Common Nature of Learning Between Back-Propagation and Hopfield-Type Neural Networks for Generalized Matrix Inversion With Simplified Models

Yunong Zhang, Member, IEEE, Dongsheng Guo, and Zhan Li, Student Member, IEEE

Abstract—In this paper, two simple-structure neural networks based on the error back-propagation (BP) algorithm (i.e., BP-type neural networks, BPNNs) are proposed, developed, and investigated for online generalized matrix inversion. Specifically, the BPNN-L and BPNN-R models are proposed and investigated for the left and right generalized matrix inversion, respectively. In addition, for the same problem-solving task, two discrete-time Hopfield-type neural networks (HNNs) are developed and investigated in this paper. Similar to the classification of the presented BPNN-L and BPNN-R models, the presented HNN-L and HNN-R models correspond to the left and right generalized matrix inversion, respectively. Comparing the BPNN weight-updating formula with the HNN state-transition equation for the specific (i.e., left or right) generalized matrix inversion, we show that such two derived learning-expressions turn out to be the same (in mathematics), although the BP and Hopfield-type neural networks are evidently different from each other a great deal, in terms of network architecture, physical meaning, and training patterns. Numerical results with different illustrative examples further demonstrate the efficiency of the presented BPNNs and HNNs for online generalized matrix inversion and, more importantly, their common natures of learning.


I. INTRODUCTION

IN RECENT years, online solution of the generalized inverse (also well-known as the Moore–Penrose generalized inverse) has been considered as one of the basic problems widely encountered in various science and engineering fields [1]–[10], e.g., robotics, signal processing, pattern recognition, and optimization. Generally speaking, if the real matrix \( A \in \mathbb{R}^{m \times n} \) is of full-rank, i.e., \( \text{rank}(A) = \min\{m, n\} \), then the unique generalized inverse \( A^\dagger \) for matrix \( A \) can be given as [5]–[10]

\[
A^\dagger := \begin{cases} 
(A^TA)^{-1}A^T, & \text{if } m > n \\
A^{-1}, & \text{if } m = n \\
A^T(AA^T)^{-1}, & \text{if } m < n
\end{cases} \tag{1}
\]

where superscript \( ^T \) denotes the transpose of a matrix/vector and superscript \( ^{-1} \) denotes the inverse of a matrix. In addition, the upper, middle, and lower parts of (1) correspond to the left generalized inverse, inverse, and the right generalized inverse, respectively. Besides, for the full-rank rectangular matrix \( A \) (i.e., \( m \neq n \)), the generalized inverse \( A^\dagger \) of matrix \( A \) satisfies the following equations [5]–[10]:

\[
\begin{align*}
A^\dagger AA^\dagger &= A^\dagger, & \text{if } m > n \\
A^\dagger AA^\dagger &= A^\dagger, & \text{if } m < n
\end{align*}
\]

It is worth pointing out here that, if \( \text{rank}(A) < \min\{m, n\} \) (i.e., matrix \( A \) is rank-deficient), \( A^\dagger A \) (or \( AA^\dagger \)) is singular [6], [7], [10]. Thus, \( A^\dagger \) cannot be obtained directly via (1). In this paper, we only consider and investigate the generalized inverses of full-rank matrices, and \( A \) hereafter is defined as a real full-rank matrix (while the generalized inverses of rank-deficient matrices can be a future research direction).

Due to the important role of the generalized inverse, many efforts have been contributed to the solution of the generalized inverse, and subsequently many algorithms (such as Newton’s iteration) have been proposed and investigated for online generalized (full-rank) matrix inversion [5], [11]–[13]. However, such serial-processing numerical algorithms may not be efficient enough for large-scale online applications. Because of the superior performances in large-scale online applications, parallel computation methods have thus been extensively developed, analyzed, and implemented for online linear or nonlinear problems solving [6]–[10], [14]–[20]. Being one of the most important parallel and distributed processing methods, the neural-dynamic approach based on recurrent neural networks has been viewed as a powerful alternative for online generalized matrix inversion [6]–[10], owing to its convenience of hardware implementation.

According to the cases of \( m > n \) and \( m < n \), in this paper, we introduce two simple-structure neural networks based on the error back-propagation (BP) algorithm, i.e., BP-type neural