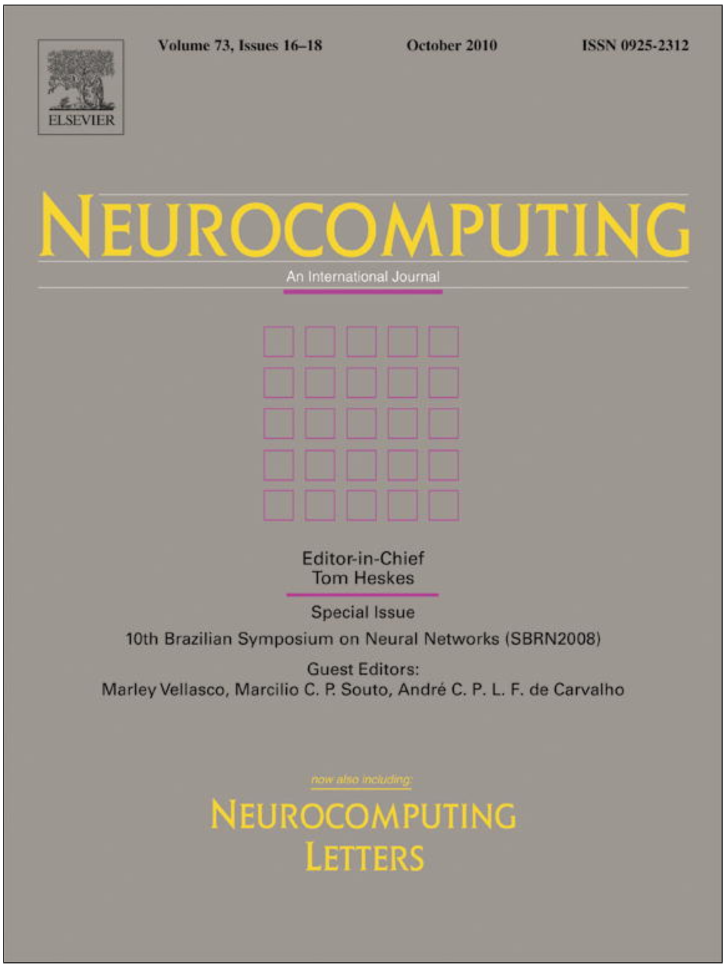
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



Volume 73, Issues 16–18 October 2010 ISSN 0925-2312

ELSEVIER

NEUROCOMPUTING

An International Journal

Editor-in-Chief
Tom Heskes

Special Issue
10th Brazilian Symposium on Neural Networks (SBRN2008)

Guest Editors:
Marley Vellasco, Marcilio C. P. Souto, André C. P. L. F. de Carvalho

now also including:

NEUROCOMPUTING
LETTERS

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Neurocomputing 73 (2010) 2928–2934

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A Pong playing agent modelled with massively overlapping cell assemblies

Kailash Nadh *, Christian R. Huyck

School of Engineering and Information Sciences, Middlesex University, London, United Kingdom

A R T I C L E   I N F O

Article history:
Received 22 March 2010
Received in revised form
1 July 2010
Accepted 30 July 2010
Communicated by G. Palm
Available online 26 August 2010

Keywords:
Neural networks
Fatiguing LIF neurons
Cell assemblies
Associative memory

A B S T R A C T

Cell assemblies (CAs) are central to many higher order cognitive processes such as perception, recognition and recollection. These processes stem from the fundamental cognitive tasks of memorisation and association, which CA models are able to perform with a viable degree of biological realism. This paper describes a virtual agent that uses CAs that emerge from fatiguing leaky integrate and fire neurons via learning from dynamic interaction. Learning is continuous and the topology is biologically motivated. The agent is able to visually perceive, learn and play a simplified game of Pong. It can learn from a user playing the game, or playing on its own. The agent's memories are encoded in the form of overlapping CAs that enable it to generalise its associations to account for previously unseen game moves. The trained agent hits the Pong ball correctly over 90% of the time. This work furthers the understanding of associative memory and CAs implemented in neural systems.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Associative memory is a fundamental cognitive process. Concepts in memory, and different types of associations between them are acquired via learning. Associations can vary semantically, for instance hierarchical associations, sequential associations, containment associations, spatial associations, and higher level semantic associations. These concepts and associations that form associative memory are critical to cognitive processing. Many connectionist accounts for associative memory exist, but cell assemblies (CAs) [1] provide a biologically and psychologically realistic basis for associative memory.

This paper describes a virtual agent modelled in simulated CAs capable of playing a simplified version of the popular arcade game, Pong. The agent is able to learn to play the game by observing a human play, or on its own. The agent learns to associate input from the environment with actions, thus learning game moves.

The agent is made entirely from fatiguing Leaky Integrate and Fire (fLIF) neurons that have a reasonable resemblance to biological neurons. Unlike many simulations, but like human neurons, learning remains on at all times.

The agent learns by encoding shared, overlapping associative memories. This allows generalisation behaviour to emerge, which further assists the agent in game play by enabling it to carry out actions in novel situations.

The paper is organised as follows: Section 2 overviews CAs; Section 3 discusses the fLIF neural network architecture where CAs emerge; Section 4 details the simulation and findings; and Section 5 discusses the impact of the simulation and highlights the findings.

2. Background

Human associative memory is a remarkable process. Throughout life, concepts continue to be acquired, learnt and associated. Any given concept is associated with many other concepts, and retrieval of an associated concept can be based on a combination of a range of base concepts with a range of contexts. Many associative memory models exist, e.g. [2–4], but Hebb's CA theory [1] provides an account that is supported by biological and psychological evidence. CAs exhibit dynamics that provide a unified explanation for long term memory and various short term memories as opposed to higher level box models of memory. There is extensive evidence that CAs are the basis of human associative memory and many other cognitive phenomena [5–10] and they have been used in computational models of associative memory [11–14].

2.1. Cell assemblies

CAs are reverberating circuits of neurons that form the neural basis of concepts. Hebb's CA theory postulates that objects, ideas, stimuli and even abstract concepts are represented in the brain by the simultaneous activation of large groups of neurons with high mutual synaptic strengths [1,11]. CAs are learnt by a Hebbian

* Corresponding author.
E-mail addresses: k.nadh@mdx.ac.uk (K. Nadh), c.huyck@mdx.ac.uk (C.R. Huyck).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.07.013

Author's personal copy

learning rule, whereby modifications in the synaptic transmission efficacy are driven by correlations in the firing activity of pre-synaptic and post-synaptic neurons [15]. When external stimuli are presented to a network, synaptic strength between neurons is adjusted so as to gain more strength if they undergo repeated and persistent activation or firing, gradually assembling themselves into a CA. A CA thus formed is bound to the object that generated the stimuli, a neural representation of that object; the behaviour of the CA causes psychological behaviour.

Such formation of CAs accounts for long term memory. When a stimulus similar to previously experienced stimuli occurs, it may excite a sufficient number of neurons of an existing CA to cause the spreading of activation within the CA, activating it fully due to recurrent activity and high mutual synaptic strength. The CA can remain active even after the stimulus is removed and this reverberating behaviour accounts for many kinds of short term memories. Ambiguous stimuli can cause contention between similar CAs, where lateral inhibition between those CAs causes competition, with one eventually winning. Thus, the CA hypothesis provides a functional account for a biologically viable associative memory mechanism.

There is extensive evidence of CAs in mammals based on a range of recording techniques and experimental paradigms (see [5,7] for reviews). CAs can also account for various psychological phenomena such as sensing [16] and determining action [17], besides different types of associative memory [18].

2.2. CAs and associative memory

Even though CAs account for memory formation, their precise neural dynamics are far from perfectly understood. In auto-associative memories, an initial state is allowed to settle into a stored memory, allowing subsequent noisy input to retrieve a stored pattern. The Hopfield Model, which is a network of units that are well connected with bidirectional weighted connections, that are used to store a set of binary patterns (typically using a Hebbian calculation), illustrates this property [19]. When an initial set of neurons is ignited, in a discrete version of the system, activation spreads through the system based on the weighted connections. In most cases, the system will settle into a stable state with no neurons switching between on and off. If the input pattern is close to a stored pattern, it will settle into that pattern's state, thus functioning as a content-addressable memory or an auto-associative memory.

Neurons may also belong to more than one CA. Hopfield patterns that share on-bits are models of CAs that share neurons. As mentioned in Section 2.1, neurons in a network may belong to different CAs, and if they are repeatedly co-activated by different versions of the same stimulus, they tend to become associated [1]. This is based on the notion that events that occur together repeatedly should somehow belong together. Every time these events occur in conjunction, they drive certain subgroups of neurons to fire in correlation, resulting in the association of the respective events [11]. A more complete review of CA based associative memory models is [20].

Repeated co-activation of neurons can lead to the formation of CAs. Similarly, repeated co-activation of multiple CAs result in the formation of multiple and sequential associations, and even new CAs. When an external stimulus activates a CA, it may excite neurons shared with a different CA that is not directly stimulated, activating it. This forms the rudimentary, neural level explanation of associative memory.

In prior work, associative memory has been explored with orthogonal and overlapping CAs. Orthogonal CAs were used to encode spatial cognitive maps, many-to-many, and context sensitive associations [18]. With orthogonal CAs, a neuron belongs to at most one CA, but with overlapping CAs, a neuron may belong to several CAs. Learnt overlapping CAs can form hierarchical categories from instances of individuals [21]. The simulations mentioned in this paragraph use a similar neural and topological architecture to the one described below.

3. The fLIF CA architecture

A computational model based on fLIF neurons, using a Hebbian learning mechanism can self-organise to form CAs. Similar to many existing models, the basic architecture of such a mechanism, explained below, is a simplification of the mammalian neural architecture.

3.1. The fLIF neuron

The fLIF neuron model is an extension of the Leaky Integrate and Fire (LIF) model [22,23]. fLIF neurons share many attributes with their biological counterparts. Like the biological neuron, the fLIF neuron integrates coincident pre-synaptic potentials until a critical threshold is reached. On exceeding the threshold, the neuron produces an action potential, or *fires*. This potential further propagates via the neuron's axonal terminal to incident post-synaptic neurons, while the firing neuron loses its activation. The neuron leaks potential if the firing threshold is not attained for prolonged periods. This leaking behaviour of fLIF neurons is similar to that of the biological neuron. This can be represented as follows:

The activation A of a neuron i at time t is

$$A_i(t) = \frac{A_i(t-1)}{\delta} + \sum_{j \in V_i} w_{ji} \quad (1)$$

The current total activation A is the sum of incoming activation and remnant activation from the previous time step $t-1$ divided by decay factor $\delta > 1$. The incoming activation is the sum of total neurons that fired at $t-1$ of all neurons $j \in V_i, V_i$ being all pre-synaptic neurons of i that fired at $t-1$, weighted by the connection from neuron j to i.

When the accumulated activation A exceeds the threshold θ, the neuron fires, losing its potential and thereby resetting A to zero. Firing is a binary event, and activation of w_{ji} is sent to all neurons i to which the firing neuron j has a connection.

Fatiguing causes the threshold to be dynamic, $\theta_{t+1} = \theta_t + F_t$. F_t is positive (F_+) if the neuron fires at t and negative (F_-) otherwise. An increase in the threshold causes the total amount of activation required for neuron firing to increase. Hence, successive firing reduces the ability of the neuron to fire. Similarly, the threshold decreases with each step the neuron does not fire, but is never less than the original threshold.

3.2. Learning

Learning in the fLIF network is dictated by a correlatory Hebbian learning rule [24], whereby synaptic connection weights are modified based on the following equation:

$$\Delta^+ w_{ij} = (1 - w_{ij}) * \lambda \quad (2)$$

$$\Delta^- w_{ij} = w_{ij} * -\lambda \quad (3)$$

w_{ij} is the synaptic weight from neuron i to j and λ is the learning rate. During each step, weights change based on the state of pre-synaptic and post-synaptic neurons. If both neurons fire, the weights increase as per the Hebbian rule (Eq. (2)). If only the

Author's personal copy

pre-synaptic neuron fires, weights decrease as per the anti-Hebbian rule (Eq. (3)). Thus w_{ij} changes and approximates k, the likelihood of j firing if i fires. Learnt CAs may have excitatory connections with each other based on the connection rule (Eq. (4)), but their inter-synaptic weights are generally low because neurons in different CAs seldom co-fire.

Each neuron is either excitatory or inhibitory, where all synapses leaving the neuron are either excitatory or inhibitory, conforming to Dale's principle [25]. A similar learning rule applies to inhibitory neurons that makes the synaptic weight approximate $k-1$ where k is the likelihood that the post-synaptic neuron fires when the pre-synaptic neuron fires.

Once learnt, external activation acting on a CA causes it to inhibit connected but inactive CAs in the same subnet via learnt inhibitory connections. Similarly, simultaneous co-activation of connected CAs increases the connection strength between them, creating associations.

3.3. Network architecture

The Pong agent described in this paper has a modular architecture, where the agent's network is divided into smaller subnets. Learning occurs within and across these subnets that are a part of the larger fLIF neural network. The neurons do not have self-connections and they synapse uni-directionally.

Intra-subnet excitatory neurons have distance-biased synaptic connections similar to biological neurons [26]. The distance-biased connectivity makes connections to neighbouring neurons. In addition, each excitatory neuron has a long distance axon which synapses to one random area of the subnet. Thus, excitatory neurons have connections to nearby neurons and neurons in another area of the subnet.

The subnets have a toroidal topology to eliminate border effects. The local connectivity rule for excitatory neurons besides the long distance connection is given by Eq. (4). There exists a connection between neurons i and j of a network only if $C_{ij} = 1$:

$$C_{ij} = 1 \quad \text{if } r < (1/(d * v))$$
$$C_{ij} = 0 \quad \text{otherwise} \tag{4}$$

where r is a random number between 0 and 1, d is the city block distance between the pre- and post-synaptic neuron, and v is a constant, the connection probability. This indicates that connections in a network are influenced by distance between neurons and the connection probability and stops if $d > 5$ as this has been observed to work well. The parameter v is 0.8 for *VisualInput* and *ControlNet* and 2.0 for the rest. Thus excitatory neurons in *VisualInput* have about 40 connections to other neurons in the net, and *PaddleNet* neurons have about 20 connections.

Parameters vary across different subnets as shown in Table 1. These have been determined largely by a process of manual parameter exploration and have been observed to be fairly robust. Inter-subnet connections are described in Section 4.1.

4. The Pong simulation

The Pong agent visually perceives a simplified version of the classic arcade game of Pong running autonomously or being played by a human. It learns by continuously encoding game moves as massively overlapping CAs. The game consists of a vertically movable paddle and a Pong ball. The goal of the game is to move the paddle along the Y-axis so as to hit the incoming ball and prevent it from crossing the paddle's Y-axis field. The agent learns to play the game by encoding spatial information, that is, its Y-axis position relative to the ball's Y-axis position. The agent is always learning and its behaviour is modified dynamically by its own actions. This in turn causes categorical behaviour to emerge, which further assists the agent in game play. CAs are categorisers, each being activated by a range of similar inputs, and the highly overlapping CAs in this simulation support generalisation.

Fig. 1 shows the visual field as seen by the agent. The ball is in the mid field and the paddle on the left, seen as regions of neuronal activation.

4.1. Network setup

Fig. 2 shows the model's network setup. All subnets have unidirectional connections, except for *PaddleNet* and *BallNet*, as indicated by the solid arrows. *VisualInput* represents the visual stimuli coming from the game; *PaddleInter* is the intermediate subnet that shows the paddle's position, but is inhibited (turned off) during testing; *BallNet* receives activation from the ball in the

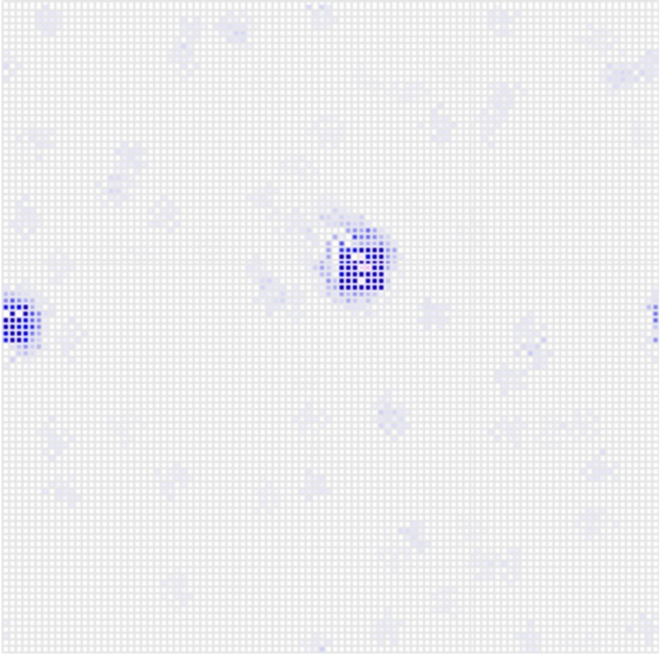


Fig. 1. Visual field of the agent showing the paddle and Pong ball.

Table 1
Network parameters.

	λ Learning	θ Threshold	$F_+ = F_-$ Fatigue	δ Decay	Excitatory neurons (%)
Visual	0.0	6.0	0.6	1.2	80
Control	0.0	1.5	0.2	1.2	0
Inter	0.0	4.5	0.6	1.2	90
Paddle	0.1	3.5	0.9	1.5	70
Ball	0.1	4.5	0.9	1.3	70

Author's personal copy

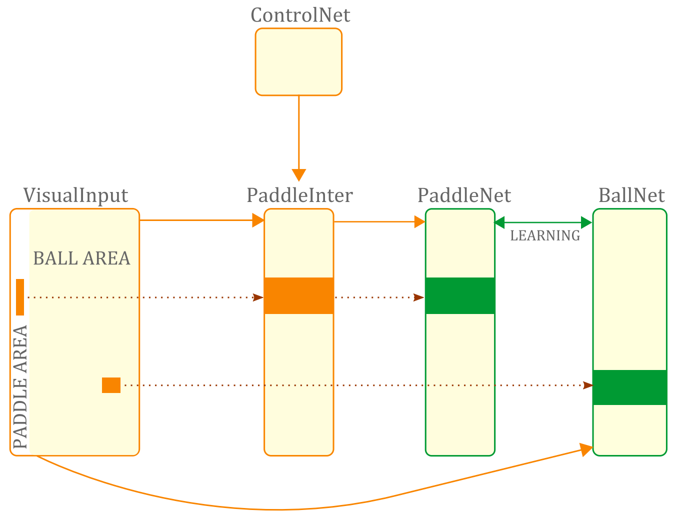


Fig. 2. Pong agent's network setup.

Table 2
Pong subnet sizes.

Subnet	No. of neurons	Rows	Columns
VisualInput	10000	100	100
ControlNet	200	20	10
PaddleInter	1000	100	10
PaddleNet	1000	100	10
BallNet	1000	100	10

visual input. Neurons in both *PaddleInter* and *BallNet* receive activation via excitatory connections from *VisualInput*. *PaddleNet* integrates activity from *PaddleInter* and associations from *BallNet*; and *ControlNet* inhibits *PaddleInter* so as to shut off visual perception of the paddle during testing.

Visual input is a direct mapping of the neuronal activation in *VisualInput* with neurons directly mapped to the paddle and ball subnets. In Fig. 2, the shaded areas in *PaddleInter*, *PaddleNet* and *BallNet* show activation received from the visual input. Each excitatory neuron in the paddle area of *VisualInput* (the five left most neurons) connects randomly to 10 neurons in the corresponding row of *PaddleInter* with a weight of 6.0. The next 90 neurons in the row of *VisualInput* connect randomly to 10 neurons in the corresponding row of *BallNet* with a weight of 6.0. Excitatory *PaddleInter* neurons connect to the 10 *PaddleNet* neurons with weight 6.0. So, the activation in the paddle and ball subnets is *Y*-axis mappings of the paddle and ball in the visual field, respectively. That is, they are representatives of the vertical positions of the paddle and ball in the game at all times. The dotted arrows in Fig. 2 from the visual field to the subnets indicate this. Every time the paddle or ball moves, their positions are updated in the paddle and ball subnets in real time by the neurons firing in the *VisualInput* subnet. Due to fatigue, not all neurons associated with the paddle and ball fire. Each *ControlNet* neuron has 100 synapses to randomly chosen *PaddleInter* neurons with synaptic weight -8.0, enabling the *ControlNet* to inhibit it. The negative weight causes the neuron to inhibit post-synaptic neurons. In the test mode, *PaddleNet* moves the paddle.

Learning takes place only within and between *PaddleNet* and *BallNet*. Initial inter-subnet connections are random low weight excitatory and inhibitory connections.

The network parameters of the Pong agent are presented in Table 1 and the subnet sizes in Table 2. For further details, the source code for the simulation is available at http://kailashnadh.name/research/files/CApong.zip.

4.2. Human trained game play

The simulation begins with a human playing the game. The paddle is controlled via keyboard input. The goal of the game is to prevent the ball from coming in contact with the wall on the paddle's side, by moving the paddle along the *Y*-axis and hitting the ball. As the paddle and ball move, their vertical positions are mapped in *PaddleInter*, *PaddleNet* and *BallNet*, respectively, by neural firing.

Since there is two way learning between *PaddleNet* and *BallNet*, the recurrent neuronal activation from *VisualInput* causes them to remain co-active. Repeated activation of a certain region causes the synaptic strength between active neurons to increase, forming CAs.[1] Co-activation causes the synaptic strength between active CAs in *PaddleNet* and *BallNet* to increase, thus associating them. For example, when there is no human input, if neurons at the bottom of the *PaddleNet* are firing, the paddle will move to the bottom of the screen in the game. Since the CAs in the paddle and ball subnets represent the vertical position of the paddle and ball, respectively, the position of the paddle and ball in relation to each other is learnt. This, in essence, encodes the player's action of moving the paddle relative to the ball and hitting the ball by aligning the paddle's position to it.

Formation of basic CAs (within a given subnet) is driven by external stimuli. As particular neurons are activated via environmental stimuli, the CAs that correspond to those neurons are strengthened, with neurons that are less central being weakened. For example, in *BallNet*, excitatory neurons have strong connections to neurons in the same row, slightly weaker connections to adjacent rows and strengths declining as the distance extends. Inhibitory neurons have a similar pattern with neurons many rows away receiving net inhibition from a given row. Activation from a CA (internal stimulation) may also spread over the overlapping region to form bigger CAs. This behaviour results in the formation of overlapping CAs that change dynamically.

Based on the player's moves, associations of CAs in paddle and ball subnets vary. For instance, consistent behaviour reinforces certain associations and causes the rest to weaken via learnt inhibition. Every time the positions of the paddle and ball change, neuronal activity in the paddle and ball subnets shift accordingly. If the movement is minimal, it causes the activity in the subnets to shift only slightly thus overlapping with the antecedent CA. That is, new vertical position mappings form by the amalgamation of positions close to each other.

In the test mode, the *ControlNet* inhibits *PaddleInter* which relays stimuli from visual input to *PaddleNet*. This in effect stops the agent's view of the paddle by disabling visual stimuli from the paddle so that paddle movement occurs solely from the internal state. Stimuli from the ball area in its visual field continue to activate CAs formed in *BallNet* based on the ball's motion. The CAs that become active in *BallNet* in turn activate the ball–paddle position CAs learnt in *PaddleNet*.

Neurons in *PaddleNet* act as motor neurons and move the paddle; movements in the game are a direct mapping of the vertical position of the spikes that occur in *PaddleNet*. Since the CAs in the paddle and ball subnets encode position variables, the paddle moves in accordance with the ball. If the action of hitting the ball was reinforced by the human player, the paddle

[1] CAs are persistent as is evident when the system is well trained and input is turned off. Activity persists in *BallNet* and *PaddleNet* for roughly 70 cycles, where one cycle is 10 ms in simulated time.

Author's personal copy

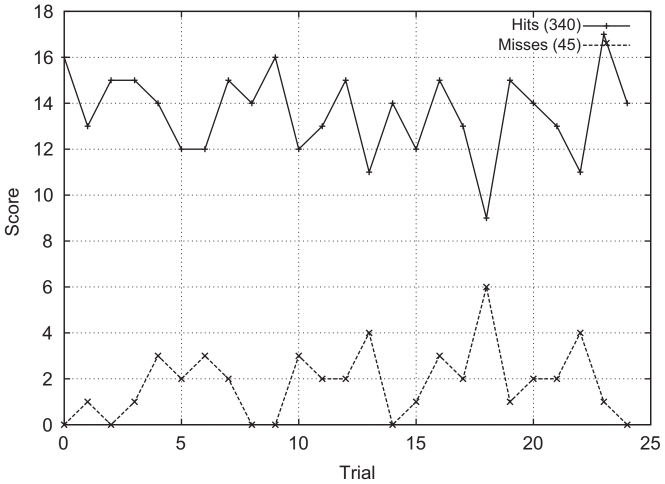


Fig. 3. Pong agent's game score in human trained game play mode.

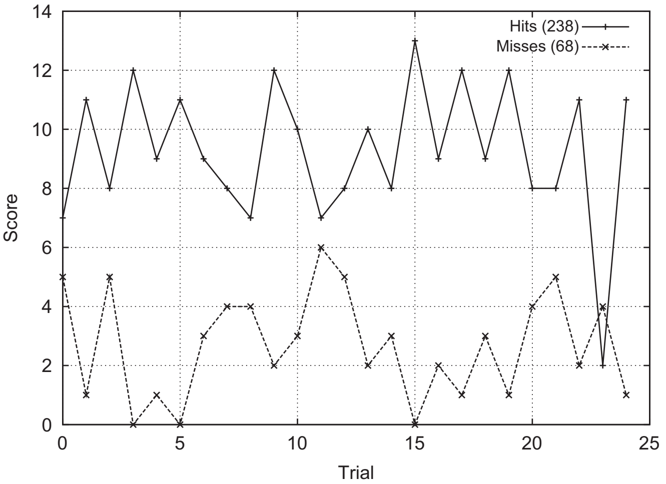


Fig. 4. Pong agent's game score in autonomous game play mode.

will correctly move up to the ball and hit it. Since the system is always learning, all learnt movements the agent makes in turn modify its own behaviour.

Due to the absence of a reward system, it is possible for erratic training to reinforce sub-optimal game moves. So, if the human intentionally misses the ball during training (inverting the game), the agent will also initially miss the ball.

A simple scoring system keeps track of the agent's performance recording hits and misses. As there is a considerable amount of randomness in the initial network topology, and in human trained game play, the game was run 25 times, each with a different network configuration. A human player played for 4 min, and the agent played on its own for another 4 min. In all trials, the agent hits the ball more times than it missed. Fig. 3 shows the score distribution, where the score is the total number of successful hits.

4.3. Autonomous game play

In the autonomous game play mode, the agent acquires game moves on its own. As there is no human intervention and no pre-existing memory, initially, the paddle stays idle at the starting position despite the ball moving around in the field. Movements are still driven by the *PaddleNet* as described in Section 4.2.

ControlNet inhibits the paddle vision so that only the ball's position is perceived. This is necessary to prevent the agent from viewing the paddle, so that learning only occurs from the visual stimulus of the ball crossing the paddle's field and not the *Y*-axis position of the stationary paddle itself.

When the ball crosses the paddle's *Y*-axis field, its neuronal activation is mapped to *PaddleNet*. This is due to the paddle staying stationary initially. At this point, *ControlNet* is shut down automatically, releasing the inhibition on paddle vision, enabling the agent to visually perceive the position of the ball which is now in the paddle's field. As explained in Section 4.2, the paddle moves in relation to the neuronal spikes in *PaddleNet*. In this case, the spikes are caused by the ball entering the paddle's visual field. The paddle quickly moves along the *Y*-axis to the position of the ball, where activation in *PaddleNet* and *BallNet* remain co-active for 50 cycles after which *ControlNet* inhibits the paddle vision. The co-occurrence triggers learning of the position of the paddle in relation to the ball, as explained in Section 4.2. The ball then bounces off and this process repeats as it comes back to the paddle's side again. After a few repetitions, enough position CAs are encoded in *PaddleNet* and *BallNet* that the agent is able to move the paddle in correspondence to the ball's *Y*-axis position and successfully prevent it from hitting the wall.

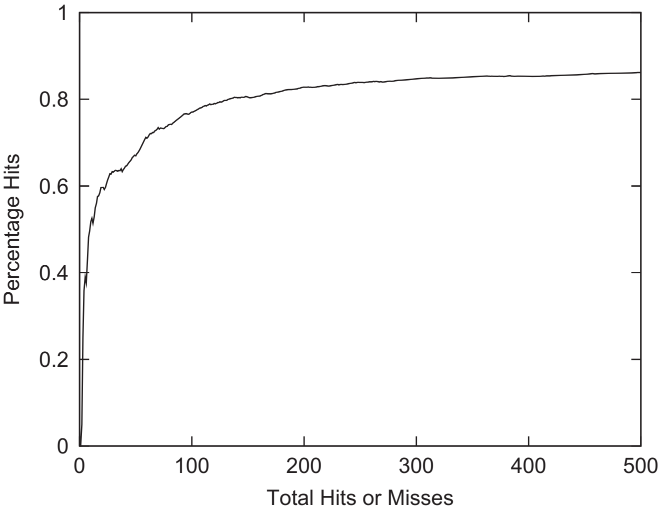


Fig. 5. Percentage of hits over time.

The game was run as 25 trials each with a different network configuration for a duration of 4 min, where the agent autonomously acquired game moves while engaged in game play. In most trials, the agent missed the first few hits until enough learning was done and then started scoring significantly better. Fig. 4 shows the score distribution across 25 trials.

Fig. 5 shows the average behaviour of 10 different network configurations playing autonomously over the first 500 hits or misses. The behaviour improves over time and is still improving at the end. The average performance after 500 interactions was 86%, and an average using a moving window of the last 100 interactions was 90%. The worst net had 72% with 500 interactions though it was 88% on the moving window. On the moving window case, the best net still only hit 95% at the end.

5. Conclusion and discussion

The agent, based solely on simulated fLIF neurons, learns to play the Pong game fairly well. It has been shown that the system learns associations between visual inputs and actions while the agent is observing or playing the game. The associations are dynamic,

Author's personal copy

changing over the course of a game. Nonetheless, they are stable, forming over all tested subnets, and persisting throughout each run.

The underlying architecture is based on reasonably biologically faithful fLIF neurons. Like biological neurons, the simulated neurons always learn. Moreover, the distance-biased topology, sparse connectivity and use of excitatory and inhibitory neurons are biologically plausible features. The authors would not argue that there is a direct relation between the neurons in the simulation and particular neurons in the brain; for instance, the *VisualInput* net is an extreme simplification of the human visual system, as is the motor output driven by *PaddleNet*. Nonetheless, it seems plausible that there is a relation between the simulated neurons involved in the associative learning (*PaddleNet* and *BallNet*) and biological neurons involved in learning to hit the Pong ball; of course this mapping would be complex.

The gross topology of the network (Fig. 2) provides connections between the possible actions (paddle positions) and the possible inputs (paddle and ball positions). For independent learning, the trace left by the ball when missed provides a driver for moving the paddle. The connections are very weak and diffuse initially. At a coarse level, associations between the actions and inputs are learnt from co-presentation. At a finer level, the synapses that support these associations are learnt via neural co-firing driven by co-presentation.

Similarly, the actual concepts that underlie input and action are formed dynamically from input. That is, initially, there are no CAs, but they are learnt via presentation of game instances. CAs both form and become associated in this simulation.

Once overlapping CAs are formed, these provide generalisation that enables the system to perform in novel situations. Other simulations have involved orthogonal CAs [18] with each CA having distinct neurons. It seems unlikely that biological CAs are orthogonal [27]. One of the benefits of overlapping CAs is that by their nature, they provide generalisation. For example, in a given run of the Pong game the ball may never come to a particular position in *VisualInput*, so a particular set of neurons in *BallNet* will never be active. However, some of those neurons may be activated when a nearby position is reached, and this will allow them to increase their mutual synaptic strength. When that particular position is first presented to the system, the CA has already formed. Existing associations already provide support for the novel situation.

Since the CAs only encode *Y*-axis positions, the agent does not necessarily predict the position of the ball during game play. It recalls the paddle's position relative to that of the ball's, learnt in previous runs that enables the agent to align the paddle with the *Y*-axis position of the ball. The overlaps provide a way to extend the learnt positions novelly. If the game were changed so that the paddle needed to be in a different place, e.g. the opposite position on the *Y*-axis, for each ball position, that would be learned as effectively. A more complex setup would be likely to allow predictive behaviour to emerge. The horizontal movements of the ball and the variations in angle would be taken into account, and meta learning, based on a reward mechanism where positive and negative reinforcements [28] for correct and incorrect moves, respectively, would be integrated.

These CAs have been described as massively parallel because each neuron participates in a large number of CAs. In these simulations, CAs persist with about 50 neurons firing persistently, so each neuron is in roughly 50 local CAs, and a larger number across subnets. Fig. 6 illustrates the successive activation of CAs formed in *PaddleNet* at an interval of five cycles from cycle 400 to 425 in a particular run. Each of the six rectangular strips is a snapshot of the state of *PaddleNet* at a particular step, with the seventh one being a composite of all states. Activation levels of neurons are visible as shades, where the darkest regions are CAs with firing neurons and white regions are inactive neurons. The visualisation shows the paddle moving up, where activation is seen to spread across multiple CAs. The active CAs are not discreet and overlap with preceding and succeeding ones as illustrated by the composite image, where no physical gaps can be seen between them.

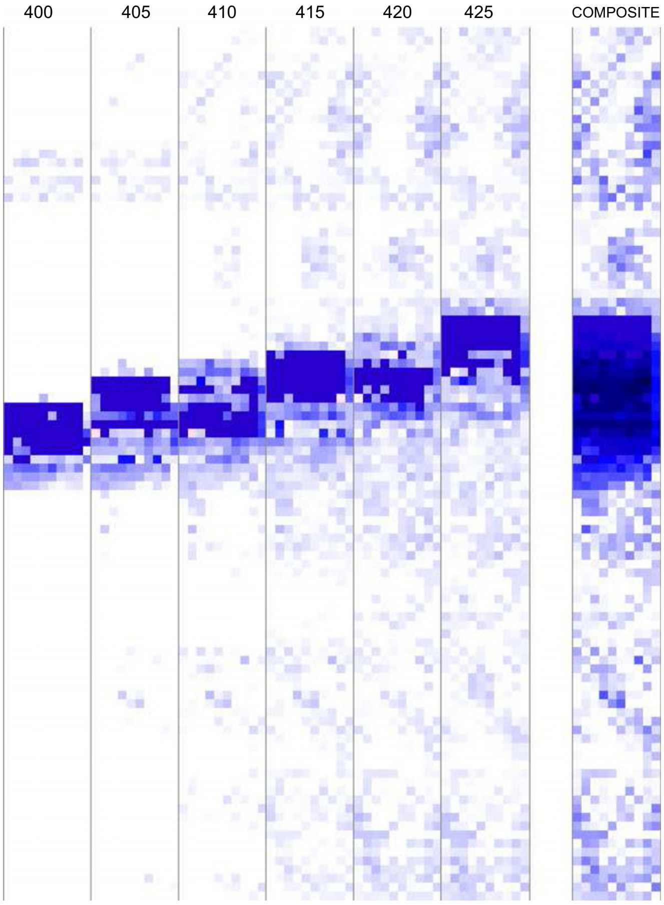


Fig. 6. Visualisation of active CAs in PaddleNet at specific intervals.

One of the problems with simulated CA formation is that when one CA becomes active, it may lead to the activation of associated CAs, which in turn lead to a cascade of activation with the whole network becoming active, i.e. simulated epilepsy [29]. In the simulations described in this paper, the spread of activation is limited and counterbalanced by inhibitory connections preventing this.

If it is reasonable to speculate on particular brain areas associated with particular subnets of Fig. 2, then the *PaddleNet* subnet can be considered to be similar to place cells that are known to occur in the Hippocampus of rats [30]. Similar results with humans indicate that neurons in the hippocampus reflect the position of the person [31]. The behaviour of the simulated neurons in the overlapping CAs that enable the agent to move about in the game is also similar to the behaviour of hippocampal place cells that are involved in spatial encoding [6]. That is, if the system described above were a model of a person acting as a Pong paddle, the neurons in *PaddleNet* subnet could correspond to some Hippocampal neurons.

The Hebbian learning mechanism used to create associations between ball and paddle positions is a general mechanism. There are, however, cases where the learning mechanism alone fails. For instance, when an action is associated with a condition and the environment indicates it is a bad association, the co-firing still

Author's personal copy

supports the association. This particular problem has been solved by a more complex system to support appropriate associations and firing [28]. Nonetheless, the associative nature of the Hebbian learning mechanism is the default. So when a sensory stimuli and action co-occur, they are associated by default.

The simulations presented in this paper exhibit the learning of a large number of associations that enable the agent to play the Pong game. These are learnt in a simulation that has a relatively high degree of biological plausibility. The continuous overlapping nature of the CAs that are learnt allows the system to generalise and thus generate reasonable performance, showing one benefit of overlapping CAs. This work is another step toward understanding the underlying neural mechanisms of associative memory.

Acknowledgements

The authors would like to thank Dan Diaper and Peter Passmore for their constructive feedback during the preparation of this paper. This work was partially supported by EPSRC Grant GR/R13975/01.

References

[1] D.O. Hebb, The Organization of Behavior, Wiley, New York, 1949.

[2] F. Schwenker, F.T. Sommer, G. Palm, Iterative retrieval of sparsely coded associative memory patterns, Neural Networks 9 (1995) 445–455.

[3] N. Levy, D. Horn, Associative memory in a multi-modular network, Neural Computation 11 (1999) 1717–1737.

[4] L.G. Valiant, Memorization and association on a realistic neural model, Neural Computation 17 (2005) 527–5557.

[5] K. Harris, Neural signatures of cell assembly organization, Nature Reviews Neuroscience 6 (2005) 399–407.

[6] A.P. Maurer, S.L. Cowen, S.N. Burke, C.A. Barnes, B.L. McNaughton, Organization of hippocampal cell assemblies based on theta phase precession, Hippocampus 16 (9) (2003) 785–794.

[7] F. Pulvermuller, Words in the brain's language, Behavioral and Brain Sciences 22 (1999) 253–336.

[8] A. Pasupathy, C. Connor, Population coding of shape in area v4, Nature Neuroscience 5 (12) (2002) 1332–1338.

[9] S. Funahashi, Neuronal mechanisms of executive control by the prefrontal cortex, Neuroscience Research 39 (2001) 147–165.

[10] J. Foster, G. Alexander, Neuron activity related to short-term memory, Science 173 (1971) 652–654.

[11] T. Wennekers, G. Palm, Cell assemblies, associative memory and temporal structure in brain signals, in: R. Miller (Ed.), Time and the Brain. Conceptual Advances in Brain Research, vol. 2, Harwood Academic Publishers2000, pp. 251–274.

[12] C. Huyck, Cell assemblies as an intermediate level of cognition, in: Emergent Neural Computational Architectures Based on Neuroscience, 2001, pp. 383–397.

[13] T. Wennekers, A cell assembly model for complex behaviour, Neurocomputing 70 (10–12) (2007) 1988–1992.

[14] A. Knoblauch, R. Kupper, M.-O. Gewaltig, U. Körner, E. Körner, A cell assembly based model for the cortical microcircuitry, Annual Review of Neuroscience 27 (2004) 649–677.

[15] W. Gerstner, W.K. Kistler, Mathematical formulations of Hebbian learning, Biological Cybernetics 87 (5–6) (2002) 404–415.

[16] G. Schoenbaum, Cell assemblies and the ghost in the machine, in: H. Eichenbaum, J. Davis (Eds.), Neuronal Ensembles: Strategies for Recording and Decoding, Wiley1998, pp. 81–116.

[17] M. Nicolelis, L. Baccala, R. Lin, J. Chapin, Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system, Science 268 (1995) 1353–1358.

[18] C.R. Huyck, K. Nadh, Multi-associative memory in fLIF cell assemblies, in: R.C.E.A. Howes, D. Peebles (Eds.), 9th International Conference on Cognitive Modeling—ICCM2009, Manchester, UK, 2009, pp. 81–87.

[19] J.J. Hopfield, Neurons with graded response have collective computation properties like those of two-state neurons, Proceedings of the National Academy of Sciences, vol. 81, 1984, pp. 3088–3092.

[20] A. Lansner, Associative memory models: from the cell-assembly theory to biophysically detailed cortex simulations, Trends in Neuroscience 32 (3) (2009) 178–186.

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

[22] W. Maas, C.M. Bishop, Pulsed Neural Networks, MIT Press, 2001.

[23] W. Gerstner, Integrate-and-fire neurons and networks, in: The Handbook of Brain Theory and Neural Networks, The MIT Press2002, pp. 577–581.

[24] C. Huyck, Overlapping cell assemblies from correlators, Neurocomputing 56 (2004) 435–439.

[25] J.C. Eccles, Chemical transmission and Dale's principle, Progress in Brain Research 68 (1986) 3–13.

[26] P. Churchland, T. Sejnowski, The Computational Brain, MIT Press, USA, 1992.

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

[28] R. Belavkin, C. Huyck, A model of probability matching in a two-choice task based on stochastic control of learning in neural cell-assemblies, in: Proceedings of the 9th International Conference on Cognitive Modelling, 2009.

[29] C. Connolly, R. Reilly, A proposed model of repetition blindness, in: A. Cangelosi, G. Bugmann, R. Borisyuk (Eds.), Modelling Language, Cognition and Action: Proceedings of the 9th Neural Computation and Psychology Workshop, World Scientific2005, pp. 279–288.

[30] G. Dragoi, G. Buzaski, Temporal encoding of place sequences in hippocampal cell assemblies, Neuron 50 (2006) 145–157.

[31] D. Hassabis, C. Chu, G. Rees, N. Weiskopf, P. Moyneus, E. Maguire, Decoding neuronal ensembles in the human hippocampus, Current Biology 19 (2009) 546–554.

Kailash Nadh received his BSc degree (cum laude) in Computer Science from the Middlesex University, London, in 2008. He is currently pursuing a Ph.D. at the Middlesex University and his research interests include neural networks, neural computer vision and cognitive modelling.

Christian R. Huyck received his BSc/AB from the University of Michigan in 1986 in Computer Science and History, his MSc in Computer Science in 1988, and his PhD in Natural Language Parsing in 1994. He is Middlesex University's Reader in Artificial Intelligence and heads a research group concentrating on the neural basis of intelligence. He works with simulations of neural models implement systems and agents using cognitive models that function in environments, making use of both neural processing and learning. This work is informed by neuroscience, psychology, linguistics, theoretical computation, and good old fashioned AI.