International Journal of Agricultural Technology 2016 Vol. 12(2):195-214
Available online http://www.ijat-aatsea.com
ISSN 2630-0192 (Online)

Artificial Neural Networks Model: Reliable Forecasting Tool in Cocoa Postharvest Losses Reduction

Adewumi, I. O.[1*], Orisaremi, K.[2], Ajisegiri, G. O.[2], Oladipo, O. O.[1], Kosemani, B. S.[1], and Adegbulugbe, T. A.[1]

[1]Department of Agricultural Engineering, Federal College of Agriculture, Ibadan, Nigeria; [2]Department of Industrial and Production Engineering, Faculty of Technology, University of Ibadan, Nigeria.

Adewumi, I. O., Orisaremi, K., Ajisegiri, G. O., Oladipo, O. O., Kosemani, B. S. and Adegbulugbe, T. A. (2016). Artificial neural networks model: reliable forecasting tool in cocoa postharvest losses reduction. International Journal of Agricultural Technology 12(2):195-214.

Abstract This research involves the development of an artificial neural network (ANN) model that forecasts the weekly production quantities of outputs for a typical cocoa processing company in order to reduce post-harvest losses. The artificial neural network was initially built with a single input and a single output with the aid of the Neurosolutions 5.07 software package. It was then trained, cross- validated and tested by carrying out a successful pilot test using raw production data obtained from the cocoa processing company. The data set consists of two input variables and two output variables, and the relationship between any input and output variable is complex. Input variables are the weekly quantities of cocoa bags tipped and batches of cocoa nibs roasted, while output variables are weekly quantities of cocoa butter and cocoa cake packaged in cartons. On training the networks, the parameters of specific networks found to give an acceptable mean square error (MSE) were recorded. The network was later modified using different combination types of input(s) and output(s). The model outputs were found to be satisfactory, lying within the defined error limit when compared to the actual outputs. The result shows that the network developed was able to predict the output quantities with a high accuracy, as the training and cross-validation errors at all times both lie within the target error of 0.0001 as specified by the software developers. The network's ability in forecasting these outputs with a high degree of accuracy goes a long way in demonstrating that artificial neural networks are highly capable of forecasting in situations when there is no closed-formed mathematical relationship between input and output.

Keywords: Artificial Neural Network, Cross-Validation, Moisture Content, Postharvest loss

Introduction

TamilNadu Agricultural University (2013) has been clearly stated that Post harvest loss reduction technology encompasses the usage of optimum harvest

***Coressponding author:** Adewumi, I. O.; **Email**: adexio2010@gmail.com

factors, reduction of losses in handling, packaging, transportation and storage with modern infrastructure machinery, processing into a wide variety of products, home scale preservation with low cost technology. ACIAR (2001) has reported the use of thermal processing, low temperature, drying, chemical and biological reactions coupled with other preservation techniques are applied to enhance the storability. Containers and packaging materials confer portability as well as extend the shelf-life. Adoption of these techniques could make available a large quantity of food by avoiding losses and provide better quality food and nutrition, more raw materials for processing, thus ensuring better returns to the farmers. Importance of Post-harvest technology lies in the fact that it has the capability to meet food requirement of growing population by eliminating losses making more nutritive food items from raw commodities by proper processing and fortification.

FAO (1997) has discovered that it is always difficult to distinguish clearly between the arbitrarily defined stages from production to consumption. The maturing/drying/processing periods will often overlap during the post-harvest period, as, for example, in the fielddrying of maize after it has reached maturity. There is nothing to be gained by defining rigid boundaries and making artificial distinctions between overlapping stages. It may be preferable to relate losses to a process or operation rather than to a definite period. The main agents causing deterioration of stored produce as discussed by Aremu (2012) includes microorganisms (fungi, bacteria and yeasts), insects and mites, rodents, birds and metabolic activity. Jane (2004) has observed that Cocoa and Coconut industries are placed 3rd and 4th in value and share of major agricultural export industries which contribute to the our national economy from export of dried cocoa beans, copra, crude copra oil and copra meal to overseas traders (e.g. buyers, processors and manufacturers)

Orisaremi (2009) has stated that the attainment of the goals of any production firm is largely dependent on the manager’s ability to refine, maintain and most importantly forecast the parameters of the business environment in which he or she operates. The output parameters are of utmost importance because their predictions are required for enhancing the quality of the decisions made by managers and policy makers of the firm regarding ways of increasing both productivity and profitability. The finance and accounting department of most firms use forecast for budgeting planning and cost control. Production and operations personnel use forecast to make periodic decisions involving process selection, capacity planning, facility layout, production planning, scheduling and inventory.

Consider a cocoa processing company where cocoa powder and cocoa butter (outputs) are processed from roasted cocoa nibs (input). Cocoa butter, the main source of sales return, is a highly demanded product by pharmaceutical companies because it serves as part of the raw materials in the manufacture of drugs that come in capsules. The respective quantities (in cartons) of the outputs: cocoa powder and cocoa butter tend to fluctuate randomly with the number of batches of the roasted cocoa nibs. This is due to sudden breakdown of some processing machines, shortage of raw materials (raw cocoa beans), quality of the supplied cocoa beans, moisture content of the cocoa beans (high or low), and frequent changeover of power source from PHCN to generating plants. All these factors have created some kind of weird and indecipherable relationship between the inputs and the outputs which cannot be analyzed mathematically. Consequently, management is faced with the problem of setting the basis for corporate long term planning.

This research work strives to develop, a prime stencil for the computation of the output quantities of a cocoa processing firm in Nigeria, taking into consideration all the above listed factors, and then build, train and test a neural network that will model all the complexities involved in the forecast of these quantities for the purpose of benefiting the firm.

Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems.' Artificial neural networks can be most adequately characterized as "computational models" with particular properties such as the ability to adapt or learn, to generalize, or to cluster or organize data, and which operation is based on parallel processing'(Krose and Van Der Smagt, 1996).

Why use neural networks

Castiglione (2000) has cited in Orisaremi (2009) has reported that Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Adewumi (2009) has highlighted other advantages of neural networks which include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

A Simple Neuron

Anderson (1987) has described an artificial neuron has a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

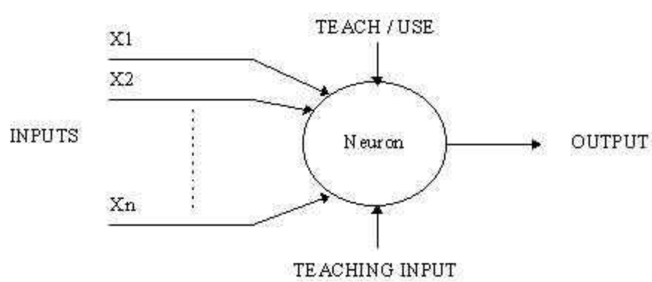

Figure 1. A simple neuron

Materials and methods

Model Development

The development of any model involves basically some steps. Every model is an abstraction of the real world. Various kinds of models do exist viz: narrative models, graphical models, mathematical models, etc. If real world situations are to be analyzed and modeled, such that their parameters desire quantitative values, then mathematical models will and always be the most appropriate as they give quantitative values which are of interest in this case. It is pertinent to state here that even though mathematical models appear to be the most suitable, a thorough understanding of the real world situation is required before it can be modeled appropriately.

The real world being modeled in this research work is the weekly amounts of the production outputs (cocoa butter and cocoa cake) of a cocoa processing company here in Nigeria, is an ever dynamic one, as so many factors both internal and external, tend to generate random fluctuations in the daily production quantities of the outputs. The two main factors that determine the quantities of the outputs are: the number of cocoa bags tipped and the number of batches of cocoa nibs roasted.

Data Collection and Classification

For ease of computation of results, the production data of a cocoa processing company over a three–year period (January 2006 to December 2008) were collected. The data consist of the following:

Inputs

I. Number of cocoa bags tipped

This simply refers to the total number of cocoa bags containing raw cocoa beans (each bag weighing approximately 64kg) that was utilized or processed during a 12 hour working period. The processing of the raw cocoa beans starts by emptying the bags into the cocoa cleaning plant manufactured by the Dutch company ‘Duyvis and Wierner’. After the cleaning process has been accomplished, the cocoa beans are micronized (i.e. subjected to longitudinal motion on the micronizer’s bed below a gas burner to absorb moisture from the cocoa beans). The cocoa beans are then broken into nibs and their chaffs separated from them by the aid of the winnower.

II. Number of batches roasted

The cocoa nibs from the winnower are transported by chutes to the Tornado bean batch roaster after winnowing. A single batch of cocoa nibs weighs approximately 1000kg and the total number of batches of cocoa nibs roasted is highly dependent on the number of cocoa bags tipped. It is therefore pertinent to state here that even though both are inputs to the company's final outputs; the total number of cocoa bags tipped serves as an input to the roaster's output (i.e. the total number of batches roasted). Management is interested in knowing the extent to which artificial neural networks can also be employed in forecasting the weekly quantities of the roaster's output by providing a new set of input data at the end of December 2009.

Outputs

I. Weekly quantities of cocoa butter packaged (in cartons)

Roasted cocoa nibs from the roaster are further processed by the aid of Pre-grinders, Fine mills and lastly through the cocoa presses to yield the finished products - cocoa butter and cocoa cake. Cocoa butter which serves as the major source of revenue for the company is packaged in 25kg cartons, and sold to some pharmaceutical companies in Spain. The respective quantities of both outputs are dependent on how well the inputs are utilized. A forecast is required to know the weekly quantities of cocoa butter that can be produced by the combination of both inputs.

II. Weekly quantities of cocoa cake sacks packaged

Another end- product of cocoa processing is cocoa cake. It is packaged in sacks. On a weekly basis, the processing of the roasted cocoa nibs from the roaster yields a higher production quantity of cocoa cake than cocoa butter. Some certain negative factors (i.e. erratic power supply, occasional breakdown of the cocoa presses, blockages in overhead pipe conveyors and chutes, etc.) sometimes do cause random fluctuations in the weekly quantities of cocoa cake sacks packaged that do not vary quite significantly with those of cocoa butter packaged. Regardless of these factors, it is of paramount importance that artificial neural networks be employed to detect the complex pattern of relationship existing between both inputs and the weekly quantities of cocoa cake sacks packaged and then providing an accurate forecast of these quantities.

The data were then divided into 4 sets namely: a supervised training data, an unsupervised training or validating data, the test data and an entirely new set of production data.

Steps in Neural Modeling

Below are the steps or set of procedures that were followed in the successful execution of this research work:

1. The necessary 'real life' data (i.e. the weekly production data of a cocoa processing company) discussed above were collected and analyzed critically for the purpose of detecting and getting rid of any discrepancies emanating from false production reports.
2. The three-year data were categorized into weeks i.e. a total of 156weeks, but later reduced to 128weeks,eliminating 28weeks of irregular/low production arising from plant breakdowns, execution of shutdown maintenance, scarcity of raw materials, erratic power supply, etc.
3. A complete set of the input (roasted batches of cocoa nibs) was modeled as a serial column onto the grid system of the Neurosolutions 5.07 software package. The aim of doing this was to determine the respective quantities of the output (cocoa butter) that can be generated by the same input.
4. By employing time series analysis and the Neurosolutions 5.07 software package, an artificial neural network (ANN) was built and trained to recognize the weird and indecipherable mathematical relationship between the input(s) and the output(s) as their variations are not quite correlated- i.e. in other words, it was trained to carry out the complex pattern recognition within the data.
5. In carrying out the preceding step, the supervised training data were used to train the built neural network, while the unsupervised data were used for cross-validation purposes and the test data for testing the accuracy of the network (i.e. by computing the mean square errors between the actual and forecasted outputs' quantities)
6. After pattern recognition within the data, the network was used to predict the weekly production quantities of the outputs given any set of input(s).
7. Using the neural graphical tool, the forecasted readings were analyzed to know when the company is likely to enjoy a differential marketing

advantage (huge profit) arising from a high productivity or a production/financial loss and how it can be curtained or prevented.

8. The results were then analyzed, conclusion drawn and recommendations made.

Results

Neural Network Development

The Neurosolutions 5.07 evaluation software package was installed and employed for the successful execution of this research work. In this program or package, there are four basic ways of constructing an artificial neural network (ANN) viz:

- Run the NeuralExpert program.
- Run the NeuralBuilder program.
- Run a pre-recorded macro (such as one of the demos) and modify the resulting network.
- Manually construct a network from a blank breadboard.

For ease of computation of results, the NeuralExpert program was loaded and used in building an artificial neural network. More advanced users who want more control of the topology and parameter settings may want to use the NeuralBuilder. It is important to note that breadboards built with either the NeuralExpert or NeuralBuilder can be modified later.

The NeuralExpert asks the programmer questions and intelligently builds a neural network. It configures the parameters and probes based on the description of the problem to be solved. Once a problem type is selected, all the questions the programmer needs to answer will be displayed in the panel on the left and one needs only to click on steps/numbers to navigate through the question and answer session. Once the network is built, the settings can be modified either directly on the breadboard or within the NeuralExpert.

The network architecture employed by the NeuralExpert is that of Multilayer perceptrons (MLPs). Multilayer perceptrons (MLPs) are layered feedforward networks typically trained with static backpropagation. These MLPs have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights). In the feedforward network, the units perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output and the units are arranged in a layered feedforward topology.

A single-layer network has severe restrictions: the class of tasks that can be accomplished is very limited. In this project work we will focus on feed-forward networks with layers of processing units.

Minsky and Papert (1969) showed that a two layer feedforward network can overcome many restrictions, but did not present a solution to the problem of how to adjust the weights from input to hidden units. An answer to this question was presented by Rumelhart and similar solutions appeared to have been published earlier (Werbos, 1974; Parker, 1985). The central idea behind this solution is that the errors for the units of the hidden layer are determined by back-propagating the errors of the units of the output layer. For this reason the method is often called the back-propagation learning rule. Back-propagation can also be considered as a generalization of the delta rule for non-linear activation functions and multilayer networks. The back-propagation algorithm was created by generalizing the Widrow-Hoff learning rule to multilayer networks as well as nonlinear differentiable transfer function. The back-propagation learning scheme compares a neural network's calculated output to a target output and calculates an error adjustment for each of the nodes in the network. The neural network adjusts the connection weights according to the error values assigned to each node, beginning with the connections between the last hidden layer and the output layer. After the network has made adjustments to this set of connections, it calculates error values for the next previous layer and makes adjustments. The back-propagation algorithm continues in this way, adjusting all of the connection weights between the hidden layers until it reaches the input layer. At this point it is ready to calculate another output (Microsoft Encarta 2008).The back-propagation uses the mean square error (MSE) as the performance function that measures the network performance by grading the network's performance.
The steps used in developing the network are:

- An initial network architecture was automatically selected (i.e. that of an MLP with multiple layers) by the NeuralExpert and the problem type set to prediction.
- The production data with an ASCII format was loaded into the NeuralExpert program; input and output columns were tagged.

- Cross validation data or level of generalization protection was set to 20% of the entire data and the level of neural network complexity was set to 'medium'.
- The network was finally built by the NeuralExpert by configuring the parameters and probes based on the description of the problem as shown in Figure 4.1.

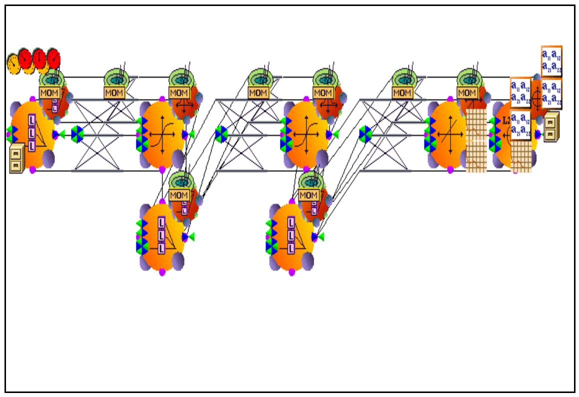

Figure 2. An artificial neural network built with one input and one output for a prediction problem

- The network was built on the basis of one input (number of batches roasted) and one output (i.e. packaged cartons of cocoa butter).
- To ensure accuracy of forecasts, the training epochs was set to 3000 and a series of iteratively conducted experiments were ran several times for the purpose of retaining the best network in terms of the minimum mean square errors (MSE) of the training and cross validation data.
- For each run, if under-learning occurs (if the cross validation and training error curves do not approach zero), the training epochs was increased to 5000.
- If over-learning should occur (i.e. when the cross validation error curve starts to rise after approaching zero such that the network starts to memorize the training data or when the network is stuck in local minima), the network was modified by removing some hidden layers and increasing the level of neural network complexity from 'medium' to 'high'.
- Once satisfactory test results were obtained, the network was rebuilt using different combinations of input(s) and output(s).

Mechanism of Development of the Models

The feed-forward neural network is a network of perceptrons with a differentiable squashing function, usually the sigmiodal function. The back propagation algorithm adjusts the weights based on the idea of minimizing the error squared. The differentiable squashing function allows the back propagation algorithm to adjust the weights across multiple hidden layers. By having multiple nodes on each layer, n-separable problems can be solved, like the Exclusive-OR, or the XOR problem, which could not be solved with only the perceptron. Figure 2 shows a fully connected feed-forward neural network; from input to output, each node is connected to every node on the adjacent layers.

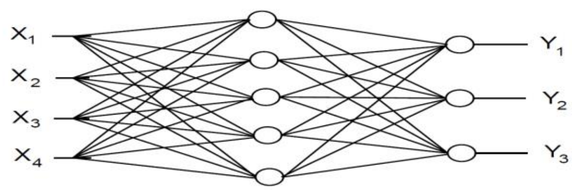

Figure 3. Fully-Connected, Feed-Forward Neural Network

In Figure 3, the individual nodes, or perceptrons, are representative of the neuron. The input to the node is the input to the neural network or, if the node is on a hidden layer or the output layer, the output from a previous layer. The node is the key to the training of the neural network. The back propagation algorithm propagates the changes to the weights through the neural network by changing the weights of one individual node at a time. With each iteration, the difference between the neural network's output and the desired response is calculated. In the case of a single output, the output of the entire neural network is the output of one individual node whose inputs are the outputs of nodes on the previous layer. By breaking the neural network down to the nodes, the training process becomes manageable. The back propagation algorithm is an LMS-like algorithm for updating the weights. Below is the derivation of the back propagation algorithm, which tries to minimize the square of the error (Rumelhart and McClelland, 1986).

The variables for the derivation of back propagation are defined as follows: X is the input vector of the node;

W is the vector of weights of the node;
y is the output of the node;
d is the desired response of the node;
e is the difference between output of the node and the desired response;
□ˆ □□□is the partial differential with respect to the weights;
s is the value inputted into the squashing function;
□□is the learning rate.

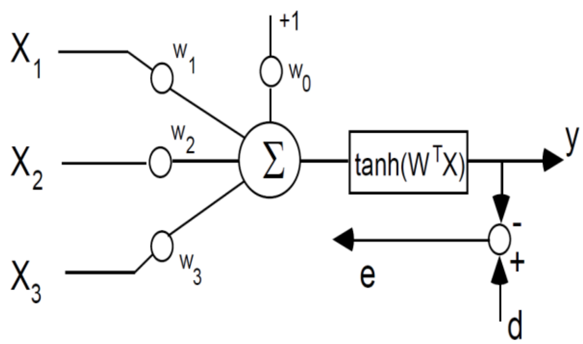

Figure 4. A Node

Equation 1 is the definition of the error

$$e = d - y \qquad (1)$$

Equations 2 and 3 are the partial differential of the error with respect the weights.

$$\nabla\hat{}\ \zeta = \partial e^T e / \partial W \qquad (2)$$

$$\nabla\hat{}\ \zeta = \partial / \partial W \{(d - y)^T (d - y)\} \qquad (3)$$

Equation 4 is the application of the chain rule to the partial differential.

$$\nabla\hat{}\ \zeta = -2\,(d - y)\,\partial y/\partial W = -2\,(d - y)\partial y/\partial s \bullet \partial s/\partial W \qquad (4)$$

Equation 5 is the derivative of the squashing function.

$$y = \tanh(s) \Rightarrow \partial y/\partial s = 1 - y^2 \qquad (5)$$

Equation 6 is the definition of s.

$$s = W^T X \qquad (6)$$

Equation 7 is the partial differential of s with respect to the weights of the node.

$$\partial s/\partial W = X \qquad (7)$$

Equation 8 is the substitution of variables into the partial derivative of the error squared.

$$\hat{\nabla} \zeta = -2\,(d - y)\,(1 - y^2)\,X \qquad (8)$$

Equation 9 is the change to the weights to be made.

$$\Delta W = -2\mu\,(d - y)\,(1 - y^2)\,X \qquad (9)$$

The equation for changing the weights is a very simple LMS-like equation that includes a single term not in the LMS equation. The term comes from the hyperbolic tangent function, whose derivative does not require much computational power. The simple weight update equation is applied to each node in the neural network. It is a gradient method that will converge to a local minimum.

During the training process, the inputs enter the neural network and get summed into the first layer of nodes. The outputs from the first layer of nodes get summed into the second layer of nodes. This process continues until the output comes from the neural network. The output is compared to the desired output, and the error is calculated. The error is used to adjust the weights backwards through the neural network. The weight adjustment equation has one shortcoming; the weights on a particular node cannot be the same as another node on the same layer because the weights will be adjusted the same for each node that has identical weights. If all of the neural network's weights were initially set at zero, the weights would be adjusted the same on each layer.

Mathematically, it would be equivalent to having a single node per layer. This is why the weights of the neural network need to be randomly initialized when there is a multi-layer neural network configuration. The other reason for randomly initializing the weights is to properly search the weight space, which is not a quadratic function, as is the linear perceptron. The randomly initialized weights make it very difficult to estimate the initial performance of the control system.

Analysis of the Pilot Test on the Network

The entire project data was analyzed by the NeuralExpert and divided into three sets (the training, cross validation and test data). In carrying out the first test on the built artificial neural network as required by the management of the company, only one input (i.e. the weekly quantities of the roasted batches) was fed into the input layer of the network. Although there are two inputs and two outputs, management's priority task was to give an estimate of how many cartons of cocoa butter could be processed from any given quantity of roasted batches. Thus the network was loaded with one input and one output. The quantities of the input were then processed by two hidden or output layers of PE's (processing elements) with hyperbolic transfer function to generate the corresponding quantities of the output (cocoa butter), and results obtained are shown in the tables below. For purpose of analysis, the column for desired outputs in all cases represents the weekly quantities of the actual outputs obtained from the company. The Neurosolutions 5.07 software package tags it "desired" to suit classification of data within its breadboard.

Table 1. Comparison of Desired and Neural Network Outputs for Cocoa

ROASTED BATCHES TRAINING DATA AND COMPARISON OF DESIRED AND NEURAL NETWORK OUTPUTS FOR COCOA BUTTER

	INPUT	OUTPUT	
Week	Roasted Batches	Des Butter(cartons)	Out Butter(cartons)
1	32	428.000000000000	424.047794684607
2	34	455.000000000000	457.066627770814
3	17	228.000000000000	223.929663693690
4	46	615.000000000000	616.574825071682
5	17	227.000000000000	227.507067013036
6	20	268.000000000000	269.795504120934
7	46	615.000000000000	614.431392050152
8	22	294.000000000000	293.884723258424
9	38	508.000000000000	509.250624407823
10	23	308.000000000000	304.374936141633
11	39	522.000000000000	522.755465401327
12	49	655.000000000000	659.351029037233
13	49	656.000000000000	655.236662353308
14	37	495.000000000000	497.726340946130
15	45	602.000000000000	598.695975781754
16	68	910.000000000000	913.775955919952
17	62	829.000000000000	833.425299851005
18	32	428.000000000000	427.202052560580
19	70	936.000000000000	932.790336084341
20	61	815.000000000000	818.708189411950
21	70	935.000000000000	933.237546841860
22	50	668.000000000000	664.745484543747
23	25	334.000000000000	332.037486684613
24	68	909.000000000000	912.883818532510
25	45	602.000000000000	603.939845559948
26	65	869.000000000000	871.831551467891
27	58	776.000000000000	776.972183569207
28	68	910.000000000000	904.401948163014
29	65	870.000000000000	873.987654907555
30	33	441.000000000000	439.074322613737
31	8	107.000000000000	122.244489710177
32	19	254.000000000000	251.988606514513
33	73	976.000000000000	976.675122304601
34	72	963.000000000000	962.618310264647
35	63	843.000000000000	840.541719944371
36	65	870.000000000000	867.028704126041
37	46	615.000000000000	611.039103109887
38	63	842.000000000000	845.090056138582

Table 1 (cont'd)

39	60	803.000000000000	803.407133901614
40	59	789.000000000000	789.654993000911
41	34	455.000000000000	455.426711250171
42	41	548.000000000000	543.757419126406
43	39	522.000000000000	522.524878036849
44	43	575.000000000000	576.949937720703
45	37	495.000000000000	495.483729870120
46	37	494.000000000000	496.114340479739
47	44	589.000000000000	588.141109671182
48	32	428.000000000000	431.493285107222
49	30	401.000000000000	399.510615904894
50	23	308.000000000000	307.838419938702
51	53	709.000000000000	710.966066359570
52	27	361.000000000000	364.951778221806
53	57	762.000000000000	762.468889587145
54	72	963.000000000000	961.394046806695
55	47	629.000000000000	629.528494386078
56	46	615.000000000000	609.850217041834
57	55	736.000000000000	732.744677458347
58	55	737.000000000000	738.571487965081
59	61	816.000000000000	818.907901758299
60	64	856.000000000000	855.146228166669
61	19	254.000000000000	258.393447141322
62	61	816.000000000000	813.338583468639
63	47	628.000000000000	630.945003052011
64	64	855.000000000000	858.708522415083
65	45	602.000000000000	601.057986015786
66	63	842.000000000000	839.887323555405
67	74	990.000000000000	982.324103925640
68	50	668.000000000000	667.388009028341
69	57	762.000000000000	759.260630631911
70	38	508.000000000000	508.983855280038

The mean square error (MSE) emanating from the training of this network was computed by the NeuralExpert program to be 0.000025059443 as the minimum error after a series of runs. This error lies within the training target error of 0.0001 per 5000 epochs of training as specified on the neurosolutions official website. The training was initially carried out on 1000 epochs but the results were not satisfactory because the training error was large, so the training was performed again using 4000 epochs before the optimum results were obtained. A closer look at the tables above, one would observe that the differences between the desired and network output quantities are relatively small, and this is mainly due to the fact that only one out of two inputs was used in building the network, and both inputs have a combined effect on any of the outputs (or both outputs together).In other words, better forecasts of the output quantities can be obtained if both inputs are loaded into the input layer of the network. In the succeeding sections, the network will be modified using both inputs and any one of the outputs as well as both inputs and the two outputs.

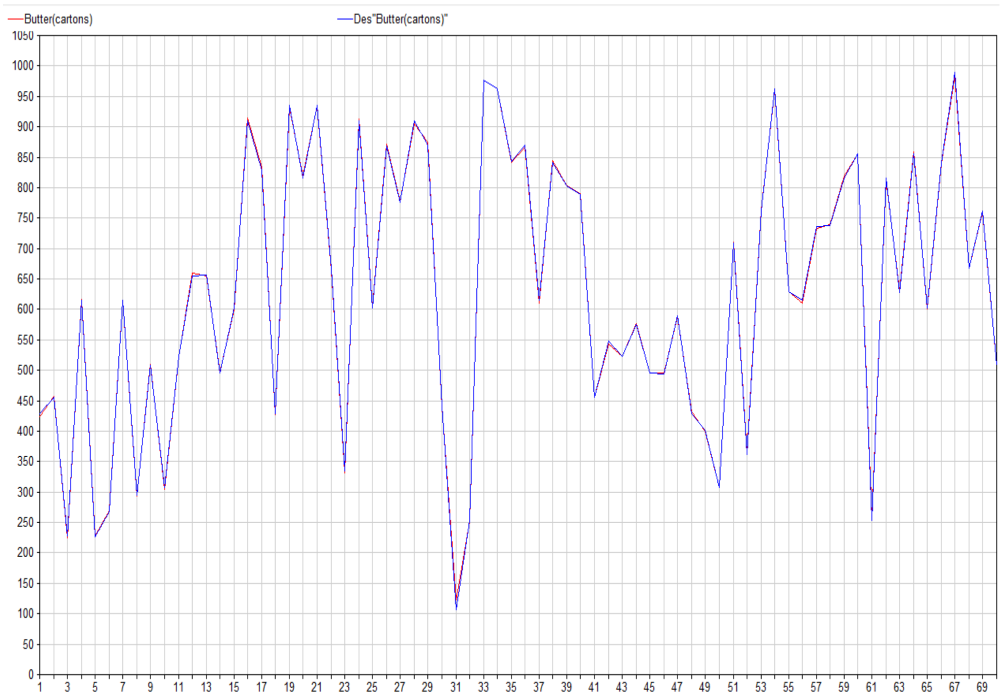

Figure 4. Graph of Desired and Network Outputs for Cocoa Butter using one Input during training

The graph above shows an almost perfect match between the desired and ANN output quantities. The red curve represents the network output, while the blue curve represents the actual or desired output. The x-axis represents the week number, while the y-axis represents the output quantities and this interpretation remains the same for all other graphs in the succeeding sections. The graph also displays random fluctuations of the actual or desired output quantities with time up to the 70th week.

Table 2. Comparison of Desired and Network Outputs for Cocoa Butter using one Input during Cross Validation

ROASTED BATCHES CROSS VALIDATION DATA AND COMPARISON BETWEEN DESIRED AND NEURAL NETWORK OUTPUTS

	INPUTS	OUTPUTS	
Week	Roasted Batches	Des Butter (cartons)	Out Butter (cartons)
77	47	629.000000000000	625.504695286585
78	36	482.000000000000	485.602053094774
79	50	669.000000000000	666.022012049196
80	56	749.000000000000	754.954201925402
81	56	747.000000000000	748.468829114389
82	12	161.000000000000	169.764636213342
83	39	522.000000000000	515.387554021377
84	53	708.000000000000	713.583926350352
85	45	601.000000000000	608.582747964472
86	37	494.000000000000	493.114344460639
87	50	670.000000000000	666.380839277016
88	32	428.000000000000	427.710894147869
89	37	496.000000000000	497.162561625593
90	23	309.000000000000	301.402628558214
91	20	268.000000000000	270.871613320014
92	21	281.000000000000	277.678624716490
93	53	709.000000000000	713.633486344797
94	30	400.000000000000	403.411581332284
95	16	214.000000000000	221.145680641413
96	5	67.000000000000	88.953925668081
97	42	562.000000000000	563.739352585311
98	8	107.000000000000	127.172922155047
99	60	803.000000000000	806.089949358001
100	71	950.000000000000	951.969734769957
101	51	682.000000000000	684.472596437695
102	48	642.000000000000	634.795436294773

The cross validation data also regarded as the level of generalization protection refers to data set aside to determine when to stop the training or to prevent over-training. During the network's training, 26 rows of the entire data was set aside as cross validation data and when this CV data was put to test by the program's testing wizard at the end of the training, the results obtained are shown in table 2 above. The cross validation data error obtained as computed by the NeuralExpert was 0.000106505246 and the plot of desired output against network's output for this CV data is shown below in Figure 4.5 with the same interpretation as Figure 4.

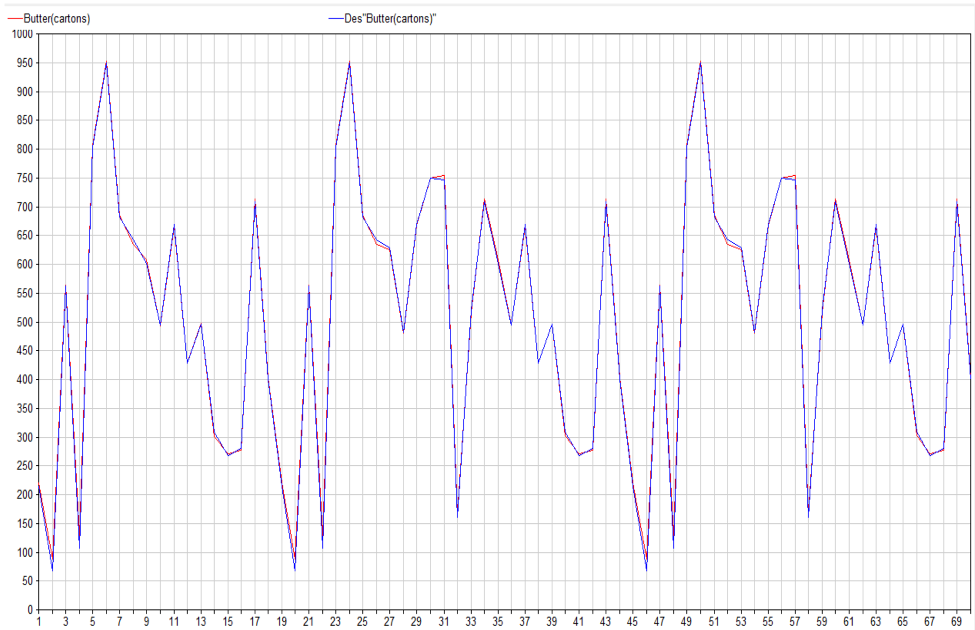

Figure 5. Graph of Desired and Network Output for Cocoa Butter using one Input during Cross Validation

References

ACIAR (2001). Postharvest Newsletter. A publication of the Australian Center for International Agricultural Research, Postharvest Technology Program. 58 pp.

Adewumi, I. O. (2009). Using genetic algorithm for solving traveling salesman problem. B.Sc Project Submitted to the Department of Industrial and Production Engineering, Faculty of Technology, University of Ibadan, Nigeria. 25 pp.

Anderson, J. A. and Rosenfeld, E. (1987). Neurocomputing: Foundations of Research. Cambridge, MA: The MIT Press.

Aremu (2012). Advance agricultural storage, Department of Agricultural & Environmental Engineering, Faculty of Technology, University of Ibadan, Nigeria.

Castiglione, F. (2001). Forecasting price increments using an artificial neural network. In Complex Dynamics in Economics, a Special Issue of Advances in Complex Systems 3:45-56.

Jane, R. O. (2004). The challenges in reducing postharvest losses: Lessons from the PNG cocoa and coconut industries. Seminar Presentation on Cocoa and Coconut Postharvest losses.

Krose, B., and Van der Smagt, P. (1996). An Introduction to Neural Networks 13 pp.

Minsky, M. and Papert, S. (1969). Perceptrons, MIT Press, Cambridge, M. A.

Orisaremi, K. (2009). Artificial neural networks for forecasting in production (A case study of a cocoa processing firm). Department of Industrial and Production Engineering, Faculty of Technology, University of Ibadan, Nigeria. 4 pp.
Parker, D. B. (1985). Learning logic: Technical Report TR-47. Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA.
Werbos, P. J. (1974). Beyond regression: New tools for prediction and analysis in the behavioural sciences . Harvard University.

(Received: 15 November 2015, accepted: 28 February 2016)