Deep Convolutional Neural Networks for Chaos Identification in Signal Processing

Andrey V. Makarenko
Institute of Control Sciences of Russian Academy Sciences, Moscow, Russia
Constructive Cybernetics Research Group, Moscow, Russia
avm.science@mail.ru

***Abstract*—This paper demonstrates effective capabilities of a relatively simple deep convolutional neural network in estimating the Lyapunov exponent and detecting chaotic signals. A major difference between this study and existing research is that our networks take raw data as input, automatically generate a selection of informative features, make a direct estimation of the Lyapunov exponent and form a decision whether a chaotic signal is present. The proposed method does not require attractor reconstruction. It also can be used for processing relatively short signals – in the experiment described here the signal length is 1024 sequence elements. The study has demonstrated that deep convolutional neural networks are effective in applications involving chaotic signals (down to narrowband or broadband stochastic processes), as well as distinct patterns, and can, therefore, be used for a number of signal processing tasks.**

***Index Terms*—deep convolutional neural networks, chaos identification, Lyapunov exponent, time series, logistic map**

I. Introduction

A distinct type among signal varieties is called chaotic signals [1]. Fundamentally, these signals rely on dynamic chaos, which is, essentially, a deterministic random phenomenon. Chaotic signals are used in digital communications [2], radar systems [3], by physicists studying real systems [4] or in medical diagnostics [5], as well as other scientific and technical fields.

One of the most common model classes used for generation of discrete-time chaotic signals $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ is discrete map, such as:

$$\mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k,\, \mathbf{p}), \tag{1}$$

with properties:

$$\mathbf{s} \in \mathrm{S} \subset \mathbb{R}^N, \quad \mathbf{p} \in \mathrm{P} \subset \mathbb{R}^L, \quad \mathbf{f} \in \mathrm{C}^0(\mathrm{S}\times\mathrm{P}),$$
$$k \in \mathrm{K} \subseteq \mathbb{Z}, \quad n \in \overline{1,\, N}, \quad l \in \overline{1,\, L},$$

where $\mathbf{s}$ – map state variable, $\mathbf{p}$ – vector of map parameters, k – discrete time.

One of the primary problems in working with chaotic signals is identification of chaotic oscillations [1], which is predominantly solved by analyzing the Lyapunov exponent Λ of the investigated system and/or process [6], [7]. This quantity describes local exponential divergence of two close trajectories. It can also be interpreted in terms of the rate of loss of information about the initial conditions during iterations (1).

In theoretical problems, Λ value is usually calculated based on the signal model (1); in practical problems, Λ is often estimated based on the observed realization of signal $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$.

In cases with a known signal model (1) and known parameters of the signal generator, the Lyapunov exponent can be calculated using the numbers in the matrix itself

$$\mathbf{M}^k_{ij} = \big[\mathbf{G}(\mathbf{s}_{k-1})\, \mathbf{G}(\mathbf{s}_{k-2}) \ldots \mathbf{G}(\mathbf{s}_0)\big]^{\frac{1}{k}}_{ij},$$
$$\mathbf{G}_{ij}(\mathbf{s}_k) = \frac{\partial\, \mathbf{f}_i}{\partial\, \mathbf{s}^{(j)}}\Big|_{\mathbf{s}=\mathbf{s}_k},$$

where $\mathbf{G}$ – Jacobian matrix of system (1). As a rule, in real-life applications, different variations of the Benettin algorithm are used in such cases [8]. In a unidimensional case $N = 1$, the problem is reduce [7] and can be calculated directly:

$$\Lambda\big(s_0\big) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \ln\Big|\frac{\mathrm{d}}{\mathrm{d}\, s} f^k(s_0)\Big|. \tag{2}$$

In cases where Λ is estimated by the observable realization of signal, the first approach, chronologically, was proposed by Wolf [9]. Numerous other approaches have been developed since, including methods that use machine learning [10]. However, they are mostly centered around attractor reconstruction (especially for $N = 1$ cases) [7], which is generally unstable when the observed trajectories are short (approximately $K < 10^4$ sequence elements). Therefore, this problem remains open.

The field of digital signal processing has been recently approaching such problems as filtering, estimation, recognition of deterministic signals in noise using deep learning algorithms [11], specifically – deep convolutional neural networks. These networks are based on digital FIR filter banks that come straight from digital signal processing. This approach has been proven viable for processing signals that have clearly distinct patterns (speech, images, etc.) [11]–[13]. Regarding chaotic (noise-like) signals that are studied in this paper, there has been no systematic investigation of convolutional neural network effectiveness or decision-making process.

This study investigates the applicability of a deep neural network for estimating the Lyapunov exponent by the observed realization of signal $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ in case $N = 1$.

II. Technique and theoretical premises

The approach to estimating parameter Λ of chaotic signals rests on a set of theoretical premises.

First, it follows from the assumptions and conclusions of the universal approximation theorem of Cybenko [14] that deep convolutional neural networks (given their high generalization

ISBN 978-90-827970-1-5

power [11]) can turn out to be an effective nonlinear approximator of attractor properties that are relevant for estimating the Lyapunov exponent [15].

Second, estimating the Lyapunov exponent from observed realization of signal $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ in case $N = 1$ relies heavily on the Takens theorem of attractor reconstruction from its observed projection [16]. The theorem lends solid mathematical foundations to general ideas of nonlinear autoregression and analyzing the structure of signal $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ in time domain, see the postulates of the method of singular spectrum analysis [17].

Combining the two aforementioned aspects and applying them to the properties of convolutional neural networks [18] necessitates convolution of signal $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ into delay matrix $\mathbf{S}$:

$$\mathbf{S} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{Nc-1} \\ s_{Nc} & s_{Nc+1} & \cdots & s_{2Nc-1} \\ \cdots & \cdots & \cdots & \cdots \\ s_{(Nr-1)Nc} & s_{(Nr-1)Nc+1} & \cdots & s_{NrNc-1} \end{bmatrix}, \quad (3)$$

where $K = Nr\,Nc$. Matrix $\mathbf{S}$ is, in turn, fed to the neural network with 2D convolutional kernels. This architecture allows for using compact kernels to estimate the $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ structure at significantly varying scales. The output layer is designed to solve the regression problem to get an estimation of Λ.

III. Experiment

A. Methodology

The proposed approach to estimating the Lyapunov exponent from signal realization was tested on time series generated by logistic map:

$$s_{k+1} = f(s_k,\, \lambda) = 4\,\lambda\, s_k\, (1 - s_k), \quad (4)$$

where: $\lambda \in (0,\, 1]$ – control parameter, $s \in (0,\, 1)$ – system phase variable. Methodologically, the value of this oscillator is that, given its relative simplicity, the structure of dynamics (4) is surprisingly rich [7] beyond the limit doubling point of period $\lambda_\infty = 0.892486418\ldots$: chaotic regimes alternate with cycles of different periods that then turn back into chaos; the value of $\lambda_{3c} = 1/4 + 1/\sqrt{2}$ is also identified, which corresponds to tangential bifurcation point (saddle-node bifurcation or transition into chaos through intermittency).

The transition into chaos occurs by the classical scenario of period doubling. Besides, this oscillator has a practical value, including its physical and technical realizations: electronic circuit [19], quantum-optic [20], possible designs of encrypted communication systems that operate on top of a chaotic signal [2], [21], [22].

The signals generated by map (4) were used in interval $k \in [1,\, K] + 1 \cdot 10^5$. Such deviation from $k = 1$ is explained by the need to minimize the spurious effect of transition in order to form a stable attractor. The control parameter ranged within $\lambda \in [0.89,\, 1]$, step $1 \cdot 10^{-4}$. For each λ value, 100 signal realizations were generated with different initial conditions $s_0 = \xi$, where $\xi \in (0,\, 1)$ – non-correlated normally distributed pseudorandom values. This neutralized the memory effect induced by the initial conditions for the signals. All signal sequences contained $K = 1024$ elements, since this value is the average length in some of the most common real-life problems [12], [13].

Non-overlapping sets of sequences $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ were formed to train and test the neural network estimator. Training was carried out in value range $\lambda \in [0.89,\, 0.94]$ (42 600 chaotic and 7 500 non-chaotic realizations); testing range was $\lambda \in (0.94,\, 1]$ (53 700 chaotic and 6 300 non-chaotic realizations). Sequences $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ were used to form matrices $\mathbf{S}$ with shape 32×32 that were fed to the neural network as input.

Deep optimization (using a genetic algorithm based on a modified version of the differential evolution method [23]) allowed us to synthesize the deep convolutional neural network structure, see Fig 1. The optimization helped minimize the network size (number of parameters) and align its depth (number of layers) with its width (layer sizes). The aim of optimization was to attain the best quality classifier at the lowest computational cost. Fig. 1 shows that the deep

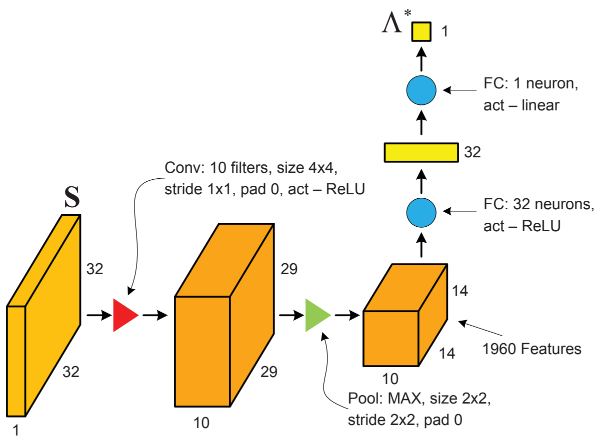

Fig. 1. CNN estimator structure.

convolutional neural network has 3 hidden layers, where 2 are convolutional and 1 is fully connected. Convolutional layers contain pooling operations (MAX configuration) as well as convolutions themselves. The activation function in all hidden layers is ReLU.

Fig 2 shows the structure of the receptive field of the first convolutional layer of the neural network estimator described in section III-A.

The data suggest that, due to the size of the convolutional filters 4×4 and the special structure of $\mathbf{S}$ (3), the filters can process four groups with four consecutive sequence elements each and also form quite complex features within the time scale of 1 to 100 sequence elements. Applying MAXPool (size 2×2, stride 2×2) after the convolutional level results in a feature structure that is sufficiently invariable to small phase shifts in the original sequence $\{\mathbf{s}_k\}_{k\in\mathrm{K}}$ [11]. The first fully connected layer (32 neurons) takes 1960 features to form meta-features that are combined into a Λ^* estimation in the output layer.

ISBN 978-90-827970-1-5

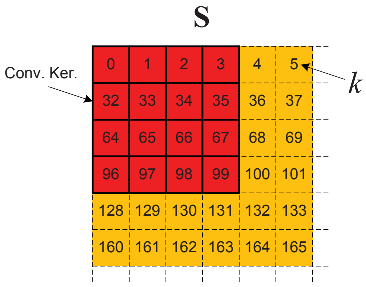

Fig. 2. Structure of the receptive field (red) of the first convolutional layer.

When training, the regression problem was solved and the loss function was minimized by MSE (mean squared error):

$$L = -\frac{1}{R}\sum_{i=1}^{R}\left(\Lambda_i^* - \Lambda_i\right)^2, \tag{5}$$

where R – training set size, Λ_i – value of the Lyapunov exponent associated with the i-th signal in the training set. To decrease overfitting, *Dropout* regularization [11] at 0.55 was applied after the pooling layer. The network was trained for 300 epochs using *rmsprop* optimizer with adaptive step size and control of the loss attribute. One mini-batch was 128 samples. All training was run with Keras Framework [24] (build 2.1.2) on top of the Microsoft CNTK [25] (build 2.3.1).

B. Results

To estimate coherence of the classifier decisions, a network uncertainty index was introduced (Λ^* – CNN and Λ – formula (2))

$$\Phi^* = \frac{1}{2}\log\frac{\mathrm{D}\left[\Lambda^*\right]_{s_0}}{\mathrm{D}\left[\Lambda\right]_{s_0}}, \tag{6}$$

where: D – operator to calculate dispersion (the dispersion is calculated from a set of initial conditions s_0), log – decimal logarithm. It follows from (6) that at $\Phi^* > 0$ the standard deviation of estimation from a set of signals is higher than the standard deviation of estimation calculated from (2), which points to an uncertainty of the neural network about its conclusions.

Fig. 3a shows median values (from the initial conditions set s_0) of Lyapunov exponent estimations: $\overline{\Lambda}$ – formula (2) and $\overline{\Lambda^*}$ – CNN. Fig. 3b shows values of the network uncertainty index (6).

As reference [26] has demonstrated, within interval $\lambda \in [0.89, 1]$ the attractor undergoes 5 structural rearrangements that result in a qualitative increase in complexity of the trajectory shape. With that, according to the experimental conditions, the training set includes only one such bifurcation $\lambda_{\mathrm{T5N}} = 0.919643377\ldots$ Although the proposed neural network classifier produces on average a higher dispersion of estimations (see Fig. 3b), when compared to a calculation from formula (2), it still produces a Lyapunov exponent estimation with an acceptable error and for attractors with markedly changed structures (see Fig. 3a). MPE (mean percentage error)

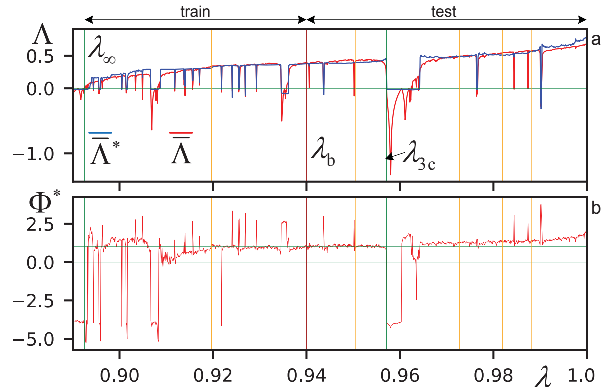

Fig. 3. Dependence on parameter λ: (a) – of estimations Λ; (b) – of the uncertainty index (6).

and MAPE (mean absolute percentage error) scores on the test interval (for chaotic trajectories with $\Lambda > 0$) have the following values:

$$\begin{array}{cccccc} \% & 5 & 25 & 50 & 75 & 95 \\ MPE & -21.75 & -6.83 & 0.83 & 6.16 & 13.13 \\ MAPE & 0.61 & 3.12 & 6.39 & 10.81 & 45.00 \end{array} \tag{7}$$

It follows from (7) that 75% of decisions have an error of less than 11% on the MAPE score.

The proposed neural network estimator for the Lyapunov exponent can also be used for detection of chaotic signals:

$$c^{\mathrm{ps}} = \mathrm{sign}\,\Phi^*. \tag{8}$$

In this case, value c^{ps} can be interpreted as decisions of a binary classifier, where $c^{\mathrm{ps}} = 1$ – meets the condition for presence of a chaotic signal ($\Lambda > 0$), event *positive*, and $c^{\mathrm{ps}} = -1$ – no chaotic signal ($\Lambda \leqslant 0$), event *negative*. For main parameters of the classifier quality, see Table I. Note

TABLE I

CLASSIFIER (8): CONFUSION MATRIX, TEST INTERVAL. LEGEND FOR TABLE [27]: TS – CLASS REFERENCE LABELS; PR – CLASSIFIER PREDICTIONS; PREC. – PRECISION METRICS; REC. – RECALL METRICS; F1 SC. – F_1 SCORE.

Signal class		Ts		Prec.	Rec.	F1 sc.
		$\Lambda \leqslant 0$	$\Lambda > 0$			
Pr	$\Lambda \leqslant 0$	6 000	1 900	0.7595	0.9524	0.8451
	$\Lambda > 0$	300	51 800	0.9942	0.9646	0.9792
Summary		6 300	53 700	0.8769	0.9585	0.9121

that in averaging of Precision, Recall and F_1 score (summary line) a pessimistic averaging strategy macro was used (not accounting for class imbalance in the training and testing sets, or prevalence of chaotic signals). Thus, $F_1 = 0.9121$ represents adequate quality of classifier (8).

In addition to the quality of the synthesized classifier shown in Table I, we look at the structure of its errors depending on λ

ISBN 978-90-827970-1-5 © EURASIP 2018

values. To this end, two values were calculated:

$$F_{\text{ch}}(\lambda) = \sum_{n=1}^{N_s} \begin{cases} 1 & \Lambda_n^*(\lambda) \leqslant 0 \wedge \Lambda_n(\lambda) > 0, \\ 0 & otherwize. \end{cases};$$

$$F_{\overline{\text{ch}}}(\lambda) = \sum_{n=1}^{N_s} \begin{cases} 1 & \Lambda_n^*(\lambda) > 0 \wedge \Lambda_n(\lambda) \leqslant 0, \\ 0 & otherwize. \end{cases};$$

where summation in the formulae is performed in the initial conditions set. Consequentially, F_{ch} corresponds to false-negatives and $F_{\overline{\text{ch}}}$ to false-positives as related to λ values. The values are shown in Fig. 4.

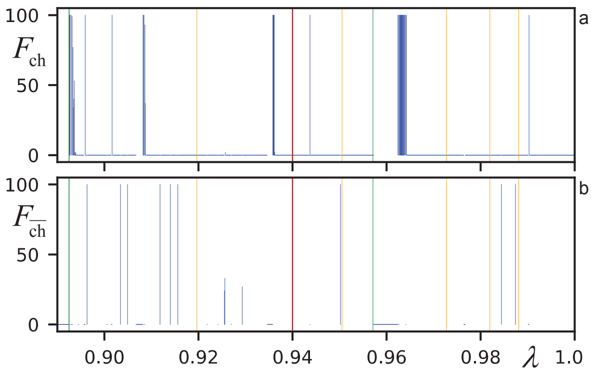

Fig. 4. Structure of classifier errors depending on λ values.

Fig. 4 shows that classifier errors group into packages that concentrate near certain values of the controlling parameter λ. Analysis of package locations taken together with the data in Fig. 3a indicates that the most common false-negative errors occur in subduction windows [28] – a sudden narrowing of the chaotic band – where oscillations change from non-chaotic to chaotic, i.e., at $\Lambda \to +0$.

IV. Conclusion

This study has demonstrated effective capabilities of a relatively simple deep convolutional neural network in estimating the Lyapunov exponent (the error rate in 75% of decisions is less than 11% on the MAPE measure) and detecting chaotic signals ($F_1 = 91.21\%$) generated by a logistic map. This is true even through in the test interval of change of control parameter $\lambda \in (0.94,\ 1]$ the generator undergoes 4 structural rearrangements of the attractor that result in a qualitative increase in complexity of the trajectory shape [26].

Unlike a number of other estimation algorithms (e.g., see [10]), our solution does not require attractor reconstruction (which is generally quite unstable) and operates on rather short signals $K = 1024$ sequence elements in the experiment. Moreover, the proposed solution works directly with raw data, automatically synthesized informative features, makes a direct estimation of the Lyapunov exponent Λ and flags the present chaotic signals. Note also that the pre-trained neural network makes estimate Λ in a non-iterative way. The network itself is quite compact in size, see Fig. 1, and contains multiplication and addition operations. The computational power it uses is therefore adequately low.

Structural analysis of the synthesized and trained convolutional network and its possible functional mechanism, as applied to the problem at hand, has shown that creating an input signal in the form of delay matrix (3) and the size of 2D convolutional filters in the first hidden layer are rather critical and affect the quality of the final decision. In this context, two issues draw certain attention: (i) – why is this specific network structure (see Fig. 1) happens to be the most effective for this application; (ii) – how much will the network structure need to change to maintain its effectiveness in case of change in the chaos structure and/or signal length. We will address these and other issues in future studies.

This leads us to conclude tentatively that a certain class of problems in digital processing of chaotic signals can be adequately solved by deep learning [11] with automatic generation of informative features and key decision-making rules. This study has also shown that deep neural networks are effective in applications involving noise-like signals (down to narrowband or broadband stochastic processes), as well as distinct patterns in signals [11]–[13].

The authors thank the anonymous referees for their useful comments.

References

[1] H. Leung, Chaotic Signal Processing. SIAM, 2013.
[2] M. Eisencraft, R. Attux, and R. Suyama, Chaotic Signals in Digital Communications. CRC Press, 2017.
[3] S.A. Harman, A.J. Fenwick and C. Williams, Chaotic signals in radar?, 2006 European Radar Conference, Manchester, pp. 49–52. IEEE, 2006.
[4] HDI. Abarbanel, Chaotic signals and physical systems, ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 113–116. IEEE, 1992.
[5] M.E. Cohen and D.L. Hudson, New chaotic methods for biomedical signal analysis, Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine. ITAB-ITIS 2000. Joint Meeting Third IEEE EMBS International Conference on Information Technol, Arlington, VA, pp. 123–128. IEEE, 2000.
[6] W. Kinsner, Characterizing chaos through Lyapunov metrics, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 36, pp. 141–151, March 2006.
[7] M. Tabor, Chaos and Integrability in Nonlinear Dynamics: An Introduction. Wiley-Interscience, 1989.
[8] G. Benettin, L. Galgani, A. Giorgilli, and J.-M. Strelcyn, Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems; A Method for Computing All of Them, Meccanica, vol. 15, pp. 9–20, 1980.
[9] A. Wolf, J.B. Swift, H.L. Swinney, and J.A. Vastano, Determining Lyapunov exponents from a time series, Physica D: Nonlinear Phenomena, vol. 16, pp. 285–317, July 1985.
[10] J. Pathak, Z. Lu, B.R. Hunt, M. Girvan, and E. Ott, Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 27, p. 121102, 2017.
[11] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[12] A.V. Makarenko, Deep learning algorithms for signal recognition in long perimeter monitoring distributed fiber optic sensors, Proceedings of the IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), Sept 2016, pp. 1–6. IEEE, 2016.
[13] O. Abdel-Hamid, A.R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, Convolutional neural networks for speech recognition, IEEE/ACM Trans. Audio, Speech and Lang. Proc., vol. 22, pp. 1533–1545, 2014.
[14] G.V. Cybenko, Approximation by Superpositions of a Sigmoidal function, Mathematics of Control Signals and Systems, vol. 2, pp. 303–314, 1989.

[15] R. Brown, P. Bryant, and H.D.I. Abarbanel, Computing the Lyapunov spectrum of a dynamical system from an observed time series, Phys. Rev. A, vol. 43, pp. 2787–2806, March 1991.

[16] F. Takens, Detecting strange attractors in turbulence, Lecture Notes in Math, vol. 898, pp. 366–381, 1981.

[17] J.B. Elsner and A.A. Tsonis, Singular Spectrum Analysis. A New Tool in Time Series Analysis. Plenum Press, 1996.

[18] S. Mallat, Understanding deep convolutional networks, Phil. Trans. R. Soc. A, vol. 374, 20150203, March 2016.

[19] M. Suneel, Electronic circuit realization of the logistic map, Sadhana, vol. 31, pp. 69–78, Feb. 2006.

[20] M.V. Gorbunkov, Yu.Ya. Maslova, V.A. Petukhov, M.A. Semenov, and Yu.V. Shabalin, Discrete Maps and the Problem of Round Trip Time Scale Nonlinear Dynamics in Solid-State Lasers, Nonlinear Maps and their Applications, Springer Proceedings in Mathematics&Statistics. vol. 112, Springer International Publishing Switzerland, pp. 159–170, 2015.

[21] P. Colet and R. Roy, Digital communication with synchronized chaotic lasers, Opt. Lett., vol. 19, pp. 2056–2058, 1994.

[22] L. Larger and J.-P. Goedgebuer, Encryption using chaotic dynamics for optical telecommunications, C. R. Physique, vol. 5, pp. 609–611, 2004.

[23] K. Price, R.M. Storn, and J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization. Springer, 2005.

[24] F. Chollet et al., Keras. GitHub, https://github.com/keras-team/keras, 2015.

[25] F. Seide and A. Agarwal, CNTK: Microsoft's Open-Source Deep-Learning Toolkit, Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16, p. 2135. ACM, 2016.

[26] A.V. Makarenko, The TQ-bifurcation in discrete dynamical systems. General properties, Proceedings of the International Conference Stability and Oscillations of Nonlinear Control Systems, ICS RAS, pp. 1–4. IEEE, 2016.

[27] C. Sammut, G.I. Webb, Encyclopedia of Machine Learning and Data Mining. Springer, 2017.

[28] C. Grebogi, E. Ott, J.A. Yorke, Crises, sudden, changes in chaotic attractors, and transient chaos, Physica D, vol. 7, pp. 181–200, 1983.

ISBN 978-90-827970-1-5 © EURASIP 2018