Learning Sensory-Motor Cortical Mappings Without Training

Mike Spratling Gillian Hayes
Department of Artificial Intelligence
University of Edinburgh
mikes@dai.ed.ac.uk gmh@dai.ed.ac.uk

Abstract.
This paper shows how the relationship between two arrays of artificial neurons, representing different cortical regions, can be learned. The algorithm enables each neural network to self-organise into a topological map of the domain it represents at the same time as the relationship between these maps is found. Unlike previous methods learning is achieved without a separate training phase; the algorithm which learns the mapping is also that which performs the mapping.

1. Introduction

A prerequisite for the performance of a skill is knowing the relationship between actions and their effects. For a simple skill this may be achieved by finding the mapping from the sensor domain to the motor domain. When these domains are represented by neural networks it is a case of finding appropriate synaptic weights to connect the motor network to the outputs from the sensor network (figure 1(a)). Methods of finding the mapping in such an architecture have previously been presented as models of the cerebral cortex [1, 9].

In the cerebral cortex there is evidence that both sensor and motor regions are topologically organised and use population coded representations [6, 3, 2, 9]: Representations are distributed over the activity of a whole population of neurons each of which respond over a range of inputs and have overlapping receptive fields (RFs) [9, 7] (figure 1(b)). Such coding is efficient for generating coordinate transformations since it allows interpolation between nodes, and is robust to node failure and noise in individual neuron activations. Learning to form appropriate connections from the sensor to motor region is equivalent to defining the receptive fields of the nodes in the motor region. In a similar way the nodes in the sensor region must learn appropriate receptive fields to represent sensory input. Various evidence has been presented to suggest that, although the cortex forms areas of functional specialisation, regions organise themselves using similar principles [11, 8, 5]. The model presented here also uses the same algorithm to learn appropriate receptive fields for both the sensor and motor region simultaneously (the same algorithm is used throughout space).

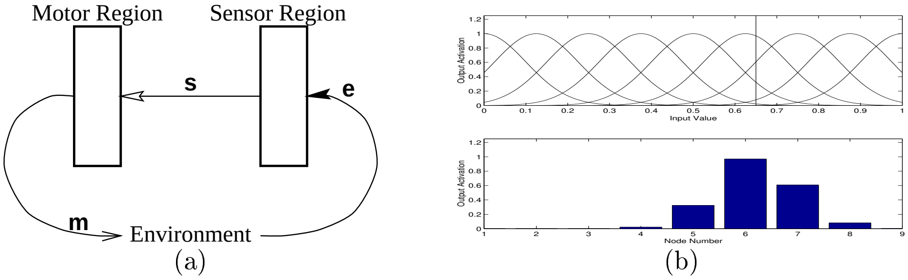

Figure 1: **(a) The architecture used to learn simple sensory-motor mappings** consists of two regions: A motor region generating the motor output and a sensor region receiving inputs in response to the motor actions. These regions are joined by connections which will learn the required mapping.
(b) An example of population coding. Top: Each curve represents the change in activation as a function of input value for each neuron in the one-dimensional array. The extent of these curves defines the receptive field for the neuron. Bottom: The activation values of the population of neurons when representing an input value of 0.65.

To learn the transformation between sensor and motor space requires training data covering the range of possible actions. Thus, most algorithms (*e.g.* [1, 9, 7]) go through a distinct training phase during which uniformly distributed random training data is generated and the inputs to the sensor region and outputs of the motor region are set to corresponding values from this training data (as if the data was generated by random motor actions). The algorithm implemented by the motor region is thus different during the training phase from that implemented when the resulting mapping is used. It is unlikely that neurons in the brain switch between behaviours or that motor actions are under 'external' control during development. Such a distinct training phase is also a practical problem since any change in the sensory-motor alignment requires a new training phase to be performed. The method presented in this paper does not require any separate training phase; the mapping is learned at the same time as the motor region generates outputs covering the whole range of actions (the same algorithm is used throughout time).

2. Implementation

Two arrays of nodes, a sensor and a motor region, are connected such that the output from the sensor region forms the input to the motor region (figure 1(a)). The output activity of the motor array is treated as a population coded value and decoded as such[1] The input to the sensor array is a population coded

[1]. The simplest means of decoding the population code [9], by taking the weighted sum of activations, is used: $value = \frac{\sum y_j X_j^{pref}}{\sum y_j}$
where y_j is the output activation of node j, and, X_j^{pref} is the preferred direction of node j.

representation of the motor action. To ensure that the motor outputs are associated with their sensory consequences the synapses which form the inputs to the motor layer are modified before the next motor output is calculated. There are thus two variations of pseudo-Hebbian learning in use: afferent connections are modified after the activity of the nodes is found, while efferents are updated before the new node activations are calculated. This allows the regions to be run sequentially while ensuring that the correct associations are learned.

The nodes in both regions form appropriate receptive fields using a novel, fully-competitive, self-organising, learning algorithm [10]. Nodes compete, via lateral inhibition, to represent inputs, and at each iteration a winning node, which is most strongly activated by the current input, is selected. The lateral inhibition increases as a function of distance from the winning node. This generates a topologically ordered map, in which neighbouring nodes have overlapping receptive fields. Local inhibition is weak so that nodes in the neighbourhood of the winner remain active. The output of the network is thus the activity of a population of nodes, centred around the winner. The selection of the winning node is affected by noise added to the activations and habituation of nodes which win the competition most frequently. Both habituation and noisy selection are essential for the topological self-organisation of each network [10]. These two mechanisms are also responsible for allowing the mapping between regions to be learned. All synaptic weights start at zero strength, so that initially when the connections to the motor region are weak, the output will be almost entirely random. As the connections become stronger there is a tendency for the current sensor input to re-activate the previous motor output, and hence produce the same sensor and motor effects continuously. However, habituation prevents this from occurring for more than a few iterations, allowing the architecture to continue learning.

3. Results

Two one-dimensional arrays of nodes where used to represent the sensor and motor regions. Figure 2 shows the synaptic weights learned after 10000 iterations with different numbers of nodes in each array. All results have been generated using identical learning algorithms (including the same values for the parameters) in both regions. It is clear that the algorithm is fairly robust to changes in the networks, and that very similar patterns of receptive fields are generated in all cases (1st column of figure 2). The networks form well ordered topological maps in which there is monotonic progression in the preferred input of each node across the array (2nd column of figure 2). The training data is not random, but is generated by the output of the motor array. Initially, the weight of lateral inhibition is zero and each node has a similar output activity; the decoded output is thus the same at each iteration (the mean of the preferred directions), and hence the mapping error is initially very low (3rd column of figure 2). As lateral inhibition increases the range of output values generated also increases, and hence so does the error, but this increased range of output

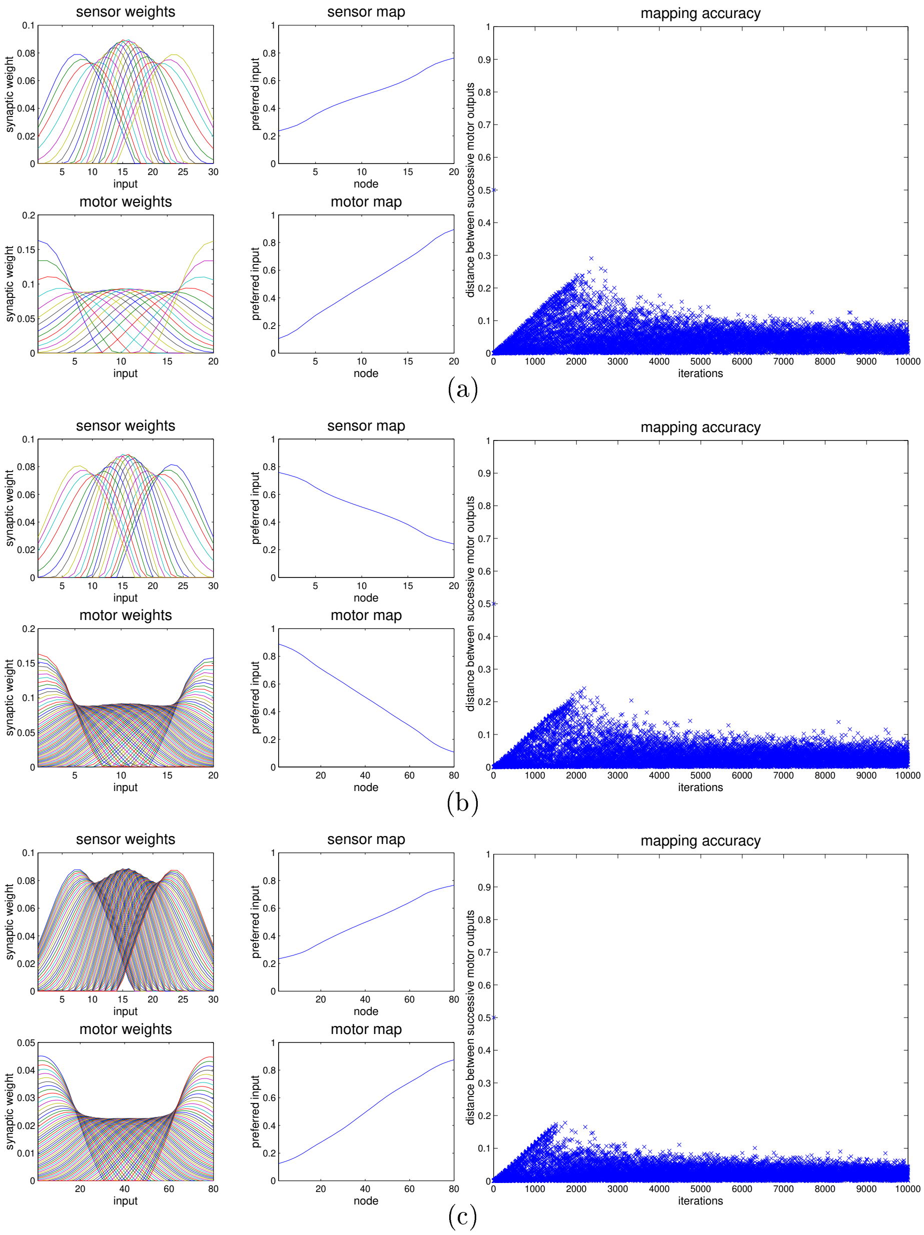

Figure 2: **Results after 10000 iterations.** (a) Both arrays contain 20 nodes. (b) The sensor array contains 20 nodes and the motor array 80 nodes. (c) Both arrays contain 80 nodes. 1st column shows the synaptic weights for all nodes in each region. 2nd column shows how the preferred input (that input which most strongly activates a node) varies along the array. 3rd column shows the variation over time of the error between the target position specified by the sensor input and the subsequent target position generated by the motor output.

values also provides training data and as the correct connections to implement the mapping are learned so the error reduces. It can be seen that the residual mapping error is reduced as the number of nodes increases.

4. Conclusions

Three requirements for a model of the development of cortical mappings are suggested in section 1.:

1. Uniformity of Algorithm over cortex:
   Since physiological evidence suggests that all cortical regions are organised by the same developmental process, models of different cortical regions should be organised by the same learning algorithm. Various information processing and organisational requirements provide constraints as to the nature of this algorithm.
2. Uniformity of Encoding over cortex:
   Since the output from one cortical region will form (part of) the input to other regions there is a need for inputs and outputs to have the same coding format. Both the requirement for topological organisation and physiological data support the use of population coding.
3. Uniformity of Algorithm over time:
   To learn a skill requires learning the relationship between motor actions and sensory effects, which requires training examples covering the range of possible actions. The same algorithm that generates the correct output for a given input must also be that which learns this mapping.

All of these requirements are met by the model described in this paper.

The architecture proposed here is very similar to that used by Salinas and Abbott [9] in that it learns the mapping between two population coded arrays. It has been shown [9] that given an array of motor neurons whose activity has sensory consequences, and an array of sensor neurons whose receptive fields are defined, it is possible to learn the mapping between these domains, provided that training data contains corresponding sensor and motor values and that the learning rules are such that the magnitude of the resulting synaptic connections are dependent on the difference between the preferred directions of the pre- and post-synaptic neurons. This algorithm meets these criteria and so, in common with their algorithm, it should generalise to networks encoding more than one variable. However, the architecture presented here improves on their work since in [9] the receptive fields of the nodes in the sensor region are predefined (fails requirement 1) and training is by the injection of random data into the motor region (fails requirement 3).

References

[1] P. M. Churchland. 'Some Reductive Strategies In Cognitive Neurobiology'. In M. A. Boden, editor, *The Philosophy of Artificial Intelligence*, chapter 14. Oxford University Press, 1990.

[2] A. P. Georgopoulos. 'Neural Networks and Motor Control'. *Neuroscientist*, 3:52–60, 1997.

[3] D. H. Hubel. *Eye, Brain, and Vision.* Scientific American Library, 1988.

[4] M. H. Johnson, editor. *Brain Development and Cognition: A Reader.* Oxford: Blackwell, 1993.

[5] M. H. Johnson. 'Constraints on Cortical Plasticity'. In *[4]*, pages 703–721. 1993.

[6] E. I. Kundsen, S. du Lac, and S. D. Esterly. 'Computational Maps In The Brain'. In J.A. Anderson, A. Pellionisz, and E. Rosenfeld, editors, *Neurocomputing 2*, chapter 22. MIT Press, 1990.

[7] B. W. Mel. *Connectionist Robot Motion Planning : A Neurally-Inspired Approach to Visually-Guided Reaching*, volume 7 of *Perspectives in Artificial Intelligence.* Academic Press, 1990.

[8] D. D. M. O'Leary. 'Do Cortical Areas Emerge From A Protocortex?'. In *[4]*, pages 323–337. 1993.

[9] E. Salinas and L. F. Abbott. 'Transfer of Coded Information from Sensory to Motor Networks'. *Journal of Neuroscience*, 15:6461–6474, 1995.

[10] M. W. Spratling and G. M. Hayes. 'A Self-Organising Neural Network For Modelling Cortical Development'. In *6th European Symposium on Artificial Neural Networks*, submitted.

[11] M. Sur. 'Visual Plasticity in the Auditory Pathway: Visual Inputs Induced into Auditory Thalamus and Cortex Illustrate Principles of Adaptive Organisation in Sensory Systems'. In M.A. Arbib and S. Amari, editors, *Dynamic Interactions in Neural Networks : Models and Data*, pages 35–51. Springer-Verlag, 1989.