolution from a more primitive size perception system and (2) both systems evolved independently in different epochs of time. According to our results, the counting system that evolves from scratch fails to excel at counting; *thus it is possible that the counting system we are familiar with might have evolved on the back of some less precise system, instead of evolving independently*.

However, it appears that better counting systems can evolve from different kinds of primitive systems (classification systems in our case) with no specific relation to size perception. Yet, this finding may be due to the size perception task that we chose (the BIG/SMALL decision) or because of the small dataset.

In addition, it took more generations to evolve a proficiency in counting in all the evolutionary runs that contained at least two tasks than when evolving only to count from scratch. This might be because the networks had to change their structure in order to adjust to a new task in the middle of the run (eg, to switch from size perception to counting, or from subitizing to counting).

The work of Kashtan, Noor, and Alon (2007) might shed some light on the reason for high counting scores that were as probable when evolving from populations previously trained for control tasks as they were when evolving from populations trained for size perception. Kashtan and coworkers made the ANNs switch between goals during an evolutionary learning process and found that as each new goal shared some subproblems with the previous goal, the networks evolved to be more modular and developed modules that were useful for both tasks. They found that

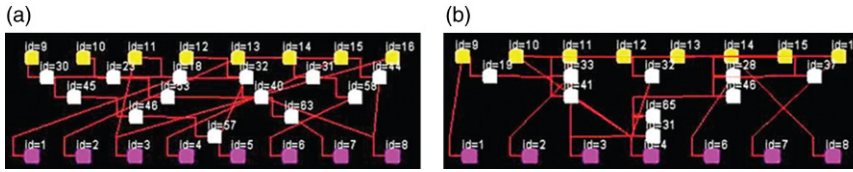


FIGURE 6.8 ANN's structure after evolutionary process is done. (a) The structure of the best ANN individual who evolved to count from scratch with discrete stimuli and tested with continuous stimuli, achieving a counting score of 57.4. (b) The structure of the best ANN individual who evolved to count from scratch with continuous stimuli and tested with discrete stimuli, achieving a counting score of 98.1. Interestingly, this individual survived with a mutation that removed input #5 node through evolution *Yellow*, output neurons; *Purple*, input neurons; *White*, inner neurons. We can see in (a) that there are more inner nodes when evolved to count with discrete and tested with continuous stimuli than the other way around (b), meaning the problem was more complex for the ANN in (a) thus additional inner nodes were required in order to perform the counting task in this case. (Katz et al., 2013).

modularly varying goals could push populations away from local fitness maxima, thus guiding them toward evolvable and modular solutions; in addition, the harder the problem, the faster the speedup. Although, in the current study, the goal was switched between size perception and counting only once, in future research we may switch the goals alternately until the ANNs excel at both tasks.

6.8.1 Complexity of the Net

Overall, when we examined the networks structure, we saw that ANNs who evolved to count with discrete stimuli had more complex networks than when evolving with continuous stimuli (ie, more inner nodes, more links between nodes). In addition, evolving with continuous but testing with discrete led to better counting scores than the other way around. When we examined the structure of the ones who evolved on discrete, we saw an economical (ie, less inner nodes, less links), well-organized structure. Fig. 6.8 shows the structures of two ANN individuals from the counting (C) runs. The bottom layer is the input layer, the top layer is the output layer, and the inner nodes are found in between them. Note that each individual ANN at the beginning of the evolutionary process has only 8 inputs and 8 outputs. The hidden nodes and the network's connections change during the evolutionary process resulting in the final networks.

References

Brannon, E. M., Lutz, D., & Cordes, S. (2006). The development of area discrimination and its implications for number representation in infancy. *Developmental Science*, *9*, F59–F64.

Cantlon, J. F., & Brannon, E. M. (2007). How much does number matter to a monkey? *Journal of Experimental Psychology: Animal Behavior Processes*, *33*, 32–41.

Cantlon, J. F., Platt, M. L., & Brannon, E. M. (2009). Beyond the number domain. *Trends in Cognitive Sciences*, *13*, 83–91.

Cantlon, J. F., Piantadosi, S., Ferrigno, S., Hughes, K., & Barnard, A. (2015). The origins of counting algorithms. *Psychological Science*, *26*(6), 853–865.

Cohen Kadosh, R., Henik, A., Rubinsten, O., Mohr, H., Dori, H., Van de Ven, V., Zorzi, M., Hendler, T., Goebel, R., & Linden, D. (2005). Are numbers special? The comparison systems of the human brain investigated by fMRI. *Neuropsychologia*, *43*, 1238–1248.

Dehaene, S. (2001). Is the number sense a patchwork? *Memory and Language*, *16*, 89–100.

Fias, W., Lammertyn, J., Reynvoet, B., Dupont, P., & Orban, G. A. (2003). Parietal representation of symbolic and nonsymbolic magnitude. *Journal of Cognitive Neuroscience*, *15*, 47–56.

Flavell, J. H. (1963). *The developmental psychology of Jean Piaget*. New York: Van Nostrand Reinhold.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Halberda, J., Mazzocco, M. M. M., & Feigenson, L. (2008). Individual differences in nonverbal number acuity correlate with maths achievement. *Nature*, *455*, 665–668.

Henik, A., Leibovich, T., Naparstek, S., Diesendruck, L., & Rubinsten, O. (2012). Quantities, amounts, and the numerical core system. *Frontiers in Human Neuroscience*, *5*, 186.

Henik, A., Rubinsten, O., & Ashkenazi, S. (2014). Developmental dyscalculia as a heterogeneous disability. In R. Cohen Kadosh, & A. Dowker (Eds.), *The Oxford handbook of numerical cognition* (pp. 662–677). Oxford: Oxford University Press.

Kashtan, N., & Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences*, *102*, 13773–13778.

Kashtan, N., Noor, E., & Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, *104*, 13711–13716.

Katz, G., Benbassat, A., Diesendruck, L., Sipper, M., & Henik, A. (2013). From size perception to counting: an evolutionary computation point of view. In: *Proceedings of the 15th annual conference companion on genetic and evolutionary computation (GECCO 2013) companion* (pp. 1675–1678). New York: ACM.

Kaufman, E. L., Lord, M. W., Reese, T. W., & Volkmann, J. (1949). The discrimination of visual number. *American Journal of Psychology*, *62*, 498–525.

Kosc, L. (1974). Developmental dyscalculia. *Journal of Learning Disabilities*, *7*, 159–162.

Lourenco, S. F., Bonny, J. W., Fernandez, E. P., & Rao, S. (2012). Nonsymbolic number and cumulative area representations contribute shared and unique variance to symbolic math competence. *Proceedings of the National Academy of Sciences*, *109*, 18737–18742.

Piaget, J. (1955). *The language and thought of the child*. Cleveland, OH: World Publishing Company.

Pinel, P., Piazza, M., Le Bihan, D., & Dehaene, S. (2004). Distributed and overlapping cerebral representations of number, size, and luminance during comparative judgments. *Neuron*, *41*, 983–993.

Price, G. R., Holloway, I., Räsänen, P., Vesterinen, M., & Ansari, D. (2007). Impaired parietal magnitude processing in developmental dyscalculia. *Current Biology*, *17*, R1042–R1043.

Rubinsten, O., & Henik, A. (2005). Automatic activation of internal magnitudes: a study of developmental dyscalculia. *Neuropsychology*, *19*, 641–648.

Rubinsten, O., & Henik, A. (2006). Double dissociation of functions in developmental dyslexia and dyscalculia. *Journal of Educational Psychology*, *98*, 854–867.

Shalev, R. S. (2007). Prevalence of developmental dyscalculia. In D. B. Berch, & M. M. M. Mazzocco (Eds.), *Why is math so hard for some children? The nature and origins of mathematical learning difficulties and disabilities* (pp. 49–60). Baltimore, MD: Paul H. Brookes Publishing Co.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, *10*(2), 99–127.

Stoianov, I., & Zorzi, M. (2012). Emergence of a "visual number sense" in hierarchical generative models. *Nature Neuroscience*, *15*(2), 194–196.

Verguts, T., & Fias, W. (2004). Representation of number in animals and humans: a neural model. *Journal of Cognitive Neuroscience*, *16*(9), 1493–1504.