A Neural Network Model for Factor Analysis

Abdelaziz Hamad Elawad, Mohammed Hassan Elzubair, Bahrom Sanugi

***Abstract*— This paper was focused on the development of a Neural Network model to solve factor analysis problems. The factor analysis concept has been presented and Multi-layer Neural Network trained model has been proposed. The performance of the proposed neural network has been measured on a numerical example.. The short time steps and the small number of steps involved in the process of the result showed that the neural network is certainly a better choice for factor analysis problem. This is the first time a factor analysis problem has been resolved by using neural network and the results are very encouraging for further investigation**

***Index Terms*— Factor Analysis, Multi-layer neural network**

I. Introduction

The idea of applying neural network for factor analysis emerges from application of neural network for the work conducted by Sanugi et. al. (1991) in the energy research and Abdelaziz Hamad Thesis(200)2) in the job scheduling problem research. There are similarities between factor analysis and job scheduling because both of them are considered very difficult to solve if the system is large and in both cases we have to do scheduling.

Factor analysis is a model for real –valued data in which correlations are explained by postulating the presence of one or more factors. These factors play the role of hidden variable, which are not directly observable, but allow the dependencies between the visible variable to be expressed in a convenient way. (Everitt, 1984) gives a good introduction to Latent or hidden variable models in general, and to factor analysis in particular. These models are widely used in psychology, social sciences and pure sciences such as Chemistry.

Many definitions are offered in the literature for factor analysis. Reymont and Joreskog (1993) provided a comprehensive definition:

“ Factor analysis is a generic term that we use to describe a number of methods designed to analyze interrelationships within a set of variables or objects (resulting in) the construction of a few hypothetical variable (or objects) called factors, that are supposed to contain the essential information in larger set of observed variable or objects that reduces the overall complexity of the data by taking advantage of inherent interdependencies a small number of factors will usually account for approximately the same a mount of information as do the much larger set of original observations.”

Manuscript received April 09, 2014.

Abdelaziz Hamad Elawad, Taif University, Kingdom of Saudi Arabia, Faculty of sciences and Arts, Department of Mathematics.

Mohammed Hassan Elzubair, Taif University, Kingdom of Saudi Arabia, Faculty of sciences and Arts, Department of Mathematics.

Bahrom Sanugi, Universiti Sains Islam, Malaysia

The procedures for factor analysis were first developed early in the twentieth century by Spearman (1904). However, due to the complicated and time –consuming steps involved in the process, factor analysis was inaccessible to many researchers until both computers and user-friendly statistical software packages became widely available (Thompson & Dennings, 1993). For the above reason the neural network approach is considered a good choice for factor analysis.

Factor analysis is a generic term for a family of statistical techniques concerned with the reduction of a set of observable variables in terms of a small number of latent factors. It has been developed primarily for analyzing relationships among a number of measurable entities (such as survey items or test scores). The underlying assumption of factor analysis is that there exists a number of unobserved latent variables (or “factors”) that account for the correlations among observed variables, such that if the latent variables are partialled out or held constant, the partial correlations among observed variables all becomes zero. In other words, the latent factors determine the values of the observed variables.

We organized this paper as follows. Section 1 Introduction, Section 2 described the purpose of the factor analysis. Section 3 outlines the steps of factor analysis and Section 4 provided an alternative solution of factor analysis using neural network with a numerical example and Section 5 the result of the paper

II. Purpose of Factor Analysis

The primary purpose of factor analysis is data reduction and summarization. Factor analysis has been widely used, especially in behavioral sciences, to assess the construct validity of a test or a scale. For example a psychologist developed a new battery of 15 subtests to measure three distinct psychological constructs and wanted to validate that battery. A sample of 300 subjects was drawn from the population and measured on the battery of 15 subtests. The 300 by 15 data matrix was submitted to a factor analysis procedure. The output from that procedure was a 15 by 3 factor loading matrix, which represented the relationships among the observed variables (the 15 subtests) and the 3 latent factors. The number of factors extracted and the pattern of relationships among the observed variables and the factors provided the researcher with information on the construct validity of the test battery.

III. Outline of Step for Factor Analysis

Normally, they are many step for doing factor analysis , The following are the steps used for factor analysis. Preparing data, selecting a factor model, Estimating communalities, Determining the Number of factors, The rotating of factor, and estimating factor scores.

IV. Neural Network and Factor Analysis

A. Outline of Step for Neural Network

There is no need to go through all the above steps for doing factor analysis by neural network because neural network gives us unique factors. The following are the steps used for training neural network.

(i) Choose small system as example for training neural network.

(ii) Based on the vector representation of the original data matrix and loading weights matrix, both of them are used as input data for neural network.

(iii) The row matrix is used as desired output for neural network. (We use the original data and not the covariance or correlation matrices).

(iv) After training, the neural network can be used direct for factoring

B. A Neural Network Model for Factor Analysis

We arbitrarily construct three linear equations of the type.

$$d_{ik} = x_{i1}y_{1k} + x_{i2}y_{2k} = D = X \times Y. \qquad (1)$$

With $p \times k$ data matrix D, $p \times j$ row matrix X and $j \times k$ column matrix (loading matrix) Y contains the weights on the recognition connections. The column factors y_{i1}, y_{i2} and y_{i3} and for the row factors x_{i1} and x_{i2} are arbitrarily chosen. The equation is shown in matrix form as equation (2).

$$\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \\ x_{51} & x_{52} \\ x_{61} & x_{62} \\ x_{71} & x_{72} \\ x_{81} & x_{82} \\ x_{91} & x_{92} \\ x_{10\,1} & x_{10\,2} \end{pmatrix} \times \begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{22} \\ d_{31} & d_{32} & d_{33} \\ d_{41} & d_{42} & d_{43} \\ d_{51} & d_{52} & d_{53} \\ d_{61} & d_{62} & d_{63} \\ d_{71} & d_{72} & d_{73} \\ d_{81} & d_{82} & d_{83} \\ d_{91} & d_{92} & d_{93} \\ d_{10\,1} & d_{10\,2} & d_{10\,3} \end{pmatrix} \qquad (2)$$

Factor analysis starting with the data matrix, and attempt to solve the reverse of what was done in generating the data matrix of equation (2). In this section we try to use neural network to find the row matrix X. In this section we assume that the data matrix is positive and the loading matrix Y is arbitrarily chosen.

The neural network that is proposed for factor analysis must be organized into three layers of processing units. There is an input layer, a hidden layer, and an output layer. The number of units in the input and output layers is dictated by the specific representation adopted for the factor analysis. In the proposed representation, the input layer contains the information describing data matrix and loading matrix in the form of a vector of continuous values. For example in equation (2) the input units contains the following:

$$unit1 = \frac{d_{i1}}{100}, \qquad (03)$$

$$unit2 = \frac{d_{i2}}{100}, \qquad (04)$$

$$unit3 = \frac{d_{i3}}{100}, \qquad (05)$$

$$unit4 = \frac{d_{i1}}{y_{11}}, \qquad (06)$$

$$unit5 = \frac{d_{i1}}{y_{21}} \qquad (07)$$

$$unit6 = \frac{d_{i2}}{y_{12}}, \qquad (08)$$

$$unit7 = \frac{d_{i2}}{y_{22}} \qquad (09)$$

$$unit8 = \frac{d_{i3}}{y_{13}} \qquad (10)$$

$$unit9 = \frac{d_{i3}}{y_{23}} \qquad (11)$$

$$unit10 = \frac{y_{11}}{10} \qquad (12)$$

$$unit11 = \frac{y_{21}}{10} \qquad (13)$$

$$unit12 = \frac{y_{12}}{10} \qquad (14)$$

$$unit13 = \frac{y_{22}}{10} \qquad (15)$$

$$unit14 = \frac{y_{13}}{10} \qquad (16)$$

$$unit15 = \frac{y_{23}}{10} \qquad (17)$$

The number of units in the hidden layer is selected by trial and error during the training phase. The final network for our example consists of 12 units in its hidden layer and 2 units in the output layer. Therefore it is known as 15-12-2 network.

C. Training neural network

To illustrate how the neural network is trained equation (18) shows 10×3 positive numerical data matrix, 10×2-row matrix and 2×3-column arbitrary matrix (loading matrix). That serves as training example for neural network. Matrix equation (16) are converted first into their vector representations by using the set of equations (1-15). The result of this pre-processing stage is presented in Table.1.To train the neural network, each vector with their desired output is presented individually at the input layer and output layer of the neural network respectively. Training is considered complete after an average of 50,000 cycles using a 15-12-2 configuration. A cycle is concluded after the network has been exposed once, in the course of the back propagation algorithm, to each one of the available training patterns. The desired output and the neural network out put after training are given in Table.2. The desired out put and the neural network are same and these indicate that the neural network was trained. The trained neural network is used to find the solution for new and similar problem.

$$\begin{pmatrix} 0 & 4 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 4 & 3 \\ 5 & 4 \\ 6 & 5 \\ 7 & 8 \\ 8 & 2 \\ 9 & 5 \end{pmatrix} \times \begin{pmatrix} 2 & 5 & 2 \\ 1 & 10 & 5 \end{pmatrix} = \begin{pmatrix} 4 & 40 & 20 \\ 3 & 15 & 7 \\ 4 & 10 & 4 \\ 6 & 15 & 6 \\ 11 & 50 & 23 \\ 14 & 65 & 30 \\ 17 & 80 & 37 \\ 22 & 115 & 54 \\ 18 & 60 & 26 \\ 23 & 95 & 43 \end{pmatrix} \quad (18)$$

Table 1: Vector representation of equation 18

<u>Input</u>

0.04	0.40	0.20	0.20	0.40	0.80	0.40	1.00	0.40	0.20	0.10	0.50	1.00	0.20	0.50
0.03	0.15	0.07	0.15	0.30	0.30	0.15	0.35	0.14	0.20	0.10	0.50	1.00	0.20	0.50
0.04	0.10	0.04	0.20	0.40	0.20	0.10	0.20	0.05	0.20	0.10	0.50	1.00	0.20	0.50
0.06	0.15	0.06	0.30	0.60	0.30	0.15	0.30	0.12	0.20	0.10	0.50	1.00	0.20	0.50
0.11	0.50	0.23	0.55	1.10	1.00	0.50	1.15	0.46	0.20	0.10	0.50	1.00	0.20	0.50
0.14	0.65	0.30	0.70	1.40	1.30	0.65	1.50	0.60	0.20	0.10	0.50	1.00	0.20	0.50
0.17	0.80	0.37	0.85	1.70	1.60	0.80	1.85	0.74	0.20	0.10	0.50	1.00	0.20	0.50
0.22	1.15	0.54	1.10	2.20	2.30	1.15	2.70	1.08	0.20	0.10	0.50	1.00	0.20	0.50
0.18	0.60	0.26	0.90	1.80	1.20	0.60	1.30	0.52	0.20	0.10	0.50	1.00	0.20	0.50
0.23	0.95	0.43	1.15	2.30	1.90	0.95	2.15	0.86	0.20	0.10	0.50	1.00	0.20	0.50

Table (2): Desired output and neural output

Desired Output		Neural network output	
0.000000	0.400000	0.044453	0.410804
0.100000	0.100000	0.098438	0.064497
0.200000	0.000000	0.199601	0.026366
0.300000	0.000000	0.294602	0.028852
0.400000	0.300000	0.361707	0.262526
0.500000	0.400000	0.501224	0.396703
0.600000	0.500000	0.641421	0.515721
0.700000	0.800000	0.684424	0.779283
0.800000	0.200000	0.804331	0.207361
0.900000	0.500000	0.892248	0.496164

The desired output and neural output are same. This indicates that the neural network train very well.

D. Numerical Example

Equation (19) shows 15×3 positive numerical data matrix, 15×2-row matrix and 2×3-column arbitrary matrix (loading matrix). Equation (19) are converted first into their vector representations by using the set of equations (3 - 16). The result of this pre-processing stage is presented in Table 3 each vector presented individually at the input layer. After an average of 50,000 cycles using a 15-12-2 configuration the output of neural network is given in Table.4.

$$\begin{pmatrix} 0 & 4 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 4 & 3 \\ 5 & 4 \\ 6 & 5 \\ 7 & 8 \\ 8 & 2 \\ 9 & 5 \\ x_{11\,1} & x_{11\,2} \\ x_{12\,1} & x_{12\,2} \\ x_{13\,1} & x_{13\,2} \\ x_{14\,1} & x_{14\,2} \\ x_{15\,1} & x_{15\,2} \end{pmatrix} \times \begin{pmatrix} 2 & 5 & 2 \\ 1 & 10 & 5 \end{pmatrix} = \begin{pmatrix} 4 & 40 & 20 \\ 3 & 15 & 07 \\ 4 & 10 & 04 \\ 6 & 15 & 06 \\ 11 & 50 & 23 \\ 14 & 65 & 30 \\ 17 & 80 & 37 \\ 22 & 115 & 54 \\ 18 & 60 & 26 \\ 23 & 95 & 43 \\ 05 & 35 & 17 \\ 10 & 40 & 18 \\ 17 & 95 & 45 \\ 22 & 100 & 46 \\ 13 & 100 & 49 \end{pmatrix} \qquad (19)$$

Table .3: Vector representation of equation (19)

0.04	0.40	0.20	0.20	0.40	0.80	0.40	1.00	0.40	0.20	0.10	0.50	1.00	0.20	0.50
0.03	0.15	0.07	0.15	0.30	0.30	0.15	0.35	0.14	0.20	0.10	0.50	1.00	0.20	0.50
0.04	0.10	0.04	0.20	0.40	0.20	0.10	0.20	0.05	0.20	0.10	0.50	1.00	0.20	0.50
0.06	0.15	0.06	0.30	0.60	0.30	0.15	0.30	0.12	0.20	0.10	0.50	1.00	0.20	0.50
0.11	0.50	0.23	0.55	1.10	1.00	0.50	1.15	0.46	0.20	0.10	0.50	1.00	0.20	0.50
0.14	0.65	0.30	0.70	1.40	1.30	0.65	1.50	0.60	0.20	0.10	0.50	1.00	0.20	0.50
0.17	0.80	0.37	0.85	1.70	1.60	0.80	1.85	0.74	0.20	0.10	0.50	1.00	0.20	0.50
0.22	1.15	0.54	1.10	2.20	2.30	1.15	2.70	1.08	0.20	0.10	0.50	1.00	0.20	0.50
0.18	0.60	0.26	0.90	1.80	1.20	0.60	1.30	0.52	0.20	0.10	0.50	1.00	0.20	0.50
0.23	0.95	0.43	1.15	2.30	1.90	0.95	2.15	0.86	0.20	0.10	0.50	1.00	0.20	0.50
0.05	0.35	0.17	0.25	0.50	0.70	0.35	0.85	0.34	0.20	0.10	0.50	1.00	0.20	0.50
0.10	0.40	0.18	0.50	1.00	0.80	0.40	0.80	0.36	0.20	0.10	0.50	1.00	0.20	0.50
0.17	0.95	0.45	0.85	1.70	1.90	0.95	2.25	0.90	0.20	0.10	0.50	1.00	0.20	0.50
0.22	1.00	0.46	1.10	2.20	2.00	1.00	2.30	0.92	0.20	0.10	0.50	1.00	0.20	0.50
0.13	1.00	0.49	0.65	1.30	2.00	1.00	2.45	0.98	0.20	0.10	0.50	1.00	0.20	0.50

Table 4: Neural output and actual output

Neural network output	Actual output
0.032736 0.414327	0 4
0.079709 0.059216	1 1
0.194076 0.020463	2 0
0.315406 0.023836	3 0
0.373179 0.277492	4 3
0.499151 0.398441	5 4
0.627957 0.506755	6 5
0.689117 0.797884	7 8
0.800554 0.208480	8 2
0.893943 0.497830	9 5
0.057241 0.296207	1 3
0.447890 0.129378	4 2
0.393223 0.757278	5 7
0.846404 0.585483	8 6
0.170915 0.830749	2 9

Table 4 shows the result of neural network output is very close to the actual output. This indicates that the neural network give us a reasonably accurate solution to the problem in comparison to conventional techniques for a small system. But for bigger system, the neural network approach will still be useful whereas in conventional techniques the process is very highly complicated.

V. Conclusions

In this paper, The factor analysis concept has been presented and Multi-layer neural network model has been proposed. The performance of the proposed network has been measured on a numerical example. This is the first time a factor analysis problem is resolved by using neural network and the results are very encouraging for further investigation. The number of the steps and the short time steps involved in the process of the result shows that the neural network is certainly a better choice for factor analysis problem.

Acknowledgment:

The main author would like to thank professor bahrom sanugi for continuing help in the research area especially the ideas for applying the neural network for factor analysis problems and thanks also go to the Head Department of Mathematics Riana Branch , Taif university Kingdom of Saudi Arabia for their Facilities

References:

[1] Everitt, B. S. (1984). An Introduction to Latent Variable Models. London: Chapman and Hall.

[2] Reymont, R., Joreskog, K.G. (1993). Applied Factor Analysis in the Natural Sciences. NewYork: Cambridge University Press.

[3] Sanugi, B. and Buang, S. (1991)." PPGs Modified Software Manual (Version1.0) for Power Station Siting Study for Tenaga Nasional Berhad (A Report on Sofware Development) UTM.

[4] Spearman, C. (1904). " General Intelligence, " Objectively Determined and Measured. American Journal of Psychology. 15, 201-293

[5] Thompson, B. and Dennings, B. (1993). "The Unnumbered Graphics Scale as a Data Collection Method: An Investigation Comparing Three Measurement Strategies in the Context of Q-Technique Factor Analysis." Paper Presented at the Annual Meeting of the Mid-South Educational Research Association, New Orleans.

[6] Abdelaziz Hamad PhD Thesis (2002) "Application of neural network for solving job scheduling problems, UTM, Malaysia

[7] Abdelaziz Hamad and Bahrom Sanugi (2011) . Neural Network and Scheduling. Germany : Lap Lambert Academic Publishing.