Letter on Convergence of In-Parameter-Linear Nonlinear Neural Architectures With Gradient Learnings

Ivo Bukovsky\textsuperscript{\textcopyright}, Senior Member, IEEE, Gejza Dohnal, Peter M. Benes\textsuperscript{\textcopyright}, Kei Ichiji\textsuperscript{\textcopyright}, and Noriyasu Homma\textsuperscript{\textcopyright}, Member, IEEE

Abstract—This letter summarizes and proves the concept of bounded-input bounded-state (BIBS) stability for weight convergence of a broad family of in-parameter-linear nonlinear neural architectures (IPLNAs) as it generally applies to a broad family of incremental gradient learning algorithms. A practical BIBS convergence condition results from the derived proofs for every individual learning point or batches for real-time applications.

Index Terms—Bounded-input bounded-state stability (BIBS), extreme learning machines, in-parameter-linear nonlinear neural architectures (IPLNAs), incremental gradient learnings, input-to-state stability (ISS), polynomial neural networks, random vector functional link networks, weight convergence.

I. INTRODUCTION

The bounded-input bounded-state (BIBS) stability concept is recently popular in neural networks. Also, the weight convergence of gradient learning is still an investigated issue. However, no paper recalls nor thoroughly presents and proves this concept for the incremental gradient-learning weight convergence of in-parameter-linear nonlinear (neural) architectures (IPLNAs) in general.

By IPLNAs, in this brief, we consider a wide family of shallow neural networks including the extreme learning machine (ELM) or random vector functional link (RVFL) networks [1]–[3], other functional links and kernel neural architectures and filters, e.g., [4]–[6], and polynomial neural networks [7], [8] including basic standalone SP structures called polynomial neural units in [9], [10].

 Manuscript received 10 April 2021; revised 23 September 2021; accepted 23 October 2021. Date of publication 15 November 2021; date of current version 4 August 2023. The work of Ivo Bukovsky was supported in part by the Fund of Strategic Priorities of the University of South Bohemia in Ceske Budejovice (FSP JU); in part by the ESI F, EU Operational Program Research, Development and Education; and in part by the Center of Advanced Aerospace Technology, Faculty of Mechanical Engineering, Czech Technical University in Prague under Grant CZ.02.1.01/0.0/0.0/16_019/0000826. The work of Gejza Dohnal was supported in part by the ESIF, EU Operational Program Research, Development and Education; and in part by the Center of Advanced Aerospace Technology, Faculty of Mechanical Engineering, Czech Technical University in Prague under Grant CZ.02.1.01/0.0/0.0/16_019/0000826. The work of Noriyasu Homma was supported in part by the JSPS KAKENHI under Grant 18K19892 and Grant 19H04479 and in part by the Smart Aging Research Center, Tohoku University. (Corresponding author: Ivo Bukovsky.)

Ivo Bukovsky is with the Department of Computer Science, Faculty of Science, University of South Bohemia in Ceske Budejovice, 370 05 Ceske Budejovice, Czech Republic, and also with the Department of Mechanics, Biomechanics, and Mechatronics, Faculty of Mechanical Engineering, Czech Technical University in Prague, 166 07 Prague, Czech Republic.

Gejza Dohnal is with the Department of Technical Mathematics, Faculty of Mechanical Engineering, Czech Technical University in Prague, 166 07 Prague, Czech Republic.

Peter M. Benes is with the Department of Rail Automation, Siemens Mobility s.r.o., 18200 Prague, Czech Republic.

Kei Ichiji and Noriyasu Homma are with the Department of Radiological Imaging and Informatics, Tohoku University Graduate School of Medicine, Sendai 980-8575, Japan.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3123533.

Digital Object Identifier 10.1109/TNNLS.2021.3123533.

This letter shows that the input-to-state stability (ISS) concept [11] and BIBS stability [12] generally apply to the gradient learning algorithms and their many modifications for IPLNAs. The most prevalent ones to mention are the (stochastic) gradient descent with normalized learning rate (also known as normalized least means squares (NLMS) [13]–[15]), recursive least squares (RLS) [16], adaptive moment estimation (ADAM) [17]. In general, there are many modifications of gradient learnings that are based on.

1) The normalization of learning rate and the adaptation of the regularization term (such as NLMS, generalized gradient descent (GNGD) [18], robust regularized NLMS (RR-NLMS) [19]);
2) the adaptation of the learning rate (such as Benveniste et al. [13], Ang and Farhang-Boroujeny [20], Mathew and Xie [21]); and
3) other variations, e.g., RLS that uses covariance matrix and momentum-enhanced methods such as ADAM and the related predecessors and followers as overviewed in [22].

The purpose of this brief is to summarize and prove the general applicability of ISS and BIBS to the weight convergence of IPLNAs and to highlight its practical aspect for the broad family of gradient learning rules of IPLNAs.

The bold letters and symbols stand for vectors and matrices, and sample index $k$ indicates time variability.

II. BACKGROUND ON IPLNAs AND GRADIENT LEARNINGS

This subsection generally defines IPLNAs and recalls BIBS stability concept applied to their gradient-based weight-update system.

\textbf{Definition 1}: The IPLNA is defined by (1) where $g(x)$ is not a function of neural weights $w$, i.e., $g(x(k)) \neq g(w)$.

The IPLNA that conforms to Definition 1 is as follows:

\begin{equation}
\dot{y}(k) = w^T(k) \cdot g(x(k))
\end{equation}

where $\dot{y}(k)$ is neural output; $w$ is the column vector of neural weights; superscript $T$ stands for transposition; $g(\cdot)$ can be:

1) vector of basis functions or kernel vector function; or
2) random vector functional link expansion as with ELMs and RVFLs;

\text{i.e., } g(x) \text{ transforms the basic feature vector } x \text{ into a new feature vector } g(x) \text{ independently of } w \text{ (Definition 1). Furthermore, in (1), } v = v(k) \text{ are additional parameters that can vary in time. The gradient learning rule details for IPLNAs (1) and some of their distinctions for variations of gradient learning algorithms are sketched in Table I, where most distinctions lie in the time-varying learning rate } \eta = \eta(k). \text{ For notational simplicity, } g(x(k)) \text{ can be shortened as } g(x) \text{ or } g(x,k) \text{ and so on.}

First, let us consider nonmomentum gradient methods, such as sketched in Table I for NLMS or RLS, where the weight updates for IPLNAs result in time-variant state-space representation as follows:

\begin{equation}
w(k + 1) = (I - \eta(k) \cdot g(x) \cdot g(x)^T) \cdot w(k) + \eta(k) \cdot y(k) \cdot g(x)
\end{equation}

where $(I - \eta(k) \cdot g(x) \cdot g(x)^T)$ is the local matrix of dynamics (LMD); $\eta(k) \cdot y(k) \cdot g(x)$ is the input term, where the time-varying...
TABLE I
GENERAL INCREMENTAL GRADIENT-LEARNING SCHEME FOR IPLNAS AND ITS THREE MOST POPULAR INCREMENTAL GRADIENT LEARNING VARIATIONS (BRIEFLY SKETCHED TO INDICATE THE PRINCIPAL DIFFERENCES WHILE THE DETAILS CAN BE FOUND IN CITED LITERATURE AND REFERENCES THEREIN)

| Learning Algorithm (in sketch) | Learning scheme: \( \mathbf{w}(k+1) = \mathbf{w}(k) - \eta(k) \frac{\partial Q(k)}{\partial \mathbf{w}} \) |
|-------------------------------|--------------------------------------------------|
| **NGD (NLMS [13–15])**       | \[
\eta(k) = \eta(g(x) + \mu) = \frac{\mu}{||g(x)||^2 + \varepsilon}
\]
where the constant learning rate \( \mu < 2 \), and the regularization term \( \varepsilon \) is small. |
| **RLS [19]**                  | \[
\eta(k) = \eta(g(x) + \mu, k) = -\gamma R^{-1}(R^{-1}(k-1)g(x), \mu)
\]
where the constant learning rate is usually \( \mu \approx 0.9 \). |
| **ADAM [17]**                 | \[
\mathbf{m}(k+1) = \beta_{m}(\mathbf{m}(k) + (1 - \beta_{m}) \frac{\partial Q(k)}{\partial \mathbf{w}})
\]

III. BIBS CONVERGENCE OF IPLNA GRADIENT LEARNINGS

This section recalls, applies, and proves the BIBS stability concept for a class of incremental weight-update systems and introduces the strict BIBS condition for weight convergence. Based on the proof, a new definition of strict BIBS stability is defined to assure the strict weight convergence of IPLNAS with gradient learning schemes as in (2) or (4), i.e., in a general form (5).

**Definition 2:** If every bounded input of a system results in a bounded state, then the system is called BIBS stable. If every bounded input of a system results in a bounded output, then that system is called bounded-input/bounded-output (BIBO) stable.

A system (5) is BIBS stable if there exist two positive constants, such that the conditions

\[
\|\mathbf{w}(k)\| \leq L_u, \forall k \geq k_0
\]

imply that \( \|\mathbf{w}(k)\| \leq L_u, \forall k \geq k_0 \). The system is BIBO stable if there exists two constants \( 0 < L_u, L_y < \infty \), such that condition (6) implies that \( \|y(k)\| \leq L_y, \forall k \geq k_0 \).

In practice, the initial weights \( \mathbf{w}(k_0) \) are random values; however, it does not violate the BIBS stability as proven further via (7)–(19).

**Theorem 1:** Time-variant discrete-time weight-update system (5) is BIBS stable if there exist constants \( L_k, M_k, B_k \) for which

\[
sup_{k \geq k_0} ||\mathbf{w}(k)|| = L_u < \infty, sup_{k \geq k_0} ||\mathbf{a}(k)|| = M_u < 1, \quad \text{and} \quad sup_{k \geq k_0} ||\mathbf{b}(k)|| = M_y < \infty.
\]

**Proof:** (based on the proof of Theorem 3 in [12]): The weight-update system (5) unfolds in \( k \) via the scheme as follows:

\[
\begin{align*}
\mathbf{w}(k) &= \mathbf{A}(k-1)\mathbf{w}(k-1) + \mathbf{B}(k-1)\mathbf{u}(k-1) \quad (7) \\
\mathbf{w}(k+1) &= \mathbf{A}(k)\mathbf{A}(k-1)\mathbf{w}(k-1) + \mathbf{A}(k)\mathbf{B}(k-1)\mathbf{u}(k-1) + \mathbf{B}(k)\mathbf{u}(k) \\
&\vdots \\
\mathbf{w}(k+1) &= \prod_{j=0}^{k-1} \mathbf{A}(k-j) \mathbf{w}(k_0) \\
&+ \sum_{i=0}^{k-1} \prod_{j=1}^{k-i} \mathbf{A}(k-j+1) \mathbf{B}(i)\mathbf{u}(i) + \mathbf{B}(k)\mathbf{u}(k) \quad (8)
\end{align*}
\]

Using the notation

\[
\mathbf{A}_k = \prod_{j=0}^{k-1} \mathbf{A}(k-j), \quad \mathbf{C}_{k,i} = \prod_{j=1}^{k-i} \mathbf{A}(k-j+1),
\]

where \( i = k_0, k_0+1, \ldots, k-1 \), and \( \mathbf{C}_{k,k} = \mathbf{I} \), we can rewrite (9) as

\[
\mathbf{w}(k+1) = \mathbf{A}_k\mathbf{w}(k_0) + \sum_{i=k_0}^{k} \mathbf{C}_{k,i} \mathbf{B}(i)\mathbf{u}(i) \quad (10)
\]
After applying a norm and using the triangle inequality, we obtain
\[ \|w(k + 1)\| \leq \|A_k\| \cdot \|w(k_0)\| + k \sum_{i=k_0}^{k} \|C_{i,k}\| \cdot \|\mathbf{B}(i)\| \cdot \|u(i)\| = D_k. \] (12)

The constraints in Theorem 1 imply the following:
\[ \|A_k\| \leq M_A^{k-k_0+1} < 1 \] (13)
\[ \|C_{i,k}\| \leq M_A^{k-i} \] (14)
\[ \|\mathbf{B}(i)\| \cdot \|u(i)\| \leq M_B \cdot L_u. \] (15)

Therefore, it holds for the right-hand side of (12) that
\[ D_k \leq M_A^{k-k_0+1} \cdot \|w(k_0)\| + M_B \cdot L_u \sum_{i=k_0}^{k} M_A^{k-i} \] (16)
where the last term represents a partial sum of a geometric sequence, i.e.,
\[ \sum_{i=k_0}^{k} M_A^{k-i} = \sum_{i=0}^{k-k_0} M_A^i = \frac{1 - M_A^{k-k_0+1}}{1 - M_A} - \frac{1}{1 - M_A}. \] (17)

Finally, it completes the proof that
\[ \|w(k+1)\| \leq D_k \leq \|w(k_0)\| + \frac{M_B L_u}{1 - M_A} \] (18)
where the weight initiation \( w(k_0) \) implies that there exists the constant \( L_w = \|w(k_0)\| + ((M_B L_u)/(1 - M_A)) \) such that
\[ \|w(k)\| \leq L_u \text{ for all } k > k_0. \] (19)

IV. CONNOTATIONS TO COMMON BIBS AND ISS STABILITY

From the bounded input assumption, it yields that there exist finite upper bounds \( M_B, L_u \) for \( \|\mathbf{B}(k)\| \) and \( \|u(k)\| \). Using the ratio criterion for convergence of the series in (12), we should find a constant \( q < 1 \) so that
\[ \rho(\mathbf{A}(i)) \leq \frac{\|C_{i,j}\|}{\|C_{i,j-1}\|} = \|\mathbf{A}(i)\| \leq q, \quad i > k_0 \] (20)
where \( \rho(\mathbf{A}) \) denotes the spectral radius of \( \mathbf{A} \).

Corollary: IPLNAs with learning-rule state-space representation (2), for which there exists a constant \( 0 \leq q < 1 \) such that
\[ \rho(\mathbf{I} - \eta(k) \cdot \mathbf{g}(x(k)) \cdot \mathbf{g}^T(x(k)), k) \leq q \] (21)
for all \( k > k_0 \), are BIBS stable. \( \square \)

Proof: It is well known that \( \|\mathbf{A}\| < 1 \) implies \( \rho(\mathbf{A}) < 1 \). Recall that for real-time learning \( \mathbf{A}(k) = (\mathbf{I} - \eta(k) \cdot \mathbf{g}(x(k)) \cdot \mathbf{g}^T(x(k)), k) \), \( \mathbf{B}(k) = \eta(k) \), and \( u(k) = y(k) \cdot \mathbf{g}(x(k)) \) as it is already indicated in (2) or as it can be extended for (4). Therefore, (21) implies (20); this assures the convergence of the series in (12) and hence,
\[ \|w(k)\| \leq \|w(k_0)\| + K \] (22)
where \( K = M_B L_u/(1 - q) \). \( \square \)

Definition 3: A system (2) is input-to-state stable (ISS) if there exist functions \( \beta \) and \( \gamma \), such that
\[ \|w(k+1)\| \leq \beta(\|w(k_0)\|, k) + \gamma(\|u\|) \] (23)
where \( \beta \) is the KL̄ function and \( \gamma \) is the \( K_\infty \) function [11]. \( \square \)

Theorem 2: IPLNAs that satisfy the condition (20) are ISS. \( \square \)

Proof: From (12), it follows that in the case of (2), we have
\[ \|w(k+1)\| \leq \|A_k\| \cdot \|w(k_0)\| + \|u\| \sum_{i=k_0}^{k} \|C_{i,k}\| \cdot \|\mathbf{B}(i)\| \] (24)

where \( \|u\| = \sup_{0 \leq i \leq 2^k} (\|u(i)\|) \), then let \( \beta(\|w(k_0)\|, k) = \|A_k\| \cdot \|w(k_0)\| \), and \( \gamma(\|u\|) = \|u\| \cdot \sum_{i=k_0}^{k} \|C_{i,k}\| \cdot \|\mathbf{B}(i)\| \).

From the proof of Corollary 1, it is now clear that \( \gamma \) is \( K_\infty \) function. Now, it is shown that \( \beta \) belongs to KL̄ function.

First, for each \( k \geq k_0 \), the function \( \beta(\|w(k_0)\|, k) \) is linear in \( \|w(k_0)\| \), and hence it is of class \( K_\infty \). Second, if the condition (21) holds, then \( \lim_{k \to \infty} \|A_k\| = 0 \). From this, it follows that \( \beta \) belongs to KL̄ class. \( \square \)

V. CONSEQUENCES TO REAL APPLICATIONS

In practice, the training data are bounded, and so is the gain matrix \( \mathbf{B}(k) \). Thus, the previous sections thoroughly prove that the sufficient condition to maintain the learning convergence under ISS’s umbrella and in the sense of BIBS stability is (20) that yields (21) for non-momentum gradient algorithms. For experimental results, e.g., with a class of polynomial neural architectures, please see paper [10], where spectral radii’s effect \( \rho(A) \) on gradient learning is studied and shown in detail. Also, this letter’s proofs are the theoretical complements to the earlier, i.e., more experimentally focused work [10] that did not explicitly show the whole theoretical kinship.

The BIBS condition (20) can be restated as
\[ \rho(\mathbf{A}(k)) \leq \|\mathbf{A}(k)\| \leq 1 \quad \forall k \] (25)
that is very strict, and also, the norm must not necessarily be kept all the time below 1 as it practically fluctuates around 1 as also shown in [10]. Thus, it is more practical to maintain the following condition
\[ \frac{\|C_{i,k}\|}{\|C_{i,k-1}\|} = \prod_{j=1}^{k-p} A(k - j + 1) \leq 1 \] (26)
where \( p \) is the custom number of samples, i.e., every \( p \)-sample sliding window, for which the condition (26) should be maintained. Any unusually large increase of (25) or (26) that extraordinary exceeds \( I \) then indicates the loss of weight-update stability that has to be avoided, e.g., see [10 Figs. 5 and 6] as the example of the violation of (20) for a class of polynomial (recurring) IPLNAs.

VI. CONCLUSION

This brief showed and proved that the ISS framework and BIBS concept are universally applicable to the weight convergence of a broad family of IPLNAs for various modifications of incremental gradient learning algorithms. For real applications, the introduced weight-update stability condition should be monitored and accordingly maintained to avoid the instability of real-time learning systems. The implementation of the weight convergence condition is extraordinarily feasible and very practical; this is because it is enough to monitor the spectral radius that can be practically substituted by calculating the Frobenius norm of the local matrix of dynamics and is achievable in real time on any HW today. Thus, it is also important for autonomous and embedded learning systems implemented by progressive technologies such as FPGA.

APPENDIX

The time-variant state-space representation of weight-updates for IPLNAs given in (2) is to be derived, considering (1) with the details in left column of Table I and respecting the proper vector multiplications, via the steps as follows:
\[ \frac{\partial \dot{Q}(k)}{\partial w} = \frac{\partial}{\partial w} \left( \frac{1}{2} \varepsilon^T(k) \right) \]
\[ = e(k) \frac{\partial}{\partial w} (y(k) - w(k)^T \cdot \mathbf{g}(x(k))) \]
\[ = -g(x(k)) \cdot e(k) \] (28)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
moments can be here rewritten in the following matrix form:

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
-(\beta_1 - 1)g(x, k)g(x, k)^T & 0 & (\beta_1 - 1)g(x, k)g(x, k)^T & \eta(k - 1) \\
\end{bmatrix} \cdot \eta(k).
\]

(36)

that returns in the form of (2) as follows:

\[
w(k+1) = (I - \eta(k) \cdot g(x(k)) \cdot g(x(k))^T)
\]

\[
\cdot \left(\frac{\partial Q}{\partial w} \right) + \eta(k) \cdot y(k) \cdot g(x(k)).
\]

(30)

State-space representation of the weight-update system for ADAM leading to (3) and (4) derives as follows:

\[
w(k+1) = w(k) - \eta(k) \cdot m(k)
\]

\[
m(k+1) = \beta_1 \cdot m(k) + (1 - \beta_1) \cdot \frac{\partial Q}{\partial \omega}
\]

\[
\cdot w(k) - \eta(k) \cdot g(x(k)) \cdot (y(k) - g(x(k))^T \cdot w(k))
\]

\[
\cdot (\beta_1 - 1) g(x, k) \cdot y(k).
\]

(31)

\[
\begin{bmatrix}
w(k) \\
w(k + 1) \\
m(k) \\
m(k + 1)
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
-(\beta_1 - 1)g(x, k)g(x, k)^T & 0 & (\beta_1 - 1)g(x, k)g(x, k)^T & \eta(k - 1) \\
\end{bmatrix} \cdot
\begin{bmatrix}
g(x, k) \\
y(k)
\end{bmatrix}
\]

(35)

where \( A(k) \) is in detail for ADAM in (36), as shown at the top of the page.

So the ADAM weight-update system (31) and (32) turns into the already shown state-space representation for IPLNAs as

\[
\xi(k+1) = A(k) \cdot \xi(k) + B(k) \cdot u(k)
\]

(37)

where the state vector \( \xi \), local matrix of dynamics \( A \), output gain matrix (or vector) \( B \), and input term \( u \) change accordingly. Therefore, further validity and application of BIBS stability applied to the weight convergence are valid and straightforward.

References

[1] W. F. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, “Feedforward neural networks with random weights,” in Proc. 11th IAPR Int. Conf. Pattern Recognit., Conf. B, Pattern Recognit. Methodol. Syst., Aug./Sep. 1992, pp. 1–4, doi: 10.1109/ICPR.1992.201708.

[2] Y.-H. Pao and Y. Takefuji, “Functional-link net computing: Theory, system architecture, and functionalities,” Computer, vol. 25, no. 5, pp. 76–79, May 1992, doi: 10.1109/2.144401.

[3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” Neurocomputing, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[4] V. J. Mathews and G. L. Sicuranza, Polynomial Signal Processing. New York, NY, USA: Wiley, 2000.

[5] A. Carini and G. L. Sicuranza, “BIBO-stable recursive functional link polynomial filters,” IEEE Trans. Signal Process., vol. 65, no. 6, pp. 1595–1606, Mar. 2017, doi: 10.1109/TSP.2016.2641395.

[6] V. Patel, S. B. Bhattacharjee, and N. V. George, “Convergence analysis of adaptive exponential functional link network,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 2, pp. 882–891, Feb. 2021, doi: 10.1109/TNNLS.2020.2979688.

[7] M. M. Gupta, L. Jin, and N. Homma, Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory. New York, NY, USA: Wiley, 2003.

[8] M. M. Gupta et al., “Higher order neural networks: Fundamental theory and applications,” in Artificial Higher Order Neural Networks for Computer Science and Engineering: Trends for Emerging Applications, M. Zhang, Ed. Hershey, PA, USA: IGI Global, 2010, pp. 397–422, doi: 10.4018/978-1-61520-711-4.ch017.

[9] I. Bukovsky, Z.-G. Hou, J. Bila, and M. M. Gupta, “Foundation of notation and classification of nonconventional static and dynamic neural units,” in Proc. 6th IEEE Int. Conf. Cognit. Informat., Aug. 2007, pp. 401–407, doi: 10.1109/CQGNIF.2007.4341916.

[10] I. Bukovsky and N. Homma, “An approach to stable gradient-descent adaptation of higher order neural units,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 9, pp. 2022–2034, Sep. 2017, doi: 10.1109/TNNLS.2016.2572310.

[11] E. D. Sontag, “Input to state stability: Basic concepts and results,” in Nonlinear and Optimal Control Theory, vol. 1932. Berlin, Germany: Springer, 2008, pp. 163–220.

[12] Z. Wang and D. Liu, “Stability analysis for a class of systems: From model-based methods to data-driven methods,” IEEE Trans. Ind. Electron., vol. 61, no. 11, pp. 6463–6471, Nov. 2014, doi: 10.1109/TIE.2014.2308146.

[13] A. Benveniste, P. Priouret, and M. Métévier, Adaptive Algorithms and Stochastic Approximations. Berlin, Germany: Springer, 1990, doi: 10.1007/978-3-642-75894-2.

[14] D. P. Mandic, S. Kanna, and A. G. Constantinides, “On the intrinsic relationship between the least mean square and Kalman filters [lecture notes],” IEEE Signal Process. Mag., vol. 32, no. 6, pp. 117–122, Nov. 2015, doi: 10.1109/MSLP.2015.2461733.

[15] Y.-S. Choi, H.-C. Shin, and W.-J. Song, “Robust regularization for normalized LMS algorithms,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 627–631, Aug. 2006, doi: 10.1109/TCSII.2006.877280.

[16] X. Hong and Y. Gong, “A constrained recursive least squares algorithm for adaptive combination of multiple models,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN). Jul. 2015, pp. 1–6, doi: 10.1109/IJCNN.2015.7280298.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[18] D. P. Mandic, “A generalized normalized gradient descent algorithm,” IEEE Signal Process. Lett., vol. 11, no. 2, pp. 115–118, Feb. 2004, doi: 10.1109/2941.131649.

[19] Q. Song, Y. Wu, and Y. C. Soh, “Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain,” IEEE Trans. Neural Netw., vol. 19, no. 11, pp. 1841–1853, Nov. 2008, doi: 10.1109/TNN.2008.2019293.

[20] W.-P. Ang and B. Farhang-Boroujeny, “A new class of gradient adaptive step-size LMS algorithms,” IEEE Trans. Signal Process., vol. 49, no. 4, pp. 805–810, Apr. 2001, doi: 10.1109/78.912925.

[21] V. J. Mathews and Z. Xie, “A stochastic gradient adaptive filter with gradient adaptive step size,” IEEE Trans. Signal Process., vol. 41, no. 6, pp. 2075–2087, Jun. 1993, doi: 10.1109/78.218137.

[22] Sebastian Ruder. (Jan. 19, 2021). An Overview of Gradient Descent Optimization Algorithms. Accessed: Mar. 8, 2016. [Online]. Available: https://ruder.io/optimizing-gradient-descent/