

# **A competitive neural model of small number detection**

Matthew C. Casey<sup>1</sup>, Khurshid Ahmad<sup>2</sup>

*<sup>1</sup>Department of Computing, School of Electronics and Physical Sciences,  
University of Surrey, Guildford, Surrey, GU2 7XH, UK*

*<sup>2</sup>Department of Computer Science, O'Reilly Institute,  
Trinity College, Dublin 2, Ireland*

## **Acknowledgements**

The authors would like to thank Tony Browne for his helpful comments on a draft of this paper and the anonymous reviewers for their constructive comments and suggestions for improvements.

## **Corresponding Author**

Matthew Casey

m.casey@surrey.ac.uk

Tel: +44 (0) 1483 689635

Fax: +44 (0) 1483 686051

Word Count: 12715

## A competitive neural model of small number detection

### **Abstract**

The ability to represent numbers is a key attribute for both humans and animals. Recent developments in the understanding of numerical processing has lead to the proposal that humans utilise two independent representations of number, one for real numbers and another for integers. We describe a computational model of small number detection to explore the relationship between these *core systems of number*. We use a combination of unsupervised and supervised neural networks to simulate the interaction between the real and integer representations. For real values we use a self-organised spatial representation of number. For integer values we use a supervised network motivated by linguistic processing. During training and testing, the networks exhibit behavioural characteristics such as the number size and numerical distance effects. Each representation is combined using the mixture-of-experts architecture that allows us to model the subitization limit (the maximum number of visual stimuli that can be accurately quantified almost immediately) as the competitive allocation of representations for number detection, where the crossover point between deploying the real and integer representations of number is obtained through a process of learning. Our results suggest that the existence of two core systems of number is at least computationally plausible and further suggests that the subitization limit emerges through the interaction of spatial and linguistic numerical processing. This provides computational evidence for one way in which small and large numbers are related in humans.

### **Keywords**

Neural modelling; Number; Subitization; Quantification; Modular systems

## 1 Introduction

The ability to represent numbers is a key attribute for both humans and animals. With the biological basis of number being investigated with techniques such as functional imaging (see for example, Hubbard, Piazza, Pinel, & Dehaene, 2005 for a review; Ansari, Garcia, Lucas, Hamon, & Dhital, 2005), models of number representations are now being proposed that consist of distinct systems to represent real and integer values independently (Feigenson, Dehaene, & Spelke, 2004). However, this is in contrast to the traditional view that there is one system of number that represents just real values, from and to which we are able to translate integers (Gallistel & Gelman, 2000) and for which there have been successful computational models (Grossberg & Repin, 2003). This highlights one of the outstanding questions of numerical cognition: what is the relationship between the representation of small and large numbers? In this article we describe a computational model of number detection that combines unsupervised and supervised learning competitively, building upon previous connectionist simulations of cognitive abilities. We use this model to explore the relationship between the proposed representations of approximate magnitudes and precise values to help address this question.

The key phenomenon that we attempt to simulate is the *subitization limit* (Kaufman, Lord, Reese, & Volkman, 1949), defined as the limiting number of items that can be quantified visually almost immediately. This is central to our models of the two separate number representations, which have developed from investigations into numerical processing. For example, in general, models of numerical processing have linked together real and integer forms of number with quantification, and recognised that at the core of numerical cognition there is an abstract representation of number (Dehaene, 1992; McCloskey, 1992); this abstract core is perhaps also the foundation of our understanding of time and space (Walsh, 2003). Such *number sense* (Dehaene, 1997) has been explored in animals and humans (see for example, Brannon & Terrace, 1998; Wynn, 1998), with the suggestion that it has a shared ancestry, which has led to the establishment of the accumulator mechanism as one of the preferred quantification models (Meck & Church,

1983). However, whilst this mechanism may account for our early understanding of small numbers, it is uncertain what impact this may have on the later development of integer processing, and indeed if the accumulator is biologically plausible (Piazza & Dehaene, 2004).

With the advent of comparative experiments between nonverbal animals, preverbal humans and psychophysical techniques, improvements in our understanding of the biological basis of number have helped to locate areas of animal and human brains that appear to explicitly encode number (Thompson, Mayers, Robertson, & Patterson, 1970; Dehaene, Dehaene-Lambertz, & Cohen, 1998; Stanescu-Cosson et al., 2000; Nieder, Freedman, & Miller, 2002; Simon, Mangin, Cohen, Le Bihan, & Dehaene, 2002; Brannon & Terrace, 2002; Nieder & Miller, 2003; Dehaene, 2003; Walsh, 2003; Orlov, Amit, Yakovlev, Zohary, & Hochstein, 2006). However, the existence of these ‘number neurons’ does not quite explain whether they form the foundation of our proposed magnitude or precise value representations. The key to understanding whether there are one or two core systems of number appears to lie in the way in which small numbers are detected and reasoned about.

The subitization limit provides a focus upon which the proposed core systems of number can be explored because of the recorded increase in error rates or reaction times in adults in quantifying a visual stimuli consisting of increasing quantities. This discontinuity in discrimination, thought to occur at about a quantity of four, is traditionally understood as a result of the deployment of the different quantification processes, such as subitization and counting (Mandler & Shebo, 1982), associated with visual processing of pre-attentive information, or through guided attention respectively (Trick & Pylyshyn, 1994). In humans, one clear influence is the development of language (Dehaene, Spelke, Pinel, Stanescu, & Tsivkin, 1999; Gelman & Butterworth, 2005), which underlies processes such as counting and allows us to draw a distinction between visual (subitization) and linguistic (counting) numerical processing. However, with evidence for numerical reasoning in animals (Brannon & Terrace, 1998; Orlov et al., 2006), and with some doubt over the evidence for the existence of a limit (Gallistel & Gelman, 2000), the influence of language in numerical development has recently been suggested as playing a lesser role (Gelman & Butterworth, 2005; Varley, Klessinger, Romanowski, & Siegal, 2005).

Once a quantity has been apprehended, whether through a process of subitization or counting, our understanding of number is subject to three other notable phenomena. First, the number size effect refers to the increase in difficulty in processing larger numbers. Second, the numerical distance effect refers to the increasing ease with which numbers can be discriminated when the distance between them becomes larger (Dehaene, 1997). Third, the first two phenomena are linked by the Weber and Fechner laws for discriminating stimulus threshold increments, the latter of which has recently been supported by psychophysical evidence (Dehaene, 2003). These phenomena provide us with characteristics that our models must explain, and these depend upon the mechanisms used for representing numbers. However, whilst the number size and numerical distance effects perhaps describe the behaviour of the individual representations, the subitization limit is related to the way in which these representations interact. Here, with two core systems of number, the implication is that the limit results from the way in which the representations are deployed. This limit is manifest in the crossover point between the two core systems, in contrast to the more traditional view that with just one system of number, the limit must result in some way from the translation from real values to integers.

In this article, we describe a computational model of small number detection in which we integrate an approximate magnitude and precise value representation of number. We combine both representations using a multiple neural network (multi-net) model with a competitive learning strategy. Here, we combine unsupervised learning in a supervised learning architecture by using a self-organising map (SOM) (Kohonen, 1982) and multi-layer perceptron with the mixture-of-experts (ME) architecture (Jacobs, Jordan, & Barto, 1991). This model brings together the neurophysiologically motivated SOM, used with some success for simulating aspects of vision (cf. Cameron, Grossberg, & Guenther, 1998), with the ME architecture, itself motivated by the simulation of ‘what’ and ‘where’ processing, but with a robust statistical foundation. The combination of these two learning strategies allows us to build a model of small number detection trained with varying parameters to simulate the dominance of either number representation. In contrast to previous approaches (for example, Peterson & Simon, 2000; Domijan, 2004), we train our model using data that simulates the exposure children have to different

quantities, modelling this through analysis of a corpus of spoken language. The results demonstrate how the combination of both representations can lead to a simulation of the subitization limit, where the discontinuity depends only upon the speed at which each representation learns how to represent numbers. In section 2 we describe our modelling approach. Section 3 details the proposed model and method for constructing the training data (equations and learning algorithm used in our study are provided in Appendix A and B respectively). Section 4 describes the experiments performed, together with the results, relating these to previous computational work to behavioural studies. Finally, in section 5 we discuss the implications of this work within a wider context, and propose further directions for research.

## **2 Modelling approach**

Several models of numerical processing, and in particular those describing quantification, have been developed that incorporate some form of number representation. Evidence from these psychological models (McCloskey, Caramazza, & Basili, 1985; Dehaene, 1997), and more recently from psychophysical experiments (Nieder et al., 2002), seems to provide us with a consistent picture of the form that such a number representation must take. Here, phenomena such as the number size and numerical distance effects, Weber and Fechner laws, and evidence for neurons that are associated with particular quantities (Nieder & Miller, 2003), points to a mental number line as the foundation of numerical knowledge (see for example, Dehaene, 2003). Much evidence also suggests that this number line is formed through the use of an accumulator mechanism (Cordes, Gelman, Gallistel, & Whalen, 2001) and some form of topographic spatial representation of number (Simon et al., 2002). There is also a distinction between the existence of dedicated neural circuitry for ‘number sense’ in contrast to the use of existing cognitive functions, such as language, for acquired mathematical abilities, such as multiplication (Spelke & Dehaene, 1999). This perhaps explains further the distinction that has been drawn between the two proposed representations of number.

The process for detecting numbers allows us to explore number representations. In particular, models of subitization and counting have been developed that explore the

distinction between what is thought to be visual and linguistic processing, respectively. Whilst Gelman and Gallistel (1978) preferred to attribute subitization to a form of preverbal counting, Mandler et al. (1982) argued that we learn to subitize through the recognition of canonical patterns. This has had substantial opposition from Wynn (1998), who argues that the ability to subitize is inborn and shared with other species. In contrast, counting is the learnt symbolic process by which an accurate, cardinal value for the numerosity of a set of items can be determined through the use of a serial set of rules (Fuson, 1988), and unlike the preverbal mechanism proposed for subitization, counting appears to rely upon the development of language.

Neural simulations of subitization have concentrated upon either psychologically or biologically plausible mechanisms, exploring the use of a spatial number line to explain phenomena such as the number size and numerical distance effects, and hence the subitization limit (Dehaene & Changeux, 1993; Peterson & Simon, 2000; Ahmad, Casey, & Bale, 2002; Domijan, 2004). In contrast, models of counting have concentrated upon the implementation of the counting procedure (Amit, 1988; Hoekstra, 1992; Rodriguez, Wiles, & Elman, 1999), but have also simulated the acquisition of the number-word sequence (Ma & Hirai, 1989; Ahmad et al., 2002). Table 1 summarises the different neural models that have been proposed.

Dehaene and Changeux (1993) first simulated subitization using a series of pre-defined networks that could convert a visual scene into an internal, abstract representation of number. Here, attempts were made to ignore location and size characteristics of the input, with learning of the association between the detected number and the required output. By using a topographic representation, they demonstrated an ordered number line, demonstrating both Fechner's law and the distance effect, with the subitization limit attributed to both the internal representation of numerosity, together with the accuracy of the visual normalisation. Ahmad, Casey and Bale (2002) also simulated visual subitization with modules for object normalisation and numerosity representation. Object normalisation was achieved using two sequentially connected networks that operated on the visual scene to produce a scale and translational invariant output. This normalised output was then presented to a SOM that learnt the topographic relationships between different numerosities, much like the model by Dehaene and Changeux (1993), but

through a process of unsupervised learning. The results demonstrated how the SOM could learn to order the numerosities into a compressive number line, demonstrating both Fechner's law and the distance effect, with the subitization limit attributed to the boundary effects from the SOM training algorithm.

Domijan (2004) recently revisited Dehaene and Changeux's model in an attempt to incorporate current biological knowledge, achieving number detection irrespective of size, density or shape of objects in a visual scene. With the focus on biological plausibility, this model demonstrated capabilities as observed in primates. The use of a winner-takes-all strategy for number detection was commensurate with the topographic nature of Dehaene and Changeux's model, highlighting the evidence for this type of mechanism as the foundation of the approximate magnitude representation.

Peterson and Simon (2000) compared two simulations of subitization using the rule-based ACT-R architecture, and a multi-layer perceptron (MLP) with backpropagation learning. The use of an MLP for connectionist simulations is a traditional method despite its apparent biological implausibility. They simulated the subitization process to produce a precise value representation by mapping a visual scene to a distinct value. However, their conclusions regarding the subitization limit in their MLP based SUBIT-PDP are similar to those of Dehaene and Changeux, noting that the limit appears to be consequent from the interaction between the number of hidden units and the size of the visual scene.

With counting, despite its recognised foundation in the development of language, the majority of simulations have focused on the counting procedure with temporal processing. For example, Amit's (1988) chime-counting multi-net simulation identified temporal sequences of stimuli using delayed synapse connections, with inputs pre-processed by a Hopfield network. Similarly, Rodriguez, Wiles and Elman (1999) focused on temporal processing in their model of formal language processing. Here, the recurrent backpropagation through time (BPTT) network was used to count a string of letters, the quantity of which was used to predict the length of a second string. Since both these temporal simulations explore the counting process without language, they can be taken as models of either language-based or preverbal counting.



In contrast, Ma and Hirai (1989) concentrated on the development of the number-word sequence. They used the combination of a heteroassociative network and a recurrent inhibitory network, with results demonstrating conventional, stable nonconventional and nonstable elements of number-word production (see for example, Fuson, Richards, & Briars, 1982). A similar approach was taken by Ahmad et al. (2002), who modelled the act of pointing at each individual object whilst repeating the word representing that number to establish a one-to-one correspondence. The results were compared with those observed in children, demonstrating a similar level of errors in the production of word sequences and in pointing.

Whilst these models have focused on individual quantification abilities, in order to explore the core systems of number it is important to explore the possible differences in the way in which integer and real (non-integer) numbers may be processed. Simulations of subitization have implicitly included the translation of real to integer values, with magnitudes represented on a notional number line that are coupled to discrete outputs (cf. Dehaene and Changeux's output clusters). Whilst this translation is not bi-directional, the implication appears to be that there is just one core system of number that operates on real values. In contrast, simulations of counting have focused upon temporal stimuli to produce integer-valued outputs. This temporal processing is a key aspect of precise valued quantification.

Two models have recently attempted to bring together real values, integers and temporal processing into a single architecture for multi-digit comparison. Grossberg and Repin's (2003) SpaN and ESpaN models incorporate a spatial numerical representation of number that learns to order quantities, motivated by the visual processing stream. Temporal processing is included in the pre-processing stage on the input, which is used to build an analogue value, constructed using the accumulator mechanism. The analogue value is input to the spatial representation. In the SpaN model, this spatial representation consists of just one-dimension to model a number line, exhibiting Weber's law in its organisation. In the extended ESpaN model, multi-digit categories are mapped against the number line with a two-dimensional representation, hence allowing the translation of real to integer values. This model of multi-digit comparison suggests the idea that there is a single real valued core system of number, especially with pre-processing of inputs to normalise

temporal and atemporal stimuli using an accumulator mechanism. However, whether the use of an accumulator fits with observed behaviour is still uncertain (see for example, Piazza & Dehaene, 2004), nonetheless aspects of the mechanism have proven viable in computational models and for pre-processing (cf. Amit, 1988; Ahmad et al., 2002).

Grossberg and Repin's key idea for the SpaN model is that spatial numerical processing has developed within the 'where' processing stream, whereas they associate multi-digit numbers with the 'what' processing stream (cf. Carpenter, Grossberg, & Leshner, 1998). A similar modular approach to cognitive simulation was developed by Jacobs et al. (1991) with the supervised learning ME architecture, first used to simulate 'what' and 'where' processing. They argued that it is appropriate to consider the modular nature of connectionist models at the level of networks, rather than at the level of individual neurons or layers because intuitively we can crudely equate areas of the brain with individual networks, rather than just neurons and layers within a network. A modular approach with unsupervised learning has also been taken by Cameron et al. (1998), when they combined together self-organising networks for visual navigation (motivated by biological mechanisms).

In this paper, we use these different approaches to construct a computational model of number detection using self-organisation within the ME architecture. In a similar way to Grossberg and Repin and other previous simulations of subitization, we develop a spatial representation of number that uses a SOM to form a number line. However, we combine this spatial magnitude representation with an independent representation for precise values that uses a supervised approach to connectionist simulations with an MLP, building on the counting simulations described above. Both representations are combined in a single ME architecture to allow each to compete to perform the number detection task. In this way, we compare computationally the capabilities of each simulated representation in order to better understand the interaction between the proposed two core systems of number.

### 3 Simulating small number detection

Our *Competitive Detection Model* (CDM) consists of four processing elements: pre-processed input, magnitude representation, precise value representation, and combined output detection (Fig. 1). For our model of approximate magnitudes, we use a spatial representation of number formed by a topographic map using a SOM. This fits well with studies of the biological basis of number representation in animals, which seem to demonstrate that groups of neurons encode specific quantities, as well as with other topographic and competitive simulations that learn to organise number spatially to exhibit known behavioural characteristics, such as the numerical distance and size effects. The topographic properties of SOM are critical for simulating such a spatial representation, where the *magnification factor* (Kohonen, 1982) phenomenon facilitates the generation of a compressive number line by making certain assumptions about the training regimen and data. In contrast, our model of precise values uses the traditional MLP with backpropagation approach to map a visual scene to a precise value.

To exploit the magnification factor in SOM learning, we weight the presentation of lower valued quantities during training. Typically, equal frequencies of training objects have been presented in such simulations (except Dallaway, 1994), and this is perhaps due to the difficulty in justifying any bias in numerical development. However, such a bias is at least intuitive when we consider the development of symbolic number understanding as characterised by linguistic sources of child and carer language. It can be argued that in learning to count or perform arithmetic, children are more frequently exposed to the lower numbers (such as in rote learning activities). To understand this, we analysed two corpora of linguistic information to determine the frequency of number words. The British National Corpus (BNC), which includes 4,124 spoken and written texts (Burnard, 1995), shows that this intuitive view is correct: the lower the number, the higher the frequency. Exceptions to this are particular number words peculiar to the spoken decimal system, such as ‘ten’ and ‘twenty’, which are more frequent. However, whilst this general language corpus contains a large sample of texts, it is necessarily focused on adult language, and hence does not demonstrate the extent to which children are exposed to particular numbers, as might be expected during the development of numerical skills.

Consequently, we performed the same number word analysis of a small corpus of conversations held between young children and adults contained in the CHILDES database (MacWhinney, 2000).

This corpus consists of transcriptions of 299 recorded conversations with British children ranging in age from 1½ to 5 years old (Wells, 1981). Performing a similar word analysis as that performed on the BNC gives rise to a model with a far higher bias towards lower numbers, which appears to decrease rapidly according to the inverse power of the number. Although this analysis provides us with evidence for the bias apparent in numerical language development, it does not tell us the effect that visual stimuli may have on numerical development (if there is such an effect at all). Here, we make the assumption that a similar visual phenomenon occurs, perhaps expressed through ideas such as the learning of canonical patterns for subitization (Mandler & Shebo, 1982).

We assume that each system has the same input, formed through a process of modality and temporal invariance to give an accumulated analogue value. Here we follow the work of Grossberg et al. (2003) to use the accumulator as a pre-processing mechanism. However, whilst this assumption replaces any modelling of sensory processing that occurs in the brain prior to quantification, we recognise that such processing may actually play an important role in numerical abilities itself because the analogue representation is used to help associate quantities together in the magnitude representation. The role of this pre-processing has been partly explored in previous simulations where it has been used to abstract an analogue value from a visual scene through a process of scale and translational invariance (Ahmad et al., 2002). However, the use of an accumulator is still a matter for ongoing debate.

With both representations competing on the analogue input in parallel, we observe the behaviour developed through a process of learning. To achieve this we use the ME architecture, which allows experts, in our case the different representations, to compete, rewarding the winner by allowing it to learn more on the selected input pattern. The constraint that this places on the model is that the output from each expert is of the same form; the output from the approximate magnitude representation, which is modelling a number line, must be the same as the output from the precise value representation. The

translation between magnitudes and precise values at this stage is necessitated by architectural constraints, and is motivated by the assumption that approximate magnitudes can be mapped to precise values, given the approximate nature of this real number representation and the assumption that this is negligible for sufficiently small numbers. The model takes a processed sensory input encoded as an analogue value and learns to transform this into a symbolic output by associating together many such values during training. This transformation is achieved through two different learning processes. First, the magnitude representation learns to order the quantities, much like the experiments performed with rhesus monkeys (Brannon & Terrace, 1998), to form a number line. This internal ordinal representation consists of self-organising process driven by the analogue inputs, coupled to a feedback driven translation to a symbolic output. Second, and in contrast, the precise value representation learns to label each analogue input directly without exploiting ordinality.

In the CDM, the pre-processed input is supplied to both the magnitude and precise value representations. These operate in parallel on input to produce an output that is weighted by the output detection stage according to an assessment of the input. This weighting is achieved by estimating which representation will give the best response, an estimate that is refined during training using the ME learning algorithm. (Appendix A comprises the equations used for each of the networks.)

*Pre-processed input:* in simulating different representations of number, several approaches to input stimuli have been taken. Whilst some have defined the input as abstract magnitudes, others have attempted to simulate a visual scene (McCloskey & Lindemann, 1992; Anderson, Spoehr, & Bennett, 1994; Dallaway, 1994). We use an analogue form of input that has an overlap for each successive value. The motivation for this is based upon the representation of an accumulated analogue value, which can be exploited by SOM learning to relate different quantities together and order them appropriately into a number line. For the purposes of this article, we assume that the accumulator mechanism has been used to generate appropriate input vectors, since it has already been extensively described for visual scenes (Ahmad et al., 2002) and modality and temporally invariant inputs (Grossberg & Repin, 2003). A single input value is used to represent a quantity, with values ranging from '1' to '20', to give 20-dimensional input

vectors, as shown in Table 2. For construction of this input, we use the number word probability distribution obtained from the CHILDES database to generate 2000 training patterns for values from '1' to '20'. Each pattern is generated by selecting a random number of objects using the defined probability model. The distribution of target and sample training data is shown in Fig. 2.

*Magnitude representation:* to simulate approximate magnitudes we use two networks connected and trained in sequence (Casey & Ahmad, 2004). The first network is a SOM used to form the number line. The unsupervised learning algorithm for the SOM facilitates competition between input signals so as to form an ordered map of similar patterns. The analogue form of the input ensures that the ordering produces a number line (see for example, Ahmad et al., 2002), whilst the frequency of individual training patterns ensures that a compressive representation is obtained. This magnification factor results from the organisation of the prototype vectors associated with each neuron during SOM learning, which depends upon the statistical properties of the input space. If, for example, there is a greater frequency of patterns of one type, then the weights associated with a larger number of neurons will be adjusted to be most like these patterns, in comparison with those of a lower frequency.

The second network is used to translate the output of the SOM into a precise value. Whilst the SOM uses unsupervised learning, the mapping of the number line to a precise value is achieved using a single layer perceptron (SLP), trained using the delta learning rule (Widrow & Hoff, 1960). This mapping models the association of precise values to a foundation of numerical magnitudes, with such transcoding central to all of the models of numerical processing discussed. Here we use a simple linear mapping, whereas we could use a bi-directional associative mapping if we were to simulate the proposed bi-directional translation from real to integer values. However, for our purposes, a single-directional mapping is sufficient. The input to the SLP is formed from the output of the SOM as a single vector that consists of elements representing each of the neurons in the map, such that the winning neuron for an input pattern, as determined by the Euclidean distance, is assigned the value '1', whilst all other neurons are assigned the value '0'.

*Precise value representation:* this builds upon the use of a multi-layer perceptron to simulate symbolic processing. A single layer of hidden neurons is used together with an output layer to assign the analogue input to a precise value through a non-linear mapping, learnt using the backpropagation algorithm (Werbos, 1974; Rumelhart, Hinton, & Williams, 1986). Whereas the magnitude representation can be visualised as a compressive number line, the encoding of number within the MLP has no neural correlate due to its abstract nature. Our motivation for this form of representation comes from the traditional route of simulating symbolic processes using an MLP (for example, Peterson & Simon, 2000). Instead, we will look at the activation values resulting from the network's output layer in order to further understand the operation of this representation.

*Output detection:* the final stage of the system combines the outputs of both the magnitude and precise value representations. The combination is weighted depending upon the particular input stimuli presented. This is achieved using the gating network from the ME architecture, which combines the output of each expert (magnitude or precise value representation) to produce the best combination by estimating the posterior probability of a correct response, learnt from the training data. This competitive process uses a winner-takes-all scheme where the weighting is also used to control learning in each expert (Jacobs, Jordan, Nowlan, & Hinton, 1991).

Training of the system proceeds on each training pattern individually for the required number of epochs. (Details of the algorithm are in Appendix B.) Once the output for a given input has been calculated, the target value is presented to generate an error signal as per a supervised learning scheme. Prior to each expert learning from the input and target, the gating network is trained upon the posterior probability of each expert being correct. The posterior probabilities are used to weight the learning of each expert independently, hence implementing the required competitive learning scheme. Whilst the learning in the precise value representation follows that of the backpropagation algorithm, the approximate magnitude representation requires both networks to be trained in sequence. This is achieved using a sequential learning algorithm that relies upon the use of an unsupervised network within the sequence (Casey & Ahmad, 2004).

## 4 Simulations and results

The modular CDM is designed to decompose the detection task to two experts, one expert represents approximate magnitudes, and the other precise values. This automatic decomposition through training allows us to explore small number detection through the seamless integration of magnitudes and precise values; we have assumed that small magnitudes can be translated into integers with negligible noise.

The integration of the two representations depends upon the capabilities of each, as determined by their competitive combination. In order to quantify the conditions under which the model exhibits a crossover between the two representations, we vary the model parameters to affect the capability of each representation to represent numbers and translate these into a precise-valued output. Training of each system on the CHILDES data set is intended to simulate the development of the number representations in children. Once trained, we test the model on the range of numbers within the input space to determine which of the experts dominates on the detection task, noting the crossover point between expert allocation. In this simulation, the crossover point is assumed to model the subitization limit, where one representation dominates detection for lower numbers, whilst the other dominates for higher numbers. Under this scenario, the allocation of experts is dependent only upon the input stimulus, and is learnt during the training period, influenced by the topologies and parameters for each network. It can be argued that the training regimen described above is our attempt to simulate ontogenetic development.

Details of the network topologies and learning parameters for the simulations using the CDM are given in Table 3, following the model equations (in Appendix A), and the algorithm (in Appendix B). Although the system consists of a number of parameters that can be varied, we focus upon three only, as these have a direct relationship to the capabilities and allocation of each expert. First we vary the size of the one-dimensional SOM ( $M$ ) in order to determine the capability of the magnitude representation to learn to detect the presented range of numbers. With a small map size, detection is potentially more difficult because there are fewer neurons available to represent all of the training patterns. This is further compounded by the skewed nature of the training data, which



biases lower numbers (as per the CHILDES probability model). With a higher frequency of low-numbered training patterns, a larger number of neurons in the map should be allocated to these patterns. In contrast, the lower frequency patterns will be allocated to far less neurons (if any), modelling the noise expected with this representation for higher numbers. By varying the map size, we can determine the effect of too few or too many neurons, and hence the highest number that can be sufficiently represented without being overcome by noise in the translation to an integer.

Second, we also vary the number of hidden neurons in the precise value MLP ( $N$ ) to explore the ability of the network to represent low and high numbers. Within an MLP, by adding hidden neurons the network is capable of modelling a greater degree of complexity. Typically, such increased complexity can lead to overfitting of the network, where the network models too closely the training data at the detriment of generalisation. By having a sufficient number of hidden neurons, the MLP should be capable of mapping the input patterns to the precise value output. However, with insufficient neurons, it is reasonable to expect the network to successfully map only a small number of inputs.

Third, we vary the learning rate of the transcoding SLP ( $L$ ). Whilst fixing the topology and learning parameters of the SOM and MLP, we can vary the allocation of the magnitude and precise value representations by increasing or decreasing the speed at which the transcoding from magnitude to precise value is learnt. This in effect models the development of categorical learning of precise values against a foundation of magnitude numerical knowledge. For example, if the magnitude representation is well established prior to the representation for precise values, then the speed at which the transcoding can be learnt will reflect directly on the integration of the two representations.

We used SOM map sizes from  $M=10$  to  $M=100$  with step size 10, MLP hidden layer sizes from  $N=1$  to  $N=10$  with step size 1, and transcoding learning rate  $L=0.1$  to  $L=1.0$  with step size 0.1, giving a total of 1000 different configurations of the system. For each system in which we varied  $M$  and  $N$ , we initialised each network's weights with random real number values, selected using a normal probability distribution of mean 0, standard deviation 1. For a particular system topology ( $M$  and  $N$ ) the same initial weights were

used for each value of the learning rate parameter  $L$ . Each system was trained on 10 presentations of the 2000 training examples (20,000 time steps  $t$ ).

#### 4.1 *Detection responses*

The trained systems were tested on the inputs ‘1’ to ‘20’. For each test, the highest consecutive allocation of either expert producing the correct response was noted. We take only consecutive correct responses to ignore any noise present at the crossover point between the representations. Finally, to help evaluate the behaviour of the model, we record the allocation of patterns within the SOM, together with the ability of the SLP and MLP to output the correct precise value responses.

First, all of the 1000 systems learnt to detect small numbers, with a consecutive minimum of ‘2’ correctly detected, and maximum ‘12’ (mean 6.0, standard deviation 2.4). This range of abilities is an expected result of the variation in network topology across the trials, which induces a variety of capabilities for the system to learn the given task. Second, we note that, irrespective of the generation of correct responses, the magnitude representation was used solely in 25.1% of the trials, and the precise value representation solely in 61.8%. In the 13.1% of simulations in which both representations were used, magnitudes were allocated to the lower numbers in 11.4% of the trials, whilst precise values were allocated to the lower numbers in only 1.7%. This profile suggests that, since the allocation of experts depends upon the speed at which each learns the task, the MLP is typically better at learning the detection task early on in training, in contrast to the SOM and the transcoding SLP. More importantly, given the use of two independent number representations, our model is at least capable of simulating the full range of scenarios of number detection: magnitudes only, precise values only, magnitudes then precise values, and precise values then magnitudes.

Whilst the system tends to favour the precise value expert in combination or exclusively, we are particularly interested in understanding the behaviour of the system when both experts are seamlessly used for detection. Looking at the subitization limit, we assume that this results from the allocation of the magnitude representation to the lower numbers, and the precise value system to the higher numbers (11.4% of the trials), as measured by consecutive correct responses. In such systems, the limit varies from ‘1’ to ‘7’ (mean 2.8,

standard deviation 1.2), whilst the corresponding maximum correct responses vary from ‘3’ to ‘11’ (mean 8.0, standard deviation 2.0). The increase in the limit tends to vary with the transcoding learning rate: with a higher learning rate, a higher limit is produced, which fits well with the observation that the limit depends upon whichever expert can learn the task earlier. For example, in the system with  $M=40$  and  $N=2$ , in which both experts were used for detection, the limit varies incrementally from ‘3’ to ‘5’ for learning rates 0.3 to 0.7. We explore this further by looking at the system with  $L=0.3$  only, which has a limit of ‘3’ (close to the average). Although this is not the best performing system, it does demonstrate the general characteristics of the simulated representations, and hence by way of an example we concentrate on this system for the results reported below for the individual representations.

The learnt output activation values for the numbers ‘1’ to ‘20’ for the selected system are shown in Fig. 3. This shows the weighted activation of the magnitude and precise value experts for each of the numbers in the input space. The weighting of the experts is learnt using the current expert and target outputs. For this system, we examine the rate at which both experts learn to represent numbers in order to determine the factors affecting the overall subitization limit. To achieve this, during training of this selected system, the responses to the test patterns were recorded after each epoch.

*Magnitude representation:* Fig. 4 depicts the number to neuron allocation of the magnitude representation SOM for our selected system after 10 epochs of training. Each line shows the normalised, inverse distance from each of the vectors ‘1’ to ‘20’. This inverse is calculated to show the peak activity of the network, such that the activity  $d$  for neuron  $i$ , given input  $\mathbf{x}$  and SOM weight vectors  $\mathbf{w}$ , is:

$$d^i = 1 - \frac{\|\mathbf{x} - \mathbf{w}_i\|}{\max_j \|\mathbf{x} - \mathbf{w}_j\|} \quad (1)$$

The activity shows that the SOM has learnt to order the numbers from ‘1’ to ‘20’ on the number line (from left to right), and that higher values have fewer neurons, with for example, ‘2’ and ‘3’ represented with a peak of activity for 9 neurons each, ‘4’ with 7, and ‘5’ with 3, which is the highest number to easily discern from this pattern of activity. Above ‘6’ the remaining neurons represent individual values less, with the difference in

distance for successive values becoming far smaller. However, the SOM boundary effects are also evident with the lower number of neurons allocated to '1', whilst the magnification factor is also apparent with the peak of activity seen on the right side of the map for '15', which is more frequent than similar high numbers in the randomly generated training data. It is also important to note that, because of the random selection process and its low probability, there is no training pattern for the number '14', yet it is still represented on the number line in between '13' and '15' in the test responses. Such generalisation is an important aspect of this self-organised representation.

During training, the SOM rapidly learns to represent the numbers, with only 1 epoch required to achieved number ordering. The key difference between this early and later stages of training is the higher number of neurons allocated to '1' and '2'. After 1 epoch, '1' is allocated 8 neurons, and '2' and 14, compared with 5 and 9 neurons, respectively, after 10 epochs. The ability of the SOM to initially represent these low numbers well is matched by the ability of the transcoding SLP to correctly map them to precise values. These two factors result in the long term allocation of low numbers to the magnitude representation in this particular system.

The characteristics of the magnitude representation demonstrated here result from the way in which the SOM learning algorithm organises patterns topologically, coupled with the probability model used for the training data, which exploits the magnification factor phenomena in the SOM algorithm. For example, if we train the same system on a data set generated using an equal probability model, this gives a uniform distribution of neurons for all of the numbers, with no overlap in the peak allocation (Fig. 5). For the CHILDES model, smaller map sizes show that the ordering of numbers is still present, but with a far greater overlap and with fewer neurons allocated to each number. For example, with a map size of  $M=10$  neurons, the peak of activity drops significantly from '3' onwards. For larger map sizes, the results show the reverse trend, with higher numbers being clearly distinguished, such as '17' for  $M=100$ . However, the variation in map size appears to have little effect on the subitization limit, which remains close to the average of 2.8. It appears that the main effect of the magnitude representation on the subitization limit is restricted to the speed at which the SOM learns to order the numbers, and especially to represent the lower numbers, coupled with the speed at which the SLP

learns to map these magnitudes to precise values. The limit is therefore independent of the topology of the representation, and only dependent upon the training data distribution leading to the compressive form of the map and the learning rate.

*Precise value representation:* in our selected system the precise value MLP learnt to represent the numbers ‘4’, ‘5’ and ‘8’ after 10 epochs (Fig. 3). Activations for ‘1’, ‘2’ and ‘3’ were near zero, indicating that the MLP had not learnt the mapping for these patterns, as expected with the competitive learning process and the capabilities of the magnitude representation. Indeed, after 2 epochs the MLP had learnt both ‘4’ and ‘5’ only. The correct responses from the MLP demonstrate how just 2 hidden neurons can learn to map a restricted number of analogue inputs to a precise value. In contrast, systems with a larger number of hidden neurons show greater capability, with for example 9 hidden neuron systems achieving ‘11’ as the highest consecutive correct response.

In order to understand the capability of our precise value representation, we trained the MLP from our selected system independently from the combined CDM for 10 epochs. Fig. 6 shows the pattern of activation for each of the input numbers ‘1’ to ‘20’ associated with this independent network. In contrast to the MLP trained within the CDM, we can see that the numbers ‘1’ to ‘3’ are represented. Also of interest is the way in which the responses for the higher numbers overlap, much like the magnitude representation, with the maximum capability of the network restricted to ‘5’.

To determine the maximum capability of the MLP to represent precise values, we trained the same 2 hidden neuron system for a total of 100 epochs, to understand if the number limitation was a factor of the training regimen. This independent 100 epoch system can successfully detect up to ‘6’, showing that increased training times have little effect on the final response. In contrast, as seen with the results for the CDM with more MLP hidden neurons, larger configuration systems can represent higher numbers. After 10 epochs of training, an MLP with 10 hidden neurons successfully represents up to ‘8’, which increases to ‘12’ after 100 epochs. Each of these results show a bias towards the lower numbers that can be attributed to the training data probability distribution. For

example, with the 10 hidden neuron system trained for 100 epochs on an equal probability distribution, the system is capable of detecting up to '19'.

These results demonstrate that the precise value representation has similar characteristics to that of the magnitude representation, albeit achieved using a supervised learning mechanism. Here, network topology affects the maximum detection capability, much like the SOM. However, the most important factor is the speed at which the representation is learnt. Whilst over a long training period the MLP appears better at representing numbers, it is slower at learning the task than the SOM, which is a consequence of the learning parameters and probability model.

*Output detection:* we can see from Fig. 3 that the magnitude representation contributes most for the responses to '1', '2', '3' and '15', whilst the precise value representation is used for the remaining numbers. The dotted line shows the threshold (0.5) above which the response is considered to be correct, giving a subitization limit of '3'. This coincides with the early learning of the lower numbers by the SOM and the transcoding SLP, and more particularly with the higher allocation of neurons in the SOM consequent from the magnification factor and the probability model. In contrast, the MLP correctly responds to '4', '5' and '8', but can also be seen to be responding to other numbers, despite the lower activity. In particular, it is possible to see a higher activation for the more frequent irregular numbers in the training data, such as '10' and '20'. This allocation of experts appears to be learnt early during training, with the limit of '3' in place after the first epoch, and with no significant changes occurring in subsequent epochs. The winner-takes-all scheme therefore ensures that individual numbers are allocated appropriately early during training, and this has a direct relation to the subitization limit.

In summary then, the two representations in the CDM successfully compete, and the results indicate that under the test conditions, it is easier for the precise value representation to dominate the processing, despite the SOM learning to represent numbers far earlier than the MLP. Here, it appears that the rate at which the transcoding is learnt affects the overall allocation of experts. In order for the SOM to be favoured, the transcoding learning rate must be high enough for the magnitude to out-perform the precise value representation. Whilst in the majority of cases there is just a single expert

for all numbers, in the selected system this allocation is different over a range of numbers. Under these limited conditions, this mechanism can therefore be successfully used to model the seamless integration of the number representations.

#### 4.2 Discussion

Our model of small number detection demonstrates two important attributes. First, the model successfully combines unsupervised and supervised learning in a single architecture. In particular we combine a SOM with a SLP sequentially, and this in parallel with an MLP using the competitive ME architecture. This combination of learning strategies allows us to algorithmically select the best method for learning a particular task, and by varying topology and learning rate parameters, we can determine the conditions under which either strategy is deployed. This approach of using a SOM within a supervised learning system (Casey, 2004) is similar, for example, to the concept adopted by Cameron et al. (1998) for visual navigation. However, whilst our system is built explicitly to model number detection, its generic combination of sequential and parallel supervised and unsupervised learning components using well-known networks may prove beneficial when applied to other domains.

Second, our connectionist model demonstrates how an analogue number input can be mapped to a symbolic output using two different number representations, as simulated with neural networks. One representation forms an ordinal number line of quantities, whilst the other translates directly into a symbolic value. Both are facilitated by the use of the analogue input, assumed to be formed by an accumulator mechanism. The key point being investigated through this computational model is the way in which the proposed two core systems of number are thought to operate, especially on the detection of small numbers as manifest in the subitization limit. With just a single number representation simulated by a neural network, the role of the subitization limit in number detection is unknown and is typically attributed to the number of hidden neurons, or equivalent, and training data (cf. Peterson & Simon, 2000). In contrast, with two neural networks simulating independent representations, the limit has a tangible realisation in the selection of which number representations are used, based upon a performance evaluation of their quantification ability. Furthermore, the performance of our two

selected models depends more on their interaction via the training data and learning speed.

We have built our model using current psychological and psychophysical evidence, coupled with experiences of previous models of number representations, and trained it using data derived from child spoken language sources. Whilst there is evidence both for and against the use of an accumulator mechanism in quantification, we use the accumulator to pre-process the sensory input to obtain an analogue value that can be exploited to learn ordinality, in the same way as Grossberg et al. (2003). Our model of approximate magnitudes uses a spatial representation of numbers to form a compressive number line, demonstrating how a one-dimensional spatial map can be constructed through a process of learning to reproduce characteristics such as the number size and numerical distance effects. This can be seen in the map through the noise associated with higher numbers, and the inherent distance between lower and higher numbers consequent from the one-dimensional organisation (Fig. 4). However, although this spatial model of number is similar to other models (Grossberg & Repin, 2003), it is more difficult to attribute characteristics of either Weber's or Fechner's law to the allocation of numbers to neurons in our model. The model displays some resemblance to the logarithmic scale of performance for monkeys (Nieder & Miller, 2003), with fixed variability (see for example, Feigenson et al., 2004) in at least the first four numbers, and hence shows some of the characteristics of Fechner's law, but the results are not conclusive.

We have used psychological evidence from linguistic processing, coupled with connectionist techniques, to construct our models of transcoding and the precise value representation. Transcoding from magnitudes to precise values is performed using a perceptron architecture using supervised learning. This is used to model the association of magnitudes to symbolic numerals, as perhaps occurs in the interaction between the 'where' and 'what' processing streams (Grossberg & Repin, 2003). In our model, this association is a one-way mapping, but a bi-directional mapping with unsupervised learning is also possible (cf. Ahmad et al., 2002).

For our model of the precise value representation we use an MLP with backpropagation to simulate a symbolic process. However, the pattern of activity that results from the



MLP does partially resemble a linear number line (Fig. 6) with scalar variability (see for example, Feigenson et al., 2004). This is similar to the SOM in that the strong activation for the low numbers results from the probability model used to generate the training data, matching the increase in activity over successive training epochs observed by Peterson and Simon (2000) in their SUBIT-PDP system. Our precise value representation does therefore partially reflect current psychological evidence. We also note that our results for the individual MLP confirm those of Peterson and Simon, in that we can observe a discontinuity in the capability of the network in our selected system, which they labelled as the subitization limit.

Our individual simulations of the approximate magnitude and precise value representations therefore appear to be sufficiently appropriate to allow us to model the integration of both for small number detection. The results from our simulations show that by varying key parameters, we can simulate each of the four possible combinations of the two core systems. However, it is important to note that in the majority case, the precise value system was preferred, and that this is attributable to the speed at which the backpropagation algorithm is capable of learning to represent the lower numbers in the MLP, faster than the transcoding mechanism for the precise values can relate the output of the SOM to the overall output, despite the capability of the SOM to organise the patterns early. Such characteristics suggest we can perhaps be resigned to the abstract nature of this computational model, given that certain values of the parameters can be used to produce all four scenarios. What is important is that the integration of the two representations does give rise to a subitization limit that at least models that observed in humans, with value varying around '3' as a distinct discontinuity. This is in contrast to previous models (Dehaene & Changeux, 1993; Peterson & Simon, 2000; Ahmad et al., 2002) in which the limit is attributed to the form of the input and architectural constraints of a single representation. In our model, the subitization limit results from the interaction of two representations, rather than resulting directly from the architecture of just a single representation. The selection of which expert to use for any given input is driven by the input stimulus alone. Although the actual criteria for the selection of representations requires some notion of time and task, our simulation demonstrates how a simple model can be constructed to automatically select representations based upon performance. As a

result, the emergent crossover point between the deployment of each representation simulates the subitization limit.

## 5 Conclusion

We have shown how a computational model that combines unsupervised with supervised learning can be used to integrate the two core systems of number, simulating the development of the subitization limit as a process of competition between the two representations. Our simulations of the magnitude and precise value representations have been developed using psychological models and psychophysical evidence, supplemented by results from previous simulations. Individually, these have been shown as being capable of representing small numbers, as well as simulating behavioural characteristics, such as the number size and numerical distance effects. Through varying model parameters, we have shown how number detection can be achieved by either the magnitude or precise value representation alone, or through a combination of both. This demonstrates that the proposed two systems of number can be modelled computationally in an integrated system. Resulting from this integration, the subitization limit emerges as a consequence of the competition between the spatial magnitude representation and the linguistically motivated precise value representation. In our example system, the spatial representation specialises in detecting small numbers, and the linguistically motivated representation specialises on the higher numbers. Our model therefore demonstrates one way in which small and large number processing can be integrated to reproduce recognised behavioural characteristics. This suggests one way in which small and large numbers are related in humans.

With advances in our knowledge of the biological basis of the number representations, improved models can be built that help us to explore the fusion of magnitudes with precise values, or indeed the integration of sensory processing for numerical functions and other abilities (cf. Jordan & Brannon, 2006). For our own model, further work is required on exploring other influences on the combination, such as timing constraints, as well as considering the implications of the use of bi-directional connections for

transcoding real values to integers, together with extension to other numerical processing abilities, such as arithmetic.

## Appendix A. Model equations

Given an  $n$ -dimensional input vector  $\mathbf{x} \in R^n$  and an  $m$ -dimensional output vector  $\mathbf{y} \in R^m$ , the CDM can be divided into the operation of each network on  $\mathbf{x}$  within the magnitude representation, precise value representation and output detection stages of the model.

### A.1 Magnitude representation

We can define the operation of the magnitude representation stage as a sequential system that maps the input vector  $\mathbf{x}$  through a sequence two networks, such that:

$$\mathbf{y}_{MA} = f_{SLP} \circ f_{SOM} \circ \mathbf{x} \quad (2)$$

which is the composition of the two functions representing each network (SLP and SOM). Here we select  $f_{SOM}$  as a SOM with weight vectors  $\mathbf{w}_{SOM,j}$  for  $1 \leq j \leq m_{SOM}$ , where  $m_{SOM}$  is the number of neurons in the map. We select  $f_{SLP}$  as an SLP with weight vectors  $\mathbf{w}_{SLP,j}$  for  $1 \leq j \leq m_{SLP}$ , where  $m_{SLP}$  is the number of output neurons, each using the binary threshold activation function  $\varphi_{SLP}$ . The  $i^{\text{th}}$  component of the output vectors  $\mathbf{y}_{SOM} = f(\mathbf{x})$  and  $\mathbf{y}_{SLP} = f(\mathbf{y}_{SOM})$  are:

$$\mathbf{y}_{SOM}^i = \begin{cases} 1 & \text{if } i = \arg \min_j \|\mathbf{x} - \mathbf{w}_{SOM,j}\| \\ 0 & \text{otherwise} \end{cases} \quad 1 \leq i \leq m_{SOM} \quad (3)$$

$$\mathbf{y}_{SLP}^i = \varphi_{SLP}(\mathbf{w}_{SLP,i}^T \cdot \mathbf{y}_{SOM}) \quad 1 \leq i \leq m_{SLP} \quad (4)$$

$$\varphi_{SLP}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (5)$$

Here we have defined the output of the SOM as a single vector that consists of elements representing each of the neurons in the map, such that the winning neuron for an input

pattern, as determined by the Euclidean distance, is assigned the value ‘1’, whilst all other neurons are assigned the value ‘0’. The overall output of this stage is then the thresholded value of the weighted summation of the input to each of the output neurons in the SLP,  $\mathbf{y}_{MA} \equiv \mathbf{y}_{SLP}$ .

### A.2 Precise value representation

The operation of the precise value representation stage uses a single network to map the input vector  $\mathbf{x}$  to the output:

$$\mathbf{y}_{PV} = f_{MLP} \circ \mathbf{x} \quad (6)$$

Here we select  $f_{MLP}$  as an MLP with  $h$  hidden neurons and  $m_{MLP}$  output neurons, with both layers using the logistic sigmoid activation function  $\varphi_{MLP}$ . Each hidden neuron has weight vector  $\mathbf{w}_{MLP,1,j}$  for  $1 \leq j \leq h$ , whilst each output layer neuron has weight vector  $\mathbf{w}_{MLP,2,j}$  for  $1 \leq j \leq m_{MLP}$ . The  $i^{\text{th}}$  component of the hidden and output layer activations are:

$$\mathbf{v}_{MLP,1}^i = \mathbf{w}_{MLP,1,i}^T \cdot \mathbf{x} \quad (7)$$

$$\mathbf{y}_{MLP,1}^i = \varphi_{MLP}(\mathbf{v}_{MLP,1}^i) \quad 1 \leq i \leq h \quad (8)$$

$$\mathbf{v}_{MLP,2}^i = \mathbf{w}_{MLP,2,i}^T \cdot \mathbf{y}_{MLP,1} \quad (9)$$

$$\mathbf{y}_{MLP,2}^i = \varphi_{MLP}(\mathbf{v}_{MLP,2}^i) \quad 1 \leq i \leq m_{MLP} \quad (10)$$

$$\varphi_{MLP}(x) = \frac{1}{1 + e^{-x}} \quad (11)$$

where  $\mathbf{v}_{MLP,j}$  and  $\mathbf{y}_{MLP,j}$  are the weighted summation and output vectors for each layer, respectively. The overall output of this stage is just the output from the MLP,  $\mathbf{y}_{PV} \equiv \mathbf{y}_{MLP,2}$ .

### A.3 Output detection

To combine the output of each representation, we use a gating network as specified in the ME architecture, which maps the input vector  $\mathbf{x}$  to a weight for each of the experts:

$$\mathbf{y}_{GAT} = f_{GAT} \circ \mathbf{x} \quad (12)$$

Here we select  $f_{GAT}$  as an SLP with weight vectors  $\mathbf{w}_{GAT,j}$  for  $1 \leq j \leq m_{GAT}$ , where  $m_{GAT}$  is the number of experts. By using a softmax activation function  $\varphi_{GAT}$ , we ensure that the sum of each the outputs of the gate equals one, so that each equates to an estimate of the probability of the corresponding expert producing the best response to the input. The  $i^{\text{th}}$  component of the output vector  $\mathbf{y}_{GAT}$  is:

$$\mathbf{y}_{GAT}^i = \varphi_{GAT}(\mathbf{w}_{GAT,i}^T \cdot \mathbf{x}) \quad 1 \leq i \leq m_{GAT} \quad (13)$$

$$\varphi_{GAT}(x) = \frac{x}{\sum_j x_j} \quad (14)$$

This enables us to calculate the output of the entire system by weighting each expert, such that the  $i^{\text{th}}$  component of the output vector  $\mathbf{y}$  is:

$$\mathbf{y} = \mathbf{y}_{GAT}^1 \cdot \mathbf{y}_{MA} + \mathbf{y}_{GAT}^2 \cdot \mathbf{y}_{PV} \quad (15)$$

## Appendix B. Learning algorithm

Given a set of  $N$  training input vectors  $X_{TR} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , such that  $\mathbf{x}_j \in R^n \quad \forall j$ , and corresponding target output vectors  $T_{TR} = \{\mathbf{t}_1, \dots, \mathbf{t}_N\}$ , with  $\mathbf{t}_j \in R^m \quad \forall j$ , the model is trained using the algorithm shown in Fig. 7. Details of each of the equations used to calculate the target values and train the networks in each stage of the model are given in the following sections.

### B.1 Magnitude representation

The weights in the SOM are modified according to the standard weight update rule:

$$\mathbf{w}_{SOM,i}(t+1) = \mathbf{w}_{SOM,i}(t) + \eta_{SOM}(t) h_{SOM}^{ci}(t) (\mathbf{x}_j - \mathbf{w}_{SOM,i}(t)) \quad 1 \leq i \leq m_{SOM} \quad (16)$$

The learning rate  $\eta_{SOM}(t)$  and neighbourhood  $h_{SOM}^{ci}(t)$  parameters are decreased exponentially to promote convergence to a stable set of prototype vectors. We use an exponential learning rate and a Gaussian neighbourhood as suggested by Kohonen (1997)

and Haykin (1999), each varying with time  $t$ , where  $c$  is the index of the winning neuron, and  $i$  the neuron whose weight is being adjusted:

$$\eta_{SOM}(t) = \eta_{SOM}(0)e^{-\frac{t}{\tau_1}} \quad (17)$$

$$h_{SOM}^{ci}(t) = e^{-\frac{\|r_c - r_i\|^2}{2\sigma_{SOM}(t)^2}} \quad (18)$$

$$\sigma_{SOM}(t) = \sigma_{SOM}(0)e^{-\frac{t}{\tau_2}} \quad (19)$$

Here,  $\eta_{SOM}(0)$  and  $\sigma_{SOM}(0)$  are the initial learning rate and neighbourhood radius,  $\tau_1$  and  $\tau_2$  are the rate at which the learning rate and neighbourhood decrease, respectively, and  $r_c \in R^2$  and  $r_i \in R^2$  the co-ordinates of the winning and current neurons within the map (which in this instance is one-dimensional).

The SLP is trained using the delta learning rule with constant learning rate  $\eta_{SLP}$  :

$$\delta_{SLP}^i = t_j^i - y_{SLP}^i \quad (20)$$

$$\mathbf{w}_{SLP,i}(t+1) = \mathbf{w}_{SLP,i}(t) + t_{GAT}^1 \eta_{SLP} \delta_{SLP}^i \mathbf{x}_j \quad 1 \leq i \leq m_{SLP} \quad (21)$$

Note that the standard form of the weight update is modified to include the posterior probability factor  $t_{GAT}^1$ , which is determined by the output detection stage, as per the ME learning algorithm (more later). This scaling is not applied to the SOM because it uses unsupervised learning.

## B.2 Precise value representation

The MLP is trained using the backpropagation with momentum learning rule, with constant learning rate  $\eta_{MLP}$  and momentum  $\alpha_{MLP}$  :

$$\delta_{MLP,2}^i = \varphi'_{MLP}(\mathbf{v}_{MLP,2}^i)(t_j^i - y_{MLP}^i) \quad 1 \leq i \leq m_{SLP} \quad (22)$$

$$\delta_{MLP,1}^i = \varphi'_{MLP}(\mathbf{v}_{MLP,1}^i) \sum_{j=1}^{m_{SLP}} \delta_{MLP,2}^j \mathbf{w}_{MLP,2,j} \quad 1 \leq i \leq h \quad (23)$$

$$\mathbf{w}_{MLP,2,i}(t+1) = \mathbf{w}_{MLP,2,i}(t) + \mathbf{t}_{GAT}^2 \eta_{MLP} \delta_{MLP,2}^i \mathbf{y}_{MLP,2}^i \quad 1 \leq i \leq m_{SLP} \quad (24)$$

$$\mathbf{w}_{MLP,1,i}(t+1) = \mathbf{w}_{MLP,1,i}(t) + \mathbf{t}_{GAT}^2 \eta_{MLP} \delta_{MLP,1}^i \mathbf{y}_{MLP,1}^i \quad 1 \leq i \leq h \quad (25)$$

where  $\varphi'_{MLP}$  is the first derivative of the activation function. Once again, the standard form of the weight update is modified to include the posterior probability factor  $\mathbf{t}_{GAT}^2$ , which is determined by the output detection stage.

### B.3 Output detection

The gating network SLP is also trained using the delta learning rule with constant learning rate  $\eta_{GAT}$ :

$$\delta_{GAT}^i = \mathbf{t}_{GAT}^i - \mathbf{y}_{GAT}^i \quad (26)$$

$$\mathbf{w}_{GAT,i}(t+1) = \mathbf{w}_{GAT,i}(t) + \eta_{GAT} \delta_{GAT}^i \mathbf{x}_j \quad 1 \leq i \leq m_{GAT} \quad (27)$$

Here we use the gradient descent, on-line learning scheme for ME (Jordan & Jacobs, 1995), where the target outputs of the gate  $\mathbf{t}_{GAT}$  are calculated as the posterior probabilities of each expert's output being correct, given the current input and overall target output:

$$p_{MA} = \mathbf{y}_{GAT}^1 e^{-\frac{1}{2}(\mathbf{t} - \mathbf{y}_{MA})^T (\mathbf{y}_{GAT} - \mathbf{y}_{MA})} \quad (28)$$

$$p_{PV} = \mathbf{y}_{GAT}^2 e^{-\frac{1}{2}(\mathbf{t} - \mathbf{y}_{PV})^T (\mathbf{y}_{GAT} - \mathbf{y}_{PV})} \quad (29)$$

$$\mathbf{t}_{GAT} = \begin{bmatrix} \frac{p_{MA}}{p_{MA} + p_{PV}} \\ \frac{p_{PV}}{p_{MA} + p_{PV}} \end{bmatrix} \quad (30)$$

These posterior probabilities are also used to weight the learning of each expert, as discussed above. Here we assume that the selection of expert follows a Gaussian probability model.

## References

- Ahmad, K., Casey, M. C., & Bale, T. (2002). Connectionist Simulation of Quantification Skills. *Connection Science, 14*, 165-201.
- Amit, D. J. (1988). Neural Networks Counting Chimes. *Proceedings of the National Academy of Sciences, USA, 85*, 2141-2145.
- Anderson, J. A., Spoehr, K. T., & Bennett, D. J. (1994). A Study in Numerical Perversity: Teaching Arithmetic to a Neural Network. In D.S.Levine & M. Aparicio (Eds.), *Neural Networks for Knowledge Representation and Inference* (pp. 311-335). Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Ansari, D., Garcia, N., Lucas, E., Hamon, K., & Dhital, B. (2005). Neural Correlates of Symbolic Number Processing in Children and Adults. *Brain Imaging, 16*, 1769-1773.
- Brannon, E. M. & Terrace, H. S. (1998). Ordering of the Numerosities 1-9 by Monkeys. *Science, 282*, 746-749.
- Brannon, E. M. & Terrace, H. S. (2002). The Evolution and Ontogeny of Ordinal Numerical Ability. In M.Bekoff, C. Allen, & G. M. Burghardt (Eds.), *The Cognitive Animal* (pp. 197-204). Cambridge, MA.: MIT Press.
- Burnard, L. (1995). British National Corpus: Users Reference Guide for the British National Corpus, Version 1.0. [British National Corpus Consortium]. Oxford, UK., Oxford University Computing Services.
- Cameron, S., Grossberg, S., & Guenther, F. H. (1998). A Self-organizing Neural Network Architecture for Navigation Using Optic Flow. *Neural Computation, 10*, 313-352.
- Carpenter, G. A., Grossberg, S., & Leshner, G. W. (1998). The What-and-Where Filter: A Spatial Mapping Neural Network for Object Recognition and Image Understanding. *Computer Vision and Image Understanding, 69*, 1-22.
- Casey, M. C. (2004). *Integrated Learning in Multi-net Systems*. Unpublished doctoral thesis, University of Surrey, Guildford, UK.
- Casey, M. C. & Ahmad, K. (2004). In-situ Learning in Multi-net Systems. In Z. R. Yang, R. Everson, & H. Yin (Eds.), *Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004), Lecture Notes in Computer Science 3177* (pp. 752-757). Heidelberg: Springer-Verlag.
- Cordes, S., Gelman, R., Gallistel, C. R., & Whalen, J. (2001). Variability Signatures Distinguish Verbal from Nonverbal Counting for Both Large and Small Numbers. *Psychonomic Bulletin & Review, 8*, 698-707.
- Dallaway, R. (1994). *Dynamics of Arithmetic: A Connectionist View of Arithmetic Skills* (Technical Rep. No. 306). Brighton, UK.: University of Sussex.
- Dehaene, S. (1992). Varieties of Numerical Abilities. In *Numerical Cognition* (pp. 1-42). Cambridge, MA.: Blackwell Publishers.



- Dehaene, S. (1997). *The Number Sense: How the Mind Creates Mathematics*. London: Allen Lane, The Penguin Press.
- Dehaene, S. (2003). The Neuronal Basis of the Weber-Fechner Law: A Logarithmic Mental Number Line. *Trends in Cognitive Sciences*, 7, 145-147.
- Dehaene, S. & Changeux, J. P. (1993). Development of Elementary Numerical Abilities: A Neuronal Model. *Journal of Cognitive Neuroscience*, 5, 390-407.
- Dehaene, S., Dehaene-Lambertz, G., & Cohen, L. B. (1998). Abstract Representations of Numbers in the Animal and Human Brain. *Trends in Neurosciences*, 21, 355-349.
- Dehaene, S., Spelke, E., Pinel, P., Stanescu, R., & Tsivkin, S. (1999). Sources of Mathematical Thinking: Behavioral and Brain-Imaging Evidence. *Science*, 284, 970-974.
- Domijan, D. (2004). A Neural Model of Quantity Discrimination. *NeuroReport*, 15, 2077-2081.
- Feigenson, L., Dehaene, S., & Spelke, E. (2004). Core systems of number. *Trends in Cognitive Sciences*, 8, 307-314.
- Fuson, K. C. (1988). *Children's Counting and Concepts of Number*. Berlin, Heidelberg, New York: Springer-Verlag.
- Fuson, K. C., Richards, J., & Briars, D. J. (1982). The Acquisition and Elaboration of the Number Word Sequence. In C.J.Brainerd (Ed.), *Children's Logical and Mathematical Cognition: Progress in Cognitive Development Research* (pp. 33-92). Berlin, Heidelberg, New York: Springer-Verlag.
- Gallistel, C. R. & Gelman, R. (2000). Non-verbal Numerical Cognition: From Reals to Integers. *Trends in Cognitive Sciences*, 4, 59-65.
- Gelman, R. & Butterworth, B. (2005). Number and Language: How Are They Related? *Trends in Cognitive Sciences*, 9, 6-10.
- Gelman, R. & Gallistel, C. R. (1978). *The Child's Understanding of Number*. Cambridge, MA.: Harvard University Press.
- Grossberg, S. & Repin, D. V. (2003). A Neural Model of How the Brain Represents and Compares Multi-digit Numbers: Spatial and Categorical Processes. *Neural Networks*, 16, 1107-1140.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. (2<sup>nd</sup> ed.) Upper Saddle River, NJ.: Prentice-Hall Inc.
- Hoekstra, J. (1992). Counting with Artificial Neural Networks: An Experiment. *Artificial Neural Networks*, 2, 1311-1314.
- Hubbard, E. M., Piazza, M., Pinel, P., & Dehaene, S. (2005). Interactions Between Number and Space in Parietal Cortex. *Nature Reviews Neuroscience*, 6, 435-448.
- Jacobs, R. A., Jordan, M. I., & Barto, A. G. (1991). Task Decomposition through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks. *Cognitive Science*, 15, 219-250.

- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive Mixtures of Local Experts. *Neural Computation*, 3, 79-87.
- Jordan, K. E. & Brannon, E. M. (2006). The Multisensory Representation of Number in Infancy. *Proceedings of the National Academy of Sciences, USA*, 103, 3486-3489.
- Jordan, M. I. & Jacobs, R. A. (1995). Modular and Hierarchical Learning Systems. In M.A.Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks* (pp. 579-582). Cambridge, MA.: MIT Press.
- Kaufman, E. L., Lord, M. W., Reese, T. W., & Volkman, J. (1949). The Discrimination of Visual Number. *American Journal of Psychology*, 62, 498-525.
- Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43, 59-69.
- Kohonen, T. (1997). *Self-Organizing Maps*. (2nd ed.) Berlin, Heidelberg, New York: Springer-Verlag.
- Ma, Q. & Hirai, Y. (1989). Modeling the Acquisition of Counting with an Associative Network. *Biological Cybernetics*, 61, 271-278.
- MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk*. (Third ed.) Mahwah, NJ.: Lawrence Erlbaum Associates.
- Mandler, G. & Shebo, B. J. (1982). Subitizing: An Analysis of its Component Processes. *Journal of Experimental Psychology: General*, 111, 1-22.
- McCloskey, M. (1992). Cognitive Mechanisms in Numerical Processing and Calculation: Evidence from Acquired Dyscalculia. In S.Dehaene (Ed.), *Numerical Cognition* (pp. 107-157). Cambridge, MA.: Blackwell Publishers.
- McCloskey, M., Caramazza, A., & Basili, A. (1985). Cognitive Mechanisms in Number Processing and Calculation: Evidence from Dyscalculia. *Brain and Cognition*, 4, 171-196.
- McCloskey, M. & Lindemann, A. M. (1992). MATHNET: Preliminary Results from a Distributed Model of Arithmetic Fact Retrieval. In J.I.D.Campbell (Ed.), *The Nature and Origins of Mathematical Skills* (pp. 365-409). North Holland: Elsevier Science Publishers B-V.
- Meck, W. H. & Church, R. M. (1983). A Mode Control Model of Counting and Timing Processes. *Journal of Experimental Psychology: Animal Behavior Processes*, 9, 320-334.
- Nieder, A., Freedman, D. J., & Miller, E. K. (2002). Representation of the Quantity of Visual Items in the Primate Prefrontal Cortex. *Science*, 297, 1708-1709.
- Nieder, A. & Miller, E. K. (2003). Coding of Cognitive Magnitude: Compressed Scaling of Numerical Information in the Primate Prefrontal Cortex. *Neuron*, 37, 149-157.
- Orlov, T., Amit, D. J., Yakovlev, V., Zohary, E., & Hochstein, S. (2006). Memory of Ordinal Number Categories in Macaque Monkeys. *Journal of Cognitive Neuroscience*, 18, 399-417.

- Peterson, S. A. & Simon, T. J. (2000). Computational Evidence for the Subitizing Phenomenon as an Emergent Property of the Human Cognitive Architecture. *Cognitive Science*, 24, 93-122.
- Piazza, M. & Dehaene, S. (2004). From Number Neurons to Mental Arithmetic: The Cognitive Neuroscience of Number Sense. In M. Gazzaniga (Ed.), *The Cognitive Neurosciences III* (3rd ed., pp. 865-876). Cambridge, MA.: MIT Press.
- Rodriguez, P., Wiles, J., & Elman, J. L. (1999). A Recurrent Neural Network that Learns to Count. *Connection Science*, 11, 5-40.
- 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* (pp. 318-362). Cambridge, MA.: MIT Press.
- Simon, O., Mangin, J. F., Cohen, L., Le Bihan, D., & Dehaene, S. (2002). Topographical Layout of Hand, Eye, Calculation, and Language-Related Areas in the Human Parietal Lobe. *Neuron*, 33, 475-487.
- Spelke, E. & Dehaene, S. (1999). Biological Foundations of Numerical Thinking. *Trends in Cognitive Sciences*, 3, 365-366.
- Stanescu-Cosson, R., Pinel, P., van de Moortele, P. F., Le Bihan, D., Cohen, L., & Dehaene, S. (2000). Understanding Dissociations in Dyscalculia: A Brain Imaging Study of the Impact of Number Size on the Cerebral Networks for Exact and Approximate Calculation. *Brain*, 123, 2240-2255.
- Thompson, R. F., Mayers, K. S., Robertson, R. T., & Patterson, C. J. (1970). Number Coding in Association Cortex of the Cat. *Science*, 168, 271-273.
- Trick, L. M. & Pylyshyn, Z. W. (1994). Why Are Small and Large Numbers Enumerated Differently? A Limited-Capacity Preattentive Stage in Vision. *Psychological Review*, 101, 80-102.
- Varley, R. A., Klessinger, N. J. C., Romanowski, C. A. J., & Siegal, M. (2005). Agrammatic But Numerate. *Proceedings of the National Academy of Sciences, USA*, 102, 3519-3524.
- Walsh, V. (2003). A Theory of Magnitude: Common Cortical Metrics of Time, Space and Quantity. *Trends in Cognitive Sciences*, 7, 483-488.
- Wells, C. G. (1981). *Learning Through Interaction: The Study of Language Development*. Cambridge, UK.: Cambridge University Press.
- Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Unpublished doctoral thesis, Harvard University, Cambridge, MA.
- Widrow, B. & Hoff, M. E. Jr. (1960). Adaptive Switching Circuits. In *IRE WESCON Convention Record* (pp. 96-104).
- Wynn, K. (1998). Psychological Foundations of Number: Numerical Competence in Human Infants. *Trends in Cognitive Sciences*, 2, 296-303.

## Figure Captions

Fig. 1. Functional diagram of the Competitive Detection Model. Each input consists of a pre-processed quantity represented as an analogue value. This is fed to both the approximate magnitude representation stage and the precise value representation stage. The output from each representation is combined using a Gaussian probability model trained to allocate processing to each expert based upon the input stimulus alone. The output of the system is the weighted combination of each expert's output.

Fig. 2. Comparison of the target CHILDES probability distribution and the sample training data distribution. Probabilities have been normalised for the numbers '1' to '20'. Note the high probability of low numbers, which rapidly decreases with the slight exception of the variation for the irregular number words 'ten' and 'twenty'.

Fig. 3. Allocation of magnitude or precise value expert representations to the output of the CDM with  $M=40$ ,  $N=2$  and  $L=0.3$  trained for 10 epochs, for each of the numbers '1' to '20'. The output of the model for each number is the weighted activation of each expert. For the numbers '1', '2', '3' and '15', the magnitude representation contributes most to the output, showing a subitization limit of '3', with '15' represented well within the SOM because of the boundary effect (see text). For the remaining numbers, the precise value representation contributes the most to the output, with activation value decreasing with increasing number. With a threshold of 0.5, shown by the dotted line, only six of the twenty numbers are correctly detected.

Fig. 4. Allocation of numbers to neurons within a one-dimensional SOM with 40 neurons trained on the CHILDES probability data set for 10 epochs. Allocation is shown as the inverse normalised Euclidean distance (see text) from a given input quantity to each neuron in the map. Numbers are ordered with increasing value from '1' on the left, hence modelling a number line. From '6' to '20' the peak allocation of numbers to neurons overlaps, demonstrating a limit in the capability of the representation. Compression is demonstrated with '1' associated with 5 neurons, '2' with 9, '3' with 9, '4' with 7 and '5'

with 3, noting the reduced number of neurons allocated to '1', and the high response for '15', resulting from the boundary effect in SOM learning.

Fig. 5. Allocation of numbers to neurons within a one-dimensional SOM with 40 neurons trained on the equal probability data set for 10 epochs. Allocation is shown as the inverse normalised Euclidean distance (see text) from a given input quantity to each neuron in the map. Numbers are ordered with increasing from '1' on the left, again modelling a number line, but with uniform distribution of numbers to neurons.

Fig. 6. Activation of the precise value MLP with 2 hidden layer neurons trained independently from the CDM on the CHILDES probability data set for 10 epochs. Each of the inputs from '1' to '20' is shown as a line representing the activation for each of the 20 output neurons. With a threshold of 0.5, shown by the dotted line, only '1', '2', '3' and '5' are correctly detected, as labelled.

Fig. 7. Training algorithm for the CDM.

**Figures**

Fig. 1.

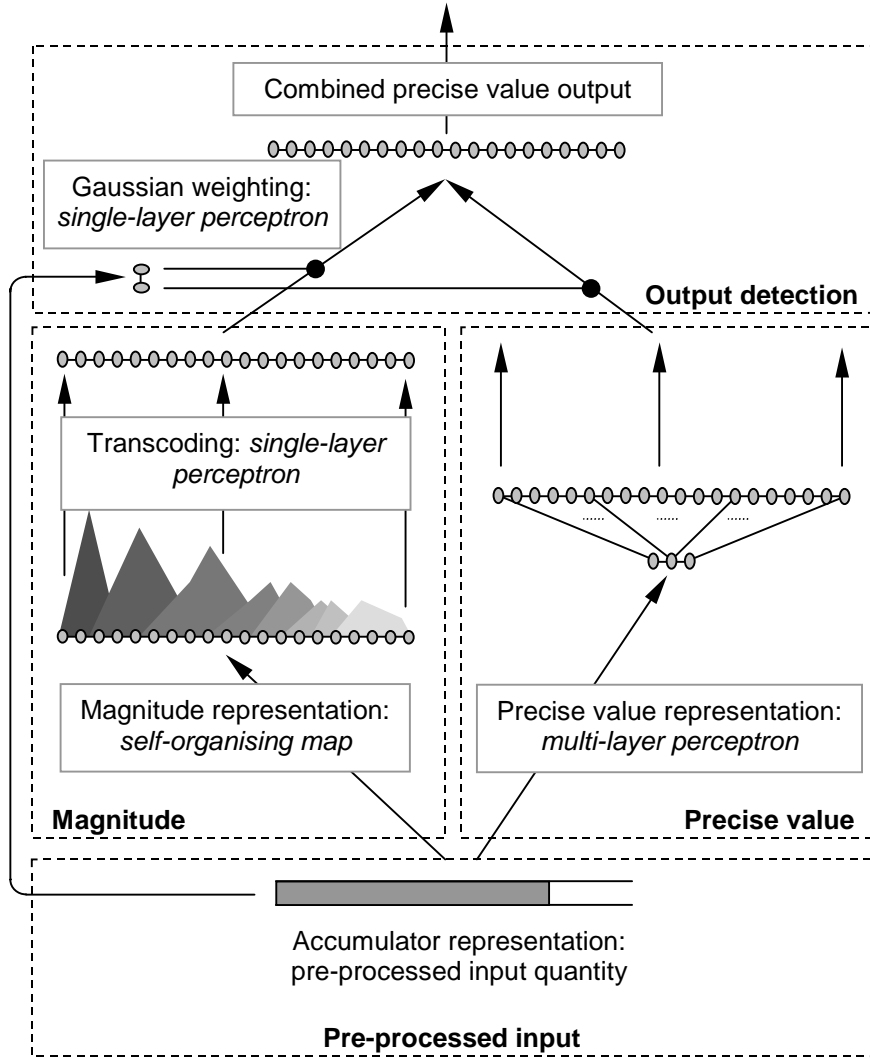


Fig. 2.

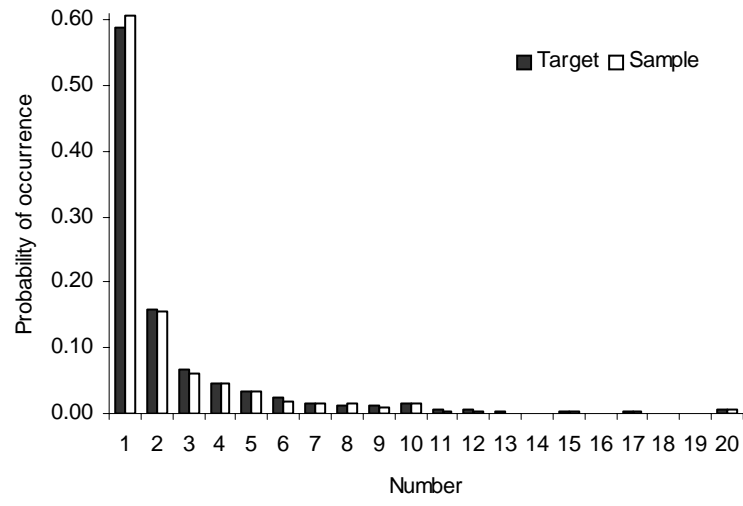


Fig. 3.

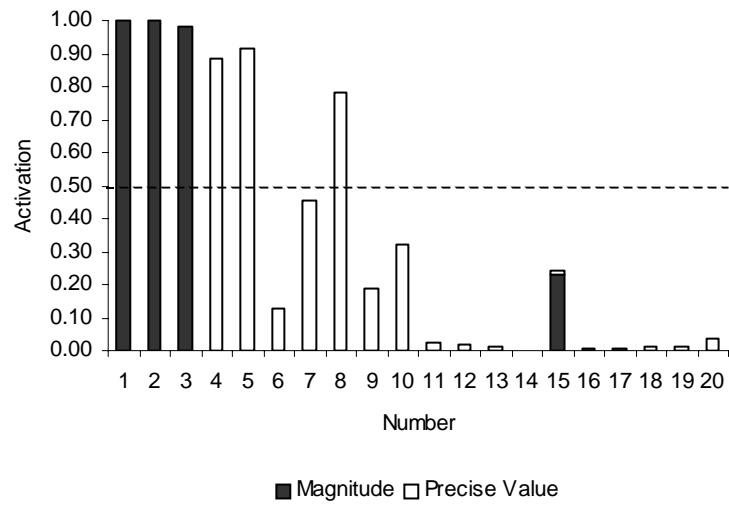




Fig. 4.

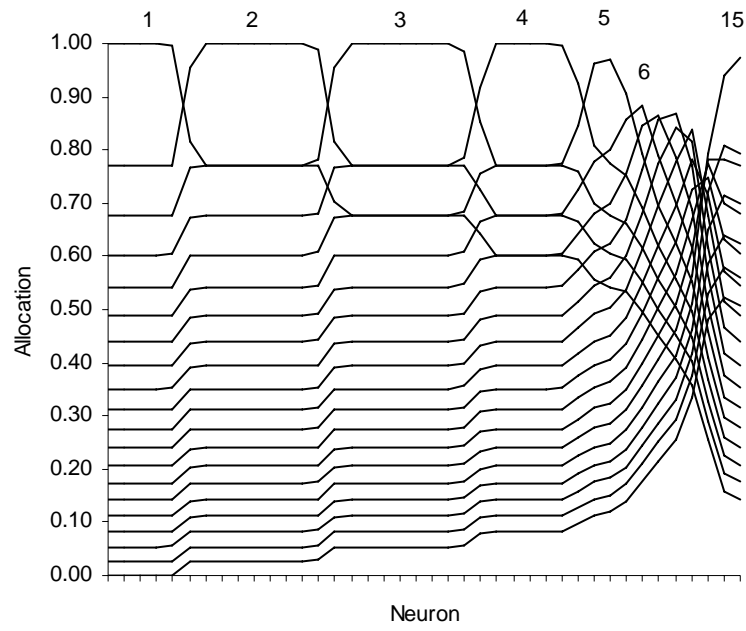


Fig. 5.

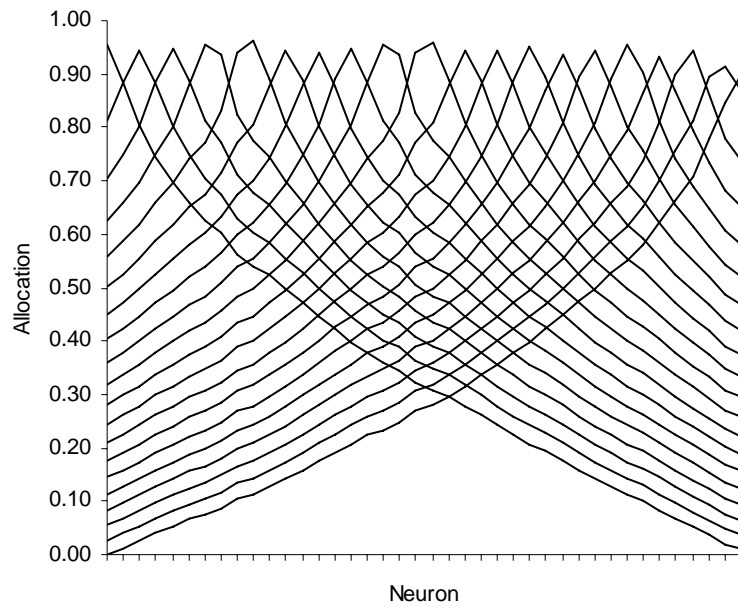


Fig. 6.

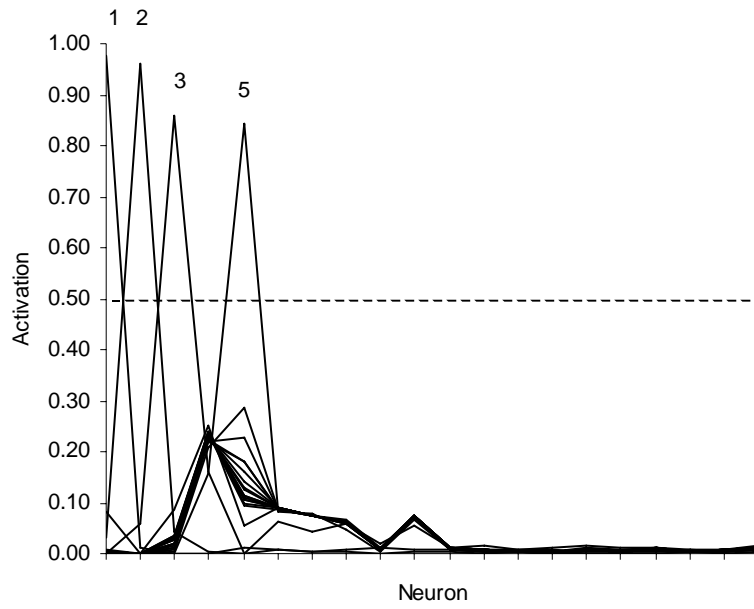


Fig. 7.

```
set  $t = 1$   
for each training epoch do  
  for each training pattern  $j = 1$  to  $N$  do  
    set  $y_{MA} = f_{SLP} \circ f_{SOM} \circ \mathbf{x}_j$   
    set  $y_{PV} = f_{MLP} \circ \mathbf{x}_j$   
    set  $y_{GAT} = f_{GAT} \circ \mathbf{x}_j$   
    set  $\mathbf{t}_{GAT}$  using  $\mathbf{t}_j$  and  $y_{GAT}$   
    train  $f_{GAT}$  using  $\mathbf{x}_j$  and  $\mathbf{t}_{GAT}$   
    train  $f_{MLP}$  using  $\mathbf{x}_j$ ,  $\mathbf{t}_j$  and  $\mathbf{t}_{GAT}$   
    train  $f_{SLP}$  using  $\mathbf{x}_j$ ,  $\mathbf{t}_j$  and  $\mathbf{t}_{GAT}$   
    train  $f_{SOM}$  using  $\mathbf{x}_j$   
    set  $t = t + 1$   
  end do  
end do
```

## Table Captions

Table 1: Example detection and number representation simulations. Previous multi-net simulations are highlighted by the grouping of their constituent networks.

Table 2: Example inputs for the quantities ‘1’ to ‘20’. Note that each number overlaps with the previous as an analogue accumulated value.

Table 3: Experimental parameters for the CDM, showing network topology and learning parameters used. Topology is shown as the input dimension and number of neurons in each layer of the network; for the SOM, the number of neurons consists of the two-dimensional map size. Learning parameters are shown as either varying for each time step  $t$ , or constant, and depend upon the algorithms defined (Appendix B). The three parameters varied during training are shown as  $L$ ,  $M$  and  $N$ : for the transcoding learning rate, SOM map size and number of hidden neurons in the precise value MLP, respectively.

## Tables

Table 1.

| Task                                   | System           | Function                              | References                        |
|--|------------------|---------------------------------------|-----------------------------------|
| Subitization                           | Multi-net {      | McCulloch-Pitts                       | Input clusters                    |
|  |                  | Difference-of-Gaussians               | Object location and normalisation |
|  |                  | McCulloch-Pitts                       | Summation clusters                |
|  |                  | McCulloch-Pitts                       | Numerosity clusters               |
| Subitization                           | ACT-R            | Recognition and counting              | (Peterson & Simon, 2000)          |
|  | Backpropagation  | Visual subitization                   |                                   |
| Subitization                           | Multi-net {      | Second Order                          | Scale invariance                  |
|  |                  | Weight Sharing                        | Translational invariance          |
|  |                  | Kohonen SOM                           | Magnitude representation          |
|  |                  | Hebbian                               | Bi-directional linkage            |
|  |                  | Kohonen SOM                           | Verbal representation             |
| Subitization                           | Winner-takes-all | Object normalisation and subitization | (Domijan, 2004)                   |
| Counting Procedure                     | Multi-net {      | Hopfield                              | Identify stimulus                 |
|  |                  | Delayed Synapses                      | Counting of abstract stimuli      |
| Counting Procedure                     | Time-Delay       | Counting without memory               | (Hoekstra, 1992)                  |
| Counting Procedure                     | BPTT             | Counting with memory                  | (Rodriguez et al., 1999)          |
| Number Word Sequence Production        | Multi-net {      | Heteroassociative                     | Number word sequence storage      |
|  |                  | Inhibitory Recurrent                  | Number word production            |
| Number Word and Next Object Production | Multi-net {      | Second Order                          | Scale invariance                  |
|  |                  | Mixture-of-experts                    | Word and next object tasks        |
|  |                  | Madaline                              | Cardinal response                 |
| Number comparison                      | Multi-net {      | Neurons with Kernels                  | Spatial number map                |
|  |                  | Directional Transient Cells           | Comparison wave                   |



Table 3.

| Network                      |     |             | Parameters     |  |
|------------------------------|-----|-------------|----------------|--|
| Magnitude representation     | SOM | 20: $M$ x1  | Learning rate: | $\eta_{SOM}(0) = \frac{1}{2}, \tau_1 = \frac{1}{4000}$                         |
|                              |     |             | Neighbourhood: | $\sigma_{SOM}(0) = \frac{1}{2}N, \tau_2 = \frac{1}{20000} \ln \sigma_{SOM}(0)$ |
| Transcoding                  | SLP | $M:20$      | Learning rate: | $\eta_{SLP} = L$   |
| Precise value representation | MLP | 20: $N$ :20 | Learning rate: | $\eta_{MLP} = 0.1$   |
|                              |     |             | Momentum:      | $\alpha_{MLP} = 0.9$   |
| Gating network               | SLP | 20:2        | Learning rate: | $\eta_{GAT} = 0.1$   |