ADAPT

IST–2001-37173

Artificial Development Approach to Presence Technologies

Deliverable Item 5.2
Basic Unit Design and Implementation

Delivery Date: February 26th, 2004
Classification: Public
Responsible Person: Dr. Harri Valpola, Geoff Nitschke
Partners Contributed: ALL

Short Description: This document details the design and implementation of a feature-map, which is a single processing unit in the systems architecture that will eventually form an integral part of an embodied autonomous system. The systems architecture is to be comprised of three types of learning systems that correspond to three essential aspects of the complete embodied autonomous system. The first is a self-supervised learning system process, which will be capable of extending motor programs that are generated by a reinforcement learning process from an existing set of motor programs. The second is a value learning system that will take a set of innate values and generate new motor programs based on a reinforcement learning process. The third is an unsupervised learning process that performs invariant feature extraction. A single component of the unsupervised learning model, a feature map, is what is described within this document. Specifically, the unsupervised learning model is to eventually govern a hierarchy of feature-maps, and this hierarchy is to constitute the invariant feature extraction model. Within this hierarchical architecture each feature-map will concurrently process a continuous stream of bottom-up sensory input data, and learn to extract invariant features. The invariant features extracted by each feature-map serve as the inputs for feature-maps at a higher-level in the hierarchy. Individual feature-maps are comprised of existing neural networks learning mechanisms, such as competitive learning. Though, the original contribution of the proposed feature-map lies in the capability to learn global connectivity to its input layer.

Project funded by the European Community under the "Information Society Technologies" Programme (1998-2002)

1. Introduction

This deliverable specifies the design and implementation of the basic building block of an invariant feature-map learning model. The invariant feature map model together with supervised and unsupervised learning processes is to be one of the key aspects of the systems architecture. Also, a reinforcement learning process will govern the derivation of new motor synergies, and a self-supervised learning process the extension of existing motor-control behaviors. The invariant feature-map learning model is to be a hierarchy of feature-maps where connectivity[1] between feature-maps is learnt during an unsupervised learning process. An individual feature-map is comprised of several previously tested mechanisms common to a variety of neural networks architectures, for example, learning to extract invariances from input data (Lowe, 1999).

A feature map is a single processing unit that is designed to accept information in a specific context, for example: cortical patterns, and learns expectations from this context. Expectations are predictions of bottom-up input[2], based on top-down and lateral stimuli, where lateral connections refer to connections between neurons within a single feature map layer. At the current stage of implementation a single feature-map consists of two layers, though this structure is to be extended to include more layers in future implementations. The upper layer is termed an invariant feature layer, and the lower layer is termed an elementary feature layer. The primary role of the elementary feature layer is to cluster bottom-up inputs within particular spatial and temporal neighborhoods at the invariant feature layer, meaning that elementary features gathered at the invariant feature layer are features that occur in close temporal and spatial proximity at the elementary feature layer. In the complete hierarchy of feature-maps the goal is to learn global bottom-up input connectivity over time, where the intended purpose of such a hierarchy is that different feature maps would represent different sensory modalities such as vision, haptic sensation, proprioception, and audition.

This document aims to place a matter of perspective on the role of a basic unit within the systems architecture, which is to eventually represent the agent's brain, and then to broadly outline how a composition of such basic units will interact with the agent's motor-system. The first aim is to delineate between the two main aspects that are to comprise the complete agent, namely the systems architecture, and the motor system, which will form an essential part of the agent's embodiment. The motor-system is to be based on a reinforcement learning process as a means of attaining motor learning and providing a mechanism for curiosity. The reinforcement learning process is implemented as a separate part of the system, and represents a value system that encapsulates potentially useful values such as novelty, which will be used to derive new motor programs. That is, to encourage exploration of novel aspects of the environment as a means of maintaining the capability to learn new things. The value signal of the reinforcement learning process is a function of novelty, where the aim of the process is to regulate novelty so as to maximize learning. The motor system implements a basic set of behaviours that form

[1] In this context, the learning of connectivity refers to weight connections learnt from bottom-up inputs to the elementary feature layer, where bottom-up input axons are locally connected to a particular neighborhood of the feature layer, and global competition for connections takes place between input axons.

[2] At this prototypical stage of the feature-map implementation, input to the elementary feature layer is implemented as a pixel array where a set of pixels corresponds to a single input at the elementary feature layer. The input data maintains an implicit topology (a well defined object) that is spatially and temporally invariant.

exploration strategies to drive the system towards novel aspects of its environment, where novelty is characterized by discrepancies detected between bottom-up inputs and top-down expectations. The second aim is to describe a basic unit as forming part of a model of the cortex, and the capability of such a model to learn to extract invariant features. The cortex as a whole is to be modeled upon an unsupervised learning process. Specifically, the cortex is modeled as a hierarchy of feature-maps that performs unsupervised learning in bottom-up manner and supervised learning in a top-down manner. Within the hierarchy of feature maps modeling the cortex, it is the interaction between supervised and unsupervised learning that is to facilitate inter-modal transfer, where the goal is to find connections between different sensory modalities. It is important to note that a single feature-map models only part of the cortex, where a single feature-map incorporates unsupervised learning of bottom-up input stimuli, as well as supervised learning of top-down stimuli.

The system will utilize an inter-play between two different types of learning as a means of attaining a complete system that constantly searches for novelty and thus maintains a state of plasticity in learning. It is the interactions between *supervised* and *unsupervised* learning processes of the feature-map and the reinforcement learning process of the motor-system that is intended to produce a constant state of plasticity in learning and motivation to learn.

The top-down supervised learning process of the feature-map attempts to predict bottom-up inputs based on context specific top-down and lateral inputs. These predictions are then termed expectations. Thus, if expectations are perfect there ceases to be novel inputs as the system has learned to predict all bottom-up inputs. Once expectations have been perfectly learnt by the supervised and unsupervised learning processes, it is the task of the motor-system, based upon a reinforcement learning process, to drive the attention of the system away from these expectancies and to explore novel aspects of the environment, thus avoiding fixation upon a given set of expectancies.

The unsupervised learning process attempts to combine multiple feature-maps in novel ways for the purpose of deriving novel multi-modal and unified representations. The learning of global connectivity of bottom-up inputs to the feature-map is to be a key contribution, where this learning is facilitated by the interaction of complementary global and local competition mechanisms.

It is expected that the combination of complementary competitive mechanisms proposed in this document will solve some of the problems that have confounded traditional unsupervised competitive learning models, such as poor scaling of training time with an increased dimension and size of data representation. Specifically, it is hypothesized that the interaction between global competition in learning and local competition for activations, and the learning of global structured connectivity to the elementary feature layer will lead to a linear scaling of computational complexity. This scaling is to be proportional to the number of representational units comprising a feature-map thus allowing concurrent representation and processing of multiple sensory modalities.

The core of this document is dedicated to the formalization of a feature-map and processes governing complementary competitive mechanisms and modification of connectivity within and between feature-maps. The ultimate goal of a feature-map is outlined as being able to integrate its functional structure into a complete and autonomous learning model that can be embodied within a physical system. Though, it is important to note that the composition of novel multi-modal representations and the derivative of sensory-motor coordinated strategies and their integration with an embodied autonomous physical system (Metta *et al.* 1999; 2001) will require an unsupervised learning process interact with phylogenetic modules dedicated to value learning (Sutton and Barto, 1998). The design and implementation of value-learning modules is to be a future research focus.

2. Literature Review and Motivation

The purpose of this section is to outline several types of mechanisms common to various unsupervised learning models described within neural networks literature. Competitive learning mechanisms used for the purpose of representation discovery are primarily surveyed. Such learning mechanisms provide inspiration for, and individually contribute to the proposed invariant feature-map.

It is generally agreed that in machine learning, supervised, unsupervised and reinforcement learning are three broad classes of learning methodologies, where supervised and unsupervised learning are more closely related to each other than reinforcement learning. In the standard reinforcement-learning model, an agent is connected to its environment via perception and action, where action changes the state of the environment, and the value of this state transition is communicated to the agent through a scalar reinforcement signal (Kaelbling *et al.* 1996). Supervised learning aims to generate a function from training data, where the training data typically consists of a set of input vectors, and a target output vector. The task of the supervised learner is to predict the value of the function for any valid input object via generalizing from a small number of training examples. The goals of unsupervised learning are typically to find useful representations in the input data, such as clusters of relevant information as accomplished by k-means (Hertz *et al.* 1991), dimensionality reduction as performed by Principle Component Analysis (Jolliffe, 1986), or the building of topographic maps as done by Self-Organizing Maps, or SOMs (Kohonen, 1989, 1996; 1997; Kohonen *et al.* 1997a; 1997b; 1997c).

2.1. Competitive Learning

The most prolific application of competitive learning in neural networks research is to the common problem of finding a suitable representation of the input data, by means of a suitable transformation. Finding a suitable representation is important for the purposes of any kind of meaningful analysis on the data, such as pattern recognition, data compression, or visualization. For computational and conceptual simplicity, such a representation is often a linear transformation of the original data[3]. Competitive learning is one type of mechanism for the discovery of useful representations in data, and typically works via de-correlating activations between individual input representations. In the proposed feature-map the competition mechanism for inputs is different from standard competition mechanisms found in

[3] Well known linear transformation methods include Principal Components Analysis, or PCA (Jolliffe, 1986), Factor Analysis (Gorsuch, 1983), and Independent Component Analysis (Hyvärinen, 1999).

neural networks literature, though several standard mechanisms are outlined herein. The three widely accepted methods for performing de-correlation are *winner-take-all*, which is able to represent only small amounts of information though can be computed in linear time simply via selecting the largest activation value, *lateral inhibition*, which is computationally complex, and *inhibitory feedback*, which often de-correlates more than is desired, and requires synchronized representations and inputs.

In methods for lateral inhibition, such as Maxnet (Lippman, 1987) and the Mexican hat (Kohonen, 1989), the output of each node feeds to others through inhibitory connections. Methods for lateral inhibition are an essential feature of many self-organizing neural networks, as they provide a mechanism through which neurons compete to represent distinct patterns within the input space. Also, they can be used to de-correlate the activations of neurons, so that the response of the network is a distributed representation of the input (Foldiak, 1990). Strong lateral inhibition can be used to generate a local encoding, where a single neuron inhibits all others (Yuille and Geiger, 1995), and by allowing the lateral inhibition strength to vary as a function of the distance between neurons it is possible to preserve the topology of the input space (Swindale, 1996; Sirosh and Miikkulainen, 1994).

In methods for inhibitory feedback, which typically occur in the guise of generative models, components of the model compete for the right to represent inputs with the goal of minimizing error. Methods for inhibitory feedback work only if the representation and inputs are synchronous, multiple neurons representing the same thing are needed, and synchronization of different parts of the system often causes an oscillation of error and a de-correlation of all input, which is biologically implausible. Though, generative models and their underlying methods for inhibitory feedback are a popular approach to sensory processing, as they are capable of synthesizing sensory input patterns from underlying hidden variables.

Hebbian and anti-Hebbian learning are two complementary learning techniques where weights are adapted according to correlations between the pre-synaptic and post-synaptic neurons. Hebbian learning uses excitatory synapses where as anti-Hebbian learning uses inhibitory synapses. Hebbian learning increases the correlations between the pre-synaptic and post-synaptic neurons while anti-Hebbian decorrelates the two types of activations. Hebbian learning memorizes input-output relations though performs no form of decorrelation, so as to guarantee that the representation has a close resemblance to the inputs. Where as, anti-Hebbian learning is typically used to decorrelate the elements of the representation in order to maximize the overall information content of the representation. Both the Hebbian and anti-Hebbian learning rules state that the connection between two neurons is strengthened if they fire simultaneously, though in the case of anti-Hebbian learning the connection is inhibitory. The Hebbian and anti-Hebbian learning rules are supported by strong biological evidence (Hebb, 1949). Whilst applications of Hebbian learning include those that utilize input patterns that are either orthogonal or uncorrelated (Bechtel and Abrahamsen, 1993; Rumelhart and McClelland, 1986), applications of anti-Hebbian learning include Kohonen's (1989) construction of a novelty filter that learned to be insensitive to familiar features in its input, lateral de-correlation of feature detectors (Barlow and Faldiak, 1989; Leen, 1991) as well as removal of temporal variations from input (Mitchison, 1991).

The utilization of the winner-take-all class of competitive learning methods largely depends upon the intended application domain. One of the most common is cluster analysis, which concerns the discovery and representation of clusters of related features within a set of input data. One example of cluster analysis is vector quantization, where a competitive unit corresponds to a cluster center and an error function is specified as the sum of squared euclidean distances between each training case and the nearest cluster center (Max, 1960; MacQueen, 1967; Anderberg, 1973; Hartigan, 1975; Hartigan and Wong, 1979; Lloyd, 1982). Vector quantization frequently appears as an algorithm known as k-means clustering, in that k-means is an unsupervised winner-take-all competitive learning algorithm for clustering data into *k* clusters, and was originally designed for fitting a mixture of Gaussians (Duda and Hart, 1973). Though, a typical problem that confounds winner-take-all classes of competitive learning models is the information bottleneck that occurs, given that these models do not learn representations that are able to concurrently process multiple streams of input data (Bradley and Fayyad, 1998).

In unsupervised competitive learning, extensions of PCA and clustering methods such as the Self-Organizing Map (SOM) are often highlighted as being two related classes of models. The difference between the models concerns only the sparseness of activations in representations that each learns. Models that extend Principle Component Analysis (PCA) and Factor Analysis (FA) such as methodologies for PCA neural networks (Wang and Oja, 1993), utilize statistical methods to give a reduced subset of linear combinations of the original input variables. Methods of estimating principal components and the dynamical tracking of eigenvectors using neural networks has become an important research field given its applications (Baldi and Hornik, 1995), (Diamantaras and Kung, 1996), (Hornik and Kuan, 1992). These methods often assume an important role in neural networks that learn invariant features, and also have applications that include face recognition (King and Xu, 1995), image compression (Wang and Oja, 1993), and signal processing (Palmieri *et al.* 1993).

A typical Self Organizing Map (Kohonen, 1989) is given a set of high dimensional input feature vectors, which are mapped onto a low dimensional space of output neurons that take part in a global competitive learning winner-take-all process. SOM methods often assume an important part of clustering and dimensionality reduction applications. The Adaptive Subspace SOM (ASSOM) is a modular neural network where the modules learn to identify input patterns that are subject to simple transformations (Kohonen, 1995a; 1995b; 1996), and was considered a generalized methodology for invariant feature extraction. ASSOM is related to SOM in that it also uses a competitive unsupervised learning process. Whilst the original SOM architecture employs both vector quantization, dimensionality reduction, and attempts to retain the topology of the input space as much as possible, in ASSOM, adaptation now takes place with relation to episodes of the input data. Episodes are collections of locally shifted finite segments of the input data that contain features common to all segments, and typically retain invariant features (Kaski, 1997). The ASSOM is typically used to extract invariant features with filters that are formed automatically based on short sequences of input samples during the learning process.

It is important to note that both PCA and a SOM can be implemented using different forms of lateral inhibition. For example, PCA typically uses anti-Hebbian learning, while a SOM uses a fixed form of global competition for activations. Where as, the approach proposed in the methodology of this deliverable will utilize a sparse form of coding that can be categorized as *middle ground* between the forms of lateral inhibition used by PCA and a SOM.

Where as in PCA, the principal components are orthogonal and the projections of the data onto them are linearly decorrelated, ICA has the goal of attaining a linear transformation to coordinates in which the data are maximally statistically independent, not merely decorrelated. That is, ICA is a method of separating independent sources, which have been linearly mixed to produce multivariate data. Such multivariate data can potentially originate from many different kinds of application domain particulars such as digital images, document databases, economic indicators and psychometric measurements (Hyvärinen and Hoyer, 2002; Hyvärinen *et al.* 2001; 2003; Pajunen, 1996; Comon, 1991; Bartlett *et al.* 1998; 2000; Bartlett and Sejnowski, 1997; Donato *et al.* 1999). ICA uses a model of statistically independent representations, and assumes that independent components are non-gaussian in constructing new features, which are the independent components of a data set.

Slow Feature Analysis (SFA) is another statistical learning technique, which is typically used for the learning of invariances from temporal input sequences (Mitchison, 1991), (Becker, 1993), (Stone, 1996a; 1996b; 1996c). Traditionally, the problem of invariant feature learning was considered one of the most difficult problems in the theory of perception, though in recent years several important advances have been accomplished such as the use of Slow Feature Analysis (SFA). It is widely agreed that invariances are objects of relatively fixed structure, which affect one's senses in greatly variable ways during a sensory experience (Malsburg, 1981). In order to learn invariances, and to draw the same conclusions from the perspective of an object irrespective of variations in appearance, it is important to reduce this variability to an invariant representation of the intrinsic structure of the object (Sutherland, 1968).

SFA is a method for learning invariant or slowly varying features from a vector input signal, where learning is based upon nonlinear expansion of the input signal and application of PCA to this expanded signal and its time derivative. SFA has the advantage that since the learned input-output functions are non-linear the algorithm can be applied repeatedly so that high-dimensional input signals can be processed and a large number of uncorrelated complex features can be extracted in a hierarchical network of SFA-modules with minimal computational effort. The hierarchical network of SFA modules can learn translation, size, rotation, contrast, or, to a lesser degree, illumination invariance for one-dimensional objects, depending on only the training stimulus, though only a few training objects suffice to achieve good generalization to new objects, and performance degrades if the network is trained to learn multiple invariances simultaneously. SFA makes the assumption that primary sensory signals vary quickly while the perceived environment changes slowly. So, if one succeeds in extracting slow features from the quickly varying sensory signal, one is likely to obtain an invariant representation of the environment.

The disadvantage of PCA and related models such as ICA (Hyvärinen *et al.* 1999; 2000a; 2000b; 2000c) and SFA (Berkes and Wiskott, 2002), (Wiskott and Sejnowski, 2002), (Oja *et al.* 1996), (Yang and Wang, 1999) is that computational complexity derived from the number learning parameters typically scales quadratically with relation to the number of neurons or representational units comprising the model. For example, in PCA the number of learning parameters is proportional to the number of inputs multiplied by the size of an input representation. So, increasing the size of the input dimensions and the size of an input representation by 10-fold, would lead to a 100-fold increase in the number of learning parameters meaning that one would require a 10-fold increase in the number of training samples. An estimate of computational complexity can then be derived as the number of learning parameters times by the number of training samples which leads to an intractable 1000-fold increase in total training time.

An important aspect common to all competitive learning models and methods is the manner in which information is represented in the learning model. In neuroscience literature it is generally agreed information could be represented either locally by the activations individual neurons or globally by an activation pattern that accounts for all neurons. The term sparse coding refers to a representational state that occurs between these two extremes (Földiák and Young, 1995). In coding schemes it is important to differentiate between local, sparse and dense coding. In local coding, each state is represented by a single active neuron in a population where all other units are non-active. In dense coding, each state is represented by approximately half of the neurons in the population being active, while in sparse coding only a small fraction of neurons are active at any one time.

The representational capacity of local codes is very limited and they can represent only as many states as the number of neurons in the pool, which is insufficient for any but the most trivial tasks. Alternatively, the representational capacity of dense codes is a very high number of states and is typically applied in minimizing the number of neurons needed to represent information. Dense codes can represent two to the power of n, where n is the number of neurons in the population. The main disadvantage is that the mapping between a dense representation and an output could result in a linearly non-separable function, therefore requiring a complex implementation of multi-layer networks and learning algorithms. Sparse coding schemes combine the advantages of local and dense coding whilst avoiding most of the disadvantages. That is, codes with low activity ratios still have sufficiently high representational capacity, while the number of input-output pairs that can be stored in an associative memory is far greater for sparse than for dense patterns (Meunier and Nadal, 1995).

3. Invariant Feature-Map

An invariant feature-map is the basic building block of the proposed complete unsupervised learning model, which is to be a hierarchy of such feature maps. As stated in the motivation section of this document, many of the design principles and theories binding an individual feature-map are modified versions of existing models found in neural networks literature. In contrast to standard competitive learning models, the proposed approach of combining complementary competitive learning approaches yields advantages of computational efficiency and the capability to concurrently process large amounts of input data. This capability was the only missing aspect that could not be found in the literature, which is to be an important requisite of the overall learning model. The advantage of the proposed learning model is that the computational complexity and the number of learning parameters scale linearly with relation to the number of representational units in the model.

This section describes the methodology of a single feature-map, which at the date of this deliverable is at the prototypical stage of implementation. As illustrated in *figure 1* the feature-map is a layered structure consisting of two layers, inter-connected by a fixed set of positive weights that are defined *a priori*. The first layer is the elementary feature layer, and the second layer is the invariant feature layer. At the present stage of feature-map implementation, learning mechanisms operate only at the elementary feature layer because there are not yet lateral or top-down connections.

The elementary feature layer processes bottom-up input data and uses competitive learning mechanisms in order to learn to extract and cluster similar elementary features in an unsupervised manner. The competitive learning mechanisms utilized at the elementary feature layer encourage similar elementary features that are implicit in the input data to be represented in spatial and temporal neighborhoods at the elementary feature layer. The invariant feature layer differs in that it processes and clusters neuron activations from neighborhoods of elementary features in order to represent invariant features. Specifically, the purpose of the invariant feature layer is to integrate the neural activations from the local neighborhoods at the elementary feature layer so as to attain similar groupings of neuronal activations as a means of representing invariant features at the invariant feature layer.

It is hypothesized that the long-range excitatory connections[4] depicted in *figure 1* as connecting spatially disparate neurons at the invariant feature layer will encourage cooperation, via forming connections between topologically distant neighborhoods of neurons. The primary advantage that the proposed invariant feature-map maintains over other invariant feature learning models such as PCA, ICA, and SFA is its computational efficiency and capability to concurrently process large input data sets. Specifically, a neighborhood of the invariant feature-map processes a finite number of inputs and not all of the inputs as a typical competitive learning model does.

[4] As part of the invariant feature layer, long-range excitatory connections are to be a subject of future research.

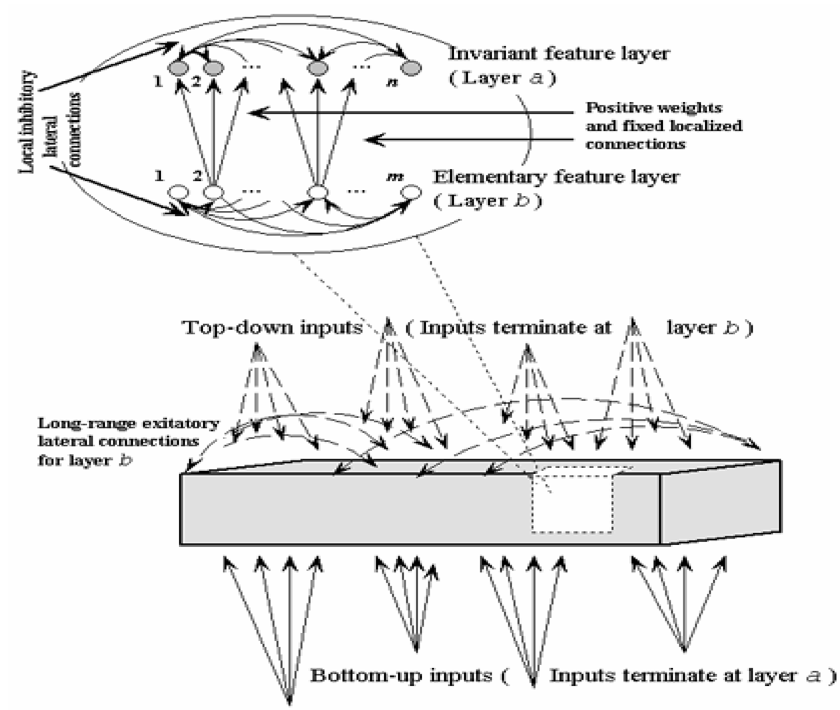

Figure 1: The elementary and invariant feature layers of a single feature-map. The interaction of complementary local and global competition mechanisms serve to cluster elementary features in temporal and spatial neighborhoods. The purpose of the invariant feature layer is to gather the overall activity of the temporal and spatial elementary feature neighborhoods in order to represent invariant features.

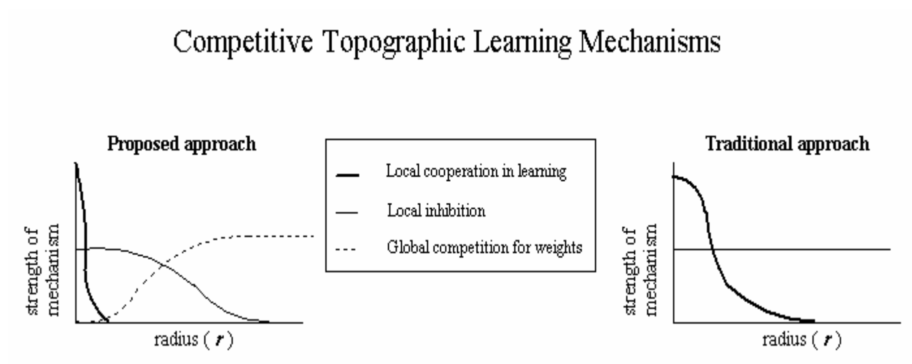

Figure 2: Comparison of competitive learning mechanisms used in the proposed feature-map versus traditional mechanisms.

For example, in our proposed model a 10-fold increase in the size of the input dimension and the size of an input representation leads to a 10-fold increase in the size of training samples, though these dimensions increase simultaneously. This leads to a 10-fold increase in the number of learning parameters and the total-training time. Hence, the number of learning parameters in our proposed model is proportional to size of a training sample multiplied by a constant. Thus, a 10-fold increase in the size of the input dimension and the size of input representation leads to 10-fold increase in the number of learning parameters, increasing total training time by only 10-fold.

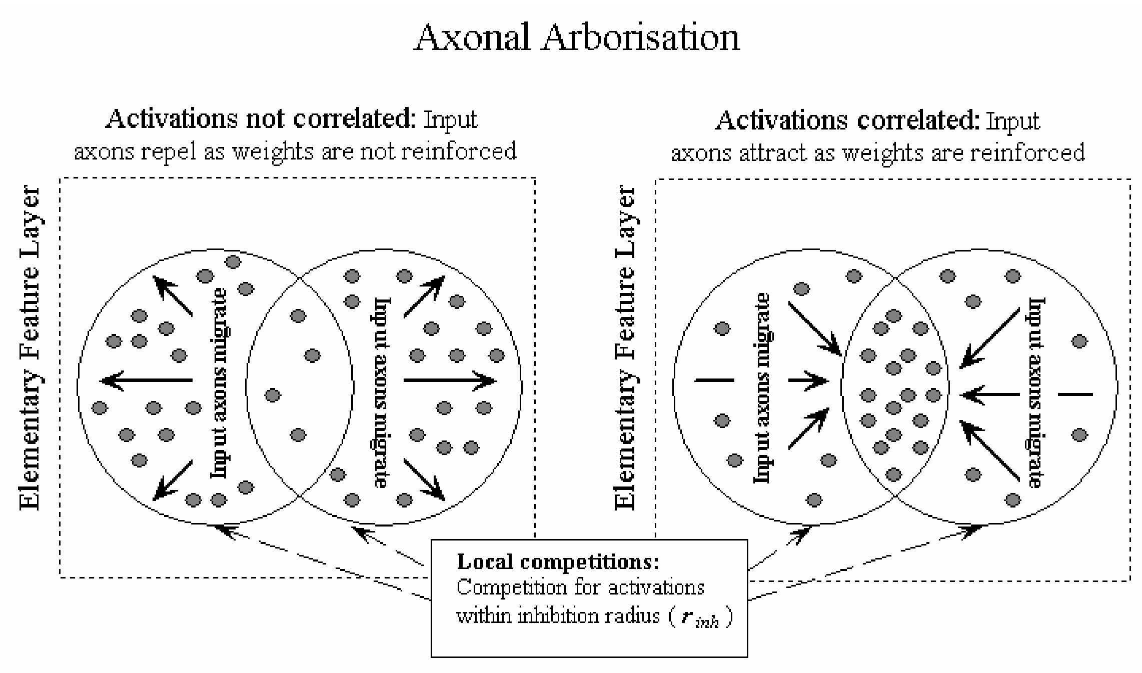

Figure 3: **Depicts the migration of input axons at the elementary feature layer. Note that, the competition radii with centers defined by the *center-of-mass* of different input axons overlap on the elementary feature layer. When input weights are few and weak, the bottom-up input axons migrate to other areas of the elementary feature layer. Where as, when input weights are relatively numerous and strong, different input axons attract as the learning rule causes these weights to be strengthened.**

3.1. Competition Mechanisms

The feature-map unsupervised learning approach combines local and global competition mechanisms, for the purpose of extracting elementary and invariant features within a layered model. Traditional global competition mechanisms such as winner-take-all have the disadvantage that they are unable to represent information in concurrent processing streams, and they are not able to process large amounts of information.

Local competition mechanisms are able to process large amounts of information but in order to avoid redundancy in the representation, inputs have to maintain restricted arbors. This works well in simple domains where the spatial structure of the problem suggests a topography, which can be used to predefine the connections. In complex multi-modal representations the problem is that it is impossible to know in advance how the bottom-up inputs from different modalities should be connected.

We propose a novel global competition mechanism, which is based on restricted axonal arbors. The elements of the representation compete locally for activations and globally for their bottom-up input connections. *Figure 2* illustrates the difference between traditional topographic learning mechanisms (right) and our approach (left). The proposed approach has the advantage that it is computationally efficient, is able to process large amounts of input data concurrently, and the training time and computational complexity scales linearly with the size of the models' representational units and the size of the input data set.

The crucial feature of the global competition mechanism is the restricted axonal arbor whose location shifts adaptively during learning. During learning, bottom-up connections can form only if the post-synaptic neuron is within a certain radius from the center of the axonal arbor. Local competition of activations only needs to take place within a similar radius because there is no risk of having redundant representation if neurons are so far away that they cannot share inputs. Our model differs from previous models with restricted arbor sizes in that the center of axonal arbors is adapted. The center is defined to be the center of mass of the weights that the arbor has. When the weights are adapted, the arbor may shift.

Figure 3 illustrates two different arborisation scenarios. First, where axon activations are correlated, they build relatively stronger weights in the area where the axons overlap and consequently two different input axons migrate to this spatial grouping, and second where activations are not correlated, the weights are weaker and different input axons migrate away from the area with overlap. It is important to note that axonal arborisation does not actually occur due to attraction and repulsion of axons but rather as a result of a particular axon finding the areas where they are allowed to build stronger weights. This clusters axons and serves to cluster related information together.

The combination of the proposed local and global competitive learning mechanisms, as well as a yet to be implemented local cooperation mechanism are illustrated together in *figure 4* as a description of the overall topographic learning model at the elementary feature layer. The areas of effect of these learning mechanisms are depicted in the figure as radii 1, 5, and 4 for local competition, local cooperation and global competition respectively. Radii 2 and 3 (not depicted in the figure) denote the areas of effect for competition mechanisms operating at the invariant feature layer, where each of these is a localized divisive competition mechanism. Specifically, these areas of effect are the spread of connections[5] from the elementary to the invariant feature layer, and the local competition radius at the invariant feature layer, respectively.

3.2. Computation for Activations

Global competition for bottom-up input weights (*figure 1*) and local competition for activations (*figure 3*) occur on different timescales. That is, local competition for activations occurs within brief time periods, where as global competition for bottom-up inputs occur within relatively longer time periods as this mechanism affects the topography of the feature-map. The local competition mechanism is subtractive and is implemented such that competition takes place within a defined neighborhood of neurons. This mechanism is generally formalized as presented in *equation 1*, where a is total activation and a' is average activation, that is, the spatial average taken across a set of elementary feature layer neurons.

[5] Given a spatial and temporal ordering of elementary features at the elementary feature layer, elementary features are integrated at the invariant feature layer in order to define and extract invariant features. In order to accomplish this a spread of connections from localized elementary feature neighborhoods to localized invariant feature neighborhoods needs to be learnt.

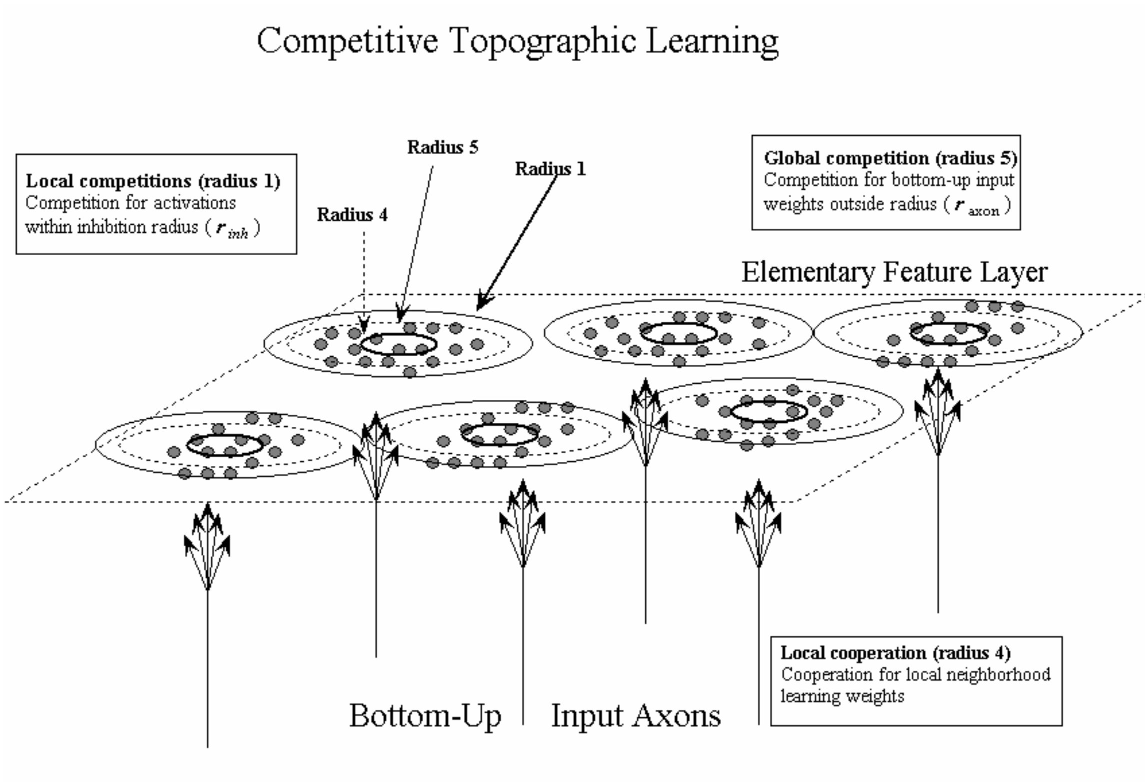

Figure 4: **Depicts competitive topographic learning at the elementary feature layer. Note the local and global competition mechanisms that are illustrated. Specifically, global competition for input weights and local competition for activations within neighborhoods of neurons.**

$$f = (a - a')$$

Equation 1: **A subtractive competition mechanism is implemented as a means of competing for activations.**

The localized competition mechanism at this feature layer is such that lateral connections within a pre-defined radius are inhibitory, and the learning of connections are discouraged with neurons that are spatially distant from the center of the competition radius. Specifically, this lateral competition mechanism is implemented as a leaky integrator, such that the greater the distance spatially from the center of the competition radius, the greater the level of inhibition. This competition mechanism is actually implemented using two functions, termed *rectify* and *smooth* presented in *equations 2* and *3* respectively.

$$\text{Rectify}(\xi) = \begin{cases} \xi, & \xi \geq 0; \\ 0, & \xi < 0 \end{cases}$$

Equation 2: **The rectify function implements a local competition mechanism at the elementary feature layer, where ξ denotes the activation of a given neuron.**

As presented in *equation 2*, the *rectify* function implements a competition mechanism where pre-processing is done, so that only nodes with activation values above the average and within a competition radius will be able to form connections to the invariant feature layer. The rectify function assigns a zero activation to neurons that have below average activation, and grades neurons that have activations above the average as winners, meaning that there are multiple winners, each positioned within multiple competition neighborhoods. Thus, each of these winners maintains an activation *x* above the neighborhood average and will be surrounded by neurons with zero activation. Note that in *equation 2*, ξ denotes the activation level of a given neuron.

$$a = \text{rectify}\left(a^* - \text{smooth}\,(a^*) \right)$$

Equation 3: **The local competition mechanism for the elementary feature layer first smoothes activation values within a local neighborhood and then inhibits lateral connections to neurons with below average activation.**

As presented in *equations 3* and *4*, the purpose of the *smooth* function is to have all elementary feature layer neurons within a certain competition neighborhood connect only to neurons that are spatially close within the invariant feature layer. So, the smooth function in effect applies a gaussian curve from the center-of-mass of a given input axon such that the greater the topographic distance that another neuron is from this center-of-mass, the less chance that a connection will be made.

$$b = \frac{b^*}{\left(\theta + \text{smooth}\,(b^*) \right)}$$

$$\text{where: } b^* = \text{smooth}(a) + \text{lateral} + \text{top-down}$$

Equation 4: **The smooth function serves to specify the radius of spread of connections from the elementary feature layer to the invariant feature layer, such that neurons are ordered in space and time at the invariant feature layer.**

The result of the smooth function is that connections between the elementary feature layer and the invariant feature layer will be localized such that connections will be made from the center-of-mass of a bottom-up input axon in a given local competition neighborhood to a particular topographical grouping of neurons at the invariant feature layer. Thus, there is local competition for inputs at the invariant feature layer but no global competition as occurs at the elementary feature layer. Regarding *equation 4*, it is important to note that *lateral* and *top-down* are to be later additions, which will specify the computation of lateral and top-down connections, and θ serves as a threshold constant with the purpose of mediating activation normalization, such that if the activation is low then it will not be normalized to too high a level.

Aside from local competition for activations, the global competition mechanism implemented for bottom-up inputs uses a learning rule as a means of gathering overall activations for localized spatial and temporal neighborhoods, and facilitating axonal arborisation of the bottom-up inputs. That is, the result of global competition for input weights (input axons) is that these axons migrate depending upon if activation weights within a particular spatial grouping correlate or not (*figure 3*).

3.3. Learning rules

The learning rule used to learn the bottom-up input weights is an activity dependent form of Hebbian learning, where learning is modulated by a "neuro-modulator" which serves to determine the level of activity within spatially and temporally arranged competitive neighborhoods of neurons. Additionally, there is a penalty for weights that are located too far from the center-of-mass of a given input axon-arbor.

This learning rule is presented in *equation 5*, where H(c) is a neighborhood function that discourages connections made to topographically disparate neurons and encourages spatial ordering of neurons. Note that, the $*$ denotes point-wise multiplication.

$$\Delta W \propto \left((a\,r) * x^{T} \right) * H(c)$$

Equation 5: **Modified Hebbian learning rule used to learn bottom-up input weights to elementary feature layer.**

Note that: H(c) can be expanded to *equation 6*, where: h_{ij} (C_{xi} , C_{yi} , C_{xj} , C_{yj}) are the coordinates which are computed based on connection weights, where *C* denotes the coordinates of the center-of-mass of a given input axon arbor, and where: *i* is *ith* input, and *j* is *jth* feature map.

$$H_{ij} = e^{-d^{2}_{ij} / 2\sigma^{2}}$$

$$\text{where: } d^{2}_{ij} = (C_{xi} - C_{xj})^{2} (C_{yi} - C_{yj})^{2}$$

Equation 6: **At the elementary feature layer H(c) is a neighborhood function that discourages connections made to topographically disparate neurons and encourages spatial ordering of neurons, where *c* denotes the coordinates of the center of mass connections in the bottom-up input axons.**

The weight update rule for time $t + 1$ is presented in *equation 7*, where *x* is input at time *t* and *b* corresponds to activations for invariant feature layer, and the smooth function serves to spatially and temporally gather neurons. Note that τ corresponds to the time scale of the integration as utilized by the leaky integrator.

$$r(t) = \left(1 - \frac{1}{\tau}\right) r\left(t - 1\right) + \left(\frac{1}{\tau}\right) \text{smooth}(b)$$

Equation 7: **Update rule for the learning rate, in the modified Hebbian learning rule, where the signal to noise ratio is mediated by *R*.**

Using the complementary local and global competitive learning mechanisms at the elementary feature layer, and a trial set of input data, we were able to execute several sets of simulations to achieve the migration of axonal arbors. These preliminary results are discussed in the following section, though it is important to note the local cooperation mechanism, as discussed in the methodology section, was implemented only in the form of spatial cooperation within a local neighborhood of neurons at the elementary feature layer. Additionally, these experiments concern only the elementary feature layer so competition mechanisms that operate at the invariant feature layer were not implemented.

4. Preliminary Results

In order to prove that the self-organizing mechanism works, an experiment was conducted in which computer-generated data, specifically the input data sample presented in *figure 5*, was presented with a certain degree of spatial correlation, but no temporal correlation. This data was used to produce a regular topology in the elementary feature layer. The data consisted of a sequence of 1-dimensional input patterns of colored noise, which were produced by smoothing and rectifying Gaussian noise. A 50x1 vector represented these input patterns, where each entry corresponded to specific axon activation.

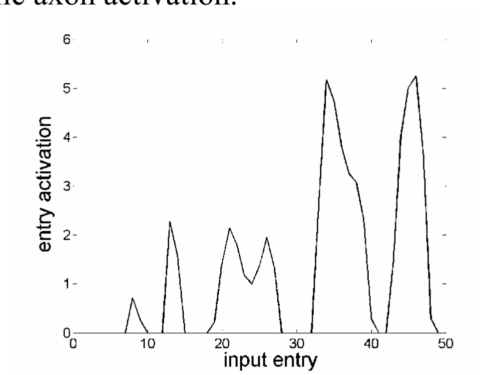

Figure 5: **Input data sample for the elementary feature layer, which was generated in Mat Lab.**

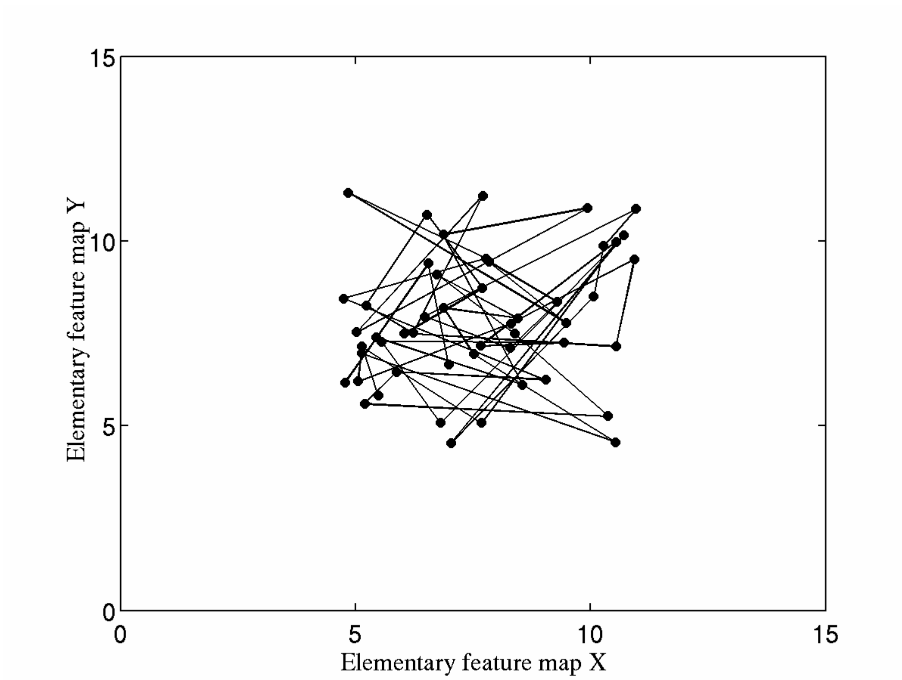

Figure 6: **The initial organization of the input axon arbors.**

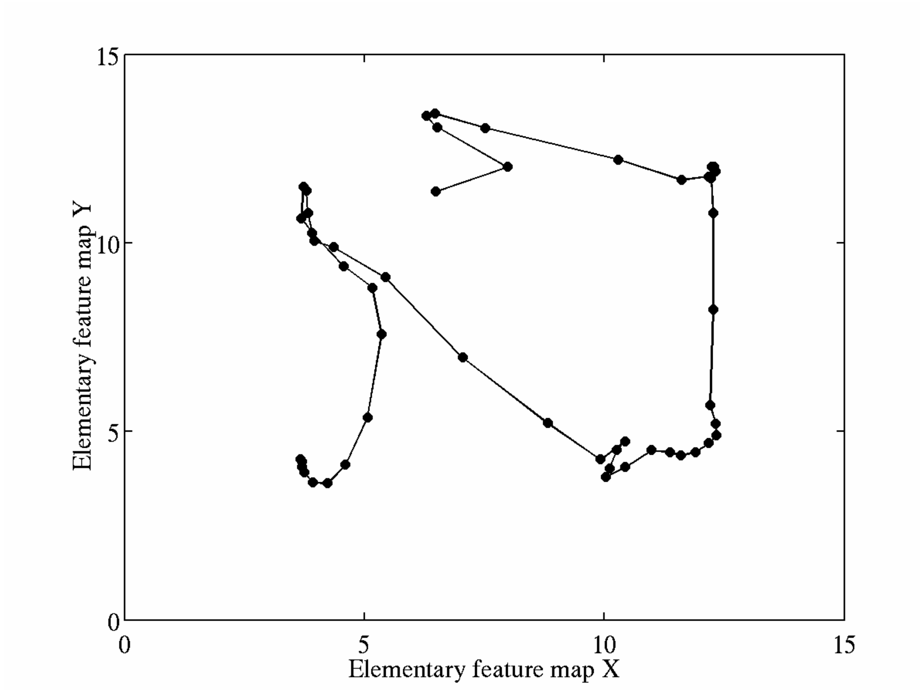

Figure 7: **The organization of the input axon arbors after the presentation of 2500 input patterns.**

Initially, the axon arbors are distributed randomly over an elementary feature layer, which is comprised of 15x15 neurons. The initial organization is depicted in *figure 6*, where the 50 positions of each axon arbor's center of mass are plotted on the elementary feature layer. The lines connecting these different centers of mass correspond to each axon's relative position in the input pattern with respect to the other axons. It is evident that at the start of the experiment the positions of these arbors on the elementary feature layer lack any meaningful organization. *Figure 6* presents the initial organization of input axon arbors, where as *figure 7* and *figure 8* show the organization of the arbors after the presentation of respectively 2500 and 5000 input patterns. Clearly can be seen that the arbors are able to disentangle themselves and distribute evenly over the elementary feature layer. In order to understand why this happens one must note the particular dynamical characteristics of axon migration. Each neuron on the elementary feature layer maintains a fixed amount of synaptic weights. All arbors compete for the weights and consequently repulsion exists between all arbors. In the meantime, arbors of axons that are quite uncorrelated repel more than those that have more correlation. Though, in our case correlation occurs because the axons are spatially close. The migration routes of arbors therefore resemble the paths of least resistance.

It is important to note that the ability of the map to organize itself is quite dependent on a number of parameters. For one, the standard deviations of the local and global competition loci should be related to the distribution of related arbors. If related arbors are far apart and the standard deviations are too small, these arbors will not be able to 'find' one another. Secondly, the feature map should provide enough room for arbors to actually move around.

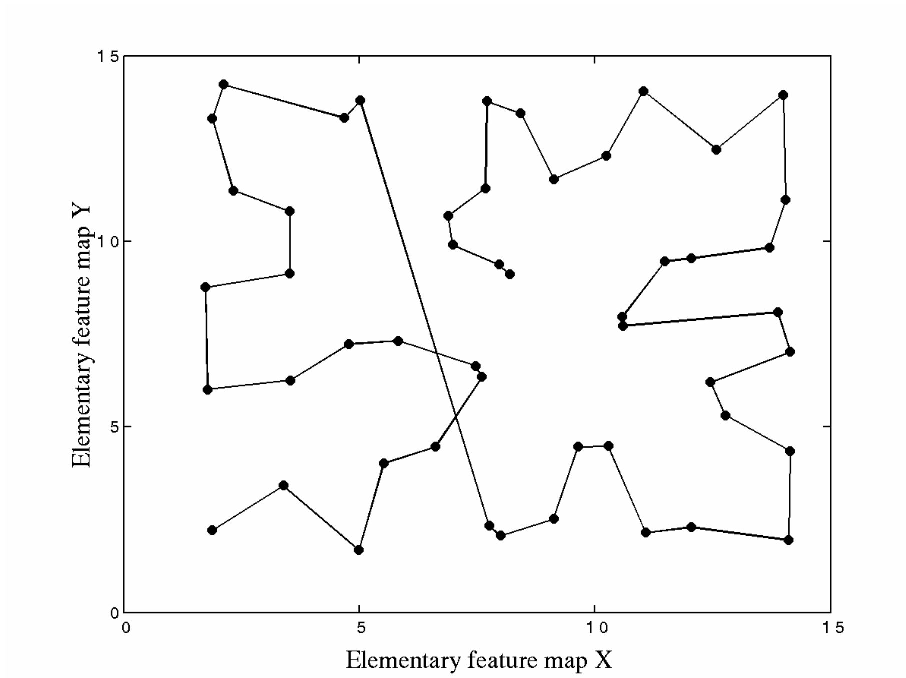

Figure 8: **The organization of the input axon arbors after the presentation of 5000 input patterns.**

Additionally, it is important to note that in our eventual implementation the arbors are already given a coarse organization on the feature map *a priori*, so the arbors do not have to migrate very much. The feature layer size can therefore be smaller, as well as the standard deviations of the competition mechanisms.

5. Conclusions

In conclusion, the experiments have highlighted that the use of complementary competitive learning mechanisms serve as a self-organizing mechanism of the elementary feature layer, and are able to organize randomly distributed axon arbors quite effectively. Although for an initial random organization the parameter tuning is reasonably sensitive, this should not affect the learning model too much, as for our purposes the arbors will be given a basic structured organization anyway, thus sparing them from the need to migrate long distances. Though as the next step in the design of the learning model it is important to illustrate that this self-organization of migrating axon arbors can lead to the formation of invariants in the invariant feature layer, which will be reported upon in the deliverable 5.3.

6. Future research

This document presented the design principles and functional outline for only a single invariant feature-map unit. *Deliverable 5.3: Initial implementation of the integration model* will report preliminary results from initial experiments that use a hierarchy of invariant feature-map units. Though, there are several important points, mentioned throughout this document that were not implemented as part of the invariant feature map unit, and are thus to be the subject of future research. Such points include that will be addressed in deliverable 5.3 include the derivation and implementation of mechanisms for habituation and novelty detection, as well as multi-modal integration and transfer within a hierarchy of feature maps for the purpose of testing the hypothesis that particular connections made between multiple invariant feature-maps will give rise to particular multi-modal representations.

References

Anderberg, M.R. (1973), Cluster Analysis for Applications. Academic Press, Inc. New York, USA.

Barlow, H., and Foldiak, P. (1989). Adaptation and decorrelation in the cortex, In, *The Computing Neuron*, editors: Miall, C., Durbin, R., and Mitchison, G., pages: 54-72. Addison-Wesley, Wokingham, England.

Baldi, P., and Hornik, K. (1995). Learning in neural networks: A survey. IEEE Transactions on Neural Networks, vol. 6(4): 837-858. IEEE Neural Networks Council, Piscataway, USA.

Bartlett, S., Lades, M., and Sejnowski, T. (1998). Independent component representations for face recognition. In, *Proceedings of the International Conference on Human Vision and Electronic Imaging III,* pages: 528-539, editors: Rogowitz, B., and Thrasyvoulos, P. SPIE International Society for Optical Engineering, Bellingham, USA.

Bartlett, S., and Sejnowski, T. (1997). Viewpoint invariant face recognition using independent component analysis and attractor networks. In, Advances in Neural Information Processing Systems 9, editors: Mozer, M., Jordan, M., and Petsche, T., pages: 817-823. MIT Press, Cambridge, USA.

Bartlett, M., Donato, G., Movellan, J., Hager, J., Ekman, P., and Sejnowski, T. (2000). Image representations for facial expression coding. In, *Advances in Neural Information Processing Systems* 12, editors: Solla, S., Leen, T., and Mueller, K. pages: 886-892. MIT Press, Cambridge, USA.
Bechtel, W., and Abrahamsen, A. (1993). *Connectionism and the mind: An introduction to parallel processing in networks*. Blackwell Publishers, Oxford, England.

Becker, S. (1993). Learning to Categorize Objects Using Temporal Coherence. In, *Advances in Neural Information Processing Systems 5*., Editors: Hanson, S., Cowan, J., and Giles, C., pages: 361-368. Morgan Kaufmann Publishers, San Mateo, USA.

Berkes, P., and Wiskott, L. (2002). Applying Slow Feature Analysis to Image Sequences Yields a Rich Repertoire of Complex Cell Properties, *Lecture Notes in Computer Science: Proceedings of the 2002 International Conference on Artificial Neural Networks*, pages: 81-87. Springer-Verlag Heidelberg, Germany.

Bradley, P., and Fayyad, U. (1998). Refining initial points for K-means clustering. In, *Proceedings of the International Conference on Machine Learning*, editors: Shavlik, J., pages 91-99. Morgan Kaufman Publishers, San Francisco, USA.

Comon, P. (1991). Independent component analysis. In, *Proceedings of the International Workshop on Higher-Order Statistics*, pages: 111-120. Institute of Electrical and Electronic Engineers, Piscataway, USA.

Diamantaras, K., and Kung, S. (1996). Principal Component Neural Networks: Theory and Applications. John Wiley and Sons. New York, USA.

Donato, G., Bartlett, M., Hager, J., Ekman, P., and Sejnowski, T. (1999). Classifying Facial Actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21(10): 974-989. Institute of Electrical and Electronic Engineers, Piscataway, USA.

Duda, R., and Hart, P. (1973). Pattern Classification and Scene Analysis. John Wiley and Sons, Indianapolis, USA.

Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, vol. 64(1): 165-170. Springer-Verlag Heidelberg, Germany.

Földiák, P. and Young, M. P. (1995). Sparse coding in the primate cortex. In, *The Handbook of Brain Theory and Neural Networks*, editor: Arbib, M., pages 895-898. The MIT Press, Cambridge, Massachusetts.

Hartigan, J.A. (1975), Clustering Algorithms. John Wiley and Sons. New York, USA.

Hartigan, J.A., and Wong, M.A. (1979). Algorithm AS136: A k-means clustering algorithm. *Applied Statistics*, vol. 28(1): 100-108. Taylor and Francis Group, London, England.

Hebb, D. (1949). The organization of behavior. John Wiley and Sons. New York, USA.

Hertz, J., Krogh, A., and Palmer, R. (1991). Introduction to the theory of Neural Computation. *Lecture Notes Volume in the Santa Fe Institute Studies in the Sciences of Complexity*. Addison-Wesley Publishing Company, Redwood City, USA.

Hornik, K., and Kuan, C. (1992). Convergence analysis of local feature extraction algorithms, Neural Networks, vol. 5(1): 229-240.

Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys*, vol. 2(1): 94-128. Lawrence Erlbaum Associates, Mahwah, USA.

Hyvärinen, A., and Hoyer, P. (2000a). Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, vol. 12(7): 1705-1720. MIT Press, Cambridge, USA.

Hyvärinen, A., and Hoyer, P. (2000b). Emergence of Topography and Complex Cell Properties from Natural Images using Extensions of ICA. In, *Advances in Neural Information Processing Systems 12*, pages: 827-833, editors: Solla, S., Leen, T., and Muller, K. MIT Press, Cambridge, USA.

Hyvärinen, A., and Hoyer, P. (2000c). Independent Component Analysis Applied to Feature Extraction from Color and Stereo Images. *Network: Computation in Neural Systems*, vol. 11(3): 191-210. Institute of Physics and IOP Publishing Limited, Bristol, England.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley and Sons. New York, USA.

Hyvärinen, A., Hoyer, P., and Inki, M. (2002). Topographic Independent Component Analysis. *Neural Computation*, 13(7): vol. 1525-1558. MIT Press, Cambridge, USA.

Hyvärinen, A., Hoyer, P., and Hurri, J. (2003). Extensions of ICA as models of natural images and visual processing. In, *Proceedings of the International Symposium on Independent Component Analysis and Blind Source Separation*, pages: 963-974, editors: Almeida, L., Cardoso, J., Jutten, C., Oja, E., and Sejnowski, T. IEEE Signal Processing Society Press, Nara, Japan.

Jolliffe, I. (1986). *Principle Component Analysis*. Spring-Verlag, New York.

Kaski, S., (1997). Data exploration using self-organizing maps. In, Mathematics, Computing and Management in Engineering Series No. 82, pages: 57-60. Finnish Academy of Technology, Espoo, Finland.

King, I., and Xu, L. (1995). Using global PCA generated receptive fields for face recognition. In, *Proceedings to the World Congress on Neural Networks*, volume II, pages 542-545. International Neural Network Society, Lawrence Erlbaum Associates, Inc. Washington D. C. USA

Kohonen, T. (1989) *Self-Organization and Associative Memory.* Springer, Berlin.

Kohonen, T. (1995a). The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In, *Proceedings of the International Conference on Artificial Neural Networks*, Editors: Fogelman-Soulié, F. and Gallinari, P., pages 3-10. European Neural Network Society. Paris, France.

Kohonen, T. (1995b). Emergence of invariant-feature detectors in self-organization. In, *Computational intelligence. A dynamic system perspective*., Editors: Palaniswami, M., Attikiouzel, Y., Marks, R., Fogel, D., and Fukuda, T., pages 17-31. IEEE Press, New York, USA.

Kohonen, T. (1996). Emergence of Invariant-Feature Detectors in the Adaptive-Subspace SOM. *Biological Cybernetics*, vol. 75(4): 281-291. Springer-Verlag, Heidelberg, Germany.

Kohonen, T. (1997). *Self-Organizing Maps*. Springer-Verlag, Berlin, Germany.

Kohonen, T., Kaski, S., and Lappalainen, H. (1997a). Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM. *Neural Computation*, vol. 9(6): 1321-1344. MIT Press, Cambridge, USA.

Kohonen, T., Kaski, S., Lappalainen, H., and Salojärvi, J. (1997b). The Adaptive-Subspace Self-Organizing Map (ASSOM). In, *Proceedings of the Workshop on Self-Organizing Maps*, editors: Kohonen, T., and Erkki, O., pages: 191-196. Helsinki University of Technology, Helsinki, Finland.

Kohonen, T., Bry, K., Jalanko, M., Riittinen, H., and Németh, G. (1997c). Spectral classification of phonemes by learning subspaces. In, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, editor: Olson, R., pages 97-100. IEEE Press, Piscataway, USA.

Leen, T. (1991). *Dynamics of learning in linear feature-discovery networks*. Network: *Computation in Neural Systems*, vol. 2(1): 85-105. Institute of Physics and IOP Publishing Limited, Bristol, England.

Lippman, R. (1987). *An introduction to computing with neural nets*, IEEE/ISSP Magazine, vol. 1(1): 4-22. IEEE Press.

Lowe, D. (1999). Object Recognition from Local Scale-Invariant Features. In, *Proceedings of the International Conference on Computer Vision*, pages: 1150-1157, editors: Tsotsos, J., Zucker, S., Ohta, Y., and Blake, A, IEEE Computer Society Press, Kerkyra, Greece.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, vol. 28(1): 129-137. Institute of Electrical and Electronic Engineers Press, Piscataway, USA.

MacQueen. J. (1967). Some methods for classification and analysis of multivariate observations. In, *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, editors: Le Cam, L., and Neyman, J., pages 281-297. University of California Press, Berkeley, USA.

Malsburg, C. (1981). The correlation theory of brain function. In, *Models of Neural Networks II*, pages: 95-119, editors: Domany, E., van Hemmen, J.L. and Schulten, K. Springer Verlag, Berlin.

Max, J. (1960). Quantizing for minimum distortion. *IEEE Transactions on Information Theory*, vol. 6(1): 7-12. Institute of Electrical and Electronic Engineers Press, Piscataway, USA.

Meunier, C., and Nadal, J. (1995). Sparsely coded neural networks. In, The Handbook of Brain Theory and Neural Networks, editor: Arbib, M., pages 899-901. MIT Press, Cambridge, USA.

Metta, G. (1999). Babybot: A study on sensory-motor development. *PhD Thesis*, Lira-Lab, University of Genova, Genova, Italy.

Metta, G., Sandini, G., and Konczak, J. (1999). A Developmental Approach to Visually Guided Reaching in Artificial Systems. *Neural Networks*, vol. 12(10): 1413-1427. Elsevier Science Press, Amsterdam, Holland.

Metta, G. Sandini, G., Natale, L. Panerai, F. (2001). Development and Robotics. In, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, editors: Hashimoto, S., and Takanishi, A., pages: 33-42. Institute of Electrical and Electronic Engineers Press, Tokyo, Japan.

Mitchison, G. (1991). Removing time variation with the anti-Hebbian differential synapse. *Neural Computation*, vol. 3(3): 312–320. MIT Press, Cambridge, USA.

Oja, E., Karhunen J., Wang, L., Vigario, R. (1996). Principal and independent components in neural networks - recent developments. In, *Proceedings of the Fifth Italian Workshop on Neural Networks*, editors: Marinaro M., and Tagliaferri R., pages: 16-35. World Scientific Publishing Company, Singapore.

Pajunen, P. (1996). Nonlinear independent component analysis by self-organizing maps. In, *Proceedings of the Sixth International Conference on Artificial Neural Networks,* editors: Malsburg, C., Seelen, W., Vorbruggen, J and Sendhoff, B., pages: 815-819. Springer-Verlag, Berlin, Germany.

Palmieri, F., Zhu, J., and Chang, C. (1993). Anti-Hebbian learning in topologically constrained linear networks: A tutorial. IEEE Transactions on Neural Networks, vol. 4(5): 748-761.
Rumelhart, D., and McClelland, J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1: Foundations. MIT Press, Cambridge, USA.

Sirosh, J., and Miikkulainen, R. (1995). Modeling Cortical Plasticity Based On Adapting Lateral Interaction. In, *The Neurobiology of Computation: Proceedings of the Third Annual Computation and Neural Systems Conference*, Editor: Bower, J., pages: 305-310. Book News, Inc. Monterey, USA.

Stone, J. (1996a). Learning Perceptually Salient Visual Parameters Using Spatio-temporal Smoothness Constraints. *Neural Computation*, vol. 8(7): 1463-1492. MIT Press, Cambridge, USA.

Stone, J. (1996b). Learning Stereo Disparity Using Temporal Smoothness Constraints: A Computational Model. *Spatial Vision*, vol. 10(1): 15-29. Brill Academic Publishers, Leiden, The Netherlands.

Stone, J. (1996c). A Canonical Micro-function For Learning Perceptual Invariances. *Perception*, vol. 25(1): 207-220. Pion Limited, London, England.

Sutton, R., and Barto A., (1998). *An Introduction to Reinforcement Learning*, MIT Press, Cambridge, USA.

Sutherland, N. (1968). Outlines of a theory of visual pattern recognition in animals and man. *Proceedings of the Royal Society Series B: Biological Sciences*. vol. 171(1): 297-317. The Royal Society, London, England.

Swindale, N. (1996). The development of topography in the visual cortex: A review of models, *Network: Computation in Neural Systems*, vol. 7(1): 161-247. Institute of Physics and IOP Publishing Limited, Bristol, England.

Wang, L., and Oja, E. (1993). Image compression by MLP and PCA neural networks. In, *Proceedings of the Eight Scandinavian Conference on Image Analysis*, editors: Høgda, K., Braathen, B., and Heia, K., pages: 1317-1324. Norwegian Society for Image Processing and Pattern Recognition, Tromsø, Norway.

Wiskott, L. and Sejnowski, T. (1998). Constrained Optimization for Neural Map Formation: A Unifying Framework for Weight Growth and Normalization. *Neural Computation*, vol. 10(3): 671-716. MIT Press, Cambridge, USA.

Yang, T., and Wang, S. (1999). Robust algorithms for principle component analysis, *Pattern Recognition Letters*, vol. 20(1): 927-933. Elsevier Academic Press, Amsterdam, Holland.

Yuille, A., and Geiger, D. (1995). Winner-take-all Mechanisms. *Chapter on The Handbook of Brain Theory and Neural Networks*, pages: 1228-1231, editor: Arbib, M. MIT Press, Cambridge, USA.