Neural Networks 44 (2013) 6–21

Contents lists available at SciVerse ScienceDirect

Neural Networks

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

Adaptive training of cortical feature maps for a robot sensorimotor controller

Samantha V. Adams [a,*], Thomas Wennekers [a], Sue Denham [b], Phil F. Culverhouse [a]

[a] Centre for Robotics and Neural Systems, School of Computing and Mathematics, University of Plymouth, PL4 8AA Plymouth, United Kingdom

[b] School of Psychology, University of Plymouth, PL4 8AA Plymouth, United Kingdom

ARTICLE INFO

Article history:
Received 1 April 2012
Received in revised form 2 March 2013
Accepted 3 March 2013

Keywords:
Spiking neuron SOM
Activity dependent learning
Robot controller

ABSTRACT

This work investigates self-organising cortical feature maps (SOFMs) based upon the Kohonen Self-Organising Map (SOM) but implemented with spiking neural networks. In future work, the feature maps are intended as the basis for a sensorimotor controller for an autonomous humanoid robot. Traditional SOM methods require some modifications to be useful for autonomous robotic applications. Ideally the map training process should be self-regulating and not require predefined training files or the usual SOM parameter reduction schedules. It would also be desirable if the organised map had some flexibility to accommodate new information whilst preserving previous learnt patterns. Here methods are described which have been used to develop a cortical motor map training system which goes some way towards addressing these issues. The work is presented under the general term 'Adaptive Plasticity' and the main contribution is the development of a 'plasticity resource' (PR) which is modelled as a global parameter which expresses the rate of map development and is related directly to learning on the afferent (input) connections. The PR is used to control map training in place of a traditional learning rate parameter. In conjunction with the PR, random generation of inputs from a set of exemplar patterns is used rather than predefined datasets and enables maps to be trained without deciding in advance how much data is required. An added benefit of the PR is that, unlike a traditional learning rate, it can increase as well as decrease in response to the demands of the input and so allows the map to accommodate new information when the inputs are changed during training.

1. Introduction

The current work is part of a larger research project which aims to transfer novel principles from the field of Computational Neuroscience to a practical robotics application. A small, autonomous humanoid robot will learn basic visuomotor coordination skills using an approach based upon the self-organising topological map as a representation of the mammalian cortex. This sort of approach to modelling the cortex is not new and much work has been done previously on the development of preferences in the visual cortex (Goodhill, 1993; Miikkulainen, Bednar, Choe, & Sirosh, 1998, 2005; Willshaw & von der Malsburg, 1976). The method has also been applied to practical sensory-motor tasks such as visuomotor control (Alamdari, 2005; Kikuchi, Ogino, & Asada, 2004; Metta, Sandini, & Konczak, 1999; Morse & Ziemke, 2009; Ogino, Kikuchi, Ooga, Aono, & Asada, 2005; Paine & Tani, 2004; Ritter, Martinez, & Schulten, 1989; Rodemann, Joublin, & Korner, 2004; Toussaint, 2006). A variety of approaches have been used for visuomotor control in these previous works such as learning robot arm kinematics and dynamics directly (Ritter et al., 1989), learning the coordination between visual input and 'motor primitives' (Kikuchi et al., 2004; Metta et al., 1999; Ogino et al., 2005), incorporating traditional search and reinforcement learning techniques (Alamdari, 2005; Toussaint, 2006) and even using evolutionary techniques to learn mappings (Paine & Tani, 2004). Whilst all these approaches have been successful, the majority of them would not be suitable for implementation in an autonomous robot operating in real time, because of the amount of computation required for some of the techniques and, more importantly, they have all required the use of a host PC to do the computations even when real robot hardware is used.

Achieving a human-like level of skill in a robot is a challenging task as the sensory pre-processing and higher level cognitive processing that is required needs significant computing power which is in conflict with the limited energy resources available on an autonomous robot. However, natural systems somehow manage to achieve speed, fault tolerance and flexibility despite having very low power requirements. Since the current capabilities of robots do not match even the simplest animal, it seems logical to explore in more depth bio-inspired approaches to robotics: in particular, where artificial neural systems are implemented using techniques inspired by a greater understanding of how real neurons, and brains work. Computational Neuroscience

* Corresponding author. Tel.: +44 0 1752 586294; fax: +44 0 1752 232540.
E-mail address: samantha.adams@plymouth.ac.uk (S.V. Adams).

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2013.03.004

has made considerable progress in recent years on spiking neuron based models of sensory and cognitive processes in the mammalian neo-cortex. Spiking Neural Networks (SNNs) are the 'third generation' of Neural Networks (Maass, 1997); the first generation being networks consisting of simple McCulloch–Pitts neurons (McCulloch & Pitts, 1943) with binary outputs and the second generation consisting of neurons with continuously-valued activation functions. Spiking neurons mimic how real neurons compute: with discrete pulses rather than a continuously varying output. The spiking neuron is, of course, still an abstraction from an actual neuron, but a much more biologically plausible one especially as models can incorporate spike-timing based learning which is believed to be an important mechanism in natural systems. Advances in software and hardware over the last ten years or so have made SNNs an increasingly feasible option for robotics applications. On the software side several general purpose spiking neuron simulators are freely available which means that researchers do not have to code a modelling framework from scratch, and they also benefit from a community of users using the same tool. Desktop computing hardware is now available that can perform parallel processing (e.g. GPU) at an affordable price. But this can only take us so far. The emerging field of Neuromorphic Engineering is making it possible to simulate large spiking neural networks in hardware in real time with modest power requirements. 'Neural chips' are massively parallel arrays of processors that can simulate thousands of spiking neurons simultaneously in a fast, energy efficient way (Jin et al., 2010; Serrano-Gotarredona et al., 2009; Silver, Boahen, Grillner, Kopell, & Olsen, 2007). To realistically be able to implement the complexity of neural networks required for human-like behaviour on-board robots in the future will require implementation in such neuromorphic hardware. Our approach of using Spiking Neural Networks has been directly motivated by the possibility of using this emerging technology to implement sensory-motor controllers directly on-board robots operating in real time and with lower power consumption than traditional computing technologies.

The underpinning concept of the current work is the cortical self-organising feature map (SOFM) which is an analogue of how biological brains manage to represent complex multidimensional information from their environment as a 2D map in the cortex. The SOFM methodology is inspired by the Kohonen Self-organising Map (SOM) (Kohonen, 1995). The original Kohonen SOM is an unsupervised learning technique most commonly used for machine learning, for example, data clustering applications. It is usually a two layer network: the 'output' layer which forms the map and the 'input' layer which passes in the data to be represented. The two layers are usually fully interconnected. The input layer has as many neurons/nodes as there are dimensions of data. During the training process, the Kohonen SOM self-organises to represent the range of input data available and in the final map the data is topologically arranged (similar inputs are mapped to similar locations in the map). The weights on the connections between the two layers 'store' the patterns and thus the number of connections to a neuron/node determines the maximum dimensionality of the map. The process of map training can be summarised as follows:

1. Present an input vector of training data.
2. Select the winning node in the output layer with the highest activation.
3. Determine a spatial neighbourhood around the winning node.
4. Adjust weights in the neighbourhood by using the learning equation $\Delta w_{ij} = k\left(x_i - w_{ij}\right) y_j$.
5. Decrease the neighbourhood size N, and the learning rate k.
6. Repeat steps 1–5 for the desired number of training cycles.

The methodology used in the current work is based upon the traditional SOM as described above but with several key differences. Essentially spiking neurons are used instead of traditional artificial neurons with continuous activation. The learning rule incorporates both spatial and temporal factors to learn the mapping of input patterns. In addition a self-regulating process is used to adapt the learning rate in an online fashion so that during training, patterns can be selected and presented on the fly rather than using predefined datasets.

The structure of the paper is as follows. Section 2 describes the details of the methodology used to create a prototype cortical motor map, including details of the spiking neural network setup, learning rules and training process. Section 3 describes the development of a simple adaptive plasticity method for regulating map training and how this has been used to replace the learning rate parameter traditionally used in SOMs. Section 4 describes the results from several experiments which demonstrate the benefits of this learning method and Section 5 includes a discussion and comments on areas for future work.

2. Methods

2.1. Overview

As outlined in the introduction, the current work has adapted the traditional SOM methodology in several key ways. Firstly, spiking neurons are used which means that instead of using the highest activation to determine the winner, it is the temporal response of the neurons that becomes important. Using spiking neurons introduces the concept of spike timing as a means to carry additional information. Traditional SOMs use only a spatial neighbourhood around the winner, but in the current work spatial and temporal neighbourhoods are used to develop the map organisation. We have also made some amendments to the traditional SOM to make it better suited for use in autonomous robotic applications where the goal is to enable the robot to learn from the information available in its environment in a completely unsupervised way.

SOM network development is generally thought of in terms of two distinct phases: initial topological ordering followed by weight convergence and in computational models the phases are usually managed explicitly by the manipulation of neighbourhood size and learning rate parameters. Both of these parameters are normally systematically reduced in a non-linear fashion during the course of training according to predefined schedules. In the case of an adaptive sensorimotor controller for a robot it is not ideal to have to predefine such schedules to control map development. Instead the development should be self-regulating as it is in natural systems. The issue of defining learning rate and neighbourhood parameter reduction in relation to traditional SOMs has been noted by several previous researchers. For example, Berglund and Sitte (2006) and more recently in Berglund (2010) the Parameter-Less SOM or PLSOM is described. These works developed a method of controlling the learning in a SOM by using the ratio of the last error between the input vector and weight vector of the winning node to the largest previous error as a scaling factor (Berglund & Sitte, 2006). In later improvements, the ratio of the last error to the diameter of the input space is used (Berglund, 2010). Shah-Hosseini and Safabakhsh (2000, 2001) developed the TASOM or Time-Adaptive SOM. Here, each neuron has its own learning rate and neighbourhood parameters and these are changed according to the distance measure between the current input vector and the synaptic weight vector of the neuron. More recently, Shah-Hosseini (2011) has developed a variant called the Binary Tree TASOM, which incorporates the removal and addition of neurons during training to allow adaptation in an environment