

A Simulation of the Broiler Growth Rate Using Artificial Neural Networks

Raquel Salazar¹, Fernando Rojano², and Abraham Rojano¹

¹ Universidad Autónoma Chapingo, Edo. México,
Mexico

² Le Magneraud Experimental station,
France

raquels60@hotmail.com

Abstract. To simulate the broiler growth the input variables were: day of year, vents opening, wind velocity, external temperature and absolute humidity, the maximum, average and minimum of the internal temperature and absolute humidity. For that purpose, two techniques were applied, a multi-layer perceptron (MLP) static Neural Network (NN) and the Layered Digital Dynamic Network (LDDN) which were applied to a set of experimental data obtained from a broiler cycle of production. The performance for both techniques was compared using: mean squared error (MSE), mean absolute error (MAE) and model efficiency (EF). The model evaluation measurements showed the superiority of the LDDN compared with MLP. The results from the sensitivity analysis found that the variable day of year was the most important variable to predict the broiler growth rate, so using this variable as the only input variable in the model, efficiency of 0.995 was reached for simulation.

Keywords: Day of a year, humidity, temperature, sensitivity analysis

1 Introduction

One of the advantages of modelling with neural networks is the ability to represent nonlinear systems with high complexity. For instance, a complex nonlinear discrete function can be approximated by a multi-layer Perceptron (MLP) model, where using sufficient number of neurons in the hidden layer can reach an acceptable approximation of any nonlinear function difficult to reach by means of using other modelling techniques.

Outnumbered applications of the Artificial Neural Networks (ANN) have been made in order to model agricultural production. One example is an ANN model used [1] to estimate performance in production of hens in a livestock building located in the South of Brazil. The use of the algorithm of back-propagation together with the training process allowed to generate an acceptable ANN model which can be used as a tool for

decision making by the technical staff in different production flocks to be based on scientifically objective criteria. Furthermore, in this work [1] the authors found that the ANN model allowed the simulation of consequences following the contribution percentage from each input to the poultry production.

An ANN model was developed by Galeano and Cerón [2] to estimate the weight of birds based only on the age of the bird, they found a very good performance with a correlation coefficient (R) of 0.99. The authors recommend the ANN as a viable option for modelling the animal production, because ANN has the ability of new variables inclusion and good adjustment between measured and predicted variables.

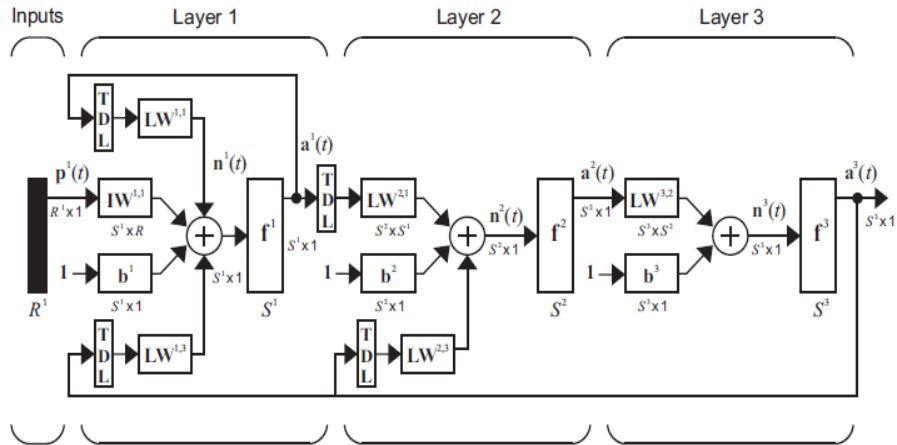
Sefat et al. [3] applied ANN for Modelling the Economic Efficiency of Broiler Production, the independent variables were the amount of consumable inputs (economic value in economic terms) and the dependent variable was the economic performance of production units. They found a NN model for the relationship between costs of different inputs used for production and benefit/cost ratio, they showed that a neural network architecture with 2 hidden layers (4 and 17 neurons in the first and second layers, respectively) provided the best results in estimating the model, with a correlation coefficient $R=0.96$ and $MSE=0.00024$. However, according to Masters [4] the additional hidden layers through which errors must be backpropagated makes the gradient more unstable, and the number of false minima increases. The only time that two hidden layers are required is when the network must learn a function having discontinuities.

The static neural networks, described above are not enough for modelling some phenomena, or when the number of patterns is small. Dynamic networks are generally more powerful than static networks (although somewhat more difficult to train). These contain delays and operate on a sequence of inputs. These dynamic networks can have purely feedforward connections, or they can also have some feedback (recurrent) connections. Their response at any given time will depend not only on the current input, but on the history of the input sequence. Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. These have applications in such diverse areas as control of dynamic systems, prediction in financial markets, fault detection, and even the prediction of protein structure in genetics [5].

Dynamic networks can be trained using standard optimization methods. However, the gradients and Jacobian matrix that are required for these methods cannot be computed using the standard backpropagation algorithm. Instead the dynamic backpropagation algorithms that are required for computing the gradients are the backpropagation-through-time (BPTT) and real-time recurrent learning (RTRL). In the BPTT algorithm, the network response is computed for all time points, and then the gradient is computed by starting at the last time point and working backwards in time [6] [7]. This algorithm is computationally efficient for the gradient calculation, but it is difficult to implement on-line. The Layered Digital Dynamic Network (LDDN) is a dynamic neural network that can be arranged in the form displayed in Fig. 1. The LDDN can be trained in the Neural Network Toolbox in Matlab [8].

As with the multilayer network, the fundamental unit of the LDDN is the layer. Each layer in the LDDN is made up of five components: A set of weight matrices that come into that layer (which may connect from other layers or from external inputs), any

tapped delay lines that appear at the input of a set of weight matrices, a bias vector, a summing junction, and a transfer function.



TDL=tapped delay line; LW= weights in the hidden layers; IW= weights in the input layer, b= bias unit; f= transfer functions between layers

Fig. 1. Layered Digital Dynamic Network (LDDN)

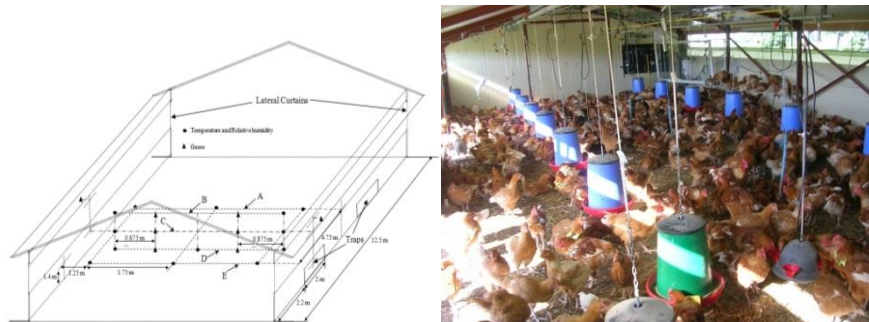


Fig. 2. Broiler geometry and sensor location.

The output of the LDDN is a function not only of the weights, biases, and the current network inputs, but also of outputs of some of the network layers at previous points in time. For this reason, it is not a simple matter to calculate the gradient of the network output with respect to the weights and biases. The weights and biases have two different effects on the network output. The first is the direct effect, which can be calculated using the standard backpropagation algorithm. The second is an indirect effect, since some of the inputs to the network are previous outputs, which are also functions of the weights and biases [9].

In this study we applied the LDNN to model the growth weight rate of the broiler production in an experimental station in France, a comparison between the performance between static and dynamic networks is made for this particular case. Also, a sensitivity

analysis is performed to detect the most important variable affecting the broiler growth rate.

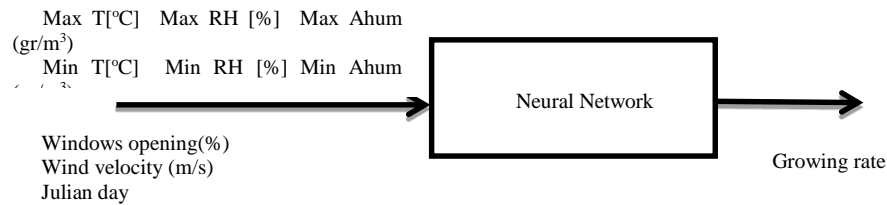


Fig. 3. Inputs and output for the Artificial Neural Network Model

2 Methodology

The data used in this study was collected from a cycle of organic production (84 days) in a broiler house at the experimental station “Le Magneraud” located in western France (46.15 N, -0.69 W). The broiler house had a volume of 158.4 m³ and was naturally ventilated through lateral curtains with a maximum opening of 0.53 m. The house had an eave height of 1.8 m. All the walls and roof were composed of foam, but an additional layer of steel panel was added to the exterior of the roof (see Fig. 2) .

The two heaters located in the house were uniformly distributed along the length, and each had a heating capacity of 4800 W. The experiment started on April 23 2014, the maximum broilers density was used, 750 broilers in an area of 75 m². The broiler house was occupied with 21,476 animals with a mortality rate during the production cycle less than 3 %. The weight gain found in daily basis is expressed in the following equation (R² = 0.993).

$$w = 0.0003346 \times d^2 + 0.0236706 \times d - 0.0245480, \quad (1)$$

where d is the n-th day of production cycle and w is the weight gain, in kg [10].

At the beginning of the growing cycle young broilers had a weight of about 0.5 kg, at the end of the growing cycle mature broilers had a weight of about 2.9 kg each. The broilers were kept indoors during the first 35 days; then two trap doors (length and height of 2 m and 0.53 m respectively) were opened to give the chickens access to a backyard during the day (9:00h to 17:00h).

Two types of neural networks were tested, first a static multilayer feedforward perceptron with one hidden layer and backpropagation algorithm. The activation function used is expressed in (2):

$$\tan \operatorname{sig}(n) = \frac{2}{1 + e^{-2n}} - 1, \quad (2)$$

Equation (2) is mathematically equivalent to hyperbolic tangent function (tanh(n)). It differs in that it runs faster than the MATLAB implementation of hyperbolic tangent function (tanh), the results can have very small numerical differences. This function is a good tradeoff for neural networks, where speed is important and the exact shape of

the transfer function is not [11]. The training set size and the hidden layer size are tied together. For the ANN's architecture, only a single hidden layer is used.

The dynamic neural network used in this work was the Layered Digital Neural Network (LDNN). was built with two delays in the input layer and two delays in the output layer, two hidden layers with three and two hidden nodes, the number of iterations was variable to find a good performance.

In both models static and dynamic, the input nodes receive an input vector, this input vector is composed of Julian day, windows opening, maximum, average and minimum temperature, maximum, average and minimum relative humidity, maximum, average and minimum absolute humidity, wind velocity, the output is the weight growth rate of the broilers as displayed in figure 3.

Three methodologies were applied to find out the number of nodes in the hidden layer, the first one was proposed by Hecht-Nielsen ($h = 2n + 1$) cited in Kůrková [11]; the second methodology was developed by Masters [4] ($h = \sqrt{mn}$); finally an additional intermedia criteria was applied ($h = n$); where n is the number of variables in the input layer, m number of variables in the output layer, in our case $m = 1$.

The artificial neural networks models generated were compared and the best were selected, based on their largest efficiency, lowest mean square error (MSE), as well as on the other statistical performance parameters described below.

3 Evaluation of the Model

The most widely used measure according to Wallach et al. [12] is the mean squared error (MSE), defined as:

$$MSE = (1/n) \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

where y_i is the measured value, \hat{y}_i is the corresponding simulated value, and n the number of measurements.

The mean absolute error (MAE) is expressed in (4):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4)$$

MAE has advantages over MSE if the objective is simply to examine the overall model error.

Model efficiency is defined as:

$$EF = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{MSE}{MSE_{\bar{y}}} \quad (5)$$

If the model is perfect then $y_i = \hat{y}_i$ for each i and $EF= 1$. If one uses the average of observed values as the predictor for every case $\hat{y}_i = \bar{y}$ for all i the $EF= 0$. A model that is a worse predictor than the average of observed values will have $EF= 0$ [12].

3.1 Sensitivity analysis

A sensitivity analysis was performed in the dynamic model to evaluate the importance of each input variable. One of the most important methods in sensitivity analysis is the backward stepwise method, it consists of step by step adding or rejecting one input variable and examining the effect on the output results. Based on the changes in performance measurements, a largest value in MSE, or a small value for efficiency, due to one input omission shows the most important input variable [13].

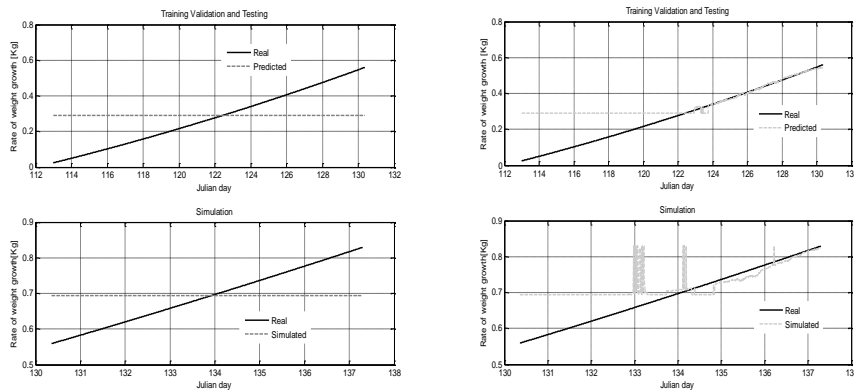


Fig. 4. Training, validation, testing and simulation of the Multilayer perceptron $h = \sqrt{mn} = 3$ (left), $h = \sqrt{mn} = 4$ (right)

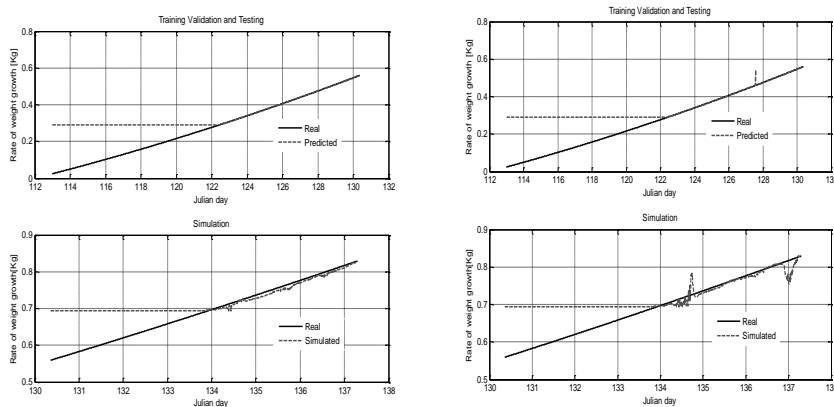


Fig. 5. Training, validation, testing and simulation of the Multilayer perceptron $n=h=12$ (left), $h=2n+1=25$ (right)

4 Results

The total data patterns available were 3500, the data was divided randomly, 2500 data patterns were used for training validation and testing and 1000 data patterns for simulation which correspond to 137.29 Julian days.

First the feedforward NN was implemented for the three cases of the number of nodes. The results are summarized in Table 1, Fig.4 and Fig.5. The performance of the static neural network is very poor especially when the first criterion is applied. The other two cases have better performance but still their efficiencies are less than 0.5.

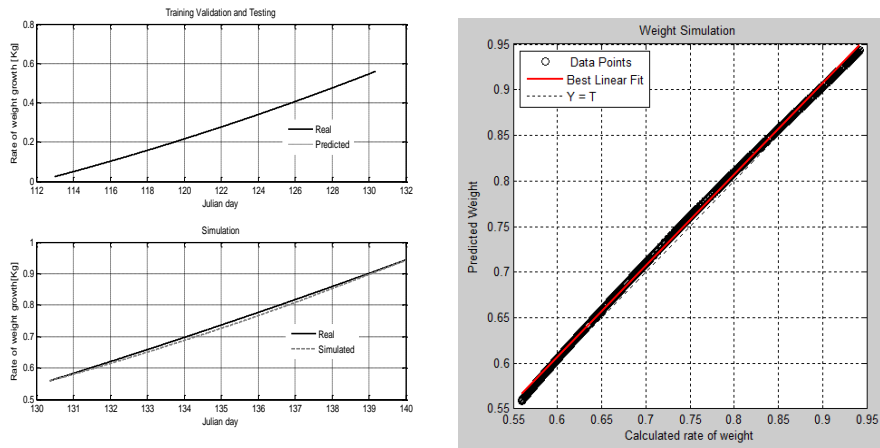


Fig. 6. Training, validation, testing and simulation of the LDNN with 20 iterations

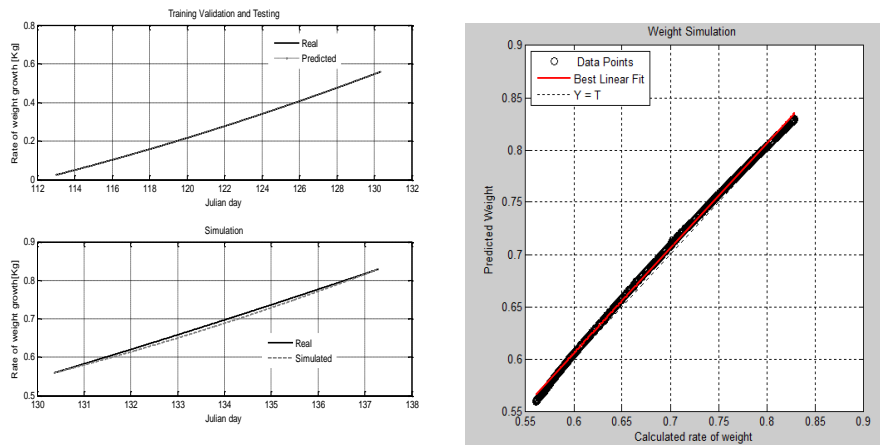


Fig. 7. Training, validation and testing of the LDNN with 50 iterations

A Dynamic Artificial Neural Network (DANN) was chosen because of its memory association and learning capability with sequential and time-varying patterns, which is most likely the biological situation [13]. Table 2, Fig. 6 and Fig. 7 displays a summary of the performance of the LDNN for different iterations. Only with 20 iterations was enough to reach an efficiency equal to one.

It is clear the superiority of the dynamic network in comparison to the static. A sensitivity analysis was performed for each input variable and results are shown in Table 3. The first row, second column of Table 3 display the efficiency of the model when all input variables are taken into account. The last column of Table 3 show a

negative efficiency when Julian day is rejected from the set of input variables, meaning that this is the most important input variable in the model. The efficiency decrease to 0.9875 when relative humidity is not in the model, which makes it the second most important variable. When windows opening or temperature are removed from the variables set the efficiency decrease to 0.9885, meaning that these two variables have the same importance in the model.

Table 1. Multilayer Perceptron Neural Network Model Performance

Number of neurons in the hidden layer	Training, validation and testing			Simulation		
	MSE	MAE	EF	MSE	MAE	EF
h=3 ($h = \sqrt{mn}$)	0.024	0.134	-0.01	0.006	0.067	-0.001
h=4 ($h = \sqrt{mn}$)	0.017	0.108	0.28	0.007	0.075	-0.265
h=12 ($n=h$)	0.013	0.076	0.422	0.003	0.037	0.475
h=25 ($h=2n+1$)	0.013	0.077	0.422	0.003	0.040	0.464

Table 2. Layered Digital Neural Network Model Performance

LDNN Iterations	Training, validation and training			Simulation		
	MSE	MAE	EF	MSE	MAE	EF
20	4.9447×10^{-8}	1.8763×10^{-4}	1	5.6825×10^{-5}	0.0069	0.995
50	7.67×10^{-8}	8.949×10^{-5}	1	3.879×10^{-5}	0.0057	0.993

Table 3. Sensitivity analysis simulation of the LDNN

Efficiency	0.993	0.9885	0.9874	0.9935	0.9885	0.9911	-0.47
Temperature	*		*	*	*	*	*
Relative Humidity	*	*		*	*	*	*
Absolute humidity	*	*	*		*	*	*
Wind opening	*	*	*	*		*	*
Wind Velocity	*	*	*	*	*		*
Julian day	*	*	*	*	*	*	

Table 4. LDNN performance using only Julian day as an input variable

LDNN iterations	Training, validation and training			Simulation		
	MSE	MAE	EF	MSE	MAE	EF
50	3.96×10^{-8}	1.631×10^{-8}	1	5.681×10^{-8}	0.0069	0.995

Given the high importance of the Julian day for the prediction, a run of the LDNN was implemented having only this input variable. Results for 50 iterations are shown in Table 4.

5 Conclusions

In this study two types of Neural Networks are compared for simulation of the growth weight rate for the broiler production in an experimental station in France, a feedforward static neural network, and the Layered Digital NN (LDNN) which is a dynamic NN with recurrent connections and delays in the inputs and output. The input nodes were Julian day, windows opening, maximum, average and minimum temperature, maximum, average and minimum relative humidity, maximum, average and minimum absolute humidity, wind velocity. The best static model reach an efficiency less than 0.5. Some of the arguments of using dynamic neural networks is that each iteration is more complicated because of gradient calculations; however in this particular case only with 20 iterations an efficiency equal to one is reached in training validation and testing and very close to one in the simulation.

Also, the sensitivity analysis shows that the most important input variable for the prediction of broiler growth rate is the Julian day, so the simulation was performed using only this variable and the results showed the superiority of the LDNN compared with static NN. The outcomes out of this model can be applied for prediction of the broiler growth rate using only the Julian day.

References

1. Salle, C.T.P., Guahyba, A.S., Wald, V.B. Silva, A.B., Salle, F.O., Nascimento, V.P.: Use of artificial neural networks to estimate production variables of broilers breeders in the production phase. *British Poultry Science* 44(2): 211–217 (2003)
2. Galeano, V.L., Cerón, M.M.: Modelación del crecimiento de pollitas Lohmann LSD con redes neuronales y modelos de regresión no lineal. *Rev. MVZ Córdoba* 18(3):3861–3867 (2013)
3. Sefat, M. Y., Borgaee, A. M., Beheshti, B., Bakhoda, H.: Application of Artificial Neural Network (ANN) for Modelling the Economic Efficiency of Broiler Production Units. *Indian Journal of Science and Technology*, 7(11):1820–1826 (2014)
4. Masters, Timothy: *Practical Neural Network Recipes in C++*. Academic Press Inc. London. (1993)
5. Laheeb, M. I.: Anomaly Network Intrusion Detection System based on Distributed Time-Delay Neural Network (DTDNN). *Journal of Engineering Science and Technology* 5(4):457–471 (2010)
6. Abdulla, Z., Kasmuri, N.H.: Comparison of Static and Dynamic Neural Network Models in Predicting Outlet Temperature of Shell and Tube Heat Exchanger Plant. In: *IEEE 9th International Colloquium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia (2013)
7. Medsker, L.R., Jain, L.C.: *Recurrent Neural Networks Design and Applications*. The CRC Press International Series on Computational Intelligence (2000)

8. Beale, M.H., Hagan, M.T., Demuth, H.B.: Neural Network Toolbox™ User's Guide. Mathworks Inc. (2015)
9. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L.: Accelerating the convergence of the backpropagation method. *Biological Cybernetics* 59: 257–263 (1988)
10. Rojano, A.F, Bournet, P. E., Robin, P. Hassouna, M.: Test of two different schemes through CFD to include heat and mass transfer induced by animals inside a broiler house. In: Proceedings International Conference of Agricultural Engineering, Zurich (2014)
11. Kůrková, V.: Kolmogorov's theorem and multilayer neural networks. *Neural networks* 5(3):501–506 (1992)
12. Wallach, D., Makowski, D., Jones, J. W., Brun, F.: Working with Dynamic Crop Models: Methods, Tools and Examples for Agriculture and Environment. Academic Press. 487 p. (2013)
13. Shojaeefard, M.H., Akbari, M., Tahani, M., Farhani, M.: Sensitivity Analysis of the Artificial Neural Network Outputs in Friction Stir Lap Joining of Aluminum to Brass. *Advances in Materials, Science and Engineering*. Volume 2013, Article ID 574914 (2013)