Factor Analysis Using Two Stages Neural Network Architecture

Sandeep Kumar, Deepak Kumar, and Rashid Ali

***Abstract*—Factor Analysis is the process of finding a suitable representation of the data in terms of lesser number of variables. There are a number of application areas where factor analysis is widely used e.g. signal processing, statistics, stock marketing, forecasting, approximation, compression, security, medical sciences etc. In this paper we will show that a very effective factor analysis scheme can be developed using a feed forward neural network. Our approach has the advantage of being able to analyze very large data sets while preserving the nature of the data.**

***Index Terms*—Factor analysis, neural network, BPN, learning, information gain.**

I. Introduction

A. Factor Analysis

Factor Analysis transforms larger dimension data (variables) to lower dimension data (factors). Factor analysis is performed by examining the pattern of correlations between the observed measures. Measures that are highly correlated either positively or negatively are likely influenced by the same factors, while those that are relatively uncorrelated are likely influenced by different factors. As given in BMDP Documentation [9] the goal of factor analysis is to express p variables ($v_1, v_2, \ldots, v_p$) by m factors ($f_1, f_2, \ldots, f_m$), where m<<p (m is considerably smaller than p). Mathematically, the model can be written as

$$X=F \bullet A$$

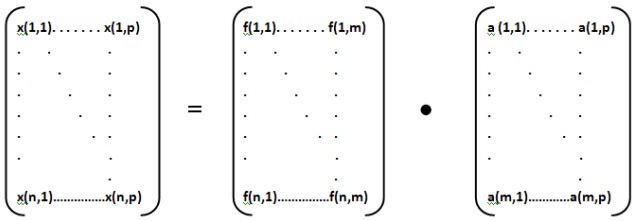

Fig. 1. Factor analysis in matrix form

where,

● :matrix multiplication operator

X :data matrix for n cases, each row represents a different case

F: Factor Score matrix

A: Factor loading matrix

Manuscript received July 17, 2012; revised September 17, 2012.This work was supported in part by Aligarh Muslim University.

Sandeep Kumar and Deepak Kumar are with the Tata Consultancy Services Limted, Pune India (e-mail: sandeep25789@gmail.com; deepakkumar@zhcet.ac.in).

Rashid Ali is with the Aligarh Muslim University. Uttar Pradesh,Aligarh India (e-mail: rashidaliamu@rediffmail.com).

Let's summarize the terminology we use. Data to be analyzed are in matrix X. Its columns represent variables, whereas its rows represent cases. The factor analysis comes out from the generic thesis saying that variables, we can observe, are just the effect of the factors, which are the real origin [8]. So we focus on factors. We also try to keep number of factors as low as possible, so we can say, "Reducing variables to factors".

The result is the pair of matrices. Matrix of factor scores F expresses the input data by factors instead of variables. Matrix of factor loadings A defines the relation between variables and factors, i.e. each row in A defines one particular factor.

B. Applications of Factor Analysis

1. Data Analysis [6]
2. Education Research[1]
3. Psychology
4. marketing[2][3][4]
5. Medical Science[5]

C. Neural Network

Neural Network is the network of neurons (processing units) which can be connected in some particular topology so as to solve the complex problems. The inherited feature of a neural network is parallel processing (all neurons can work in parallel).

Feed-forward neural networks are easy to analyze and design. In this paper we will be discussing the implementation of factor analysis scheme using feed forward neural network. The topological architecture of neural network is given in Fig. 2.

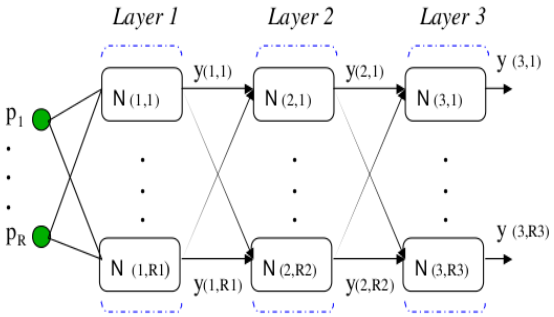

Fig. 2. Topological structure of a feed forward neural network

Layer1, where input is applied, is called Input Layer. Layer3, where output is present, is called output layer. Layer2, which is an intermediate layer between input and output, is called hidden layer. We can have multiple hidden layers in a feed-forward neural network. The edge connecting two neurons is called a synaptic and each synaptic has its weight. As a signal pass through a synaptic it gets multiplied by the weight of that synaptic.

D. Learning

As we grow, we learn from our experience. Similarly, we can make the neural network to learn (learning means setting the weights of synaptics) by supplying some input set and corresponding output set. Such sets are called examples and this process of making the neural network to learn is called training.

There are a number of algorithms to train a neural network and selection of algorithms highly depends upon the neural network topology. In case of Feed-forward neural network we generally use Back propagation learning so we will use the same learning in factor analysis.

E. Related Work

A number of literatures have been published for developing factor analysis using different techniques. In [10] and [11], neural network based binary factor analysis has been discussed using Hope-filed like neural network. In [12]and [15] formal concept analysis is used for factor analysis. Some other neural network based schemes for factor analysis have been discussed in [13],[14] and [16].

After Finishing Introduction, we will discussed the organization and architecture of the proposed system in Section-II. In Section-III we will discuss the metrics that can be used to evaluate the performance of a factor analysis scheme and hence the proposed system. Section-IV dealt with the result of simulation work that we have carried out with our proposed system. Finally in Section-V detailed list of the entire referenced article is given.

II. Factor Analysis using Two Stage Neural Network

In this section we will discuss about our proposed system for factor analysis using two different neural networks.

A. Organization of System

The system comprised of two neural networks (viz. as two stages), one network to factorize the given data and second one to reconstruct the original data from the factorized data.

a) Stage-I (divider neural network)

Following Fig. 3 shows the schematic diagram of Stage-I. This stage receives original data (variables pattern) and transforms the same into its corresponding factor pattern.

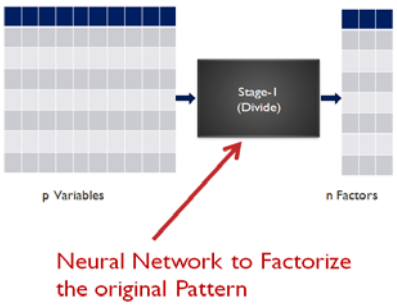

Fig. 3. Divider neural network

b) Stage-II (reconstructor neural network)

Following Fig. 4 shows the schematic diagram of Stage-II. This stage receives factors pattern at its input and transforms the same into its corresponding variable pattern (original data).

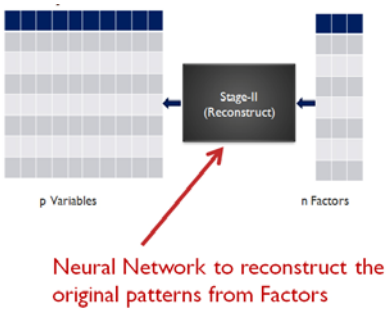

Fig. 4. Reconstructor Neural Network

B. Operational Phases

The proposed system requires a setup procedure before it can be used for actual factor analysis. So, the system will work in two phase:

1. Learning Phase(Training Phase)
2. Working Phase

a) Learning phase (training phase)

In this mode we will give some random inputs variables patterns to stage-I and will get the corresponding factor patterns.

Then we use these patterns to train the stage-II neural network. For training we will supply the output of Stage-I at the input of stage-II and treat the input of Stage-I as the output for Stage-II. Then the BPN algorithm is executed using this given input-output relation (example sets).

We will train the Stage-II neural network until we start getting the output without error (means the output of stage-II must be equal to the input of Stage-I).

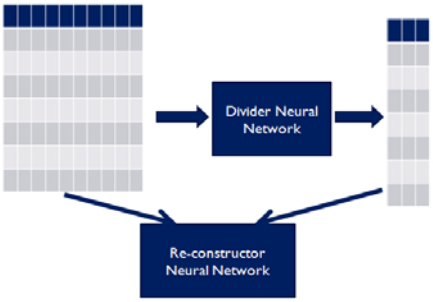

Fig. 5. Setup Mode

b) Working mode

As soon as the reconstructor neural network (i.e. Stage-II) get trained the system is ready to use for actual factorization. Now, when our Reconstructor neural network has been trained properly, we can provide the actual patterns to be factorized at the input of the Stage-I (i.e. Divider Neural Network) and it will give the factors corresponding to this input pattern. And whenever required we can reconstruct the original data from these factors using Reconstructor Neural Network.

III. Evaluation Criteria for the Quality of Factorization

As discussed in [7] the efficiency of factorization system can be measures with the help of parameter termed as informational advantage.

The factorization scheme is optimal when informational advantage received during the transition to coding of signals

is maximal. The calculation of informational advantage is done as follows:

Let Ho = amount of information required to record the signals without the involvement of factors.

H1 = amount of information required to record the signals encoded in factors

The information gain or informational advantage is given by the following equation:

$G=(Ho - H1)/Ho$

The possible rages of values of G is from $-\infty$ to $+\infty$.

Positive value of G means that the encoding of signals by means of factor is more effective compared to the original encoding. On the other hand, If the value of G is close to zero or negative, then factors emerged incorrectly and hence the factorization is not effective and useful.

IV. Simulation and Simulation Results

A. Simulation

We have developed a simulation version of the system. The simulation details are as follow:

- Learning Method: BPN
- For Stage –I
  - Input layers : 10*
  - Hidden layers : 10
  - Output Layers : 3*
- For Stage-II
  - Input layers: 3*
  - Hidden layers: 3
  - Output layers: 10*

Some points to be noted here are

- the number of input layers at Stage-I (or output layers at Stage-II) corresponds to the numbers of variables in an unfactorized pattern.
- Similarly, the number of output layers at Stage-I (or input layers at Stage-II) corresponds to the number of factors in a factorized pattern.
- Finally, the number of hidden layer will affect the time required for learning, encoding (factorization) and decoding (reconstruction) of patterns.

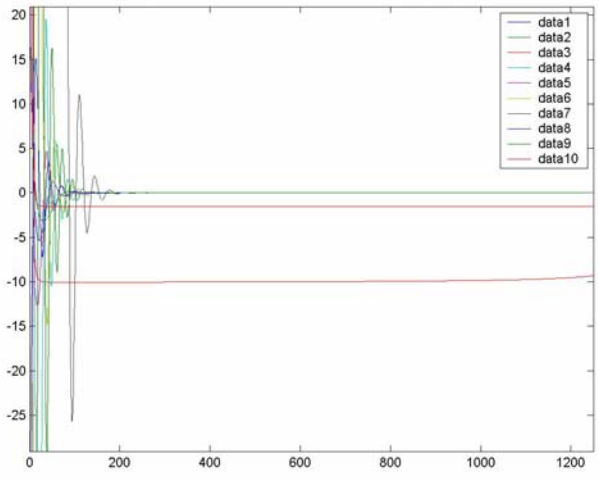

Fig. 6. Error pattern during learning of Stage-II neural Network (For 1200 Samples)

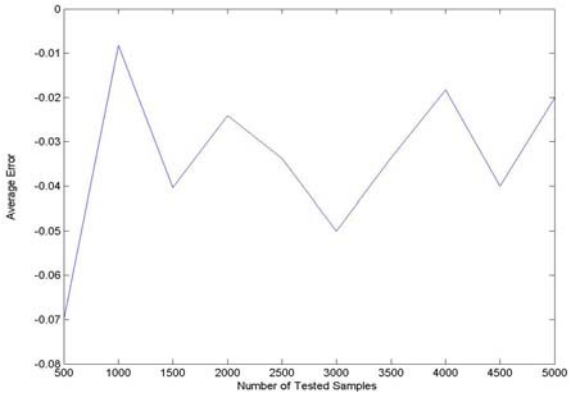

Fig. 7. Average Error in data samples for different number of samples

B. Results

Suppose the space required to store a single variable (Item) is V.S.(Variable size).

Informational Advantage G can be calculated as

Ho=10*10*V.S.

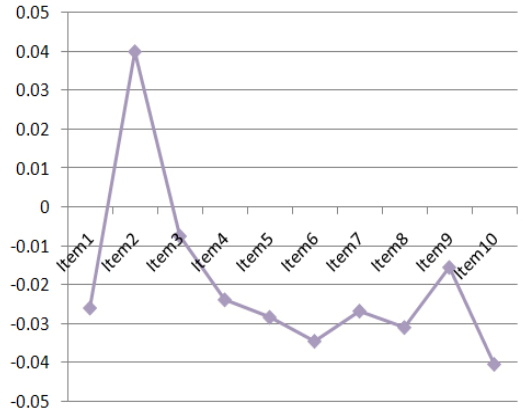

Fig. 8. Average Error in Individual items of original data pattern

H1=10*3*V.S.

G=((100-30)*V.S.)/100 V.S.

G=0.7

G=70%

So, the informational gain is 70%.

Means if original data requires 100MB to store the data the factorized data will require only 30MB of space to store it. This result shows it is very effective scheme.

But as we factorize the data using stage-I and reconstruct the original data again using Stage-II instead of getting exactly original data some error is induced in the data(which is obvious as neural network just approximates the process). That error is illustrated in Fig. 7 and 8. Fig. 7 shows the average error in samples for different numbers of samples taken at a time. Fig. 7 shows the average error in all individual items (variables) of the original data. As par the calculation the average system error comes out to be 0.02 (~2%). Generally, the application areas where the factor analysis is used, we can work with such a small average error.

So, as a conclusion, we would like to say that the system is very much effective in the sense that the degree of compression is very high. Moreover, even the system error is not zero but it is acceptable in the concerned areas like image compression, stock forecasting etc.

REFERENCES

[1] Dijana Oreški and Petra Peharda, "Application of Factor Analysis in Course Evaluation," *Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces*, June 23-26, 2008, Cavtat, Croatia.

[2] B. Xia, B. Pan, and H. Xia, "Appraisal on the Competitiveness of Commercial Bank of China Based on Factor Analysis," *International Symposium on Intelligent Information Technology Application Workshops* 2008 IEEE Computer Society.

[3] Man-zhi Liu and Mei-hua Zhou, "Empirical Study of Market Segmentation Based on Factor Clustering Analysis Model," *Management and Service Science*, 2009. MASS '09, International Conference on Sep 2009.

[4] Ying Yue, Xinghua Ma, and Chenghu Zhang, "Comprehensive Performance Evaluation of the Listed Companies in Coal Mining Industry Based on Factor Analysis and Cluster Analysis," *2010 Asia-Pacific Conference on Wearable Computing Systems IEEE.*

[5] Christiaan Schiepers, Carl K. Hoh, Johan Nuyts, Hsiao-Ming Wu, Michael E. Phelps, and Magnus Dahlbom, "Factor Analysis in Prostate Cancer: Delineation of Organ Structures and Automatic Generation of In- and Output Functions," *IEEE Transactions on Nuclear Science*, vol. 49, no. 5, October, 2002.

[6] Liu Tao, Tian Hongxiang, and Guo Wenyong, "Application of Factor Analysis to a Type Diesel Engine SOA," *2010 International Conference on Measuring Technology and Mechatronics Automation*, IEEE Computer Society.

[7] Pavel Polyakov, Alexander A. Frolov, Dusan Husek, "Comparison of Two Neural Networks Approaches to Boolean Matrix Factorization," *First International Conference on Networked Digital Technologies,* 2009.

[8] Karl ¨*Uberla: Faktorenanalyse* (2nd edition). Springer–Verlag, Berlin–Heidelberg– New York, 1971. ISBN 3-540-04368-3, 0-387-04368-(slovensk´y p˘reklad: Alfa, Bratislava, 1974)

[9] BMDP (Bio-Medical Data Processing). A statistical software package. SPSS. [Online]. Available: http://www.spss.com/

[10] A. A. Frolov, A. M. Sirota, D. H´usek, I. P. Muraviev, P. A. Polyakov, *Binary factorization in Hopfield-like neural networks: Single-step approximation and computer simulations*. 2003.

[11] D. H´usek, A. A. Frolov, H. ˇ Rezankov´a, V. Sn´a˘sel, "Application of Hopfield-like Neural Networks to Nonlinear Factorization," *Proceedings in Computational Statistics Compstat 2002*, Humboldt-Universit¨at Berlin, Germany, 2002.

[12] A. Hotho and G. Stumme, "Conceptual Clustering of Text Clusters," in *Proceedings of FGML Workshop*, pp. 37–45. Special Interest Group of German Informatics Society, 2002.

[13] D. H´usek, A. A. Frolov, I. Muraviev, H. ˇRezankov´a, V. Sn´a˘sel, and P. Polyakov, "Binary Factorization by Neural Autoassociator," *AIA Artifical Intelligence and Applications IASTED International Conference*, Benalm´adena, M´alaga, Spain, 2003.

[14] A. Keprt, "Binary Factor Analysis and Image Compression Using Neural Networks," in *proceedings of Wofex 2003*, Ostrava, 2003.

[15] A. Keprt, "Using Blind Search and Formal Concepts for Binary Factor Analysis," in *Dateso 2004 - proceedings of 4th annual workshop*. Ed. V´aclav Sn´a˘sel, Jaroslav Pokorn´y, Karel Richta, V˘SB Technick´a Univerzita Ostrava, Czech Republic; CEUR WS – Deutsche Bibliothek, Aachen, Germany; 2004, pp. 120-131.

[16] A.M. Sirota, A. A. Frolov, D. H´usek, "Nonlinear Factorization in Sparsely Encoded Hopfield-like Neural Networks," *ESANN European Symposium on Artifical Neural Networks*, Bruges, Belgium, 1999.

Sandeep Kumar is a Software developer with Tata Consultancy services limited, working with Microsoft client for Online services development. He has 14 research publications and has attended various national/international conferences. The main area of interest included Artificial Intelligence, Neural Network and Application of computational techniques and programming in solving / analyzing biological problems / phenomenon.

Deepak Kumar is a Software developer with Tata Consultancy services limited. He has several research publications and has attended various national/international conferences. The main area of interest included Handheld development and open source systems.

Rashid Ali is a reader in department of Computer Engineering, Aligarh Muslim University.