Implementing Hebbian Learning in a Rank-Based Neural Network

Manuel Samuelides[1,2], Simon Thorpe[3] and Emmanuel Veneau[1,3]

1 Ecole Nationale Supérieure de l'Aéronautique et de l'Espace,
BP 4032, 31055 ,Toulouse Cedex, France.

2 ONERA-CERT, Computer Science Department,
BP 4025, 31055 Toulouse Cedex France.

3 Centre de Recherche Cerveau et Cognition, CNRS,
Faculté de Médecine, 133 route de Narbonne - 31062, Toulouse, France.

Abstract. Recent works have shown that biologically motivated networks of spiking neurons can potentially process information very quickly by encoding information in the latency at which different neurons fire, rather than by using frequency of firing as the code.In this paper, the relevant information is the rank vector of latency order of competing neurons. We propose here a Hebbian reinforcement learning scheme to adjust the weights of a terminal layer of decision neurons in order to process this information. Then this learning rule is shown to be efficient in a simple pattern recognition task. We discuss in conclusion further extensions of that learning strategy for artificial vision.

1 Introduction

In the vast majority of artificial neural-network architectures, the activation state of the individual units is either a binary variable (as in the original McCulloch-Pitts formulation), or a continuous function, typically taking values between 0 and 1. Continuous activation functions are generally believed to correspond to the firing rates of biological neurones. However, as [2] pointed out, there are situations in which the speed of neural computation is too fast to be able to make use of firing rate codes, simply because individual neurones will only have enough time to generate one spike. One way of overcoming such temporal constraints is to take advantage of the fact that even the most simple integrate-and-fire neurones can be effectively thought of as analog-to-delay convertors in that the time needed for such a neuron to reach threshold and generate a spike will depend on the strength of its input. Such ideas have received increasing interest in the last few years [1, 3, 6]. Recently it has been proposed in [4] that, instead of using the relative latency values directly, one could make use of the order in which the neurons fire.
In this paper, after providing a mathematical description of this model, we propose a Hebb-like reinforcement learning mechanism with constant learning rate and test it in a one-dimensional pattern classification problem. Then

we conclude by discussing the extension of this learning algorithm to multi-layered networks.

2 Model of a rank-based neural network

We use a simple integrate and fire model to describe spike generation by the input neurons to mimic the spike emission. The evolution equation of the activation potential of a neuron is

$$v(t+1) = \begin{cases} (1-\alpha)v(t) + f[u(t)] & \text{when } v(t) < v_f \\ v_0 & \text{when } v(t) \geq v_f \end{cases} \tag{1}$$

where v_f is the neuron firing threshold, v_0 is the neuron ground potential, α is the leakage rate per time unit, and f is a transfer integrating function of the neuron input $u(t)$ at time t.

The network architecture under consideration is a feed-forward architecture with two layers. The first layer is a preprocessing layer which is activated by a particular intensity profile, considered to be constant during processing. In the model we present here, the neurons of the first layer are in a one-to-one correspondance with the pixels of the presented pattern. So the transfer integrating function is an increasing sigmoidal function of the pixel intensity u. Since the architecture we study is dedicated to a recognition task, the neuron of the second layer are decision neurons. They receive input signals from all the neurons of the first layer. These signals are ponderated by synaptic weights and summed to form the input of the transfer integration function of the decision neuron. We shall propose in section 3 a Hebbian learning rule to adjust these weights

We define the *latency* λ of a neuron as the time of the first spike. From the equation (1) it is easy to compute the expression of the latency (in the low α approximation) $\lambda = \frac{v_f - v_0}{f(u)}$

However, although measuring delays between the arrival times of spikes originating from different sources is used in a great number of sensory systems (including echolocation in bats, electric fish, and auditory localisation), it tends to require very large amounts of neuronal machinery. In contrast, determining the order in which the inputs to a neurone arrive is in principle much easier. It was proposed in [4] to use a mechanism similar to the sorts of activity-dependent synaptic depression reported recently [5] to progressively desensitise the target neuron as a function of the number of inputs that have already been activated. According to this hypothesis, the response of a post-synaptic neurone to one of its inputs would depend not only on the effective weight of the synapse, but also on a modulatory factor that controls the neurons sensitivity. In our model, we implement the following mechanism: the synaptic efficiency w_i^j of the synapse from neuron j to neuron i is decreased by a given modulation rate β each time a spike is received by neuron i. After the image is processed, it is reset at its original value. Then the modulated