

Reducing the Complexity of the Multilayer Perceptron Network using the Loading Matrix

Mohamed Loay Dahhan
PhD Student
Dept. Statistics and Information Systems
Faculty of Economics
University of Aleppo, Syria

Yasser Almoussa
Professor
Dept. Statistics and Information Systems
Faculty of Economics
University of Aleppo, Syria

ABSTRACT

In this paper, Researchers present three models of a Multilayer Perceptron Network (MLPs) based on the Factor Analysis with Principal Components method (PC) to reduce the degree of complexity of the neural network. In the first model, a neural network was built with all the variables in the input layer. In the second model, the results of the FA were adopted instead of the basic variables in the input layer, and in the third model, the Loading matrix was used to determine the number of nodes in the hidden layer and the weights that are associated with the input layer. Then compared the results of the models by determining the number of network weights that reflect the complexity of the network, in addition to the time of building and training the model and the accuracy of classification. The results of applying the models to a hypothetical database for the purposes of scientific research titled Bank Marketing showed that the model that inserted the factors in the hidden layer and preserved the high loading factors only is the best model in terms of low degree of complexity and maintaining classification accuracy.

General Terms

Complexity, Artificial Neural Networks ANN, Multivariate Analysis.

Keywords

Multilayer Perceptron Network MLP, Factor Analysis FA, Principle Component Analysis PCA, Complexity, Loading Matrix

1. INTRODUCTION

Artificial neural network (ANN) is a kind of cross-subject, which combines with Brain Science, Neuroscience, Cognitive Science, Psychology, Computer Science, and Mathematics [1]. There are many successful real world applications using ANN such as industry, business, science and classification problems [2-3]. Example of applications that have implemented using ANN are bankruptcy prediction [4], handwriting recognition [5], Time Series Forecasting [6]. Neural network has been more than 70 years until now; during these years, hundreds of network algorithm models have been proposed [7], and back-propagation (BP) neural network is one of the most mature and most widespread algorithms [8]. ANN with different structure gives different output for the same data set. Therefore, architectural design of ANN is very crucial and may be defined as an optimization problem [9]. MLP structure has three types of layers, the number of processing units (neurons) at input and output layer are fixed while these vary at the hidden layers. There is no straightforward criterion to calculate number of neurons in hidden layers nor is there any supporting theory for calculating the hidden layers. These architectural attributes are

involved in the performance of ANN because a network with few layers and few neurons can cause under fitting while a large network easily leads to overfitting. [10]

Plenty of algorithms have been proposed for optimizing training rule as well as ANN architecture, which can suggest an optimal, or nearest optimal ANN structure but problem still needs further exploration. Many papers have used Multivariate Analysis (Principle Component Analysis (PCA), Factor Analysis (FA), Cluster Analysis (CA) ...) to improve ANN Architectures. Gross and Luttermann first proposed a hybrid architecture combining PCA with a multilayer perceptron for recognizing human faces [11]; after that many papers applied the Principle Component in Neural Network in many sectors, to predict software faults during testing [12], prediction and data analysis applications in meteorology and oceanography [13]. Another paper used an ANN for classification and a PCA in the preprocessing stage to classify the quality of food products by a feedforward ANN with one hidden layer and the backpropagation algorithm. The PCA as a feature extractor was applied to spectral data before training the ANN to reduce the amount of redundant information [14]. Bucinski et al. combined PCA and ANN in medicine using backpropagation ANN to classify patients into two categories, and PCA to extract some important features predictive for patients' survival [15]. Song et al. analyzed the high-spectral remote sensing data. By the integration of factor analysis (FA) and NN [16]. Zhang used integrating principal component analysis (PCA) with neural network in clinical diagnosis, diagnosed the swelling through measuring the amount of nucleoside in urine [17]. Gopi Combined independent component analysis (ICA) and NN to extract the features of forged images, to identify the forged images through NN [18]. Sousa et al used PCA analysis with varimax rotation to extract factors using ozone concentrations data. The results showed that principal components as inputs improved MLR and ANN models' prediction by eliminating data collinearity and the number of predictor variables [19]. Ravi and Pramodh placed principal components instead of the hidden layer of a multilayer perceptron (MLP) network. They proposed a new principal component neural network (PCNN) architecture to solve bankruptcy prediction problem in commercial banks. They replaced the hidden layer of an ANN by a 'principal component layer', which consists of a few selected principal components that perform the function of hidden nodes. The advantage of such architecture is that it reduces the number of weights by eliminating connections between the input layer and the principal component layer [20]; Miguel used tied hidden Factors to model speaker and session variability and a MAP adaptation of some of the parameters of the model [21]. Ding proposed a back-propagation (BP) neural network algorithm based on factor analysis (FA) and cluster analysis (CA), which is combined with the principles of FA and CA,

and the architecture of BP neural network. The new algorithm reduces the feature dimensionality of the initial data through FA to simplify the network architecture [22]; Zekić Combined Two nonparametric methods to model entrepreneurial intentions: principal component analysis (PCA) and artificial neural networks (ANNs), it reveals benefits from the combination of the PCA and ANNs in modeling entrepreneurial intentions [23]

The main goal of this paper is to reduce the complexity degree of MLPs through reduction the number of connections (weights) between layers depending on the loading matrix that derived from factor analysis FA based on principal components analysis method. By building three models with different nodes and connections between layers, each model works only for single hidden layer. Then will compare the models based on confusion matrix and the needed time to build & train the model. This paper is organized as follows. In Section 2 presents neural networks and loading matrix. Section 3 discusses Proposed models. Section 4 presents an experiment to show the accuracy of models. Conclusions are presented in Section 5.

2. NEURAL NETWORKS & LOADING MATRIX

2.1 Neural Network Architecture

An artificial neural network, or simply neural network, is a type of artificial intelligence that attempts to mimic the way the human brain processes and stores information. It works by creating connections between mathematical processing elements, called neurons. Knowledge is encoded into the network through the strength of the connections between different neurons, called weights, and by creating groups, or layers, of neurons that work in parallel [24]. The system learns through a process of determining the number of neurons or nodes and adjusting the weights for the connections based upon training data. The architecture consists of not only input and output layer but also one or more intermediary layers called hidden layers. The computation units of the hidden layer called as hidden neurons. The function of hidden neurons is to intervene between the external input and the network output in some useful manner [25]. The performance of it fully depends upon the selection of architecture and training algorithm. MLP architecture can be tweaked using several parameters, such as the number of hidden layers, number of hidden neurons at each hidden layer and number of connections between layers [9].

The most widely used architecture in ANNs has been the Multiple Layer Perceptron (MLP), trained with the Back Propagation (BP) error-learning algorithm [26]. In this architecture, nodes are partitioned into layers numbered 0 to L, where the layer number indicates the distance of a node from the input nodes. The lowermost layer is the input layer numbered as layer 0, and the topmost layer is the output layer numbered as layer L. layers numbered between 0 and L are hidden layers [27-28-29], The ω number of weights and biases of the NN can be seen in Eq. (1) [30].

$$\omega = (N_i + 1)N_h + (N_h + 1)N_h * N_l + (N_h + 1)N_o \quad (1)$$

Where N_i , N_h , and N_o are the numbers of the nodes in the input layer, the hidden layer, and the output layer. N_l is the number of hidden layers as presented in Fig. 1.

In classification problems, the most common activation function is Softmax. Because the output vector has a

component corresponding to each target class, and the softmax output is interpreted as the probability of the input belonging to the corresponding class.

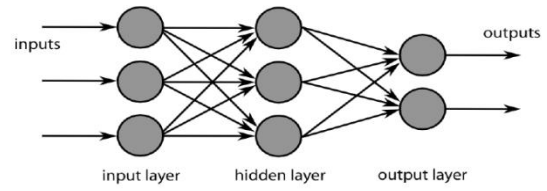


Figure 1: Neural Network Architecture

The softmax function differs from sigmoid functions in that it does not operate element-wise on a vector. Rather, the softmax function applies to an entire vector. If $x = [x_1, x_2, \dots, x_n]$, then its softmax $\mu(x) = [\mu(x_1), \mu(x_2), \dots, \mu(x_n)]$, where [31]:

$$\mu(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (2)$$

The backpropagation algorithm is a generalization of the least mean squared algorithm that modifies network weights to minimize the mean squared error between the desired and actual outputs of the network. Backpropagation uses supervised learning in which the network is trained using data for which inputs as well as desired outputs are known. Backpropagation is a simple training method that uses the calculated gradients of a neural network to adjust the weights of the neural network. As these weights are adjusted the neural network should produce more desirable output. The global error of the neural network should fall as it is trained [27-32].

ANN model trained by Cross-Validation Method, in k-fold cross-validation, the available learning set is partitioned into k disjoint subsets of approximately equal size. Fold refers to the number of resulting subsets. This partitioning is performed by randomly sampling cases from the learning set without replacement. The model is trained using k - 1 subsets, which, together, represent the training set. Then, the model is applied to the remaining subset, which is denoted as the validation set, and the performance is measured. This procedure is repeated until each of the k subsets has served as validation set. The average of the k performance measurements on the k validation sets is the cross-validated performance [33].

2.2 MLP Complexity degree

Complexity degree of Multi-Layer Perceptron could be estimated by the needed time to train and generalize the desired model. The essential storage requirement for the BP algorithm consists of all the N_w weights of the network. The computational complexity per iteration of the BP is around N_w multiplications for the forward pass, around $2N_w$ multiplications for the backward pass [34]. The process of presenting all the examples in the pattern set, with each example being presented once, is called an epoch. Neural networks are trained by presenting all the examples cyclically by epoch, until the convergence criteria is reached. The training examples should be presented to the network in a random order during each epoch [35].

Complexity degree of a Multi-Layer Perceptron is determined by the number of free parameters that it has (weights and biases), which in turn is determined by the number of neurons [36]. According to Eq(1) number of weights calculates by the sum of weights between each layer on sequence, as it

clarified:

$$\omega = \sum_{i=0}^{L-1} (\eta_i + b_i) * \eta_{(i+1)} \quad (3)$$

- ω : Total number of weights.
- η_i : Number of nodes in the layer i.
- b_i : Bias node in the layer i.
- L : Number of layers.

2.3 Factor Analysis and Loading Matrix

The essential purpose of factor analysis FA is to describe, if possible, the covariance relationships among many variables in terms of a few underlying, but unobservable, random quantities called Factors. Factor analysis can be considered an extension of principal component analysis PCA. Both can be viewed as attempts to approximate the covariance matrix. The approximation based on factor analysis model is more elaborate [37]. Despite similarities, there is a fundamental difference between them: PCA is a linear combination of variables; FA is a measurement model of a latent variable. The common estimation method of factors is principle factors, used the principal component method.

The observable random vector \mathbf{X} , with p components has mean μ and covariance matrix Σ . The factor model postulates that \mathbf{X} is linearly dependent upon a few unobservable random values F_1, F_2, \dots, F_m called Common factors, and p additional sources of variation $\epsilon_1, \epsilon_2, \dots, \epsilon_p$, called errors or sometimes called specific factors. The factor analysis model is:

$$\begin{aligned} X_1 - \bar{x}_1 &= \ell_{11} F_1 + \ell_{12} F_2 + \dots + \ell_{1m} F_m + \epsilon_1 \\ X_2 - \bar{x}_2 &= \ell_{21} F_1 + \ell_{22} F_2 + \dots + \ell_{2m} F_m + \epsilon_2 \\ &\vdots \\ X_p - \bar{x}_p &= \ell_{p1} F_1 + \ell_{p2} F_2 + \dots + \ell_{pm} F_m + \epsilon_p \end{aligned}$$

- \bar{x}_i : mean of variable i.
- ϵ_i : i th specific factor.
- F_j : j the common factor.
- ℓ_{ij} : Loading of the i_{th} on the j_{th} factor.

Or, in matrix notation,

$$X - \mu_{(p*1)} = L_{(p*m)} * F_{(m*1)} + \epsilon_{(p*1)} \quad (4)$$

Therefore, L is the matrix of factor loadings [37]. One widely used method to determine factors number is to let the number of factors equal the number of eigenvalues of the sample correlation matrix that are greater than 1. It is called Kaiser—Guttman criterion. Initial solution of factor loadings should be rotated to find a simple structure. Several methods for rotation to a simple structure have been proposed. The rotational methods are classified as orthogonal and oblique rotations, among the orthogonal rotations; Kaiser's vari-max rotation is by far the most frequently used. Each factor keeps variables whose coefficient of loading is greater than 0.5 [38].

3. PROPOSED MODELS

This paper supposes a neural network with three layers (input layer- hidden layer- output layer) to classify a binary variable with two classes, and:

- m : number of nodes in the input layer.
- h : number of nodes in the hidden layer.
- α : number of nodes in the output layer, its value is two.

In addition, there is a bias node in both input layer and hidden layer. According to Eq(2) the complexity degree (number of weights) can be calculated as follows:

$$\omega = (m + 1) * h + (h + 1) * 2 \quad (5)$$

- $(m + 1) * h$: represent the number of connections between input layer and hidden layer.
- $(h + 1) * 2$: represent the number of connections between hidden layer and output layer.

Then:

$$\omega = (m * h + h) + (2h + 2) \quad (6)$$

The number of nodes in hidden layer depends on trial and error concept. With too few nodes, the network may not be powerful enough for a given learning task. With a large number of nodes, computation is too expensive.

In this paper, according to Rule of thumb method researchers suppose the nodes in hidden layer is the average of the nodes number between input and output layers, so $h = \frac{m+2}{2}$ [39]; Then the number of weights in the model as following:

$$\begin{aligned} \omega &= \left((m + 1) * \left(\frac{m + 2}{2} \right) \right) + 2 \left(\frac{m + 2}{2} + 1 \right) \\ \omega &= \left(m * \frac{m + 2}{2} + \frac{m + 2}{2} \right) + \left(2 * \frac{m + 2}{2} + 2 \right) \\ \omega &= \frac{m^2}{2} + m + \frac{m}{2} + 1 + m + 2 + 2 \\ \omega &= \frac{m^2}{2} + \frac{5m}{2} + 5 \end{aligned} \quad (7)$$

In this paper, three different models of the neural network were studied, to evaluate the models researchers depended on the following measures:

1. Time to build and train the model.
2. Valid classification percentage.

3.1 First model (ANN)

Multi-layers neural network without any modification on the network architecture; the number of weights given as follows:

$$\omega_1 = \frac{m^2}{2} + \frac{5m}{2} + 5 \quad (8)$$

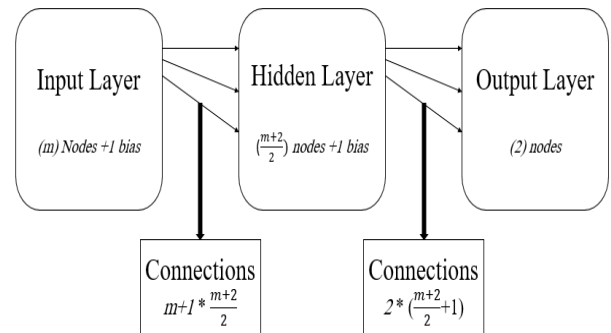


Figure 3. Architecture of Model 1

3.2 Second Model: (ANN with FA in Input Layer)

We applied factor analysis FA with principal component analysis PCA method; the factors p was merging into the neural network architecture by replacing the input layer that had m node with the factors p . and:

$$1 \leq p \ll m \quad (9)$$

Then the number of weights given as following:

$$\omega_2 = \frac{p^2}{2} + \frac{5p}{2} + 5 \quad (10)$$

The complexity degree (number of weights) of the second model are always lower than complexity degree (number of weights) of the first model because of expression (9). Then the second model needs a lower training time:

$$\omega_2 \ll \omega_1 \quad (11)$$

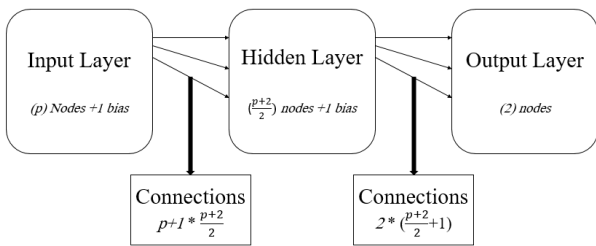


Figure 4. Architecture of Model 2

3.3 Third Model: (ANN with FA in Hidden Layer)

We also applied FA with PCA method. Then merge previous analysis into the neural network architecture by identifying the hidden layer nodes based on the factors p and the weights that linking it with the input layer nodes, which produced from the loading matrix $L_{ij} > 0.5$. The number of nodes in input layer could be lower than m by α , which represent the number of unloaded variables on any factor p . The number of connections between input layer nodes and hidden layer nodes is β , which represent the loaded variables on factors and bigger than 0.5, considering validity of $m^2 \gg \beta$. Then the complexity degree (number of weights) given as following expression:

$$\begin{aligned} \omega_3 &= \beta + (p + 1) * 2 \\ \omega_3 &= \beta + 2p + 2 \end{aligned} \quad (12)$$

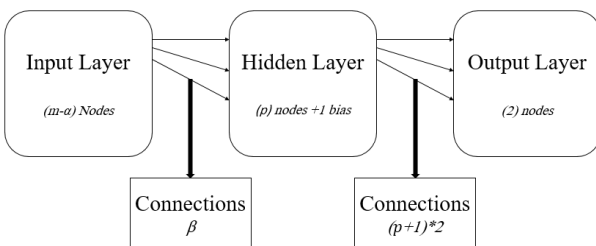


Figure 4. Architecture of Model 3

In addition, considering that:

$$\begin{aligned} p &< m \\ \beta &\ll m^2 \end{aligned}$$

Then:

$$\omega_3 < \omega_2 < \omega_1 \quad (13)$$

4. PRACTICAL EXPERIMENT

This study applied on a bank-marketing database [40] created to predict the success of marketing campaigns were based on phone calls. The number of studied cases was 41188, the missing data recoded with the "unknown" label for the categorical variables, series average for the numeric variables.

The attributes divided into 21 variables explained in the following table:

Table.1. Variables Dictionary

Numeric Variables	
Code	Description
X1	Age
X11	Last contact duration, in seconds.
X12	number of contacts performed during this campaign and for this client
X13	number of days that passed by after the client was last contacted from a previous campaign
X14	number of contacts performed before this campaign and for this client
X16	employment variation rate - quarterly indicator
X17	consumer price index - monthly indicator
X18	consumer confidence index - monthly indicator
X19	euribor 3 month rate - daily indicator
X20	number of employees - quarterly indicator
Nominal variables (0="No", 1="yes")	
X5	Has credit in default?
X6	Has housing loan?
X7	Has personal loan?
Y	Has the client subscribed a term deposit?
Categorical Variables (1 for the 1 st category...)	
X2	type of job
X3	marital status
X4	Education Level
X8	contact communication type
X9	last contact month of year
X10	last contact day of the week
X15	outcome of the previous marketing campaign

4.1 FA with PCA result:

FA was applied on the variables (X1, X2 ...X20), considering Y is the target variable. By using SPSS the result as following:

Multi-Collinearity: the value of determinant is (0.00007 \neq 0) which means factor analysis can be applied on the studied dataset, and then none of variables need to be deleted.

KMO Test: As in Table.2, the value of KMO is 0.543, which is bigger than 0.5. Then the sample size is good enough to apply the analysis.

Bartlett's Test: from the table 2, the significant level is 0.000. Therefore, researchers refuse the null hypothesis that the correlation matrix is an identity matrix.

Table.2 KMO & Bartlett's Test Results

KMO Value	0.543
Bartlett (sig)	0.000

Communalities: the default value of communalities is (1) based on PCA method and Correlation matrix. The value of communalities refer to the percentage of variable variances explained by the common factors. For example, the value of variable X15 is 0.964; refer to 96.4% of the variable X15 variances was explained by common factors.

In general most of common factors values explain a good percentage of variable variances, and the less percentage was for X4 (0.486). As cleared in Table.3.

Table 3 Communalities

Variable	default	Selected	Variable	default	Selected
X1	1	.688	X11	1	.560
X2	1	.536	X12	1	.550
X3	1	.569	X13	1	.920
X4	1	.486	X14	1	.940
X5	1	.290	X15	1	.964
X6	1	.645	X16	1	.948
X7	1	.654	X17	1	.800
X8	1	.741	X18	1	.504
X9	1	.808	X19	1	.976
X10	1	.957	X20	1	.888

Total Explained Variance: variables (X1, X2....X20) represented by nine factors. Based on Kaiser Rule, first nine factors was taken because of its Eigenvalues (EV>1).

Table 4 Total Explained Variance

	EV	Explained Variance %	Cumulative %
F1	4.254	21.27%	21.27%
F2	1.699	8.497%	29.766%
F3	1.556	7.778%	37.545%
F4	1.478	7.392%	44.936%
F5	1.286	6.432%	51.368%
F6	1.082	5.411%	56.779%
F7	1.042	5.21%	61.989%
F8	1.026	5.13%	67.119%
F9	1.001	5.003%	72.122%

As shown in table 4 the cumulative percent of explained variances is 72.122% of the total variance, and the first factor has the highest EV than the other factors 4.254 explain 21.27% of the total variances for the studied variables.

Loading matrix after rotation: It shows the correlation degree between variables and each factor of the selected factors using Varimax rotation. The variables with $\ell_{ij} < 0.5$ was ignored

Results: Nine factors was derived from variables summarizes in the table 6 below:

Table 4 Factors Result

Factor	Loaded variables	Variance explained
F1	X8,X16,X17,X19,X20	21.27%
F2	X13,X14	8.497%
F3	X1,X3	7.778%
F4	X8,X9	7.392%
F5	X13,X15	6.432%
F6	X6,X7	5.411%
F7	X2,X4	5.21%
F8	X11,X12	5.13%
F9	X10	5.003%

Then the representation of factor model formalized as next expressions:

$$F_1 = 0.511 X_8 + 0.958 X_{16} + 0.793 X_{17} + 0.964 X_{19} + 0.864 X_{20}$$

$$F_2 = 0.708 X_{13} - 0.887 X_{14}$$

$$F_3 = 0.825 X_1 - 0.748 X_3$$

$$F_4 = 0.859 X_9 - 0.664 X_8$$

$$F_5 = 0.967 X_{15} - 0.605 X_{13}$$

$$F_6 = 0.799 X_6 + 0.806 X_7$$

$$F_7 = 0.710 X_2 - 0.611 X_4$$

$$F_8 = 0.712 X_{12} - 0.726 X_{11}$$

$$F_9 = 0.977 X_{10}$$

Total explained variances was 72.122% .Then we can depend on the proposed factors because total explained variances greater than 60%.

In addition, two variables were ignored (X5, X18) because there are not loaded on any factor.

4.2 Applying Models:

Multi-layer Perceptron architecture has a good fitting with proposed classification issue, the researchers determined the static attributes for each of the proposed models summarizes as following:

Table 7 Neural Network static attributes

Total Number of Instances	41188
Training and Testing Method	Cross-Validation
Variables Scaling	Standard

Optimization Algorithm	Softmax
Seeds	0
Learning Rate	$\eta = 0.4$
Momentum	$\alpha = 0.3$
Validation Threshold	20

4.2.1 Model Architecture:

First Model (ANN): The 20 variables represented in input layer and the target variable as in output layer, so the number of output layer nodes is two based on the variable form. Number of nodes in the hidden layer is $\frac{20+2}{2} = 11$ node. Moreover, the number of weights is the connections between each node in input layer with the nodes in hidden layer, and each node in hidden layer connected with the nodes in output layer. Then number of connections or weights calculated as follows:

$$\omega_1 = \frac{20^2}{2} + \frac{5 * 20}{2} + 5 = 255$$

Second model (ANN with FA in input layer): Factors represented in input layer (variables are replaced by the derived factors as inputs in input layer). Therefore, the nodes number in input layer is 9 nodes, and the number of nodes in hidden layer is $\frac{9+2}{2} \cong 6$. The nodes number in output layer is 2. Then number of the connections or weights calculated as follows:

$$\omega_2 = \frac{9^2}{2} + \frac{5 * 9}{2} + 5 = 68$$

Third model (ANN with FA in hidden layer): Input layer represented with total number of variables except (X5, X18), so number of nodes in input layer is 18. And for the number of nodes in hidden layer determined based on factor analysis results by number of derived factors values that its EV greater than 1 which is $p=9$ nodes. Like the previous models the nodes of output layer number is 2. Then for the weights number, it is calculated based on number of loaded variables on factors $\beta = 20$ added to the number of connections between hidden and output layer $(p+1)*2$ as following:

$$\omega_3 = 20 + (2 * 9 + 2) = 40$$

Table (8) clear the calculated components of neural networks architecture based on the discussed models above:

Table 8 Components of neural network architecture

Model	ANN	ANN with FA in input layer	ANN with FA in hidden layer
Input Layer	20 Node	9 Node	18 Node
Hidden Layer	11 Node	6 Node	9 Node
Output Layer	2 Node	2 Node	2 Node
Total Weights	255	68	40

4.2.2 Confusion Matrix

It used to describe the performance of a classification model on a set of test data for which the true values are known. The following tables clear the confusion matrix for each model.

ANN model: Model classified accuracy is 90.89%. Number of true classified instances is 37436

Table 9 Confusion Matrix for ANN model

Actual	Classifier	
	a=F	b=T
a=F	35120	1428
b=T	2324	2316
True classified percentage	$35120 + 2316 / 41188 = 90.89\%$	

ANN with FA in input layer model: Model classified accuracy is 90.77%. Number of true classified instances is 37389.

Table 10 Confusion Matrix for ANN with FA in Input Layer model

Actual	Classifier	
	a=F	b=T
a=F	35324	1224
b=T	2575	2065
True classified percentage	$35324 + 2065 / 41188 = 90.77\%$	

ANN with FA in hidden layer model: Model classified accuracy is 90.85%. Number of true classified instances is 37421.

Table 11 Confusion Matrix for ANN with FA in Hidden Layer model

Actual	Classifier	
	a=F	b=T
a=F	35413	1135
b=T	2632	2008
True classified percentage	$35413 + 2008 / 41188 = 90.85\%$	

4.3 Discussion

Table (12) reviews results comparison between three models. Classification Rate at the ANN Model was the greatest 90.89%, then ANN with FA in hidden layer model and the lowest for the ANN with FA in input layer model. In addition, it is interesting to note that classification rate for the ANN with FA in hidden layer model was so close to the highest one with a low difference equal to 0.04%. The highest value for kappa statistic was for the ANN model 0.5024. Whereas the lowest value for the ANN with FA in hidden layer model 0.4676, with small different between them.

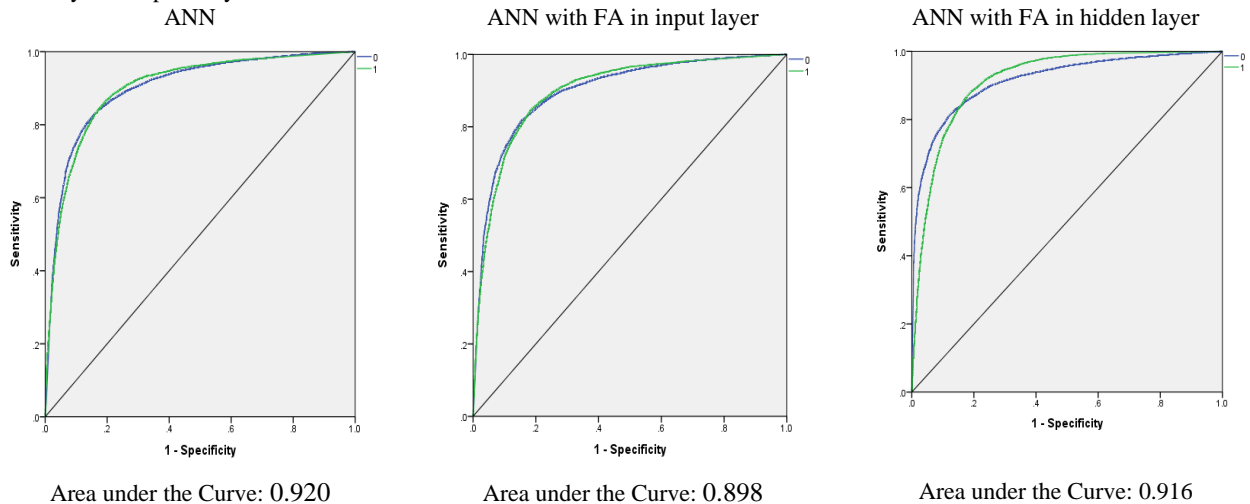
Time needed to build and train the first model was 1239 seconds, which is extremely high, and then the second model with 486 seconds for building and training and the lowest time was for the third model around 319 seconds. Finally notice that model sensitivity for the classifier (b=T) could return to the origin of the proposed issue in dataset, we could predict to deposit subscription (a=F) because of quantity and diversity of

the instances.

Table 12 Models implementation results summary

Model	ANN	ANN with FA in input layer	ANN with FA in hidden layer
True classified instances	37436	37389	37421
False classified instances	3752	3799	3767
Kappa statistic	0.5024	0.4715	0.4676
True classifier percentage	% 90.890	% 90.776	% 90.854
Model building & training time	1239 sec	486 sec	319 sec
Complexity degree (Weights number)	255	68	40

Fig (1) shows the results of the comparison between models, and clearly the superiority of the **ANN with FA in hidden**



layer model in complexity degree building & training time, and its proximity to the best correct classification

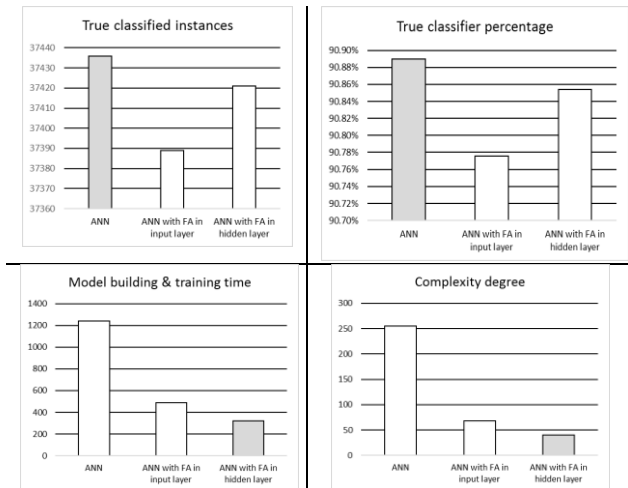


Fig 1: Comparison Charts between Models

Fig (2) shows ROC curve for each model, it is clear that the largest area under the curve for the ANN model, then ANN with FA in hidden layer model, and the lowest for the ANN with FA in input layer model. This is reflected in the accuracy of the True classifier percentage of these models.

5. CONCLUSIONS

- Using multi-layer perceptron made a good performance with binary variable classification based on multi-dimension database.
- When depending on factors in the input layer, some of variable had no connection with any node in the hidden layer, because this variable unloaded on any factor, whereas all variable was connected with hidden layer nodes without using factor analysis.
- Replacing the original variables with factors in ANN with FA in input layer model has a good impact on reduction the complexity degree for the network, and a good classification accuracy comparing to the original model with less building and training time.
- Reducing the complexity degree of ANN by building an ANN with FA in hidden layer model,

the building and training time has been reduced by 75% of the original model with keeping on the classification accuracy.

- Applied FA with PCA method on multi-dimensional database with many different data types before ANN model Building and merging the results of FA in the hidden layer will contribute in reducing the ANN complexity degree then improving in building time and classification accuracy.

6. REFERENCES

- McCulloch W, Pitts W, 1943, "A logical calculus of the ideas immanent in nervous activity". Bull Math Biophys 10(5):115–133
- P. Jeatrakul, K.W. Wong, 2009. "Comparing the Performance of Different Neural Networks for Binary Classification Problems". IEEE

- [3] B. Widrow, D. Rumelhard, and M. A. Lehr, 1994. "Neural networks: Applications in industry, business and science," *Communications of the ACM*, vol.37, pp. 93-105.
- [4] C. Charalambous, A. Charitou, and F. Kaourou, 2000, "Comparative analysis of artificial neural network models: application in bankruptcy prediction" in *Neural Networks, IJCNN'99. International Joint Conference on Neural Networks*, vol.6, pp. 3888-3893.
- [5] L. Gang, B. Verma, and S. Kulkarni, 2002. "Experimental analysis of neural network based feature extractors for cursive handwriting recognition," in *Neural Networks. IJCNN '02. Proceedings of the 2002 International Joint Conference on Neural Networks*, pp. 2837-2842.
- [6] R. Rabuñal Juan, Dorado Julián, 2006. "Artificial Neural Networks in real life applications", Idea group publishing, London.
- [7] Cheng H, Cai X, Min R, 2009. "A novel approach to color normalization using neural network". *Neural Computing and Applications* 18(3), pp. 237–247.
- [8] Rumelhart D, Hinton G, Williams R, 1986, "Learning representation by back-propagating errors". *Nature.com* vol. 323, no. 6088, pp. 533–538,
- [9] Gupta Tarun Kumar, Raza Khalid, 2018. "Optimizing Deep Neural Network Architecture: A Tabu Search Based Approach". *Computer Science*, springer.
- [10] Agah Arvin, 2013. "Medical Applications of Artificial Intelligence", CRC Press, Tylor & Fracis Group. New York p. 204.
- [11] M. Gross, H. Luttermann, "Combining principal component analysis and neural networks for the recognition of human faces- a case study for man - machine - communication, in: S.Y. Shin, T.L. Knuii (Eds.),1993. *Computer Graphics & Applications*, Proceedings of the Pacific Graphics, vol. 1, pp. 176–192.
- [12] T.M. Khoshgoftaar, R.M. Szabo, 1996. "Using neural networks to predict software faults during testing", *IEEE Transactions on Reliability*, pp: 456–462.
- [13] W.W. Hseih, B. Tang, 1998. "Applying neural network models to prediction and data analysis in meteorology and oceanography", *Bulletin of the American Meteorological. Soc.* 79 pp: 1855–1870.
- [14] O'Farrella, et al., 2005. "Combining principal component analysis with an artificial neural network to perform online quality assessment of food as it cooks in a large-scale industrial oven", *Sensors and Actuators B*, Vol. 107, pp. 104–112.
- [15] Bucinski, A., Baczek, T., Wasniewski, T., and Stefanowicz, M., 2005. "Clinical data analysis with the use of artificial neural networks (ANN) and principal component analysis (PCA) of patients with endometrial carcinoma", *Reports on Practical Oncology and Radiotherapy*, Vol. 10, pp. 239-248.
- [16] Song J, Feng Y, 2006. "Hyperspectral data classification by independent component analysis and neural network". *Remote sensing technology and application*, pp. 115–119
- [17] Zhang Y, 2007. "Artificial neural networks based on principal component analysis input selection for clinical pattern recognition analysis". *Talanta* 73(1) pp:68–75.
- [18] Gopi E, 2007. "Digital image forgery detection using artificial neural network and independent component analysis". *Appl Math Comput* 194(2) pp:540–543.
- [19] Sousa, S.I.V. , Martins, F.G. , Alvim-Ferraz, M.C.M., Pereira, M.C., 2007. "Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations", *Environmental Modelling & Software*, Vol. 22, pp. 97-103.
- [20] Ravi, V., Pramodh, C., 2008. "Threshold accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks", *Applied Soft Computing*, Vol. 8, pp. 1539–1548.
- [21] Miguel Antonio, Llombart Jorge, Ortega Alfonso and Lleida Eduardo, 2017. "Tied Hidden Factors in Neural Networks for End-to-End Speaker Recognition", In *Proc. Interspeech*, pp 2819–2823, Stockholm, Sweden.
- [22] Ding Shifei et al. et al., 2010. "Research of neural network algorithm based on factor analysis and cluster analysis", *Neural Comput & Applic* (2011) 20:297–302 springer-Verlag London Limited.
- [23] Zekić-Sušac Marijana et al., 2013. "Combining PCA analysis and artificial neural networks in modelling entrepreneurial intentions of students", *Croatian Operational Research Review*, Croatia, 4, P306-317.
- [24] Taylor, Brian J., 2006. "Methods and Procedures for the Verification and Validation of Artificial Neural Networks", Springer Science & Business Media.
- [25] P. Arumugam, R. Ezhilarasi, 2017. "Data Mining based Neural Network Model for Rainfall Forecasting", *International Journal of Computer Applications*, Volume 170 – No.4.
- [26] Shafi Imran et al., 2006. "Impact of Varying Neurons and Hidden Layers in Neural Network Architecture for a Time Frequency Application", *IEEE International Multitopic Conference*.
- [27] Mehrotra K., Mohan C., Ranka S., 1997. "Element of Artificial Neural Networks", MIT Press, USA.
- [28] S. Haykin, 2008. "Neural networks and learning machines", third edition, Prentice Hall.
- [29] R. Rojas, 1996. "Neural networks - a systematic introduction", Springer-Verlag.
- [30] Dang Tuan Linh, Hoshino Yukinobu, 2019. "Improved PSO Algorithm for Training of Neural Network in Co-design Architecture", *International Journal of Computer Applications*, Volume 182 - No.44.
- [31] Leskovec Jure, Rajaraman Anand, Ullman Jeffrey David, 2020. "Mining of Massive Datasets", Third edition, Cambridge University Press, pp. 517-520.
- [32] Heaton Jeff, 2011. "Introduction to the Math of Neural Networks", Heaton Research, Inc., USA.
- [33] Berrar Daniel, 2018. "Cross-validation", *Encyclopedia of Bioinformatics and Computational Biology*, Volume 1, Elsevier, pp. 542–545.
- [34] Kamarthi, S. V., & Pittner, S. 1999. "Accelerating neural network training using weight extrapolations". *Neural Networks*, 12, 1285–1299.

- [35] Du Ke-Lin, Swamy M. N. S., 2019. "Neural Networks and Statistical Learning", Springer-Verlag London, UK.
- [36] Hagan Martin, Demuth Howard, Beale Mark, De Jesús Orlando, "Neural Network Design", Oklahoma State University, Stillwater, OK, United States, 2nd Edition, P13-1.
- [37] Johnson Richard, Wichern Dean, 1998. "Applied Multivariate Statistical Analysis", fourth edition, prentice hall, Upper Saddle River, New jersey 07458.
- [38] J,Salkind Neil, 2010. "Encyclopedia of Research Design", 1st Edition, Kindle Edition, SAGE Publications, pp. 460-466.
- [39] Saurabh Karsoliya, 2012. "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture" International Journal of Engineering Trends and Technology- Volume 3 Issue 6.
- [40] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, In press, <http://dx.doi.org/10.1016/j.dss.2014.03.01>