

Time Series Forecasting using Polynomial Artificial Neural Networks

Yoltic Jassiel García Guzmán

Faculty of Engineering

Universidad La Salle

Mexico City, Mexico

yjgg@lasallistas.org.mx

Abstract—The Polynomial Artificial Neural network is a different approach to a design of a universal structure identifier, which uses a statistical method to adapt to a structure by means of maximum likelihood techniques, giving an efficient design for data forecasting in high performance computing where the model is complex in nature.

Index Terms—PANN; Time series forecasting; Neural Networks

I. INTRODUCTION

Artificial Neural Networks, in general, are function approximations. They identify patterns in a dataset according to their relationship. An ANN is trained to correctly classify the test information, and then it is fed with new information to make an attempt of forecasting the correct answer. Polynomial neural networks, unlike ANNs, can handle the interaction between multiple layers of a neuron. It is considered a polynomial paradigm because of the multiple variables and parameters the PANN uses to transform the input into a series of weighted combinations using time-delayed dynamic periodic activation functions (Zjavka, 2011).

Polynomial approximations are used in the fields of data mining, knowledge discovery, prediction, complex systems modeling, optimization and pattern recognition. They were first introduced with the introduction of Group method of data handling (GMDH), a mathematical model of multi-parametric data sets that feature fully automatic structural and parametric optimization of models (“Group method of data handling,” 2020), in the hope to capture the complexity of a process and decompose it into simple relationships described by a processing function of a neuron. Since then, other types of approximations appeared, such as NARMAX, High Order Neural Network, Non-linear interconnection, and Polynomial Neural Networks.

In section II we describe the PANN functionality, in section III we discuss the methodology used to train and test a PANN, showing the corresponding results in section V, to later analyze the output and discuss the results in section VI.

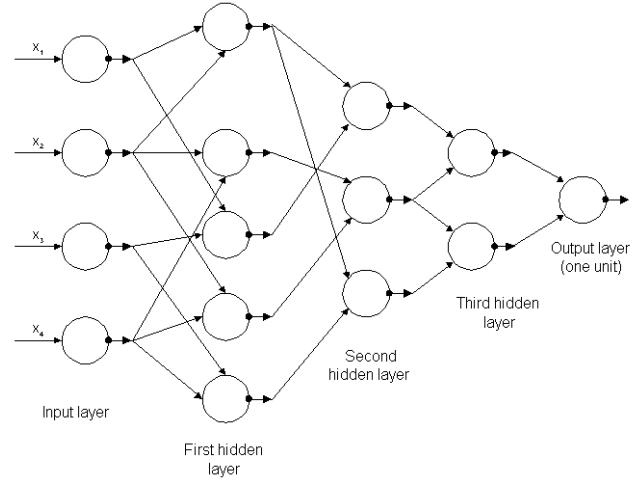


Fig. 1.1 - GMDH polynomial neural network

II. POLYNOMIAL ARTIFICIAL NEURAL NETWORKS

The PANN model can be described as the following equation:

$$\hat{y}_k = [\phi(x_{1,k}, x_{2,k}, \dots, x_{n_i,k}, x_{1,k-1}, x_{2,k-1}, \dots, x_{n_i,k-n_1}, \dots, y_{k-1}, y_{k-2}, \dots,$$

Where: - $\hat{y}_k \in \mathbf{R}$ is the approximation of a function, or the network output, - $\phi(x, y) \in \mathbf{R}$ is a non linear function, - $x_i \in X$ are the inputs, - $i = 1, \dots, n_i$; n_i = number of inputs, - $y_{k-j} \in Y$ are the previous values of the input, - $j = 1, \dots, n_2$, n_1 are the number of delays in the input, - n_2 is the number of delays in the output, - X, Y are compact spaces of \mathbf{R}

The non-linear $\phi(z)$ function is described by:

$$[\phi(z)]_{\phi_{\min}}^{\phi_{\max}} = \begin{cases} \phi_{\max} & \phi(z) \geq \phi_{\max} \\ \phi(z) & \phi_{\min} < \phi(z) < \phi_{\max} \\ \phi_{\min} & \phi(z) \leq \phi_{\min} \end{cases}$$

Where ϕ_{\max} and ϕ_{\min} are maximum and minimum limits respectively.

In summary:

$$z = \{x_{1,k}, x_{2,k}, \dots, x_{n_1,k}, \dots, y_{k-1}, y_{k-2}, \dots, y_{k-n_2}\} = \{z_1, z_2, z_3, \dots,$$