Episodic Clustering of Data Streams Using a Topology-Learning Neural Network

Marko Tscherepanow[1,2] and **Sina Kühnel**[2,3] and **Sören Riechers**[1,2]

Abstract. In this paper, an extension of the unsupervised topology-learning TopoART neural network is presented. Like TopoART, it is capable of stable incremental on-line clustering of real-valued data. However, it incorporates temporal information in such a way that consecutive input vectors with a low distance in the input space are summarised to episode-like clusters. Inspired by natural memory systems, we propose two recall methods enabling the selection and retrieval of these episodes. They are demonstrated at the example of a video stream recorded in a natural environment.

1 Introduction

Incremental on-line learning is a branch of machine learning and artificial intelligence that has been gaining increasing interest over the recent years (e.g., [2], [7], [10], [14], and [20]). In contrast to traditional models requiring distinct training, validation, and test phases, such approaches allow for a continuous extension of existing knowledge while they are already in application. Hence, they are particularly useful for tasks involving incomplete knowledge or non-stationary data distributions such as the representation of visual [10] and multi-modal [2] categories in robotic scenarios or dynamic topic mining [12]. These methods process incoming data sample by sample; i.e., in a temporal order. However, this aspect is barely accounted for, although it might provide additional information.

Humans and animals exploit this information in a natural way. Consequently, sequence learning is considered as "the most prevalent form of human and animal learning" [16, p. 67]. This is reflected by the formed representations, which relates to the vast research area of memory.

Memory has been classified along time (short-term, long-term), processes (encoding, storage, retrieval) as well as regarding stored content (procedural, perceptual, semantic, episodic) [13]. Episodic memory is the highest developed system that allows us to remember our own past in detail. During encoding of episodic memory information from sensory systems, semantic knowledge as well as perceptual and procedural information is connected to one coherent event. This complex event knowledge allows us to perform mental time travel when remembering past experiences [21]. The importance of episodic memory becomes clearer when evolutionary traits and today's requirements are taken into account. When interacting in social situations we rely strongly on our ability to encode semantic as well as episodic memories of events. For example, established impressions of people and situations can be reevaluted and updated over time [11].

From the machine learning perspective, Sun and Gilles [16] distinguish between four major categories of sequence learning approaches: sequence prediction, sequence generation, sequence recognition, and sequential decision making. Some popular approaches are content-addressable memories for temporal patterns [3] (sequence generation), echo state networks [8] (sequence prediction and generation), hidden Markov models [15] (sequence prediction, generation, and recognition), as well as reinforcement learning [17] (sequential decision making). Unsupervised vector quantizers for time series such as Recursive Self-Organizing Maps (RSOMs) [22] constitute a further approach to sequence learning: they learn a mapping from subsequences to prototype sequences.

The approaches mentioned above deal with sequential data within a limited time frame. They do not have an absolute representation of time. Consequently, retrieval of past sequences as performed by natural memory systems is not possible. Furthermore, they are limited by a predefined model structure and capacity.

Machine learning approaches including those dedicated to sequence learning have been frequently employed so as to develop artificial memory systems. The CLARION architecture [6] possessing procedural and semantic memory components and the memory of the humanoid robot ISAC [9] comprising short-term and long-term memory components for processing procedural, perceptual, semantic, and episodic data are two examples for complex artificial memory systems. In addition, specific aspects of natural memory systems such as consolidation processes for procedural learning [1] and the categorisation of perceptual patterns [6] have been emulated.

In this paper, we present a novel approach to incremental on-line clustering (see Section 3) which incorporates temporal information for the life-long learning of episode-like clusters. In addition to the common prediction functionality, two recall methods for retrieving learnt information and reconstructing past episodes based on these clusters are proposed. As our approach originates from the TopoART neural network (see Section 2) [18][20], it inherits its capabilities of fast and stable on-line clustering of possibly noisy or non-stationary data. Therefore, we call our approach Episodic TopoART. In Section 4, we demonstrate the recall methods and show that the inclusion of temporal information may be advantageous given that input is provided in a meaningful temporal order.

2 TopoART

Adaptive Resonance Theory (ART) neural networks learn top-down expectations which are matched with bottom-up input. These expectations, which encode different regions of the input space, are called

[1] Applied Informatics, Faculty of Technology, Bielefeld University, Germany, email: marko@techfak.uni-bielefeld.de, marko@tscherepanow.de
[2] CITEC, Cognitive Interaction Technology, Center of Excellence, Bielefeld University, Germany
[3] Physiological Psychology, Faculty of Psychology and Sport Sciences, Bielefeld University, Germany

categories. Their maximum size is controlled by the vigilance parameter ρ.

TopoART (TA) [18][20] is an Adaptive Resonance Theory (ART) neural network consisting of two modules called TA a and TA b.[4] These modules are closely related to Fuzzy ART [4]. They have a three-layered structure and the input layer $F0$ is shared by them (see Fig, 1). Input to TA b is filtered by TA a, which renders the network insensitive to noise.

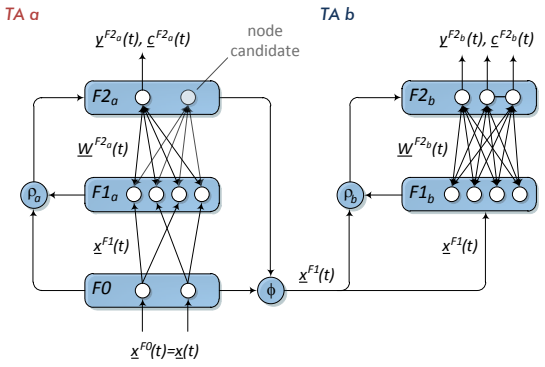

Figure 1. TopoART architecture. TopoART networks comprise two modules (TA a and TA b) sharing the input layer $F0$.

As TopoART is an incremental neural network that can be trained on-line, training and prediction steps can be mixed arbitrarily. In both cases, input is presented in discrete time steps t. Each input vector $\underline{x}(t)$ consists of d real-valued elements $x_i(t)$:

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

In the $F0$ layer, $\underline{x}(t)$ is complement coded. The resulting vector

$$\underline{x}^{F1}(t) = \big[x_1(t), \ldots, x_d(t), 1-x_1(t), \ldots, 1-x_d(t)\big]^T \tag{2}$$

is propagated to the respective $F1$ layer. Due to the usage of complement coding, each of the elements $x_i(t)$ has to lie in the interval $[0, 1]$.

2.1 Training

During training, $\underline{x}^{F1}(t)$ is first propagated to the $F2$ layer of TA a where the neurons (also called nodes) are activated (choice function):

$$z_j^{F2}(t) = \frac{\big\|\underline{x}^{F1}(t) \wedge \underline{w}_j^{F2,s}(t)\big\|_1}{\alpha + \big\|\underline{w}_j^{F2,s}(t)\big\|_1} \quad \text{with} \quad \alpha = 0.001. \tag{3}$$

The activation $z_j^{F2}(t)$ measures the similarity between $\underline{x}^{F1}(t)$ and the category of node j, which is encoded in the weight vector $\underline{w}_j^{F2,s}(t)$. $\wedge$ denotes an element-wise minimum operation.

In addition, a match value

$$\zeta_j^{F2,s}(t) = \frac{\big\|\underline{x}^{F1}(t) \wedge \underline{w}_j^{F2,s}(t)\big\|_1}{\big\|\underline{x}^{F1}(t)\big\|_1} \tag{4}$$

is computed for all $F2$ nodes j. It constitutes a measure for the size of the extended category that includes $\underline{x}^{F1}(t)$.

[4] In general, the number of modules must be larger than or equal to 1.

The maximum category size $S^{\max}$ depends on the dimensionality of the input space d and the vigilance parameter ρ:

$$S^{\max} = d(1-\rho). \tag{5}$$

Therefore, the weights $\underline{w}_j^{F2,s}(t)$ of an $F2$ node j are only allowed to be adapted if

$$\zeta_j^{F2,s} \geq \rho. \tag{6}$$

In order to learn a new input vector, the nodes with the highest and the second highest activation while fulfilling Eq. 6 (match function) are sought. They are referred to as the best-matching node (bm) and the second-best-matching node (sbm), respectively. Only the weights of these two neurons are adapted:

$$\begin{aligned} \underline{w}_j^{F2,s}(t+1) \quad = \quad & \beta_j\big(\underline{x}^{F1}(t) \wedge \underline{w}_j^{F2,s}(t)\big) \\ & +(1-\beta_j)\underline{w}_j^{F2,s}(t) \\ & \text{with } j \in \{bm, sbm\} \text{ and } \beta_{bm} = 1. \end{aligned} \tag{7}$$

In addition to its weight vector, each $F2$ node j possesses a counter n_j which is incremented whenever its weights are adapted. Furthermore, if two neurons bm and sbm that fulfil Eq. 6 were found, they are connected by an edge so as to learn the topological structure of the input data.

In order to reduce the sensitivity to noise, all $F2$ nodes with $n_j < \phi$ including their edges are removed every τ learning cycles.[5] Therefore, such nodes are called node candidates. If $n_j \geq \phi$, node j is permanent.

TA b learns in an identical way like TA a using a higher value for its vigilance parameter ρ_b:

$$\rho_b = \frac{1}{2}(\rho_a + 1). \tag{8}$$

In addition, $\underline{x}^{F1}(t)$ is only propagated to TA b if the best-matching node of TA a is permanent. As a consequence, TA b learns a refined clustering which is less prone to noise.

If the respective $F2$ layer does not contain any node yet or no node fulfilling Eq. 6 could be found, a new $F2$ node with $\underline{w}_{new}^{F2,s}(t+1) = \underline{x}^{F1}(t)$ is incorporated.

2.2 Prediction

During prediction steps, learnt cluster labels are associated to unknown input. After complement coding, presented input vectors are directly propagated to both modules. Here, the nodes of the respective $F2$ layer are activated using a modified activation function:

$$z_j^{F2}(t) = 1 - \frac{\Big\|\big(\underline{x}^{F1}(t) \wedge \underline{w}_j^{F2,s}(t)\big) - \underline{w}_j^{F2,s}(t)\Big\|_1}{d}. \tag{9}$$

In contrast to Eq. 3, Eq. 9 is independent of the category size.

After activation, the node with the highest activation is chosen as the best-matching node bm of the respective module. The match function is not checked. Then, both modules provide an output vector $\underline{y}^{F2}(t)$ with

$$y_j^{F2}(t) = \begin{cases} 0 & \text{if } j \neq bm \\ 1 & \text{if } j = bm \end{cases} \tag{10}$$

[5] The learning cycles are individually counted for each module.

and a clustering vector $\underline{c}^{F2}(t)$ containing the cluster labels of the $F2$ neurons. These cluster labels are determined by a labelling algorithm assigning unique integer labels to connected components of $F2$ nodes.

3 Episodic TopoART

Episodic TopoART (ETA) contains a TopoART network as a major learning component (see Fig. 2). This component is extended in order to enable the encoding and the retrieval of information within its spatio-temporal context.

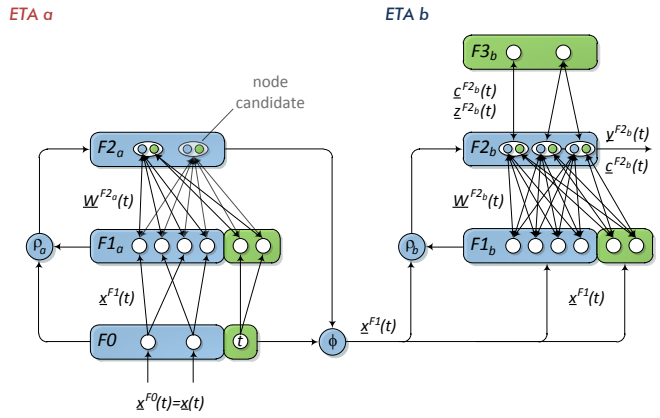

Figure 2. Structure of Episodic TopoART networks. Like TopoART, Episodic TopoART consists of two modules sharing a common input layer $F0$. The structures adopted from TopoART (blue) are extended by neurons representing temporal information and an additional layer required for recall (green).

Due to the structural similarities of TopoART and Episodic TopoART, both networks can easily be substituted by each other, e.g., in order to serve as components of more complex networks fulfilling alternative tasks such as the supervised TopoART-R network [19]. However, there are several functional differences between Episodic TopoART and TopoART, which are explained in the following. In addition to the training and prediction steps known from TopoART, Episodic TopoART provides a more complex recall functionality.

3.1 Training

The input of Episodic TopoART networks is equal to TopoART; i.e., individual input vectors $\underline{x}(t)$ comprise d elements $x_i(t)$. $\underline{x}(t)$ is complement coded and propagated to the respective $F1$ layer. Thus, Episodic TopoART is able to learn spatial relations of samples in the input space like TopoART and it requires a normalisation of all $x_i(t)$ into the interval $[0, 1]$.

In addition to the nodes representing the current input, the $F0$ layer of Episodic TopoART networks contains a single node representing the current time step $t{=}t^{F0}(t)$. It reflects the total number of performed training steps. Its actual value is not crucial as long as it is incremented by 1 after each training step. Therefore, it constitutes a subjective, internal representation of time.

The problem with clustering temporal data in conjunction with presented input consists in the different characteristics of this information. While the elements of the input vector $\underline{x}(t)$ are real-valued and normalised, $t^{F0}(t)$ is a positive integer value which is strictly increasing during learning and not bounded. Therefore, it is not possible to use complement coding for $t^{F0}(t)$. However, the effects of complement coding can be emulated. In particular, $\underline{x}^{F1}(t)$ corresponds to a category comprising only $\underline{x}(t)$ as a single point in the input space where $x_i(t)$ and $x_{i+d}(t)$ encode for the lower and upper bounds along dimension i, respectively. During learning, a category grows; i.e., it spans a certain range along different dimensions. Regarding $t^{F0}(t)$, a similar effect is achieved by the following encoding:

$$\underline{t}^{F1}(t) = \left[t_1^{F1}(t), t_2^{F1}(t)\right]^T. \tag{11}$$

Here, $t_1^{F1}(t)$ encodes the minimum time step and $t_2^{F1}(t)$ the maximum time step that is represented. For an individual sample, both values are equal to $t^{F0}(t)$.

Due to the different type of information processed, all $F2$ nodes j have two types of weights: the spatial weights $\underline{w}_j^{F2,s}(t)$ adopted from TopoART and the temporal weights

$$\underline{w}_j^{F2,t}(t) = \left[w_{j,1}^{F2,t}(t), w_{j,2}^{F2,t}(t)\right]^T. \tag{12}$$

Like in TopoART networks, the activation of the $F2$ nodes is computed according to Eq. 3; i.e., it reflects only spatial similarities. However, an additional temporal match value

$$\zeta_j^{F2,t}(t) = \frac{t^{max} - \min\left(t_2^{F1}(t) - w_{j,1}^{F2,t}(t), t^{max}\right)}{t^{max}} \tag{13}$$

is computed in order to incorporate temporal information in the learning process. The match values $\zeta_j^{F2,s}(t)$ and $\zeta_j^{F2,t}(t)$ are combined in a new match function:

$$\zeta_j^{F2,s} \geq \rho \quad \text{and} \quad \zeta_j^{F2,t} \geq \rho. \tag{14}$$

As a result, the $F2$ nodes represent spatial similarities which were encountered within a certain time frame bounded by t^{max}.

Using Eq. 14 for the processing of data streams causes a new problem. As explained in Section 2.1, edges are added between two nodes fulfilling the match function. However, if input is arriving as a data stream and temporal information is also considered, the overlap of categories is less probable, since new nodes are only added if no existing node can fulfil Eq. 14. Hence, the chance to find two nodes fulfilling the match function is considerably smaller. As a result, categories belonging to a cluster cannot be connected. Therefore, nodes need to be added earlier utilising a stricter match function for the determination of the best-matching nodes (cf. Eqs. 5 and 6):

$$\zeta_j^{F2,s} \geq \frac{1}{2}(\rho + 1) \quad \text{and} \quad \zeta_j^{F2,t} \geq \frac{1}{2}(\rho + 1). \tag{15}$$

If new input is to be learnt and the $F2$ nodes have been activated, the node with the highest activation while fulfilling Eq. 15 is determined. If such a node can be found, it becomes the best-matching node bm. Otherwise, a new node with $\underline{w}_{new}^{F2,s}(t){=}\underline{x}^{F1}(t)$ and $\underline{w}_{new}^{F2,t}(t){=}\underline{t}^{F1}(t)$ is added. This new node automatically fulfils Eq. 15 and, therefore, becomes the new best-matching node.

Afterwards, a second-best-matching node is sought. Here, Eq. 14 is applied as match function; i.e., the unmodified value of the respective vigilance parameter (ρ_a or ρ_b) is used. Hence, the categories can reach the same size in the input space as with the original TopoART. Furthermore, nodes rejected as best-matching nodes before can still become the second-best-matching node.

The spatial weights $\underline{w}_j^{F2,t}(t)$ and the temporal weight $w_{j,2}^{F2,t}(t)$ of the nodes bm and sbm are adapted according to Eq. 7. However, $w_{j,1}^{F2,t}(t)$ remains constant once a node has been created, as the time step t is strictly increasing and $w_{j,1}^{F2,t}(t)$ denotes the lower temporal bound of the respective category.

Like in the original algorithm, node candidates are removed every τ learning cycles, ETA b is trained in an identical way to ETA a using a vigilance value of ρ_b according Eq. 8, and input to ETA b is filtered by ETA a. As a result, ETA b learns a refined and noise-reduced clustering. Therefore, the output of ETA a is neglected for recall; the main function of this module consists in directing the attention of the network to relevant areas of the input space (cf. [19]).

3.2 Prediction of Cluster Labels

The prediction of cluster labels is performed in an identical way to TopoART (see Section 2.2). Temporal information is completely neglected and $t^{F0}(t)$ is not incremented. However, the formed clusters reflect the spatio-temporal relationships encountered during training; i.e., each cluster summarises similar samples which were learnt in close succession.

3.3 Recall of Spatio-Temporal Relationships

For recall, the formed clusters are interpreted as episodes, as they represent related input vectors (stimuli) in their temporal order. To recall information within the respective spatio-temporal context, Episodic TopoART distinguishes between two principal procedures: inter-episode recall and intra-episode recall. While inter-episode recall provides access to different episodes comprising stimuli similar to the presented input, intra-episode recall reconstructs episodes starting from a time step when a stimulus similar to the presented input vector was observed. Like the prediction mechanism, both procedures require that the $F2$ nodes of ETA b have activated according to Eq. 9 and labelled.

3.3.1 Inter-Episode Recall

The procedure for inter-episode recall is strongly related to the iterative recall procedure used by TopoART-AM [20] for recalling associations between real-world associative keys. However, TopoART-AM is not able to account for temporal relationships.

The actual recall mechanism is realised by the temporary $F3$ layer of ETA b. It is created after a stimulus has been presented. Each node of this layer represents an individual episode and is connected to all of its $F2$ nodes. The activation

$$z_l^{F3}(t) = \max_{j, c_j^{F2_b}(t)=l} z_j^{F2}(t) \tag{16}$$

of an $F3$ node l is equal to the maximum activation of the connected $F2$ nodes; i.e., it is a measure for the similarity of the presented stimulus with this episode. After the activation of the $F3$ nodes, the iterative recall process is initiated:

1. set iteration counter i to 1
2. find the $F3$ node r_i with the highest activation
3. inhibit all $F2$ nodes j with $z_j^{F2}(t) < z_{r_i}^{F3}(t)$ which are connected to r_i
4. find the $F2$ node bm_i with the highest activation within the current episode
5. return the output vector $\underline{y}(t,i)$ of the current iteration i
6. reset r_i $\left(z_{r_i}^{F3}(t) = -1\right)$
7. increment i
8. start next iteration (go to step 2)

The recall process either stops if all $F3$ nodes have been reset or a desired number of recall steps has been performed. Afterwards, the $F3$ layer is removed.

The output vector $\underline{y}(t,i)$ is computed as the centre of gravity of the respective best-matching category bm_i:

$$\underline{y}(t,i) = \frac{1}{2} \begin{bmatrix} w_{bm_i,1}^{F2,s}(t) & + & 1 - w_{bm_i,d+1}^{F2,s}(t) \\ \vdots & & \vdots \\ w_{bm_i,d}^{F2,s}(t) & + & 1 - w_{bm_i,2d}^{F2,s}(t) \end{bmatrix} \tag{17}$$

3.3.2 Intra-Episode Recall

Intra-episode recall requires an $F2$ node j as a starting point. For example, the best-matching nodes bm_i determined by means of inter-episode recall can be applied here.

After a suitable $F2$ node j has been chosen, the temporal order of all its topological neighbours n is analysed. Those nodes which were created after j, i.e. $w_{n,1}^{F2,t}(t) > w_{j,1}^{F2,t}(t)$, are put into the set $\mathcal{N}^+(j)$. Then, a best-matching node bm_i is computed as

$$bm_i = \arg \max_{n \in \mathcal{N}^+(j)} \left\| \underline{w}_j^{F2,t}(t) - \underline{w}_n^{F2,t}(t) \right\|_1. \tag{18}$$

Like with inter-episode recall, Eq. 17 is used, to generate an output for bm_i. Afterwards, bm_i is used as the starting node for the next intra-episode recall cycle. The recall process is stopped, if $\mathcal{N}^+(j) = \emptyset$. In this case, one possible end of the episode has been reached.

4 Results

We conducted two different experiments in order to analyse Episodic TopoART. First, we compared the prediction results of TopoART[6] and Episodic TopoART using a synthetic dataset (see Section 4.1). Then, we investigated its prediction and recall capabilities by means of real-world video data (see Section 4.2).

4.1 Synthetic Data

For the first experiment, we employed the well-known two spiral dataset (see Fig. 3a) [5]. It consists of two intertwined spirals comprising 97 points each. For validation, we randomly determined 250 additional samples for each spiral (see Fig. 3b). During training, both spirals were presented one after another. Furthermore, the samples of each spiral were presented with increasing radius. Thereby, both spirals can be considered as two consecutive episodes.

The clustering results for TopoART and Episodic TopoART are shown in Figs. 3c–f. The parameters ρ, β_{sbm}, and ϕ of both approaches were obtained by grid search using the validation dataset. Here, the Rand index R [23] for separating both spirals into two distinct clusters/episodes was maximised. Based on previous experiments (e.g., in [19]), τ was set to 200. As the new parameter t^{max} of Episodic TopoART denotes a time frame like τ, t^{max} was also set to 200. Each training sample was only presented once to each network.

[6] LibTopoART (version 0.37), available at `www.LibTopoART.eu`

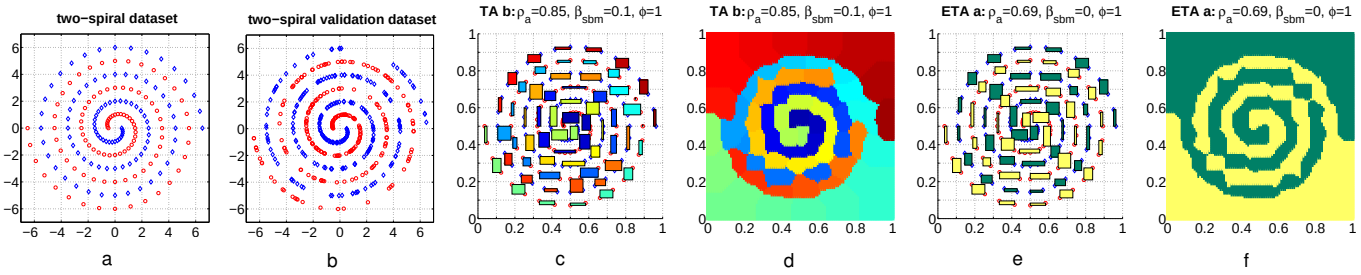

Figure 3. Clustering results for the two-spiral problem. The categories formed after training with the two-spiral dataset (a) are depicted as coloured rectangles (c and e). Here, categories connected to the same cluster share a common colour. In addition, the cluster labels were predicted for 101×101 equidistant test points distributed in the entire input space (d and f).

Figure 3 shows that both neural networks were able to learn the training samples after a single presentation. While Episodic TopoART correctly created two clusters corresponding to the two spirals (see Figs. 3e and 3f), TopoART created numerous clusters (see Figs. 3c and 3d), since its categories could not be linked appropriately. Furthermore, some categories enclose samples from both spirals. We therefore conclude that the inclusion of temporal information and the modified learning mechanism of Episodic TopoART supported the clustering process.

4.2 Real-World Data

In order to examine Episodic TopoART under more realistic conditions, we recorded a video stream by cycling with a mountain bike in the Teutoburg Forest. In comparison to indoor scenarios, outdoor environments are less structured and more diverse. In addition, they probably had a higher impact on human and animal evolution. The experimental setup and the generation of training data is explained in Fig. 4.

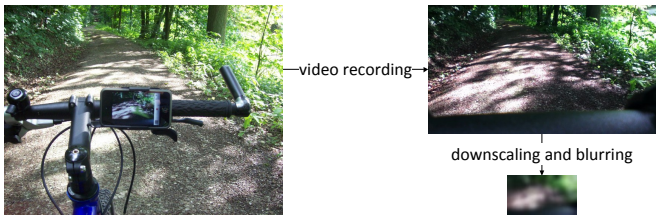

Figure 4. Experimental setup. An iPod touch 4G mounted on the handlebar of a mountain bike was used for recording image sequences in HD 720p with 30 frames per second. These images were downscaled to 64×36 pixels and subjected to a Gaussian blur (11×11 pixels). Images with even indices were used for training, while the remaining images were reserved for test purposes.

After preprocessing, a total of $29,747$ training and $29,747$ test images was available. Several Episodic TopoART networks were trained in a systematic way. However, to our knowledge there is no commonly accepted quality criterion for evaluating episodes. Rather, episodes formed based on the same event may even differ between different persons. Therefore, we resorted to manual evaluation. Figure 5 depicts the assignment of episodes[7] to video scenes computed by two different networks.

[7] prediction for the test images

Figure 5. Assignment of episodes. The video is represented by 100 test images taken at every 600th time step (left to right and top to bottom). The colour bars over each image visualise the assignment of episodes for two different networks (top bar: ρ_a=0.5; bottom bar: ρ_a=0.7; remaining parameters for both networks: β_{sbm}=0.25, ϕ=5, τ=200, t^{max}=400). Each episode is denoted by an individual colour.

Here, it needs to be emphasised that the episode length is not predefined. Rather, episodes are split if the input vectors differ considerably[8] for a longer[9] time interval. This is reflected by Fig. 5. While two episodes suffice to group the presented test images for ρ_a=0.5, the episodes are refined for ρ_a=0.7. In particular, Episodic TopoART formed a reasonable set of episodes with episode changes mainly caused by visible scene changes for ρ_a=0.7. A further increase of ρ_a would result in a higher number of created episodes. Hence, higher values of ρ_a result in a decline of the average episode length.

The more complex recall functionality of Episodic TopoART is demonstrated in Fig. 6. Here, the network trained with ρ_a=0.7 for the previous experiment (cf. Fig. 5) was used again.

Figure 6 demonstrates that the similarity of the test stimulus to the episodes provided by inter-episode recall decreases with each itera-

[8] defined by ρ_a, see Eqs. 4 and 14
[9] defined by ρ_a and t^{max}, see Eqs. 13 and 14

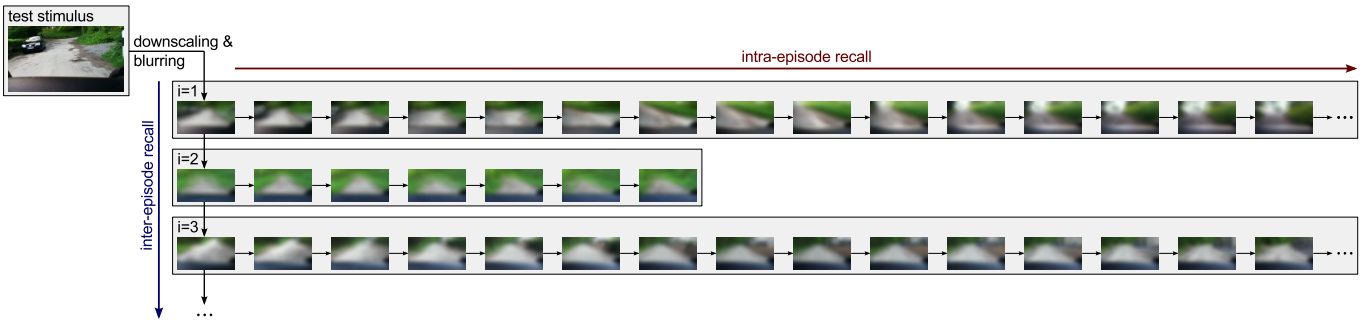

Figure 6. Recall functionality of Episodic TopoART. An exemplary test image was applied as a stimulus for initiating the inter-episode recall process. It originates from the last 10% of the video, between the second and the third image in the bottom line of Fig. 5. In each iteration i, the node bm_i was further used as the starting point for intra-episode recall. The recall results are limited to the first three iterations of the inter-episode recall algorithm and a maximum of 14 cycles for each intra-episode recall call.

tion. In this example, the best matching node bm_1 of the first iteration encodes the correct episode. The reconstruction of this episode by means of intra-episode recall shows how the input changed from the second image to the third image in the bottom line of Fig. 5.

5 Conclusion

We extended the TopoART neural network in such a way that it can create spatio-temporal representations of presented input vectors. In particular, input is grouped in episode-like clusters which can be accessed by two novel recall methods. Furthermore, the modified training procedure may be superior to TopoART provided that the input is presented in a meaningful temporal order. In the future, additional recall methods could be developed, in particular for intra-episode recall, as each episode is an undirected graph, which can be traversed in numerous ways. In addition, multi-modal data and semantic information could be applied in order to create episodes being more similar to their natural counterparts.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Heni Ben Amor, Shuhei Ikemoto, Takashi Minato, Bernhard Jung, and Hiroshi Ishiguro, ‘A neural framework for robot motor learning based on memory consolidation’, in *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms*, volume 4432 of *LNCS*, pp. 641–648. Springer, (2007).

[2] Elmar Berghöfer, Denis Schulze, Marko Tscherepanow, and Sven Wachsmuth, ‘ART-based fusion of multi-modal information for mobile robots’, in *Proceedings of the International Conference on Engineering Applications of Neural Networks*, volume 363 of *IFIP AICT*, pp. 1–10, Corfu, Greece, (2011). Springer.

[3] J. Buhmann and K. Schulten, ‘Noise-driven temporal association in neural networks.’, *Europhysics Letters*, **4**(10), 1205–1209, (1987).

[4] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen, ‘Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system’, *Neural Networks*, **4**, 759–771, (1991).

[5] Stephan K. Chalup and Lukasz Wiklendt, ‘Variations of the two-spiral task’, *Connection Science*, **19**(2), 183–199, (2007).

[6] Sylvain Chartier, Gyslain Giguère, and Dominic Langlois, ‘A new bidirectional heteroassociative memory encompassing correlational, competitive and topological properties’, *Neural Networks*, **22**(5–6), 568–578, (2009).

[7] Shen Furao and Osamu Hasegawa, ‘An incremental network for on-line unsupervised classification and topology learning’, *Neural Networks*, **19**, 90–106, (2006).

[8] Herbert Jaeger, ‘Adaptive nonlinear system identification with echo state networks’, in *Neural Information Processing Systems*, pp. 593–600. MIT Press, (2002).

[9] Kazuhiko Kawamura, Stephen M. Gordon, Palis Ratanaswasd, Erdem Erdemir, and Joseph F. Hall, ‘Implementation of cognitive control for a humanoid robot’, *International Journal of Humanoid Robotics*, **5**(4), 547–586, (2008).

[10] Stephan Kirstein and Heiko Wersing, ‘A biologically inspired approach for interactive learning of categories’, in *Proceedings of the International Conference on Development and Learning*, pp. 1–6. IEEE, (2011).

[11] Stanley B. Klein, Leda Cosmides, Cynthia E. Gangi, Betsy Jackson, John Tooby, and Kristi a. Costabile, ‘Evolution and Episodic Memory: An Analysis and Demonstration of a Social Function of Episodic Recollection’, *Social Cognition*, **27**(2), 283–319, (April 2009).

[12] Jean-Charles Lamirel, Ghada Safi, Navesh Priyankar, and Pascal Cuxac, ‘Mining research topics evolving over time using a diachronic multi-source approach’, in *International Conference on Data Mining Workshops (ICDMW)*, pp. 17–24. IEEE, (2010).

[13] Hans J Markowitsch and Angelica Staniloiu, ‘Amnesic disorders’, *The Lancet*, **6736**(11), 1–12, (April 2012).

[14] Yann Prudent and Abdellatif Ennaji, ‘An incremental growing neural gas learns topologies’, in *Proceedings of the International Joint Conference on Neural Networks*, volume 2, pp. 1211–1216. IEEE, (2005).

[15] Lawrence R. Rabiner, ‘A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition’, *Proceedings of the IEEE*, **77**(2), 257–267, (1989).

[16] Ron Sun and C. Lee Giles, ‘Sequence learning: From recognition and prediction to sequential decision making’, *IEEE Intelligent Systems*, **16**(4), 67–70, (2001).

[17] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning – An Introduction*, MIT Press, 4th edn., 2002.

[18] Marko Tscherepanow, ‘TopoART: A topology learning hierarchical ART network’, in *Proceedings of the International Conference on Artificial Neural Networks*, volume 6354 of *LNCS*, pp. 157–167. Springer, (2010).

[19] Marko Tscherepanow, ‘An extended TopoART network for the stable on-line learning of regression functions’, in *Proceedings of the International Conference on Neural Information Processing*, volume 7063 of *LNCS*, pp. 562–571. Springer, (2011).

[20] Marko Tscherepanow, Marko Kortkamp, and Marc Kammer, ‘A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data’, *Neural Networks*, **24**(8), 906–916, (2011).

[21] Endel Tulving, ‘Episodic memory: from mind to brain’, *Annual Review of Psychology*, **53**, 1–25, (2002).

[22] Thomas Voegtlin, ‘Recursive self-organizing maps’, *Neural Networks*, **15**(8–9), 979–991, (2002).

[23] Rui Xu and Donald C. Wunsch II, *Clustering*, Wiley–IEEE Press, 2009.