Parameter definitions and specifications for population modelling simulations of English past-tense acquisition

Michael S. C. Thomas

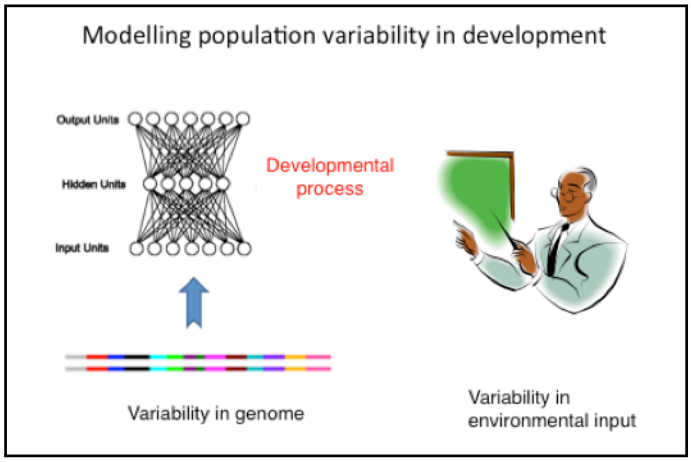
0. Introduction

This document contains technical details to accompany several papers that employ population modelling of the acquisition of English past tense formation. These simulations are either directly applied to capturing empirical data of children learning the past tense (Thomas, Ronald & Forrester, submitted; Thomas & Knowland, in preparation), or employ the past tense domain as an abstract learning problem within cognition (Thomas, Knowland & Karmiloff-Smith, 2011a, 2011b; Thomas, Forrester, & Ronald, in preparation a, in preparation b).

These simulations explore a population of networks acquiring the past tense domain, where individual variability is included both in the parameters of the artificial neural networks which model the children's learning systems, and the learning environment to which they are exposed. The parameters of the artificial neural networks are encoded in an artificial genome. Population variability in parameters is created by generating and breeding populations of artificial genomes. Each genome is realised as a parameterised network. The network is exposed to an individualised learning environment, generating a trajectory of behavioural development. The inclusion of an artificial genome level in the simulations allows two avenues of exploration. First, it allows us to study the associations that can arise between values on the artificial genome and behavioural variability that is the product of an implemented developmental process (see Figure 2). Second, it allows us to generate

individuals with different levels of genetic similarity, such as parents, siblings, monozygotic (identical) twins, and dizygotic (fraternal twins); siblings can be exposed to a shared family learning environment, unique learning events, or a combination of the two. Measurements of the similarity of behaviour between related individuals then permit the simulation of behaviour genetic designs, such as twin studies. The simulations therefore provide a framework to study heritability of behaviour within a developmental framework.

Figure 1: Schematic of the population modelling simulations



This document describes the computational parameters that varied in the artificial neural networks. It outlines how the range of variation for each parameter in the population was established. It then describes the method for designing the artificial genome, and the assumptions that this method embodies. The details of the breeding process are described. Finally, a set of lookup tables is included detailing how values on the artificial genome were mapped to computational parameter values in the artificial neural networks. Two closely related models are considered, both of which are trained to output the phonological form of the past tense of the English verb; one

model is given the phonological form of the verb stem as input, the second model additionally is given lexical-semantic information about the verb as input. For each of these models, two sets of tables is included, one that specifies a wide range of computational parameter variation in the population, a second that specifies a relatively narrow range of variation. By combining these with training sets that also can vary widely or narrowly in quality, the relative contributions of internal versus external influences on individual differences in behaviour can be assessed.

Section 1 outlines the parameters and artificial genome for the phonology-to-phonology past tense network. Section 2 outlines the equivalent parameters and genome for the phonology-and-semantics-to-phonology past tense network. Section 3 contains a brief note about the simplifications contained in the design of the artificial genome, and the extent to which it corresponds to a 'blueprint' (or otherwise).

The technical specifications are provided to support these papers:

Thomas, M. S. C., Knowland, V. C. P., & Karmiloff-Smith, A. (2011a). Mechanisms of developmental regression in autism and the broader phenotype: A neural network modeling approach. *Psychological Review*, *118(4)*, 637-654.

Thomas, M. S. C., Knowland, V. C. P., & Karmiloff-Smith, A. (2011b). Variability in the severity of developmental disorders: A neurocomputational account of developmental regression in autism. In: E. Davelaar (Ed.), *Proceedings of the 12th Neurocomputational and Psychology Workshop*, (p. 309-325). World Scientific.

Thomas, M. S. C., Ronald, A., & Forrester, N. A. (submitted). *Modelling socio-economic status effects on language development*. Manuscript submitted for publication.

Thomas, M. S. C., Forrester, N. A., & Ronald, A. (in preparation, a). *Modelling associations between levels of description: What can gene-behaviour associations tell us about cognitive process?* Manuscript in preparation.

Thomas, M. S. C. & Knowland, V. C. P. (in preparation). *Modelling mechanisms of persisting and resolving delay in language development.* Manuscript in preparation

Thomas, M. S. C., Forrester, N. A. & Ronald, A. (in preparation, b). *A simulated twin study exploring the heritability of past tense acquisition in a population of neural network models.* Manuscript in preparation.

If you have any questions on the contents of this document, please contact Michael Thomas (m.thomas@bbk.ac.uk).

Copies of papers may be downloaded from the homepage of the Developmental Neurocognition Laboratory (http://www.psyc.bbk.ac.uk/research/DNL/).

1. The phonology-to-phonology architecture

1.1 Model architecture and parameters

The original connectionist model employed a three-layer artificial neural network, comprising an input layer, a layer of internal or 'hidden' units, and an output layer. It was trained using the backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986), a type of supervised learning. The free parameters in the model were the number of hidden units, the learning rate, and the momentum (see below). An expanded set of 14 parameters was employed in the current simulations, in many cases to allow for additional analogues to known neurocomputational properties. However, backpropagation itself is not viewed as biologically plausible. We use it here in place of a more biologically plausible error-correction algorithm (see Thomas & McClelland, 2008, for discussion). An introduction to the idea that parameters in connectionist models can explain types of cognitive variability can be found in Thomas and Karmiloff-Smith, 2002a). The parameters and model architecture are depicted schematically in Figure 2. The parameters were as follows:

Building the network:

- *Architecture*: In addition to the 3-layer network, a 2-layer network without a layer of hidden units, and a fully connected network were used. A 2-layer network has less computational power than a 3-layer network but learns more quickly. A fully connected network contains both direct connections from input to output and a hidden layer, and produces a computationally more powerful system. Networks could therefore have 1, 2, or 3 layers of connection weights. Previous connectionist models have proposed single or multiple pathways may be available to connect input and output (e.g., Westermann, 1998; Zorzi, Houghton & Butterworth, 1998),

and that differential use of routes may explain individual differences in behaviour (Harm & Seidenberg, 2004; Plaut, 1997; Thomas & Karmiloff-Smith, 2002b). Recent functional brain imaging of reading lend some support to this proposal (e.g., Richardson et al., 2011; Seghier et al., 2008).

- *Hidden units*: For networks with a hidden unit layer, the number of hidden units could vary. Variations of the number of hidden units have been proposed to account for developmental deficits such as dyslexia (e.g., Harm & Seidenberg, 1999) and autism (e.g., Cohen, 1998), as well as individual differences (Richardson et al., 2006a, b). We did not vary the number of hidden layers. More hidden units within a layer increases computational power and the rate of learning, while more layers of hidden units increases computational power but slows down learning, since error must be propagated from the output more deeply into the network to improve learning (see Richardson et al., 2006a,b, for a comparison of these conditions).
- *Sparseness*: The architecture determined how many layers of connection weights existed. Of the potential connections in a layer, only a certain proportion was created. The sparseness parameter set the probability that any given connection would be created. Greater connectivity increases computational power, but can lead to slower learning. Under some conditions, it can also lead to poorer generalisation, since greater integration of information causes more item-specific and context-specific learning (see McClelland, 2000, for a proposal that conjunctive coding may cause autistic symptoms, and conversely, Beversdorf, Narayanan & Hughes, 2007, for a proposal that the symptoms arise from sparse connectivity).
- *Weight variance*: Connection weights were assigned an initial random value within a range depending on this parameter. E.g., if set to 0.5, weights would be randomised between +/- 0.5. Large initial weights take time to unlearn, which slows

learning (an effect known as entrenchment; see Munakata & McClelland, 2003, for discussion).

Processing dynamics:

- *Processing noise*: The net activation a receiving unit receives from a given sending unit is a product of the sending unit's activation and the connection strength between them. Transmission noise was added to this net activation. Gaussian noise was used and the parameter specified the standard deviation of the noise distribution around zero. Noise has been used to simulate under-specified representations in development (e.g., to simulate Specific Language Impairment: Joanisse & Seidenberg, 2003; or as a candidate explanation of autistic symptoms: Simmons et al., 2007), and has also been proposed as an essential primitive in neural processing (McClelland, 1993).
- *Unit threshold function*: A receiving unit sums the net activation from all sending units and uses an activation function to determine its consequent output. We used a common non-linear activation function, the sigmoid or logistic function, equivalent to a smoothed threshold. This function has a parameter, the temperature, which makes the smoothed threshold either steeper or shallower. The activation function was:

$$Output = \frac{1}{1 + e^{-temperature \times (netinput + bias)}}$$

where *netinput* is the summed activation to a unit, *bias* is the negative of the unit's threshold, and *Output* is the unit's activation state in response to this input. A shallow function (low temperature) denies a unit the opportunity to make large output changes in response to small changes in net input, whereas a steep function (high temperature) approximates a non-smoothed threshold, thereby producing a

unit with binary response characteristics. Variations in the slope of the sigmoid function have been proposed as candidate explanations of disorders such as specific language impairment (Thomas, 2005) and schizophrenia (Cohen & Servan-Schreiber, 1992), as well as ageing (Li & Lindenberger, 1999). Changes to the slope of the sigmoid have a number of effects on learning. A shallow slope means that processing units are less sensitive to small differences in their input. This poor discriminability means they will be slow to learn categorisations that rely on small distinctions in the input. Secondly, in the backpropagation algorithm, weight update for a given error signal is proportional to the slope on the sigmoid (the differential of the function). If the function resembles a gentle S-shape, then the slope across the range of unit activations will be small. A shallow sigmoid will lead directly to slower learning. Conversely, if the temperature is very high, producing a sigmoid similar to a step function, for most inputs to a unit, it will be jammed on or off ('saturated') rather than in its dynamic range. When a unit is saturated, the slope on the sigmoid function is flatter (the regions below or above the step). When it is in its dynamic range it is steep (the step). If a unit is predominantly saturated due to a high temperature, the flat slope will again lead to small weight changes for a given error signal and therefore slow learning. Finally, units with high temperatures flip between being saturated on or off. They are therefore ill suited to learning mappings requiring graduations of activation states. In sum, temperatures that are either too high or too low can delay learning.

Network maintenance:

- *Connection weight decay*: each connection's magnitude was reduced by a small proportion on each presentation of a training pattern, according to the weight decay

parameter. The approximate range of weight decay values was derived by estimating a percentage of weight value that could plausibly be lost overall all of training (e.g., 50%), and then dividing this proportion by the number of training epochs (e.g., 1000) and the number of training patterns presented on each epoch (e.g., 508), to give a proportional reduction in the connection weights to be applied on each pattern presentation (e.g., $0.5/1000/508=9.84 \times 10^{-7}$). To my knowledge, weight decay has not been used as a candidate mechanism to explain individual variability.

- We did not simulate the increase in synaptic density observed in human cortex during infancy and early childhood; we did, however, implemented the pruning of spare resources from mid-childhood (Huttenlocher, 2002). The *pruning process* eliminated small connection weights. Variations in pruning have been proposed as an explanation of autistic symptoms, and specifically developmental regression (Thomas, Knowland & Karmiloff-Smith, 2011). The pruning process involved three parameters: onset, threshold, and probability:
- *Connection pruning – onset*: Connections that were not being used were probabilistically pruned away after a certain point in training. The onset parameter determined the point in training when pruning began (see Thomas & Johnson, 2006, for simulations of pruning applied to sensitive periods in plasticity).
- *Connection pruning – threshold*: Connections stood a chance of being pruned after onset only if their magnitude fell below a threshold determined by this parameter. The rationale is that small weights are assumed not to transmit strong activations and therefore not to be playing a key role in computations. They may therefore be removed to save on resources.

- *Connection pruning – probability*: If the magnitude of a connection fell below threshold after pruning had begun, it was eliminated probabilistically based on this parameter. High probability leads to faster loss of unused connections. Low probability leads to slower loss.

Network adaptation:

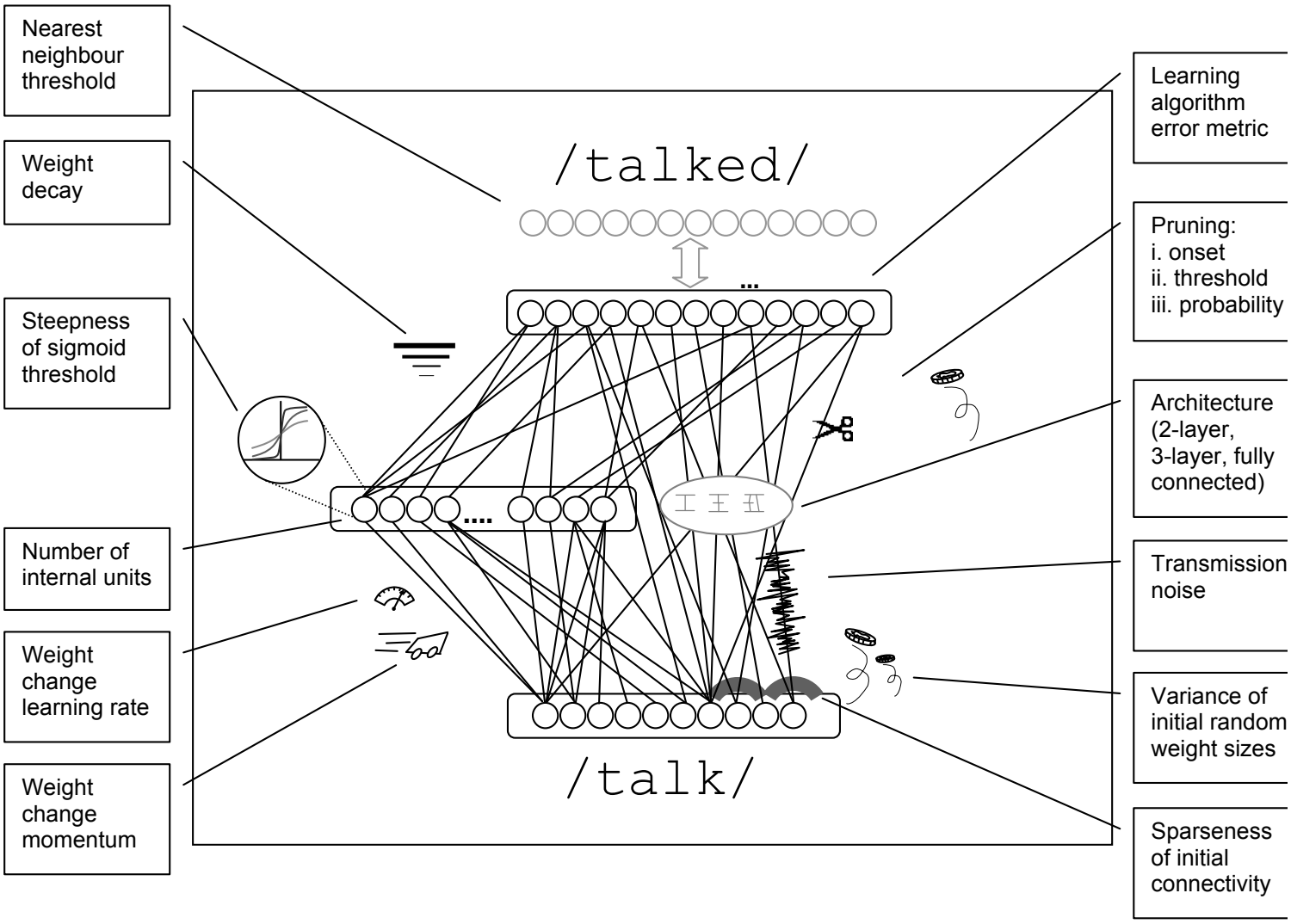
- *Learning algorithm error measure*: The backpropagation algorithm was used with two different metrics to determine the error signal marking the disparity between the network's current output and its intended target. These were Euclidean distance and cross-entropy (Hinton, 1989). The Euclidean distance metric produces less weight change for a unit when it is committed to an erroneous response than the cross-entropy measure. That is, when a unit is stuck on in a saturated state but the learning algorithm requires it to be off, or vice versa, cross-entropy will lead to faster changes to its weights to change its activation state than Euclidean distance. Under some conditions, cross-entropy can therefore be a more plastic learning algorithm, leading to faster learning and higher ceiling performance.
- *Learning rate*: This parameter determined how much the connection weights were altered in response to a certain disparity between output and target during supervised learning. A large learning rate produces a system that learns more quickly but that also may be unstable, flipping between good performance on different parts of the problem domain. Differences in learning rate have been proposed as explanations of individual differences in cognitive ability (Richardson et al., 2006a,b) and general intelligence (Garlick, 2002), as well as developmental deficits (e.g., dyslexia; Harm & Seidenberg, 1999).

- *Momentum*: This parameter allowed some proportion of the weight change on the previous learning trial to be carried over. It serves a smoothing function to prevent learning from getting stuck in local, sub-optimal solutions. While a parameter often varied in connectionist models of development, it has not to my knowledge been used as a candidate explanation for individual differences in learning.

Network response:

- *Nearest neighbour threshold*: Network output comprised a vector of continuous activation values between 0 and 1, while legal responses of the network were binary vectors. An algorithm determined which legal phoneme was closest to the activation patterns at onset, nucleus, and coda. However, the phoneme was only recognised as a response if the activation was sufficiently close to the legal phoneme (using a root mean square or RMS measure). This was determined by the nearest neighbour threshold. (The legal phonemes could of course still be the incorrect ones for the target verb). The nearest neighbour computation may be viewed as equivalent to the settling of an unimplemented recurrent attractor network into a particular response state (see Plaut et al., 1996, for a model of reading development in which this attractor network was implemented). The nearest neighbour threshold parameter then indexes the efficiency of this attractor network to generate a response within some notional deadline. A high threshold allows an approximate output to be recognised as correct (i.e., larger error is tolerated); a low threshold requires a more exact initial output. The use of a nearest neighbour algorithm allowed the network to generate accuracy levels. Differences in the functioning of the attractor network (sometimes called 'clean-up' units) have been proposed as a candidate explanation of developmental deficits (e.g., dyslexia; Harm & & Seidenberg, 1999).

Figure 2: Architecture of the connectionist model of English past-tense acquisition, showing the internal parameters that varied in the population.



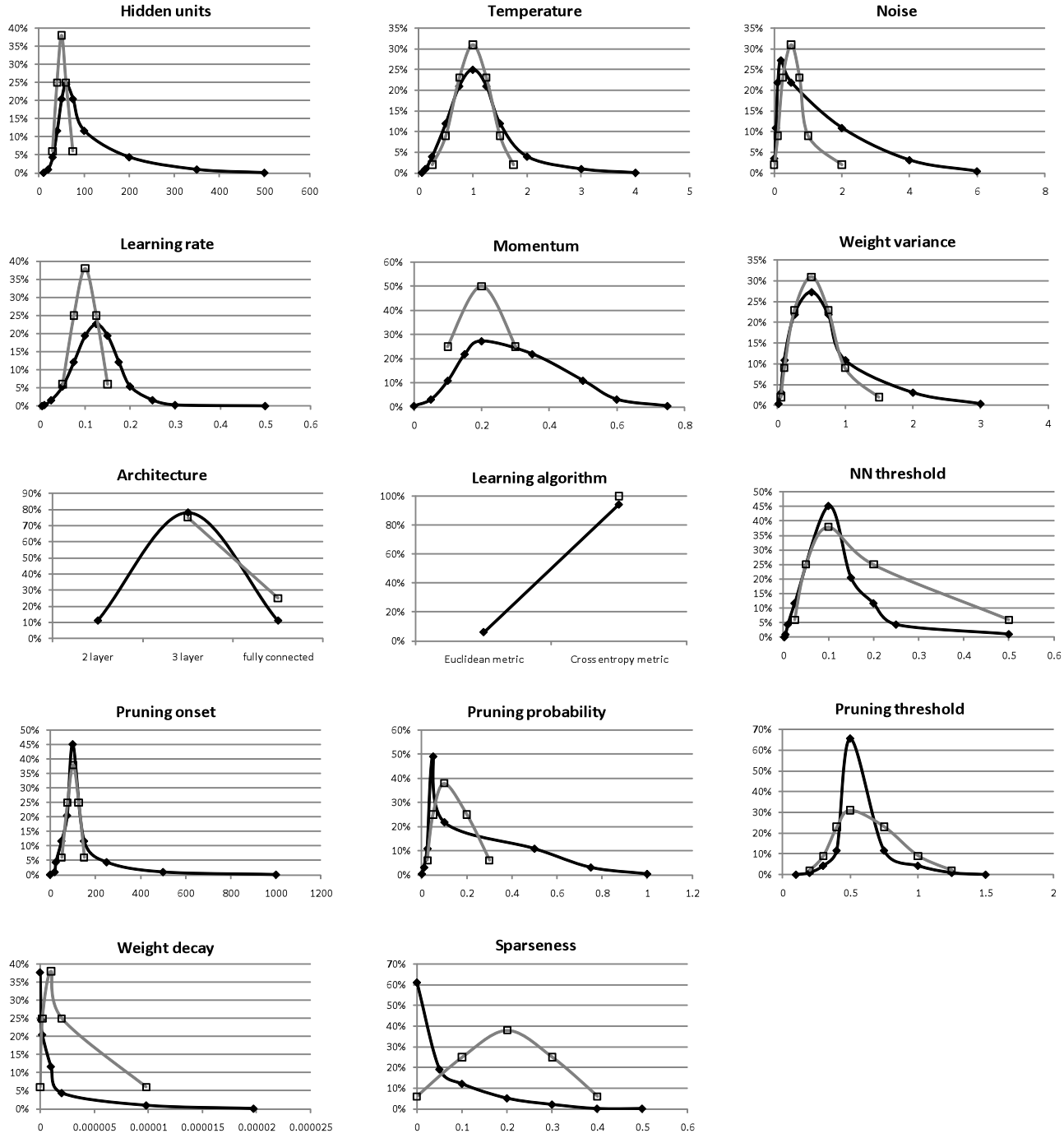
1.2 Calibrating parametric variation

Two of these parameters were categorical: the architecture and learning algorithm metric. The others were continuously valued. In order to produce variability in the population according to these remaining parameters, they were calibrated them as follows. An initial 'normal' set of parameters was defined. These were estimated based on previous research. Each of the continuously valued parameters was then varied in turn, holding the all other parameters at their initial values. For each parameter, the range was derived that produced failure of learning up to highly successful learning. In some cases, parameters had a monotonic relationship to performance (e.g., hidden units, where more was better); in other cases, there was an optimal intermediate value (e.g., activation function). The aim was to determine an average or adequate value for each parameter, which was defined heuristically as 'just enough to succeed and then a little bit more'. Values were then derived that would cause increasingly poorer or increasingly better performance around this value. We attempted to make poorer and better performance roughly symmetrical around average performance for each parameter. This caused some parameter ranges to be skewed. For example, 50 hidden units was determined as the average value in a 3-layer network. Values of 40 or 30 would cause poorer performance. However, to achieve equivalent differences above average level, 100 or 200 hidden units might be necessary. We chose to emphasise behavioural symmetry around the average parameter value rather than parametric symmetry, on the grounds that the symmetrical bell curve is a common pattern observed in human abilities. The ranges for each parameter for the phonology-to-phonology network are included in Figure 3.

We chose not to vary the input and output coding scheme. Our previous work suggests that, within certain limits, varying the problem encoding has similar effects on the developmental trajectory to altering computational parameters (Thomas & Karmiloff-Smith, 2003). However, recoding the problem domain can in principle have extreme effects on learnability, if key distinctions in the input or output are lost in the recoding. Some models of developmental language impairment and dyslexia propose that differences in the representation of phonology cause subsequent behavioural deficits in grammar and reading acquisition (e.g., Harm & Seidenberg, 1999; Hoeffner & McClelland, 1993; Joanisse, 2004).

Although only main effects of each parameter were considered as sources of variability during calibration, we expected interactions between these neurocomputational parameters in subsequent learning. To pick four examples: (i) large numbers of hidden units can partially compensate for a shallow sigmoid function in those processing units; (ii) having a more sparse initial connectivity is likely to reduce the amount of weights eliminated via pruning because their magnitudes will be larger; (iii) high weight decay can be countered by a higher learning rate; (iv) an over-aggressive pruning process (e.g., with a high threshold and high probability) can be alleviated if its onset occurs very late in training when weights have become large, but exacerbated if the onset is early. Large numbers of parameter combinations were possible within our scheme: given the number of levels specified for each parameter, approximately two thousand billion unique parameter combinations were available.

Figure 3: Parameter values and target population frequencies



Parameter values (x-axis) and their target frequencies in the population (y-axis) for the wide-genetic (black) and narrow-genetic (grey) variation conditions, for each of the 14 computational parameters. Each gene had two alleles, coded as binary values. Several genes coded for each parameter value. Sets of binary values were summed and a look-

up table used to derive each parameter value. The numbers of binary alleles for each parameter were as follows. *G-Wide* = hidden units: 10; temperature: 10; noise: 8; learning rate: 12; momentum: 8; weight variance: 8; architecture: 6; learning algorithm: 4; nearest neighbour threshold: 10; pruning onset epoch: 10; pruning probability: 8; pruning threshold: 10; weight decay: 10; sparseness: 12 (total 126 bits). *G-Narrow* = hidden units: 4; temperature: 6; noise: 6; learning rate: 4; momentum: 2; weight variance: 6; architecture: 2; learning algorithm: 4; nearest neighbour threshold: 4; pruning onset epoch: 4; pruning probability: 4; pruning threshold: 6; weight decay: 4; sparseness: 4 (total 60 bits).

1.3 Specifying an artificial genome for the model and the mechanism for breeding

The use of genetic algorithms entails creation of an artificial genome to encode the neural network's parameter values, such that all possible genomes correspond to legal parameter sets. In creating the genome, we made the following assumptions:

- There were two copies of each gene, with genes residing on pairs of chromosomes.
- For simplicity, each gene had only two variants or alleles.
- The two alleles produced different outcomes in the functionality of the neurocomputational parameter which they encoded.
- The influence of genes was intended to be *additive*: we did not include dominant or recessive effects, and genes had the same effect in combination as in isolation. This constraint was motivated by the finding within behavioural genetics that the effect of gene variants is predominantly additive on phenotypic outcomes (Plomin et al., 2008). Nevertheless, our method of implementing the mapping between gene variants and neurocomputational parameters did turn out to inadvertently include non-additive effects (see Section 4).
- All neurocomputational parameters were *polygenic*. That is, their value was determined by the additive action of a collection of genes.
- In the first instance, we assumed that the action of genes was not *pleiotropic*; that is, with respect to neurocomputational parameters, we assumed that no gene affected the value of more than one parameter at once. This simplification likely will not hold in many cases, and certainly the current theoretical view is

that the relationship between genes and *cognitive processes* is pleiotropic (see, e.g., Kovas & Plomin, 2006).

The assumption of polygenicity was motivated by the fact that we are using computational models to capture cognitive-level phenomena, and is a point worth emphasising. We expect many low-level neural variations to influence neurocomputational functions at the level of cognitive processes in neural circuits. We therefore view it as unlikely that a single gene would modulate a neurocomputational parameter responsible for normal cognitive variation.

By way of illustration, the following list gives some examples (from Sapolsky, 2005) of the low-level variations one might expect. At the level of individual neurons, conservatively, one might expect variation between individuals in the number of dendritic spines, the number of axon terminals, the level of resting potentials, the size of the dendritic wavelet caused by pre-synaptic activity, the excitability of the axon hillock, and the speed of propagation of the axon potential. At the level of two neurons communicating, one might expect individual variations in the amounts of neurotransmitter released, the numbers of receptors, the efficiency of receptors in binding neurotransmitters, the efficiency of producing neurotransmitters, the efficiency of producing receptors, and the proportions of different types of receptors. At the level of long-term potentiation, one might expect variation between individuals in how much glutamate neurotransmitter is released, the number of glutamate receptors, the ratio of glutamate receptor types, the level of calcium ion release, and the level of phosphoration of the receptors. It is likely that a range of gene variants contribute to each of these neural parameters. Our higher-level models encode much coarser neurocomputational parameters such as “level of processing noise” or “learning rate”,

which would correspond to the combined effect of many of the more detailed neural properties.

We assumed that the combination of alleles for each polygenic neurocomputational parameter had a deterministic relation to the value of that parameter in the instantiated network: that is, the allele set alone determined the parameter value. We assumed (and did not instantiate) a much larger part of the genome that was species universal and coded for the basics of, for example, creating the processing units, the connections, the activation dynamics, the sensorium, the input-output connectivity pathways, and the mechanics of experience-dependent systems.

Turning to mechanisms of breeding, we assumed that there was sexual reproduction, so that each gene had a 50% chance of being passed on to a gamete (egg or sperm), which combined with a gamete of another individual to create a new offspring. Although reproduction was sexual in this sense, we did not consider sex effects in these simulations (i.e., there were no genetic differences between males and females). During breeding, we assumed that there was uniform crossover and no linkage disequilibrium, the latter falling beyond the scope of our simulations. That is, the presence or absence of a given allele in a gamete was independent of the presence or absence of any others. This assumption is violated in humans because genes on the same chromosome have a greater than 50% chance of being transferred into a gamete together, and the closer they lie on a chromosome, the higher the chance.

When Genetic Algorithms are used for machine learning optimisation, the most successful individuals of the previous generation are often retained in the next generation. In our case, after breeding, the previous generation died. Breeding enabled the creation of individuals with different degrees of relatedness, for instance as twins

or siblings. For some conditions, we created identical (monozygotic; MZ) or fraternal (dizygotic; DZ) twin pairs. MZ twins shared the same genome, while DZ twins and siblings were created by generating two offspring from the same set of parents, but from a different sperm and egg. DZ twins and siblings shared 50% of their alleles on average. Also in contrast to the more common use of Genetic Algorithms, we did not include genetic mutation during reproduction. In humans, the mutation rate is extremely low (e.g., Strachan & Read, 2003, cite a rate of between 1 and 4 mutations per 100,000 genes per generation). Mutations serve to reduce the average genetic similarity of siblings below 50%, and our preference was to maintain retain the 50% value, since it is the one deployed in standard behavioural genetic models. Several other aspects fell beyond the scope of the simulations. We did not model the effects of epistasis (interactions between genes) or epigenetic effects on gene expression; we did not model assortative mating – in our simulations, mates were selected at random from the population; and we did not model gene-environment correlations (Plomin et al., 2008) – variation in the composition of the environment had no correlation with the nature of an individual's genotype.

1.4 Parameter values and their link to the artificial genome for the phonology-to-phonology network

For the phonology-to-phonology network, the total of number of genes used to encode the value of the 14 computational parameters was 126 (or two copies of 63) as follows – hidden units: 10; temperature: 10; noise: 8; learning rate: 12; momentum: 8; weight variance: 8; architecture: 6; learning algorithm: 4; nearest neighbour threshold: 10; pruning onset epoch: 10; pruning probability: 8; pruning threshold: 10; weight decay: 10; sparseness: 12 (total 126 bits).

Figure 3 plots the range of values for each parameter against their target frequency of occurrence in the population. The translation of a genome into a parameter set was implemented by assigning alleles the value of 1 or 0, and then deriving the total for all the genes influencing the parameter (thereby ensuring additivity). The parameter value was calculated from the total using a lookup table, created by hand for each parameter to reflect the range of values identified during the calibration stage. The lookup tables for the 14 parameters (in the Wide Genetic) condition are shown below.

Table 1. Lookup table linking the artificial genome to the Hidden Unit parameter, for the Wide Genetic Variation condition

	Hidden Unit Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.01	0.04	0.12	0.21	0.25	0.21	0.12	0.04	0.01	0.001
Parameter value	10	20	30	40	50	60	75	100	200	350	500

Table 2. Lookup table linking the artificial genome to the Temperature parameter, for the Wide Genetic Variation condition

	Temperature Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.01	0.04	0.12	0.21	0.25	0.21	0.12	0.04	0.01	0.001
Parameter value	0.0625	0.125	0.25	0.5	0.75	1	1.25	1.5	2	3	4

Table 3. Lookup table linking the artificial genome to the Noise parameter, for the Wide Genetic Variation condition

	Noise Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8

Population probability	-	0.04	0.11	0.22	0.27	0.22	0.11	0.03	0.00
Parameter value	0	0	0.05	0.1	0.2	0.5	2	4	6

Table 4. Lookup table linking the artificial genome to the Learning Rate parameter, for the Wide Genetic Variation condition

	Learning Rate Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	0.0002	0.0029	0.02	0.05	0.12	0.19	0.23	0.19	0.12	0.05	0.02	0.0029	0.0002
Parameter value	0.005	0.01	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.25	0.3	0.5

Table 5. Lookup table linking the artificial genome to the Momentum parameter, for the Wide Genetic Variation condition

	Momentum Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8
Population probability	0.004	0.03	0.11	0.22	0.27	0.22	0.11	0.03	0.004
Parameter value	0	0.05	0.1	0.15	0.2	0.35	0.5	0.6	0.75

Table 6. Lookup table linking the artificial genome to the Weight Variation parameter, for the Wide Genetic Variation condition

	Weight Variation Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8
Population probability	0.004	0.03	0.11	0.22	0.27	0.22	0.11	0.03	0.004
Parameter value	0.01	0.05	0.1	0.25	0.5	0.75	1	2	3

Table 7. Lookup table linking the artificial genome to the Architecture parameter, for the Wide Genetic Variation condition. (0 = 2-layer, 1 = 3-layer, 2 = fully-connected)

	Architecture Parameter Value						
Number of 1-	0	1	2	3	4	5	6

valued alleles							
Population probability	-	0.109	-	0.781	-	0.109	-
Parameter value	0	0	1	1	1	2	2

Table 8. Lookup table linking the artificial genome to the Learning Algorithm parameter, for the Wide Genetic Variation condition. (0 = Euclidean distance error metric, 1 = cross-entropy error metric)

	Learning Algorithm Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.063	0.938	-	-	-
Parameter value	0	1	1	1	1

Table 9. Lookup table linking the artificial genome to the Nearest Neighbour Threshold parameter, for the Wide Genetic Variation condition

	Nearest Neighbour Threshold Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.010	0.044	0.117	0.451	-	0.205	0.117	0.044	0.011	-
Parameter value	0.0025	0.005	0.01	0.025	0.1	0.1	0.15	0.2	0.25	0.5	0.5

Table 10. Lookup table linking the artificial genome to the Pruning Onset parameter, for the Wide Genetic Variation condition

	Pruning Onset Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.01	0.04	0.12	-	0.45	0.21	0.12	0.04	0.01	0.001
Parameter value	1000	500	250	150	100	100	75	50	25	20	0

Table 11. Lookup table linking the artificial genome to the Pruning Probability parameter, for the Wide Genetic Variation condition

	Pruning Probability Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8
Population probability	0.004	0.03	0.11	-	0.49	0.22	0.11	0.03	0.004
Parameter value	0	0.01	0.025	0.05	0.05	0.1	0.5	0.75	1

Table 12. Lookup table linking the artificial genome to the Pruning Threshold parameter, for the Wide Genetic Variation condition

	Pruning Threshold Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability (%)	0.001	0.01	0.04	0.12	-	0.66	-	0.12	0.04	0.01	0.001
Parameter value	0.1	0.2	0.3	0.4	0.5	0.5	0.5	0.75	1	1.25	1.5

Table 13. Lookup table linking the artificial genome to the Weight Decay parameter, for the Wide Genetic Variation condition

	Weight Decay Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	-	-	-	-	0.38	0.25	0.21	0.12	0.04	0.01	0.001
Parameter value	0	0	0	0	0	$1x10^{-7}$	$2x10^{-7}$	$9.8x10^{-7}$	$19.7x10^{-7}$	$98.4x10^{-7}$	$196.9x10^{-7}$

Table 14. Lookup table linking the artificial genome to the Sparseness parameter, for the Wide Genetic Variation condition

	Sparseness Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	-	-	-	-	-	-	0.61	0.19	0.12	0.05	0.02	0.003	0.0002
Parameter value	0	0	0	0	0	0	0	0.05	0.1	0.2	0.3	0.4	0.5

A more constricted range of genetic variation was also considered for each computational parameter in the phonology-to-phonology network, shown in Figure 3 in grey. This required fewer genes to encode, leading to a genome with only 60 genes (2 copies of 30). For one of the parameters, learning algorithm, there were only had two values in the original formulation; we restricted the range of variation by fixing the parameter to use cross-entropy, thus removing variation in this gene. On average, the parameters of the narrow condition had 40% of the range of variation of the wide condition. The lookup tables for the Narrow Genetic condition are shown below.

Table 15. Lookup table linking the artificial genome to the Hidden Unit parameter, for the Narrow Genetic Variation condition

	Hidden Unit Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	30	40	50	60	75

Table 16. Lookup table linking the artificial genome to the Temperature parameter, for the Narrow Genetic Variation condition

	Temperature Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.25	0.5	0.75	1	1.25	1.5	1.75

Table 17. Lookup table linking the artificial genome to the Noise parameter, for the Narrow Genetic Variation condition

	Temperature Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6

Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0	0.1	0.25	0.5	0.75	1	2

Table 18. Lookup table linking the artificial genome to the Learning Rate parameter, for the Narrow Genetic Variation condition

	Learning Rate Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	0.05	0.075	0.1	0.125	0.15

Table 19. Lookup table linking the artificial genome to the Momentum parameter, for the Narrow Genetic Variation condition

	Momentum Parameter Value		
Number of 1-valued alleles	0	1	2
Population probability	0.25	0.50	0.25
Parameter value	0.1	0.2	0.3

Table 20. Lookup table linking the artificial genome to the Weight Variation parameter, for the Narrow Genetic Variation condition

	Weight Variation Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.05	0.1	0.25	0.5	0.75	1	1.5

Table 21. Lookup table linking the artificial genome to the Architecture parameter, for the Narrow Genetic Variation condition. (0 = 2-layer, 1 = 3-layer, 2 = fully-connected)

	Architecture Parameter Value		
Number of 1-valued alleles	0	1	2
Population	-	0.75	0.25

probability			
Parameter value	1	1	2

Table 22. Lookup table linking the artificial genome to the Learning Algorithm parameter, for the Narrow Genetic Variation condition. (0 = Euclidean distance error metric, 1 = cross-entropy error metric)

	Learning Algorithm Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	-	-	1.00	-	-
Parameter value	1	1	1	1	1

Table 23. Lookup table linking the artificial genome to the Nearest Neighbour Threshold parameter, for the Narrow Genetic Variation condition

	Nearest Neighbour Threshold Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	0.025	0.05	0.1	0.2	0.5

Table 24. Lookup table linking the artificial genome to the Pruning Onset parameter, for the Narrow Genetic Variation condition

	Pruning Onset Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	50	75	100	125	150

Table 25. Lookup table linking the artificial genome to the Pruning Probability parameter, for the Narrow Genetic Variation condition

	Pruning Probability Parameter Value

Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	0.025	0.05	0.1	0.2	0.3

Table 26. Lookup table linking the artificial genome to the Pruning Threshold parameter, for the Narrow Genetic Variation condition

	Pruning Threshold Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.2	0.3	0.4	0.5	0.75	1	1.25

Table 27. Lookup table linking the artificial genome to the Weight Decay parameter, for the Narrow Genetic Variation condition

	Weight Decay Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	98.0×10^{-7}	20.0×10^{-7}	10.0×10^{-7}	2.0×10^{-7}	0

Table 28. Lookup table linking the artificial genome to the Sparseness parameter, for the Narrow Genetic Variation condition

	Sparseness Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	0.4	0.3	0.2	0.1	0

During our simulations, we considered two other conditions of genetic variation for the phonology-to-phonology network. These were used to assess the power of association analyses to predict behavioural variability from the values of the genes. In

the first populations, allele values of 1 and 0 had equal frequency. In these latter populations, the frequency of 1 and 0 alleles was unbalanced. In one population, alleles had a value of 1 with a probability of 70% and 0 with a probability of 30%. In the second, alleles had a value of 1 with a probability of 30% and a value of 0 with a probability of 70%.

2. The phonology-and-semantics-to-phonology architecture

2.1 Introduction

The simple phonology-to-phonology network of Plunkett & Marchman (1991, 1993) was altered in later work to include the influence of lexical semantic information at input (Joanisse & Seidenberg, 1999; Karaminis & Thomas, 2010; Thomas & Karmiloff-Smith, 2003; Woollams, Joanisse & Patterson, 2009). We also generated populations with parametric variability for an architecture including lexical-semantic information at input, which we refer to as the phonology-and-semantics-to-phonology architecture.

Two addition parameters were created for this architecture. Prior work has demonstrated that the phonological input is more influential in the learning of regular verbs, and particularly the extension of the regular rule to novel verbs, while the lexical semantic input is more influential in the learning of exception verbs (Joanisse & Seidenberg, 1999; Thomas & Karmiloff-Smith, 2003; Thomas, 2005). We added two parameters that would allow these respective influences to vary across the population. The two parameters served to modulate the learning rate on the connections from each input type. The Phonological Learning Rate was a value between 0 and 100% that modulated the learning rate in the connections from the phonological input units, whether to the hidden units or direct to the phonological output layer. The Semantic Learning Rate was a value between 0 and 100% that modulated the learning rate in the connections from the lexical semantic input units, whether to the hidden units or direct to the phonological output layer.

2.2 Parameter values and their link to the artificial genome for the phonology-and-semantics-to-phonology network

Calibration for a Wide range of genetic variation yielded an artificial genome with 156 genes (2 copies of 78) as follows: hidden units: 12; temperature: 12; noise: 10; learning rate: 12; phonological learning rate: 12; semantic learning rate: 12; momentum: 8; weight variance: 10; architecture: 8; learning algorithm: 4; nearest neighbour threshold: 6; pruning onset epoch: 10; pruning probability: 8; pruning threshold: 10; weight decay: 12; sparseness: 10 (total 156 bits). The lookup tables for the mapping between artificial genome and parameters were as follows.

Table 29. Lookup table linking the artificial genome to the Hidden Unit parameter, for the Wide Genetic Variation condition

	Hidden Unit Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	0.0002	0.003	0.02	0.05	0.12	0.19	0.23	0.19	0.12	0.05	0.02	0.003	0.0002
Parameter value	6	8	10	15	20	22	25	30	40	60	100	200	500

Table 30. Lookup table linking the artificial genome to the Temperature parameter, for the Wide Genetic Variation condition

	Temperature Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	0.0002	0.003	0.02	0.05	0.12	0.19	0.23	0.19	0.12	0.05	0.02	0.003	0.0002
Parameter value	0.0625	0.125	0.25	0.5	0.75	1	1.25	1.5	1.75	2	2.5	3	4

Table 31. Lookup table linking the artificial genome to the Noise parameter, for the Wide Genetic Variation condition

	Noise Parameter Value

Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.01	0.04	0.12	0.21	0.25	0.21	0.12	0.04	0.01	0.001
Parameter value	0	0.05	0.1	0.2	0.25	0.5	0.75	1	2	4	5

Table 32. Lookup table linking the artificial genome to the Learning Rate parameter, for the Wide Genetic Variation condition

	Learning Rate Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	0.0002	0.0029	0.02	0.05	0.12	0.19	0.23	0.19	0.12	0.05	0.02	0.0029	0.0002
Parameter value	0.005	0.01	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.25	0.3	0.5

Table 33. Lookup table linking the artificial genome to the Phonological Learning Rate parameter, for the Wide Genetic Variation condition

	Phonological Learning Rate Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	-	-	-	0.07	0.12	0.19	0.23	0.19	0.12	0.05	0.02	-	-
Parameter value	1	1	1	1	0.75	0.5	0.25	0.1	0.05	0.01	0	0	0

Table 34. Lookup table linking the artificial genome to the Semantic Learning Rate parameter, for the Wide Genetic Variation condition

	Semantic Learning Rate Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	-	-	-	-	0.19	0.19	0.23	0.19	0.12	0.05	0.02	0.003	-
Parameter value	1	1	1	1	1	0.75	0.5	0.25	0.1	0.05	0.01	0	0

Table 35. Lookup table linking the artificial genome to the Momentum parameter, for the Wide Genetic Variation condition

	Momentum Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8
Population probability	0.004	0.03	0.11	0.22	0.27	0.22	0.11	0.03	0.004
Parameter value	0	0.05	0.1	0.15	0.2	0.35	0.5	0.6	0.75

Table 36. Lookup table linking the artificial genome to the Weight Variation parameter, for the Wide Genetic Variation condition

	Weight Variation Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.01	0.04	0.12	0.21	0.25	0.21	0.12	0.04	0.01	0.001
Parameter value	0.005	0.01	0.05	0.1	0.25	0.5	0.75	1	1.5	2	2.25

Table 37. Lookup table linking the artificial genome to the Architecture parameter, for the Wide Genetic Variation condition. (0 = 2-layer, 1 = 3-layer, 2 = fully-connected)

	Architecture Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8
Population probability	-	-	0.14	-	0.71	-	0.14	-	-
Parameter value	0	0	0	1	1	1	2	2	2

Table 38. Lookup table linking the artificial genome to the Learning Algorithm parameter, for the Wide Genetic Variation condition. (0 = Euclidean distance error metric, 1 = cross-entropy error metric)

	Learning Algorithm Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.063	0.938	-	-	-
Parameter value	0	1	1	1	1

Table 39. Lookup table linking the artificial genome to the Nearest Neighbour Threshold parameter, for the Wide Genetic Variation condition

	Nearest Neighbour Threshold Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.005	0.01	0.025	0.05	0.1	0.2	0.5

Table 40. Lookup table linking the artificial genome to the Pruning Onset parameter, for the Wide Genetic Variation condition

	Pruning Onset Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability	0.001	0.01	0.04	0.12	-	0.45	0.21	0.12	0.04	0.01	-
Parameter value	1000	500	250	150	100	100	75	50	25	0	0

Table 41. Lookup table linking the artificial genome to the Pruning Probability parameter, for the Wide Genetic Variation condition

	Pruning Probability Parameter Value								
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8
Population probability	0.004	0.03	0.11	-	0.49	0.22	0.11	0.03	0.004
Parameter value	0	0.01	0.025	0.05	0.05	0.1	0.5	0.75	1

Table 42. Lookup table linking the artificial genome to the Pruning Threshold parameter, for the Wide Genetic Variation condition

	Pruning Threshold Parameter Value										
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10
Population probability (%)	0.001	0.01	0.04	0.12	-	0.66	-	0.12	0.04	0.01	0.001
Parameter value	0.1	0.2	0.3	0.4	0.5	0.5	0.5	0.75	1	1.25	1.5

Table 43. Lookup table linking the artificial genome to the Weight Decay parameter, for the Wide Genetic Variation condition

	Weight Decay Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	0.001	0.01	0.04	0.12	0.21	0.45	-	0.12	0.06	-	-	-	-
Parameter value	984.3×10^{-7}	196.9×10^{-7}	98.4×10^{-7}	19.7×10^{-7}	9.8×10^{-7}	2×10^{-7}	2×10^{-7}	1×10^{-7}	0	0	0	0	0

Table 44. Lookup table linking the artificial genome to the Sparseness parameter, for the Wide Genetic Variation condition

	Sparseness Parameter Value												
Number of 1-valued alleles	0	1	2	3	4	5	6	7	8	9	10	11	12
Population probability	-	-	-	-	-	0.62	0.21	0.12	0.04	0.01	0.001	0.003	0.0002
Parameter value	0	0	0	0	0	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7

Finally, a condition for Narrow Variation was created for the phonology-and-semantics-to-phonology network. This yielded an artificial genome with 80 genes (2 copies of 40) as follows: hidden units: 4; temperature: 6; noise: 6; learning rate: 6; phonological learning rate: 6; semantic learning rate: 4; momentum: 4; weight variance: 6; architecture: 2; learning algorithm: 4; nearest neighbour threshold: 6; pruning onset epoch: 4; pruning probability: 6; pruning threshold: 6; weight decay: 6; sparseness: 4 (total 80 bits). The lookup tables for the mapping between artificial genome and parameters were as follows.

Table 45. Lookup table linking the artificial genome to the Hidden Unit parameter, for the Narrow Genetic Variation condition

	Hidden Unit Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	15	20	25	30	35

Table 46. Lookup table linking the artificial genome to the Temperature parameter, for the Narrow Genetic Variation condition

	Temperature Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.25	0.5	0.75	1	1.25	1.5	1.75

Table 47. Lookup table linking the artificial genome to the Noise parameter, for the Narrow Genetic Variation condition

	Temperature Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0	0.05	0.1	0.2	0.25	0.5	0.75

Table 48. Lookup table linking the artificial genome to the Learning Rate parameter, for the Narrow Genetic Variation condition

	Learning Rate Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.05	0.075	0.1	0.125	0.15	0.175	0.2

Table 49. Lookup table linking the artificial genome to the Phonological Learning Rate parameter, for the Narrow Genetic Variation condition

	Phonological Learning Rate Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.11	-
Parameter value	0.05	0.1	0.25	0.5	0.75	1	1

Table 50. Lookup table linking the artificial genome to the Semantic Learning Rate parameter, for the Narrow Genetic Variation condition

	Semantic Learning Rate Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.063	0.625	-	0.250	0.063
Parameter value	0.25	0.5	0.5	0.75	1

Table 51. Lookup table linking the artificial genome to the Momentum parameter, for the Narrow Genetic Variation condition

	Momentum Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.0625	0.25	0.375	0.25	0.0625
Parameter value	0	0.1	0.15	0.2	0.25

Table 52. Lookup table linking the artificial genome to the Weight Variation parameter, for the Narrow Genetic Variation condition

	Weight Variation Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.01	0.05	0.1	0.25	0.5	0.75	1

Table 53. Lookup table linking the artificial genome to the Architecture parameter, for the Narrow Genetic Variation condition. (0 = 2-layer, 1 = 3-layer, 2 = fully-connected)

	Architecture Parameter Value		
Number of 1-valued alleles	0	1	2
Population probability	-	0.75	0.25
Parameter value	1	1	2

Table 54. Lookup table linking the artificial genome to the Learning Algorithm parameter, for the Narrow Genetic Variation condition. (0 = Euclidean distance error metric, 1 = cross-entropy error metric)

	Learning Algorithm Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	-	-	1.00	-	-
Parameter value	1	1	1	1	1

Table 55. Lookup table linking the artificial genome to the Nearest Neighbour Threshold parameter, for the Narrow Genetic Variation condition

	Nearest Neighbour Threshold Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06
Parameter value	0.01	0.04	0.07	0.1	0.13

Table 56. Lookup table linking the artificial genome to the Pruning Onset parameter, for the Narrow Genetic Variation condition

	Pruning Onset Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.06	0.25	0.38	0.25	0.06

Parameter value	50	75	100	150	200

Table 57. Lookup table linking the artificial genome to the Pruning Probability parameter, for the Narrow Genetic Variation condition

	Pruning Probability Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.01	0.025	0.05	0.1	0.15	0.2	0.3

Table 58. Lookup table linking the artificial genome to the Pruning Threshold parameter, for the Narrow Genetic Variation condition

	Pruning Threshold Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.31	0.23	0.09	0.02
Parameter value	0.1	0.2	0.3	0.4	0.5	0.6	0.75

Table 59. Lookup table linking the artificial genome to the Weight Decay parameter, for the Narrow Genetic Variation condition

	Weight Decay Parameter Value						
Number of 1-valued alleles	0	1	2	3	4	5	6
Population probability	0.02	0.09	0.23	0.55	-	0.11	-
Parameter value	98.4×10^{-7}	19.7×10^{-7}	9.8×10^{-7}	1.0×10^{-7}	1.0×10^{-7}	0	0

Table 60. Lookup table linking the artificial genome to the Sparseness parameter, for the Narrow Genetic Variation condition

	Sparseness Parameter Value				
Number of 1-valued alleles	0	1	2	3	4
Population probability	0.063	0.250	0.375	0.250	0.063
Parameter value	0	0.05	0.1	0.15	0.2

3. A note on the artificial genome

One can view the artificial genome simply as a method to generate a population with variability in the learning abilities of individual, while providing the basis to encode the similarity between siblings of different types (MZ or DZ twins). In terms of a model of the actual relation between levels of description, the artificial genome is clearly remote from real cellular function.

Nevertheless, it is worth noting that the relation of genotype to neural substrate respects several distinct genetic contributions. The parameters corresponded not only to how the network was built (e.g., number of layers, internal units, and connections) but also how it ran (processing noise, threshold functions), how it was maintained (weight decay and pruning), and how it adapted (learning rate, momentum). Some of these parameters can be seen as analogous to genetic effects on brain development operating early on and not thereafter (e.g., neurogenesis, neural migration), while others can be seen as analogous to on-going gene expression in the maintenance of function (neural dynamics) and adaptive processes (plasticity, pruning, decay). The artificial genome does not represent a design blueprint but stipulates many aspects of ongoing functioning.

4. A note on additivity

The action of genes on the artificial genome was intended to be additive. No dominant effects or interactions between genes were implemented. Nevertheless, our use of lookup tables to map between the multiple genes influencing a given neurocomputational parameter and the value of that parameter turned out to inadvertently implement non-additive effects. Recall, the lookup table for each

parameter summed the number of 1-valued gene variants amongst the set of artificial genes influencing each parameter and retrieved the associated parameter value. However, the function linking the number of 1-valued variants with the parameter value was not necessarily linear, nor necessarily symmetrical around the 'normal' or 'average' value of the parameter (see Figure 3). The reason for this was that during calibration, we had attempted to make poorer and better *behavioural outcome* roughly symmetrical around average performance for each parameter. This required that the change in parameter value be sometimes non-linear. For example, a few less hidden units below average could cause performance to drop off quickly, but many more hidden units above average were required to give an equivalent performance gain. However, the consequence of this assumption was that the effect on the final parameter value of adding one extra 1-valued variant to the sum could differ, depending on whether there were many other 1-valued or few other 1-valued variants amongst the group. No dominant or interactive effects were encoded, in the sense that the artificial genes for a parameter were interchangeable in their effects. Nevertheless, the consequence of designing in behavioural symmetry was that strict additivity did not hold in the way gene variants combined. We ran further populations ensuring linearity in the mapping between artificial genes and parameters. The results of these simulations have not yet been reported.

5. References

Beversdorf, D. Q., Narayanan, A., Hillier, A., & Hughes, J. D. (2007). Network model of decreased context utilization in autism spectrum disorder. *Journal of Autism and Developmental Disorders*, *37*, 1040– 1048.

Cohen, I. L. (1998). Neural network analysis of learning in autism. In D. J. Stein & J. Ludik (Eds.), *Neural networks and psychopathology* (pp. 274–315). New York, NY: Cambridge University Press.

Harm, M. W. & Seidenberg, M. S. (1999). Phonology, reading acquisition, and dyslexia: Insights from connectionist models. *Psychological Review*, *106*, 491-528.

Harm, M. W., & Seidenberg, M. S. (2004). Computing the meanings of words in reading: Cooperative division of labor between visual and phonological processes. *Psychological Review*, *111*, 662-720.

Hinton, G. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*, 185–234.

Hoeffner, J. H. & McClelland, J. L. (1993). Can a perceptual processing deficit explain the impairment of inflectional morphology in developmental dysphasia? A computational investigation. In E.V. Clark (Ed), *Proceedings of the 25th Child language research forum*, (pp. 1-25). Stanford University Press.

Huttenlocher, P. R. (2002). *Neural plasticity: The effects of environment on the development of the cerebral cortex.* Cambridge, MA: Harvard University Press.

Joanisse, M. F., & Seidenberg, M. S. (1999). Impairments in verb morphology after brain injury: A connectionist model. *Proceedings of the National Academy of Sciences of the United States of America*, *96*, 7592-7597.

Joanisse, M. F. (2004). Specific language impairments in children: Phonology, semantics and the English past tense. *Current Directions in Psychological Science*, *13(4)*, 156-160.

Joanisse, M. F., Seidenberg, M. S. (2003). Phonology and syntax in Specific Language Impairments: Evidence from a connectionist model. *Brain and Language*, *86*, 40-56.

Karaminis, T. N., & Thomas, M. S. C. (2010). A cross-linguistic model of the acquisition of inflectional morphology in English and Modern Greek. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of 32nd Annual Conference of the Cognitive Science Society*, August 11-14, 2010, Portland, Oregon, USA.

Kovas, Y., & Plomin, R. (2006). Generalist genes: implications for the cognitive sciences. *Trends in Cognitive Sciences*, *10(5)*, 198-203.

Li, S.-C. & Lindenberger, U. (1999). Cross-level unification: A computational exploration of the link between deterioration of neurotransmitter systems and the dedifferentiation of cognitive abilities in old age, (pp. 103-146) in L.-G. Nilsson & H. Markowitsch (Eds.). *Cognitive neuroscience of memory*. Toronto: Hogrefe & Huber

McClelland, J. L. (1993) Toward a theory of information processing in graded, random, interactive networks. In D. E. Meyer & S. Kornblum (Eds.), *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence and cognitive neuroscience* (pp. 655-688). Cambridge, MA: MIT Press.

McClelland, J. L. (2000). The basis of hyperspecificity in autism: A preliminary suggestion based on properties of neural nets. *Journal of Autism and Developmental Disorders*, *30*, 497–502

Munakata, Y., & McClelland, J. L. (2003). Connectionist models of development. *Developmental Science*, *6(4)*, 413-429.

Plaut, D. C. (1997). Structure and function in the lexical system: Insights from distributed models of word reading and lexical decision. *Language and Cognitive Processes*, *12*, 767-808.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. E. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*, 56-115.

Plomin, R., DeFries, J. C., McClearn, G. E., & McGuffin, P. (2008). *Behavioral genetics (5th Edition)*. New York: Worth Publishers.

Plunkett, K. & Marchman, V. (1991). U-shaped learning and frequency effects in a multilayered perceptron: Implications for child language acquisition. *Cognition*, *38*, 1-60.

Plunkett, K. & Marchman, V. (1993). From rote learning to system building: acquiring verb morphology in children and connectionist nets. *Cognition*, *48*, 21-69.

Richardson, F. M., Forrester, N. A., Baughman, F. D., & Thomas, M. S. C. (2006b). Computational modeling of variability in the conservation Task. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (p. 2010-2015), July 26-29, Vancouver, BC, Canada.

Richardson, F. M., Seghier, M. L., Leff, A. P., Thomas, M. S. C., & Price, C. J. (2011). Multiple routes from occipital to temporal cortices during reading. *Journal of Neuroscience*, *31(22)*, 8239-8247.

Richardson, F.M., Baughman, F.D., Forrester, N. A., & Thomas, M.S.C. (2006a). Computational modeling of variability in the balance scale task. *Proceedings of*

the 7th International Conference of Cognitive Modeling, (pp 256-261). Trieste, Italy: Edizioni Goliardiche.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland and The PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 318-362). Cambridge, MA: MIT Press.

Sapolsky, R. (2005). *Biology and human behavior: The neurological origins of individuality (2nd Ed.)*. Chantilly, VA: The Teaching Company.

Seghier, M. L., Lee, H. L., Schofield, T., Ellis, C. L., & Price, C. (2008). Inter-subject variability in the use of two different neuronal networks for reading aloud familiar words. *Neuroimage*, *42(3-3)*, 1226-1236.

Simmons, D. R., McKay, L., McAleer, P., Toal, E., Robertson, A., & Pollick, F. E. (2007). Neural noise and autism spectrum disorders. *Perception*, *36(Suppl.)*, 119–120.

Strachan, T., & Read, P. (2003). *Human molecular genetics 3*. Garland Publishing.

Thomas, M. S. C. (2005). Characterising compensation. *Cortex*, *41(3)*, 434-442.

Thomas, M. S. C. & Johnson, M. H. (2006). The computational modelling of sensitive periods. *Developmental Psychobiology*, *48(4)*, 337-344.

Thomas, M. S. C. & Karmiloff-Smith, A. (2002b). Are developmental disorders like cases of adult brain damage? Implications from connectionist modelling. *Behavioral and Brain Sciences*, *25(6)*, 727-788.

Thomas, M. S. C. & Karmiloff-Smith, A. (2003). Modelling language acquisition in atypical phenotypes. *Psychological Review*, *Vol. 110*, *No.4*, 647-682.

Thomas, M. S. C. & Karmiloff-Smith, A. (2003a). Connectionist models of development, developmental disorders and individual differences. In R. J. Sternberg, J. Lautrey, & T. Lubart (Eds.), *Models of Intelligence: International Perspectives*, (p. 133-150). American Psychological Association.

Thomas, M. S. C., & McClelland, J. L. (2008). Connectionist models of cognition. In R. Sun (Ed.), *Cambridge handbook of computational cognitive modelling* (pp. 23-58). Cambridge: Cambridge University Press.

Thomas, M. S. C., Knowland, V. C. P., & Karmiloff-Smith, A. (2011). Mechanisms of developmental regression in autism and the broader phenotype: A neural network modeling approach. *Psychological Review*, *118(4)*, 637-654.

Westermann, G (1998) Emergent modularity and U-shaped learning in a constructivist neural network learning the English past tense. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 1130-1135). Hillsdale, NJ: Erlbaum.

Woollams, A. M., Joanisse, M., & Patterson, K. (2009). Past-tense generation from form versus meaning: Behavioural data and simulation evidence. *Journal of Memory and Language*, *61*, 55-76.

Zorzi, M., Houghton, G., & Butterworth, B. (1998). Two routes or one in reading aloud? A connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance*, *24*, 1131-1161.

Document last edited 4/5/12

MT.