Comparison of NNARX, ANN and ARIMA Techniques to Poultry Retail Price Forecasting

Alireza Karbasi

University of Zabol, College of Agriculture,
Agricultural Economics Department,
P.O.Box. 98615-538
Zabol
Iran

Somayeh shirzadi Laskukalayeh

University of Zabol, College of Agriculture,
M.Sc student of Agricultural Economics Department,
P.O.Box. 98615-538
Zabol
Iran

Seiad Mohammad Fahimifard

University of Zabol, College of Agriculture,
M.Sc student of Agricultural Economics Department,
P.O.Box. 98615-538
Zabol
Iran

Contributed Paper prepared for presentation at the International Association of Agricultural Economists Conference, Beijing, China, August 16-22, 2009

Abstract

The lack of study among the economic forecasting literature that can empirically proves the hypothesis of being more powerfulness of dynamic neural networks in comparison with the static neural networks models for forecasting, is the most important motivation of this study. In this paper, the utilization of NNARX as a nonlinear dynamic neural network model, ANN as a nonlinear static neural network model and ARIMA as a linear model were compared to forecast poultry retail price. As a case study on Iranian poultry retail price, we compare forecast performance of these models for three forecasts (1, 2 and 4 week ahead). Results show that NNARX and ANN models outperform ARIMA model, and also NNARX model outperforms ANN model for all three forecasts.

Keywords: NNARX; Poultry Retail Price; Forecasting.

1. Introduction

In the last few decades, many forecasting models have been developed [19.]. Which among them, the autoregressive integrated moving average (ARIMA) model has been highly popularized, widely used and successfully applied not only in economic time series forecasting, but also as a promising tool for modeling the empirical dependencies between successive times and failures [13.]. Recently, it is well documented that many economic time series observations are non-linear while, a linear correlation structure is assumed among the time series values therefore, the ARIMA model can not capture nonlinear patterns and, approximation of linear models to complex real-world problem is not always satisfactory. While nonparametric nonlinear models estimated by various methods such as Artificial Intelligence (AI), can fit a data base much better than linear models and it has been observed that linear models, often forecast poorly which limits their appeal in applied setting [22.].

Artificial Intelligence (AI) systems are widely accepted as a technology offering an alternative way to tackle complex and ill-defined problems [15.]. They can learn from examples, are fault tolerant in the sense that they are able to handle noisy and incomplete data, are able to deal with non-linear problems, and once trained can perform prediction and generalization at high speed [16.]. They have been used in diverse applications in control, robotics, pattern recognition, forecasting, medicine, power systems, manufacturing, optimization, signal processing and social/psychological sciences. AI systems comprise areas like expert systems, ANNs, genetic algorithms, fuzzy logic and various hybrid systems, which combine two or more techniques [16.]. Among the mentioned AI systems, according to Haykin [12.], a neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. Also, the greatest advantage of a neural network is its ability to model complex nonlinear relationship without a priori assumptions of the nature of the relationship like a black box [17.].

On the order hand, neural networks can be classified into dynamic and static categories. Static (feed-forward) networks have no feedback elements and contain no delays; the output is calculated directly from the input through feed-forward connections. In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network. Dynamic networks are generally more powerful than static networks (although somewhat more difficult to train). Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns [22.].

Concerning the application of neural nets to time series forecasting, there have been mixed reviews. For instance, Lapedes and Farber [18.] reported that simple neural networks can outperform conventional methods, sometimes by orders of magnitude. Sharda and patil [28.] conducted a forecasting competition between neural network models and traditional forecasting technique (namely the Box-Jenkins method) using 75 time series of various natures. They concluded that simple neural nets could forecast about as well as the Box-Jenkins forecasting system. Wu [29.] conducts a comparative study between neural networks and ARIMA models in forecasting the Taiwan/US dollar exchange rate. His findings show that neural networks produce significantly better results than the best ARIMA models in both one-step-ahead and six-step-ahead forecasting. Similarly, Hann and Steurer [10.], Zhang and Hu [30.] find results in favor of neural network. Gencay [8.] compares the performance of neural network with those of random walk and GARCH[1] models in forecasting daily spot exchange rates for the British pound, Deutsche mark, French franc, Japanese yen, and the Swiss franc. He finds that forecasts generated by neural network are superior to those of random walk and GARCH models.

We couldn't big body of study among the economic forecasting literatures that can empirically proves the hypothesis of being more powerfulness of dynamic neural networks in compare with the static neural networks for economic time series forecasting. This lack motivated us to prepare this study. In this paper we compare the utilization of NNARX[2] as a nonlinear dynamic neural network model, ANN[3] as a nonlinear static neural network model and ARIMA as a linear model for forecasting. In order to comparison of mentioned models we use the common forecast performance measures such as Absolute fraction of variance (R^2), Mean Absolute Deviation (MAD), Mean Square Error (MSE) and Root Mean Square Error (RMSE). As an empirical application, we compare the various forecasting performance of mentioned models for three perspectives (1, 2 and 4 week ahead) of Iran poultry retail price weekly time series via common forecast performance measures. We obtained the weekly poultry retail price time series of Iran for the period 2002:3-2007:12 from the website of Iran State Livestock Affairs Logistics. Also, we consider the period 2002:3-2006:3 (70% of total observations) and 2006:3-2007-12 (30% of total observations) for training and testing of all models, respectively.

2. Auto-Regressive Integrated Moving Average (ARIMA) Model

Introduced by Box and Jenkins [1.], in the last few decades the ARIMA model has been one of the most popular approaches of linear time series forecasting methods. An ARIMA process is a mathematical model used for forecasting. One of the attractive features of the Box-Jenkins approach to forecasting is that ARIMA processes are a very rich class of possible models and it is usually possible to find a process which provides an adequate description to the data. The original Box-Jenkins modeling procedure involved an iterative three-stage process of model selection, parameter estimation and model checking. Recent explanations of the process (e.g., [20.]) often add a preliminary stage of data preparation and a final stage of model application (or forecasting).
Also, the *ARIMA (p, d, q)* model for variable *y* is as follow:

$$y_t = f(t) + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + e_t + \theta_1 e_{t-1} + ... + \theta e_{t-q} \quad (1)$$

Where *y* is estimated by the following equation:

$$y_t = \Delta^d x_t = (1-L)^d x_t \quad (2)$$

Where y_t and e_t are the target value and random error at time t, respectively, $\phi_i(i = 1,2,..., p)$ and $\theta_j(j = 1,2,..., q)$ are model parameters, p and q are integers and often referred to as orders of autoregressive and moving average polynomials.

3. Artificial Neural Network model

---

1 . Generalized Auto-Regressive Conditional Hetroskedastic.
2 . Neural Network Auto-Regressive model with eXogenous inputs.
3 . Artificial Neural Network.

The major advantage of neural networks is their flexible capability of nonlinear modeling. With ANN, there is no need to specify a particular model. Rather, the model is adaptively based on the features presented from the data [11.]. This data-driven approach is suitable for many empirical researches where no theoretical guidance is available to suggest an appropriate data generating process. The most common types of ANN models have been shown in figure1:

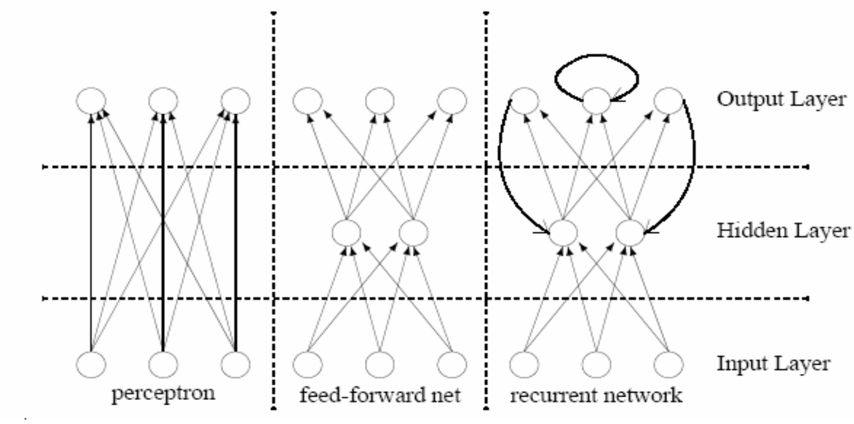


Fig1. Most common types of ANN models

For the purposes of this paper, the feed-forward backpropagation neural network (also known as a MLP[4] network) is the neural network model most widely used in time series forecasting, because it is capable of resolving a wide variety of problems [27.]. MLP network is made up of an input layer, an output layer and one or more hidden layers of neurons. As the fig2 shows, each input is weighted with an appropriate *w*. The sum of the weighted inputs and the bias forms the input to the transfer function *f*.

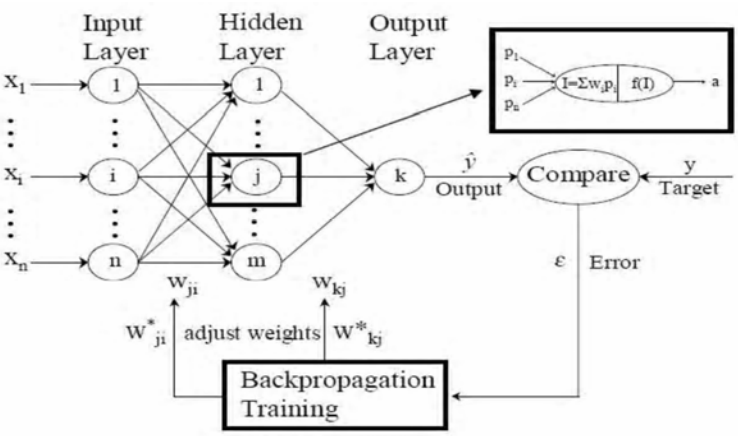


Fig.2. A typical Back-Propagation neural network

Neurons can use any differentiable transfer function *f* to generate their output. In general, transfer function introduces a degree of nonlinearity that is valuable for most ANN applications and ideally, it should be continuous, differentiable, and monotonic. Feed-forward networks often have hidden layer(s) of sigmoid neurons followed by an output layer of linear neurons.

Two stages may be considered in the MLP network: the running stage, in which an input pattern is presented to the trained network and transmitted through successive layers of neurons until reaching an output, and the training or learning stage in which the weights or parameters of the network are iteratively modified on the basis of a set of input–output

[4] . Multilayer Perceptron.

patterns known as a training set, in order to minimize the deviance or error between the output obtained by the network and the user's desired output. This is why MLP network learning is said to be supervised. The learning rule commonly used in this type of network is the back propagation algorithm or gradient descent method, developed and disseminated by Rumelhart, Hinton and Williams [26.]. In this research, we use the following three-layer feed-back networks:

$$F = F\left[\beta_0 + \sum_{j=1}^{J} \beta_j G\left[\sum_{k=1}^{K} \gamma_{kj} X_J\right]\right] \tag{3}$$

Where F is the output function of the output layer unit, β_0 is the bias unit (equal to 1), G is the output function of the hidden layer units j, γ_{kj} denotes the weight for the connection linking input k to the hidden unit j, β_j is the weight of outputs from the hidden layers in the output layer unit, and X is the input vector.

4. Neural Network Auto-Regressive model with eXogenous inputs (NNARX):

Neural networks can be classified into dynamic (e.g. NNARX) and static (e.g. ANN) categories. Static networks have no feedback elements and contain no delays; the output is calculated directly from the input through feed-forward connections. In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network. Dynamic networks are generally more powerful than static networks (although somewhat more difficult to train). Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns [21.]. This model has a parametric component plus a nonlinear part, where the nonlinear part is approximated by a single hidden layer feed-forward ANN. The neural network autoregressive with exogenous inputs (NNARX) is current dynamic network, with feedback connections enclosing several layers of the network. The NNARX model is based on the linear ARX model, which is commonly used in time-series modeling. Also, this has applications in such disparate areas as prediction in financial markets [24.], channel equalization in communication systems [7.], phase detection in power systems [16.], sorting ([14.], fault detection [3.], speech recognition [23.], and even the prediction of protein structure in genetics [9.].

The defining equation for the NNARX model is as follow:

$$y(t) = f(y(t-1), y(t-2), ... y(t-n_y), u(t-1), u(t-2), ..., u(t-n_u)) \tag{4}$$

Where, the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal. The output is feed back to the input of the feed-forward neural network as part of the standard NNARX architecture, as shown in the left fig3. Because the true output is available during the training of the network, we could create a series-parallel architecture [25.], in which the true output is used instead of feeding back the estimated output, as shown in the right fig3. This has two advantages. The first is that the input to the feed-forward network is more accurate. The second is that the resulting network has purely feed-forward architecture, and static backpropagation can be used for training.

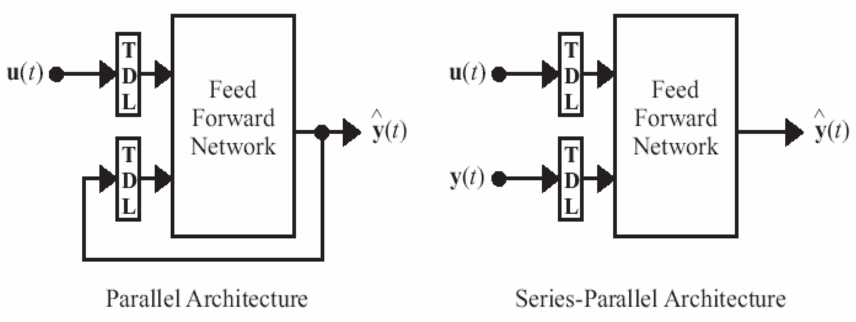


Fig3. Parallel and Series-Parallel Architectures

Dynamic networks are trained in the same gradient-based algorithms that were used in "Backpropagation." Although they can be trained using the same gradient-based algorithms that are used for static networks, the performance of the algorithms on dynamic networks can be quite different, and the gradient must be computed in a more complex way [5.]. A diagram of the resulting network is shown by fig 4, where a two-layer feed-forward network is used for the approximation.

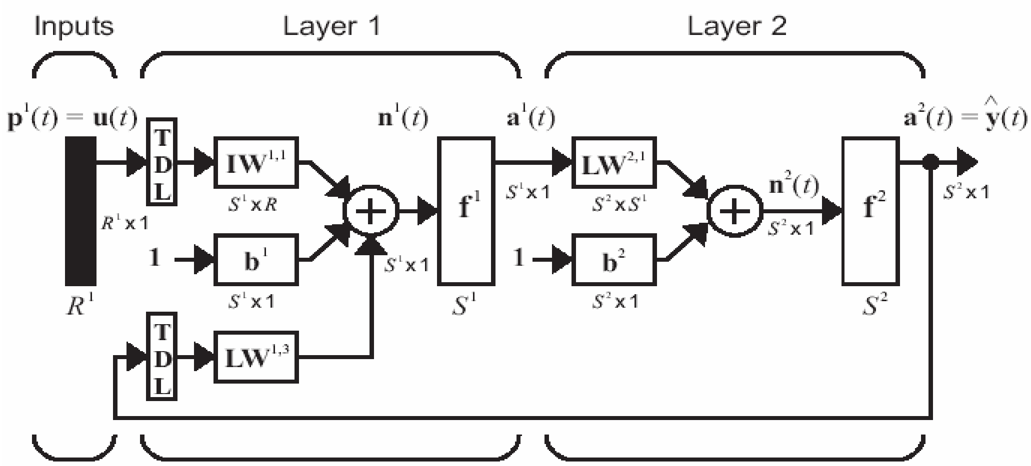


Fig4. A typical neural network auto-regressive with exogenous inputs (NNARX)

This type of network's weights has two different effects on the network output. The first is the direct effect, because a change in the weight causes an immediate change in the output at the current time step (This first effect can be computed using standard backpropagation.). The second is an indirect effect, because some of the inputs to the layer, such as $a(t,1)$, are also functions of the weights. To account for this indirect effect, we must use dynamic backpropagation to compute the gradients, which are more computationally intensive [6.]. Expect dynamic backpropagation to take more time to train, in part for this reason. In addition, the error surfaces for dynamic networks can be more complex than those for static networks. Training is more likely to be trapped in local minima. This suggests that you might need to train the network several times to achieve an optimal result [4.].

5. Forecast Performance Measures

Forecast researchers need measures in order to compare the forecasting performance of various models. Commonly, these measures are including of R^2, MAD, MSE and RMSE that the following is their definition and general formulas:

Table 2.Four common types of forecast performance measures

Measure Definition	Formulate
Absolute fraction of variance (R^2)	$R^2 = 1 - \frac{\sum(\hat{y}_t - y_t)^2}{\sum \hat{y}_t^2}$
Mean Absolute Deviation (*MAD*)	$MAD = \frac{\sum \lvert \hat{y}_t - y_t \rvert}{n}$
Mean Square Error (*MSE*)	$MSE = \frac{\sum(\hat{y}_t - y_t)^2}{n}$
Root Mean Square Error (*RMSE*)	$RMSE = \sqrt{\frac{\sum(\hat{y}_t - y_t)^2}{n}}$

Where y_t, $\hat{y}_t$ and n are the target value, output value and number of observations, respectively. Clearly, the best score for R^2 measure is 1 and for other measures is zero.

6. Data construction

For the exercise which is follows, we modeled the Iran poultry retail price as a function of past prices. Clearly, this has the shortcoming that our models are somewhat naive from the perspective of theoretical macroeconomics. However, there is a large body of literature in economics suggesting that very parsimonious models, such ARIMA model, perform better than more complex models, at least from the perspective of forecasting [2.]. We obtained the weekly poultry retail price time series of Iran for the period 2002:3-2007:12 from the website of Iran State Livestock Affairs Logistics (www.IranSLAL.com). Also, we consider the period 2002:3-2006:3 (70% of total observations) and 2006:3-2007-12 (30% of total observations) for training and testing of all models, respectively.

7. Results and discussion

In this section we presented the empirical results of comparing the forecasting performance of mentioned models for 1, 2 and 4 week ahead of Iran poultry retail price weekly time series via common forecast performance measures. For ARIMA model we identified the degree of integration (*d*), autoregessive (*p*) and moving average (*q*) by Dikey-Fuller, correlation and partial correlation diagrams, respectively. We applied the Schwartz-Bayesian criterion for identification of lag number. For ANN model and nonlinear part of NNARX we investigated the various architectures of feed-forward backpropagation network. Table3 shows the summary of these results:

Table 3.Comparision of NNARX, ANN and ARIMA models for forecasting

A. ANN /ARIMA

Week(s) ahead	Architecture	R^2		MAD		MSE		RMSE	
		Train	Test	Train	Test	Train	Test	Train	Test
1	5-2-1-1	1.04211	1.05383	0.16279	0.36039	0.04000	0.07143	0.14462	0.30163
	5-3-2-1-1	1.04127	1.04950	0.58326	0.70598	0.20000	0.30357	0.43696	0.58696
	5-4-3-2-1-1	1.04169	1.05393	0.38884	0.43555	0.12000	0.10714	0.31088	0.36232
	5-5-4-3-2-1-1	1.04211	1.05372	0.10884	0.25507	0.04000	0.03571	0.08962	0.21377
	5-6-5-4-3-2-1-1	1.04044	1.05340	0.73581	0.97477	0.40000	0.57143	0.60754	0.80888
2	5-2-1-1	1.04573	1.05586	0.91681	0.96206	0.46875	0.59016	0.08377	0.82920
	5-3-2-1-1	1.04866	1.05628	0.16303	0.33637	0.03125	0.08197	0.15656	0.29335
	5-4-3-2-1-1	1.04761	1.05766	0.59076	0.78806	0.18750	0.39344	0.45869	0.68014
	5-5-4-3-2-1-1	1.04845	1.05893	0.31008	0.48862	0.06250	0.14754	0.24783	0.42286
	5-6-5-4-3-2-1-1	1.04814	1.05681	0.41597	0.57663	0.12500	0.21311	0.34200	0.50326
4	5-2-1-1	1.05266	1.05459	0.36386	0.42514	0.07692	0.10606	0.29206	0.37170
	5-3-2-1-1	1.05276	1.05714	0.67631	0.54732	0.05128	0.22727	0.51805	0.55647
	5-4-3-2-1-1	1.05308	1.05799	0.14699	0.39059	0.02564	0.09091	0.14159	0.34184
	5-5-4-3-2-1-1	1.05024	1.05449	0.80723	0.92839	0.38462	0.53030	0.67629	0.80701
	5-6-5-4-3-2-1-1	1.05308	1.05874	0.16948	0.22784	0.02564	0.03030	0.14714	0.19991

B. NNARX/ARIMA

Week(s) ahead	Architecture	R^2		MAD		MSE		RMSE	
		Train	Test	Train	Test	Train	Test	Train	Test
1	5-2-1-1	1.04211	1.05478	0.12744	0.25453	0.00000	0.03571	0.10816	0.21377
	5-3-2-1-1	1.04211	1.05520	0.03907	0.07625	0.00000	0.01786	0.03337	0.06431
	5-4-3-2-1-1	1.04211	1.05435	0.20558	0.34723	0.00000	0.07143	0.17985	0.29121
	5-5-4-3-2-1-1	1.04211	1.05456	0.06047	0.08941	0.00000	0.01786	0.05748	0.07790
	5-6-5-4-3-2-1-1	1.04211	1.05446	0.06140	0.13714	0.00000	0.01786	0.05624	0.11685
2	5-2-1-1	1.04866	1.05893	0.09076	0.16439	0.00000	0.01639	0.07626	0.14429
	5-3-2-1-1	1.04866	1.05904	0.06975	0.11735	0.00000	0.01639	0.06181	0.10300
	5-4-3-2-1-1	1.04866	1.05851	0.11008	0.18665	0.00000	0.01639	0.09705	0.16341
	5-5-4-3-2-1-1	1.04866	1.05935	0.03866	0.03237	0.00000	0.01639	0.04275	0.03738
	5-6-5-4-3-2-1-1	1.04866	1.05808	0.09832	0.08093	0.00000	0.01639	0.09359	0.07171
4	5-2-1-1	1.05308	1.05927	0.10843	0.21783	0.00000	0.03030	0.09828	0.19126
	5-3-2-1-1	1.05308	1.06086	0.56627	0.54181	0.00000	0.18182	0.48029	0.47339
	5-4-3-2-1-1	1.05308	1.05927	0.11486	0.22784	0.00000	0.03030	0.10328	0.19991
	5-5-4-3-2-1-1	1.05308	1.05937	0.05863	0.04306	0.00000	0.01515	0.06607	0.05539
	5-6-5-4-3-2-1-1	1.05308	1.05927	0.04900	0.03455	0.00000	0.01515	0.05664	0.07443

C. NNARX/ANN

Week(s) ahead	Architecture	R^2		MAD		MSE		RMSE	
		Train	Test	Train	Test	Train	Test	Train	Test
1	5-2-1-1	1.00000	1.00090	0.78286	0.70624	0.00000	0.50000	0.74786	0.70871
	5-3-2-1-1	1.00080	1.00543	0.06699	0.10800	0.00000	0.05882	0.07638	0.10957
	5-4-3-2-1-1	1.00040	1.00040	0.52871	0.79723	0.00000	0.66667	0.57853	0.80375
	5-5-4-3-2-1-1	1.00000	1.00080	0.55556	0.35054	0.00000	0.50000	0.64138	0.36441
	5-6-5-4-3-2-1-1	1.00160	1.00100	0.08344	0.14069	0.00000	0.03125	0.09257	0.14446
2	5-2-1-1	1.00281	1.00291	0.09899	0.17087	0.00000	0.02778	0.91034	0.17400
	5-3-2-1-1	1.00000	1.00261	0.42784	0.34887	0.00000	0.20000	0.39483	0.35111
	5-4-3-2-1-1	1.00100	1.00080	0.18634	0.23684	0.00000	0.04167	0.21159	0.24026
	5-5-4-3-2-1-1	1.00020	1.00040	0.12466	0.06625	0.00000	0.11111	0.17249	0.08839
	5-6-5-4-3-2-1-1	1.00050	1.00120	0.23636	0.14035	0.00000	0.07692	0.27365	0.14249
4	5-2-1-1	1.00040	1.00443	0.29801	0.51237	0.00000	0.28571	0.33650	0.51455
	5-3-2-1-1	1.00030	1.00352	0.83729	0.98994	0.00000	0.80000	0.92712	0.85070
	5-4-3-2-1-1	1.00000	1.00120	0.78142	0.58333	0.00000	0.33333	0.72941	0.58481
	5-5-4-3-2-1-1	1.00271	1.00463	0.07264	0.04639	0.00000	0.02857	0.09770	0.06863
	5-6-5-4-3-2-1-1	1.00000	1.00050	0.28910	0.15165	0.00000	0.50000	0.38491	0.37229

- In part A of the above table:

Columns 3-10 represent quantities of R^2, MAD, MSE and RMSE measures of some designed network architectures of ANN model divided to R^2, MAD, MSE and RMSE of ARIMA (2,1,1) model, for 1, 2 and 4 week ahead of weekly poultry retail price time series, respectively. Because the quantities of column 3 and 4 are bigger than 1 and the quantities of columns 4-10 are less than zero, for all observations allocated to train and test sets, we can find that ANN network architectures outperform the ARIMA (2,1,1) model for all considered perspectives.

- In part B of the above table:

Similarly, columns 3-10 represent quantities of R^2, MAD, MSE and RMSE measures of some designed network architectures of NNARX model divided to R^2, MAD, MSE and RMSE of ARIMA (2,1,1) model, for 1, 2 and 4 week ahead of weekly poultry retail price time series, respectively. Similarly, because the quantities of column 3 and 4 are bigger than 1 and the quantities of columns 4-10 are less than zero, for all observations allocated to train and test sets, we can find that NNARX network architectures outperform the ARIMA (2,1,1) model for all considered perspectives.

- In part C of the above table:

Similarly, columns 3-10 represent quantities of R^2, MAD, MSE and RMSE measures of some designed network architectures of NNARX model divided to R^2, MAD, MSE and RMSE of some designed network architectures of ANN model, for 1, 2 and 4 week ahead of weekly poultry retail price time series, respectively. Similarly, because the quantities of column 3 and 4 are bigger than 1 and the quantities of columns 4-10 are less than zero, for all observations allocated to train and test sets, we can find that NNARX network architectures outperform the ANN model for all considered perspectives.

Also, table 4 shows:

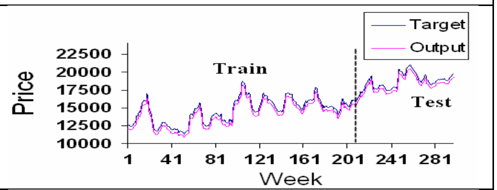
- The quantities of R^2, MAD and RMSE measures of the best designed architectures of NNARX, ANN and ARIMA (2,1,1) model for 1, 2 and 4 week ahead of weekly poultry retail price time series.
- The fitness of the best designed architectures of NNARX, ANN and ARIMA (2,1,1) model for 1, 2 and 4 week ahead of weekly poultry retail price time series.

Table4. Best designed architectures of NNARX, ANN and ARIMA models

NNARX — **Fitness**

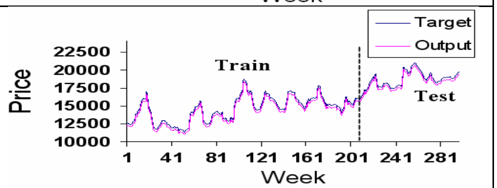
1 week ahead — 5-3-2-1-1

R^2		MAD		RMSE	
Train	Test	Train	Test	Train	Test
0.99990	0.99980	0.00042	0.00139	0.00054	0.00142



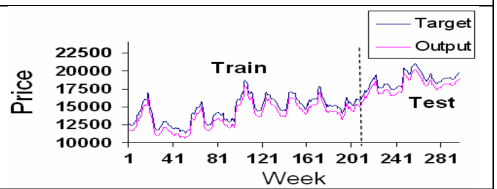
2 week ahead — 5-5-4-3-2-1-1

R^2		MAD		RMSE	
Train	Test	Train	Test	Train	Test
0.99990	0.99950	0.00046	0.00064	0.00074	0.00086



4 week ahead — 5-6-5-4-3-2-1-1

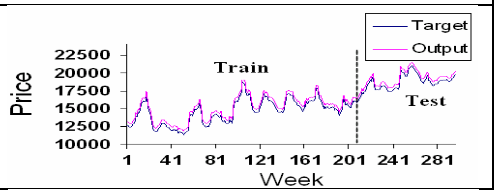
R^2		MAD		RMSE	
Train	Test	Train	Test	Train	Test
0.99990	0.99730	0.00061	0.00069	0.00102	0.00172



ANN — **Fitness**

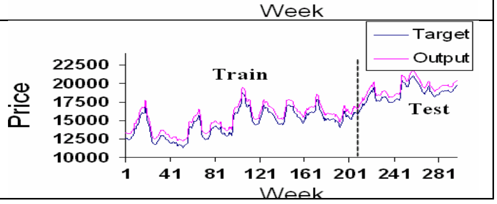
1 week ahead — 5-5-4-3-2-1-1

R^2		MAD		RMSE	
Train	Test	Train	Test	Train	Test
0.99990	0.99840	0.00117	0.00465	0.00145	0.00472

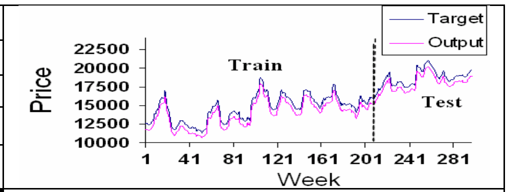


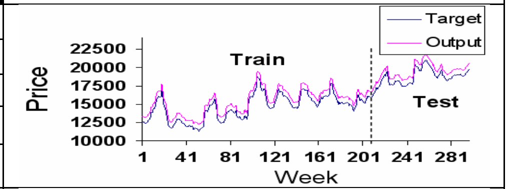
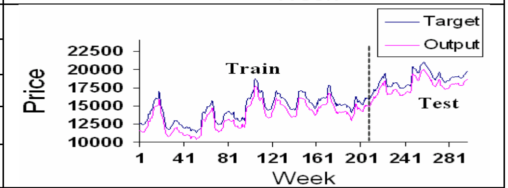
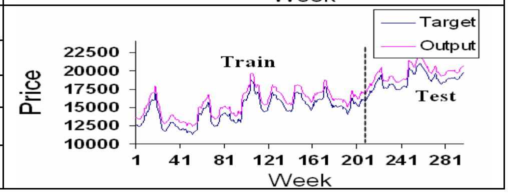
2 week ahead — 5-3-2-1-1

R^2		MAD		RMSE	
Train	Test	Train	Test	Train	Test
0.99990	0.99660	0.00194	0.00665	0.00271	0.00675



4 week ahead					
5-6-5-4-3-2-1-1					
R^2		*MAD*		*RMSE*	
Train	Test	Train	Test	Train	Test
0.99990	0.99680	0.00211	0.00455	0.00265	0.00462



ARIMA						Fitness
1 week ahead						
(2,1,1)						
R^2		*MAD*		*RMSE*		
Train	Test	Train	Test	Train	Test	
0.95950	0.94750	0.01075	0.01823	0.01618	0.02208	
2 week ahead						
(2,1,1)						
R^2		*MAD*		*RMSE*		
Train	Test	Train	Test	Train	Test	
0.95350	0.94350	0.01190	0.01977	0.01731	0.02301	
4 week ahead						
(2,1,1)						
R^2		*MAD*		*RMSE*		
Train	Test	Train	Test	Train	Test	
0.94950	0.94150	0.01245	0.01997	0.01801	0.02311	

According to the above table:

- The best network architectures for forecasting 1, 2 and 4 week ahead of weekly poultry retail price time series via NNARX model are 5-3-2-1-1, 5-5-4-3-2-1-1 and 5-6-5-4-3-2-1-1 architectures, respectively. Because of the highest quantity of R^2 measure, the lowest quantity of MAD and RMSE measures among the other architectures.
- Similarly, the best network architectures for forecasting 1, 2 and 4 week ahead of weekly poultry retail price time series via ANN model are 5-5-4-3-2-1-1, 5-3-2-1-1 and 5-6-5-4-3-2-1-1 architectures, respectively. Because of the highest quantity of R^2 measure, the lowest quantity of MAD and RMSE measures among the other architectures.
- Similarly, The best architecture for forecasting 1, 2 and 4 week ahead of weekly poultry retail price time series via ARIMA model is (2,1,1) architecture. According to identified degree of integration (*d*), autoregessive (*p*) and moving average (*q*) by Dikey-Fuller, correlation and partial correlation diagrams.

8. Conclusions and Discussions

Non-linear processes are usually too complicated for accurate modeling by traditional and statistical models, therefore there are always rooms for alternative model types such as the data based models. Clearly, more research is needed to see if and how the proposed scheme could help the development of efficient models.

We did not find big body of study done among the economic forecasting literature that can empirically prove the hypothesis of being more powerfulness of dynamic neural networks in comparison with the static neural networks for forecasting. This lack motivated us to carry out this study. We compared the utilization of NNARX as a nonlinear dynamic neural network model, ANN as a nonlinear static neural network model and ARIMA as a linear model for forecasting. As an empirical application, we compared the various forecasting performance of mentioned models for three perspectives (1, 2 and 4 week ahead) of Iran poultry retail price weekly time series via common forecast performance measures. We obtained the weekly poultry retail price time series of Iran for the period 2002:3-2007:12

from the website of Iran State Livestock Affairs Logistics. Also, we considered the period 2002:3-2006:3 (70% of total observations) and 2006:3-2007-12 (30% of total observations) for training and testing of all models, respectively. We found that, NNARX and ANN models outperform ARIMA model and NNARX model outperforms ANN model for all perspectives. Also, we found that the best network architectures for forecasting 1, 2 and 4 week ahead of weekly poultry retail price time series via NNARX model are 5-3-2-1-1, 5-5-4-3-2-1-1 and 5-6-5-4-3-2-1-1 architectures, via ANN model are 5-5-4-3-2-1-1, 5-3-2-1-1 and 5-6-5-4-3-2-1-1 architectures and via ARIMA model is (2,1,1) architecture, respectively.

Reference

[1.] G. E. P. Box, G. M. Jenkins, Time series analysis: Forecasting and control, San Francisco: Holden-Day, 1970.

[2.] X. Chen, J. Racine, R. N. Swanson, Semiparametric ARX Neural Network Models with an Application to Forecasting Inflation. Neural Networks 12 (2001) 674-683.

[3.] G. Chengyu, K. Danai, Fault diagnosis of the IFAC Benchmark Problem with a model-based recurrent neural network. Proceedings of the 1999 IEEE International Conference on Control Applications 2(1999)1755–1760.

[4.] O. De Jesús, J.M. Horn, M.T. Hagan, Analysis of Recurrent Network Training and Suggestions for Improvements. Proceedings of the International Joint Conference on Neural Networks, Washington, DC, July 15–19 (2001)2632–2637.

[5.] O. De Jesús, M.T. Hagan, Backpropagation through Time for a General Class of Recurrent Network. Proceedings of the International Joint Conference on Neural Networks, Washington, DC, July 15–19 (2001)2638–2642.

[6.] O. De Jesús, M.T. Hagan, Forward Perturbation Algorithm for a General Class of Recurrent Network. Proceedings of the International Joint Conference on Neural Networks, Washington, DC, July 15–19 (2001)2626–2631.

[7.] J. Feng, C.K. Tse, F.C.M. Lau, A neural-network-based channel-equalization strategy for chaos-based communication systems. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 50(2003)954–957.

[8.] R. Gencay, Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. Journal of International Economics 47(1999)91–107.

[9.] P. Gianluca, D. Przybylski, B. Rost, P. Baldi, Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins: Structure, Function, and Genetics 47 (2002)228–235.

[10.] T. H. Hann, E. Steurer, Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. Neurocomputing10 (1996) 323–339.

[11.] Z. Haoffi, X. Guoping, Y. Fagting, Y. Han, A Neural Network Model Based on the Multi-Stage Optimization Approach for Short-Term Food Price Forecasting in China. Expert Systems with Applications 33(2007) 347-356.

[12.] S. Haykin, Neural Networks: a Comprehensive Foundation. Macmillan, New York, 1994.

[13.] S. L. Ho, M. Xie, The use of ARIMA models for reliability and analysis. Computers and Industrial Engineering 35(1998) 213–216.

[14.] Jayadeva, S. A. Rahman, A neural network with O(N) neurons for ranking N numbers in O(1/N) time. IEEE Transactions on Circuits and Systems I: Regular Papers 51 (2004) 2044–2051.

[15.] S.A. Kalogirou, Artificial intelligence for the modeling and control of combustion processes: a review. Progress in Energy and Combustion Science 29(2003)515–566.

[16.] I. Kamwa, R. Grondin, V. K. Sood, C. Gagnon, V. T. Nguyen, J. Mereb, Recurrent neural networks for phasor detection and adaptive identification in power system control and protection. IEEE Transactions on Instrumentation and Measurement 45(1996)657–664.

[17.] N.B. Karayiannis, A.N. Venetsanopoulos, Artificial Neural Networks: Learning Algorithms. Performance Evaluation and Applications. Kluwer Academic Publishers, Boston, USA, 1993.

[18.] A. Lapedes, R. Farber, Nonlinear signal processing using neural networks: Prediction and system modeling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM, 1987.
[19.] S. e. Markridakis, The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of Forecasting 1 (1982) 111-153.
[20.] S. e. Makridakis, S.C. Wheelwright, R.J. Hyndman, Forecasting: methods and applications, New York: John Wiley & Sons, 1998.
[21.] L.R. Medsker, L.C. Jain, Recurrent neural networks: design and applications, Boca Raton, FL: CRC Press, 2000.
[22.] J. S. Racine, On The Nonlinear Predictability of Stock Returns Using Financial and Economic Variables, forthcoming. Journal of Business and Economic Statistics 19 (2001) 80-382.
[23.] A.J. Robinson, An application of recurrent nets to phone probability estimation. IEEE Transactions on Neural Networks 5 (1994) 298-305.
[24.] J. Roman, A. Jameel, Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns. Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences 2 (1996) 454–460.
[25.] F. Rosenblatt, Principles of Neurodynamics, Washington, D.C.: Spartan Press, 1961.
[26.] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation. In D. E. Rumelhart, & J. L. McClelland (Eds.), Parallel distributed processing (pp. 318–362). Cambridge, MA: MIT Press, 1986.
[27.] W.S. Sarle, Neural network FAQ. Retrieved 11 February 2003, from ftp://ftp.sas.com/pub/neural/FAQ.html.
[28.] R. Sharda, R. Patil, Neural Systems: Foundations, Paradigms, Applications, and Implementations. Pergamon Press, 1990.
[29.] B. Wu, Model-free forecasting for non-linear time series (with application to exchange rates). Computational Statistics and Data Analysis 19 (1995) 433–459.
[30.] G. Zhang, M.Y. Hu, Neural Network Forecasting of the British Pound/US Dollar Exchange Rate. International Journal of Management Science 26 (1998) 495–506.