2007 International Symposium on Nonlinear Theory and its Applications
NOLTA'07, Vancouver, Canada, September 16-19, 2007

An Adaptive Cellular Nonlinear Network and its Application

H. Koeppl† and L. O. Chua‡

†School of Computer and Communication Sciences
Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
‡Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA, USA
Email: heinz.koeppl@epfl.ch, chua@eecs.berkeley.edu

Abstract—We propose the extension of the classical framework of Cellular Nonlinear Networks (CNN) to incorporate adaptivity of the cells. Adaptivity means that coupling template coefficients can evolve over time according to some specified rule. Here, the rule is described in terms of a differential equation for each template coefficient. It is proposed that this dynamics can be obtained from the gradient flow of an objective function imposed on the network. The extension is exemplified for a signal processing application, namely the principal subspace analysis (PSA). The application illustrates a top-down approach to self-organization from a global objective to local processing and adaptation rules.

1. Introduction

The CNN framework [1] follows the paradigm, observed in most biological as well as human-made multi-agent systems, that each agent has limited information about the state of the whole collective it is a member of. The agent can sense and interact within a region confined to its vicinity. This appealing concept of locality triggered and holds promise to trigger novel, massive parallel, fault-tolerant engineering architectures and to enhance our mathematical understanding of self-organization. Many of these multi-agent systems furthermore also exhibit adaptivity. That is, the agent can choose from a finite or infinite set of strategies, that modulate its states and interactions. The agent chooses its strategy according to some local criterion that depends on its own state and on the state of its local environment. Examples thereof are, cell differentiation in multicellular organisms, local routing of network traffic, formation change in fish schools and flocks, spike timing dependent plasticity in neuron ensembles, etc. The observation suggests an extension of the classical CNN framework to incorporate this adaptivity. Due to CNN's regular structure it is particularly amenable to mathematical analysis and thus attractive as a model system to study the effects of such adaptive agents. Furthermore, this work provides evidence that this extension also has implications for the design of engineering systems.

The remaining part of the work proceeds as follows. The extension to the classical CNN framework and the necessary notations are introduced in Section 2. Section 3 discusses the considered engineering application, namely principal subspace analysis (PSA). In section 4 we propose the adaptive CNN for PSA, while Section 5 draws the conclusions.

2. The Adaptive CNN

The evolution of an adaptive CNN is governed by two sets of equations, i.e., the processing equations and the adaptation equations. The processing equations represent the standard[1] CNN dynamics for the cell i

$$\dot{x}_i = -\alpha x_i + \sum_{j\in\mathcal{N}(i)} A_{ij} y_j + \sum_{j\in\mathcal{N}(i)} B_{ij} u_j + z_i \quad \text{with} \quad y_i = f(x_i), \tag{1}$$

where i is a linear position index over an arbitrary multi-dimensional array, $\mathcal{N}(i)$ is the set of linear indices associated with the neighboring cells of cell i (and itself), $\alpha \geq 0$ is the self-feedback gain and z_i the bias of the cell i. The function $y_i = f(x_i)$ is the piecewise linear saturation function. The adaptation equations for the coupling connections to cell i can take the general form

$$\begin{aligned} \dot{A}_{ij} &= g_{ij}(A_{i\mathcal{N}(i)}, y_{\mathcal{N}(i)}, u_{\mathcal{N}(i)}, z_i) \\ \dot{B}_{ij} &= h_{ij}(B_{i\mathcal{N}(i)}, y_{\mathcal{N}(i)}, u_{\mathcal{N}(i)}, z_i), \end{aligned} \tag{2}$$

with $y_{\mathcal{N}(i)} = \cup_{j\in\mathcal{N}(i)} y_j$, that is the set of output signals of cells in the neighborhood $\mathcal{N}(i)$ of cell i. Similarly, $A_{i\mathcal{N}(i)} = \cup_{j\in\mathcal{N}(i)} A_{ij}$ refers to the set of coefficients associated with the coupling of the cell i with its neighbors with positions $\mathcal{N}(i)$. Analogous definitions apply to $u_{\mathcal{N}(i)}$ and $B_{i\mathcal{N}(i)}$. The functions g_{ij} and h_{ij} are undetermined functions, that can depend on the position (i, j) of the coupling coefficient. With respect to the daunting degrees of freedom introduced by spatial variant adaptation equations, the case of spatial invariant adaptation equations is more appealing. If we compare this spatial invariant case to the classical spatial invariant CNN framework, it is recognized that the homogeneity has been shifted to a second level. In other words, a spatial *invariant* adaptation dynamics controls the templates A_{ij} and B_{ij} and in general will result in spatial *variant* templates. Except for boundary effects, the subsequent application considers spatial invariant adaptation dynamics. The adaptation equations (2) are general

[1] In what follows we develop the adaptive CNN in terms of the standard CNN [1, p. 16], but it should be clear that the same extension can be made to a more general class of CNNs as defined in [1, p. 5]

but very unspecific. The main part of this work is devoted to obtain an explicit form of these equations for a particular problem.

Note, that in contrast to approaches for template learning of classical CNNs [2], with (2) we impose the constraint that the adaptation dynamics can only take into account local information available to the individual cell. Furthermore, we emphasize the online aspect of the adaptation, where in general templates through (2) and states through (1) co-evolve. Nevertheless, to justify the distinction between processing and adaptation equations, the two equation sets should have different time scales, i.e., the time constant for the adaptation is assumed to be larger than that of the processing.

Most of this work will deal with two-dimensional CNN arrays. In this case the linear position index is chosen to be the columns-wise enumeration of the two-dimensional array. That is, the (β, δ)-position in a $N \times L$ two-dimensional array is associated with the linear index i by $i = (\delta - 1)N + \beta$. We subsequently refer to the column index δ as the stage number of the array. For later purposes let us define two classes of matrices.

Definition 1: A $m \times n$ matrix G_{ij} with $m \leq n$ belongs to the class of band matrices $\mathbb{B}^{m \times n}(k_1, k_2)$ with bandwidth $b = k_1 + k_2 + 1$ if $G_{ij} = 0$ for $j < i - k_1$ or $j > i + k_2$ with $k_1, k_2 \geq 0$.

Definition 2: The class of tridiagonal square matrices $\mathbb{T}^{n \times n}$ is defined as $\mathbb{T}^{n \times n} = \mathbb{B}^{n \times n}(1, 1)$.

Interpreting the coupling coefficients A_{ij} and B_{ij} as elements of matrices, we can rewrite (1) in matrix notation as

$$\dot{\mathbf{x}} = -\alpha\mathbf{x} + \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} + \mathbf{z} \quad \text{with} \quad y_i = f(x_i), \tag{3}$$

where for $j \notin \mathcal{N}(i)$ the elements of A_{ij} with $i = 1, \ldots, NL$, $j = 1, \ldots, NL$ were set to zero. The same holds for B_{ij}. Assuming nearest neighbor coupling and von Neumann or Dirichlet boundary conditions the matrices $\mathbf{A}$ and $\mathbf{B}$ have block-tridiagonal structure. This is subsequently exemplified for $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} \mathbf{C}_1 & \mathbf{N}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}_2 & \mathbf{C}_2 & \mathbf{N}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_3 & \mathbf{C}_3 & \mathbf{N}_3 & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_L & \mathbf{C}_L \end{pmatrix}, \tag{4}$$

where the submatrices $\mathbf{P}_k$, $\mathbf{C}_k$ and $\mathbf{N}_k$ refer to the coupling matrices at stage k with the previous $(k-1)$, the current (k) and the next $(k+1)$ stage, respectively. All of those submatrices have once again tridiagonal structures, assuming von Neumann or Dirichlet boundary conditions. Thus, for a $N \times L$ two-dimensional array we have $\mathbf{P}_k \in \mathbb{T}^{N \times N}$ with $k = 2, \ldots, L$, $\mathbf{C}_k \in \mathbb{T}^{N \times N}$ with $k = 1, \ldots, L$ and $\mathbf{N}_k \in \mathbb{T}^{N \times N}$ with $k = 1, \ldots, L-1$. For the case of periodic boundary conditions the Töplitz-like structure of (4) and its submatrices has to be replaced by the corresponding circulant-like structure.

3. Application to PSA

PSA and its variant principal component analysis (PCA) [3] are widely used linear signal processing operations for signal and image compression, data pre-processing, noise removal, feature extraction, etc. Roughly, the PSA operation involves the projection of a multivariate random process $\mathbf{u} = (u_1, \ldots, u_N)^T$ to a lower M-dimensional subspace, spanned by those eigenvectors $\mathbf{v}_k$ of the covariance matrix $\mathbf{\Phi} \equiv \mathsf{E}(\mathbf{u}\mathbf{u}^T)$, that are associated with the M largest eigenvalues of $\mathbf{\Phi}$.

3.1. Processing and Adaptation

Linear feedforward neural networks have been used to implement PSA, where the eigenvector decomposition of $\mathbf{\Phi}$ is performed without a direct estimation of $\mathbf{\Phi}$. The proposed network is based on two seminal works. Firstly on [4], that introduced a bi-directional symmetric coupling in a feedforward neural network. Secondly on [5], that extended the isospectral Toda flow to solve the PSA problem. We extend these works by considering a fully recursive system and allow only for local coupling between all cells. Furthermore, the obtained adaptation dynamics does not involve any approximation. With that we propose the following processing structure

$$\begin{aligned} \dot{\mathbf{x}}_1 &= -\alpha\mathbf{x}_1 + \mathbf{W}_1\mathbf{x}_2 + \alpha^2\mathbf{u} \\ \dot{\mathbf{x}}_2 &= -\alpha\mathbf{x}_2 + \mathbf{W}_2\mathbf{x}_3 + \mathbf{W}_1^T\mathbf{x}_1 \\ &\vdots \\ \dot{\mathbf{x}}_L &= -\alpha\mathbf{x}_L + \mathbf{W}_{L-1}^T\mathbf{x}_{L-1}, \end{aligned} \tag{5}$$

with $\mathbf{W}_1^T \in \mathbb{B}^{M \times N}(0, l)$ with $l \geq N - M$ and $\mathbf{W}_k \in \mathbb{T}^{M \times M}$ with $k = 2, \ldots, L-1$. The constraint on l guarantees full rank of $\mathbf{W}_1$. The network topology of (5) is indicated in Fig. 1. According to the above time-scale considerations, we assume now that the processing equations (5) are in steady state. Furthermore, we assume stability of (5). Thus, the network instantaneously maps the process sample $\mathbf{u}$ to a lower M-dimensional encoding state. This state is $\mathbf{x}_2$ and the encoder is thus the matrix $\mathbf{E} : \mathbf{u} \mapsto \mathbf{x}_2$. Let us further define the steady state propagators $\mathbf{P}_k : \mathbf{u} \mapsto \mathbf{x}_k$. They can be written as

$$\mathbf{P}_k = \alpha^{-(k-2)}\mathbf{S}_k^{-1}\mathbf{W}_{k-1}^T \cdots \mathbf{S}_2^{-1}\mathbf{W}_1^T\,\mathbf{S}_1^{-1}, \tag{6}$$

for $k = 2, \ldots, L$, with the backward recursion for

$$\mathbf{S}_{j-1} = \mathbf{I} - \alpha^{-2}\mathbf{W}_{j-1}\mathbf{S}_j^{-1}\mathbf{W}_{j-1}^T, \tag{7}$$

with $j = 2, \ldots, L$ and $\mathbf{S}_L = \mathbf{I}$. For $\mathbf{P}_2 = \mathbf{E}$ we find the alternative form

$$\mathbf{P}_2 = (\mathbf{S}_2 - \alpha^{-2}\mathbf{W}_1^T\mathbf{W}_1)^{-1}\mathbf{W}_1^T. \tag{8}$$

We can now define the global objective of the network as

$$\min_{\mathbf{W}_1, \ldots, \mathbf{W}_{L-1}} J(\mathbf{W}_1, \ldots, \mathbf{W}_{L-1}) \tag{9}$$

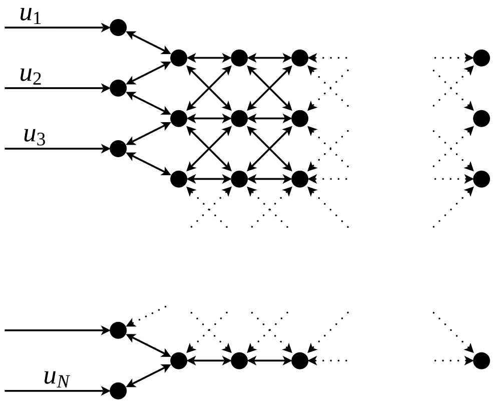

Figure 1: General topology of the proposed L-stage locally coupled PSA network for arbitrary N and $M = N - 1$.

with

$$J(\mathbf{W}_1, \ldots, \mathbf{W}_{L-1}) \equiv \tfrac{1}{2}\, \mathsf{E}\, \|\mathbf{u} - \mathbf{W}_1\mathbf{E}\mathbf{u}\|_2^2. \tag{10}$$

The vector $\boldsymbol{\nu} \equiv \mathbf{W}_1\mathbf{E}\mathbf{u}$ gives the network reconstruction of $\mathbf{u}$ from the state $\mathbf{x}_2$. Thus, $\mathbf{W}_1$ is the decoder matrix. The solution to (9) gives the network configuration resulting in an encoding $\mathbf{x}_2$ that minimizes the expected ℓ_2-norm between $\mathbf{u}$ and its reconstruction $\boldsymbol{\nu}$. Thus inevitable, the subspace spanned by the optimal $\mathbf{x}_2$ is identical to the PSA subspace. Computing the gradient of (10) with respect to $\mathbf{W}_k$ we obtain the exact gradient flow

$$\begin{aligned} \dot{\mathbf{W}}_1 &= \mathbf{J}\mathbf{U}\mathbf{P}_2^T \\ \dot{\mathbf{W}}_k &= \alpha^{-1}\mathbf{P}_k\mathbf{U}\mathbf{P}_{k+1}^T \quad \text{for} \quad k = 2, \ldots, L-1, \end{aligned} \tag{11}$$

with

$$\mathbf{U} \equiv 2\boldsymbol{\Phi} - \mathbf{W}_1\mathbf{E}\boldsymbol{\Phi} - \boldsymbol{\Phi}\mathbf{W}_1\mathbf{E} \tag{12}$$

$$\mathbf{J} \equiv \mathbf{I} + \alpha^{-2}\mathbf{W}_1\mathbf{E}. \tag{13}$$

For $\mathbf{E}\mathbf{W}_1 \underline{\rightarrow} \mathbf{I}$ one arrives at the simplified flow by replacing $\mathbf{U} \rightarrow \bar{\mathbf{U}} \equiv \boldsymbol{\Phi} - \mathbf{W}_1\mathbf{E}\boldsymbol{\Phi}$ in (11). The resulting dynamics can be associated with the flow in [5]. These averaged dynamics (11), involving the covariance matrix $\boldsymbol{\Phi}$, can be approximated through stochastic approximation [6, pp. 144] $\boldsymbol{\Phi} \approx \mathbf{u}\mathbf{u}^T$. Assuming positive definiteness of $\mathbf{J}$ in (11), that can be ensured by an appropriate choice of α, $\mathbf{J}$ can be replaced by $\mathbf{I}$ without changing the stable equilibrium points of the dynamics (11). With this we arrive at the simple stochastic learning dynamics

$$\begin{aligned} \dot{\mathbf{W}}_1 &= \mu(\mathbf{u} - \boldsymbol{\nu})\mathbf{x}_2^T + \mu\mathbf{u}(\mathbf{x}_2 - \boldsymbol{\xi}_2)^T \\ \dot{\mathbf{W}}_k &= \gamma(\mathbf{x}_k - \boldsymbol{\xi}_k)\mathbf{x}_{k+1}^T + \gamma\mathbf{x}_k(\mathbf{x}_{k+1} - \boldsymbol{\xi}_{k+1})^T, \end{aligned} \tag{14}$$

for $k = 2, \ldots, L-1$ with $\boldsymbol{\xi}_k \equiv \mathbf{P}_k\boldsymbol{\nu}$ and μ denotes a small adaptation rate consistent with the stochastic approximation and $\gamma \equiv \mu\alpha^{-1}$. The correspond stochastic dynamics for the approximate flow with $\bar{\mathbf{U}}$ reads

$$\begin{aligned} \dot{\mathbf{W}}_1 &= \mu(\mathbf{u} - \boldsymbol{\nu})\mathbf{x}_2^T \\ \dot{\mathbf{W}}_k &= \gamma(\mathbf{x}_k - \boldsymbol{\xi}_k)\mathbf{x}_{k+1}^T. \end{aligned} \tag{15}$$

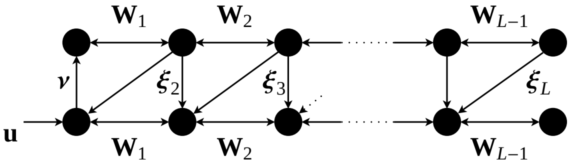

Figure 2: Topology corresponding to (14) with an signal propagation layer and a reconstruction propagation layer.

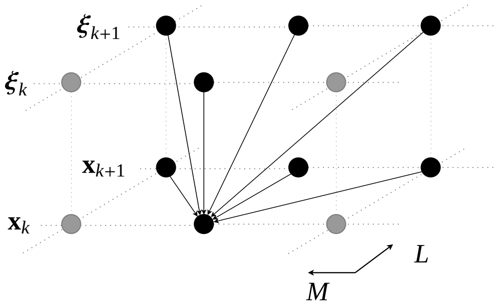

Figure 3: Neighborhood of a single cell, that is required to compute the learning dynamics (14) for the coupling coefficients of the cell.

Both sets of equations have an interesting structure. They involve the difference between the input $\mathbf{u}$ and the network reconstruction $\boldsymbol{\nu}$ and correspondingly the difference between the network states $\mathbf{x}_k$ and some novel states $\boldsymbol{\xi}_k$. From their definition we see, that the latter states would arise if we instead of $\mathbf{u}$ apply $\boldsymbol{\nu}$ to the network. The key to construct a local adaptation dynamics is to introduce a second, overlayed network in addition to the existing processing network. The resulting topology for (14) is shown in Fig. 2. The neighborhood for a single cell in this two-layered network for (14) is indicated in Fig. 3. For the simplified dynamics (15) the number of neighbors reduces to four.

3.2. Stability Analysis

Here we investigate the stability of the nonlinear matrix differential equation (11) and its approximation using $\bar{\mathbf{U}}$, that describe the average dynamics of (14) and (15), respectively. The dynamics (11) performs a exact gradient descend on the positive definite objective (10). Thus, the objective function can be utilized as a Lyapunov function if we can show that the time-derivative of (10) along the trajectory $\mathbf{W}_k(t)$ is always negative. This holds true for all gradient systems and in particular we have

$$\frac{\partial J}{\partial t} = -\sum_{k=1}^{L-1} \operatorname{Tr}\left[\left(\frac{\partial \mathbf{W}_k}{\partial t}\right)^T \frac{\partial \mathbf{W}_k}{\partial t}\right] < 0. \tag{16}$$

In the following we show that (10) is also a local Lyapunov function for the simplified matrix flow with $\bar{\mathbf{U}} = \mathbf{U} - \boldsymbol{\Phi}(\mathbf{I} - \mathbf{W}_1\mathbf{E})$. The local region in state space, where (10) is a Lyapunov function for the simplified flow is defined as $\mathcal{W} \equiv \{\{\mathbf{W}_1^T \in \mathbb{B}^{M\times N}(0, l), \mathbf{W}_2 \in \mathbb{T}^{M\times M}, \ldots, \mathbf{W}_{L-1} \in \mathbb{T}^{M\times M}\} \mid \mathbf{E}\mathbf{W}_1 = \mathbf{I}\}$, with $l \geq N - M$. For this to make sense we have to show that a trajectory starting in the set $\mathcal{W}$ will always stay within that set. Let us define the trajectory as the set of coupling matrices that are solutions to the approximate flow as $\mathbf{W}(t) \equiv \{\mathbf{W}_1^T(t) \in \mathbb{B}^{M\times N}(0,l), \mathbf{W}_2(t) \in \mathbb{T}^{M\times M}, \ldots, \mathbf{W}_{L-1}(t) \in \mathbb{T}^{M\times M}\}$ with $t \in [0,\infty)$ and $l \geq N-M$. After some algebra one obtains that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{E}(t)\mathbf{W}_1(t)\right] = 0 \quad \text{for} \quad \mathbf{W}(0) \in \mathcal{W}, \tag{17}$$

Thus, with $\mathbf{W}(0) \in \mathcal{W}$ the trajectory stay within that region $\mathbf{W}(t) \in \mathcal{W}$ for all times t. For the positive definite (10) to be a local Lyapunov function in $\mathcal{W}$ it remains to show that it decays along the trajectory. We find that

$$\frac{\partial J}{\partial t} = -\operatorname{Tr}\left[(\mathbf{J}\bar{\mathbf{U}}\mathbf{E}^T)^T(\mathbf{J}\bar{\mathbf{U}}\mathbf{E}^T)\right] < 0 \quad \text{for} \quad \mathbf{W}(t) \in \mathcal{W}.$$

4. The Subspace CNN

We now interpret the obtained dynamical system of Section 3 in term of the adaptive CNN framework. The proposed adaptive CNN has two planar layers that are coupled. One layer propagates the input signal $\mathbf{u}$, while the other layer propagates the reconstructed signal $\boldsymbol{\nu}$. Both layers have the identical coupling coefficients. Thus, with the corresponding feedback matrix

$$\mathbf{A} = \begin{pmatrix} -\alpha\mathbf{I} & \mathbf{W}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{W}_1^T & -\alpha\mathbf{I} & \mathbf{W}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2^T & -\alpha\mathbf{I} & \mathbf{W}_3 & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{W}_{L-1}^T & -\alpha\mathbf{I} \end{pmatrix}, \tag{18}$$

we obtain the processing equations as

$$\begin{aligned} \dot{\mathbf{x}} &= -\alpha\mathbf{x} + \mathbf{A}\mathbf{x} + \mathbf{B}\bar{\mathbf{u}}, \qquad & \mathbf{x}(0) &= \mathbf{x}_0 \\ \dot{\boldsymbol{\xi}} &= -\alpha\boldsymbol{\xi} + \mathbf{A}\boldsymbol{\xi} + \mathbf{B}\bar{\boldsymbol{\nu}}, \qquad & \boldsymbol{\xi}(0) &= \boldsymbol{\xi}_0, \end{aligned} \tag{19}$$

with $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_L^T)^T$ and $\boldsymbol{\xi} = (\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \ldots, \boldsymbol{\xi}_L^T)^T$ and with $\bar{\mathbf{u}} = (\mathbf{u}^T, \mathbf{0})^T$, $\bar{\boldsymbol{\nu}} = (\boldsymbol{\nu}^T, \mathbf{0})^T$, $\boldsymbol{\nu} = \mathbf{W}_1\mathbf{x}_2$ where the vector $\mathbf{0}$ is of dimension $1 \times M(L-1)$. As PSA is a linear transformation, the resulting cellular network (19) is linear. Alternatively, it can be interpreted as a CNN operated in the linear region of $y_i = f(x_i)$. Furthermore we have

$$\mathbf{B} = \begin{pmatrix} \alpha^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{20}$$

where $\mathbf{I}$ is the $N \times N$ unit matrix. Concerning the adaptation equations for the coupling coefficients, we observe that for the proposed processing topology of Fig. 1 for stage $1 < k < L$, each cell has to adapt three coupling coefficients (apart from boundary effects). These are the coupling coefficient toward the next stage. For the linear indexing scheme, the three indices for the neighboring cell in the consecutive stage can be obtained by adding the stage dimension M, $M-1$ and $M+1$ to the linear index position of the cell that we consider (except boundary effects). Thus, in addition to the signal and reconstruction propagation equation (19) we obtain the nonlinear coupling adaptation equation for cell i in the signal propagation layer as

$$\dot{A}_{ij} = \gamma(x_i - \xi_i)x_j + \gamma x_i(x_j - \xi_j), \tag{21}$$

with $j \in \mathcal{N}(i) = \{i+M-1, i+M, i+M+1\}$. In the same way the coupling adaptation equation based on the approximate flow (15) reads

$$\dot{A}_{ij} = \gamma(x_i - \xi_i)x_j, \tag{22}$$

with $j \in \mathcal{N}(i) = \{i+M-1, i+M, i+M+1\}$. For the stage $k = 1$ and thus for $i \leq N$ the dynamics are $\dot{A}_{ij} = \mu(\bar{u}_i - \bar{\nu}_i)x_j + \mu\bar{u}_i(x_j - \xi_j)$ and $\dot{A}_{ij} = \mu(\bar{u}_i - \bar{\nu}_i)x_j$ with $j \in \mathcal{N}(i)$, for (14) and (15), respectively. The neighbor positions $\mathcal{N}(i)$ for $i \leq N$ of size $l+1$ can be read off from $\mathbf{W}_1^T \in \mathbb{B}^{M\times N}(0,l)$.

5. Conclusion

We proposed an extension of the classical CNN framework to allow for adaptivity of the processing cell and showed a first application of such a network to PSA. The local adaptation rules are derived from a gradient flow on a global objective that is a function of all states and inputs of the network. The new framework can serve as a model system to understand self-organizing processes and to design novel robust, fault-tolerant engineering systems. Interpreting a CNN as a lattice approximation of a continuous medium, the adaptive extension would correspond to a *smart* medium, that would change its spatial inhomogeneity according to external stimuli. Open issues involve the design of nonconservative local tests for stability of the processing equations in the discussed application and in adaptive CNNs in general.

References

[1] L. O. Chua, *CNN: A Paradigm for Complexity*, ser. World Scientific Series on Nonlinear Science. World Scientific, 1998.

[2] T. Kozek, T. Roska, and L. O. Chua, "Genetic algorithm for CNN template learning," *IEEE Transactions on Circuits and Systems–I*, vol. 40, no. 6, pp. 392–402, 1993.

[3] R. O. Duda, P. E. Hart, and S. D. G., *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[4] L. Xu, "Least mean square error reconstruction principle for self-organizing neural networks," *Neural Networks*, vol. 6, pp. 627–648, 1993.

[5] R. Brockett, "Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems," *Linear Algebra and its Applications*, vol. 146, pp. 79–91, 1991.

[6] L. Ljung and T. Söderstrom, *Theory and Practice of Recursive Identification*. The MIT Press, Cambridge, MA, 1983.