A Hierarchical ART Network for the Stable Incremental Learning of Topological Structures and Associations from Noisy Data

Marko Tscherepanow[a,*], Marco Kortkamp[a], Marc Kammer[a,b]

[a] *Bielefeld University, Applied Informatics*
Universitätsstraße 25, 33615 Bielefeld, Germany
[b] *Bielefeld University, Cognitive Interaction Technology – Center of Excellence*
Universitätsstraße 21–23, 33615 Bielefeld, Germany

Abstract

In this article, a novel unsupervised neural network combining elements from Adaptive Resonance Theory and topology-learning neural networks is presented. It enables stable on-line clustering of stationary and non-stationary input data by learning their inherent topology. Here, two network components representing two different levels of detail are trained simultaneously. By virtue of several filtering mechanisms, the sensitivity to noise is diminished, which renders the proposed network suitable for the application to real-world problems. Furthermore, we demonstrate that this network constitutes an excellent basis to learn and recall associations between real-world associative keys. Its incremental nature ensures that the capacity of the corresponding associative memory fits the amount of knowledge to be learnt. Moreover, the formed clusters efficiently represent the relations between the keys, even if noisy data is used for training. In addition, we present an iterative recall mechanism to retrieve stored information based on one of the associative keys used for training. As different levels of detail are learnt, the recall can be performed with different degrees of accuracy.

Keywords: topology learning, associative memory, incremental learning, hierarchical representations, Adaptive Resonance Theory

1. Introduction

For numerous tasks, the traditional off-line learning approach with separate training, validation, and test phases is not sufficient. The diagnosis of genetic abnormalities (Vigdor & Lerner, 2006), interactive teaching of a humanoid robot (Goerick et al., 2009), and the subcellular localisation of proteins (Tscherepanow et al., 2008) constitute several examples for such problems. As a consequence, incremental on-line learning has become more popular in recent years, since such machine learning techniques are required to gradually complete knowledge or adapt to non-stationary input distributions.

In this article, the TopoART network (Tscherepanow, 2010) is presented. It combines incremental and fast on-line clustering with topology learning. As TopoART originates from Adaptive Resonance Theory (ART) networks, in particular Fuzzy ART (Carpenter et al., 1991), TopoART creates stable representations while retaining its ability to learn new data. In order to render TopoART more suitable for real-world applications, it was designed in such a way that it becomes insensitive to noise. Furthermore, it creates a hierarchical representation of the input distribution reflecting different levels of detail.

TopoART can be extended to a hierarchical hetero-associative memory called TopoART-AM. Here, an iterative recall mechanism provides missing keys in decreasing order of confidence. Due to the properties inherited from TopoART, namely insensitivity to noise as well as the ability of incremental and fast on-line learning, this associative memory is particularly well-suited to real-world applications.

Related approaches are discussed in Sect. 2. Afterwards, details of TopoART and its extension TopoART-AM are introduced in Sect. 3. In Sect. 4, the results of TopoART and TopoART-AM applied to different types of datasets are compared to several state-of-the-art methods. Here, their ability to cope with noise and to incrementally learn new input data from non-stationary distributions will be shown. In addition, the iterative recall mechanism of TopoART-AM will be demonstrated. Finally, Sect. 5 summarises the most important points of this article.

2. Related Work

As we intend to solve two different types of problems using TopoART, namely clustering and the learning of associations, we discuss related work from both research fields.

2.1. Clustering Techniques

The k-means algorithm (MacQueen, 1967), which constitutes a very well-known unsupervised learning technique, de-

*Corresponding author. Tel.: +49 521 106 12222; fax: +49 521 106 2992.
Email addresses: marko@techfak.uni-bielefeld.de (Marko Tscherepanow), mkortkam@techfak.uni-bielefeld.de (Marco Kortkamp), mkammer@cit-ec.uni-bielefeld.de (Marc Kammer)

termines a partitioning of an input distribution into k regions or rather clusters. Each cluster is represented by a reference vector. The choice of the number of required clusters constitutes a crucial problem. For this reason, the Linde-Buzo-Gray (LBG) algorithm (Linde et al., 1980) was developed. Based on a fixed training set, it successively computes sets of reference vectors of increasing size until a stopping criterion is fulfilled. The topological structure of the input data is not considered by this algorithm.

In 1982, the Self-Organising Feature Maps (SOFMs), which map input data to a lattice of neurons, were introduced by Kohonen. Here, the reference vectors are encoded by the weights of the neurons. The lattice possesses a predefined topological structure, the dimension of which is usually lower or equal to the dimension of the input space. If the input distribution is not completely known in advance, an appropriate lattice structure is difficult to choose. This problem was solved by the Growing Neural Gas (GNG) algorithm (Fritzke, 1994). It allows for the incremental incorporation of new neurons and the learning of the input distribution's topology by adding and deleting edges between different neurons.

The GNG algorithm is contained as a special case in a recently proposed extension, which is called the limited branching tree Growing Neural Gas (lbTreeGNG) (Kortkamp & Wachsmuth, 2010). It creates hierarchical codebooks that locally preserve the topology of the input space, while allowing a very efficient mapping from input samples to codewords and avoiding overfitting during training.

However, the above-mentioned methods do not directly employ mechanisms that deal with the *stability-plasticity dilemma* (Grossberg, 1987). A continuing presentation of input data results in a continuing adaptation of the neurons' weights, i.e. the reference vectors, and the network topology. Thus already-learnt structures may get altered or even lost. This can occur, for instance, if the input distribution is complex or due to small changes of the input probabilities. The sequencing of the input data may cause a similar effect.

Adaptive Resonance Theory (ART) networks have been proposed as a solution to the stability-plasticity dilemma (Grossberg, 1987). These networks learn top-down expectations which are matched with bottom-up input. The expectations, which are called categories, summarise sets of input data into clusters. Depending on the type of ART network, the categories exhibit different shapes such as a hyperspherical shape (Anagnostopoulos & Georgiopoulos, 2000), a hyperelliptical shape (Anagnostopoulos & Georgiopoulos, 2001), or a hyperrectangular shape (Carpenter et al., 1991). Besides enabling ART networks to create stable and plastic representations, the categories allow for an easy novelty detection. But in contrast to SOFMs and GNG, ART networks do not capture the topology of the input data. Furthermore, their ability of stable learning leads to an increased sensitivity to noise.

In 2006, the Self-Organising Incremental Neural Network (SOINN) was introduced by Furao & Hasegawa. Similar to GNG, SOINN clusters input data by incrementally adding neurons, the weights of which represent reference vectors, and the topology is reflected by edges between the nodes. But it has several additional features: Firstly, SOINN has a two-layered structure representing the input distribution at different levels of detail. Additionally, this structure reduces the sensitivity to noise. The second layer is trained after the training of the first layer has been finished. Secondly, novelty detection can be performed based on an adaptive threshold. Thirdly, each neuron has an individual learning rate which decays if the amount of input samples that it represents increases. In this way, a more stable representation is achieved. But the weights of the neurons do not stabilise completely. Furthermore, a high number of relevant parameters (8 parameters per layer) has to be set in order to apply SOINN.

The Enhanced Self-Organising Incremental Neural Network (ESOINN) (Furao et al., 2007) solves some of the above-mentioned problems: By removing the second layer and one condition for the insertion of new neurons, the number of required parameters is considerably reduced (4 in total). Furthermore, the whole network can be trained on-line. But similar to SOINN, the weights do not stabilise completely. Moreover, ESOINN loses the ability to create hierarchical representations.

TopoART combines the advantages of ART and topology-learning networks (see Sect. 3.1). From its ART ancestors, it inherits the ability of fast and stable on-line learning using expectations (categories). These categories are extended by edges reflecting the topology of the input distribution. Therefore, they enable the formation of arbitrarily shaped clusters. In addition, TopoART adopts the ability to represent input data at different levels of detail from SOINN; but unlike SOINN, it learns both levels simultaneously.

2.2. Associative Memories

There exist several approaches to associative memories, which are based on clustering methods. Some examples are the bidirectional hetero-associative memories of Chartier et al. (2009) and of Ichiki et al. (1993), which incorporate SOFMs, as well as SOIAM (Sudo et al., 2009), an associative memory based on a simplified version of SOINN. In contrast to traditional approaches such as Hopfield networks (Hopfield, 1982) and bidirectional associative memories (BAMs) (Kosko, 1988), they do not have to be trained with noise-free input patterns and perform information compression: The underlying clusterer summarises similar input samples to clusters, which may be considered as a simple type of categorisation. As a consequence, these approaches reduce the amount of data to be stored which is a major aspect of the principle of *cognitive economy* (Goldstone & Kersten, 2003). This is particularly beneficial for artificial agents such as robots operating in real-world environments, as they have to process large amounts of noisy and corrupted data.

The capacity of Hopfield networks and BAMs depends on the size of the associative keys (Hopfield, 1982; Kosko, 1988). After the maximum capacity has been reached, further training results in forgetting the previously learnt data. SOFM-based associative memories suffer from a similar problem, although they are capable of generalisation, which increases the capacity. Since the application of SOFMs requires the topology and network size to be chosen in advance (e.g., Chartier et al., 2009;

Ichiki et al., 1993), the capacity of these methods is limited as well. Furthermore, SOFMs do not create stable representations. Hence, catastrophic forgetting might result from training with non-stationary data. In contrast, the capacity of SOIAM is not limited, as it is an incremental network. Its capacity rather fits the learnt knowledge. But similar to SOINN, the knowledge is not completely stable. Furthermore, since SOIAM is based on a one-layered version of SOINN, no hierarchical clustering is performed. This hierarchical clustering might have been beneficial for real-world tasks, as it enables the representation of further abstraction levels.

Another important aspect, which needs to be considered, is the type of information which can be processed. While Hopfield networks require binary input, BAMs allow for the storage of real-valued data. Associative memory models incorporating clustering techniques can be applied to real-valued data as well. But one data type, which typically occurs in real-world scenarios, is often neglected: colour images. Of course, such images can be transformed into vectors of real-valued data and learnt with the respective methods, but explicit evaluations are usually missing (cf. Chartier et al., 2009; Kosko, 1988; Sudo et al., 2009).

In 2007, Yáñez-Márquez et al. presented an auto-associative memory dedicated to the learning of RGB images. This model is a direct extension of binary approaches and does not involve transformations to real-valued data or clustering. But since the evaluation was performed using ten different images only, its suitability to solve real-world problems can barely be estimated.

As TopoART combines features of ART and topology-learning networks, it constitutes an excellent basis to construct an associative memory for real-world tasks: By virtue of TopoART's incremental nature, the capacity of the corresponding associative memory would not be limited and learnt associations would be stable. Furthermore, the usage of complex data, such as colour images, should be possible. Therefore, we decided to apply TopoART to the task of associative learning as well. This extension of TopoART is called TopoART-AM. Similar, to the associative memory introduced by Ichiki et al. (1993), the learning procedures of the underlying clusterer remain untouched. As a result, the properties of TopoART are directly transferred to TopoART-AM. But in order to realise the recall process, some additional mechanisms are required (see Sect. 3.2).

3. TopoART

In this section, the principal functioning of TopoART is described. Afterwards, this algorithm is extended in order to allow for the construction of an associative memory, which is referred to as TopoART-AM.

3.1. Using TopoART for Clustering

The basic structure and the computational framework of TopoART are strongly related to Fuzzy Art (Carpenter et al., 1991), which constitutes a very efficient ART network utilising hyperrectangular categories. TopoART is composed of two

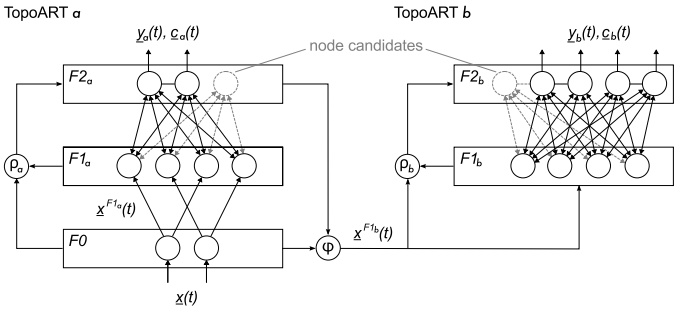

Figure 1: Structure of TopoART. TopoART consists of two Fuzzy ART-like components called TopoART a and TopoART b, which share the input layer $F0$. The propagation of input vectors to TopoART b depends on the activation of TopoART a. Furthermore, the $F2$ nodes of each component are connected by edges defining a topological structure. In order to reduce the sensitivity to noise, TopoART evaluates the benefit of neurons (node candidates) before they are fully incorporated.

Fuzzy ART-like components – TopoART a and TopoART b. These components possess a three-layered structure with a shared initial layer $F0$ (see Fig. 1). They function in an identical way and are trained in parallel. In order to create representations at different levels of detail and to reduce the sensitivity to noise, the propagation of input vectors to TopoART b is controlled by TopoART a. Additionally, the maximum category size of TopoART b is diminished in comparison to TopoART a.

The input vectors $\underline{x}(t)$ are presented to the shared initial layer $F0$.

$$\underline{x}(t) = \left[x_1(t), \ldots, x_d(t)\right]^T \tag{1}$$

At the initial layer, input is encoded using complement coding, which constitutes a concatenation of $\underline{x}(t)$ and its complement $\underline{x}^c(t)$.

$$\underline{x}^c(t) = \left[1 - x_1(t), \ldots, 1 - x_d(t)\right]^T \tag{2}$$

The complement coded vectors are denoted by $\underline{x}^{F1}(t)$.

$$\underline{x}^{F1}(t) = \left[\underline{x}(t)^T, \underline{x}^c(t)^T\right]^T \tag{3}$$

As a consequence of the usage of complement coding, each component $x_i(t)$ of an input vector $\underline{x}(t)$ has to lie in the interval $[0, 1]$.

The encoded input vectors $\underline{x}^{F1}(t)$ are transmitted to the respective comparison layer $F1$. The activation $z_i^{F2}(t)$ (choice function) of the $F2$ nodes is computed as follows:

$$z_i^{F2}(t) = \frac{\left|\underline{x}^{F1}(t) \wedge \underline{w}_i^{F2}(t)\right|_1}{\alpha + \left|\underline{w}_i^{F2}(t)\right|_1} \tag{4}$$

$z_i^{F2}(t)$ constitutes a measure for the similarity between $\underline{x}^{F1}(t)$ and the category represented by neuron i. $|\cdot|_1$ and $\wedge$ denote the city block norm and a component-wise minimum operation, respectively. The parameter α must be set slightly higher than zero. The choice of the actual value is not crucial.[1] In general, $z_i^{F2}(t)$ prefers small categories to large ones.

[1] α was set to 0.001 for all experiments presented in this article.

After all $F2$ neurons have been activated, the best-matching neuron *bm*, i.e. the neuron with the highest activation, is selected. But the category represented by its weights $\underline{w}_{bm}^{F2}(t)$ is only allowed to grow and enclose a presented input vector if resonance occurs, i.e. if the match function (5) is fulfilled.

$$\frac{\left|\underline{x}^{F1}(t) \wedge \underline{w}_{bm}^{F2}(t)\right|_1}{\left|\underline{x}^{F1}(t)\right|_1} \geq \rho \tag{5}$$

The vigilance parameter ρ limits the maximum size of the categories and has, therefore, a strong influence on the resulting clusters. But as the match function has directly been adopted from Fuzzy ART (Carpenter et al., 1991) and its supervised counterpart Fuzzy ARTMAP (Carpenter et al., 1992), it is possible to resort to existing experiences and procedures for finding adequate values for ρ (e.g., Tscherepanow & Kummert, 2007; Tscherepanow et al., 2008).

Assuming a neuron was not able to fulfil (5), its activation is reset. Then a new best-matching node is chosen. If no suitable best-matching neuron is found, a new neuron representing $\underline{x}(t)$ is incorporated and resonance occurs.

In case of resonance, the weights $\underline{w}_{bm}^{F2}(t)$ of the chosen neuron are adapted and the output $\underline{y}(t)$ of the respective TopoART component is set:

$$\underline{w}_{bm}^{F2}(t+1) = \left(\underline{x}^{F1}(t) \wedge \underline{w}_{bm}^{F2}(t)\right) \tag{6}$$

$$y_i(t) = \begin{cases} 0 & \text{if } i \neq bm \\ 1 & \text{if } i = bm \end{cases} \tag{7}$$

Using (6) the network is trained in fast-learning mode; i.e., each learnt input is enclosed by the category that matches it best. Moreover, shrinking of the categories is impossible. Hence, the formed representations are stable.

Rather than only determining the best-matching neuron *bm* and modifying its weights, the neuron *sbm* with the second highest activation that fulfils (5) is adapted as well. Here, the learning rate β_{sbm} should be chosen smaller than 1, as neuron *sbm* – in contrast to neuron *bm* – is only intended to partly learn $\underline{x}^{F1}(t)$. Its weights $\underline{w}_{sbm}^{F2}(t)$ are adapted as follows:

$$\begin{aligned} \underline{w}_{sbm}^{F2}(t+1) = {} & \beta_{sbm}\left(\underline{x}^{F1}(t) \wedge \underline{w}_{sbm}^{F2}(t)\right) \\ & +(1-\beta_{sbm})\underline{w}_{sbm}^{F2}(t) \end{aligned} \tag{8}$$

As a result of this procedure, the insensitivity to noise is increased, since the categories are more likely to grow in relevant areas of the input space. But in comparison to ρ, β_{sbm} is considerably less influential. Therefore, its purpose consists in the fine tuning of the clustering results.

Each $F2$ neuron i of both components has a counter denoted by n_i^a and n_i^b, respectively, which counts the number of input samples it has learnt. An encoded input vector is only propagated to TopoART *b* if resonance of TopoART *a* occurred and $n_{bm}^a \geq \phi$. Every τ learning cycles, all neurons with a counter smaller than ϕ are removed. Therefore, such neurons are called *node candidates*. Once n_i equals or surpasses ϕ, the corresponding neuron can no longer be removed; i.e., it becomes a permanent node. The fraction ϕ/τ gives the minimum activation frequency for neurons to become permanent. Due to this relationship, ϕ and τ function similar to a single parameter, which facilitates their choice. In addition, τ specifies the number of training steps considered for the estimation of the activation frequencies. Thus, the reliability of the estimates increases for higher values of τ. Using this mechanism for the removal of rarely activated nodes, the network is rendered more insensitive to noise but is still able to learn stable representations.

In order to enable TopoART to learn topologies, a lateral connection or rather edge between the neurons *bm* and *sbm* is created, if a second-best-matching neuron can be found. These edges define a topological structure. They are not used for activating other neurons. If the neurons *bm* and *sbm* have already been connected by an edge, it remains unchanged, since the edges do not possess an age parameter in contrast to the edges in ESOINN, SOINN, and GNG networks. They are removed if one of the adjacent neurons is removed. As a consequence, edges between permanent nodes are permanent, while edges from or to node candidates can be eliminated. In addition, it is always possible to create new edges. This mechanism constitutes an extension of Fuzzy ART's solution to the stability-plasticity dilemma, which enables the representation of new input while retaining the already-learnt representations.

The permanent nodes and edges constitute a kind of long-term memory of the network, as they enable the access to information from the past. We consider this property to be important for numerous tasks such as the life-long learning of artificial agents or categorisation processes which require the learnt categories to remain stable if new data is incorporated into the network.

The current size $S_i(t)$ of a category can be derived from the weights $\underline{w}_i^{F2}(t)$ of the corresponding neuron i:

$$S_i(t) = \sum_{j=1}^{d} \left|\left(1 - w_{i,d+j}^{F2}(t)\right) - w_{i,j}^{F2}(t)\right| \tag{9}$$

In addition to the vigilance parameter ρ, the maximum category size $S^{\max}$ is determined by the dimension of the input space d.

$$S^{\max} = d(1-\rho) \tag{10}$$

In order to refine the representation of TopoART *a* by means of TopoART *b*, ρ_b should be higher than ρ_a. Therefore, ρ_b is determined according to (11), which diminishes the maximum category size $S^{\max}$ by 50%.

$$\rho_b = \frac{1}{2}(\rho_a + 1) \tag{11}$$

In this way, TopoART *b* learns a more detailed representation which is less influenced by noise. Connections between categories of TopoART *a* can be split by TopoART *b* resulting in a hierarchical representation of the input data.

In addition to the output $\underline{y}(t)$ (7), each component provides the cluster labels of the F2 nodes as a vector termed $\underline{c}(t)$. These labels are determined as follows (cf. Furao & Hasegawa, 2006; Furao et al., 2007): First, an initial label (integer number) is chosen. Then, starting from an unlabelled neuron, all connected neurons receive this label. Afterwards, the label is increased and a new unlabelled neuron is searched for. The complete procedure is repeated until no unlabelled neurons remain. As a result, all sets of interconnected nodes or rather clusters have received a unique label. For reasons of stability, only permanent nodes are considered for the computation of $\underline{c}(t)$. By this, the clusters can grow and fuse, but they are prevented from shrinking.

In order to map unknown data to learnt clusters, TopoART determines the permanent node with the highest activation. The cluster label of this node is then returned as the result.

The original activation (4) depends on the category size. This might be disadvantageous in situations where a trained network is applied to unknown test patterns. In such a case, the presented input samples are not guaranteed to lie in existing categories. Furthermore, as no learning takes place, the current category size is irrelevant. Therefore, the alternative activation proposed by Tscherepanow et al. (2008) could be beneficial. It constitutes the city block distance between an input sample and the respective category. In order to render it more suitable for the application as an activation function, it is inverted and normalised to values from the interval [0, 1]:

$$z_i^{F2}(t) = 1 - \frac{\left|\left(\underline{x}^{F1}(t) \wedge \underline{w}_i^{F2}(t)\right) - \underline{w}_i^{F2}(t)\right|_1}{\left|\underline{x}^{F1}(t)\right|_1} \tag{12}$$

The resulting values of $z_i^{F2}(t)$ reflect the similarity of an input with a category and are not influenced by the category size.

3.2. Learning and Recalling Associations with TopoART-AM

TopoART can directly be utilised for learning associations between two keys denoted by the vectors $\underline{k}_1(t)$ and $\underline{k}_2(t)$, respectively. Both vectors have just to be concatenated and fed as input into the network:

$$\underline{x}(t) = \left[\underline{k}_1(t)^T, \underline{k}_2(t)^T\right]^T \tag{13}$$

As each $F1$ node receives only input from one $F0$ node, the $F1$ nodes and the $F0$ nodes as well as their connections correspond either to $\underline{k}_1(t)$ or to $\underline{k}_2(t)$ (see Fig. 2). Furthermore, all connections between the $F2$ layer and the $F1$ layer are assigned to one of the keys as well. Each $F2$ node has both types of connections.

This assignment to one of the keys is exploited for recall, which constitutes the key aspect of TopoART-AM. During recall, one key is presented and the corresponding second key is independently generated by both components of the network (TopoART-AM *a* and TopoART-AM *b*). For reasons of clarity, the indices denoting the respective component of TopoART-AM are omitted in the following.

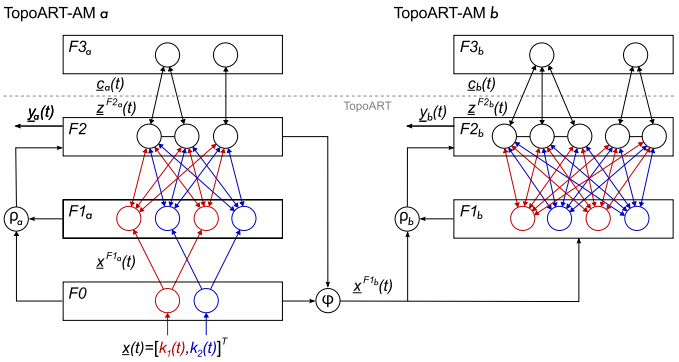

Figure 2: Structure of TopoART-AM. TopoART-AM consists of a TopoART network which is trained by inputs that are a concatenation of two keys $\underline{k}_1(t)$ and $\underline{k}_2(t)$. For simplicity, scalar keys are used in this figure. The subnets, which refer to either $\underline{k}_1(t)$ or $\underline{k}_2(t)$, are drawn in red and in blue colour, respectively. In addition to TopoART, TopoART-AM has a fourth layer ($F3$), which is used for recall. During training, the $F3$ layer is not required.

In order to recall associated keys, the respective known key is directly used as input vector $\underline{x}(t)$ leading to:

$$\underline{x}^{F1}(t) = \begin{cases} \left[\underline{k}_1(t)^T, \underline{k}_1^c(t)^T\right]^T & \text{if } \underline{x}(t) = \underline{k}_1(t) \\ \left[\underline{k}_2(t)^T, \underline{k}_2^c(t)^T\right]^T & \text{if } \underline{x}(t) = \underline{k}_2(t) \end{cases} \tag{14}$$

Due to this modification, the activation of the $F2$ nodes exclusively depends on the presented key and the subnets responsible for processing it (see Fig. 2).

$$z_i^{F2}(t) = \begin{cases} z_i^{F2,1}(t) & \text{if } \underline{x}(t) = \underline{k}_1(t) \\ z_i^{F2,2}(t) & \text{if } \underline{x}(t) = \underline{k}_2(t) \end{cases} \tag{15}$$

The weights corresponding to these subnets are indicated by $\underline{w}_i^{F2,1}(t)$ and $\underline{w}_i^{F2,2}(t)$, respectively. Since no learning takes place during recall, the alternative activation function (12) is applied.

$$z_i^{F2,1}(t) = 1 - \frac{\left|\left(\underline{x}^{F1}(t) \wedge \underline{w}_i^{F2,1}(t)\right) - \underline{w}_i^{F2,1}(t)\right|_1}{\left|\underline{x}^{F1}(t)\right|_1} \tag{16}$$

$$z_i^{F2,2}(t) = 1 - \frac{\left|\left(\underline{x}^{F1}(t) \wedge \underline{w}_i^{F2,2}(t)\right) - \underline{w}_i^{F2,2}(t)\right|_1}{\left|\underline{x}^{F1}(t)\right|_1} \tag{17}$$

In contrast to a training cycle, the match function (5) is not checked during recall. After all $F2$ nodes have been activated, the clusters are labelled; then the recall layer $F3$ is created. Each $F3$ node represents an individual cluster and is connected to all $F2$ nodes which have received the corresponding label l. Their activation constitutes the maximum activation of the $F2$ nodes it is connected to.

$$z_l^{F3}(t) = \max_{i, c_i(t)=l} z_i^{F2}(t) \tag{18}$$

After the $F3$ nodes have been activated, the iterative recall process starts: First, the $F3$ neuron with the maximum activation is determined. If multiple neurons exhibit this activation, the neuron with the smallest index is selected. This $F3$ neuron

inhibits all connected $F2$ nodes which are less activated than itself. By this procedure, the best-matching $F2$ neuron bm of the respective cluster is selected. Based on the weights of this neuron, the output of the network is computed:

$$\underline{y}(t,i) = \begin{cases} \mathrm{CoG}\left(\underline{w}_{bm}^{F2,2}(t)\right) & \text{if } \underline{x}(t) = \underline{k}_1(t) \\ \mathrm{CoG}\left(\underline{w}_{bm}^{F2,1}(t)\right) & \text{if } \underline{x}(t) = \underline{k}_2(t) \end{cases} \quad (19)$$

Here, $\mathrm{CoG}\left(\underline{w}(t)\right)$ denotes the centre of gravity of the category defined by the weight vector $\underline{w}(t)$. For n-dimensional weight vectors, it can be determined as follows:

$$\mathrm{CoG}\left(\underline{w}(t)\right) = \frac{1}{2}\begin{bmatrix} w_1(t) & + & 1 - w_{\frac{n}{2}+1}(t) \\ \vdots & & \vdots \\ w_{\frac{n}{2}}(t) & + & 1 - w_n(t) \end{bmatrix} \quad (20)$$

After the output $\underline{y}(t,i)$ for the current iteration i has been computed, the selected $F3$ neuron is reset; i.e., its activation is set to 0. Then, a new iteration starts. If i=1, the output constitutes the best solution to the reconstruction of the respective missing key. With each further iteration, the recall error increases. The recall process can be stopped if the recall error reaches a certain limit. Otherwise, it will be finished when all $F3$ nodes have been reset. After the recall process has been stopped, the $F3$ layer is removed.

4. Results

In this section, the clustering capability of TopoART is demonstrated. Then, we show that after extending TopoART by the recall mechanism proposed in Sect. 3.2, it is capable of learning and recalling associations of complex data.

4.1. TopoART

TopoART was evaluated using three different types of data: stationary artificial data, non-stationary artificial data, and stationary real-world data.

4.1.1. Stationary Artificial Data

As stationary artificial input distribution, a two-dimensional data distribution copying the one used for the evaluation of SOINN (Furao & Hasegawa, 2006) was chosen. It comprises two Gaussian components (A and B), two ring-shaped components (C and D), and a sinusoidal component (E) composed from three subcomponents ($E1$, $E2$, and $E3$). Each component encompasses 18,000 individual samples. Additionally, the input distribution includes uniformly distributed random noise amounting to 10% of the total sample number (10,000 samples). This dataset was used to train four different types of networks: Fuzzy ART, lbTreeGNG, SOINN, and TopoART. Figure 3 depicts the applied data distribution and the respective clustering results. During training, each input sample was once presented to the respective network.

As Fuzzy ART constitutes the basis of TopoART, it was analysed first. For comparison reasons, β was set to 1. Therefore, the weights of the best-matching neurons are adapted in the same manner as with TopoART. ρ was selected in such a way that the edge length of square categories roughly fits the thickness of the elliptic and sinusoidal components of the input distribution. As this network does not possess any means to reduce the sensitivity to noise, virtually the whole input space was covered by categories.

For the analysis of TopoART, the value of the vigilance parameter was transferred from Fuzzy ART (ρ_a=ρ). Since τ fulfils a similar task as the SOINN parameters λ and $\mathrm{age}_{\mathrm{dead}}$, its value was adopted from Furao & Hasegawa (2006) where a very similar dataset was used and both λ and $\mathrm{age}_{\mathrm{dead}}$ were set to 100. The remaining parameters ϕ and β_{sbm} were manually adjusted in such a way as to copy the results of SOINN published by Furao & Hasegawa (2006).[2]

In contrast to Fuzzy ART, both TopoART components created representations reflecting the relevant regions of the input distribution very well. This is remarkable since the value of ρ_a was equal to the value of the vigilance parameter ρ of the Fuzzy ART network. The representation of TopoART was refined from TopoART a to TopoART b: While TopoART a comprises one cluster, TopoART b distinguishes five clusters corresponding to the five components of the input distribution. By virtue of the filtering of samples by TopoART a and due to the fact that ρ_b is higher than ρ_a (11), the categories of TopoART b reflect the input distribution in more detail. This property is particularly useful if small areas of the input space have to be clustered with high accuracy. Here, TopoART a could filter input from other regions and TopoART b could create the desired detailed representation.

The lbTreeGNG network was trained with the default values for the parameters ϵ_b, ϵ_n, $a_{\max}$, α, d, and λ as stated in (Kortkamp & Wachsmuth, 2010). In addition, the parameter b for the limiting branching factor was set to 90 and the error threshold m to 0.0001. Here, the branching factor was chosen relatively large in order to produce a two-layered codeword tree providing results comparable to TopoART and SOINN. As can be seen in Fig. 3, the levels 1 and 2 of the lbTreeGNG network, denoted by lbTreeGNG 1 and lbTreeGNG 2 respectively, reasonably captured the topological structure of the input space. Since noise is a significant part of the input distribution, the lbTreeGNG system learnt codewords in those noisy regions as well. However, the trained GNG networks show a much higher resolution in relevant parts of the input distribution. Due to the hierarchical space partitioning of the network, a single GNG network in the second layer only encodes local topological structures within a single Voronoi cell of the first layer. In contrast to SOINN and TopoART, lbTreeGNG does not directly provide a labeling of clusters. Rather the labels are implicitly represented by the hierarchical taxonomy.

For SOINN, the values of λ, $\mathrm{age}_{\mathrm{dead}}$, and c were manually selected in such a way that results comparable to those published by Furao & Hasegawa (2006) were achieved. Although Furao & Hasegawa used a very similar dataset, it was not possi-

[2] According to the guidelines mentioned in Sect. 3.1, ϕ was increased until those neurons which represent noise were successfully removed based on their comparably low activation frequencies. Then, β_{sbm} was varied for fine tuning.

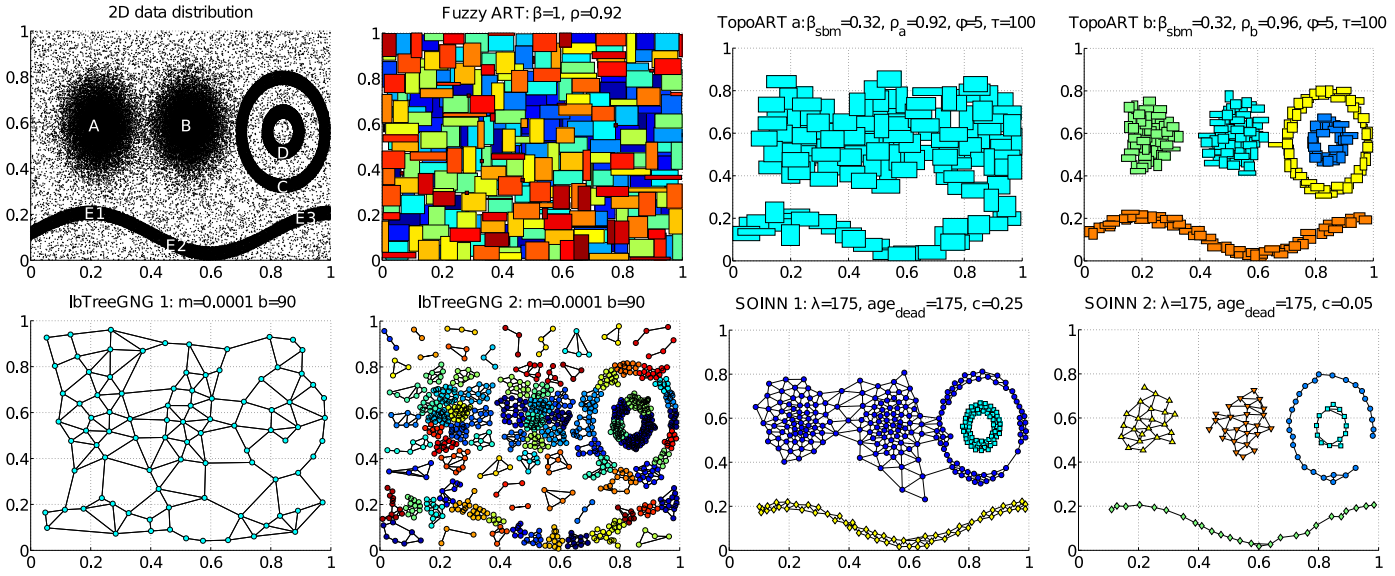

Figure 3: Data distribution and clustering results of several types of neural networks. Due to the noise contained in the data, Fuzzy ART covered virtually the complete input space with its rectangular categories. In contrast, TopoART learnt a noise-insensitive representation in which the categories were summarised to arbitrarily shaped clusters. The representation of TopoART *a* was refined by TopoART *b*. Here, all categories of an individual cluster are painted with the same colour. The first level of the lbTreeGNG network represents the input space globally. At the second level, the representations are locally refined and the topological structure is locally maintained. Noise regions are represented by the lbTreeGNG network as well. But the node density is much lower than in the relevant regions of the input space. Finally, the data distribution was successfully clustered by SOINN. Here, the representation is refined from SOINN 1 (first layer) to SOINN 2 (second layer). Reference vectors belonging to the same cluster share a common colour and symbol.

ble to obtain such results on our data using their parameter settings. In order to find more appropriate values, individual parameter settings for both layers (SOINN 1 and SOINN 2) were allowed. In contrast to λ, age_{dead}, and c, the values of α_1, α_2, α_3, β, and γ could directly be adopted from (Furao & Hasegawa, 2006) for both layers (1/6, 1/4, 1/4, 2/3, 3/4). Figure 3 shows that SOINN, was able to create a hierarchical representation of the input distribution: The three clusters of SOINN 1 were refined by SOINN 2 which distinguishes five clusters. Similar to TopoART *b*, SOINN 2 exhibits a reduced sensitivity to noise.

4.1.2. Non-stationary Artificial Data

In this experiment, we compared TopoART, lbTreeGNG, and SOINN regarding their ability to represent changing data distributions. In doing so, the respective networks were successively trained with all available samples from the subdistributions *A*+*E*3, *B*+*E*2, and *C*+*D*+*E*1 (cf. Fig. 3). Additionally, the subdistributions include 10% of uniformly distributed random noise. As in the previous experiment, the input samples were only presented once and learned immediately. Each row in Fig. 4 depicts snapshots of the different networks after training with the corresponding data. In order to account for the changing input data distribution, the parameters c (SOINN) and β_{sbm} (TopoART) were manually modified to achieve results comparable to those of Furao & Hasegawa (2006). The remaining parameter settings could be transferred from the previous experiment, which demonstrates some degree of insensitivity of the considered networks.

Figure 4 shows that the lbTreeGNG system created and maintained a reasonable codebook over time. Similar to the results for the stationary data, we can observe that the topological structure of the input space was locally preserved and that relevant regions were represented with a much higher resolution than noise regions. Since the data distribution changed over time, the node density was adapted accordingly. As a result, lbTreeGNG could learn novel or modified data distributions and already-represented structures may have been forgotten. This effect can be observed by comparing the different representations of *E*3 created by lbTreeGNG 2, for example. However, for deeper lbTreeGNGs the upper levels got more and more stable since the adaption rule of intermediate winning nodes allows gradually less plasticity in higher levels of the tree. In comparison to the results of the previous experiment, the size of the leaf GNGs in the regions *A* and *E*3 has decreased. The explanation for this is that the network tries to grow in breadth before it grows in depth. Thus, while learning *A*+*E*3 the network created a higher resolution in the first layer since it had capacity left. As a consequence, smaller leaf GNGs were produced at the second level.

As the second layer of SOINN can only be trained after the first layer has finished learning, only the first layer (SOINN 1) could be applied to learn the non-stationary data. The results resemble the results obtained in the previous experiment (cf. Fig. 3). But here, the respective clusters were incorporated subsequently, depending on the current data distribution. Learnt representations remained virtually stable and were only slightly modified due to noise.

Finally, Fig. 4 shows that both components of TopoART incrementally learnt the presented input. Similar to SOINN, already-created representations remained stable when the input distribution changed. As in the stationary case, TopoART *b* per-

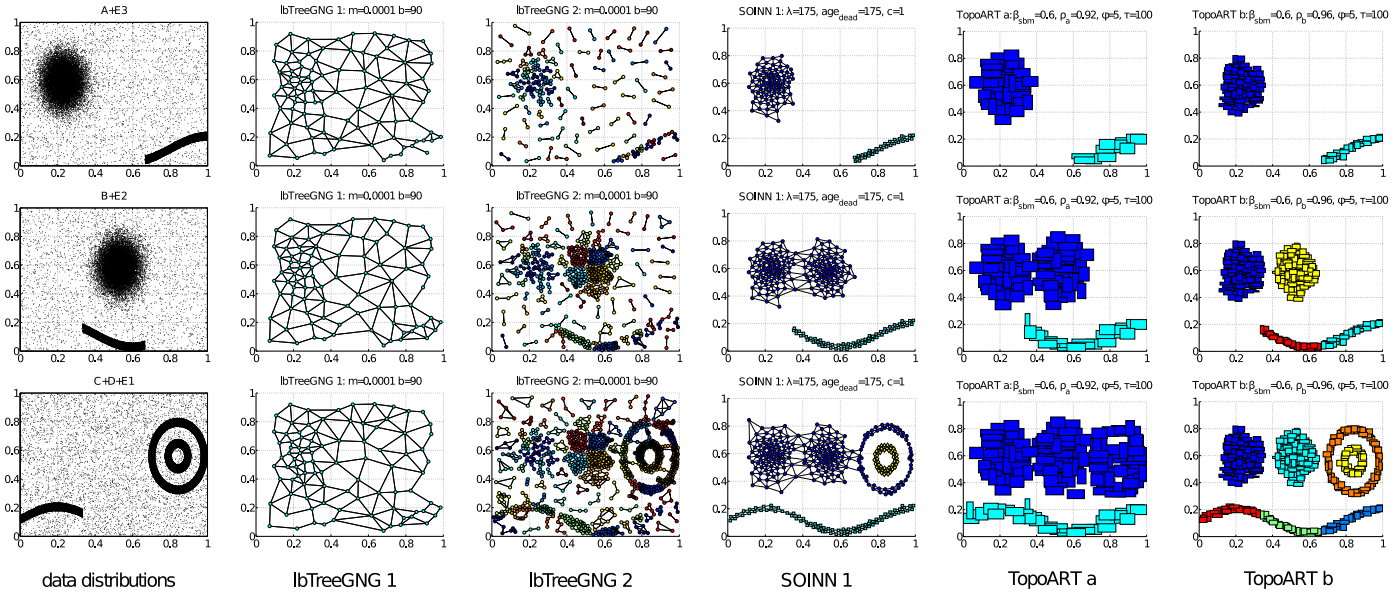

Figure 4: Training results for a changing data distribution. In order to simulate non-stationary data, the networks were successively trained with samples from the subdistributions A+$E3$, B+$E2$, and C+D+$E1$, which are depicted in the leftmost column. Each row shows the formed representations of all considered networks after finishing the respective training period. Here, each cluster of SOINN and of TopoART as well as each connected component of lbTreeGNG has been drawn using an individual colour. All networks were able to incrementally incorporate the new data. The representations created by SOINN and by TopoART are stable; i.e., learnt structures are not forgotten if the input distribution changes. In contrast, representations learnt by lbTreeGNG may be destroyed during the learning process, for instance, the representation of subdistribution $E3$ created by lbTreeGNG 2.

formed a refinement of the representation of TopoART a. But here, the sub-regions $E1$, $E2$, and $E3$ were separated, since the corresponding input samples were presented independently and could not be linked. TopoART a was able to compensate for this effect, as its lower vigilance parameter ρ_a allowed for larger categories which could form connections between the sub-regions.

4.1.3. Stationary Real-World Data

Finally, a dataset originally used to investigate methods for the direct imitation of human facial expressions by the user-interface robot iCat was applied (Tscherepanow et al., 2009). From this dataset, the images of all 32 subjects (12 female, 20 male) that were associated with predefined facial expressions were selected, which resulted in a total of 1783 images. These images had been acquired using two lighting conditions: daylight and artificial light. In order to reduce the dimensionality of the input data, the images were processed according to the procedure shown in Fig. 5, resulting in 45-dimensional feature vectors: The recorded face images were cropped, scaled to a size of 64×64 pixels, and successively processed by principal component analysis keeping 90% of the total variance.

After training SOINN and TopoART systems with the entire dataset until convergence of the permanent nodes' weights and edges (TopoART) or until a maximum number of iterations (SOINN)[3] was reached, the resulting clusters were compared to the partitionings based on labels reflecting the individual

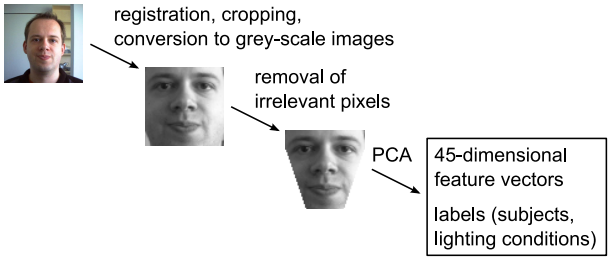

Figure 5: Processing of the facial images. First, the original images were converted to grey-scale images, aligned according to the position of the eyes, cropped, and scaled to a size of 64×64 pixels. Then, pixels lying in face regions which are irrelevant for the imitation of facial expression by the iCat were removed. The resulting images were further processed by principal component analysis keeping 90% of the total variance.

subjects and the two different lighting conditions. Here, two standard measures were used: the Jaccard coefficient J and the Rand index R (Xu & Wunsch II, 2009). Both provide values between 0 and 1, with higher values indicating a higher degree of similarity. As always more than one criterion for partitioning these images (subjects, lighting conditions, facial expressions, gender, usage of glasses, etc.) influences the clustering process, values considerably lower than 1 may also indicate similarity. The splitting of clusters caused by these additional criteria, for example, entails a decrease of the Jaccard coefficient. The results are shown in Fig. 6.

Here, the alternative activation (12) was used to analyse the clusters formed by TopoART to be independent from the category size. In order to analyse the networks' capability to create appropriate representations of the learnt data, the relevant par-

[3] Unlike TopoART, SOINN does not differ between permanent nodes and node candidates. Therefore, the adaptation of the network structure and the weights does not reach a stable state. In order to solve this problem, the maximum number of iterations was set to 10.

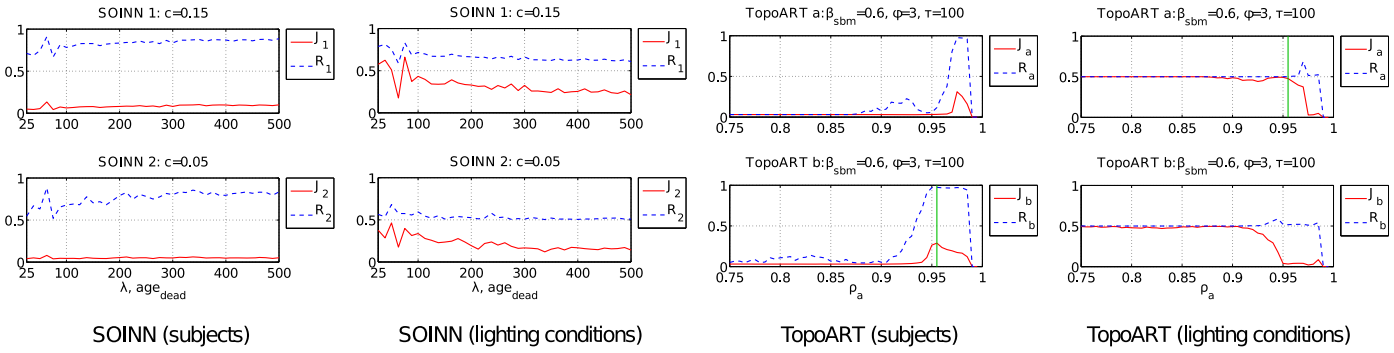

Figure 6: Clustering of stationary real-world data. The preprocessed facial images were clustered by a SOINN and a TopoART network. Then, how accurately the resulting clusters reflected the different subjects and the two lighting conditions was analysed. Here, the Rand index R and the Jaccard coefficient J were applied to measure the similarity. For both networks, the results were plotted depending on the parameters, which have the strongest influence on the cluster size – namely λ and age_{dead} (SOINN) as well as ρ_a (TopoART). The vertical green line in the TopoART graphs marks the value of ρ_a, which enables an optimal representation of both partitionings in a single network. In this case, TopoART a would represent the lighting conditions and TopoART b the subjects.

ameters have to be set accordingly. Due to the dimensionality of the input space, a manual choice is rather difficult. Therefore, we have decided to apply an automated procedure exploiting the available labels. For both networks, three parameters were optimised by means of grid search in order to maximise the Jaccard coefficient for the complete dataset. In particular, the TopoART parameters β_{sbm}, ϕ, and ρ_a as well as the SOINN parameters c (one for each layer) and λ were iterated over representative values from the relevant intervals[4] and selected in such a manner as to maximise the Jaccard coefficient for the partitioning according to the subjects. For SOINN, age_{dead} was set equal to λ. While each SOINN layer has its own parameter c, λ and age_{dead} were applied to the entire network, in order to simplify the optimisation process. The remaining parameters were transferred from the previous experiments.

Although the parameter optimisation was performed in such a way as to maximise the Jaccard coefficient for the partitioning according to the subjects, the maximum values reached by SOINN ($J_1^{\max}$=0.134, $J_2^{\max}$=0.080) are comparatively low. Moreover, the results of SOINN 2 appear as an impaired version of the results of SOINN 1. In contrast, TopoART achieved results which are more than twice as high ($J_a^{\max}$=0.312, $J_b^{\max}$=0.306). Regarding the Rand index, TopoART shows improvements as well. Therefore, TopoART is more suitable for representing this partitioning.

The results with respect to the partitioning according to the lighting conditions are equally high. Here, the results of SOINN 2 again resemble an impaired version of the results of SOINN 1. In contrast, the similarity of the clusterings provided by TopoART, strongly depends on the vigilance parameter ρ_a. But the results are not impaired from TopoART a to TopoART b. They are rather shifted along the ρ_a-axis, which reflects the different levels of detail represented by both components. As a consequence, the representations of both partitionings can be combined in a single TopoART network, provided that ρ_a is chosen appropriately, e.g., ρ_a=0.955 (see Fig. 6). In this case, TopoART a represents the coarser partitioning with respect to the lighting conditions and TopoART b the finer partitioning according to the subjects.

In order to solve clustering tasks, the procedure used for parameter optimisation within the scope of our analysis cannot be applied, as labeling information is usually not available here. It was only applied so as to show the principal capability of the networks to represent the respective partitionings. Considering the difficulty to select appropriate values for the network parameters, the application of TopoART is advantageous as well, as it only requires 4 parameters to be set in comparison to the 16 parameters of SOINN. Nevertheless, the parameters must be set using some kind of prior knowledge about the input data distribution (see Sect. 3.1). This task may be difficult, in particular, as TopoART is always trained on-line. In order to solve this problem, the hierarchical structure of TopoART can be exploited, since it provides alternative clusterings of the input data distribution. By means of interaction during the learning process, these clusterings could be evaluated with respect to the current task or other criteria.

4.2. *TopoART-AM*

TopoART-AM does not differ from TopoART during learning. Therefore, the results of the previous experiments can be directly transferred to TopoART-AM. The focus of this section is on the recall procedure, which constitutes the main difference to the original TopoART. Since we want to show that TopoART-AM can be applied to real-word tasks, the evaluation is based on two sets of colour images (RGB), which were recorded in real experimental setups. The first dataset consists of a subset of the facial images applied for analysing TopoART (cf. Sect. 4.1.3). The second dataset comprises images of different objects taken by a webcam (Logitech Webcam Pro 9000). This setup constitutes the first step in developing a new imitation learning approach for a humanoid robot. Both datasets are described in the following. In order to demonstrate the capability of TopoART-AM to serve as a hetero-associative memory, facial images from the first dataset were associated with images showing specific objects.

[4] $\beta_{sbm} \in [0,1]$, step size: 0.05; $\phi \in \{1,2,3,4,5\}$; $\rho_a \in [0.75, 0.995]$, step size: 0.005; $c \in [0.05, 1]$, step size: 0.05; $\lambda \in [25, 500]$, step size: 12.5

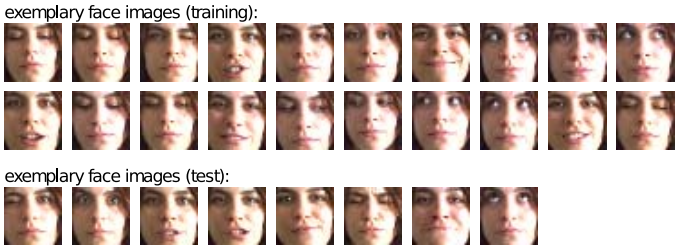

Figure 7: Facial images of an exemplary subject. For each subject, 28 images depicting different facial expressions were available. These 28 images were randomly divided into 20 training and eight test images.

The facial images were obtained by choosing 16 subjects (8 male, 8 female) from the dataset used to investigate methods for the direct imitation of human facial expressions by the user-interface robot iCat (Tscherepanow et al., 2009). From these subjects, those 28 images that show predefined facial expressions in daylight were selected. These images were aligned according to the position of the eyes, cropped, and scaled to a size of 32×32 pixels. A conversion to grey-scale images did not take place. The resulting images of one subject are shown in Fig. 7.

In contrast to other approaches (e.g., Chartier et al., 2009; Sudo et al., 2009; Yáñez-Márquez et al., 2007), the generalisation capability of which is demonstrated using learnt input samples disturbed by artificial noise, we decided to evaluate TopoART-AM using distinct test and training datasets. By means of this procedure, we want to provide an evaluation more suited to real-world applications. In order to reach this goal, 20 images of each subject were randomly selected for training, while the remaining eight images were reserved for the test set.

The object images were automatically recorded using the experimental setup introduced by Kammer et al. (2011). After an image of an object had been taken, a region of interest including the object was computed. Successively this region was rotated in such a way that the direction of its longest extension was aligned with the x-axis. Then it was cropped and scaled to a size of 45×23 pixels. This size was chosen in such a way that the areas covered by the object images and the face images are approximately equal. The automatic rotation results in the generation of images showing the respective object in two different orientations differing in a rotational angle of 180°. In order to compile a set of object images, 16 different objects that can usually be found in an office environment were used. From each of these objects, images were recorded at five different positions. Two of these positions resulted in one orientation and three in the other. Furthermore, as the camera is not vertically fixed and the positions of the placed objects vary, the automatic rotation can lead to deformed object appearances. Figure 8 shows the images taken of six exemplary objects.

The objective of TopoART-AM consisted in the learning of associations between persons and objects. Thus, $\underline{k}_1(t)$ and $\underline{k}_2(t)$ corresponded to serialised face and object images, respectively. For computational purposes, the elements of $\underline{k}_1(t)$ and $\underline{k}_2(t)$ were normed to values between 0 to 1. Each person was assigned to exactly one object. But both the persons and the objects were represented by multiple images. In particular, each

Figure 8: Images of six exemplary objects. Of each object, five images were taken. The varying orientation and partially deformation of the objects result from the technical conditions of the experimental setup.

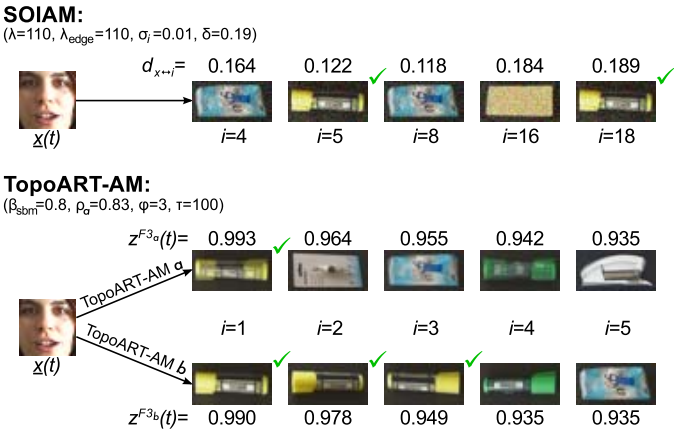

Figure 9: Recall results for an exemplary test image. The recalled images are shown depending on the iteration i in which they were generated. While SOIAM iterates over all nodes and returns an image if the distance $d_{x\leftrightarrow i}$ between the input $\underline{x}(t)$ and the weights of the respective neuron is smaller than the threshold δ, the recall results of both TopoART-AM components are ordered according to the activation $z^{F3}(t)$ of the corresponding $F3$ nodes. For SOIAM, the complete recall set is shown. In contrast, the recall of TopoART-AM was stopped after five iterations. The images of the object corresponding to the person the test image originates from are marked by a green tick.

face image of the training set was randomly associated with one of the five images of the corresponding object. As a result, the training set encompasses 320 associations. The test set consists of the remaining 128 face images. The object images corresponding to these face images were to be provided by the recall procedure. Figure 9 depicts exemplary recall results of a trained TopoART-AM system in comparison to a SOIAM network.

During recall, SOIAM determines the distance $d_{x\leftrightarrow i}$ between the input $\underline{x}(t)$ and the weights of all nodes independent of the cluster the nodes belong to. If $d_{x\leftrightarrow i}$ is smaller than the threshold δ, the second key $\underline{k}_2(t)$ is generated from the weights of the representative node of the respective cluster and returned. The images produced for an exemplary test image are shown in Fig. 9. They are ordered according to the iteration i in which they were returned. As a consequence of SOIAM's recall mechanism, multiple nodes of the same cluster can fulfil the condition $d_{x\leftrightarrow i}<\delta$ and generate the same image. Furthermore, the object images are returned in the order of the iterations and not according to $d_{x\leftrightarrow i}$.

In contrast to SOIAM, TopoART-AM generates only one

output for each cluster and its recall results are ordered according to the activation of the $F3$ nodes. Therefore, in the first iteration, the best-fitting recall is produced while with each further iteration the confidence in the recall result decreases. In principle, a threshold similar to the parameter δ of SOIAM could be applied in order to stop the recall process of TopoART-AM, if $z^{F3}(t)$ becomes too small. Both TopoART-AM components perform the recall independently of each other. As TopoART-AM b creates a more detailed representation of the input data than TopoART-AM a, here, the association of a person with an object can be split into different clusters which reflect the different orientations of the object.

In comparison to TopoART-AM, the images recalled by SOIAM appear considerably more noisy. This is a result of the training procedure of SOIAM, which introduces artificial noise sampled from a Gaussian distribution with zero mean and standard deviation σ_i.

In order to compare SOIAM and TopoART-AM more thoroughly, networks of both types were trained with the training set.[5] Afterwards, the recall error was computed for the test images. Due to the random assignment of images from an individual object to the face images of the corresponding subject, there exists no unique result. Rather, any image showing the correct object is appropriate. Therefore, we determined the recall error in comparison to the most similar image of the correct object as a measure for evaluating the recall procedures. This minimum recall error, which is denoted by $E^{\min}$, is computed as the mean absolute difference over all pixels and colour channels. As SOAIM returns the recall result in the form of an unordered set, $E^{\min}$ was additionally averaged over all generated images. The corresponding mean recall error is denoted by $\bar{E}^{\min}$. If SOIAM replied that the test key is unknown without providing any recalled image, $\bar{E}^{\min}$ was set to its maximum value 1. In contrast, using TopoART, the recall result of the first iteration can always be applied, as it constitutes the most accurate reply for the current test image.

The parameters of both networks were optimised using grid search in their most relevant intervals.[6] For each parameter setting, a network was trained and the recall error was averaged over all test images. The minimum recall error achieved by SOIAM amounts to $\bar{E}^{\min}$=0.019. The components of the TopoART-AM system reached errors of $E_a^{\min}$=0.001 and $E_b^{\min}$=0.002, respectively. Although the errors of both networks are very small, the recall error of SOIAM is about one order of magnitude higher in comparison to TopoART-AM.

Finally, we analysed how accurately the 16 underlying associations of a person to an object were reflected by the clustering structure. Similar to the experiment described in Sect. 4.1.3, we applied the Jaccard coefficient J as a similarity measure. The parameters of both networks were again iterated in their respective intervals. But now, they were chosen in such a way as to maximise the Jaccard coefficient for the SOIAM network and TopoART-AM a, respectively. In Fig. 10, the results depending on the parameters which have the highest influence on the cluster size are shown. In addition, the mean recall errors for the test set were plotted into this figure.

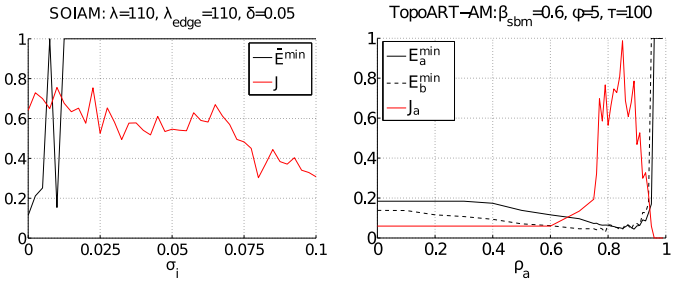

Figure 10: Analysis of the underlying clustering structure. Provided that ρ_a is chosen appropriately, TopoART-AM a reflects the 16 basic associations more accurately than SOIAM. Furthermore, the recall errors of TopoART-AM decrease to lower values. Since the clustering of TopoART-AM b exhibits a higher level of detail, $E_b^{\min}$ is smaller than $E_a^{\min}$ for large ranges of ρ_a.

Figure 10 demonstrates the advantages of using TopoART-AM for solving the task at hand. Besides causing smaller recall errors, it is more suited for reflecting the underlying clustering structure.

5. Conclusion

TopoART – the neural network presented in this article – successfully combines properties from ART and topology learning approaches: The categories originating from ART systems are connected by means of edges. In this way, clusters of arbitrary shapes can be formed. In addition, a filtering mechanism reduces the sensitivity to noise. Similar to SOINN, representations exhibiting different levels of detail are formed. But TopoART enables parallel learning at both levels requiring only 4 parameters (β_{sbm}, ϕ, ρ_a, τ) to be set, which constitutes a reduction of 75% compared to SOINN. Moreover, representations created by TopoART are completely stable.

The capability of TopoART to capture hierarchical relations and the topology of presented data might be of interest for numerous tasks, e.g., the representation of complex sensory and semantic information in robots. In principle, TopoART could even be extended to a multi-level structure that captures hierarchical relations more comprehensively.

By means of incorporating an appropriate recall procedure, TopoART can be extended to the hetero-associative memory TopoART-AM. TopoART-AM enables the learning of associations between complex data, such as different colour images. Nevertheless, it keeps the advantageous features of TopoART, namely its stability, robustness to noise and the ability of incrementally learning and representing different levels of detail. Therefore, it is preferable to alternative associative memory models, in particular, if real-world problems are to be solved.

[5] Similar to the training procedures described in Sect. 4.1.3, the training was stopped when the permanent nodes' weights and edges had converged (TopoART-AM) or a maximum number of iterations (SOIAM) had been reached. Here, the maximum number of iterations for the SOIAM network was increased to 50 in order to compensate for the smaller number of training samples.

[6] $\beta_{sbm}\in[0,1]$, step size: 0.05; $\phi\in\{1,2,3,4,5\}$; $\rho_a\in[0,0.99]$, step size: 0.1 (0 to 0.7), step size: 0.01 (0.75 to 0.99); τ=100; $\sigma_i, \delta\in[0,0.01]$, step size: 0.0025; $\lambda\in[10,500]$, step size: 10; $\lambda_{\text{edge}}=\lambda$

Acknowledgements

This work was partially funded by the German Research Foundation (DFG), Excellence Cluster 277 "Cognitive Interaction Technology".

References

Anagnostopoulos, G. C., & Georgiopoulos, M. (2000). Hypersphere ART and ARTMAP for unsupervised and supervised incremental learning. In *Proceedings of the International Joint Conference on Neural Networks, vol. 6* (pp. 59–64).

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001). Ellipsoid ART and ARTMAP for incremental clustering and classification. In *Proceedings of the International Joint Conference on Neural Networks, vol. 2* (pp. 1221–1226).

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759–771.

Chartier, S., Giguère, G., & Langlois, D. (2009). A new bidirectional heteroassociative memory encompassing correlational, competitive and topological properties. *Neural Networks*, *22*, 568–578.

Fritzke, B. (1994). A growing neural gas network learns topologies. In *Neural Information Processing Systems* (pp. 625–632).

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, *19*, 90–106.

Furao, S., Ogura, T., & Hasegawa, O. (2007). An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks*, *20*, 893–903.

Goerick, C., Schmüdderich, J., Bolder, B., Janßen, H., Gienger, M., Bendig, A., Heckmann, M., Rodemann, T., Brandl, H., Domont, X., & Mikhailova, I. (2009). Interactive online multimodal association for internal concept building in humanoids. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (pp. 411–418).

Goldstone, R. L., & Kersten, A. (2003). Concepts and categorization. In A. F. Healy, & R. W. Proctor (Eds.), *Handbook of psychology, Volume 4: Experimental psychology* (pp. 599–621). Hoboken, NJ, United States: Wiley.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, *11*, 23–63.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*, 2554–2558.

Ichiki, H., Hagiwara, M., & Nakagawa, M. (1993). Kohonen feature maps as a supervised learning machine. In *Proceedings of the IEEE International Conference on Neural Networks, vol. 3* (pp. 1944–1948).

Kammer, M., Tscherepanow, M., Schack, T., & Nagai, Y. (2011). A perceptual memory system for affordance learning in humanoid robots. In *Proceedings of the International Conference on Artificial Neural Networks*. In press.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kortkamp, M., & Wachsmuth, S. (2010). Continuous visual codebooks with a limited branching tree growing neural gas. In *Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354* (pp. 188–197). Berlin, Germany: Springer.

Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, *18*, 49–60.

Linde, Y., Buzo, A., & Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, *COM-28*, 84–95.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, vol. 1* (pp. 281–297).

Sudo, A., Sato, A., & Hasegawa, O. (2009). Associative memory for online learning in noisy environments using self-organizing incremental neural network. *IEEE Transactions on Neural Networks*, *20*, 964–972.

Tscherepanow, M. (2010). TopoART: A topology learning hierarchical ART network. In *Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354* (pp. 157–167). Berlin, Germany: Springer.

Tscherepanow, M., Hillebrand, M., Hegel, F., Wrede, B., & Kummert, F. (2009). Direct imitation of human facial expressions by a user-interface robot. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (pp. 154–160).

Tscherepanow, M., Jensen, N., & Kummert, F. (2008). An incremental approach to automated protein localisation. *BMC Bioinformatics*, *9*, 445.

Tscherepanow, M., & Kummert, F. (2007). Subcellular localisation of proteins in living cells using a genetic algorithm and an incremental neural network. In *Proceedings of the Workshop "Bildverarbeitung für die Medizin"* (pp. 11–15). Berlin, Germany: Springer.

Vigdor, B., & Lerner, B. (2006). Accurate and fast off and online fuzzy ARTMAP-based image classification with application to genetic abnormality diagnosis. *IEEE Transactions on Neural Networks*, *17*, 1288–1300.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Hoboken, NJ, United States: Wiley–IEEE Press.

Yáñez-Márquez, C., Cruz-Meza, M. E., Sánchez-Garfias, F. A., & López-Yáñez, I. (2007). Using alpha-beta associative memories to learn and recall RGB images. In *Proceedings of the International Symposium on Neural Networks, LNCS 4493* (pp. 828–833). Berlin, Germany: Springer.