Conflict Resolution and Learning Probability Matching in a Neural Cell-Assembly Architecture

Roman V. Belavkin[a,*], Christian R. Huyck[a]

[a] *School of Engineering and Information Sciences, Middlesex University, London NW4 4BT, UK*

Abstract

Donald Hebb proposed a hypothesis that specialised groups of neurons, called cell-assemblies (CAs), form the basis for neural encoding of symbols in the human mind. It is not clear, however, how CAs can be re-used and combined to form new representations as in classical symbolic systems. We demonstrate that Hebbian learning of synaptic weights alone is not adequate for all tasks, and that additional meta-control processes should be involved. We describe an earlier proposed architecture (Belavkin & Huyck, 2008) implementing an adaptive conflict resolution process between CAs, and then evaluate it by modelling the probability matching phenomenon in a classic two-choice task. The model and its results are discussed in view of mathematical theory of learning and existing cognitive architectures.

Keywords: artificial intelligence, cognitive science, neuroscience, decision making, intelligent agents, learning, Bayesian modeling, computational neuroscience, human experimentation

1. Introduction

There exists a variety of artificial systems and algorithms for learning and adaptation. Most of them can be classified as sub-symbolic (e.g. Bayesian and connectionist networks) or symbolic systems (e.g. rule-based systems). Known natural learning systems use neural networks, and therefore can be classified as using sub-symbolic computations. A distinguishing feature of the human mind, however, is the ability to use rich symbolic representations and language.

From an information-theoretic point of view, symbols are elements of some finite set that are used to encode discrete categories of sub-symbolic information. They enable communication of information about the environment or a complex problem in a compact form. One obvious benefit is that with language, one can learn not only from one's own experience, but also from experiences of others. The benefits of reading a guidebook before going abroad are obvious.

The duality between sub-symbolic and symbolic approaches has been studied in cognitive science. There exist sub-symbolic (i.e. connectionist), symbolic (e.g. Soar, Newell, 1990) and hybrid architectures (e.g. Act-r, Anderson & Lebiere, 1998) for cognitive modelling. These different approaches, however, have not yet explained where the symbols are in the human mind, or how the brain implements symbolic information processing (though see Jilk, Lebiere, O'Reilly & Anderson, 2008).

It was proposed by Hebb (1949) that symbols are represented in the brain not by individual neurons, but by correlated activities of groups of cells, called *cell-assemblies* (CAs). The Cell-Assemblies robot project (CABot) set out to test and demonstrate this idea in an engineering task by building an artificial agent, situated in a virtual environment, capable of complex symbolic processing, and implemented entirely using CAs of simulated neurons. Some of the objectives have already been achieved and reported elsewhere (e.g. Huyck & Belavkin, 2006; Huyck, 2007; Belavkin & Huyck, 2008). The architecture and some of these works will be discussed in the next section.

The work described in this paper is concerned with a particular aspect of the project — a stochastic conflict resolution and meta-control mechanism that modulates Hebbian learning to allow for re-use and combination of CAs into new representations, such as learning logical implications (i.e. procedural knowledge). As will be discussed in this paper, this cannot be achieved by using a Hebbian learning mechanism alone. A unique contribution of this work is evaluation of the meta-control mechanism in a cognitive model of the probability matching phenomenon in a two-choice experiment (Friedman, Burke, Cole, Keller, Millward & Estes, 1964). The results suggest that a proposed mechanism is a plausible model. Some neurophysiological studies and hypotheses about the brain circuitry will be discussed supporting the biological plausibility of the architecture.

In the next section, we describe briefly the neural model that is used in our architecture, how simulated neurons form cell assemblies and how we use them to test the CA hypothesis of symbolic processing. Then we discuss the problem of learning connections between existing CAs. This process is important for learning new symbolic knowledge by re-using and combining existing symbolic representations. In particular, we focus on the problem of learning the correct set of rules from the set of all possible rules connecting existing antecedents and conse-

[*] Corresponding author

Email addresses: R.Belavkin@mdx.ac.uk (Roman V. Belavkin), C.Huyck@mdx.ac.uk (Christian R. Huyck) quents. Here we draw the parallel with the ACT-R conflict resolution mechanism. Using a mathematical theory of stochastic learning, we argue that utility (or reinforcement) and stochastic noise are essential components of this process, and that they are not included in the Hebbian principle for adaptation of synaptic weights. The neural architecture implementing the utility-based stochastic learning of the connections between CAs is explained in Section 4, and its performance is demonstrated in an experiment. Section 5 presents the same architecture simulating the probability matching phenomenon as observed by Friedman et al. (1964), and a comparison with the hybrid model based on the ACT-R architecture is drawn. We then summarise contributions of this work and discuss its potential future development.

2. Cell-Assemblies as the Basis of Symbols

In this section, we outline some of the basic features of the CABOT architecture as well as the CA hypothesis.

2.1. *Neural Information Processing in* CABOT

It is widely accepted that human cognition is the result of the activity of approximately 10^{11} neurons in the central nervous system that interact with each other as well as with the outside world via the peripheral nervous system. Biological neurons are complex systems, and they have been modelled with various levels of details (McCulloch & Pitts, 1943; Hodgkin & Huxley, 1952). In our system, we use fatiguing, leaky, integrate and fire (fLIF) neurons.

The 'integrate and fire' component is based on the classical idea that the neuron 'fires' (or spikes) if its action potential, A, exceeds a certain threshold value θ:

$$y = \begin{cases} 1 & \text{if } A \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

The action potential, A, is a function of the integral (inner product) $\langle x, w \rangle = \sum_{i=1}^{k} x_i w_i$ of the stimulus (pre-synaptic) vector $x \in \mathbb{R}^k$ and the synaptic weight vector $w \in \mathbb{R}^k$ of the neuron. Here, $\mathbb{R}^k$ is a k-dimensional Euclidean space, where k is the number of synapses to the neuron. We use binary signals, and therefore x is a k-dimensional binary vector.

The 'leaky' property refers to a more complex (non-linear) dependency of the action potential on the pre- and post-synaptic activity:

$$A_{t+1} = \frac{A_t}{d_t} + \langle x_t, w_t \rangle\,, \quad d_t = \begin{cases} \infty & \text{if fired } (y_t = 1) \\ d \geq 1 & \text{otherwise} \end{cases} \qquad (1)$$

Thus, the action potential is accumulated over several time moments if the neuron does not fire. Parameter $d \geq 1$ allows for some of this activation to 'leak' away. This is the LIF model (Maas & Bishop, 2001).

The 'fatigue' property refers to a dynamic threshold that is defined as follows:

$$\theta_{t+1} = \theta_t + F_t\,, \quad F_t = \begin{cases} F_+ \geq 0 & \text{if fired } (y_t = 1) \\ F_- < 0 & \text{otherwise} \end{cases} \qquad (2)$$

where values F_+ and F_- represent the *fatigue* and fatigue *recovery* rates. Thus, if a neuron fires at time t, its threshold increases, and it is less likely to fire at time $t + 1$.

The fatiguing and leaky properties of the neural model allow for a non-trivial dynamics of the system. Repetitive stimulation of excitatory synapses increases the probability of a neuron to fire, even if the weights have small (positive) values. On the other hand, if the neuron fires repetitively, its threshold increases reducing the chance of it firing again. Thus, frequencies of pre- and post-synaptic activities are important factors in our system.

The weights of a neuron adapt according to the following compensatory rule (Huyck, 2007):

$$\Delta w_{ij} = \begin{cases} \alpha(1 - w_{ij})e^{W_B - W_i} & \text{if } x_t = 1, y_t = 1 \\ -\alpha w_{ij} e^{W_i - W_B} & \text{if } x_t = 1, y_t = 0 \end{cases}$$

where $\alpha \in [0, 1]$ is the learning rate parameter, W_B is a constant representing the average total synaptic strength of the pre-synaptic neuron, and W_i is the current total synaptic strength (see Huyck, 2007, for details). Note that absolute values of the weights w_{ij} here are in the interval $[0, 1]$, and the rule ensures that the new weight depends on the correlation between the pre-synaptic, x_t, and the post-synaptic, y_t, activities, which is an implementation of the Hebbian principle.

The above described properties are known characteristics of biological neurons, and our model is a compromise between computational efficiency and biological plausibility that is important for the emerging dynamics that we discuss below.

2.2. *Neural Cell-Assemblies*

Networks of neurons can be used as general function approximators and applied in a variety of tasks including control, pattern recognition and classification. Our system, CABOT, uses recurrent, partially connected networks (a mesh) of fLIF neurons with a largely pre-defined topology, which is usually determined by a specific task. For example, CABOT was used to develop an agent situated in a virtual environment, and it used many sub-systems including simulation of visual cortex, action-selection and natural language parsing to process text commands from a user. The non-linearity of the cells and the topology of the network lead to a complex dynamics of the system similar to that in attractor nets (e.g. Hopfield, 1982), where some of the states are more probable. These more 'stable' states can be characterised by groups of neurons that remain significantly more active than the other neurons in the system. Following Hebb, we refer to such reverberating groups of cells as *cell-assemblies* (CAs).

In our system, the formation of CAs depends on the topology of the network, and it is facilitated by the adaptation of the weights between connected cells. Therefore, CAs can be used for pattern classification of sensory stimuli (i.e. patterns from external connections). This leads to functional *specialisation* of neurons in the network based on CAs — two cells are functionally different if they belong to different CAs, even though they are similar architecturally. Such specialisation is observed in many neural networks, such as in self-organising maps (Kohonen, 1982) and particularly in the human brain. Note that CAs are not necessarily disjoint sets of cells. A single cell may be a member of several overlapping CAs. This feature can be used to encode hierarchies of patterns (Huyck, 2007).

An important property of CAs' dynamics is their persistence (Kaplan, Sontag & Chown, 1991). When enough neurons fire to start the reverberating circuit, the CA ignites. Once ignited, the activity within the cells in a CA may be sufficient to support itself. There are several parameters that influence this effect in our system. These are parameters of the network topology and parameters of the fatiguing and leaky properties of the cells. The main network topology parameters define the total number of cells in the network (module), sparsity of the connections, the default connection strength and the likelihood of a cell being inhibitory. The main parameters of the cells are the initial activation threshold θ, the decay parameter d controlling the activation leak in equation (1), the fatigue F_+ and recovery rate F_- parameters in the threshold equation (2).

After defining the network topology, the parameters of the cells can be used to achieve the desired persistence of the CAs in the network. For example, one can increase or decrease the recovery parameter F_- to achieve longer or shorter persistence. Table 1 in Section 4 lists the main parameter settings of the four networks used in the described experiments.

Note also that a CA's activity does not only depend on the external patterns, but also on the activity of other CAs in the system as they can ignite and extinguish each other. Thus, the activity of several CAs can be characterised by different patterns of ignition order and so on. The ignition of a CA in the system can be interpreted as an activation of a certain symbol in the system. It was demonstrated earlier that such state transitions in the system of CAs are sufficiently controllable to implement a broad range of tasks simulating symbolic processing that will be discussed below.

2.3. Symbols and Human Cognition

Many models of biological neurons suggest that synaptic weights may represent the memory for statistical and sub-symbolic information of the stimulus. In particular, in many algorithms for training artificial neural networks (e.g. Oja, 1982), the weight vector $w \in \mathbb{R}^k$ adapts to one of the principal eigenvectors of the covariance matrix $E\{xx^\dagger\}$ of the input vectors $x \in \mathbb{R}^k$ that have been observed. On the other hand, human cognition, and a great deal of human knowledge in particular, is encoded using symbolic representations, and the link between the symbols and neural models is less clear.

It was proposed by Hebb (1949) that CAs may be considered as the neural basis of symbols. Indeed, CAs can be easily mapped to some discrete categories of the stimuli, and their activity patterns can model serial processing typical for symbolic algorithms. Testing this hypothesis experimentally is one of the main objectives of the CABot project. However, many challenges had to be overcome to make a purely CA-based system performing some non-trivial symbol processing task.

Previously, we reported a system performing a counting task that consisted of 7 modules and 40 CAs (Huyck & Belavkin, 2006). A more recent system, CABot 2, is an artificial agent functioning in a virtual 3D environment that has a model of visual information processing, and is capable of natural language processing and action-selection (Belavkin & Huyck, 2008). One of the advantages of such a CA-based architecture is that neural CAs, that we associate with symbolic representations, also integrate all the sensory (i.e. sub-symbolic) information, which can be a natural solution to the *symbol grounding* problem. An associated phenomenon of symbolic processing is *grounding transfer* — combination and re-use of existing symbols to form new representations (Jamshed & Huyck, 2009).

The re-use of symbols is also important for learning procedural knowledge. Indeed, a logical implication (i.e. a production rule) may use combinations of symbols both in the antecedent and the consequent, and generally there are many more possible combinations than the number of rules that are actually used. Hybrid architectures, such as Act-r, rely on statistical (sub-symbolic) computations to 'filter' out the unwanted rules in the process called *conflict resolution*. In CABot, associations between CAs are learnt due to the Hebbian learning mechanism. However, as will be pointed out below, this mechanism alone is not sufficient to implement learning of particular associations between CAs representing existing symbols. To resolve this problem, an additional stochastic meta-control mechanism, moderating the Hebbian learning, has been introduced (Belavkin & Huyck, 2008). Here, we use this mechanism to model the probability matching in a classical two-choice experiment, and in this way evaluate its plausibility.

3. Learning as Optimisation with Information Constraints

Learning as a process can be understood in different ways, but learning systems in general can be characterised by some optimisation criteria and information constraints. The latter characteristic facilitates the mathematical description and analysis of learning (Belavkin, 2009, 2010) based on information value theory, as opposed to a reinforcement learning approach (Kaelbling et al., 1996; Sutton & Barto, 1998) based on the optimal control theory. We begin by describing the problem of optimal choice in a two-choice task, which was used in the probability matching experiments described later. Then we outline a few theoretical results from the theory of optimisation with information constraints.

3.1. Two-Choice Task

Let s, a_1 and a_2 be three symbols, where s represents a stimulus (antecedent), and a_1, a_2 represent two alternative responses (consequents). Thus, we have a conflict between two implications $s \to a_1$ and $s \to a_2$ shown on the diagram below

$$\begin{tikzcd}
 & s \arrow[dl] \arrow[dr] & \\
a_1 & & a_2
\end{tikzcd}$$

This is the simplest two-choice task (a more complex two-choice task may involve a set of different stimuli). The choice of a_1 or a_2 is followed by some reinforcement events E that may have different utility values (e.g. a success after choosing a_1 or a failure after choosing a_2). If the utility values $u(s,a)$ were known, then one would prefer to choose a_1 in presence of s (i.e. preferring rule $s \to a_1$) if and only if $u(s,a_2) \leq u(s,a_1)$. If the positive reinforcement event is not deterministic, but occurs with probability $P(E) = \pi \in [0,1]$, then one can use the expected utility to choose a_1 or a_2. This is the maximum expected utility principle, which is fundamental in theories of games, statistical decisions and control (von Neumann & Morgenstern, 1944; Wald, 1950; Bellman, 1957), and it also has been used in the Act-r conflict resolution to model properties of human choice behaviour.

If, however, the utility function or the probability distribution is not known, then one needs to learn them from experience. As demonstrated in many experiments with animals and human participants, the frequency of choosing a_1 adapts to the probability π of reinforcement with high utility — a phenomenon referred to as the *probability matching*. This phenomenon can be explained based on the theories of optimal statistical decisions and information value (Stratonovich, 1965).

3.2. *The Effect of Information Constraints*

Let us consider an abstract system with input $s \in S$ and output $a \in A$. Optimisation corresponds to some preference relation on the input-output pairs $(s,a) \in S \times A$. In a deterministic setting, this preference relation can be represented by a utility function $u : S \times A \to \mathbb{R}$, while in a stochastic setting, one considers conditional probability distributions $P(u \mid s,a)$ on values of utility $u \in \mathbb{R}$. If the utility function $u = u(s,a)$ or the joint distribution $P(u,s,a)$ is known (and hence $P(u \mid s,a)$), then given input s, the optimal output $\bar{a} \in A$ maximises the conditional expected utility:

$$\bar{a}(s) = \arg\max_{a \in A} \left\{ E_P\{u \mid s,a\} = \sum_u u\, P(u \mid s,a) \right\}$$

(assuming that the maximum exists). Here, $E_P\{\cdot\}$ denotes the expected value with respect to probability distribution P. Note that in the deterministic case, the conditional expected utility $E_P\{u \mid s,a\}$ coincides with the utility function $u = u(s,a)$. The *greedy* strategy $s \mapsto \bar{a}(s)$ of always choosing the optimal output can be expressed by the following conditional probability:

$$P(a \mid s) = \begin{cases} 1 & \text{if } a = \bar{a}(s) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In learning problems, either the utility function $u = u(s,a)$ or the distribution $P(u,s,a)$ is not known. Instead, one has some data from past occurrences of $(u,s,a) \in \mathbb{R} \times S \times A$. This represents a constraint on information, which can be taken into account explicitly as part of the optimisation problem: Maximise the expected utility subject to the constraint that information is not greater than some value I. We shall refer to the optimal value of the expected utility under the information constraint as $U(I)$ (called the *value of information* according to Stratonovich, 1965). Analytical solution of this optimisation problem leads to an optimal conditional probability that is different from the greedy strategy (3) (see Belavkin, 2009, 2010, for details). For an important case, when information is represented as the Kullback-Leibler divergence (e.g. Kullback, 1959) of posterior distribution P relative to the prior distribution Q, the optimal conditional distribution belongs to the one-parameter exponential family (also known as the Gibbs, Boltzmann or the 'soft-max' distribution):

$$P(a \mid s) = \frac{1}{Z}\, Q(a \mid s)\, \exp\{\beta\, \tilde{u}(s,a)\} \tag{4}$$

Here, $Q(a \mid s)$ is the distribution corresponding to the minimum of information (e.g. no data), β is the inverse 'temperature' parameter related to the amount of information, $\tilde{u}(s,a)$ is the empirical estimation of $u(s,a)$, and $Z = \sum_A Q(a \mid s)\, \exp\{\beta\, \tilde{u}(s,a)\}$ is defined from the normalisation condition $\sum_A P(a \mid s) = 1$.

It is important to note that the temperature parameter β^{-1} appears in the solution as the Lagrange multiplier related to the information constraint I. Its optimal value is given by the derivative of the optimal information value function $U(I)$ computed at I:

$$\beta^{-1} = U'(I) \tag{5}$$

Analysis shows that the function above is decreasing so that $\beta^{-1} \to 0$ (or $\beta \to \infty$) as information increases (Belavkin, 2010), and the exponential distribution (4) converges to the greedy strategy (3). However, if information is incomplete, then $\beta < \infty$ in distribution (4), and the optimal strategy is randomised.

Exponential distribution is often used for selecting the output of a system in machine learning and stochastic optimisation algorithms. It is also used in the Act-r cognitive architecture to model some stochastic properties of behaviour. In particular, it was used in the Act-r model of the two-choice experiment, discussed below. However, the 'temperature' parameter β^{-1} is usually set to some constant value or determined from some arbitrary 'annealing' schedule. The relation of β^{-1} to entropy of success in Act-r was proposed in (Belavkin, 2003), and it was shown that it improves the match between the models and data. The derivation of optimal function $\beta^{-1} = U'(I)$ can be found in (Stratonovich, 1965) and more generally in (Belavkin, 2009).

In the next section, we outline a neural CA-based architecture that uses the described above principles of utility and dynamic stochastic noise to learn preferable connections between existing CAs. This architecture can be used to learn new symbolic knowledge by re-using and combining existing CA-based symbolic representations. Similar to the Act-r conflict resolution mechanism, the use of utility and stochastic noise will allow us to simulate data from a probability matching experiment. An important difference of our neural implementation, however, is that the level of stochasticity depends on the experience resembling the antitone relation (5) between the temperature β^{-1} and the amount of information.

4. Stochastic Meta-Control of Hebbian Learning

The output of a neuron depends on its weight vector $w \in \mathbb{R}^k$, which, according to Hebb's hypothesis, adapts to the correlation between the pre- and post-synaptic activities in the past. It is attractive to conclude, therefore, that Hebbian learning is a particular implementation of statistical learning. However, the utility is clearly missing in this description of neural plasticity. What criteria does such a process of changing the weights optimise? If in a two-choice task the system accidentally chooses the 'incorrect' cell-assembly a_2, then the weights associating s with neurons in a_2 increase due to the correlation-based Hebbian learning. This can only increase the chance of $s \to a_2$ igniting in the future, even though the reinforcing event E following the choice of $s \to a_2$ has a low utility (i.e. a failure). Thus, some additional process should be involved to increase the chance of the 'correct' combination $s \to a_1$ after the reinforcing event E. Such a process appears to be especially useful if the CA-based symbolic representations, formed earlier, are to be re-used. Below we describe a neural implementation of such a meta-control of Hebbian learning based on the utility feedback (Belavkin & Huyck, 2008) following principles of statistical learning.

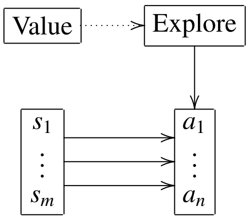

Figure 1: Components and connections of the Value and Explore modules controlling Hebbian learning of connections between CAs in modules S and A. Solid and dashed arrows show excitatory and inhibitory connections respectively.

Table 1: Main parameter settings in the four modules used in the simulations.

	Module			
Parameter	Stimuli	Responses	Value	Explore
Cells #	400	400	400	400
Connectivity %	40%	20%	40%	40%
Inhibitory %	20%	20%	30%	35%
Connection strength	.02	.02	.02	.02
Spontaneous activation	Off	Off	Off	On
Activation threshold θ	4.0	4.0	4.0	4.0
Decay d	1.5	2.0	2.0	2.0
Fatigue F_+	1.0	1.0	1.0	1.0
Recovery F_-	2.4	2.2	1.0	1.0
Learning (post-synaptic)	On	Off	Off	Off
Learning rate	.15			

4.1. CA Implementation

The meta-control process involves two specialised modules: Value and Explore. Their connections in the system are shown on Figure 1. Here, $S = \{s_1, \dots, s_m\}$ and $A = \{a_1, \dots, a_n\}$ are sets of CAs representing m stimuli and n responses respectively. Table 1 shows the main parameter settings for the four modules.

Initially, there are excitatory connections from every CA in S to all CAs in A, which means that all pairs (s, a) (i.e. all rules $s \to a$) are equally preferred. Thus, given input $s \in S$, any response $a \in A$ can be selected. However, due to Hebbian learning, the connection $s \to a$ is reinforced if a particular pair of CAs ignite together, giving the pair a higher chance to ignite together in the future. Thus, simply by virtue of Hebbian learning, the system can learn eventually to prefer some random pairs. The purpose of the Value and Explore modules is to make this process selective according to the feedback and its utility.

The Value module is a network of sparsely, recursively and randomly connected cells which form a single cell assembly. In the simulations described below, we used the Value module with 400 cells. The output activity of the Value module represents the utility value u associated with the pair (s, a) selected on the previous step. The input to the module can be configured according to the application (e.g. using sensory information).

The Explore module has a similar structure and number of cells to the Value module, but it contains cells that can be active without any external stimulation due to spontaneous activation. The purpose of this module is to randomise the activity of the response CAs (i.e. CAs in set A). The cells in the Explore module send excitatory signals to all CAs in module A, and the weights of these connections do not change. Thus, the activity in the Explore module can randomly trigger any response CA, and this process does not have a memory. The Explore module implements the effect of parameter β^{-1} in the exponential distribution.

The Value module sends inhibitory connections to the Explore module, so that high activity of the Value cells may shut down the activity in the Explore module. As a result, any response CA that has been ignited in module A will persist longer, because it is less likely to be shut down by another CA. Such a connectivity implements the following learning scheme: If a particular pair (s, a) results in a high utility value, then high activity of the Value module inhibits the Explore module, the responsible (s, a) pair is allowed to persist longer, and the $s \to a$ connection increases relative to others due to Hebbian learning.

Learning the 'correct' rules (subset $R \subset S \times A$) contributes to better performance of the system (i.e. higher expected utility). As a consequence, the average activity of the Value module increases with time, which in turn decreases the activity of the Explore module. This dynamic resembles the decrease of parameter β^{-1} as a function of information (5) making the system less random and more deterministic.

4.2. Performance

The working of the described meta-process has been implemented and tested in our system based on fLIF neurons. Here we report its performance in a fairly simple experiment of learning dichotomies. The code of the system and the described below experiment is available at

```
http://www.cwa.mdx.ac.uk/CABot/CANT.html
```

In this simple experiment, there are two or four CAs in the Stimulus module (s_1, s_2 or $s_1, \dots, s_4$) and two CAs in the Response module (a_1, a_2). It is assumed that these CAs correspond to existing representations of stimuli and responses in the given task. Each module consisted of 400 cells, with 200 cells in each CA. The modules were set up with connections with low weights from every stimulus CA to all response CAs, shown by four dashed arrows on the left diagram below. Table 1

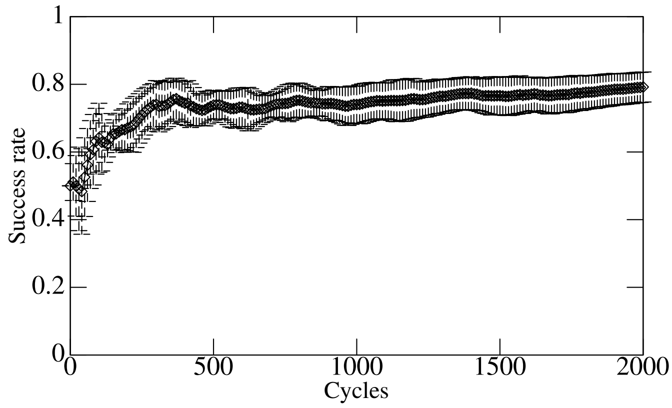

Figure 2: The proportion of correct response (ordinate) as a function of cycles (abscissa) in a 2 by 2 experiment. Error bars represent standard deviations from the mean in a series of trials.

lists settings for the main parameters in each module. The task was to learn two rules, shown by two solid arrows on the right diagram, by increasing the connection weights.

$$\begin{tikzcd}
s_1 \arrow[r, dashed] \arrow[dr, dashed] & a_1 & & s_1 \arrow[r] \arrow[dr, dash, dotted] & a_1 \\
s_2 \arrow[r, dashed] \arrow[ur, dashed] & a_2 & & s_2 \arrow[r] \arrow[ur, dash, dotted] & a_2
\end{tikzcd}$$

The training procedure consisted of a random presentation of an input pattern activating one of the stimulus CAs every 100 cycles. It takes on average 10–20 cycles for one of the response CAs to ignite, which is manifested by the high activity of cells in the CA. A threshold value can be used in the system to determine that a CA starts to ignite (10% in this simulation). The selected response is associated with the CA that has the highest activity. Although there can be an increase of activity in more than one CA, the inhibitory connections between them ensure that the CA with the highest activity quickly extinguishes other CAs.

If the correct response is selected, then the activation of the Value module inhibits the Explore module after another 10–20 cycles, and the activities of the stimulus and response CAs persist until a new pattern is presented. Otherwise, if an incorrect response is selected, the activity from the Explore module causes another response CA to ignite after approximately another 10–20 cycles.

Figures 2 and 3 show the proportion of the correct response (vertical axis) as a function of cycle number (horizontal axis) in a system with two and four input CAs respectively. The charts show the averaged results of several experiments, and the error bars show standard deviations. One can see that the system initially makes only half of the choices correctly. After about 2000 cycles, the proportion of correct choices increases to 70–90%. Note that the stimulus may change up to 10 times per 1000 cycles (i.e. every 100 cycles). Because the stimulus sequence was randomly generated in each experiment, there is a variance in the results represented by error bars on Figure 2 and 3. The increase of the probability of success corresponds to an increase in the empirical expected utility $\tilde{u}(s, a) = U$.

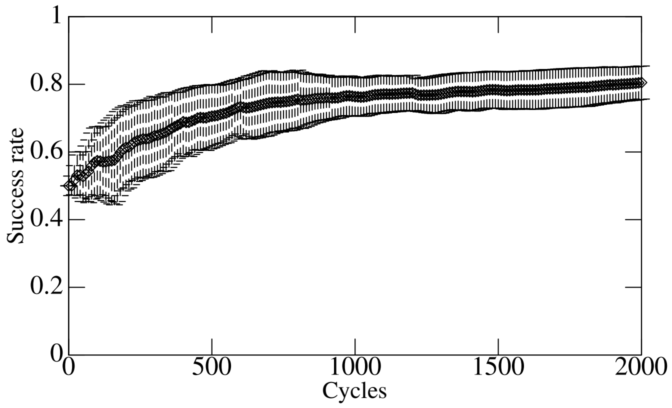

Figure 3: The proportion of correct response (ordinate) as a function of cycles (abscissa) in a 4 by 2 experiment. Error bars represent standard deviations from the mean in a series of trials.

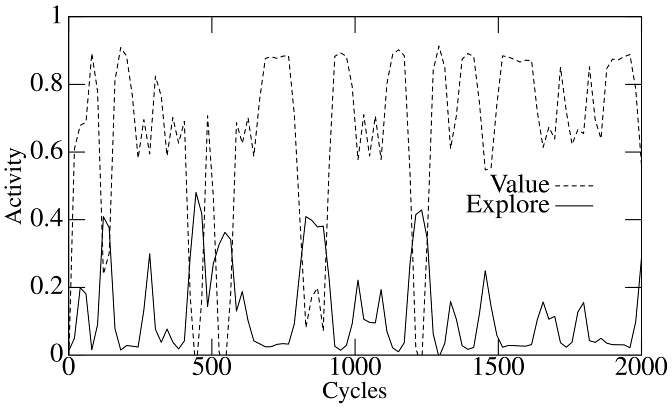

Figure 4: Activities of the Value and Explore modules in one experiment.

Figure 4 shows the percentage of neurons firing per cycle in the Value and the Explore modules in one of the experiments. One can clearly see that the activities anticorrelate; an increase in the Value module coincides with the decrease of the Explore module activity. More significantly, the chart shows that the average activity of the Value module increases as learning progresses, while the average activity of the Explore module decreases. As expected, this dynamic resembles the decrease of the noise temperature parameter β^{-1} as a function of information (5).

Because learning of the connections between the correct pairs of CAs depends on the differences between the times the 'correct' and 'incorrect' CAs persist in the system, the parameters controlling the dynamics of CAs in the modules may significantly influence the effect of the meta-process and the ability of the system to learn. For example, the values of the fatigue and fatigue recovery rates of the cells influence the persistence of the CAs as well as how rapidly one CA may extinguish another. Another important parameter is the connectivity of the cells in the module. The networks in the system are sparsely connected, and the average number of cells each cell is connected to can also significantly contribute to the behaviour of the CAs. The learning rate parameter of the Hebbian learning rule can also significantly influence the performance of the system. If the rate is too high, then association of an incorrect pair of CAs may occur before the meta-process has its effect. Table 1 lists settings of the main parameters in the system that was used in the experiments reported here.

5. Modelling Probability Matching

To test how adequately the above mechanism can represent properties of human cognition, we evaluate its performance against data from a classic two-choice experiment due to Friedman et al. (1964). The choice of this dataset was motivated not only by its quality and detailed description of the procedures, but also because it was used to 'calibrate' stochastic properties of other cognitive architectures, such as Act-r (Anderson & Lebiere, 1998). The complete description of the experiment and data can be found in the original paper (Friedman et al., 1964). Here we give a basic outline.

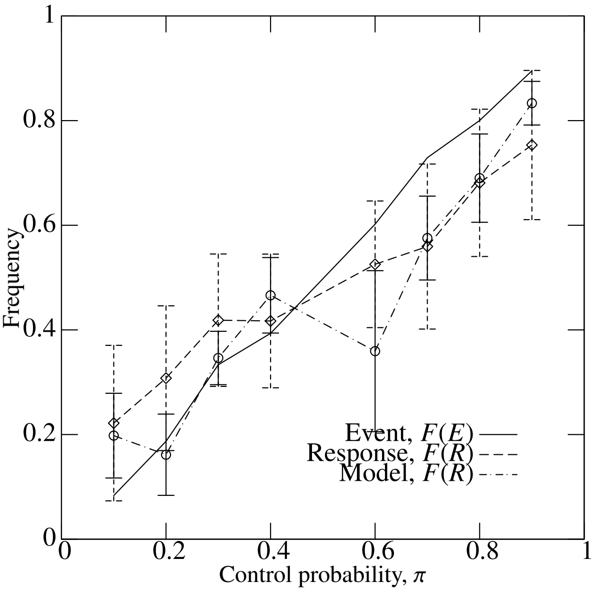

Figure 5: Frequency of response (ordinates) as a function of the probability of reinforcing this response (abscissae). Solid line show frequency of the reinforcing event, $F(E)$. Dashed lines show the average response frequencies, $F(R)$, in 48–trials of the participants in Friedman et al. (1964) and of the CABot model (RMSE=8.937%). The error bars represent standard deviations.

5.1. Experiment Description and Previous Work

In this experiment, participants were asked to select one of two responses on presentation of a stimulus. After the response was selected, a reinforcement event E occurred with probability $P(E) = \pi$ that did not depend on prior responses. Each participant had to perform this task in three sessions, each session consisted of 8 blocks, each block consisted of 48 trials. The probability $P(E) = \pi$ changed between each 48–trial block. This paper will report only simulations of results in Sessions 1 and 2. In these two sessions, blocks 1, 3, 5 and 7 had $P(E) = .5$, and blocks 2, 4, 6, and 8 were with $P(E) \in \{.1, .2, .3, .4, .6, .7, .8, .9\}$ that was assigned according to a random pattern. Thus, probability $P(E) = \pi$ was alternating between .5 and some value above or below .5 between 48-trial blocks. The data recorded the number of times Response 1 was chosen in each 48-trial block.

Figure 5 shows the results of these experiments, reported by Friedman et al. (1964). The charts show frequencies of Response 1, $F(R)$, and reinforcement events, $F(E)$, as functions of the control probability $P(E) = \pi$. One can see that the frequency of the reinforcement event $F(E)$ approximates the control probability $F(E) \approx P(E)$. The response frequency $F(R)$ also matches the probability $P(E)$, but it differs significantly at the lower and higher ends of the range: When $P(E)$ is low ($\pi = .1$), the participants overestimate the probability ($F(R) \geq P(E)$); when $P(E)$ is high ($\pi = .9$), the participants underestimate it ($F(R) \leq P(E)$). Thus, the response appears to be less certain than the reinforcing event.

As suggested by Anderson & Lebiere (1998), this experimental evidence indicates against using the greedy strategy (3) for choosing the response. The data was modelled in Act-r by sampling responses from exponential distribution with some $\beta^{-1} > 0$. This agrees with equations (4) and (5), where $\beta^{-1} \to 0$ only when there are no constraints on information. We now describe a model of this experiment implemented in CABot.

5.2. Model Description

The model used the same architecture, shown in Figure 1, and the same parameter settings, shown in Table 1, as in the experiments described earlier. Module S consisted of CAs representing one or more stimuli, and module A contained two CAs representing two alternative responses. Initially, there were excitatory connections with low weights from module S to all CAs in module A. The weights on these connections, however, could adapt according to a Hebbian rule increasing associations $s \to a$ between active CAs. The fatigue and leak parameters of the A network were set in such a way that CAs ignite only when an external stimulus is present (see Table 1). The CAs in module A inhibited each other so that only one of the CAs in A was active at any moment. The Explore module had excitatory connections with a small proportion of cells in module A. These connections were distributed uniformly, and the weights did not adapt. Spontaneous activation in the Explore module could randomly trigger either of the two response CAs in module A. The activity of the Explore module could be inhibited by the output activity from the Value module that was triggered in each trial according to probability $P(E) = \pi$ of the reinforcement event, controlled by the experimental sequence.

When the Explore module is inhibited by the reinforcing activity of the Value module, the active pair (s, a) is allowed to persist longer, strengthening the connections $s \to a$ relative to other connections. We found that the robustness of this effect depends on the time (i.e. number of cycles) these CAs are allowed to persist. In this model, it takes approximately between 10–20 cycles for a response CA in module A to ignite, and if the Explore module is active, then the response CA may change during another 10–20 cycles. In this experiment, the system ran for 100 cycles per trial which was sufficient for the control of learning to have a robust effect. The complete code of the simulation is available online from the CABot project website.

5.3. Results

The model was used to simulate Sessions 1 and 2 of eight 48-trial blocks each with variable control probabilities π (Friedman et al., 1964). The results comparing response frequency of the model with the data are shown on Figure 5. The model approximates the data fairly well (RMSE=8.937%) showing the probability matching effect that also overestimates and underestimates the low and high value of the control probability π respectively. Note that here we did not change any of the parameters of the system, such as those shown in Table 1, and perhaps the model could fit the data better with a different set of parameter values.

6. Conclusions

In this paper, we discussed the CABot architecture and some challenges associated with implementing the CA hypothesis of symbolic processing in the brain. The problem of re-use and combination of symbols, particularly in learning procedural knowledge, pointed at one significant shortcoming of the standard Hebbian learning mechanism — adaptation of weights based purely on correlations does not take into account the optimisation criteria that a system may have to satisfy. To resolve this problem, stochastic conflict resolution and meta-control of learning based on utility feedback was introduced into the system.

It is attractive to speculate about the existence of the Value and Explore modules in the brain. Some researchers have proposed that tonically active cholinergic neurons in the basal ganglia and striatal complex play an important role in conflict resolution and learning procedural knowledge (Granger, 2006). These neurons account for a small proportion of the connections that are quite uniform and non-topographic, and the activity of these neurons was suggested to play the role of stochastic noise, similar to the activity of cells in the Explore module (see Fig. 1). Interestingly, the activation of the tonically active cholinergic neurons is inhibited by the activation from the reward path, similar to the function of the Value module in our system. Other studies of mechanisms for exploratory behaviour in the brain are also in favour of the exponential distribution model (Daw, O'Doherty, Dayan, Seymour & Dolan, 2006).

Setting these speculations aside, this work has demonstrated that the proposed mechanism can be used for controlling Hebbian learning in networks of relatively biologically faithful models of neurons. The mechanism allows for selective learning of connections between specialised groups of cells (CAs), and following Hebb's hypothesis it shows not only that CAs can indeed be associated with symbols, but also shows how such representations can be re-used and combined to learn new knowledge. Simulation of the probability matching effect has demonstrated that the mechanism is also a plausible cognitive model of conflict resolution. We anticipate that the proposed architecture can also be used to model other psychological phenomena, such as the effect of reinforcement values on speed of learning, and this is one possible direction of our future research.

7. Acknowledgements

This work was supported by EPSRC grant EP/DO59720. The authors also would like to thank anonymous reviewers for a number of very useful comments.

References

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum.

Belavkin, R. V. (2003). *On Emotion, Learning and Uncertainty: A Cognitive Modelling Approach*. PhD thesis The University of Nottingham Nottingham, UK.

Belavkin, R. V. (2009). Bounds of optimal learning. In *2009 IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (pp. 199–204). Nashville, TN, USA: IEEE.

Belavkin, R. V. (2010). Information trajectory of optimal learning. In M. J. Hirsch, P. M. Pardalos, & R. Murphey (Eds.), *Dynamics of Information Systems: Theory and Applications*. Springer volume 40 of *Springer Optimization and Its Applications Series*.

Belavkin, R. V., & Huyck, C. (2008). Emergence of rules in cell assemblies of fLIF neurons. In *The 18th European Conference on Artificial Intelligence*.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, *441*, 876–879.

Friedman, M. P., Burke, C. J., Cole, M., Keller, L., Millward, R. B., & Estes, W. K. (1964). Two–choice behaviour under extended training with shifting probabilities of reinforcement. In R. C. Atkinson (Ed.), *Studies in Mathematical Psychology* (pp. 250–316). Stanford, CA: Stanford University Press.

Granger, R. (2006). Engines of the brain: The computational instruction set of human cognition. *AI Magazine*, *27*, 15–32.

Hebb, D. O. (1949). *The Organization of Behavor*. New York: John Wiley & Sons.

Hodgkin, A. L., & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, *117*, 500–544.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, *79*, 2554–8.

Huyck, C. (2007). Creating hierarchical categories using cell assemblies. *Connection Science*, *19*, 1–24.

Huyck, C., & Belavkin, R. V. (2006). Counting with neurons, rule application with nets of fatiguing leaky integrate and fire neurons. In D. Fum, F. D. Missier, & A. Stocco (Eds.), *Proceedings of the Seventh International Conference on Cognitive Modeling*. Trieste, Italy: Edizioni Goliardiche.

Jamshed, F., & Huyck, C. (2009). Grounding symbols: Labelling and resolving pronoun resolution with fLIF neurons. In *8th International Conference on Machine Learning and Applications*.

Jilk, D. J., Lebiere, C., O'Reilly, R. C., & Anderson, J. R. (2008). SAL: an explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, *20*, 197–218.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

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

Kohonen, T. (1982). Self–organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kullback, S. (1959). *Information Theory and Statistics*. John Wiley and Sons.

Maas, W., & Bishop, C. (2001). *Pulsed Neural Networks*. MIT Press.

McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior*. (1st ed.). Princeton, NJ: Princeton University Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, Massachusetts: Harvard University Press.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

Stratonovich, R. L. (1965). On value of information. *Izvestiya of USSR Academy of Sciences, Technical Cybernetics*, *5*, 3–12. In Russian.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press.

Wald, A. (1950). *Statistical Decision Functions*. New York: John Wiley & Sons.