Neural Networks 96 (2017) 137–149

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Lifelong learning of human actions with deep neural network self-organization

German I. Parisi [a,*], Jun Tani [b], Cornelius Weber [a], Stefan Wermter [a]

[a] Knowledge Technology Institute, Department of Informatics, University of Hamburg, Germany
[b] Cognitive Neurorobotics Research Unit, Okinawa Institute of Science and Technology (OIST), Japan

ARTICLE INFO

Article history:
Received 1 March 2017
Received in revised form 23 August 2017
Accepted 1 September 2017
Available online 20 September 2017

Keywords:
Lifelong learning
Action recognition
Unsupervised deep learning
Self-organizing neural networks

ABSTRACT

Lifelong learning is fundamental in autonomous robotics for the acquisition and fine-tuning of knowledge through experience. However, conventional deep neural models for action recognition from videos do not account for lifelong learning but rather learn a batch of training data with a predefined number of action classes and samples. Thus, there is the need to develop learning systems with the ability to incrementally process available perceptual cues and to adapt their responses over time. We propose a self-organizing neural architecture for incrementally learning to classify human actions from video sequences. The architecture comprises growing self-organizing networks equipped with recurrent neurons for processing time-varying patterns. We use a set of hierarchically arranged recurrent networks for the unsupervised learning of action representations with increasingly large spatiotemporal receptive fields. Lifelong learning is achieved in terms of prediction-driven neural dynamics in which the growth and the adaptation of the recurrent networks are driven by their capability to reconstruct temporally ordered input sequences. Experimental results on a classification task using two action benchmark datasets show that our model is competitive with state-of-the-art methods for batch learning also when a significant number of sample labels are missing or corrupted during training sessions. Additional experiments show the ability of our model to adapt to non-stationary input avoiding catastrophic interference.

1. Introduction

The robust recognition of other people's actions represents a crucial component underlying social cognition. Neurophysiological studies have identified a specialized area for the visual coding of articulated motion in the mammalian brain (Perrett, Rolls, & Caan, 1982), comprising neurons selective to biological motion in terms of time-varying patterns of form and motion features with increasing complexity of representation (Giese & Rizzolatti, 2015). The hierarchical organization of the visual cortex has inspired computational models for action recognition from videos, with deep neural network architectures producing state-of-the-art results on a set of benchmark datasets (e.g. Baccouche, Mamalet, Wolf, Garcia, & Baskurt, 2011; Jain, Tompson, LeCun, & Bregler, 2015; Jung, Hwang, & Tani, 2015).

Typically, visual models using deep learning comprise a set of convolution and pooling layers trained in a hierarchical fashion for yielding action feature representations with an increasing degree of abstraction (Guo, Liu, Oerlemans, Lao, Wu, & Lew, 2016). This processing scheme is in agreement with neurophysiological studies supporting the presence of functional hierarchies with increasingly large spatial and temporal receptive fields along cortical pathways (Hasson, Yang, Vallines, Heeger, & Rubin, 2008; Taylor, Hobbs, Burroni, & Siegelmann, 2015). The training of deep learning models for action sequences has been proven to be computationally expensive and to require an adequately large number of training samples for the successful learning of spatiotemporal filters. The supervised training procedure comprises two stages: (i) a forward stage in which the input is represented by the current network parameters and the prediction error is used to compute the loss cost from ground-truth sample labels, and (ii) a backward stage which computes the gradients with respect to the parameters and updates them using back-propagation through time (BPTT, Mozer, 1995). Different regularization methods have been proposed to boost performance such as parameter sharing and dropout. However, the training process requires samples to be (correctly) labeled in terms of input–output pairs. Consequently, standard deep learning models for action recognition do not account for learning scenarios in which the number of training samples is not be sufficiently high and ground-truth labels may be occasionally missing or noisy.

* Correspondence to: Knowledge Technology Institute, Department of Informatics, University of Hamburg, Vogt-Koelln-Strasse 30, Hamburg 22527, Germany.
E-mail address: parisi@informatik.uni-hamburg.de (G.I. Parisi).

http://dx.doi.org/10.1016/j.neunet.2017.09.001

Furthermore, the above-described approaches have been designed for learning a batch of training actions, thus assuming a given training set. This training set should contain all necessary knowledge that can be readily used to predict novel samples. Consequently, such a training scheme is not suitable for more natural scenarios where an artificial agent should incrementally process a set of perceptual cues as these become available over time.

Lifelong learning is considered to be essential for cognitive development and plays a key role in autonomous robotics for the progressive acquisition of knowledge through experience and the development of meaningful internal representations during training sessions (Lee, 2012; Zhou, 1990). When dealing with non-stationary data, processing systems should account for the integration of new information while preserving existing knowledge (Grossberg, 1987). This problem is known as the *stability–plasticity dilemma* and has been well studied in both biological systems and computational models (see Ditzler, Roveri, Alippi, & Polikar, 2015 for a survey). In particular, the purpose of stability–plasticity theories is to avoid *catastrophic interference*, i.e., the process of new knowledge overwriting old knowledge. For error-driven connectionist models such as back-propagation networks, catastrophic interference can be addressed by the use of special processing structures (McClelland, McNaughton, & O'Reilly, 1995).

In this paper, we introduce a deep neural architecture for the lifelong learning of body actions. Our proposed architecture consists of a series of hierarchically arranged self-organizing neural networks for learning action representations from pose–motion features (Fig. 2). Each layer of the architecture comprises a novel variant of the Growing When Required (GWR) network (Marsland, Shapiro, & Nehmzow, 2002), the Gamma-GWR, which equips neurons in the network with recurrent processing with increasingly large spatiotemporal receptive fields. Each network is followed by a pooling mechanism for learning action features robust to translation and scale variance. Lifelong learning is achieved in terms of prediction-driven neural dynamics, accounting for the incremental learning of non-stationary input distribution. In the self-organizing hierarchy, the growth and adaptation of the Gamma-GWR networks are driven by their capability to predict neural activation sequences from the previous network layer. The development of associative connections between visual representations and symbolic labels allows learning action-label mappings over time, without requiring a predefined number of action classes and with tolerance to missing and corrupted sample labels.

The architecture is based on three assumptions consistent with neurophysiological evidence from the mammalian visual system: (i) complex motion is analyzed in parallel by two separate pathways and subsequently integrated to provide a joint percept (Vangeneugden, Pollick, & Vogels, 2009); (ii) both pathways contain hierarchies to extrapolate shape and optic-flow features with increasing complexity, from low- to high-level representations of the visual stimuli (Giese & Poggio, 2003; Hasson et al., 2008); and (iii) input-driven self-organization is crucial for the cortex to tune the neurons according to the distribution of the inputs (Nelson, 2000; Willshaw & von der Malsburg, 1976). In previous research, we showed that a deep architecture comprising a hierarchy of self-organizing networks can learn spatiotemporal action features with increasing complexity of representation (Parisi, Weber, & Wermter, 2015). The main advantage of this method over traditional supervised learning approaches is that visual representations are learned in an unsupervised fashion. The incremental development of associative connections between visual representations and symbolic labels yield the correct formation of action—label mappings also in the absence of a large percentage of sample labels (Parisi, Tani, Weber, & Wermter, 2016). However, the experiments for these studies were conducted using hand-crafted action features, thus against the idea of deep neural architectures that learn significant features by iteratively tuning internal representations. Furthermore, the temporal processing of pre-processed features was modeled in terms of neurons in higher-level layers computing the concatenation of neural activation trajectories from lower-level layers, which increases the dimensionality of neural weights along the hierarchy. In this paper, we address these two issues through the development of a self-organizing hierarchy with increasingly large spatiotemporal receptive fields.

We report a series of experiments with the Weizmann (Gorelick, Blank, Shechtman, Irani, & Basri, 2005) and the KTH (Schuldt, Laptev, & Caputo, 2004) action benchmark datasets showing that our model: (i) achieves competitive classification results with respect to the state of the art for batch learning, (ii) learns robust action-label mappings also in the case of occasionally missing or corrupted class labels during training sessions, and (iii) accounts for lifelong learning in terms of adapting to non-stationary input avoiding catastrophic interference.

2. Related work

Experience-driven development plays a crucial role in the brain (Nelson, 2000), with topographic maps being a common feature of the cortex for processing sensory inputs (Willshaw & von der Malsburg, 1976). We focus on the use of recurrent neural self-organization motivated by the process of input-driven self-organization exhibited by cortical maps and the computational efficiency of recurrent self-organizing neural networks. However, self-organizing methods are the result of simplified modeling assumptions with respect to biological findings. Furthermore, numerous approaches have been proposed in the literature that also effectively account for spatiotemporal processing. We point the reader to Section 5.2 for a discussion on additional approaches with different processing principles and levels of biological plausibility.

2.1. Computational models of neural self-organization

A number of studies suggested that visual input is crucial for normal cortical organization, since the cortex tunes itself to the input distribution (Blakemore & Cooper, 1970; Blakemore & Van Sluyters, 1975; Hirsch & Spinelli, 1970; Sengpiel & Stawinski, 1999). Computational models implementing experience-driven self-organization have been used to demonstrate that the preferences of neural response can result from statistical learning with the nonlinear approximation of the distribution of visual inputs. The goal of self-organizing learning is to cause different parts of the network to respond similarly to certain input samples starting from an initially unorganized state. Typically, during the training phase, these networks build a map through a competitive process so that a set of neurons represents prototype vectors encoding a submanifold in the input space. In doing so, the network learns significant *topological relations* of the input without supervision.

Different models of neural self-organization have been proposed to resemble the dynamics of basic biological findings on Hebbian-like learning and map plasticity. The most well-established model is the self-organizing feature map (SOM, Kohonen, 1990) algorithm that nonlinearly projects a high-dimensional input space onto a low-dimensional (typically two-dimensional) discretized representation. It consists of a layer with competitive neurons connected to adjacent neurons by a neighborhood relation. The SOM represents the input distribution using a finite set of prototype neurons, where the number of neurons must be decided before the training phase starts, and the topology of the network (neighborhood relation) is fixed. The network learns by iteratively reading each vector-valued training sample and organizes the neurons so that they describe the domain space of the input.

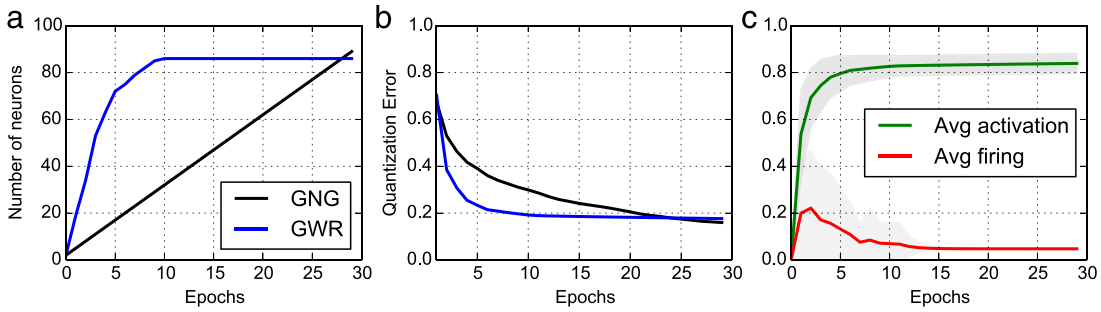

Fig. 1. Comparison of GNG and GWR growth behavior: (a) number of neurons, (b) quantization error, and (c) GWR average activation and firing counter (solid lines) and standard deviation (shadowed area) through 30 training epochs for the Iris dataset (150 four-dimensional samples).

The ability of a network to create new neurons (and remove unused ones) for adapting to novel incoming signals is crucial for better dealing with non-stationary input distributions. A well-known model is the growing neural gas (GNG, Fritzke, 1995), in which neurons are added at fixed intervals to minimize local errors. In contrast to the GNG, the growing when required (GWR) network proposed by Marsland et al. (2002) creates new neurons whenever the activity of trained neurons is smaller than a given threshold. As a criterion for neural growth, the training algorithm considers the amount of network activation at time t computed as a function of the distance (typically the Euclidean distance) between the current input $\mathbf{x}(t)$ and its best-matching neuron $\mathbf{w}_b$:

$$a(t) = \exp(-\|\mathbf{x}(t) - \mathbf{w}_b\|). \tag{1}$$

Additionally, the algorithm considers the number of times that neurons have fired so that recently created neurons are properly trained before creating new ones. The network implements a firing counter $\eta \in [0, 1]$ used to express how frequently a neuron has fired based on a simplified model of how the efficacy of an habituating synapse reduces over time (Stanley, 1976). The GWR algorithm will then iterate over the training set until a given stop criterion is met, e.g. a maximum network size or a maximum number of iterations.

The use of an activation threshold and firing counters to modulate the growth of the network leads to the creation of a larger number of neurons at early stages of the training and then tune the weights of existing neurons through subsequent training epochs. This behavior is particularly convenient for incremental learning scenarios since neurons will be created to promptly distribute their receptive fields in the input space, thereby yielding fast convergence through iterative fine-tuning of the topological map. It has been shown that GWR-based learning is particularly suitable for novelty detection and cumulative learning in robot scenarios (Marsland, Nehmzow, & Shapiro, 2005; Marsland et al., 2002). A comparison between GNG and GWR learning in terms of the number of neurons, quantization error (average discrepancy between the input and representative neurons in the network), and parameters modulating network growth (average network activation and firing rate with standard deviation) is shown in Fig. 1 over 30 training epochs for the Iris dataset[1] which contains 150 four-dimensional samples.

Self-organizing networks have shown state-of-the-art results for classification tasks with different techniques proposed to attach symbolic labels to prototype neurons (e.g. Beyer & Cimiano, 2011; Parisi et al., 2015). While the average discrepancy between the input and its representation in the network should decrease for a larger number of prototype neurons, there is not such a straightforward relation between the number of neurons and the classification performance. In fact, networks with a small number of neurons could perform better than bigger ones depending on input distribution properties such as the linear separability of input classes (Parisi et al., 2016). This is because the classification process consists of predicting the label of novel samples by retrieving attached labels to the input's best-matching neurons, with the actual distance between the novel input and the selected neurons being irrelevant for this task.

The standard GNG and GWR learning algorithms do not account for temporal sequence processing. Consequently, these models were extended with recurrent connectivity while preserving desirable learning properties such as computational efficiency and learning convergence.

2.2. Recurrent self-organizing networks

A number of temporal extensions of self-organizing networks has been proposed that implement recurrent connectivity so that neural activation in the map is driven by multiple time steps. The first example was the Temporal Kohonen Map (TKM, Chappell & Taylor, 1993) where the distance of a recurrent neuron $\mathbf{w}_i$ from the input sequence $(\mathbf{x}(t-n), \ldots, \mathbf{x}(t))$ with similarity measure d_W (e.g. the Euclidean distance) is computed as

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{x}(t), \mathbf{w}_i) + (1 - \alpha) \cdot \tilde{d}_i(t-1), \tag{2}$$

where $\alpha \in (0; 1)$ modulates the signal decay. However, in the TKM, there is no explicit back-reference to previous map activity because the context is only implicitly represented by the weights. Therefore, the sequence representation domains are restricted to the superposition of values.

Different approaches using context learning have been proposed that use a compact reference representation for an arbitrary lattice topology. Context learning as proposed by Strickert and Hammer (2005) combines a compact back-reference with a weighted contribution of the current input and the past. Each neuron is equipped with a weight vector $\mathbf{w}_i$ and a temporal context $\mathbf{c}_i$ (with $\mathbf{w}_i, \mathbf{c}_i \in \mathbb{R}^d$), the latter representing the activation of the entire map at the previous time step. The recursive activation function of a sequence is given by the linear combination

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{x}(t), \mathbf{w}_i) + (1 - \alpha) \cdot d_W(\mathbf{C}(t), \mathbf{c}_i), \tag{3}$$

$$\mathbf{C}_i = \beta \cdot \mathbf{w}_{I(t-1)} + (1 - \beta) \cdot \mathbf{c}_{I(t-1)}, \tag{4}$$

where d_W is the distance in the map grid, $\alpha, \beta \in (0, 1)$ are fixed parameters, $\mathbf{C}(t)$ is a global context vector, and $I(t-1)$ denotes the index of the winner neuron at time $t-1$. Thus, neurons in the map of models using context learning encode temporal input assuming an exponential decay of the influence of the past. The training is

[1] http://archive.ics.uci.edu/ml/datasets/Iris.

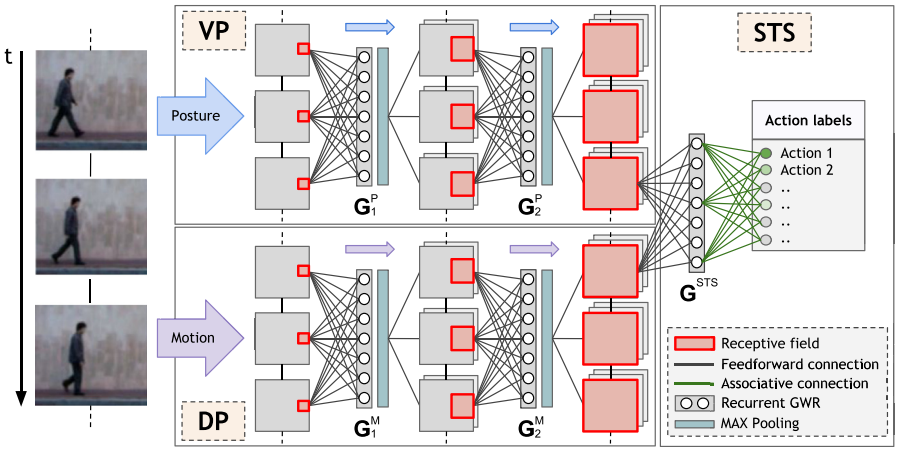

Fig. 2. Diagram of our deep neural architecture with recurrent GWR networks for action recognition. Posture and motion action cues are processed separately in the ventral (VP) and the dorsal pathway (DP), respectively. Each network consists of a recurrent competitive network. At the STS stage, the recurrent GWR network learns associative connections between prototype action representations and symbolic labels.

carried out by adapting the weight and the context vector towards the current input and context descriptor according to

$$\triangle \mathbf{w}_i = \epsilon_i \cdot h_\sigma(d_N(i, I_t)) \cdot (\mathbf{x}(t) - \mathbf{w}_i), \tag{5}$$

$$\triangle \mathbf{c}_i = \epsilon_i \cdot h_\sigma(d_N(i, I_t)) \cdot (\mathbf{C}(t) - \mathbf{c}_i), \tag{6}$$

where ϵ_i is the learning rate, h_σ is usually a Gaussian function, and $d_N : N \times N \rightarrow \mathbb{R}$ is a neighborhood function that defines the topology of the network.

Context learning can be applied to lattices with an arbitrary topology as well as to incremental approaches that vary the number of neurons over time. For instance, a GNG model equipped with context learning (MergeGNG, Andreakis, von Hoyningen-Huene, and Beetz, 2009) uses the activation function defined in Eq. (3) to compute winner neurons and creates new neurons with a temporal context. In Parisi, Magg, and Wermter (2016), we presented a GWR network equipped with recurrent neurons with one context descriptor, yielding a decreased temporal quantization error for a time-series prediction task with respect to recurrent models of the SOM and the GNG. This formulation of context learning can be extended to equip each neuron with an arbitrary number of context descriptors, thus increasing the temporal receptive field of neurons in the map, and leading to a reduced temporal quantization error. This is due to an increase in memory depth and temporal resolution following the idea of a Gamma memory model (de Vries & Príncipe, 1992). Experiments using SOM and GNG networks with Gamma memory showed a reduced temporal quantization error with respect to traditional context learning models for trained networks using multiple context descriptors in nonlinear time series analysis (Estévez & Hernández, 2011; Estévez & Vergara, 2012).

2.3. Lifelong learning

The purpose of stability–plasticity theories is to understand how to avoid *catastrophic interference* so that the acquisition of new information does not result in forgetting old knowledge. Specifically for self-organizing feature maps, Richardson and Thomas (2008) investigated three factors that modulate the effects of catastrophic interference for non-stationary inputs: (1) the conditions of map plasticity, (2) the available resources to represent information, and (3) the similarity between new and old knowledge. The functional plasticity of the map can be modulated through *critical periods*, i.e., a particularly sensitive period in which the map is more responsive to the effects of experience. In fact, a body of literature has suggested that early experiences are particularly influential on human and animal behavior (see e.g. Hubel and Wiesel, 1962, Johnson and Newport, 1991; Senghas, Kita, and Ozyurek, 2004). In the SOM (Kohonen, 1990), two training phases are used to achieve a topographic organization resembling the two phases of a critical period: the organization phase in which the network is trained with a high learning rate and large neighborhood size, and the tuning phase in which these parameters are reduced to fine-tune the map and learn more detailed distinctions of the input. Since the number of neurons in the SOM (i.e., the resources available for storing information) is fixed, critical periods play a crucial role in the formation of maps with good topological organization.

In growing self-organizing networks such as the GNG (Fritzke, 1995) and the GWR (Marsland et al., 2002), available resources to allocate new information can be created in terms of new neurons and network connectivity driven by the input distribution. In this case, the learning parameters are kept fixed and map topology develops through competitive Hebbian learning (Martinetz, 1993). Thus, growing models do not directly implement critical periods but they rather incrementally adapt to non-stationary input. A mechanism used to control the durability of information in a growing network is the connection age. When a neuron is fired, the age of the connections from the neuron to its neighbors is set to 0, while the age of the rest of the connections is increased by a value of 1. At each iteration, connections of large age and neurons without connections are deleted. Removing a neuron from the network means that the information learned by that unit is permanently forgotten. Therefore, a convenient maximum age of connections μ_{max} must be set so that the network removes neurons that no longer fire while avoiding catastrophic forgetting.

From the perspective of hierarchical processing, predictive models with interactions between top-down predictions and bottom-up regression may provide a computational mechanism to account for the learning of dynamic input distributions in an unsupervised fashion (Jung et al., 2015). Predictive coding (Huang & Rao, 2011; Rao & Ballard, 1999) has been widely studied for understanding many aspects of brain organization. In particular, it has been proposed that the visual cortex can be modeled as a hierarchical network with reciprocal connections where top-down feedback connections from higher-order cortical areas convey predictions of lower-order neural activity and bottom-up connections carry the residual prediction errors. Tani (2003) and Tani and Nolfi (1999) proposed that sensory-motor patterns for on-line planning in a robot learning scenario can be generated and recognized by using

recurrent neural network models trained through prediction error minimization. However, neural network models that implement a predictive learning scheme to achieve lifelong learning have not yet been fully investigated.

3. Proposed method

3.1. Hierarchical neural self-organization

Our architecture consists of a series of hierarchically arranged self-organizing neural networks for processing body actions from pose–motion features (Fig. 2). Each layer in the hierarchy comprises a recurrent variant of the Growing When Required (GWR) network (Marsland et al., 2002), the Gamma-GWR, and a pooling mechanism for learning action features with increasingly large spatiotemporal receptive fields. In the last layer, neural activation patterns from distinct pathways are integrated. For the purpose of action classification, we propose an associative Gamma-GWR (AG-GWR) to develop connections between visual representations of pose–motion patterns and symbolic labels.

Hierarchies provide a convenient trade-off in terms of invariance-selectivity by decomposing a complex task in a hierarchy of simpler ones (Poggio & Smale, 2003). From a computational perspective, a hierarchical structure has the advantage of increased computational efficiency by sharing functionalities across multiple levels, e.g., low-level networks represent a dictionary of features that can be shared across multiple tasks. A hierarchical organization is consistent with neurophysiological evidence for increasingly large spatiotemporal receptive windows in the human cortex (Hasson et al., 2008; Lerner, Honey, Silbert, & Hasson, 2011; Taylor et al., 2015), where simple features manifest in low-level layers closest to sensory inputs while increasingly complex representations emerge in deeper layers.

The proposed deep architecture is composed of two distinct processing streams for pose and motion features, and their subsequent integration in the superior temporal sulcus (STS). This is in line with the biological findings suggesting that the mammalian brain processes body motion in two neural pathways: the ventral and the dorsal pathways (Felleman & Van Essen, 1991; Ungerleider & Mishkin, 1982). The ventral pathway recognizes sequences of snapshots of body posture, while the dorsal pathway recognizes movements in terms of optic-flow patterns. Both pathways comprise hierarchies that extrapolate visual features with increasing complexity of representation. Tan, Singer, Serre, Sheinberg, and Poggio (2013) have shown that the modeling of STS neurons as the simple linear weighted sum of inputs over a short temporal scale has produced good fits to data of STS neural responses in macaque monkeys for body action perception tasks. From a computational perspective, it has been shown that the separate processing of body pose and motion features improves the accuracy of action recognition (Gorelick et al., 2005; Parisi et al., 2016, 2015).

In the following sections, we introduce the proposed Gamma-GWR (Section 3.2), the neural mechanisms driving lifelong learning (Section 3.3), the pooling mechanism for yielding visual representations invariant to scale and position (Section 3.4), and the associative learning process for attaching sample labels to visual representations (Section 3.5). The details on the type of network input as well as the parameters for hierarchical learning are discussed in Section 4.

3.2. Gamma-GWR network

We introduce a temporal GWR network, the Gamma-GWR, that equips each neuron with an arbitrary number of context descriptors to increase the memory depth and temporal resolution in the spirit of a Gamma memory model (de Vries & Príncipe, 1992). In this paper, we modify the GWR model's activation function and learning rules to account for spatiotemporal processing using Gamma filtering. The proposed training algorithm is illustrated by Algorithm 1 (except for Steps 3, 9.c, and 10.b that are implemented by the AG-GWR only).

We equip the GWR network with recurrent neurons following previous formulations of context learning (Strickert & Hammer, 2005), so that the best-matching neuron $\mathbf{w}_b$ for the input $\mathbf{x}(t)$ is computed as follows:

$$b = \arg\min_i \{d_i\}, \tag{7}$$

$$d_i = \alpha_w \cdot \|\mathbf{x}(t) - \mathbf{w}_i\|^2 + \sum_{k=1}^{K} \alpha_k \cdot \|\mathbf{C}_k(t) - \mathbf{c}_i^k\|^2 , \tag{8}$$

$$\mathbf{C}_k(t) = \beta \cdot \mathbf{w}_{b(t-1)} + (1-\beta) \cdot \mathbf{c}^{k-1}_{b(t-1)}, \tag{9}$$

for each $k = 1, \ldots, K$, where $\alpha, \beta \in (0; 1)$ are constant values that modulate the influence of the current input and the past activations, $b(t-1)$ is the index of the best-matching neuron at $t-1$, and $\mathbf{c}^0_{b(t-1)} \equiv \mathbf{w}_{b(t-1)}$. This formulation of context learning combines a compact back-reference to the previous best-matching neuron with a separately controllable contribution of the current input and the past with an arbitrary network topology. It is not necessary to have a batch of training samples to initialize the network. Instead, neurons are incrementally created and tuned over time. Thus, we initialize our context descriptors $\mathbf{c}^k_{b(0)} = 0$ (Algorithm 1, Step 1). For $K = 1$, i.e. only one context descriptor, the Gamma-GWR is reduced to the temporal GWR as introduced by Parisi et al. (2016). Strickert and Hammer (2005) showed that a SOM with context learning converges to an efficient fractal encoding of given sequences with high temporal quantization accuracy.

A decay function with decreasing values of the parameter α_i gradually leaks an exponentially smaller amount of input over time. Since the definition of the context descriptors is recursive, setting $\alpha_w > \alpha_1 > \cdots > \alpha_{K-1} > \alpha_K > 0$ has been shown to reduce the propagation of errors from early filter stages to higher-order contexts for the Gamma-SOM (Estévez & Hernández, 2011) and the Gamma-GNG (Estévez & Vergara, 2012). (For our proposed definition of a decay function, see Section 4.1).

The training is carried out by adapting the weight and the context vectors of the best-matching neurons and its neighbors (Algorithm 1, Step 10.a) according to

$$\Delta\mathbf{w}_i = \epsilon_i \cdot \eta_i \cdot (\mathbf{x}(t) - \mathbf{w}_i), \tag{10}$$

$$\Delta\mathbf{c}_i^k = \epsilon_i \cdot \eta_i \cdot (\mathbf{C}_k(t) - \mathbf{c}_i^k), \tag{11}$$

where ϵ_i is the learning rate that modulates neural update. Different from the SOM, the learning rate does not decrease over time. Instead, the firing counter η_i is used to modulate the amount of learning. The firing counter of a neuron i is initialized at 1 (Algorithm 1, Step 9.a) and decreases according to the following habituation rule (Marsland et al., 2002):

$$\Delta\eta_i = \tau_i \cdot \kappa \cdot (1 - \eta_i) - \tau_i, \tag{12}$$

where κ and τ_i are constants that control the behavior of the curve (Algorithm 1, Step 12). This mechanism causes neurons that have fired more often to be trained less, thereby fostering network convergence and to some extent resembling SOM-like implementations that gradually reduce the kernel width of the neighborhood function and the learning rate. To be noted is that, different from the SOM, the network topology of the GWR is not fixed but it rather develops over time following Hebbian-like learning, i.e. neurons

Algorithm 1 Associative Gamma-GWR (AG-GWR)

1: Start with a set of two random neurons, $A = \{\mathbf{w}_1, \mathbf{w}_2\}$ with empty context vectors $\mathbf{c}_i^k$ for $k = 1, ..., K$, $i = 1, 2$.
2: Initialize an empty set of connections $E = \emptyset$.
3: [AG-GWR only] Initialize an empty label matrix $H(i, l) = \emptyset$.
4: Initialize K empty global contexts $\mathbf{C}_k = 0$.
5: At each iteration, generate an input sample $\mathbf{x}(t)$ with label ξ.
6: Select the best and second-best matching neurons (Eq. 8):
$b = \arg\min_{i \in A} d_i(t)$, $s = \arg\min_{i \in A/\{b\}} d_i(t)$.
7: Update context descriptors:
$\mathbf{C}_k(t) = \beta \cdot \mathbf{c}_{b(t-1)}^k + (1 - \beta) \cdot \mathbf{c}_{b(t-1)}^{k-1}$.
8: Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.
9: If $(\exp(-d_b(t)) < a_T)$ and $(\eta_b < f_T)$ then:
a: Add a new neuron r ($A = A \cup \{r\}$):
$\mathbf{w}_r = 0.5 \cdot (\mathbf{x}(t) + \mathbf{w}_b)$, $\mathbf{c}_r^k = 0.5 \cdot (\mathbf{C}_k(t) + \mathbf{c}_i^k)$, $\eta_r = 1$.
b: Update edges between neurons:
$E = E \cup \{(r, b), (r, s)\}$ and $E = E/\{(b, s)\}$.
c: [AG-GWR only] Associate the sample label ξ to the neuron r:
If $(\xi \neq \emptyset)$: $H(r, \xi) = 1$, $H(r, l) = 0$, with $l \in L/\{\xi\}$.
10: If no new node is added:
a: Update weight and context of the winning neuron and its neighbors:
$\Delta\mathbf{w}_i = \epsilon_i \cdot \eta_i \cdot (\mathbf{x}(t) - \mathbf{w}_i)$, $\Delta\mathbf{c}_i^k = \epsilon_i \cdot \eta_i \cdot (\mathbf{C}_k(t) - \mathbf{c}_i^k)$.
b: [AG-GWR only] Update label values of b according to the sample label ξ:
If $(\xi \neq \emptyset)$: $\Delta H(b, \xi) = \delta^+$, $\Delta H(b, l) = -\delta^-$, with $l \in L/\{\xi\}$.
11: Increment the age of all edges connected to b of 1.
12: Reduce the firing counters of the best-matching neuron and its neighbors:
$\Delta\eta_i = \tau_i \cdot \kappa \cdot (1 - \eta_i) - \tau_i$.
13: Remove all edges with ages larger than μ_{max} and remove neurons without edges.
14: If the stop criterion is not met, repeat from step 5.

that are co-activated are bound together (Algorithm 1, Step 8). Different from the standard GWR where the network activation is given in Eq. (1), in the Gamma-GWR this function is replaced with $a_t = \exp(-d_b)$ with d_b as defined in Eq. (8) (Algorithm 1, Step 9).

3.3. Lifelong learning

In growing self-organizing networks such as the GWR (Marsland et al., 2002), available resources to allocate new information can be created in terms of new neurons and network connectivity driven by the input distribution. With the use of hierarchical Gamma-GWR networks as described in Section 3.1, lifelong learning can be achieved in terms of prediction-driven neural dynamics. In order to allocate resources for new information, the growth of the networks is modulated by their capability to reconstruct neural activation patterns from the previous network layer.

Given the two contiguous network layers G^{L-1} and G^L in our architecture, neural activations from G^{L-1} are sent to G^L via feed-forward connections. Neural growth in the learning algorithm is modulated by the ability of G^L to reconstruct activation sequences from G^{L-1} (Algorithm 1, Step 9). Given the best-matching neuron $\mathbf{w}_b$ for the current input $\mathbf{x}(t)$ as defined by the minimum distance in Eq. (7), the network activation is computed as $a(t) = \exp(-d_b(t))$ and the firing counter η_b of the neuron is updated (Algorithm 1, Step 12). A new neuron will be added when $a(t) < a_T$ and $\eta_b < f_T$, where a_T is the activation threshold that sets the maximum discrepancy (distance) between the input sequence from G^{L-1} and its best-matching neuron in G^L. Therefore, $a(t)$ can be seen as the prediction error of neural activation sequences in G^L which reconstruct prototype sequences from G^{L-1} so that each higher-level network can learn to recurrently reconstruct input sequences from lower-level networks. The firing threshold f_T assures that existing neurons are trained before allocating new ones, thus yielding the fine-tuning of the topological map.

The self-organizing learning dynamics of the hierarchy account for avoiding catastrophic interference by updating neurons according to competitive Hebbian learning to represent new information based on its similarity with existing knowledge or by allocating new neurons whenever the existing ones do not sufficiently represent the new information.

When a neuron is fired, the age of the connections from the neuron to its topological neighbors is set to 0 (Algorithm 1, Step 8), while the age of the rest of the connections is increased by a value of 1 (Algorithm 1, Step 11). Thus, the connection age mechanism is used to control the durability of information in a growing network.

3.4. Pooling layers

Computational models with deep architectures obtain invariant responses by alternating layers of feature detectors and nonlinear pooling neurons using the maximum (MAX) operation, which has been shown to achieve higher feature specificity and more robust invariance with respect to linear summation (see Guo et al., 2016 for a review). Robust invariance to translation has been obtained via MAX and average pooling, with the MAX operator showing faster convergence and improved generalization in deep neural network architectures (Scherer, Müller, & Behnke, 2010). Mechanisms of MAX-pooling in hierarchical models are also compatible with neurophysiological data (Riesenhuber & Poggio, 1999), e.g. showing that the response of neurons in the inferotemporal (IT) cortex is dominated by the stimulus producing the highest firing rate (Sato, 1989).

In our architecture, we implement MAX-pooling layers after each competitive layer (see Fig. 2). For each input image patch, a best-matching neuron $\mathbf{w}_b^{(n-1)} \in \mathbb{R}^m$ will be computed in the competitive layer $n-1$ and only its maximum weight value $\widetilde{\mathbf{w}}^{(n)} \in \mathbb{R}$ is forwarded to the next layer n:

$$\widetilde{\mathbf{w}}^{(n)} = \max_{0 \leq i \leq m} \mathbf{w}_{b,i}^{(n-1)}, \tag{13}$$

where b is computed according to Eq. (7). The tilde superscript on $\widetilde{\mathbf{w}}^{(n)}$ indicates that this value is not an actual neural weight of layer n, but rather a pooled activation value from layer $n-1$ that will be used as input in layer n to learn prototype neural weights. Since the spatial receptive field of neurons increases along the hierarchy, this pooling process will yield scale and position invariance.

3.5. Associative learning and classification

The aim of classification is to predict action labels from unseen action samples. For this purpose, the last network G^{STS} is equipped with an associative learning mechanism to map sample labels to prototype neurons representing action segments.

During the training phase, neurons in G^{STS} can be assigned a label l (with l from a set L of label classes) or remain unlabeled. The AG-GWR training algorithm for this network is illustrated by Algorithm 1. The associative matrix $H(i, l)$ stores the frequency-based distribution of sample labels in the network, i.e. each neuron i stores the number of times that a given sample label $l \in L$ has been associated to its neural weight. This labeling strategy does not require a predefined number of action classes. When a new neuron r is created and provided that ξ is the label of the input sample $\mathbf{x}(t)$, the associative matrix H is increased by one row and initialized according to $H(r, \xi) = 1$ and $H(r, l) = 0$ with $l \in L/\{\xi\}$ (Algorithm 1, Step 9.c). When instead an existing best-matching neuron b is updated, we increase $H(b, \xi)$ by a value δ^+ and decrease $H(b, l)$ of

δ^- with $\delta^+ > \delta^-$ (Step 10.b). If the sample label ξ is not present in L, a new column in H is created and initialized to $H(b, \xi) = 1$ and $H(b, l) = 0$, whereas if the input sample is not labeled, then H is not updated.

This labeling mechanism yields neurons associated to most frequent labels, thus also handling situations in which sample labels may be occasionally missing or corrupted. Neurons in the G^{STS} network are activated by the latest $K + 1$ input samples, i.e. from time t to $t - K$. The label that we take into account is the one of the most recent input $\mathbf{x}(t)$. To predict the label λ of a novel sample $\tilde{\mathbf{x}}(t)$ after the training is completed, we compute the label class with the highest value of the associative matrix for the best-matching neuron b of $\tilde{\mathbf{x}}(t)$ as $\lambda = \arg\max_{l \in L} H(b, l)$.

4. Experiments and results

We report experimental results with two action benchmarks: the Weizmann (Gorelick et al., 2005) and the KTH (Schuldt et al., 2004) datasets. We trained our architecture with each dataset using similar training parameters but different experimental setups for compatibility with the evaluation schemes used in the literature and a direct comparison in terms of overall classification accuracy. We conducted a series of experiments showing that for both datasets our model: (i) achieves competitive classification results with respect to the state of the art for batch learning, (ii) learns robust action-label mappings also in the case of occasionally missing or corrupted class labels during training sessions, and (iii) accounts for lifelong learning in terms of adapting to non-stationary input avoiding catastrophic interference.

4.1. Training parameters

For the training sessions, the ventral pathway is fed with sequences of actions at 10 frames per second with grayscaled images resized to 30×30 pixels, while the dorsal pathway is fed with motion-flow images computed from two consecutive frames.

In the first layer, we compute image patches of 3×3 pixels for both posture and motion sequences, i.e. a total of 784 patches for each 30×30 input image. In each pathway, the image patches are fed into the recurrent networks G_1^P and G_1^M, respectively, both comprising a Gamma-GWR with 1 context descriptor ($K = 1$). In the second layer, the input is represented by the pooled activation from the first two networks, yielding a 28×28 representation for each processed frame. From this representation, we compute 4×4 patches that are fed into G_2^P and G_2^M with $K = 3$. In the third layer, the information from the ventral and the dorsal pathways are integrated, i.e. the pooled activation from the pathways is concatenated producing 14×7 matrices for each frame used to train G^{STS} with $K = 5$. If we consider the hierarchical processing of the architecture, the last network yields neurons that respond to the latest 10 frames which correspond to 1 second of video.

The network parameters are summarized in Table 1. Although the best classification results may be obtained by empirically testing different configurations of parameters over multiple training sessions, adequate network parameters can be conveniently defined on the basis of the following considerations.

The number of context descriptors (K) in each layer is designed to learn action representations with increasing degree of complexity in the spatiotemporal domain. Due to the recursive definition of context descriptors in each network, decreasing values of α_i have been shown to reduce the propagation of errors from lower-level layers to higher-level layers (Estévez & Vergara, 2012). Thus, we assign decreasing values to α_i according to the following function:

$$\mathbf{p}_N = \left[\frac{\alpha^i}{\sum_i \alpha^i} : \alpha^i = \frac{1}{N} - \exp(-(i+2)) \right], \quad i = 1, \ldots, N \qquad (14)$$

Table 1
Training parameters for the Gamma-GWR architecture.

Parameter	Value
Insertion thresholds (a_T)	$G_1 = 0.6, G_2 = 0.9, G^{STS} = 0.7$
Firing threshold	$f_T = 0.1$
Firing counter	$\tau_b = 0.3, \tau_i = 0.1, \kappa = 1.05$
Context descriptor	$\beta = 0.7$
Learning rates	$\epsilon_b = 0.1, \epsilon_n = 0.001$
Maximum age	$\mu_{max} = 200$
Labeling	$\delta^+ = 1, \delta^- = 0.1$

Table 2
Weights for the recurrent activation of the Gamma-GWR.

Network	$\mathbf{p}_N$
G_1^P/G_1^M	$\mathbf{p}_2 = [0.536, 0.464]$
G_2^P/G_2^M	$\mathbf{p}_4 = [0.318, 0.248, 0.222, 0.212]$
G^{STS}	$\mathbf{p}_6 = [0.248, 0.178, 0.152, 0.142, 0.139, 0.138]$

with $N = K + 1$, i.e. the number of context descriptors plus the current weight vector. For our three-layer architecture, this function yielded the values shown in Table 2. The context descriptor parameter β, which defines the contribution of the present and the past input, has been shown to yield smaller temporal quantization error for values $\beta \in [0.6, 0.7]$ (Parisi et al., 2016; Strickert & Hammer, 2005).

The insertion threshold f_T and the firing counter parameters τ_b, τ_i, κ, as well as the learning rates for the best-matching neuron ϵ_b and its topological neurons ϵ_n were selected on the basis of previous studies showing that these values yield an adequate network response (neural growth) and levels of stability (learning convergence) when exposed to both stationary and non-stationary input distributions (Marsland et al., 2005, 2002; Parisi et al., 2015).

4.2. Classification accuracy with batch learning

4.2.1. Weizmann dataset

The Weizmann dataset (Gorelick et al., 2005) contains 90 low-resolution color image sequences with 10 actions performed by 9 subjects. The actions are *walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack*, and *skip*. Sequences are sampled at 180×144 pixels with static background and are about 3 s long. We used aligned foreground body shapes by background subtraction included in the dataset (Fig. 3).

For compatibility with Schindler and Van Gool (2008), we trimmed all sequences to a total of 28 frames, which is the length of the shortest sequence, and evaluated our approach by performing *leave-one-out* cross-validation, i.e., 8 subjects were used for training and the remaining one for testing. This procedure was repeated for all 9 permutations and the results were averaged. Results for the recognition of 10-frame snippets are shown in Table 3.

Our experiments yield an overall accuracy of 98.7%, which is a very competitive result with respect to the state of the art of 99.64% reported by Gorelick et al. (2005). In their approach, the authors extract action features over a number of frames by concatenating 2D body silhouettes in a space–time volume. These features are then fed into simple classifiers: nearest neighbors and Euclidean distance. Schindler and Van Gool (2008) obtained an accuracy of 99.6% by combining pose and motion cues. In their two-pathway architecture, the filter responses are MAX-pooled and then compared to a set of learned templates. The similarities from both pathways are concatenated to a feature vector and classified by a bank of linear classifiers. Their experiments show that local body pose and optic flow for a single frame are enough to achieve around 90% accuracy, with snippets of 5–7 frames (0.3–0.5 s of video) yielding similar results to experiments with 10-frame snippets.

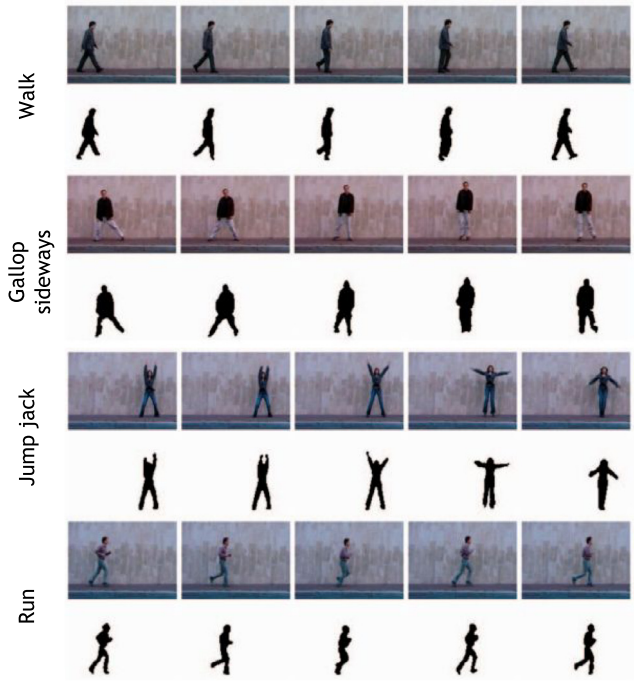

Fig. 3. Sample action frames and segmented silhouettes from the Weizmann dataset.
Source: Adapted from Gorelick et al. (2005).

Table 3
Results on the Weizmann dataset for 10-frame snippets and full action sequences. Results from Jung et al. (2015) with 3 models: [1]CNN, [2]MSTNN, and [3]3D-CNN. Results appear as reported by their authors.

10-second action snippets	Accuracy (%)
Gorelick et al. (2005)	99.64
Schindler and Van Gool (2008)	99.6
Our approach	**98.7**
Jung et al. (2015)	92.9[1], 95.3[2], 96.2[3]
Full action sequences	**Accuracy (%)**
Gorelick et al. (2005)	100
Blank, Gorelick, Shechtman, Irani, and Basri (2007)	100
Schindler and Van Gool (2008)	100
Our approach	**100**
Jhuang, Serre, Wolf, and Poggio (2007)	98.8
Thurau and Hlavác (2008)	94.4
Niebles, Wang, and Fei-Fei (2008)	90

Our results outperform the overall accuracy reported by Jung et al. (2015) with three different deep learning models: convolutional neural network (CNN, 92.9%), multiple spatiotemporal scales neural network (MSTNN, 95.3%), and 3D CNN (96.2%). However, a direct comparison of the above-described methods with ours is hindered by the fact that they differ in the type of input and number of frames per sequence used during the training and the test phase.

Most of the results in the literature are reported at the level of correctly classified sequences. Therefore, we also evaluated our approach on full sequence classification to compare it to the state of the art. For each action sequence, we predicted labels from 10-frame snippets and then considered the prediction with the highest statistical mode as the output label for that sequence. Results at a sequence level are shown in Table 3. Although this evaluation scheme yields better results, the main drawback is that all the frames of a sequence have to be processed to predict one action label.

4.2.2. KTH dataset

The KTH action dataset (Schuldt et al., 2004) contains 25 subjects performing 6 different actions: *walking, jogging, running, boxing, hand-waving* and *hand-clapping*, for a total of 2391 sequences. Action sequences were performed in 4 different scenarios: indoor, outdoor, variations in scale, and changes in clothing (see Fig. 4). Videos were collected with a spatial resolution of 160 × 120 pixels taken over homogeneous backgrounds and sampled at 25 frames per second, containing considerable variations in sequence duration and viewpoint.

Following the evaluation schemes proposed in the literature for this dataset, we trained our model with 16 randomly selected subjects and used the other 9 subjects for testing. The overall classification accuracy averaged across 5 trials achieved by our model is shown in Table 4. Our result is competitive with the two best results: 95.6% by Ravanbakhsh et al. (2015) and 95.04% by Gao et al. (2010). Ravanbakhsh et al. (2015) used a hierarchical CNN model to capture sub-actions from complex ones. Higher levels of the hierarchy represent a coarse capture of an action sequence and lower levels represent fine action elements. Furthermore, key frames are extracted using binary coding of each frame in a video which helps to improve the performance of the hierarchical model (from 94.1% to 95.6%). To be noted is that the reported accuracy represents the result of the best trial, while an averaged accuracy across multiple trials has not been reported by the authors. Gao et al. (2010) used a method for computing interest points with substantial motion, the MoSIFT (Chen & Hauptmann, 2009), which first applies the SIFT algorithm to find distinctive visual components in the spatial domain and then detect spatiotemporal interest points with motion constraints. This results in high computational requirements for the estimation of ad-hoc interest points. To be noted is that the results reported by Gao et al. (2010) correspond to the average on the five best runs over a total of 30 trials, while the classification accuracy decreases to 90.93% if averaging the five worst ones.

Our model performs better than other proposed hierarchical models that do not rely on handcrafted features, such as Ji et al. (2013) using a 3D convolutional neural network (3D-CNN, 90.2%) and Baccouche et al. (2011) combining a 3D-CNN with a long short-term memory (LSTM, 94.39%).

These comparative experiments are the result of training the model with a batch of labeled samples and a predefined number of action classes, whereas the purpose of our model is to exhibit robust performance also in the case of unavailable training labels and the incremental learning of action sequences that become available over time.

4.3. Classification with missing or corrupted labels

An additional experiment consisted of decreasing the percentage of available and correct sample labels during the training phase. Visual representations are progressively learned without supervision, thus the removal or corruption of a number of sample labels will not have an impact on the correct development of neural representations. However, missing and corrupted labels will negatively impact on the development of associative connections between the neural map and the symbolic label layer in terms of a decreased classification accuracy. This accuracy decrease should be attenuated by the associative labeling method introduced in Section 3.5 as long as a sufficient number of available and correct labels are present.

We decreased the percentage of available labels from 100% to 0% from randomly chosen samples. The average classification accuracy with different percentages of omitted labels over 10 training runs for both datasets is displayed in Fig. 5. Although an increasing number of missing labeled samples during the training phase has a negative impact on the classification accuracy

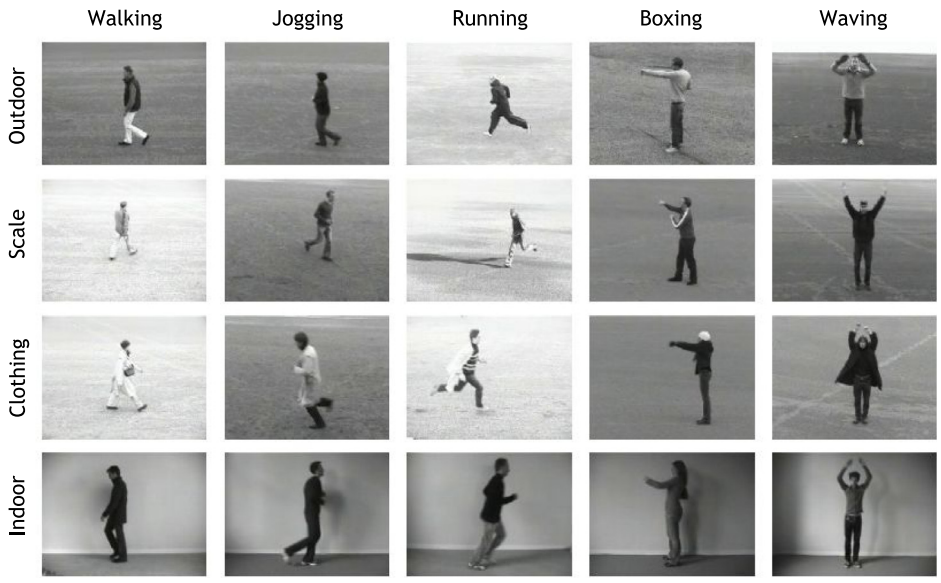

Fig. 4. A sample of the 6 actions and 4 scenarios from the KTH dataset.
Source: Adapted from Schuldt et al. (2004).

Table 4
Results on the KTH dataset. Unless otherwise stated, results show the accuracy averaged across 5 trials. Results as reported by the authors.

	Accuracy (%)
Ravanbakhsh, Mousavi, Rastegari, Murino, and Davis (2015) (best trial)	95.6
Gao, Chen, Hauptmann, and Cai (2010) (5 best trials out of 30)	95.04
Our approach	**94.91**
Baccouche et al. (2011)	94.39
Wang, Klaser, Schmid, and Liu (2011)	94.2
Schindler and Van Gool (2008)	92.7
Jhuang et al. (2007)	91.7
Ji, Xu, Yang, and Yu (2013)	90.2
Niebles et al. (2008)	83.3
Schuldt et al. (2004)	71.7

for both datasets, this decline in performance is not proportional to the number of omitted labels. For the Weizmann dataset, as soon as 10% of labeled samples are available during the training, the model shows an accuracy of 45% and accuracy values above 95.7% can be observed for at least 50% available labeled samples. A similar performance behavior was obtained with the KTH dataset, yielding an accuracy above 91.5% for at least 50% available labeled samples.

In the case of a number of missing labels during training sessions, a direct comparison of the performance of our model with the approaches discussed in Section 4.2 is hindered by the fact that most of the learning methods are supervised, i.e. labels are needed during the training procedure as a target function. A possible comparison can be performed by introducing a number of randomly labeled samples in order to examine the robustness of the models exposed to a set of corrupted labels. In a recent study, Zhang, Bengio, Hardt, Recht, and Vinyals (2016) inspected the behavior of supervised neural networks trained with a varying level of label noise, showing a slowdown of the convergence time with an increasing level of corrupted labels. Furthermore, as the noise level approaches 100%, the generalization errors converge to 90%. In our case, a set of corrupted labels should have a similar impact as missing labels, without significantly compromising the overall accuracy as long as a sufficient number of correct labels are present (Fig. 5, dashed lines). For both datasets, the overall accuracy shows a slight decrease for up to 40% of corrupted labels (from 98.7% to 96.2% for the Weizmann and from 94.91% to 91.8% for the KTH), with a more significant accuracy decrease for over 50% of corrupted labels.

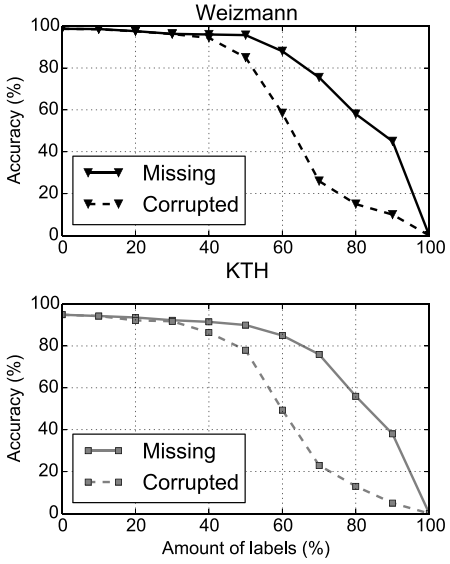

Fig. 5. Average classification accuracy on the Weizmann and the KTH dataset over 10 runs for an increasing percentage of missing and corrupted sample labels.

4.4. Learning dynamics

The self-organizing hierarchy yields progressively specialized neurons with increasingly large spatiotemporal receptive fields,

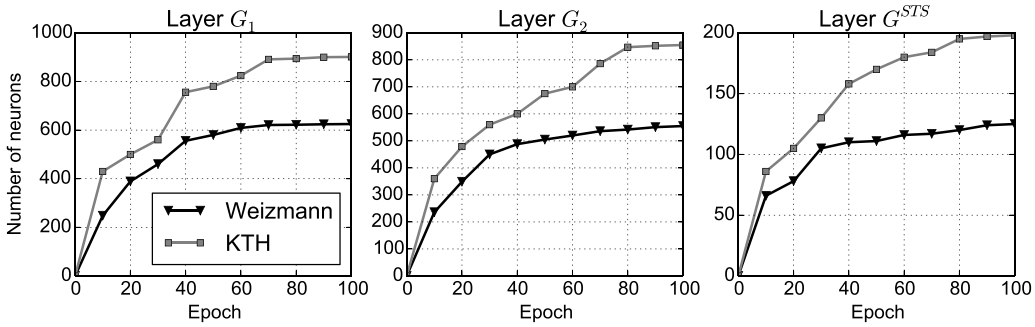

Fig. 6. Number of neurons created over 100 epochs. For the network layers G_1 and G_2, we show the averaged number of neurons from the ventral and the dorsal pathway.

i.e. neurons in each layer will fire for larger parts of the image for an increasing number of image frames. Therefore, the number of neurons created in each layer is driven by the spatiotemporal resolution of its input, with prototype action sequences being distributively encoded in the neurons of the map in terms of recursively activated neurons. In the last layer of our architecture, STS neurons compute up to 1 s of information, i.e. 10 frames.

To analyze the dynamics of the self-organizing hierarchy, we plot the process of neural growth for each network layer over a number of training epochs for both datasets. The number of neurons created over 100 training epochs is shown in Fig. 6. For the network layers G_1 and G_2, we show the averaged number of neurons from the ventral and the dorsal pathway. The KTH dataset led to a greater number of neurons created in the first layer with respect to the Weizmann dataset, since the former contains action sequences with a cluttered background in different scenarios, whereas the latter contains segmented body silhouettes on a uniform background. For both datasets, it can be seen that in the first network layer the number of neurons tends to converge earlier than in the subsequent layer suggesting that the neural weights of G_1 are recombined in G_2 to represent more complex spatiotemporal properties of the input. Similarly, the number of neurons in G_2 converges faster than the ones in G^{STS}. Such a neural growth behavior suggests that prototype neural weights are shared across multiple levels of the hierarchy.

4.5. Catastrophic interference

The goal of lifelong learning systems is to avoid catastrophic interference (see discussion in Section 3.3). We argue that the mechanism to prevent the acquisition of new information from forgetting existing knowledge is embedded in the dynamics of the self-organizing learning algorithm that allocates new neurons or updates existing ones based on the discrepancy between the input distribution and the prototype neural weights. To support this claim, we conducted an additional experiment in which we explore how our model accounts for avoiding catastrophic interference when learning new action classes.

For both datasets, we first trained the model with a single action class and then scaled up progressively to all the others in order to observe how the performance of the model changes for an increasing number of action classes. The results are shown in Fig. 7, where the classification accuracy was averaged across all the combinations for a given number of action classes. Although the performance decreases as the number of action classes is increased, this decline is not catastrophic (from 100% to 98.7% $\pm$ 0.8% for the Weizmann dataset and from 100% to 94.91% $\pm$ 1.1% for the KTH). However, it is complex to establish whether this accuracy decrease is caused by catastrophic interference or the labeling strategy. We observed that the overall quantization error of the networks tends to decrease over the training epochs, suggesting that the prototype neurons are effectively allocated and fine-tuned to better represent the input distribution.

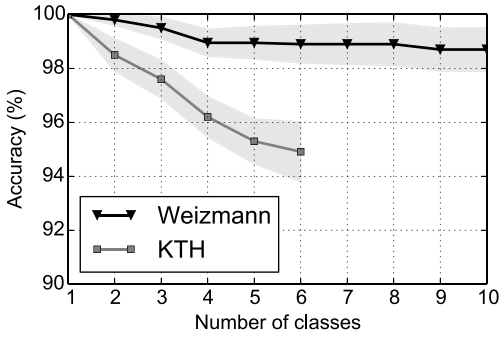

Fig. 7. Effects of incremental learning: classification accuracy averaged across all the combinations for a given number of action classes.

5. Discussion

5.1. Summary

In this paper, we propose a deep neural architecture with a hierarchy of self-organizing GWR networks for the lifelong learning of action representations. Different from previous models in which action representations are learned from handcrafted 3D features (Parisi et al., 2016, 2015), we use a hierarchy of recurrent networks to learn prototype action features with increasingly large spatiotemporal receptive fields. We propose a temporal extension of the GWR, the Gamma-GWR, showing that a self-organizing hierarchy accounts for the learning of spatiotemporal properties of the input with increasing complexity of representation. Pooling layers are used to maximize the variance of projection of each layer yielding invariance to scale and position along the hierarchy. For the purpose of classification, visual representations obtained through unsupervised learning are incrementally associated with symbolic action labels. This is achieved through the use of an associative mechanism in the Gamma-GWR that attaches labels to prototype neurons based on their occurrence frequency.

We compare the classification accuracy of our model to state-of-the-art learning methods on the Weizmann (Gorelick et al., 2005) and the KTH (Schuldt et al., 2004) action benchmarks, showing competitive performance for the batch learning scenario with different evaluation schemes. Additional experiments show that our learning architecture can adapt to non-stationary input avoiding catastrophic interference and handle situations in which a number of sample labels is missing or corrupted. Future work directions include the evaluation of our method with larger-scale datasets, e.g. the UCF action recognition dataset (Soomro, Zamir, & Shah, 2012) containing 101 classes of human actions and different amounts of available and noisy labels.

5.2. Deep neural self-organization

Motivated by the process of input-driven self-organization exhibited by topographic maps in the cortex (Nelson, 2000; Willshaw & von der Malsburg, 1976), we proposed a learning architecture encompassing a hierarchy of growing self-organizing networks. The proposed hierarchical architecture yields progressively specialized neurons encoding latent spatiotemporal dynamics of the input. Neurons in higher-level layers will encode prototype sequence-selective snapshots of visual input, following the assumption that the recognition of actions must be selective for temporal order (Giese & Poggio, 2003). In previous research (Parisi et al., 2016, 2015), the temporal processing of features was explicitly modeled in terms of neurons in higher-level layers computing the concatenation of neural activation trajectories from lower-level layers, which increases the dimensionality of neural weights along the hierarchy. This issue has been addressed in this paper, where we proposed a novel temporal extension of the GWR with context learning (Strickert & Hammer, 2005) and a Gamma Memory model (de Vries & Príncipe, 1992), the Gamma-GWR, showing that hierarchically arranged GWR networks with recurrent connections can account for the learning of action features with increasingly large spatiotemporal receptive fields.

Similar to previous self-organizing models using context learning, the Gamma-GWR computes the best-matching neuron as a linear combination between the current input and the temporal context. Each Gamma-GWR network in the hierarchy has a fixed set of context descriptors weighted by a decay function (Eq. (14)). Other recurrent models with state-of-the-art performance in supervised sequence classification tasks such as the long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) learn the distribution of the decay function, thus not assuming an exponential decay as in the case of self-organizing networks with context learning. LSTM networks do not use an activation function within their recurrent components but rather implement a set of gates (i.e. input, forget, and output gates) that control the information flow into or out of a memory cell. On the other hand, this type of processing requires pre-established gate values for overcoming the vanishing gradient problem of recurrent networks that implement backpropagation through time (Schmidhuber, 1992).

We argue that computational models of hierarchical neural self-organization enable the efficient learning of latent spatiotemporal properties of the input over time. Nevertheless, our GWR-based architecture is the result of simplified modeling assumptions, diverging significantly from a set of biological findings on cortical brain areas. The GWR algorithm is an efficient computational model that incrementally adapts to dynamic input. However, its process of neural growth does not resemble biologically plausible mechanisms of neurogenesis (e.g. Eriksson, Perfilieva, Bjork-Eriksson, Alborn, Nordborg, Peterson, & Gage, 1998; Ming & Song, 2011). Furthermore, a characteristic of GWR models as well as other well-established self-organizing networks such as SOM and GNG is that best-matching neurons are computed according to a winner-take-all competition, which requires biologically unrealistic knowledge about the global state of the network (Nelson, 2000). Implausible mechanisms of Hebbian learning inherited by SOM-like models have been extended with additional stabilization and competition processes for biologically plausible experience-driven learning (e.g. Zenke & Gerstner, 2017; Zenke, Gerstner, & Ganguli, 2017).

In our proposed architecture, each network layer learns action representations with an increasing degree of complexity from pose–motion features. While the processing of body action features in two distinct neural pathways and their subsequent integration in the STS are in line with neurophysiological evidence (Giese & Poggio, 2003; Giese & Rizzolatti, 2015), our model is a strong simplification with respect to biological mechanisms where the two streams comprise interactions at multiple levels (Felleman & Van Essen, 1991). Additionally, recurrent GWR networks followed by MAX-pooling shape a functional hierarchy that shares lower-level features for composing higher-level representations, without aiming to model the response of neurons in any specific brain area (e.g. Stevens, Law, Antolík, & Bednar, 2013; Yamins & DiCarlo, 2016). In terms of neural activity, more biologically plausible models using spiking neural networks have been proposed in which spatiotemporal information is encoded as locations of synapses and neurons taking into account the time of their spiking activity (Buonomano & Maass, 2009; Kasabov, 2014; Kasabov, Doborjeh, & Doborjeh, 2017; Zenke, Agnes, & Gerstner, 2015).

5.3. Lifelong learning of action representations

Most of the current deep neural network models in the literature rely on the assumption that a batch of samples is available before the training session, thus they do not account for the processing of non-stationary datasets. A well-known issue of lifelong learning is the *stability–plasticity dilemma* aimed to avoid *catastrophic interference* in computational models, i.e., the process of new knowledge overwriting old knowledge. Specifically for self-organizing networks, catastrophic interference is modulated by the conditions of map plasticity, the available resources to represent information, and the similarity between new and old knowledge (Richardson & Thomas, 2008).

In our architecture, Gamma-GWR networks are used to implement lifelong learning in terms of prediction-driven neural dynamics, so that neural growth in each layer is driven by the capability of a network to reconstruct neural activation patterns from the previous layer with a given accuracy. This principle roughly resembles schemes of predictive coding (Huang & Rao, 2011; Rao & Ballard, 1999) proposing that the visual cortex can be modeled as a hierarchical network with reciprocal connections where top-down feedback connections from higher-order cortical areas convey predictions of lower-order neural activity and bottom-up connections carry the residual prediction errors. In our case, we do not explicitly model connections conveying prediction errors but rather the condition necessary to minimize prediction error is embedded in the process modulating neural growth (Algorithm 1, Step 9). During this prediction-driven learning, associative connections will be developed in network layer G^{STS} between the neural representation and the symbolic labels taken from the dataset.

We performed an additional experiment to observe how the classification accuracy changes for an increasing number of action classes used to train the model. Experimental results show that the effects of novel information are not catastrophic for the classification task, yielding accuracy values from 100% for one action class to 98.7% $\pm$ 0.8% for all the classes of the Weizmann dataset and from 100% to 94.91% $\pm$ 1.1% for the KTH. We observed that the overall quantization error of the networks tends to decrease over the training epochs, suggesting that the prototype neurons are effectively allocated and fine-tuned to better represent the input distribution. However, although a larger number of neurons reduces the average quantization error between the input and its neural representation, there is no linear relation between the number of neurons and the classification performance. This occurs because the classification process consists of predicting the label of novel samples by retrieving attached labels to the inputs' best-matching neurons, irrespective of the actual distance between the novel inputs and the selected neurons. Therefore, convenient insertion thresholds should be chosen by taking into account the distribution of the input and, in the case of a classification task, the classification performance could also be used to modulate neural

growth. More specifically, feedback connectivity from the symbolic layer containing action labels could have modulatory effects on the growth rate of lower-level networks so that a sufficient number of prototype neurons are created as a dictionary of primitives subsequently used to learn spatiotemporal statistics of the input. This mechanism may be employed to modulate the amount of learning necessary to adapt to the dynamic input distribution and develop robust action representations. In this case, convenient threshold values should be chosen so that the layers adapt to dynamic input for yielding smaller prediction errors while showing convergence with stationary input.

6. Conclusion

In contrast to batch learning approaches, lifelong learning is crucial for the incremental development of knowledge based on progressively available perceptual cues. In this paper, we showed that lifelong learning can be developed in terms of prediction-driven neural dynamics with action representations emerging in a hierarchy of self-organizing neural networks. Our learning model exhibits competitive performance with respect to state-of-the-art deep learning models trained with a predefined number of action classes, showing robust performance also in the case of missing or corrupted sample labels and adapting to non-stationary input distributions.

The proposed architecture can be considered a further step towards more flexible neural network models for learning robust visual representations that develop and fine-tune over time on the basis of visual experience. Additional principles that play a role in lifelong learning such as the influence of reward-driven self-organization (Aswolinskiy & Pipa, 2015) and attentional functions (Ivanov, Liu, Clerkin, Schulz, Friston, Newcorn, & Fan, 2012) in the development of topological maps will be subject to future research.

Acknowledgments

This research was supported by the DAAD German Academic Exchange Service (Kz:A/13/94748) and the German Research Foundation (DFG) under project Transregio Crossmodal Learning (TRR 169).

References

Andreakis, A., von Hoyningen-Huene, N., & Beetz, M. (2009). Incremental unsupervised time series analysis using merge growing neural gas. In *Lecture Notes in Computer Science: vol. 5629. WSOM* (pp. 10–18). Springer.

Aswolinskiy, W., & Pipa, G. (2015). RM-SORN: a reward-modulated self-organizing recurrent neural network. *Frontiers in Computational Neuroscience*, *9*, 36.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In *Human behavior understanding: Second international workshop, HBU 2011, Amsterdam, the Netherlands, November 16, 2011. Proceedings* (pp. 29–39). Berlin, Heidelberg: Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-25446-8_4.

Beyer, O., & Cimiano, P. (2011). Online labelling strategies for growing neural gas. In H. Yin, W. Wang, & V. Rayward-Smith (Eds.), *LNCS: vol. 6936. Proceedings of the international conference on intelligent data engineering and automated learning*. (IDEAL), (pp. 76–83). Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-23878-9_10.

Blakemore, C., & Cooper, G. F. (1970). Development of the brain depends on the visual environment. *Nature*, *228*(5270), 477–478.

Blakemore, C., & Van Sluyters, R. C. (1975). Innate and environmental factors in the development of the kitten's visual cortex. *Journal Physiology*, *248*(3), 663–716.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253.

Buonomano, D., & Maass, W. (2009). State-dependent computations: Spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, *10*(2), 113–125.

Chappell, G. J., & Taylor, J. G. (1993). The temporal Kohonen map. *Neural Networks*, *6*(3), 441–445. http://dx.doi.org/10.1016/0893-6080(93)90011-K.

Chen, M.-y., & Hauptmann, A. (2009). MoSIFT: Reocgnizing human actions in surveillance videos. CMU-CS-09-161. Carnegie Mellon University.

de Vries, B., & Príncipe, J. C. (1992). The gamma model–A new neural model for temporal processing. *Neural Networks*, *5*(4), 565–576.

Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, *10*(4), 12–25. http://dx.doi.org/10.1109/MCI.2015.2471196.

Eriksson, P. S., Perfilieva, E., Bjork-Eriksson, T., Alborn, A.-M., Nordborg, C., Peterson, D. A., & Gage, F. H. (1998). Neurogenesis in the adult human hippocampus. *Nature Medicine*, *4*(11), 1313–1317. http://dx.doi.org/10.1038/3305.

Estévez, P. A., & Hernández, R. (2011). Gamma-filter self-organizing neural networks for time series analysis. In *WSOM*, (pp. 151–159).

Estévez, P. A., & Vergara, J. R. (2012). Nonlinear time series analysis by using gamma growing neural gas. In *WSOM*, (pp. 205–214).

Felleman, D., & Van Essen, D. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, *1*(1), 1–47.

Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in neural information processing systems, Vol. 7* (pp. 625–632). MIT Press.

Gao, Z., Chen, M.-y., Hauptmann, A. G., & Cai, A. (2010). Comparing evaluation protocols on the KTH dataset. In *Human behavior understanding: First international workshop, HBU 2010, Istanbul, Turkey, August 22, 2010. Proceedings* (pp. 88–100). Berlin, Heidelberg: Springer Berlin Heidelberg.

Giese, M. A., & Poggio, T. (2003). Neural mechanisms for the recognition of biological movements.. *Nature Reviews Neuroscience*, *4*(3), 179–192. http://dx.doi.org/10.1038/nrn1057.

Giese, M. A., & Rizzolatti, G. (2015). Neural and computational mechanisms of action processing: Interaction between visual and motor representations. *Neuron*, *88*(1), 167–180. http://dx.doi.org/10.1016/j.neuron.2015.09.040.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2005). Actions as space-time shapes. In *ICCV*, (pp. 1395–1402).

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, *11*(1), 23–63. http://dx.doi.org/10.1016/S0364-0213(87)80025-3.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. http://dx.doi.org/10.1016/j.neucom.2015.09.116. Recent Developments on Deep Big Vision.

Hasson, U., Yang, E., Vallines, I., Heeger, D. J., & Rubin, N. (2008). A hierarchy of temporal receptive windows in human cortex. *The Journal of Neuroscience*, *28*(10), 2539–2550. http://dx.doi.org/10.1523/jneurosci.5487-07.2008.

Hirsch, H., & Spinelli, D. (1970). Visual experience modifies distribution of horizontally and vertically oriented receptive fields in cats. *Science*, *168*(3933), 869–871.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *8*(9), . http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Huang, Y., & Rao, R. P. N. (2011). Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, *2*(5), 580–593.

Hubel, D. H., & Wiesel, T. H. (1962). Receptive fields, binocular and functional architecture in the cat's visual cortex. *Journal Physiology*, *160*, 106–154.

Ivanov, I., Liu, X., Clerkin, S., Schulz, K., Friston, K., Newcorn, J. H., & Fan, J. (2012). Effects of motivation on reward and attentional networks: an fMRI study. *Brain and Behavior*, *2*(6), 741–753. http://dx.doi.org/10.1002/brb3.80.

Jain, A., Tompson, J., LeCun, Y., & Bregler, C. (2015). MoDeep: A deep learning framework using motion features for human pose estimation. In *Computer vision –ACCV 2014: 12th Asian conference on computer vision, Singapore, Singapore, November 1-5, 2014, Revised selected papers, part II* (pp. 302–315). Cham: Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-16808-1_21.

Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007). A biologically inspired system for action recognition. In *International conference on computer vision, ICCV*.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Johnson, J., & Newport, E. (1991). Critical period effects on universal properties of language: the status of subjacency in the acquisition of a second language. *Cognition*, *39*(3), 215–258.

Jung, M., Hwang, J., & Tani, J. (2015). Self-organization of spatio-temporal hierarchy via learning of dynamic visual image patterns on action sequences. *PLoS One*, *10*(7), e0131214. http://dx.doi.org/10.1371/journal.pone.0131214.

Kasabov, N. K. (2014). NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, *52*, 62–76.

Kasabov, N. K., Doborjeh, M. G., & Doborjeh, Z. G. (2017). Mapping, learning, visualization, classification, and understanding of fMRI data in the NeuCube evolving spatiotemporal data machine of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(4), 887–899.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*, 1464–1480.

Lee, J. (2012). Cumulative learning. In *Encyclopedia of the sciences of learning* (pp. 887–893). Boston, MA: Springer US, (Chapter).

Lerner, Y., Honey, C. J., Silbert, L. J., & Hasson, U. (2011). Topographic mapping of a hierarchy of temporal receptive windows using a narrated story. *The Journal*

of Neuroscience, *31*(8), 2906–2915. http://dx.doi.org/10.1523/jneurosci.3684-10.2011.
Marsland, S., Nehmzow, U., & Shapiro, J. (2005). On-line novelty detection for autonomous mobile robots. *Robotics and Autonomous Systems*, *51*(2–3), 191–206.
Marsland, S., Shapiro, J., & Nehmzow, U. (2002). A self-organising network that grows when required.. *Neural Networks*, *15*(8–9), 1041–1058.
Martinetz, T. M. (1993). Competitive Hebbian learning rule forms perfectly topology preserving maps. In *ICANN'93: International conference on artificial neural networks* (pp. 427–434). Amsterdam: Springer.
McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why There are Complementary Learning Systems in the Hippocampus and Neocobelrtex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419–457.
Ming, G.-l., & Song, H. (2011). Adult neurogenesis in the mammalian brain: Significant answers and significant questions. *Neuron*, *70*(4), 687–702. http://dx.doi.org/10.1016/j.neuron.2011.05.001. URL. http://dx.doi.org/10.1038/nrn2147.
Mozer, M. (1995). A focused backpropagation algorithm for temporal pattern recognition. In *Hillsdale* (pp. 137–169). NJ: Lawrence Erlbaum Associates.
Nelson, C. A. (2000). Neural plasticity and human development: the role of early experience in sculpting memory systems. *Developmental Science*, *3*(2), 115–136.
Niebles, J. C., Wang, H., & Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, *79*(3), 299–318.
Parisi, G. I., Magg, S., & Wermter, S. (2016). Human motion assessment in real time using recurrent self-organization. In *Proc. of the IEEE international symposium on robot and human interactive communication, RO-MAN* (pp. 71–76). New York, US.
Parisi, G. I., Tani, J., Weber, C., & Wermter, S. (2016). Emergence of multimodal action representations from neural network self-organization. *Cognitive Systems Research*. http://dx.doi.org/10.1016/j.cogsys.2016.08.002.
Parisi, G. I., Weber, C., & Wermter, S. (2015). Self-Organizing neural integration of pose-motion features for human action recognition. *Frontiers in Neurorobotics*, *9*(3), . http://dx.doi.org/10.3389/fnbot.2015.00003.
Perrett, D., Rolls, E., & Caan, W. (1982). Visual neurons responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, *47*, 329–342.
Poggio, T., & Smale, S. (2003). The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, *50*, 2003.
Rao, R., & Ballard, D. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*, 79–87.
Ravanbakhsh, M., Mousavi, H., Rastegari, M., Murino, V., & Davis, L. S. (2015). Action recognition with image based CNN features. CoRR abs/1512.03980.
Richardson, F. M., & Thomas, M. S. C. (2008). Critical periods and catastrophic interference effects in the development of self-organising feature maps. *Developmental Science*, *11*(3), 371–389.
Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*, 1019–1025.
Sato, T. (1989). Interactions of visual stimuli in the receptive fields of inferior temporal neurons in macaque. *Experimental Brain Research*, *77*, 23–30.
Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In *Proceedings of the 20th international conference on artificial neural networks: Part III*. ICANN'10, (pp. 92–101). Berlin, Heidelberg: Springer-Verlag.
Schindler, K., & Van Gool, L. J. (2008). Action snippets: How many frames does human action recognition require? In *CVPR*. IEEE Computer Society.
Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, *4*, 234–242.
Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *Proceedings of the pattern recognition, 17th international conference on (ICPR'04) Volume 3 - Volume 03*. ICPR '04, (pp. 32–36). Washington, DC, USA: IEEE Computer Society.
Senghas, A., Kita, S., & Ozyurek, A. (2004). Children creating core properties of language: Evidence from an emerging sign language in nicaragua. *Science*, *305*(5691), 1779–1782. http://dx.doi.org/10.1126/science.1100199.
Sengpiel, F., & Stawinski, P. T. B. (1999). Influence of experience on orientation maps in cat visual cortex. *Nature Neuroscience*, *2*(8), 727–732.
Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR*.
Stanley, J. C. (1976). Computer simulation of a model of habituation. *Nature*, *261*, 146–148. http://dx.doi.org/10.1038/261146a0.
Stevens, J.-L. R., Law, J. S., Antolík, J., & Bednar, J. A. (2013). Mechanisms for stable, robust, and adaptive development of orientation maps in the primary visual cortex. *Journal of Neuroscience*, *33*(40), 15747–15766.
Strickert, M., & Hammer, B. (2005). Merge SOM for temporal data. *Neurocomputing*, *64*, . http://dx.doi.org/10.1016/j.neucom.2004.11.014.
Tan, C., Singer, J. M., Serre, T., Sheinberg, D., & Poggio, T. A. (2013). Neural representation of action sequences: how far can a simple snippet-matching model take us? In *Advances in neural information processing systems 26: 27th Annual conference on neural information processing systems 2013. Proceedings of a meeting held December 5-8, 2013*, Lake Tahoe, Nevada, United States. (pp. 593–601).
Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks*, *16*(1), 11–23.
Tani, J., & Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, *12*(7–8), 1131–1141.
Taylor, P., Hobbs, J. N., Burroni, J., & Siegelmann, H. T. (2015). The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions. *Scientific Reports*, *5*(18112), . http://dx.doi.org/10.1038/srep18112.
Thurau, C., & Hlavác, V. (2008). Pose primitive based human action recognition in videos or still images. In *CVPR*. IEEE Computer Society.
Ungerleider, L., & Mishkin, M. (1982). Two cortical visual systems. In *Analysis of visual behavior* (pp. 549–586). Cambridge: MIT Press.
Vangeneugden, J., Pollick, F., & Vogels, R. (2009). Neural and computational mechanisms of action processing: Interaction between visual and motor representations. *Cerebral Cortex*, *19*(3), 593–611. http://dx.doi.org/10.1093/cercor/bhn109.
Wang, H., Klaser, A., Schmid, C., & Liu, C.-L. (2011). Action recognition by dense trajectories. In *Proceedings of the 2011 IEEE conference on computer vision and pattern recognition* (pp. 3169–3176). IEEE Computer Society.
Willshaw, D. J., & von der Malsburg, C. (1976). How Patterned Neural Connections Can Be Set Up by Self-Organization. *Proceedings of the Royal Society of London B: Biological Sciences*, *194*(1117), 431–445.
Yamins, D. L. K., & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, *19*(3), 356–365.
Zenke, F., Agnes, E. J., & Gerstner, W. (2015). Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature Communications*, *6*(6922).
Zenke, F., & Gerstner, W. (2017). Hebbian plasticity requires compensatory processes on multiple timescales. *Philosophical Transactions of the Royal Society, Series B (Biological Sciences*, *372:20160259*.
Zenke, F., Gerstner, W., & Ganguli, S. (2017). The temporal paradox of Hebbian learning and homeostatic plasticity. *Current Opinion in Neurobiology*, *43*, 166–176.
Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. CoRR abs/1611.03530.
Zhou, H. H. (1990). CSM: A computational model of cumulative learning. *Machine Learning*, *5*(4), 383–406.