Procedia Computer Science

Volume 88, 2016, Pages 198–204

7th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2016

PlaNeural : Spiking Neural Networks that Plan

Ian Mitchell[1], Christian Huyck[1], and Carl Evans[1]

Middlesex University, London, UK
i.mitchell,c.huyck, c.evans@mdx.ac.uk

Abstract
PlaNeural is a spike-based neural network that has the ability to plan. The network is a spreading activation network implemented with Cell Assemblies; this combination has built a dynamic network of nodes that is able to interact with an environment and respond appropriately. PlaNeural uses Cell Assemblies to make decisions and plan - there is no pre-determined code managing the decision process that leads to planning. PlaNeural is the planning component of a virtual robot in a virtual environment. This paper describes PlaNeural 's behaviour in two virtual environments, programmed independently of it; actions are completed in a closed-loop. PlaNeural was programmed in PyNN, executed with Nest and on a neuromorphic platform, SpiNNaker. PlaNeural has been tested on two environments and results show a successful performance; in both cases PlaNeural takes appropriate actions to fulfil user selected goals based on environmental changes.

Keywords: PyNN, Cell Assembly, Spiking Neural Networks, Planning

1 Introduction

Bostrom [1] states that intelligence requires three conditions: learning, logic and planning. Planning is the ability to take pre-conditions and facts from an environment and satisfactorily deliver a goal. Many agents are capable of interacting with an environment, so what makes PlaNeural different? PlaNeural is based on the third generation of neural network models, Spiking Neural Networks (SNN) [12] based on integrate and fire neurons [2] and Cell Assemblies (CA) [7]. This means all decisions are completed via spiking neurons.

1.1 Cell Assemblies and Neural Networks.

Cell Assemblies (CAs) [7] are groups of neurons that are interconnected and when neuron firing within the group exceeds a threshold, the CA fires. CAs can be active but not firing, i.e. neurons firing within the group are insufficient to exceed the thresholds, when in this state the CA is said to be primed. There is growing support and evidence that they are used in brains to represent concepts [11]. CAs are a distributed and decentralised model and when many CAs are networked they can produce dynamical systems that solve complex problems from classification [10] to parsing language [8]. CAs can be implemented in SNNs and thus classified as third generation neural networks, CAs can be used to solve

 Selection and peer-review under responsibility of the Scientific Programme Committee of BICA 2016

doi:10.1016/j.procs.2016.07.425

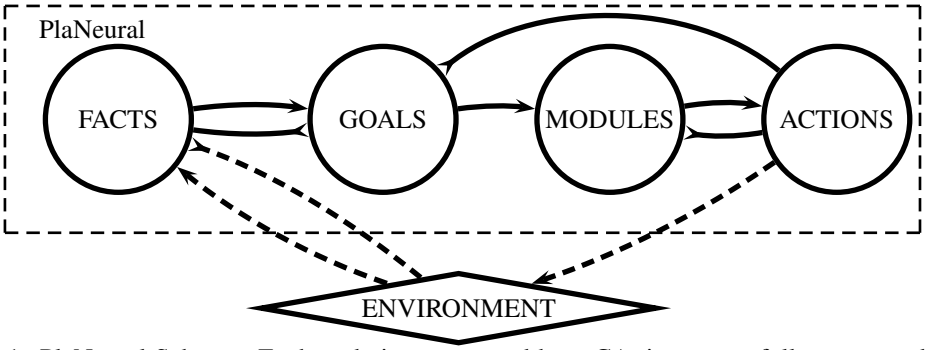

Figure 1: PlaNeural Schema: Each node is represented by a CA, in tests, a fully connected (no self-connections) five neuron cell assembly using Integrate and Fire neurons. Facts are connected to Goals. Goals in turn excite Modules and finally Modules excite the appropriate Action. To ensure that there is no prolonged activity, Actions inhibit the appropriate Module and Goal. Actions will change the Environment, which will give feedback to the Facts and complete the closed loop.

classic AI problems. CAs can be used to implement cognitive and other types of models, e.g. applying them to FSM [5]. In §2 a Maes network-inspired model is discussed.

1.2 Maes networks

Maes [13] uses a network of competence modules to develop plans, illustrated in figure 1. Each competence module has successor, predecessor and conflicter connections. Activation is spread throughout the network with constraints drawn from successor, predecessor and conflicter connections. Maes networks are distributed and dynamical systems of inhibitor and excitatory connections and hence drawing inspiration from them. Briefly, Maes networks are: i) not hierarchical; ii) not centralised; and iii) have no explicit representation of the environment, but rather communicate with the environment and react to any changes by choosing suitable actions. Transforming Maes-like networks to SNNs seemed a logical step and is explained in §2.

2 PlaNeural

PlaNeural is a SNN that plans and satisfies goals. Goals are entered by the user and appropriate actions fire. The consequence of these actions can change the environment and generate new facts. The environment is separate from PlaNeural , which depends on stimulus from user-input for goals and environmental facts. When combined these two inputs change the activation in the network and converge on an overall action. On completion of the action the associated goal and module are inhibited.

2.1 Implementation

Programming was completed using PyNN [3], a simulator independent spiking neuron based programming language. To ensure the results were robust, two simulators were chosen: Nest (2.6) and SpiNNaker [6], the latter being neuromorphic hardware [1].

[1] on loan from APT at the University of Manchester

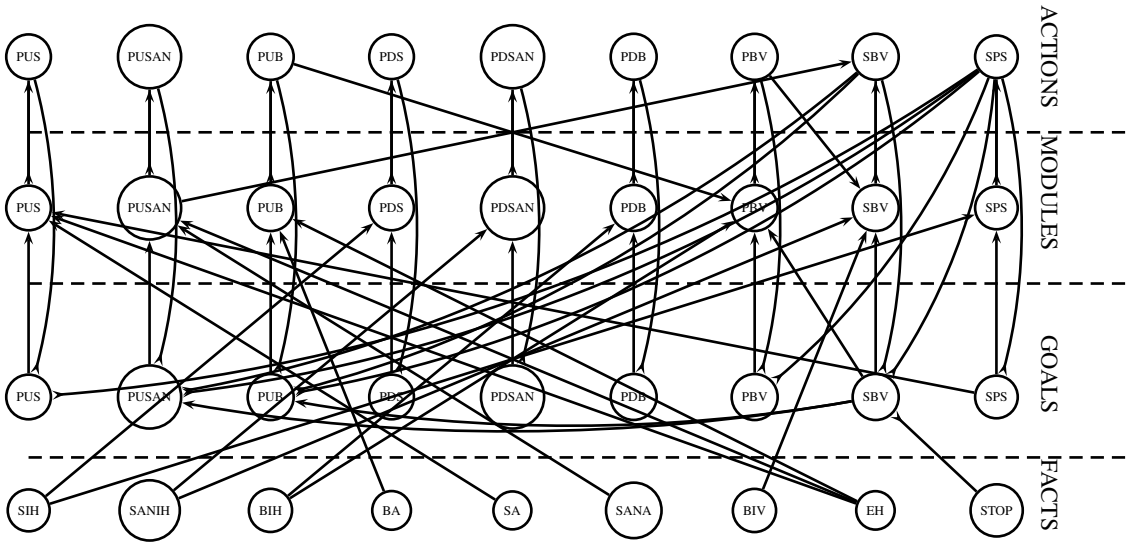

Figure 2: PlaNeural Structure for Robot. Key is referred to Table 1.

Key	Description	Key	Description	Key	Description
BA	Board Available	PDSAN	Put Down Sander	BIV	Board in Vice
SA	Spray Available	PDS	Put Down Spray	SANIH	Sander in Hand
SANA	Sander Available	PBV	Put Board in Vice	BIH	Board in Hand
PUS	Pick up Spray	SBV	Sand Board in Vice	SIH	Spray in Hand
PUSAN	Pick up Sander	EH	Empty Hand	SPS	Spray Paint Self
PUB	Pick up Board	STOP	Stop	PDB	Put down Board

Table 1: Commands for Maes Robot

2.2 Experiment 1: The Sander Task

Briefly, the virtual agent is a robot with two hands; some objects that can be grasped by the robot are: sander, board and spray-paint. The goals can be divided into two groups: single-objective; and multi-objective. The single objective goals are: pick up object, put down object, where object is a board, sander or spray-paint. The multi-objective goals are: sand board in vice; put board in vice; sand board in hand; and spray-paint self. The last objective requires the robot to shut down and wait for paint to dry, so it cannot achieve any other goals once this has been achieved. The test of this is to sand the board and spray paint self and thus use all single objectives in a correct sequence. This is the original Maes task and more details can be found in [13].

Figure 2 details the network as a graph and four distinct groups of CAs, each represented by a circle and a name. The connections are either inhibitory or excitatory. Both sets of Synaptic weights are static. There is a further type of connection not represented in Fig. 2 and that is internal connections within the CA, these neurons are fully connected (disallowing self connections) and are also static. Otherwise all connections between populations are fully connected. The standard set up for each neuron in the network is shown in Table 2. The code can be found at

Description	Value	Description	Value
Synaptic Time Constant (excitatory)	2.0ms	Resting Potential	-65.0mV
Synaptic Time Constant (inhibitory)	5.0ms	Refractory time	5.0ms
Threshold	-51.0mV	Reset potential	-70.0mV
offset	0.0	distance	0.1cm

Table 2: Neuron Parameters for PlaNeural

www.cwa.mdx.ac.uk/NEAL/code/bica16plan/code.tar.gz

Figure 3 is a rastergram of the spiking neurons after given goals are injected to the network. The two goals are described in the list below.

0ms : Facts show that the agent has Empty Hand (EH) (see Table 1 for abbreviations) and the environment has Spray-paint, Board and Sander available (SA,BA,SANA).
200ms : The goal Sand-Board-in-Vice (SBV) fires. This cascades approx. 10ms later to PUB and PUSAN sub-goals.
≈**275ms** : Actions indicate that both the Board and Sander have been picked-up, PUB and PUSAN fire. Approx. 5-10ms later this is reflected in the facts by SANA, BA and EH are no longer persisting. This is also reflected in facts SANIH and BIH persisting.
≈**300ms** : Action PBV fires and as a result ≈ 10ms later BIV fact persists. This is also reflected by BIH no longer persisting and EH persisting.
≈**340ms** : Action SBV fires, which matches the original goal, and in turn inhibits PUB, PUSAN and SBV in Goals
≈ **400-700ms** : is a period of rest, no new goal are given to the network. Facts EH, BIV, SA and SANIH persist during this period.
≈ **700ms** : Goal SPS fires.
$approx$ 760ms : Action PUS fires and as a consequence Facts, EH and SA no longer persist, and SIH persists
≈ **800ms** : Action SPS (clipped in figure) fires and as a consequence inhibits the Goals SPS.

2.3 Experiment 2: Simple Cognitive Mapping

The agent navigates through a series of rooms with objects in them. The agent is to approach the object (further work is to identify and build a spatial map) and then enter the next room. The agent is executed on a neuromorphic platform, SpiNNaker. More details about the environment can be found in [9].

Table 3 explains the commands - there are essentially 4 actions: forward; right; left; and backward (not used). There is an issue of going through doors; whilst in the corridor an object could be recognised in another room, so when in the corridor an inhibit object (IO) is activated that prevents objects being recognised until the agent gets through the door. The network was developed using the same code as in the previous experiment and is data-driven, i.e. only the data sets fed into the network changed. All graphs are created based on the network data fed into the program.

Table 4 plots spiking neurons over a period of 2000ms after given goals are injected to the network and are described in the list below.

0ms : small test, TR goal is activated followed by approx. 10ms later the action TR. Note that the fact LEFT and PYR is on, but the goal TR (turn right) takes priority.
≈**100-300ms** : Goal EXP is activated along with facts LEFT and PYR. Subsequently, GP and TP sub-goals are activated. The agent turns towards the object and centres the object in the vision field.
≈**300-600ms** : CENTRE fact is activated and the response of the agent is to move forward towards the object.

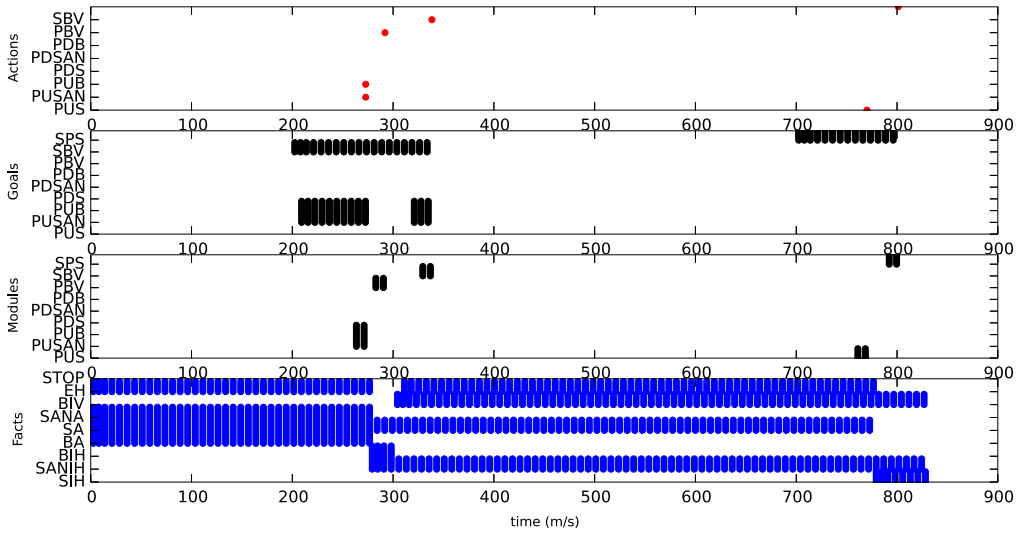

Figure 3: Rastergram for goals: i) Sand board in vice goal set at 200ms; and ii) Spray paint self goal set at 700ms. Neuron number is represented on the vertical access with each CA represented by five neurons.

Key	Description	Key	Description	Key	Description
TR	Turn Right	TL	Turn Left	FW	Forward
BW	Backward	BUMP	Bump	FO	Object Forward
LO	Object Left	RO	Object Right	STAL	Object 1
PYR	Object 2	DOOR	Door	RIGHT	Object on Right
LEFT	Object on Left	CENTRE	Object in Centre	ENDOOR	End of Door
STOP	Stop agent	EXP	Explore	UO	Inhibit Object
IO	Identify Object	TP	Turn to Object 2	TS	Turn to object 1
GP	Go towards Object 2	GS	Go towards Object 1	GD	Go towards Door

Table 3: Commands for the mapping experiment.

≈**600-800ms** : IO is deactivated and GD is activated as a sub-goal. DOOR and RIGHT are activated facts and the door is centred by turning right.

≈**800-1000ms** : DOOR and CENTRE are activated and the agent moves towards the DOOR.

≈**1000-1500ms** :UO goal is active and ENDOOR is the activated fact. The agent keeps moving forward until it is clear of the door and looks for the next object in the new room.

≈**1500-1600ms** :UO is deactivate and IO is activated. The object, STAL, is identified and the agent moves forward.

≈**1600-2100ms** :The DOOR and RIGHT, followed by CENTRE are activated facts. The actions TR follow by a series of FW through the next door. Eventually the agent is halted by STOP fact activated causing inhibition throughout the network and leaving the agent to rest.

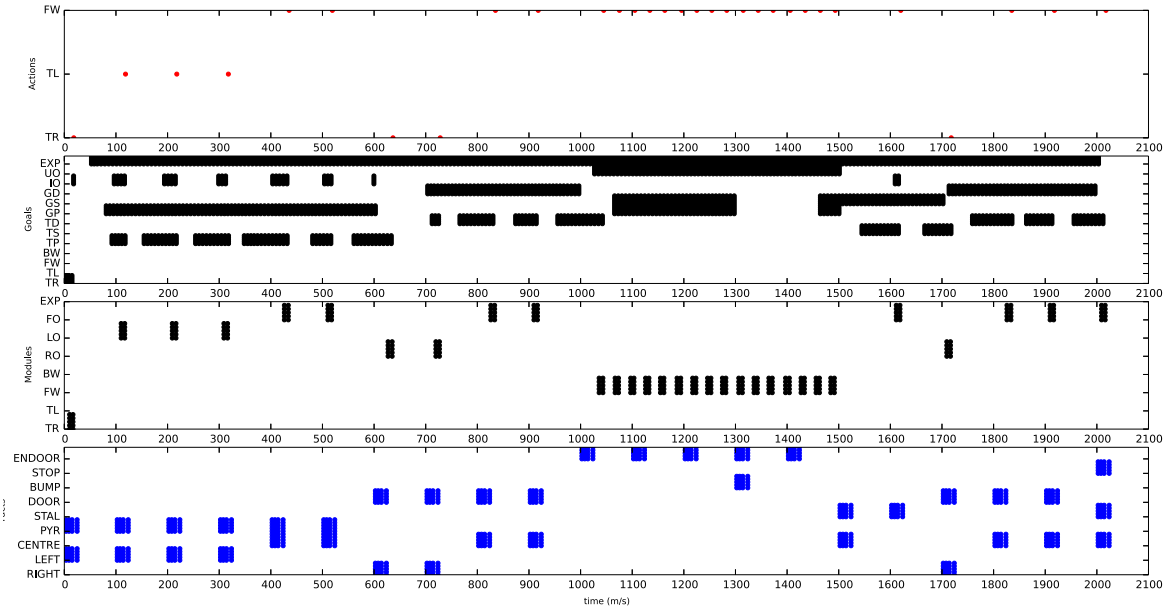

Figure 4: Results for goals in the mapping experiment. i) turn right at 0ms; and ii) explore at 100ms

3 Conclusion

Creating plans using CAs is the main contribution. Building a Maes-inspired network to cope with planning has been a success based on the results of the two environments. Artificial Intelligence requires planning. PlaNeural is a SNN for planning where all decisions are made and based on neurons firing. The actions from PlaNeural are fed into the environment and changes are made appropriately. These are represented by changes in facts that feed into PlaNeural and hence a closed-loop.

The two main challenges that have been met:

1. Planning with SNN. The topology describes a network that demonstrates the ability to plan in two environments under two different implementations, Nest and Spinnaker, using the concept of Maes-inspired Networks combined with Cell Assemblies.

2. Topology: Systematically building a framework for future agents as seen in Figures 3 and 4. This systematic approach will improve areas of planning in the development of agents.

In addition to planning Bostrom [1] states two other conditions for artificial intelligence: learning; and logic. Unlike other spiking networks [4], PlaNeural does not learn, it is a systematic approach to the development of plans using spiking neural networks. Future work is to add plasticity and the ability to generate plans, in essence it is the "spikification" of a Maes net. The main contribution of this paper is the development of plans using spiking neural networks, something which has been overlooked in AI literature since much is devoted to learning and logic to solve problems. Planning, combined with learning and logic, aims to build better agents. In summary this paper has focussed on the development of a planning agent, PlaNeural . The development was completed in PyNN using Nest and on a neuromorphic chip, SpiNNaker. The authors have provided a systematic way to implement a method to plan using SNNs with the combination of Maes-inspired networks and cell assemblies.

PlaNeural ran successfully in two different environments and could form the basis of any planning in an agent relying on SNN.

Acknowledgements: This work was supported by the Human Brain Project Grant 604102, Neuromorphic Embodied Agents that Learn. The authors also wish to thank the reviewers for their helpful comments.

References

[1] Nick Bostrom. *Superintelligence: Paths, dangers, strategies*. OUP Oxford, 2014.

[2] R. Brette and W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.*, 94:3637–3642, 2005.

[3] A. Davison, D. Brüderle, J. Eppler, E. Muller, D. Pecevski, L. Perrinet, and P. Yqer. PyNN: a common interface for neuronal network simulators. *Frontiers in neuroinformatics*, 2, 2008.

[4] C. Eliasmith, T.C. Stewart, X. Choo, T. Bekolay, Y. Tang T. DeWolf, and D. Rasmussen. A large-scale model of the functioning brain. *Science*, 338:1202–1205, 2012.

[5] Y. Fan and C. Huyck. Implementation of finite state automata using flif neurons. In *IEEE Systems, Man and Cybernetics Society*, pages 74–78, 2008.

[6] S. Furber, D. Lester, L. Plana, J. Garside, E. Painkras, S. Temple, and A. Brown. Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, 62(12):2454–2467, 2013.

[7] D. Hebb. *The Organization of Behavior*. John Wiley and Sons, 1949.

[8] C. Huyck. A psycholinguistic model of natural language parsing implemented in simulated neurons. *Cognitive Neurodynamics*, 3(4):316–330, 2009.

[9] C. Huyck, R. Belavkin, F. Jamshed, K. Nadh, P. Passmore, E. Byrne, and D.Diaper. CABot3: A simulated neural games agent. In *7th Intl Workshop on Neural-Symbolic Learning and Reasoning, NeSYS'11*, pages 500–544, 2011.

[10] C. Huyck and I. Mitchell. Post and pre-compensatory Hebbian learning for categorisation. *Computational Neurodynamics*, 8:4:299–311, 2014.

[11] C. Huyck and P. Passmore. A review of cell assemblies. *Biological Cybernetics*, 107:3:263–288, 2013.

[12] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[13] P. Maes. How to do the right thing. *Connection Science*, 1:3:291–323, 1989.