

Multilayer Neural Network with Multi-Valued Neurons in Time Series Forecasting of Oil Production

Igor Aizenberg¹, Leonid Sheremetov², and Luis Villa-Vargas³

¹ Texas A&M University-Texarkana, USA

igor.aizenberg@tamut.edu

² Mexican Petroleum Institute, Mexico

sher@imp.mx

³ Computer Science Research Center of the IPN, Mexico

lvilla@cic.ipn.mx

Abstract. In this paper, we discuss the long-term time series forecasting using a Multilayer Neural Network with Multi-Valued Neurons (MLMVN). This is complex-valued neural network with a derivative-free backpropagation learning algorithm. We evaluate the proposed approach using a real-world data set describing the dynamic behavior of an oilfield asset located in the coastal swamps of the Gulf of Mexico. We show that MLMVN can be efficiently applied to univariate and multivariate multi-step ahead prediction of reservoir dynamics. This paper is not only intended for proposing a novel model of forecasting but to study carefully several aspects of the application of ANN models to time series forecasting that could be of the interest for pattern recognition community.

Keywords: time series forecasting, MLMVN neural networks, oil production.

1 Introduction

An oilfield is described by a set of time series (TS) of fluids from petroleum wells (oil, gas and water), which are characterized by different starting points and mutual influence. Production performance is both controlled by the reservoir properties and is affected by operational constraints and surrounding wells performance. Since the rock and fluid properties of the reservoirs are highly nonlinear and heterogeneous in nature, production TS comprise high-frequency multipolynomial components, represent a long memory process and are often discontinuous (or piecewise continuous).

Several important tasks of petroleum reservoir engineering are concerned with the forecasting of oil production from the reservoir. Usually, production prediction problem is considered within several different settings [1]: i) prediction of existing wells based on that well's previous production data, ii) spatial prediction of a new infill drilling well based on the reservoir's model, and iii) backward prediction, known as "backcasting", for some brown fields with no record of the measured wells' production. In this paper, we limit the discussion to the former case.

Traditional methods of production prediction in petroleum engineering include decline curve analysis (DCA), black oil model history matching, exploration analogies

and exploration trend extrapolations. The main disadvantage of such tools is that they are based on subjective data interpretation: to pick the proper slope, to tune the parameters of the numerical simulation model in such a way that they keep the reasonable values, to interpret reservoir geology [2].

The underlying idea of TS forecasting is that patterns associated with past values in a data series can be used to project future values [3]. In real-life dynamic systems the task for a TS forecasting can be stated as follows: given measurements of one component of the state vector, reconstruct the (possibly) chaotic dynamics of the phase space and thereby predict the evolution of the measured variable [4]. The paper studies an oilfield behavior reflected in the oil well's monthly production TS analyzing both the architecture and the parameters (time lag, memory size, etc.) which better describe and are able to predict its dynamics on long time intervals.

Recently, we reported the application of pattern recognition techniques (the associative model) to oil production prediction [5]. The Gamma model showed very competitive behavior on short prediction horizons (up to one year) but, in the current state of the development of forecasting algorithm, had some difficulties on longer intervals. Several artificial neural networks (ANN) topologies have been studied in the literature in their application to the prediction of oil and gas production both in univariate and multivariate settings. Multi-layer perceptron (MLP) and recursive neural networks (RNN), such as NARX, Elman and Jordan RNN, can be applied for multi-step-ahead TS forecasting. In [1, 6] a forecasting model based on the use of MLP was suggested to predict existing and infill oil well performance using only production data. Garcia and Mohaghegh [8] used recurrent neural networks for forecasting natural gas production in the United States. In [9] the NARX networks have been studied for univariate forecasting of oil monthly production, Chakra et al. described higher order neural networks (HONN) applied to forecast water, oil and gas production [10]. On relatively short-term (6 - 18 months) forecasting intervals and rather small data sets (up to 10 TS) most of these models outperformed DCA results with mean absolute percentage error (MAPE) about 14 - 16% [6], but long-term forecasting is still a challenge.

In this paper, we analyze the problem of long-term forecasting using a Multilayer Neural Network with Multi-Valued Neurons (MLMVN) introduced in [11] and further developed in [12, 13]. We illustrate the representation of TS patterns with MLMVN, several aspects of the prediction problem as the prediction horizon increases (for up to 5-15 years) and compare the univariate and multivariate forecasting with real data from an oilfield located in the coastal swamps of the Gulf of Mexico.

2 The MLMVN Neural Network in Time Series Forecasting

MLMVN [13] consists of multi-valued neurons (MVN) with complex-valued weights, and this is its main distinction from a classical feedforward neural network. Using complex-valued inputs/outputs, weights and activation functions, it is possible to increase the functionality of a single neuron and a neural network, to improve their performance, and to reduce the training time [13, 14].

The discrete MVN was introduced in [15]. It implements a mapping between n inputs and a single output. Let $\epsilon_k = e^{i2\pi/k}$, where i is an imaginary unit, be the primitive k -th root of unity. Let $E_k = \{\epsilon_k^0, \epsilon_k, \epsilon_k^2, \dots, \epsilon_k^{k-1}\}$. An MVN input/output mapping is described by a function of n variables $f(x_1, \dots, x_n)$, which can be either of $f: E_k^n \rightarrow E_k$ or $f: O^n \rightarrow E_k$ (discrete MVN) or $f: O^n \rightarrow O$ (continuous MVN), where O is a set of points located on the unit circle. Such a function can be represented using $n+1$ complex-valued weights as follows [13]:

$$f(x_1, \dots, x_n) = P(w_0 + w_1 x_1 + \dots + w_n x_n), \quad (1)$$

where x_1, \dots, x_n ($x_j \in E_k, j=1, \dots, n$) are neuron inputs and w_0, w_1, \dots, w_n are the weights. P is the activation function of the neuron, which is for a discrete MVN:

$$P(z) = e^{i2\pi j/k}, \text{ if } 2\pi j/k \leq \arg z < 2\pi(j+1)/k, \quad (2)$$

where $j=0, 1, \dots, k-1$ are values of the k -valued logic, $z = w_0 + w_1 x_1 + \dots + w_n x_n$ is the weighted sum, $\arg z$ is the argument of the complex number z . Function (2) divides a complex plane onto k equal sectors and maps the whole complex plane into a subset of points belonging to the unit circle.

The MVN learning is based on the error-correction learning rule [16]:

$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|} (D - Y) \bar{X}, \quad (3)$$

where \bar{X} is the vector of neuron inputs complex-conjugated, n is the number of neuron inputs, D is the desired output of the neuron, $Y = P(z)$ is the actual output of the neuron, r is the number of the learning step, W_r is the current weighting vector, W_{r+1} is the following weighting vector, C_r is a learning rate (it is complex-valued in general, but in all simulations, which we have done in this work, we used $C_r = 1$), and $|z_r|$ is the absolute value of the weighted sum obtained on the r^{th} learning step.

The use of MVN as a basic neuron in a MLMVN was suggested in [11, 16]. It's most important advantage is the derivative-free backpropagation learning algorithm [11-13], which is constructed in the following way. Let w_i^{js} be the weight corresponding to the i^{th} input of the j^{th} neuron (j^{th} neuron of the s^{th} layer), Y_{js} be the actual output of the j^{th} neuron from the s^{th} layer ($j=1, \dots, m$), and N_s be the number of the neurons in the s^{th} layer. It means that the neurons from the $s+1^{\text{st}}$ layer have exactly N_j inputs. Let x_1, \dots, x_n be the network inputs. To obtain the local errors for all neurons, the global error ($\delta_{jm}^* = D_{jm} - Y_{jm}$) must be shared with these neurons. Hence, the errors of the m^{th} (output) layer neurons are:

$$\delta_{jm} = \delta_{jm}^* / t_m, \quad (4)$$

where jm specifies the j^{th} neuron of the m^{th} layer; $t_m = N_{m-1} + 1$. The errors of the hidden layers neurons are calculated as follows:

$$\delta_{js} = \frac{1}{t_s} \sum_{j=1}^{N_{s+1}} \delta_{js+1} (w_i^{js+1})^{-1}, \quad (5)$$

where js specifies the j^{th} neuron of the s^{th} layer ($j=1, \dots, m-1$); $t_s = N_{s-1} + 1$, $s=2, \dots, m$ is the number of all neurons in the layer $s-1$, and $t_1 = n+1$ (n is the number of network inputs). The weights for all neurons are corrected using the error-correction learning rule (3) adapted to MLMVN.

3 Time Series Prediction Using MLMVN

Time series prediction using MLMVN was first considered in [13], where univariate forecasting of financial time series was studied. It was shown that MLMVN learning capability does not depend on the size of the network, a big network can be successfully used for a long term time series prediction. We use here two natural models, which are very suitable for MLMVN.

3.1 Univariate Forecasting Model

The model of a "classical time series" [17], is based on the assumption that the following series member depend only on a certain amount of its direct predecessors. Suppose we have historical data for some TS $x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}, \dots, x_r$ and there exist some functional dependence among the series members, according to which the $n+1^{\text{st}}$ member's value is a function of the preceding n members' values (6). Our task is to predict the following members of the series, that is x_{r+1}, x_{r+2}, \dots , which are not known. According to our assumption, (6) holds for our TS, but f is unknown. However, we can approach this function using some machine learning tool. This means that we have to form a learning set from the known TS members. Since the first r members of the TS are known, and according to (6) each following member is a function of the preceding n members, our learning set should contain the learning samples and desired outputs. As soon as the learning process is completed, f can be implemented as its approximation \hat{f} , which is resulted from the learning process, and future members of the TS can be predicted according to (7). The hat sign in (7) means that the corresponding value is not a true value, but the predicted one.

3.2 Multivariate Forecasting Model

This model is a generalization of the first model for such a case when a TS member to be predicted depends not only on the preceding members of the same series, but on

the members of another TS. Let $y_0 = y(t_0), y_1 = y(t_1), \dots, y_n = y(t_n), \dots$ be another TS changing and measured synchronously with the first one. Let us suppose that there exist some functional dependence among the series members, according to which the $n+1^{\text{st}}$ member's value is a function of the preceding n members' values of both series.

$$\begin{aligned}
 x_n &= f(x_0, \dots, x_{n-1}) & \hat{x}_{r+1} &= \hat{f}(x_{r-n+1}, \dots, x_r) \\
 x_{n+1} &= f(x_1, \dots, x_n), & \hat{x}_{r+2} &= \hat{f}(x_{r-n+2}, \dots, x_r, \hat{x}_{r+1}), \\
 x_{n+2} &= f(x_2, \dots, x_{n+1}), & \hat{x}_{r+3} &= \hat{f}(x_{r-n+3}, \dots, x_r, \hat{x}_{r+1}, \hat{x}_{r+2}), \\
 \dots & & \dots & \\
 x_{n+j} &= f(x_j, \dots, x_{n+j-1}). & &
 \end{aligned} \tag{6}$$

Suppose we have historical data for some TS $x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}, \dots, x_r$ and $y_0, y_1, \dots, y_{n-1}, y_n, y_{n+1}, \dots, y_r$. Suppose that there exist the functional dependence (8) among their members. Our task is to predict the following members of the series, that is x_{r+1}, x_{r+2}, \dots , which are not known. According to our assumption, (8) holds for our TS, but f is unknown. In the same way as for the first model, f can be implemented as its approximation \hat{f} , which is resulted from the learning process, and future members of the time series can be predicted as in (9).

$$\begin{aligned}
 x_n &= f(x_0, y_0, \dots, x_{n-1}, y_{n-1}) & \hat{x}_{r+1} &= \hat{f}(x_{r-n+1}, y_{r-n+1}, \dots, x_r, y_r) \\
 x_{n+1} &= f(x_1, y_1, \dots, x_n, y_n), & \hat{x}_{r+2} &= \hat{f}(x_{r-n+2}, y_{r-n+2}, \dots, x_r, y_r, \hat{x}_{r+1}, y_{r+1}), \\
 x_{n+2} &= f(x_2, y_2, \dots, x_{n+1}, y_{n+1}), & \hat{x}_{r+3} &= \hat{f}(x_{r-n+3}, y_{r-n+3}, \dots, x_r, y_r, \hat{x}_{r+1}, y_{r+1}, \hat{x}_{r+2}, y_{r+2}) \\
 \dots & & \dots & \\
 x_{n+j} &= f(x_j, y_j, \dots, x_{n+j-1}, y_{n+j-1}). & &
 \end{aligned} \tag{8}$$

4 Experimental Settings and Results

In order to test the proposed forecasting model, it was applied to the data set comprising monthly production from 15 wells of an oilfield asset located in the coastal swamps of the Gulf of Mexico. Both models (7) and (9) have been tested using MLMVN software simulators. For the multivariate model only two dynamic inputs: oil and gas-oil ratio (GOR) were used. From a number of experiments it follows that $n=60$ is an optimal value in (7) and (9). Since the series members were measured once in a month, then this means that a current member depends on the ones for the preceding 60 months (exactly 5 years). The optimal length of the learning set corresponds to 120 months (exactly 10 years). This makes it possible to perform a long-term prediction (5 years or even more) using both models. Thus, the wells from the data set meet the following criterion: to have a production history for at least 15 years.

4.1 Experimental Settings

The first model (univariate) was tested using MLMVN $60 \rightarrow 2 \rightarrow 32768 \rightarrow 1$ (60 inputs, 2 neurons in the 1st hidden layer, 32768 neurons in the 2nd hidden layer, and 1 output neuron). The second model (multivariate) was tested using MLMVN $120 \rightarrow 8 \rightarrow 1$ (120 inputs - they still correspond to 60 months -, 8 hidden neurons in a single hidden layer, and 1 output neuron).

To transform TS values into numbers located on the unit circle (to be able to use them as MLMVN inputs and outputs), the following transformation was used:

$$u_j \in [a, b] \Rightarrow \varphi_j = \frac{u_j - a}{b - a} (2\pi - \delta) \in [0, 2\pi - \delta[; x_j = e^{i\varphi_j} ; j = 0, \dots, n, \quad (10)$$

where u_0, u_1, \dots, u_n is the original time series. If $u_{\min} = \min_{j=0,1,\dots,n} u_j$ and $u_{\max} = \max_{j=0,1,\dots,n} u_j$,

then a and b were chosen in the following way $a = \begin{cases} y_{\min} - 0.125(y_{\max} - y_{\min}), & y_{\min} > 0 \\ 0, & y_{\min} = 0 \end{cases}$

and $b = y_{\max} + 0.125(y_{\max} - y_{\min})$. This extension of the range is important to avoid closeness to each other of the numbers on the unit circle corresponding to minimal and maximal values of a time series. For the same purpose, a shift δ was used. In our experiments, we used $\delta = \pi/4$. Evidently, to return back to the original data scale, the inverse transformation is necessary:

$$x \in O, \arg x = \varphi; u = \frac{\varphi(b-a)}{2\pi - \delta} + a \quad (11) \quad x \in e^{is2\pi/k}; u = \frac{s(b-a)}{k-1} + a \quad (12)$$

According to previous experimental results [13], the MLMVN software simulator shows better accuracy when the standard deviation of the TS data is "squeezed" to a value of order $[0.1, 10.0]$. To achieve this effect, the initial data were normalized by division of all the series members by 10,000 or 100,000 depending on their initial range. The resulting series members were multiplied by the same number accordingly.

Since the MLMVN software simulator supporting the multivariate model can produce only discrete output (the output neuron of the network is discrete, that is (2) is its activation function), to approach the continuous output, the large number of sectors ($k=4096$) was used. This makes a sector's angular size negligently small and the neuron's performance is practically equivalent to the continuous case. To transform the actual data into the format suitable for MVN/MLMVN and vice versa, the equations (10) - (11) should be accordingly transformed into equation (12).

To control the learning process, the root mean square error (RMSE) was used. The results are compared on two different error measures: the Mean Square Error (MSE) and the adjusted Mean Absolute Percent Error, also known as symmetric (though actually it is not symmetric since over- and under-forecasts are biased) MAPE (SMAPE) normalized between 0% and 100%. These error metrics are computed as shown in equations (13) and (14) respectively, where y_i is the actual and \hat{y}_i - is the predicted value.

$$E_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (13)$$

$$E_{SMAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)} \quad (14)$$

4.2 Experimental Results

Experimental results are resumed in Table 1, which contains both the results for reference models (exponential smoothing – ES, ARIMA (1,1,2) and NARX neural network [10]) and the errors for learning and prediction periods measured in a number of time points. All the TS were used in all models but for different forecasting periods. It is important to note, that reference models were used for much shorter prediction periods than MLMVN (12 months for the NARX model and 24 months for smoothing and ARIMA) since there accuracy drastically degrades with time when stationarity hypothesis fails. It can be seen for W3, W7 and W13 cases (shadowed) on the observed 2-years intervals. The NARX model gives competitive results on a one-year period, but it requires more study to be applied for longer periods. In turn, the univariate MLMVN model was studied on a 5-years period and the multivariate one – for up to 15 years period. Fig. 1 illustrates the forecasting for two typical TS with low (W1) and high (W11) errors.

Table 1. Experimental results

	Mean	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Period		24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
SMAPE (ES), %	26.02	4.49	5.14	59.04	28.98	10.69	7.58	100	1.42	8.91	14.75	2.39	4.01	100	7.08	35.78
SMAPE (ARIMA), %	17.28	7.68	5.30	60.81	8.38	4.96	6.98	18.10	1.85	8.84	5.31	1.09	1.61	91.82	7.45	28.96
Period		12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
SMAPE (NARX), %	12.20	6.08	11.22	18.57	12.93	10.88	9.58	38.42	7.52	10.63	10.70	4.61	3.93	N/A	9.29	16.47
Period		120	122	142	113	121	145	156	141	163	165	156	130	52	44	63
MSE (LM)		4.30E+08	3.32E+07	1.25E+08	4.25E+08	4.64E+08	6.40E+07	2.39E+07	9.87E+08	2.19E+08	9.33E+07	3.98E+08	6.87E+08	4.76E+06	1.32E+07	1.37E+08
SMAPE (LM), %		2.3	1.97	1.68	2.41	1.47	1.53	1.11	6.32	3.87	15.1	3.12	3.47	8.28	1.53	7.43
Period		60	61	61	61	61	61	61	61	61	61	61	61	41	31	61
MSE (M1)		1.96E+08	1.76E+09	2.75E+09	9.57E+08	3.95E+09	2.24E+10	1.54E+09	1.55E+09	7.97E+08	8.58E+08	2.67E+09	2.67E+07	1.42E+07	1.57E+08	2.88E+07
SMAPE (M1), %	25.99	3.05	25.05	9.57	25.65	17.54	33.16	26.32	44.84	23.96	26.62	81.1	4.48	36.12	8.72	23.74
Period		144	61	61	71	120	144	154	80	102	104	95	129	21	13	62
MSE (M2)		9.51E+08	1.68E+08	6.89E+08	1.39E+09	1.34E+10	1.50E+09	3.68E+08	6.34E+08	1.26E+08	2.03E+08	4.57E+07	7.91E+07	1.27E+07	1.17E+09	4.66E+07
SMAPE (M2), %	17.81	7.87	9.36	28.54	27.05	29.84	12.31	10.51	13.13	14.92	17.34	6.14	11.51	25.78	N/A	35.03

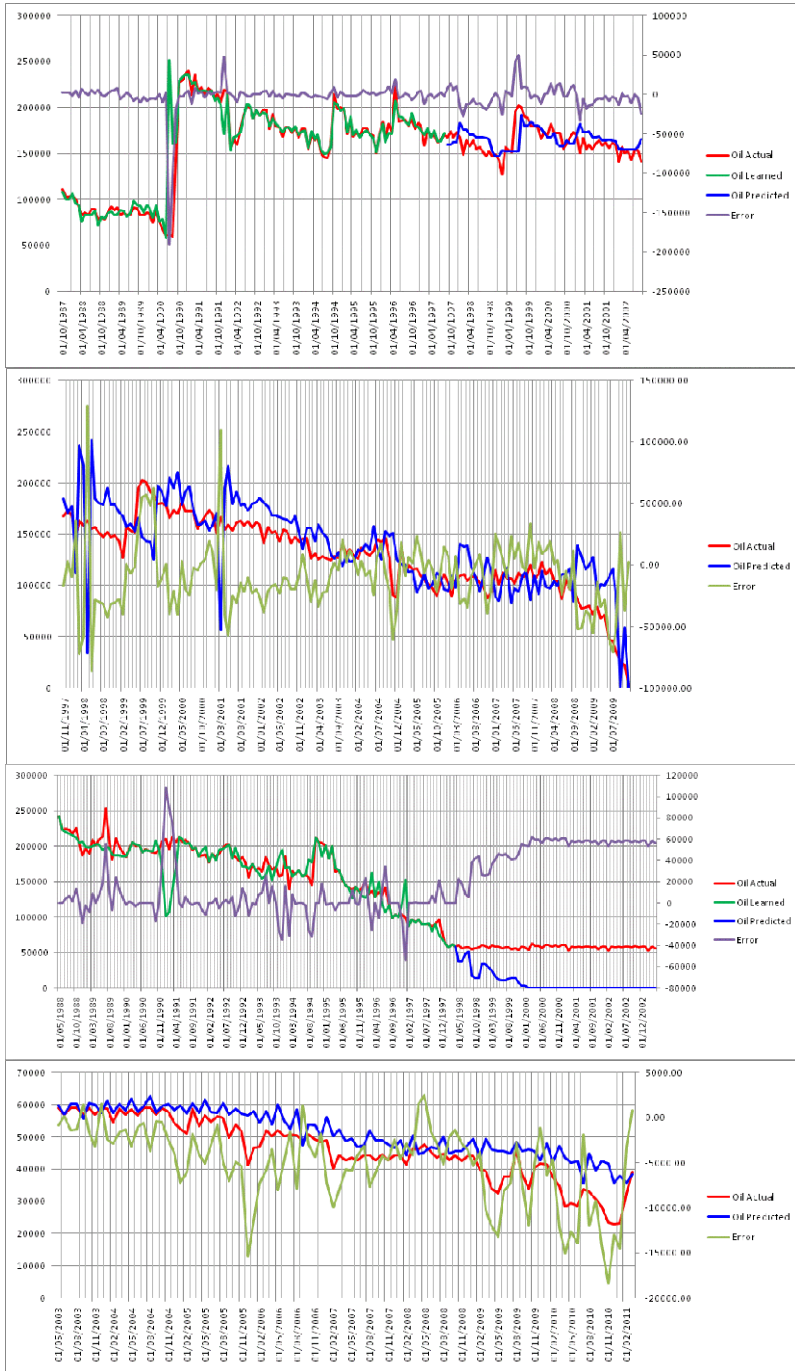


Fig. 1. Comparison of the univariate and multivariate forecasting results (in barrels per month): TS W1 model 1, TS W1 model 2, TS W11 model 1 and TS W11 model 2

Since only in one case (W3) the MSE and SMAPE showed the opposite results for both models, we'll use SMAPE for the rest of the discussion. As we can see the average SMAPE for the learning period is about 4%, being higher than 10% only for one TS (W10). In general, multivariate forecasting showed better performance (average SMAPE of 17% vs. 25%), though for several TS (W1, W3-W5, W12, W15) the univariate model showed better results (shadowed cells in Table 1). As we can see, the average error of model 2 is comparable to the errors reported in the literature (see the Introduction) for short-term forecasting models. Moreover, it permitted to considerably reduce the error for a number of TS which used to be very difficult for univariate forecasting (W2, W6-W11).

Let us consider the example of the W11 TS, showing extremely high error of 81% for the univariate model. For the forecasting period (comprising 12 years) the GOR pattern was relatively similar to the training period. As we can see from Fig. 1, for longer periods, the univariate model doesn't reflect properly the changes in the well's behavior: during the 10-years long training period till April 1998 the univariate network learns the declined behavior of the well and predicts it's shut-off for February 2000. Such prediction, by the way, is consistent with the DCA results for the same period (if the last slop is selected). Nevertheless, the bivariate model behaves differently and shows the SMAPE of 6.14% on a 12 years horizon. This example clearly shows the difference between DCA and ANN forecasting for long-term intervals.

Both models showed considerable difficulties while learning from (W10) and predicting (W3-W5, W13-W15) TS with zero values of production, which unfortunately is a typical situation when the wells are closed for some time for workovers or due to an accident. An interesting observation is that for the wells with high errors (average SMEPE 29%), the error of cumulative oil production for the tested period was as low as 6%. Another advantage of the multivariate model is the ability of forecasting of infill wells (mentioned as a second forecasting task in the Introduction), but this discussion is out of the scope of this paper.

Given that the network has up to 32768 neurons, it took about one hour to learn the model on the Xeon X5550 CPU @ 2.67 GHz (CPU mark 5414) and 64 bits OS.

5 Conclusions

In this paper, the application of MLMVN to long-term TS prediction has been studied on a real-life example of oil monthly production. Both univariate and multivariate models were developed. Compared to DCA method the advantage of ANN forecasting models is that prediction allows local variation instead of smooth curve projection as in the DCA.

As the experimental results showed, both models can be efficiently used and achieve results which are significantly better than prior published work in terms of the predicted interval and enables long-term prediction for up to 15 years (see W7, model 2). When there are no "gaps" in the data or these gaps are minimal, the qualitative long-term predictions are possible.

Acknowledgements. The authors would like to thank the CONACYT-SENER project 146515 for partial support of this work.

References

1. He, Z., Yang, L., Yen, J., Wu, C.: Neural-Network Approach to Predict Well Performance Using Available Field Data. In: SPE Western Regional Meeting, Bakersfield, California, pp. 26–30. SPE 68801 (March 2001)
2. Weiss, W.W., Balch, R.S., Stubbs, B.A.: How artificial intelligence methods can forecast oil production. In: SPE/DOE Improved Oil Recovery Symposium, Tulsa, Oklahoma, April 13–17, pp. 1–16 (2002)
3. Batyrshin, I., Sheremetov, L.: Perception based approach to time series data mining. *J. of Applied Soft Computing*. Elsevier Science 8(3), 1211–1221 (2008)
4. Davey, N., Hunt, S.P., Frank, R.J.: Time series prediction and neural networks. *J. of Intelligent and Robotic Systems* 31(1–3), 91–103 (1999)
5. López-Yáñez, I., Sheremetov, L., Yáñez-Márquez, C.A.: Novel Associative Model for Time Series Data Mining. *Pattern Recognition Letters* 41, 23–33 (2014)
6. Olivares-Velazquez, G., Escalona-Quintero, C., Gimenez, E.: Production monitoring using artificial intelligence. SPE paper 149594 (2012)
7. Silva, L.C.F., Portella, R.C.M., Emerick, A.A., Ebecken, N.F.F.: Predictive Data-Mining Technologies for Oil-Production Prediction in Petroleum Reservoir. SPE 107371 (2007)
8. Garcia, A., Mohaghegh, S.D.: Forecasting of natural gas production into year 2020: a comparative study. SPE 91413 (2004)
9. Sheremetov, L.B., González-Sánchez, A., López-Yáñez, I.: Time Series Forecasting: Applications to the Upstream Oil and Gas Supply Chain. In: IFAC Conference on Manufacturing Modelling, Management, and Control, vol. 7(1), pp. 957–962 (2013)
10. Chakra, C., Song, K.-Y., Saraf, D.N., Gupta, M.M.: Production forecasting of petroleum reservoir applying higher-order neural networks (HONN) with limited reservoir data. *Int. J. of Computer Applications* 72(2), 23–35 (2013)
11. Aizenberg, I., Moraga, C.: Multilayer Feedforward Neural Network based on Multi-Valued Neurons and a Backpropagation Learning Algorithm. *Soft Computing* 11(2), 169–183 (2007)
12. Aizenberg, I., Paliy, D., Zurada, J., Astola, J.: Blur Identification by Multilayer Neural Network based on Multi-Valued Neurons. *IEEE Transactions on Neural Networks* 19(5), 883–898 (2008)
13. Aizenberg, I.: *Complex-Valued Neural Networks with Multi-Valued Neurons*. SCI, vol. 353. Springer, Heidelberg (2011)
14. Hirose, A.: *Complex-Valued Neural Networks*, 2nd edn. SCI, vol. 400. Springer, Heidelberg (2012)
15. Aizenberg, N., Aizenberg, I.: CNN Based on Multi-Valued Neuron as a Model of Associative Memory for Gray-Scale Images. In: Proc. of the 2nd IEEE Int. Workshop on Cellular Neural Networks and their Applications, Munich, Germany, pp. 36–41 (1992)
16. Aizenberg, I., Moraga, C., Paliy, D.: A Feedforward Neural Network based on Multi-Valued Neurons. In: Reusch, B. (ed.) *Comp. Intelligence, Theory and Applications. Advances in Soft Computing*, vol. XIV, pp. 599–612. Springer, Heidelberg (2005)
17. Box, G., Jenkins, G., Reinsel, G.: *Time series analysis: forecasting and control*, 4th edn. Wiley (2008)