Learning wave phenomena on the CNN universal machine

Samuel Xavier-de-Souza, Johan A.K. Suykens, Joos Vandewalle

K.U. Leuven, ESAT-SCD-SISTA
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee) Belgium
e-mail: samuel.xavierdesouza@esat.kuleuven.be

Abstract— In recent years it was discovered that cellular neural networks with local and space-invariant connections are able to generate a wide range of two-dimensional spatiotemporal behavior. Many of these dynamics can be directly mapped into natural phenomena occurring in physics, chemistry, and biology. These mappings make cellular neural network a suitable tool for modeling and simulation of such phenomena. With the advent of advanced VLSI implementations of this network and its inherent parallelism, simulations can be executed on-chip in a fraction of the time that would be necessary with actual digital computer implementations. In this work we introduce a methodology for the learning of this kind of dynamics. The problem is treated as an optimization problem and is based on trajectory learning for recurrent neural networks. In order to adapt this to the learning of two-dimensional dynamics, we proposed a cost function which can incorporate time instants into the set of variables to be optimized. As a result it can be observed that the network can also learn any frequency modulation of the original dynamics. Besides simulation, the proposed methodology can also be applied directly to a VLSI implementation of the network. Experiments were performed for the spiral autowave.

1. Introduction

Recently, the existence of a variety of complex phenomena have been discovered to exist in Cellular Neural Networks (CNN) systems [1], which have proven to be very good tools for modeling and simulation. This, and the growing interest and success in implementing CNNs on silicon, have contributed to enforce the new paradigm of *active wave computing* [2][3]. This paradigm allows for silicon based implementations of algorithms for robot navigation, artificial retina, finger print enhancement, etc. Among these complex phenomena, two and higher dimensional spatiotemporal behavior are of special importance. They appear in diverse fields of physics, chemistry, and biology, and can be observed in an active medium which can be modeled by arrays of coupled non-linear circuits like CNN cells. Autowaves, spiral waves, traveling waves and many other of these phenomena have been reported to emerge from an active medium consisting of an array of CNN cells. This paper describes the methodology proposed in [4] to systematically learn such phenomena on CNNs systems.

The methodology to train spatiotemporal behavior with CNNs is based on trajectory learning. The novelty lies in the derivation of a cost function that also assimilates the time instants into the set of parameters to be optimized allows for the learning of the desired behavior with different speeds rather than restricting it to the original speed of the dynamics. Besides the positive effect that this feature may have for increasing the speed of existing applications, it also reduces the necessity of generating a perfect training set, which is certainly the most important issue for the learning of 2-dimensional spatiotemporal behavior on CNNs.

2. Trajectory Learning and Cellular Neural Networks

The mapping of trajectory learning with Recurrent Neural Networks (RNN) [5] into learning of spatiotemporal behavior with CNNs is straightforward. One only needs to consider the equivalence between output neurons and output cells. Although trajectory learning becomes increasingly complicated with the number of neurons N in the RNN as the number of weights increases quadratically, by assuming zero elements in the weight matrices, the computational burden can be reduced. This is essentially what happens with CNNs. More precisely, only weights of neighboring cells are taken into account with the remaining values of the weight matrices equal to zero. Moreover, CNNs weights are often also space invariant. These weights, called CNN templates, are local and invariant equivalents of the weights in a RNN. Therefore, the description of a first order CNN system with local and invariant weights can be reduced to the description of the behavior of a single cell/neuron:

$$\frac{dx_{i,j}}{dt} = -x_{i,j}(t) + Ay_{i,j} + Bu_{i,j} + z, \qquad (1)$$

with A and B being the local and invariant weights, z denoting the bias term, and i, j, the indices of the given cell in a regular grid.

Other models that are especially important for modeling of complex behavior are multi-layer second order CNN systems. Their dynamics can be represented by the following equations:

$$\frac{dx_{i,j;1}}{dt} = -x_{i,j;1}(t) + A_{1,1}y_{i,j;1} + A_{1,2}y_{i,j;2} + B_1u_{i,j;1} + z_1$$

$$\frac{dx_{i,j;2}}{dt} = -x_{i,j;2}(t) + A_{2,2}y_{i,j;2} + A_{2,1}y_{i,j;1} + B_2u_{i,j;2} + z_2$$

where the index after the semi-column identifies the layer and the indices i, j locate the given cell within the layer.

Locality and invariance are the main advantages of CNNs. Despite the condensed set of values that describe these systems, a large variety of dynamical phenomena can be observed. Moreover, locality and invariance alone make these systems very suitable for VLSI implementation which is a trend that has been emerging in the past years. Nowadays, high end silicon versions of CNN Universal Machines (CNN-UM) are already available for the development of image processing application at extremely high speed [6].

3. Spatiotemporal Learning with Cellular Neural Networks

The fact that the training of CNN systems is reduced to the learning of the matrices A, B, and the bias z makes the process much simpler. Nevertheless, in spite of being very efficient and widely used, gradient descent techniques are hard to be applied in this case because of the non-linear output function $y = f(x)$ that often assumes the piecewise linear non-differentiable form $f(x) = \frac{1}{2}(|x+1| - |x-1|)$ in CNN models. This complicates the derivation of an analytical form for the gradient of the cost function and would require techniques from non-differentiable optimization. Although gradient methods are more efficient to solve convex and local optimization problems, there are no guarantees about the global optimality of the task to be learned for the problem considered here. Global optimization methods can be used here with reasonable confidence since the number of parameters to be optimized is not very high. Additionally, these methods are not limited to convex problems and can do surprisingly well on problems with many local minima.

Another point that needs to be carefully addressed is the generation of training sets. While a trajectory can be described by a sequence of values for every output, e.g. $\sin(t)$, spatiotemporal behavior in a grid of cells needs to be described by a continuous 2-D image sequence where the values of the pixels correspond to the output of a single cell. Due to the couplings between cells, the desired values of every individual pixel trajectory can not be derived independently and need to be considered as a whole. As a consequence, the images in a training set may represent snapshots of the desired system's output which are irregular in time. In order to avoid the necessity of a strict match between a irregular time evolution of the desired and resulting behavior, a new cost function is introduced. This cost function assimilates the time intervals between snapshots of the system's output into the set of optimization parameters. The problem of learning spatiotemporal behavior with CNNs can be presented in this way as

$$\min_{A,B,z,\Delta t_k,\Delta t_{k+1},\cdots,\Delta t_T} \left[E = \sum_{i,j} \sum_{k=1}^{T} (y^d_{i,j;k} - y_{i,j}(A,B,z,t_k))^2 \right], \tag{2}$$

where $\Delta t_k = t_k - t_{k-1}\ \forall\ k = 0, \cdots, T$, representing the time interval between two output samples with T being the finite number of samples and $t_0 = 0$. $y^d_{i,j;k}$ denotes the desired output value of a pixel in the kth image of a given sequence of T images. The initial conditions $x(0)$ are also assumed to be given. The value $y_{i,j}(A,B,z,t_k)$ denotes the output value of a pixel as the system has evolved to the time instant t_k with weight matrices A and B, and bias z.

Observe that the desired outputs $y^d_{i,j;k}$ do not depend on time but rather on the index k of time instants whereas the obtained outputs $y_{i,j}(A,B,z,t_k)$ do depend on the time instants t_k but these are also parameters to be optimized. A way to exemplify the results of such measure in view of trajectory learning is to think that if the trajectory to be trained is e.g. $\sin(t)$, the resulting trajectory is allowed to be any approximation of $\sin(\omega t)$, i.e. the desired trajectory becomes not only $\sin(t)$ but also any frequency modulation of it. Additionally, also observe that Δt_k is the parameter to be optimized rather than t_k itself. In this case all the desired time instants t^d_k are irrelevant, only the order in which the images are learned matters, not at which specific time instants the images are generated by the CNN. This is also the reason why the error term $(y^d_{i,j;k} - y_{i,j}(A,B,z,t_k))^2$ is summed over all k samples rather than integrated over time the interval $[0, t_T]$. The right choice for the value of T will depend on how difficult the learning problem is and on how much time and processing resources are available.

The relaxation of the schedule for the resulting spatiotemporal behavior is an important issue for CNNs. Fixing a rigid schedule for the spatiotemporal trajectory to be learned would magnify the importance of a physically feasible training set and this needs to be avoided for a simple reason: in many cases nothing can guarantee that suggested time stamps for the desired spatiotemporal trajectory have a fixed relation between themselves in a real system.

The training of complex behavior like autowaves can be given an extra degree of freedom which concerns the number of CNN layers involved. In a 2-layer CNN for example, autowaves occur simultaneously in both layers but often with different waveforms. Frequently the outcome of what happens in one layer is sufficient for some applications and in this case only the output of this layer needs to be taken into account for the calculation of the cost function. However, besides the cases where the outcome of both layers is important, the inclusion of the output of the second layer in the cost function calculation can sometimes bring more insight about the location of a globally optimal solution. For this case it is only necessary to include a sum over the number of layers in Eq. (2), and an index for the layers in the output and desired output values.

4. A Learning Example of Spatiotemporal Behavior

A very interesting class of spatiotemporal behavior is autowaves. The term autowave is an abbreviation of ”autonomous wave” commonly used to characterize self-sustained signals that induce a local release of stored energy in an active medium, and use this energy to trigger the same process in neighbor regions. Many waves occurring in nature share the same properties. Typical examples include: waves in the cerebral cortex, epidemic waves, combustion waves, reaction-diffusion processes, etc. In cellular neural networks, autowaves represent the means for the new paradigm of active wave computing and have been used for example to guide robots along obstacles towards a target [7]. An example of autowaves known as spiral wave is shown in Fig. 1.

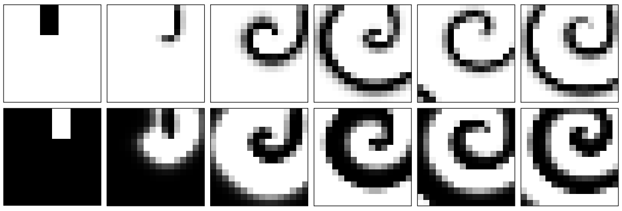

Figure 1: Spiral autowave phenomena in a 2-layer CNN simulator: initial conditions and time snapshots of the output in the two layers. Each row represents one layer.

In CNN systems, autowaves have been observed to emerge in 2-D arrays of second or higher order cells [8, 9] and delayed type first order cells [10]. Due to the fact that the arrays of regular first order cells can not generate the necessary active local dynamics, autowaves can not be observed in these systems. In this paper the model chosen to produce autowaves is a 2-layer array of second order CNN cells described by Eq. (2).

In order to train a spiral wave like the one of Fig. 1, a few issues need to be addressed. The optimization method used to minimize the cost function in Eq. (2) was Adaptive Simulated Annealing (ASA) [11]. This method has proved to be efficient for tuning fixed output templates for VLSI implementations [12]. It can be observed that if one makes $T = 1$ in Eq. (2) and t_1 is removed from the optimization and made sufficiently long, this cost function is reduced to the fixed output case. The same approach of relaxation of constraints and search boundaries used in [12] can also be made useful for learning in the following way: (a) in the beginning of the learning process no limits are imposed to the weight values and thus the maximum range of values are available; (b) after this process converges, better solutions are obtained by limiting the weight values to values that are close to the first solution and/or incrementally relaxing existing constraints, e.g. symmetry, non-zero values, etc; (c) the last step is then repeated until any stopping criteria is reached.

Only immediate neighbor cells are assumed to have a non-zero weight, which means that A and $B \in \mathbb{R}^{3\times3}$. However, if prior knowledge about the template matrices is known, the number of parameters that actually needs to be optimized in A and B can be reduced, e.g. with symmetry this number can go from 9 to 5 parameters per matrix. Without any prior knowledge, the number of parameters to be optimized are the values of a full template plus the number of time intervals T. For simplification, the input images in Eq. (1) will be set to zero, and therefore the input weight matrix B will be assumed zero.

Initial conditions are another important aspect to consider when trying to generate autowaves. Here the initial condition of the first layer will be given as an image and the image for the second layer it will be assumed to be the same image of the first layer inverted and shifted one or two pixels in the direction of the desired propagation. Observe that, this is not the only way for generating initial conditions for autowaves [13] but it is certainly a simple one.

The images in Fig. 1 were produced by simulation and used as training set for experiment with autowaves. The results of the learning process are shown in Fig. 2. The training was performed in a simulator although the same procedure could be used to train a CACE1k CNN-UM chip [14]. Although the images for the training set and the learning itself were generated with the same simulator, the original and the resulting template are not the same. Observe that there was a good generalization of outputs further in time that were not used for the learning process. Another important remark to be made w.r.t. Fig. 2 is that the speed of the template execution changed. The speed of the evolution of the learned behavior is approximately 20% faster than the original one although no measures were taken with respect to this. The difference in speed and a clear sight of the generalization of the learned template can be seen in Fig. 3, where original and trained templates can be seen in a larger grid of cells. Observing how the speed changed, one may assume that the learned template could also have randomly generated a behavior which was slower or even faster. Hence changes in speed of a given CNN operation are possible, without modifying time constants.

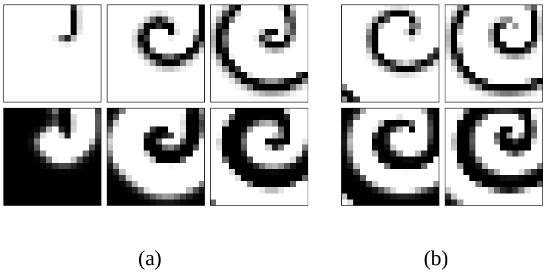

Figure 2: Spiral autowave that was learned from the three first columns of Fig. 1. (a) Trained behavior; and (b) generalized behavior further in time.

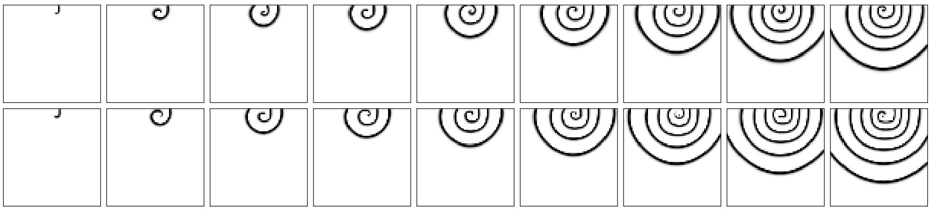

Figure 3: Evolution of a 64x64 CNN grid for the original (row 1) and trained (row 2) templates. The columns represent snapshots taken in equivalent time instants. Generalization further in time and change in speed can be clearly seen.

5. Conclusion

In this paper the problem of learning 2-dimensional spatiotemporal behavior with cellular neural networks was approached. Due to the locality and space invariance of CNN weights, the number of parameters to be optimized in these networks is not large. The learning is thus considerably easier favoring the use of global optimization methods to avoid the need of an expression for the gradient of the cost function. The generation of an efficient training set for the CNN problem is not straightforward as in classical trajectory learning and thus customized solutions need to be devised. Therefore, a new cost function and methodology was introduced for learning of spatiotemporal behavior. This cost function assimilates time intervals also as parameters to be optimized and with that also reduce the importance of generating perfect training sets. This cost function also allows for the learning of the desired behavior at arbitrary speed of the dynamics. From an learning example it was possible to show that the CNN could learn and generalize the task at hand. Another important feature shown in this example was the possibility to modify the speed of an operation. A deeper exploration of this feature is ongoing research but it is believed that existing CNN applications can benefit from faster execution by pushing the template operations to its speed limit.

Acknowledgments Research supported by: • Research Council KUL: GOA-Mefisto 666, GOA-AMBioRICS, BOF OT/03/12, several PhD/postdoc & fellow grants; • Flemish Government: ○ FWO: PhD/postdoc grants, G.0407.02, G.0080.01, G.0211.05, G.0499.04, G.0226.06, research communities (ICCoS, ANMMM); ○ IWT: PhD Grants, ○ Tournesol 2005 • Belgian Federal Science Policy Office IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling'). J. Suykens and J. Vandewalle are an associate and full professor with K.U. Leuven Belgium, respectively.

References

[1] M. E. Yalcin, J.A.K. Suykens, and J.P. Vandewalle. *Cellular Neural Networks, Multi-Scroll Chaos and Syncronization*, volume 50 of *Nonlinear Science/A*. World Scientific, 2005.

[2] B. Shi, P. Arena, and A. Zarándy, editors. *IEEE Trans. on Circuits and Systems-I - Special issue on CNN technology and active wave computing*, volume 51 (5). May 2004.

[3] T. Roska. Computational and computer complexity of analogic cellular wave computers. *J. Circuits, Systems, and Computers*, 12(4):539–56, 2003.

[4] S. Xavier-de-Souza, J. A. K. Suykens, and J. Vandewalle. Learning of Spatiotemporal Behavior in Cellular Neural Networks. *International Journal of Circuit Theory and Applications*, 2005. Submitted for publication.

[5] B. A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks - a survey. *IEEE Trans. Neural Networks*, 6(5):1212–1228, Sep 1995.

[6] L. O. Chua, T. Roska, T. Kozek, and Á. Zardy. CNN Universal chips crank up computing power. *IEEE Circuits and Devices*, 12(4):18–28, 1996.

[7] A. Adamatzky, P. Arena, A. Basile, R. Carmona-Galan, B.D.L. Costello, L. Fortuna, M. Frasca, and A. Rodriguez-Vazquez. Reaction-diffusion navigation robot control: from chemical to VLSI analogic processors. *IEEE Trans. on Circuits and Systems-I*, 51(5):926–938, May 2004.

[8] V. Pérez-Muñuzuri, V. Pérez-Villar, and L. O. Chua. Autowaves for image processing on a two-dimensional cnn array of excitable nonlinear circuits: flat and wrinkled labyrinths. *IEEE Trans. on Circuits and Systems-I*, 40(3):174–181, Mar 1993.

[9] A.P. Munuzuri, V. Perez-Munuzuri, M. Gomez-Gesteira, L.O. Chua, and V. Perez-Villar. Spatiotemporal structures in discretely-coupled arrays of nonlinear circuits: a review. *Int. J. of Bifurcation and Chaos*, 5(1):17–50, Feb 1995.

[10] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer. Simulating nonlinear waves and partial differential equations via CNN—part I: Basic techniques. *IEEE Trans. on Circuits and Systems-I*, 42(10):807–815, Oct 1995.

[11] L. Ingber. Adaptive simulated annealing (ASA): lessons learned. *J. Control and Cybernetics*, 25(1):33–54, 1996.

[12] S. Xavier-de-Souza, M. E. Yalcin, J. A. K. Suykens, and J. Vandewalle. Toward CNN Chip-specific robustness. *IEEE Trans. on Circuits and Systems-I*, 51(5):892–902, May 2004.

[13] P. Arena, S. Baglio, L. Fortuna, and G. Manganaro. Self-organization in a two-layer CNN. *IEEE Trans. on Circuits and Systems-I*, 45(2):157–162, Feb 1998.

[14] R. Carmona-Galan, F. Jimenez-Garrido, C.M. Dominguez-Mata, R. Dominguez-Castro, S.E. Meana, I. Petras, and A. Rodriguez-Vazquez. Second-order neural core for bioinspired focal-plane dynamic image processing in cmos. *IEEE Trans. on Circuits and Systems-I*, 51(5):913–925, May 2004.