# Creating Hierarchical Categories Using Cell Assemblies

Christian R. Huyck

Middlesex University

c.huyck@mdx.ac.uk

### Abstract

Highly recurrent neural networks can learn reverberating circuits called Cell Assemblies (CAs). They can be used to categorise input, and in this paper we explore the use of CAs in developing hierarchical categories.

Using our simulator, we initially derive networks that act as hierarchical categorisers. The calculated network is based on a Hopfield network, but is gradually transformed into a network of neurons that are fatiguing leaky integrators. This transformed model is an improved categoriser. The model takes advantage of overlapping CAs where neurons may participate in more than one CA.

We then use a compensatory learning rule, a Hebbian learning rule, to learn the network in an unsupervised manner. The networks learn a hierarchy of categories that correctly categorise 97% of presentations of the input in our test.

## 1 Introduction

Cell Assemblies (CAs), groups of neurons that form reverberating circuits, have been proposed as a mechanism for an intermediate level of cognition [Huyck, 2001]. They are more complex than neurons (a lower level of cognition), and can form the basis of symbols among other things. One answer to the question *how do we think?* is we use our neurons (the lower level); another answer is that we use our CAs; and yet another is that we use symbols [Newell, 1990] (the higher level). CAs provide a bridge between the more traditional answers.

CAs are more complex than neurons, so what do we gain by modelling cognition in this more complex manner. CAs answer many questions, which a theory that focuses solely on neurons cannot: short-term persistence of memory [Hebb, 1949], long-term persistence of memory [Hebb, 1949], categories [Hebb, 1949], figure-ground separation [Hebb, 1949], completion effects [Hebb, 1949, Fu, 2003], ambiguity resolution [Fu, 2003], cognitive maps

[Chown, 1999], and as we show in this paper hierarchies. The interplay between neurons and groups of neurons is crucial in explaining the system's behavior.

A neural network can contain many CAs and the long term dynamic of the neural network can be seen as a process of CA growth, modification and fractionation all achieved by learning. Initially a CA will form and grow by increasing intra-CA synaptic strength. Uncommitted neurons, weakly committed neurons, or strongly related neurons (via synaptic strength) may be recruited into a CA even though they never occur in the external stimulus. This will cause the CA to contain more neurons. Once formed, a CA may also break into two or more sub-CAs through a process called fractionation [Hebb, 1949].

A network of neurons that contains CAs is a categoriser. An instance is presented to the net, and the CA that represents the category to which the instance belongs is ignited and remains active. Categorisation via CA ignition is the short-term dynamic. A well known facet of human memory is that we categorise things hierarchically (e.g. [Rosch and Mervis, 1975]), so that a particular instance, say a sighting of an animal, might activate several categories, say *Lassie*, *Collie*, *Dog*, *Mammal*, *Animal*, and *Thing*. A complete model of memory would need to account for this so a model of memory must either handle hierarchy or defer to another mechanism. Fortunately, as this paper shows, CAs can both learn and use hierarchical categories.

Our current model of hierarchical CAs is based on overlapping sets of neurons. A CA for a super-category contains many of the neurons that are also contained in the CAs that represent its subcategories. We believe this model is incomplete but is sufficient to represent hierarchical CAs. Moreover, this hierarchy of categories can be learned in an unsupervised manner.

In this paper, we first present a background of prior work in CAs, defend their biological and computational plausibility, and give a brief background of hierarchies. Section 3 describes our computational implementation of CA theory and shows how it relates to some similar models. Section 4 describes a simulation of a net of hierarchical categories that we calculate; this calculation uses a Hopfield network [Hopfield, 1982] as a starting point, and modifies that net to make it a net that is both more biologically plausible and has more computational power. Section 5 describes the results of a simulation to learn a simple hierarchy from presentations of instances of the basic categories. In section 6 we discuss the importance of this result and show how it might be extended. We conclude with a brief discussion of ongoing and future CA work.

## 2 Background

CAs were proposed by Hebb [Hebb, 1949]. Long-term memories are represented by neurons that have high mutual connection strength. When a small percentage of these neurons are activated from input outside these neurons, they raise the activity of the other neurons, and cause them to fire [Kaplan et al., 1991]. This causes a cascade of firing, called CA ignition, that creates a reverberating circuit that can persist after the external stimulus ceases.

Hebb is better known for Hebbian learning. Two neurons that co-fire tend to have the strength between them increased. The high mutual connection strengths between neurons in a CA are formed by Hebbian learning. The Hebbian rule is combined with an anti-Hebbian rule, which reduces strength when one neuron fires and the other does not. When combined the rules define the long term behavior of a synapse based solely on the behavior of the neurons it connects.

The brain requires at least two types of dynamics: short-term dynamics and long term dynamics undone cite . Short-term dynamics are needed to recognise things. This is done by the network receiving external activation from the environment. Neurons are activated from the environment; these neurons in turn activate other neurons leading to the ignition of a particular CA. The item that caused the external activation is categorised as belonging to the class of items that the ignited CA represents. This short-term dynamic is rapid and recognition usually occurs in less than a second [Kieras et al., 1997].

Long-term dynamics are needed to build these CAs. CAs represent semantic categories[1] and the CAs are learned by having many instances of the category presented to the network. Over many presentations, Hebbian learning causes synaptic weights within a category to increase. This eventually leads to a CA which is an attractor state. If a stimuli is presented that is within the basin of attraction, the CA will ignite. The ignited CA is an

[1] The CAs that we are discussing in this work are long term categories. It is clear that memories can be learned from a single presentation; this episodic memory is not the topic of this paper though see section 7. Long term semantic memory is needed as a prerequisite for this impressive one shot learning and this semantic memory is the problem on which we are currently concentrating.

attractor state that many initial states lead to, so all of these initial states will be categorised as the same CA.

In the untrained network that we are simulating, no CAs are activated. As the network is modified through synaptic change based on external stimulation, CAs are learned. This dynamic takes many presentations and may take days, months, or even years to form a CA in a mammal undone cite. However, when any of a range of stimuli are presented a CA will ignite.

## 2.1 Psuedo-Stable States

In addition to being an attractor state, a CA is a pseudo-stable state. A CA is a group of neurons that are fired and more or less continue to fire. Thus an item is presented to the network, and the network settles into a stable state where the neurons in one CA fire and the other neurons do not. The initial part of this process, known as CA ignition, is closely related to stable states of Hopfield networks [Aleksander, 1989].

However, the network needs to move on to other tasks, so it needs to move from this stable state. Fatigue is part of the solution to this problem [Horn and Usher, 1989]. Neurons fatigue, so the state of an ignited CA is not entirely stable. It is pseudo-stable firstly because neurons in the CA are firing in one cycle but not the next due to fatigue. Secondly, the overall fatigue in the CA builds until it actually shuts down the CA [Kaplan et al., 1991], thus freeing the network to process another input[2].

Ambiguous input must also lead to one of these pseudo-stable states [Huyck, 2002b]. An ambiguous input could activate one of several CAs, but these CAs compete with each other via mutually inhibitory connections. Once one CA has won the competition, it inhibits the others as well as keeping its own pseudo-stable state active.

Inhibition prevents neurons outside of the active CA from firing. In the extreme case, inhibition prevents simulated epilepsy, the state when the whole network starts to fire[3].

[2] Fatigue alone may not be responsible for shutting off a CA. Another CA may need to ignite to shut the first one down.

[3] This would be a system with just two stable states: all neurons firing, or all neurons off. Such a system is not a very useful.

## 2.2 Recruitment and Fractionation

In the long-term, a CA may also break into two or more sub CAs through a process called fractionation [Hebb, 1949]. Early in the development of the network, the presentation of an external stimulus may activate a CA that consists of say 1000 neurons. For example's sake, we number these neurons 1 to 1000. As the network develops and many new items are presented, the synaptic strength within these neurons will change so that the first 500 neurons have larger intra-synaptic strength than in the initial network. The second 500 neurons also have larger intra-synaptic strength, while synaptic strength between the two groups of 500 is diminished. At some point, the presentation of a new item is presented and only the first 500 neurons will become active, whereas a different item may activate the second 500 neurons. In this case, the CA has fractionated into two new CAs. Neurons 1 to 500 do not activate alongside 501 to 1000 because they have high mutual inhibition.

Fractionation is difficult because the Hebbian learning rule strengthens connections between co-firing neurons. Whenever the initial CA is activated, neurons in both sub-groups are coactive. Let us call this the Mule problem. The system is presented with many instances of horses, and it forms a *Horse* CA. When a horse is presented, many neurons in the *Horse* CA become active and it is ignited. Now the system is presented with a mule. As there is no *Mule* CA and mules have many features a horse has, the *Horse* CA is activated. This is probably a good thing, because the mule can largely be treated as a horse. However, if the system starts to run into many mules, the system should be able to learn a mule CA. Unfortunately, every time a mule is presented neurons in the *Horse* CA come on, and their intra-synaptic strength is increased. A good system should be able to overcome the Mule problem.

## 2.3 Neurons in Multiple CAs

A given neuron may participate in more than one CA [Sakurai, 1998]. Work has been done in studying overlapping CAs [Wickelgren, 1999], but this was based on biologically implausible neurons. To follow our earlier example, when the *Horse* CA is formed it consists of neurons 1 to 1000. When this fractionates into two CAs it may fractionate into the new *Horse* CA consisting of neurons 1 to 800 and the new *Mule* CA consisting of neurons 300 to 1000. So, a horse is presented by externally stimulating some of the neurons between 1 and 800, while a mule is presented by stimulating some neurons between 300 and 1000. If an item only activated neurons between 300 and 800 it would be ambiguous; an item that also activated some neurons between 1 and 300, and between 800 and 1000 would also be ambiguous.

If a solution to the Mule problem can be found, hierarchies may emerge. One solution would follow the above example; the *Proto-Horse* CA consists of neurons 1 to 1000. When the new *Horse* and *Mule* CAs form they consist of neurons 1 to 800 and 300 to 1000 respectively. The new *Proto-Horse* CA could be the neurons 300 to 800. That is, if the neurons 300 to 800 are active it is a horse-like thing, if neurons 1 to 800 are active it is a horse, and if neurons 300 to 1000 are active it is a mule. The *Proto-Horse* CA is a super-category.

## 2.4 Subcategorisation Hierarchies

The type of hierarchical relationship that we are currently discussing is the subcategorisation relationship, typically labelled *IS-A*. The categories are related in a type of semantic net [Quillian, 1967]. In this case a particular item is a member of a class, but every member of this class is also a member of another class. For example, *Lassie* is a member of the *Collie* class, and all *Collies* are members of the *Dog* class; of course all *Retrievers* are also a member of the *Dog* class, and all *Dogs* and *Cats* are members of the *Mammal* class.

Subcategorisation is a very powerful technique, allowing a huge range of generalisations to be made. For instance, you may never have seen *Lassie*, but knowing *Lassie* is a *Dog*, you know that *Lassie* almost certainly barks; knowing *Lassie* is a *Mammal* and a *Female* you know that *Lassie* feeds her children with milk. This gives a great deal of representational efficacy. Subcategorisation is so powerful that it is a hallmark of Object Oriented Programming (OOP); in OOP inheritance is subcategorisation.

Other types of hierarchical relationships exist. An obvious one is the part-of relationship where an object, such as a steering wheel, is part-of another object, such as an automobile. We believe that CAs can also generate this type of hierarchical relationship, but this paper concentrates on subcategorisation hierarchies.

### 2.4.1 Psychological Foundations of Categories and Hierarchies

There has been a great deal of research on categorisation from machine learning a nd psychological perspectives. For instance, Shepard et al. [Shepard et al., 1961] show how it is easier for humans to learn categories that differ by only one feature; however, they can learn categories that are not linearly separable.

A great deal of machine learning work has been done developing symbolic and statistical algorithms for supervised and unsupervised category formation. For example, Hanson and Bauer [Hanson and Bauer, 1989] have developed clustering techniques based on feature similarity and difference within and between categories. This work is explicitly linked with human categorisation data.

One obvious factor, explored by Hanson and Bauer, is that concepts are polymorphic. That is, they are not usually defined by necessary and sufficient conditions. This problem was discussed by Wittgenstein [Wittgenstein, 1953], and the result is that categories are more like family resemblances. This type of category is quite simple for neural networks to learn by unsupervised training.

Reseach on categorisation has shown that humans form hierarchical categories. The original idea of semantic nets was developed by Quillian [Quillian, 1967] to account for human categories. It has been shown that a spreading activation model based on semantic nets can account for a range of experimental results [Collins and Loftus, 1975].

Rosch [Rosch and Mervis, 1975] has made a detailed exploration of hierarchical categories and has found that humans have basic level categories. These are the most important categories, are the first learned, are more salient, and this effect is generally cross-cultural.

In this paper we are not trying to account for all of the psychological effects of hierarchy or even an appreciable subset of these effects. Instead we are attempting to show how a neural system could account for hierarchies.

It is however important that these effects are not impossible for our system. We feel that our system is capable of all of these behaviors though we have not yet simulated them. For instance, our system can account for linear separability by the use of hidden neurons. These neurons can be recruited into CAs via spontaneous neural activation [Huyck and Bowles, accp].

### 2.4.2 Other Connectionist Systems for Hierarchies

Since hierarchy is crucial to categorisation and memory, there has been a great deal of work on connectionist systems that model hierarchy. One common approach is to use standard feed forward networks but to add recurrent connections. Another approach is to use a more biologically plausible network topology such as ART networks [Grossberg, 1987].

Several researchers have looked into hierarchy formation using feed forward networks with recurrent connections. For example, Pollack's RAAM system [Pollack, 1990], Elman's work [Elman, 1990], and Hanson and Negishi's work [Hanson and Negishi, 2002] have all used this class of model. They use standard feed forward networks with supervised learning but add extra back connections or extra layers with connections. These extra layers are not part of the standard feed forward mechanism, but take output from the system and feed it back in as input. So in any given time step there is input from the environment and from the system.

As Elman points out [Elman, 1990], this has the benefit of adding memory to the system. This memory enables these recurrent neural networks to handle sequences, and to some extent develop hierarchies and rules. These systems are useful in understanding how distributed memory representations function and how neurons can combine transduction with memory.

Adaptive Resonance Theory (ART) [Grossberg, 1987] has neural plausibility as one of its goals and can account for hierarchical categorisation. There is a layered architecture and a node for each category. If an input is not associated with a new category a new node is allocated for it. Hierarchy is handled by a change in vigilance; the range of inputs that is accepted is gradually increased so that new nodes account for larger categories. So, a new node may account for several old nodes and will thus be a super-category.

This work is excellent but violates our understanding of neurons. Clearly one neuron cannot account for a concept or we would lose that concept if the neuron died. However, this model may approximate the behavior of groups of neurons, and thus meets its goal of neural plausibility.

## 3 The Simulator

We have developed a computational model to enable us to simulate CAs. Like all models, it is a simplification, but we hope that it includes the essential properties of neurons. We have used this simulator to explore categorisation, ambiguity resolution and other phenomena. The simulator has remained largely unchanged[4]. We have made computational efficiency a priority but have attempted to mimic mammalian cortical neural function.

Mammalian neurons work in continuous time, but this is expensive to

[4] It can be found at http://www.cwa.mdx.ac.uk/cant20/CANT20.html; the simulations from this paper and other simulations are available.

simulate in a digital computer. We have broken time into 10 ms cycles because neurons rarely fire more than once in 10 ms [Abbott et al., 1997]. This time scale enables us to simulate thousands of neurons in real time.

Our neurons are fatiguing leaky integrators; it is clear that biological neurons integrate activation from incoming synapses, that activation leaks away, and that they fire sending activation to the synapses leading from them [Churchland and Sejnowski, 1992].

In our model, neurons collect activation and fire if activation is above a threshold $\theta$. If neurons do not fire in a given time step, some but not all of the activation leaks away.

$h_{i_t} = \frac{h_{i_{t-1}}}{d} + \sum_{j \in V_i} w_{ij}$ Equation 1

Equation 1 describes the activation of a neuron $i$ at time $t$, if it has not fired in the previous cycle. It retains some of its activation from time *t-1*, but this is reduced by a decay constant $d$. The neuron integrates the retained activation with the new activation from the active inputs of all neurons $j \in V_i$, $V_i$ being the set of all neurons that fired at *t-1* that are connected to $i$, weighted by the value of the synapse from neuron $j$ to neuron $i$.

If neurons do fire, all activation is lost; in Equation 1, $d = \infty$. Activation (or inhibition) equivalent to the synaptic weight is sent from the firing pre-synaptic neuron to all if its post-synaptic neurons.

Neurons also fatigue so that the more steps they fire the more difficult it becomes for them to fire. This is modelled by increasing the activation threshold if a neuron fires as described by equation 2.

$T_t = T_{t-1} + F_c$ Equation 2

In equation 2 the threshold $T$ at time $t$ is set to the threshold at time *t-1* + the fatigue constant $F_c$. If the neuron does not fire the threshold is reduced toward the base resting level as in equation 3.

$T_t = T_{t-1} + F_r$ Equation 3.

The threshold is reduced by the fatigue recovery constant $F_r$ though it never becomes less than the base resting level.

Neurons may be inhibitory or excitatory, but they obey Dale's principle [Eccles, 1986] so that a neuron cannot have both inhibitory and excitatory synapses leading from it. In the experiments described in this paper, the ratio is 80/20 excitatory/inhibitory as in the mammalian cortex [Bratenberg, 1989].

Neurons have a small number of synapses to and from them. Like the mammalian brain, excitatory neurons are likely to connect to neurons that are nearby. In this paper, excitatory neurons also have one long distance axon with several synapses. So a neuron connects to nearby neurons and to neurons in one other area of the net. These connections are assigned randomly, so each new net is different from another net with the same number of neurons.

The topology we use is a torus, a grid with the sides connected and the top connected. This is a simplifying assumption and this topology is not biologically plausible, though it does avoid boundary effects.

When learning, our simulations use Hebbian learning rules. Synaptic weight is changed based solely on the properties and firing of the pre and post-synaptic neurons.

## 3.1 The Hopfield Model

A compuationally similar model is the popular Hopfield network [Hopfield, 1982]. A Hopfield network consists of nodes that are well connected. Each node has a threshold associated with it, and if the node receives enough activation to surpass its threshold in a given step, it fires and sends activation to all the other nodes. Each connection may have a separate weight. Hopfield nets are not biologically plausible, and table 1 shows some differences between our nets and Hopfield nets.

| Property | Hopfield | Mammalian Brain/CAs |
|---|---|---|
| Connections | Bidirectional | Unidirectional |
| Connectivity | Well Connected | Sparsely Connected and Distance Biased |
| Fatigue | None | Fatiguing |
| Decay | No Activity Retained | Activity Leaks Away |
| Threshold | Positive or Negative | Positive(Constant in CAs) |
| Dale's Principle | Neurons can have positive and negative synapses | Synapses from a neuron are either all positive or all negative |
| Learning | Various | Hebbian but Open Question |

Table 1: CAs vs. Hopfield Nets

The Hopfield model comes from spin glass models that have been used in physics. These models have been extensively studied, and we have extensive knowledge of the computional theoretical properties of these networks. For instance, Hopfield networks can store at most $n$ non-orthogonal patterns in a network of $n$ neurons [Abu-Mostafa and Jaques., 1985].

### 3.2 Related Neural Models

Both our model and the Hopfield model are one of a large family of attractor neural networks. To the best of our understanding, the mammalian brain is an attractor neural network. There are many models that attempt to model the brain from the neural level.

undone other non-neural attractor nets (from amits book).

There has been a long history of systems that have attempted to simulate CA behavior using simulated neurons. Rochecester et al. [Rochester et al., 1956] worked with 512 neurons almost 50 years ago, but failed to develop sequences of CAs. Hetherington and Shapiro [Hetherington and Shapiro, 1993] used neurons with continuous synapses, instead of spiking neurons; their system formed CAs that were reliably activated, and persisted.

Bower and Beemer [Bower and Beeman, 1995] have attempted to model the neuron at a very low level modeling a host of phenomena including calcium gating during firing . Fransen et al. [Fransen et al., 1992] have used the SWIM simulator to model CA from low level neural models similar to Bower and Beemer. Amit and Brunel [Amit and Brunel, 1997] have used mean-field descriptions and a simulation to model CAs. This model includes sophisticated time integration and spontaneous neural activity. Brunel [Brunel, 1996] has used a model similar to [Amit and Brunel, 1997] but added Hebbian learning rules that adhere to some current neurophysilogical theory. Amit and Mongillo [Amit and Mongillo, 2003] have used a model similar to [Amit and Brunel, 1997] that has quite sound formation of CAs. Sougne has developed the INFERNET simulator [Sougne, 2001]. It has a complex spike timed neural model and is simulating variable binding activity (see section 7). Crucially, all of these CA simulations have used non-overlapping CAs.

### 3.3 Discussion of Models

*All models are wrong; some models are useful -George Box*

We think it is essential to develop models that are closer and closer to mammalian neural functioning to improve our understanding of that function. However, while this work may have ramifications for more sophisticated processing, but it may be unnecessary to develop effective categorisers. For instance, a more fine grained time step may not substantially improve categorisation. We do not discount this work, but a more accurate neural model may not make a better categoriser, and will increase the simulation time.

Our model focuses on what we hope are the critical computational properties of neurons. We hope that we can use this model for complex AI problems as well as cognitive simulations. In particular we show below that we can use this model to simulate hierarchical category formation in populations of neurons that implment overlapping CAs. In the long run, we hope that our model is close enough to mammalian neurons to solve problems that only mammalian neural systems are currently capable of solving. Hierarchy is obviously an early step in this process.

## 4 Simulation 1: Calculating a Hierarchy

Our first step in developing a system that could learn hierarchical categories using CAs was to calculate a network that had a hierarchy of categories. Once we had calculated, or automatically created, such a network, we could explore its properties and decide what was needed to learn the network from a random initial net (see section 5).

Unlike all other simulations of categories using biologically plausible neurons, we have done simulations of overlapping categories [Huyck, 2002a] (see section 2.3), and overlapping categories are essential to our approach to hierarchical categorisation. We hypothesise that a superordinate category consists mostly of neurons that occur in the prototypical subordinate category. This approach of sharing neurons has been used to simulate hierarchical categories using Hopfield networks [Rupin and Usher, 1990]. For instance, the *Mammal* superordinate category would largely consists of neurons that were in most of its subordinate categories (e.g. *Cat*, *Dog*, and *Rat*).

To exhibit subcategorisation, what is needed is a network with four possible pseudo-stable states: one for each of the three subordinate categories, and one for the superordinate category. These states could share neurons, but need to be distinct. Using a technique described in [Huyck and Mitchell, 2002] many Hopfield networks [Hopfield, 1982] can be translated into more biologically plausible networks. So, we needed to create a Hopfield network that had a hierarchy of categories.

The Hopfield net in Figure 1 has four stable states. Note that arcs with a weight of 0 (e.g. 0-6) are omitted from the figure.

As for all Hopfield nets, all connections are bi-directional. In this net, the activation threshold for all neurons is *.5*. There are four stable states corresponding to the categories of interest: *0-3-4-5-8*, *1-3-4-6-9*, *2-3-4-7-8*, and *3-4-8*. For expository convenience, these states correspond to *Dog*, *Cat*, *Rat* and *Mammal*. The individual neurons correspond to features. Neurons

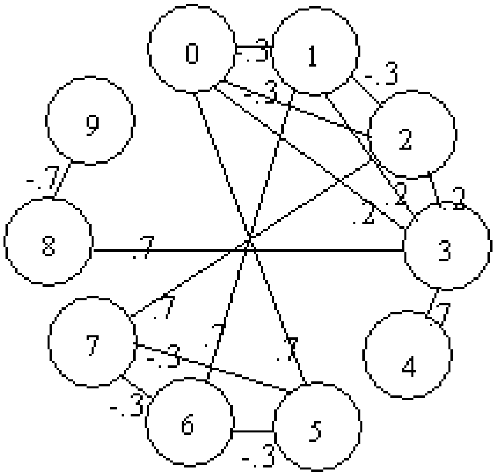

Figure 1: Hopfield Net Representing a Hierarchy

0 to 2 refer to features that a specific animal has with 0 referring to features that dogs have but cats, rats, and all other mammals do not, 1 to features cats have, and 2 to features rats have with restrictions as before. Neurons 3 and 4 refer to features all mammals have. 5 to 7 refer to features that relate to the tails of animals; 5 to tails that wag, 6 to fluffy tails, 7 to furless tails. Neurons 8 and 9 refer to features relating to what the animals eat, 8 refers to omnivores, and 9 to carnivores. Given the 4 basic stable states that are desired, it is straight forward to solve the simultaneous equations to get the connection weights and the thresholds [Hopfield, 1982].

As CAs (ignoring the differences), this Hopfield net performs poorly on many metrics. For instance, CAs should exhibit completion effects; presenting part of the pattern should lead to the whole pattern becoming active. Presenting only part of the *cat* pattern, say *1-9* leaves only part of it active, *1-6-9* in this case. Our CA nets typically run with many more than 10 neurons; using the above pattern as a base, we can create a 20x20 net with 40 sets of these patterns. Of course, feature *1* now refers to all 40 neurons that represent feature *1*, or a subset of those neurons.

The complete *Dog*, *Cat*, and *Rat* patterns have 200 individual neurons active, 5 of the 10 in the Hopfield model times 40. The *Mammal* pattern has 120 neurons active. Figure 2 shows an example of one of the external

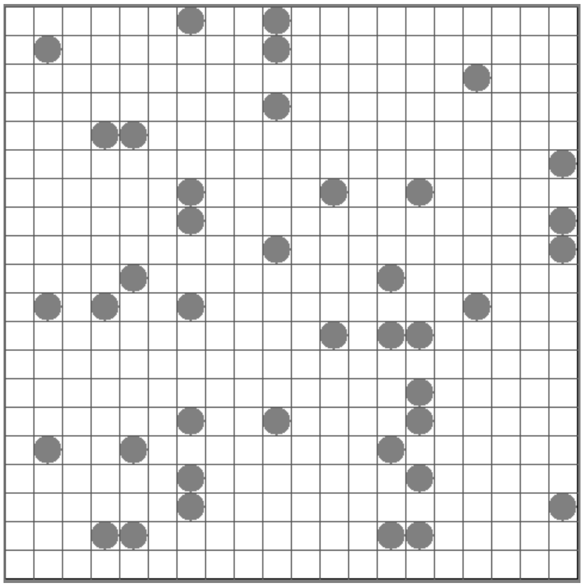

Figure 2: 40 Neurons in the Dog Pattern

activations of a *Dog* pattern. All the neurons are from columns 1, 3, 4, 6, 9, 11, 13, 14, 16 and 19.

With the above Hopfield net run by externally activating 40 randomly selected neurons from a pattern, less than half of the neurons in the basic categories are activated. Over 100 trials an average of 90.5 *Dog*, *Cat*, or *Rat* neurons are activated when an appropriate stimuli is presented; over 100 trials an average of 83.46 *Mammal* neurons are activated when 40 randomly selected *Mammal* neurons are presented.

We also use Pearson's Product correlation to show correlation between nets. Table 2 shows the correlation values when all of the neurons of a given pattern are active.

| | Dog | Cat | Rat | Mammal |
|---|---|---|---|---|
| Dog | 1.0 | -.20 | .20 | .65 |
| Cat | | 1.0 | -.20 | .22 |
| Rat | | | 1.0 | .65 |

Table 2: Correlations Between Fully Activated CAs

Ideally a wide range of partial activations would lead to the same CA being activated. So if 40 of the 200 *Dog* neurons are activated externally, the system runs and a pseudo-stable state should be the result. If a different set of 40 *Dog* neurons are activated, a similar pseudo-stable state should be the result. Pearson's Product correlation measures how similar the states are.

Table 3 shows that different version of a pattern activate similar neurons and different patterns are much less correlated. Each column and row refer to an instance of a particular pattern where 40 of the neurons are randomly selected. The cells represent the correlation between the eventual stable states when the network is run on these patterns. Table 4 shows the same is true with the *Mammal* categories. The intra-category correlations are higher than the inter-category comparisons, but they are quite close. Thus based on sparse presentation, it is difficult to categorise the input using this Hopfield net; that is, the Hopfield net is poor at completion, and thus a poor categoriser.

undone dog1

| | Dog 2 | Dog 3 | Cat 1 | Cat 2 | Cat 3 | Rat | Rat 2 | Rat3 |
|---|---|---|---|---|---|---|---|---|
| Dog 1 | | .249 | .061 | -.016 | -.015 | .175 | .242 | .265 |
| Dog 2 | | .249 | .061 | -.016 | -.015 | .175 | .242 | .265 |
| Dog 3 | | | .065 | .061 | -.087 | .149 | .041 | .112 |
| Cat 1 | | | | .259 | .248 | -.007 | -.052 | .030 |
| Cat 2 | | | | | .138 | .015 | -.001 | -.089 |
| Cat 3 | | | | | | -.028 | .100 | .082 |
| Rat 1 | | | | | | | .398 | .317 |
| Rat 2 | | | | | | | | .393 |

Table 3: Correlations Between Hopfield Subcategory Runs

Fortunately, our nets are more flexible than Hopfield nets. For instance, it is simple to add a connection from an element in one pattern to an element in another. A connection is made from each element, to the corresponding element in the next pattern and the connection has a weight of .3. This minor change dramatically increases the separation for the categories based on correlation. Table 5 shows these correlations. The number of neurons activated also increases. Over 100 trials an average of 115.37 *Dog*, *Cat*, or *Rat* neurons are activated when an appropriate stimuli is presented; over

| | Mammal 2 | Mammal 3 | Dog 1 | Cat 1 | Rat 1 |
|---|---|---|---|---|---|
| Mammal 1 | .639 | .658 | .273 | -.021 | .313 |
| Mammal 2 | | .647 | .299 | -.132 | .386 |
| Mammal 3 | | | .219 | -.099 | .215 |

Table 4: Correlations Between Superordinate and Subordinate Hopfield Runs

100 trials an average of 85.23 *Mammal* neurons are activated.

| | Dog 2 | Dog 3 | Cat 1 | Cat 2 | Cat 3 | Rat 1 | Rat 2 | Rat3 |
|---|---|---|---|---|---|---|---|---|
| Dog 1 | .378 | .327 | -.081 | -.069 | -.069 | .126 | .103 | .125 |
| Dog 2 | | .414 | -.081 | .064 | .064 | .166 | .072 | .132 |
| Dog 3 | | | -.059 | .076 | .076 | .034 | .086 | .025 |
| Cat 1 | | | | .457 | .135 | -.131 | -.131 | -.049 |
| Cat 2 | | | | | .284 | .064 | -.147 | -.178 |
| Cat 3 | | | | | | .001 | -.132 | -.113 |
| Rat 1 | | | | | | | .372 | .437 |
| Rat 2 | | | | | | | | .368 |

Table 5: Correlations Between Modified Hopfield Runs

We can continue this process of loosening constraints from the initial Hopfield net until we get one of our own networks. We described this technique of converting a Hopfield net into one of our nets in [Huyck and Mitchell, 2002].

Continuing this process leads to the final calculated net that has our normal topology, except that we calculate the synaptic weights. This topology is distance biased, with one long distance axon. That is, a neuron is likely to have connections to nearby neurons, and connections to one other randomly selected area of the network. Each neuron has an average of 40 synapses, with 10 of these being on the long-distance axon.

Neurons obey Dale's Principle so all synapses from a neuron are excitatory or all are inhibitory. Synaptic weights are calculated based on the row of the pre-synaptic neuron and the row of the post-synaptic neuron. These weights are minor variations on the initial Hopfield weights.

The simulation is run without neural decay or fatigue. Virtually all of the neurons come on when 40 are externally stimulated. Over 100 trials an average of 198.95 *Dog*, *Cat*, or *Rat* neurons are activated when an appropriate stimuli is presented; over 100 trials an average of 119.21 *Mammal* neurons are activated. Consequently, the Pearson's correlations are almost exactly the ideal weights of table 2.

We could continue to improve our calculated net, but it is clear that this topology can support the simple hierarchy that we are simulating. All that is needed is for the system to learn these or similar synaptic weights. The system will then have learned a hierarchy in an unsupervised manner.

# 5 Simulation 2: Learning a Hierarchy

The behavior of a network in categorising input patterns is based on its synaptic weights. It is easy enough to calculate such a network (Hopfield solution), but the real question is what is needed to learn these synaptic weights?

The network must be presented with stimuli, and it must learn to categorise it. In this section we first describe the learning rule, then the specific network that was simulated, and the training regime. We then show how the trained network categorises input.

## 5.1 Learning Rule

Our model commits us to a Hebbian learning rule. However, Hebbian learning only commits us to basing the synaptic weights on properties of the pre and post-synaptic neurons. This leaves us with a wide range of options. As there is a wide range of possibilities, we would like to be directed by neurophysiological data. There is neural evidence for the Hebbian learning rule [Tsodyks et al., 1998], but the precise nature of the Hebbian learning rule is still not known. Consequently, we have used a compensatory learning rule.

Compensatory learning considers the total synaptic strengths of the pre and post-synaptic neurons as well as their firing behaviour. Like standard correlational learning, when both neurons fire, the synaptic weight is increased, and when the pre-synaptic neuron fires and the post-synaptic neuron does not fire the weight is decreased. However the changes are multiplied by a compensatory modifier.

The correlatory Hebbian learning rule is based on the increase of equation 4 where $R$ is the learning rate. In this simulation, the learning rate was 0.1. The anti-Hebbian learning rule is based on a decrease of equation 5. These learning rules alone force the weight toward the correlation percentage; that is how likely is the post-synaptic neuron to fire when the pre-synaptic neuron fires.

$\Delta^{+} w_{ij} = (1 - w_{ij}) * R$ Equation 4.

$\Delta^{-} w_{ij} = (w_{ij}) * -R$ Equation 5.

The increasing compensatory modifier is equation 6. The decreasing compensatory modifier is equation 7. $W_B$ is a constant which represents the average total synaptic strength of the pre-synaptic neuron, and $W_i$ is the current total synaptic strength. 5 here is a rather arbitrary constant, and any value greater than 1 would do, though smaller values take longer to converge, and larger values are more likely to arrive at bad stable states early on.

$\Delta^{+}_{mod} w_{ij} = 5^{(W_B - W_i)}$ Equation 6.

$\Delta^{-}_{mod} w_{ij} = 5^{(W_i - W_B)}$ Equation 7.

The base increase and decrease (equations 4 and 5) make the weight tend toward a correlation value near the percentage of time the post-synaptic neuron fires when the pre-synaptic neuron fires. The compensatory modifier forces the total synaptic weight towards the constant $W_B$ because there is more of an increase when the current total synaptic weight is less that $W_B$, less of an increase when the total is greater, and a parallel decrease when the anti-Hebbian rule is used. In this simulation, $W_B$ was 18. This enables neurons with low correlation weights to have enough synaptic strength to participate in reverberating circuits. It also prevents neurons that are frequently present in external stimuli from having very large weights. This in turn prevents simulated epilepsy. A compensatory mechanism is biologically plausible because the overall activation (or inhibition) a neuron can emit is limited. Since a neuron is a biological cell, it has limited resources, and synaptic efficiency may well be one such resource.

There are parallel rules for inhibitory neurons. The correlatory weights reflect the value *percentage-1*, so that neurons that frequently co-fire have little inhibition, but those that rarely co-fire have a great deal of inhibition. The compensatory modifier also prevents inhibitory neurons from having too great an effect and encourages CAs to form.

## 5.2 Experimental Design

The untrained network has an identical connectivity topology to that in the final calculated simulation: a 20x20 network connected in a distance biased fashion. Initial synaptic weights were random and low. During the training

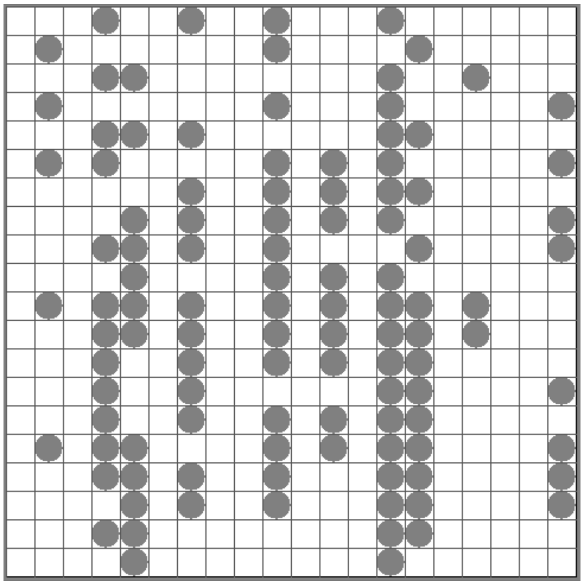

Figure 3: An Activated Dog CA

phase, the network was presented instances of the three basic patterns. As earlier there were an average of 40 synapses per neuron with 10 of those on the long distance axon. There is an 80/20 ratio of excitatory/inhibitory neurons.

The firing threshold was set to 4 for all neurons so it takes several neurons to cause another neuron to fire. The fatigue rate was .6, and the fatigue recovery rate was .9. The decay rate was set to 3 so that a neuron retained only a third of its activation if it did not fire in a time step.

The patterns were the same as those in the calculated simulation: *Dog 0-3-4-5-8*, *Cat 1-3-4-6-9*, and *Rat 2-3-4-7-8*. A given instance had 40 of the 200 neurons randomly selected. Each instance was presented for one cycle and a total of 800 instances were presented.

In the testing phase, learning was turned off, and a randomly selected instance of 40 neurons was presented for 5 cycles. After 5 cycles, firing patterns were measured.

Figure 3 shows a network 5 cycles after the presentation of a *Dog* pattern; this is the externally activated pattern that is shown in Figure 1. Note that in this particular instance, only neurons from the *Dog* pattern are activated but this is not always the case. Figure 4 shows the network 5 cycles after

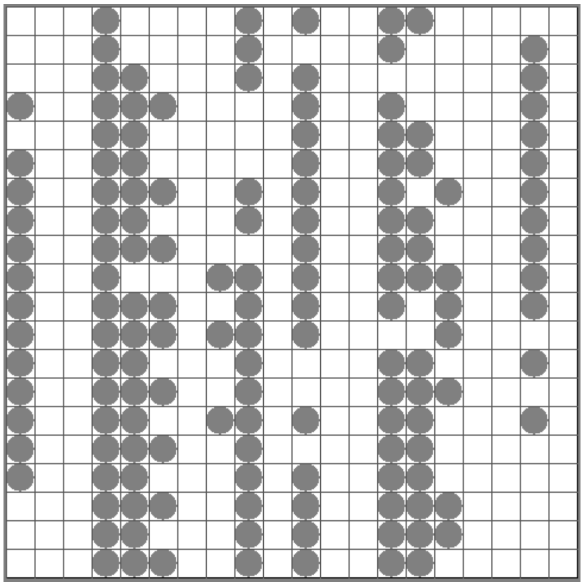

Figure 4: An Activated Cat CA

the presentation of a *Cat* pattern. Four of the 20 columns are active in both, 4 are inactive in both, and 12 are active in only one of these two patterns.

### 5.3 Results

For one network, the Pearson's Product correlations are shown in Tables 6 and 7. In each case, the Pearson's measurement shows that the subordinate categories are closer to other versions of the same pattern than to any versions of any other pattern. In one case, the correlation between two *Mammal* patterns is lower than a *Mammal-Dog* pair. This is reasonable as *Dogs* are *Mammals*, but fortunately the best correlation is with another *Mammal*.

Tables 6 and 7 are an example of one network. We repeated the same process on a total of 10 networks. For each net, we compared each of the subcategory patterns with the other patterns. We found the closest pattern using Pearson's measurements, and this formed the basis of our categorisation. So, if a given pattern was nearest[5] to a *Dog*, it was categorised as a *Dog*. In 87 of the 90 cases or 97% of the cases, the pattern was correctly categorised.

[5] Nearest here means that it had the highest Pearson's measurement.

| | Dog 2 | Dog 3 | Cat 1 | Cat 2 | Cat 3 | Rat | Rat 2 | Rat3 |
|---|---|---|---|---|---|---|---|---|
| Dog 1 | .482 | .538 | .218 | .210 | .211 | .223 | .202 | .272 |
| Dog 2 | | .418 | .165 | .207 | .159 | .238 | .201 | .193 |
| Dog 3 | | | .197 | .202 | .202 | .259 | .247 | .280 |
| Cat 1 | | | | .648 | .585 | .148 | .091 | .142 |
| Cat 2 | | | | | .532 | .129 | .098 | .190 |
| Cat 3 | | | | | | .142 | .073 | .123 |
| Rat 1 | | | | | | | .350 | .302 |
| Rat 2 | | | | | | | | .447 |

Table 6: Correlations Between CA Runs on One Net

| | Mammal 2 | Mammal 3 | Dog 1 | Cat 1 | Rat 1 |
|---|---|---|---|---|---|
| Mammal 1 | .533 | .512 | .522 | .329 | .393 |
| Mammal 2 | | .611 | .376 | .220 | .391 |
| Mammal 3 | | | .357 | .187 | .314 |

Table 7: Correlations Between Superordinate and Subordinate CA Runs on One Net

Similarly, we found the average Pearson's measurements between categories. As expected, patterns of the same category were highly correlated and different categories were more lowly correlated. Table 8 shows these correlations.

| | Dog | Cat | Rat |
|---|---|---|---|
| Dog | .4019 | .0782 | .2779 |
| Cat | | .4846 | .0753 |
| Rat | | | .4498 |

Table 8: Correlations Between Categories

It should be noted that *Dog* and *Rat* are much closer than other categories. This is of course because they share the *omnivore* feature.

Another measurement that is instructive is average excitatory weight between features. This is described in Table 9 where F. 0 refers to a neuron of feature 0.

| | F. 0 | F. 1 | F. 2 | F. 3 | F. 4 | F. 5 | F. 6 | F. 7 | F. 8 | F. 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| F. 0 | .74 | .02 | .02 | .21 | .20 | .86 | .02 | .02 | .41 | .02 |
| F. 1 | .02 | .68 | .02 | .24 | .26 | .02 | .41 | .02 | .01 | .79 |
| F. 2 | .02 | .02 | .64 | .24 | .19 | .02 | .02 | .62 | .36 | .02 |
| F. 3 | .90 | .59 | .58 | .58 | .56 | .71 | .82 | .92 | .61 | .67 |
| F. 4 | .39 | .70 | .67 | .64 | .56 | .63 | .63 | .67 | .71 | .78 |
| F. 5 | .95 | .02 | .02 | .24 | .20 | .57 | .02 | .02 | .40 | .02 |
| F. 6 | .02 | .65 | .02 | .25 | .23 | .02 | .72 | .02 | .01 | .73 |
| F. 7 | .02 | .02 | .73 | .20 | .25 | .02 | .02 | .66 | .48 | .02 |
| F. 8 | .82 | .02 | .54 | .53 | .49 | .61 | .02 | .80 | .75 | .02 |
| F. 9 | .02 | .56 | .02 | .28 | .24 | .02 | .82 | .02 | .01 | .76 |

Table 9: Average Synaptic Weight from One Feature to Another

As expected, weights between mutually exclusive feature like *0-1* or *0-6* are low, either .02 or .01. Inhibitory weights (not shown here) are also closer to -1 between these mutually exclusive features, while co-occurring features have inhibitory weights much nearer 0. So, exclusive features inhibit each other. However, unlike the calculated Hopfield nets, the weights are more complete. In the Hopfield case, we knew feature *0* would inhibit feature *1*, but omitted feature *0* inhibiting feature *6*. Learning the weights has enabled us to be more complete.

The weights from feature *3* and *4* are weighted toward co-occurrence. Weights to each other are around .6, to omnivore around .5, and to all others around .2. Weights to *3* and *4* are quite large, above .6; this is the effect of the compensatory modifier. These neurons have more strength to give, so they activate *3* and *4* while getting comparatively less activation back. This is the basis of the hierarchical effect. Feature *8* falls somewhere between *3* and *4*, and the other features sending more activation than *3* and *4* to those it activates but less than the others.

## 6 Discussion and Future Work on Hierarchies

These simulations show a hierarchy of categories emerging via Hebbian learning. The subcategory CAs emerge as prior work in categorising with CAs would suggest. Additionally, a supercategory also emerges containing the most common features in the subcategories.

This allows a more flexible categoriser. If a new member of a new subordinate category is presented (say a *Bear*), the superordinate category will be activated. Eventually, the new subordinate category may be learned.

Similarly, default reasoning can be used for the new *Bear* subordinate category. In the absence of any evidence, the system would guess that the new item is an *Omnivore* because it is a *Mammal*.

The compensatory learning rule is essential to this process. In addition to providing flexible categories, it forces a neuron to concentrate on the most important neurons with which it interacts. This limits simulated epilepsy, but also supports the formation of the superordinate stable state.

There is a wide range of further work to be done on categorisation and hiearchies using CAs. CAs can be used for current AI tasks like classification. We need to improve our theoretical understanding of CA systems. We are hoping to use hierarchical CAs as a mechanism for resolving prepositional phrase ambiguity in natural language processing. Further in the future we are hoping to explore basic level effects, and to use our model to simulate timing data in psychological categorisation tasks.

We have begun exploration of the use of CAs for some standard classification tasks from a standard repository [Blake and Merz, 1998]. Our work on some simple classification and information retrieval (IR) tasks has been successful and are currently working on a manuscript. Both the IR and classification tasks were natural categories with polymorphic categories. This work is far from complete and we need to extend this work to more complex tasks, including linearly separable categories and highly interleaved categories. It is likely that a greater understanding of CA dynamics is needed before CAs can be applied to the full range of current data-mining problems.

More worrisome is the biologically implausible nature of the training regime used for the simulation of section 5. During training, we have presented each pattern for only one cycle. If we use a more plausible mechanism, presenting it for 20 cycles and allowing run on, we get only one CA forming. It is the superordinate category including the features of all subordinate categories. This is theoretically legitimate, but is not the desired result.

We plan on exploring this problem next. One aspect of the problem is that once a subordinate neuron is fired by the superordinate CA, it is rapidly recruited into it. This is the smallest version of the Mule problem mentioned in section 2.3. The problem has occurred in earlier experiments and has been solved by reducing the learning rate [Huyck, 2002a], and preventing the subcategories being recruited by the supercategories. Perhaps a better mechanism would allow the superordinate category to form first, but then allow the subordinate CAs to fractionate off.

This problem also relates to the long term dynamics of recruitment and fractionation. We are in the process of developing a computational theory of CA development. This will be related to the theory of Hopfield nets but is separate.

It would be simple to scale this work to categorise several superordinate and subordinate sets. Less clear is how well this will work for several hierarchical levels of categories. We are hoping to work on the prepostional phrase attachment task from natural language processing. This involves the understanding of the semantics of the verb, noun and prepostional phrases in a sentence like  I saw the girl with the telescope. [Hindle and Rooth, 1993]. We are hoping that hierarchical categories of these phrases can be used to learn the semantic relationships used in selecting the correct attachment. We will be use our novel machine learning technique to this problem.

Eventually we would like to look into basic level effects and timing data. We abide by Hebb's original idea that CAs are the basis of human concepts. Consequently, we would like to develop neural models that implment CAs to account for the full range of categorisation data.

## 7 Ongoing Work, Future Work, and Conclusion

We have proposed CAs as a good level to model cognition, and shown among other things that a system using CAs can form hierarchies. However, to be the basis of cognition, CAs must form many other structures.

Hierarchies are related to associative memory. We are currently working on associative memory including the related task of priming. In the next few years, we hope to move from theoretical work to practical data-mining and psychological simulation tasks involving categorisation, associative memories and hierarchies.

A related question is how CAs grow into unstimulated regions. Initially, stimulus only comes from the environment and most neurons in the mammalian brain are not directly connected to the environment. Large initial weights may help, but we have begun exploration of spontaneous neural activity to drive neural recruitment [Huyck and Bowles, accp]. There is evidence [Abeles et al., 1993, Bevan and Wilson, 1999] that neurons spontaneously fire without external input; our work relates to Amit and Brunel's [Amit and Brunel, 1997]. When combined with a compensatory learning mechanism, this allows unstimulated neurons to be recruited into CAs. This process must relate to category formation, and relationships between categories.

Until recently, we have directly stimulated neurons to simulate external activity, but recently, we have begun to explore how simple sensors can be integrated into the system. Our current work is on numbers for data-mining, but we hope to develop more sophisticated mechanisms. We draw hope from the extensive neurophysiological work on vision and other sensing. Similarly, we would also like to integrate actuators into the model.

The final great hurdle is the variable binding problem [VonDerMalsberg, 1986, Sun, 1995]. The above work deals with long-term relationships. The binding problem asks how very short-term relationships can be formed. Current thinking is that this can be with models of synchronous firing [Eckhorn et al., 1988, Sougne, 2001]. We are currently uncommitted but feel that there are probably several types of variable binding. Each type may be driven by the different CA dynamics caused by the brain areas they inhabit. For instance, episodic memory may be driven by synaptic change [Shastri, 2001] that is quicker than semantic memory, but still long-lasting. This difference may be due to the vast numbers of synapses in the hippocampus (10 times more per neuron than in the cortex), the area that is so essential to episodic memory [Shastri, 2001]. Variable binding is probably also used in simple natural language processing. In this system, the memories may be combined via synaptic modification, but these are more like registers which are quickly over-written. Clearly this is highly speculative, but it can be seen how variable binding might be accomplished by simply changing the learning rate for different CA based memory systems.

Finally, this associative memory can be combined with variable binding to form a significantly different computational model. It would be Turing complete because it could implement rules and thus arbitrarily complex functions, but it would be based more on mammalian processing. Different subsystems could vary certain parameters, but this basic CA architecture could be used to develop a more complete cognitive architecture, which is the ultimate goal of this work.

**Acknowledgements:**

This work was supported by EPSRC grant GR/R13975/01. We would also like to thank Marius Usher for comments on a draft of this paper.

## References

[Abbott et al., 1997] Abbott, L., A. Verela, K. S., and Nelson, S. (1997). Synaptic depression and cortical gain control. *Science*, 275:220–4.

[Abeles et al., 1993] Abeles, M., Bergman, H., Margalit, E., and Vaadia., E. (1993). Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *Journal of Neurophysiology*, 70(4):1629–38.

[Abu-Mostafa and Jaques., 1985] Abu-Mostafa, Y. and Jaques., J. S. (1985). Information capacity of the hopfield model. *IEEE Trans. Inform. Theory*, 31(4):461–4.

[Aleksander, 1989] Aleksander, I. (1989). *An Introduction to Neural Computation.* Van Nostrand Reinhold Co.

[Amit and Brunel, 1997] Amit, D. and Brunel, N. (1997). Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral Cortex*, 7:237–52.

[Amit and Mongillo, 2003] Amit, D. and Mongillo, G. (2003). Spike-driven synaptic dynamics generating working memory states. *Neural Computation*, 15:565–96.

[Bevan and Wilson, 1999] Bevan, M. and Wilson, C. (1999). Mechanisms underlying spontaneous oscillation and rhythmic firing in rat subthalamic neurons. *Journal of Neuroscience*, 19:7617–28.

[Blake and Merz, 1998] Blake, C. and Merz, C. (1998). Uci repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html].

[Bower and Beeman, 1995] Bower, J. and Beeman, D. (1995). *The Book of GENESIS.* Springer-Verlag.

[Bratenberg, 1989] Bratenberg, V. (1989). Some arguments for a theory of cell assemblies in the cerebral cortex. In Nadel, Cooper, Culicover, and Harnish, editors, *Neural Connections, Mental Computation.* MIT Press.

[Brunel, 1996] Brunel, N. (1996). Hebbian learning of context in recurrent neural networks. *Neural Computation*, 8:1677–1710.

[Chown, 1999] Chown, E. (1999). Making predictions in an uncertain world: Environmental structure and cognitive maps. *Adpative Behavior*, 7.

[Churchland and Sejnowski, 1992] Churchland, P. S. and Sejnowski, T. (1992). *The Computational Brain.* MIT Press.

[Collins and Loftus, 1975] Collins, A. and Loftus, E. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, 82:6:407–28.

[Eccles, 1986] Eccles, J. (1986). Chemical transmission and dale's principle. *Prog. Brain Research*, 68:3–13.

[Eckhorn et al., 1988] Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., and Reitboeck, H. (1988). Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, 60:121–30.

[Elman, 1990] Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

[Fransen et al., 1992] Fransen, E., Lanser, A., and Liljenstrom, H. (1992). A model of cortical associative memory based on hebbian cell assemblies. In Niklasson, L. and Boden, M., editors, *Connectionism in a Broad Perspective*, pages 165–71. Ellis Horwood Springer-Verlag.

[Fu, 2003] Fu, L. L. (2003). *An Analysis of Hebb's Cell Assembly as a Mechanism for Perceptual Generalisation.* PhD thesis, University of Michigan, Ann Arbor.

[Grossberg, 1987] Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–63.

[Hanson and Bauer, 1989] Hanson, S. and Bauer, M. (1989). Conceptual clusetring, categorization, and polymorphy. *Machine Learning*, 3:343–72.

[Hanson and Negishi, 2002] Hanson, S. and Negishi, M. (2002). On the emergence of rules in neural networks. *Neural Computation*, 14:2245–68.

[Hebb, 1949] Hebb, D. O. (1949). *The Organization of Behavior.* J. Wiley & Sons.

[Hetherington and Shapiro, 1993] Hetherington, P. and Shapiro, M. (1993). Simulating hebb cell assemblies: the necessity for partitioned dendritic trees and a post-net-pre ltd rule. *Network: Computation in Neural Systems*, 4:135–53.

[Hindle and Rooth, 1993] Hindle, D. and Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, 19:1:103–20.

[Hopfield, 1982] Hopfield, J. (1982). Neural nets and physical systems with emergent collective computational abilities. *Proc. of the Nat. Academy of Sciences USA*, 79:2554–8.

[Horn and Usher, 1989] Horn, D. and Usher, M. (1989). Neural networks with dynamical thresholds. *The American Physical Review*, 40:2:1036–44.

[Huyck, 2002a] Huyck, C. (2002a). Cell assemblies and neural network theory: From correlators to cell assemblies. Technical report, Middlesex University.

[Huyck, 2002b] Huyck, C. (2002b). Describing low level psychological phenomena through cell assemblies. In Bullinaria and Lowe, editors, *Proceedings of the Seventh Neural Computation and Psychology Workshop: Connectionist Models of Cognition and Perception.*

[Huyck and Bowles, accp] Huyck, C. and Bowles, R. (accp). Spontaneous neural firing in biological and artificial neural systems. *Journal of Cognitive Systems.*

[Huyck and Mitchell, 2002] Huyck, C. and Mitchell, I. (2002). Cell assemblies, self organising maps and hopfield networks. In *Proceedings of the Sixth International Conference on Cognitive and Neural Systems.*

[Huyck, 2001] Huyck, C. R. (2001). Cell assemblies as an intermediate level model of cognition. In Wermter, S., Austin, J., and Willshaw, D., editors, *Emerging Neural Architectures based on Neuroscience.* Springer.

[Kaplan et al., 1991] Kaplan, S., Sontag, M., and Chown, E. (1991). Tracing recurrent activity in cognitive elements(trace): A model of temporal dynamics in a cell assembly. *Connection Science*, 3:179–206.

[Kieras et al., 1997] Kieras, D., Wood, S., and Meyer, D. (1997). Predictive engineering models based on the epic architecture for a multimodal highperformance human-computer interaction task. *ACM Transactions on Computer Human Interaction*, 4:3:230–75.

[Newell, 1990] Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.

[Pollack, 1990] Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence*, 46:77–105.

[Quillian, 1967] Quillian, M. (1967). Word concepts: A theory of simulation of some basic semantic capabilities. *Behavioral Science*, 12:410–30.

[Rochester et al., 1956] Rochester, N., Holland, J., Haibt, L., and Duda, W. (1956). Tests on a cell assembly theory of the action of the brain using a large digital computer. *IRE Transaction on Information Theory*, 2:80–93.

[Rosch and Mervis, 1975] Rosch, E. and Mervis, C. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605.

[Rupin and Usher, 1990] Rupin, E. and Usher, M. (1990). An attractor neural network modle of semantic fact retrieval. *Network*, 1:325–44.

[Sakurai, 1998] Sakurai, Y. (1998). The search for cell assemblies in the working brain. *Behavioural Brain Research*, 91:1–13.

[Shastri, 2001] Shastri, L. (2001). Smriti: a computational model of episodic memory formation inspired by the hippocampal system. In Wermter, S., Austin, J., and Willshaw, D., editors, *Emerging Neural Architectures based on Neuroscience*. Springer.

[Shepard et al., 1961] Shepard, R., Hovland, C., and Jenkins, H. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied*, 517.

[Sougne, 2001] Sougne, J. (2001). Binding and multiple instantiation in a distributed network of spiking neurons. *Connection Science*, 13:99–126.

[Sun, 1995] Sun, R. (1995). Robust reasoning: integrating rule-based and similarity based reasoning. *Artificial Intelligence*, 75:241–95.

[Tsodyks et al., 1998] Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. *Neural Computation*, 10:821–35.

[VonDerMalsberg, 1986] VonDerMalsberg, C. (1986). Brain theory. In Palm, G. and Aersten, A., editors, *Emerging Neural Architectures based on Neuroscience*, pages 161–76. Springer-Verlag.

[Wickelgren, 1999] Wickelgren, W. (1999). Webs, cell assemblies, and chunking in neural nets. *Canadian Journal of Experimental Psychology*, 53:1:118–131.

[Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical Investigations.* Blackwell, Oxford.