# Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand

G.E. Nasr, E.A. Badr and C. Joun
School of Engineering and Architecture
Lebanese American University, Byblos, Lebanon
e-mail: genasr@lau.edu.lb

**Abstract**

This paper applies artificial neural networks to forecast gasoline consumption. The ANN is implemented using the cross entropy error function in the training stage. The cross entropy function is proven to accelerate the backpropagation algorithm and to provide good overall network performance with relatively short stagnation periods. To forecast gasoline consumption (GC), the ANN uses previous GC data and its determinants in a training data set. The determinants of gasoline consumption employed in this study are the price (P) and car registration (CR). Two ANNs models are presented. The first model is a univariate model based on past GC values. The second model is a trivariate model based on GC, price and car registration time series. Forecasting performance measures such as mean square errors (MSE) and mean absolute deviations (MAD) are presented for both models.

## Introduction

The estimation of future demand for gasoline consumption is central to the planning of transportation systems. Such planning is improved through investigating a whole range of forecasting techniques. In addition, in order to guarantee a regular supply of gasoline, vital to the economic cycle of a country, it is necessary to keep a reserve. Depending on how fast this form of energy can become available, this reserve bears the name of spinning reserve or of cold reserve. Overestimating the future gasoline consumption results in unused spinning reserve and underestimating the future consumption is equally detrimental. Thus, using accurate forecasting techniques becomes essential (Darbellay and Slama, 2000). For many years, researchers have sought to understand consumer response to changes in the price of gasoline so as to design effective energy and environmental policy (Puller and Greening, 1999). The majority of studies have estimated the price elasticity of gasoline demand using aggregate-level data. This large number of studies has produced greatly varying estimates of the price elasticity depending on the specification and the data used. Also, estimating gasoline demand using household level data rather than aggregate data provides estimates that reflect more closely how individual consumers respond to changes in gasoline prices or household income (Kayser, 2000).

Recent research activities in artificial neural networks (ANNs) have shown that ANNs have powerful pattern classification and pattern recognition capabilities. Thus, one major application area of ANNs is forecasting (Sharda, 2000). Also, ANNs are well suited for problems whose solutions require knowledge that is difficult to specify but for which there are enough data or observations (Zhang, Patuwo and Hu, 1998). Research efforts on ANNs and other statistical means used for forecasting are considerable (Nasr, Badr and Younes, 2001; Nasr, Badr and Dibeh, 2000; Saab, Badr and Nasr, 2000).

In this paper, two neural network models suited to forecast monthly gasoline consumption in Lebanon are built. The first model is a univariate and fully connected model based on past GC values. The second model is a trivariate not fully connected model based on past GC values, gasoline price (P) and car registration (CR). Historically, a large number of studies offer a direct relationship between energy consumption and the price of energy. This paper presents no exception in this regard, but also ties gasoline consumption to car registration, since a change in the volume of operating cars would also affect the consumption of gasoline. The data used in this study is extracted from a governmental report (Statistical Administration, 1993-1999). The predictions are made using feed-forward neural networks since they are able to learn nonlinear mappings between inputs and outputs. Also, we propose to replace the common mean square error (MSE) by the cross-entropy error function in which the error signal associated with the output layer is directly proportional to the difference between the desired and actual output values. The cross entropy function is proven to accelerate the backpropagation algorithm and to provide good overall network performance with relatively short stagnation periods.

## ANN Implementation

The study period spans the time period from 1993 to 1999. This period is used to train, test and evaluate the ANN models. The training of the models is based on a five-year training set, January 1993 to December 1997, while the testing stage covers the period from January 1998 to December 1998. The evaluation stage covers the period between January 1998 and December 1999. The backpropagation algorithm is used for training since it is proven to be highly successful in training of multilayered neural nets for forecasting purposes.

The training of the network by backpropagation consists of three stages:

- The feedforward of the input training pattern.
- The calculation and backpropagation of the associated error.
- The adjustment of the weights.

Before applying the BPN algorithm, two steps are required:

1. Initialize Weights: the weights from the input layer to the hidden layer and the weights from the hidden layer to the output layer are randomly chosen to have values between –0.5 and 0.5.
2. Normalize data: The monthly data are normalized in order to have values between 0.1 and 0.9. The formula used is the following:

$$\frac{Data - Min}{Max - Min} * (Hi - Lo) + Lo$$

Where:

Min = Monthly minimum data.
Max = Monthly maximum data.
Hi = 0.9 = Maximum value for the normalized data.
Lo = 0.1 = Minimum value for the normalized data.

The ANN models are implemented through an object oriented programming method using Java programming language. This program is used to train the net, test it and evaluate it. The program involves many steps:

- The weights are chosen randomly.
- The minimum test error is initialized to the maximum real value.
- The training data set is passed to the network more than once.
- Backpropagation is performed using the cross entropy error function as a stop criterion for learning without exceeding the specified maximum number of cycles or by using a fixed number of epochs.
- The network is tested using the testing data set and the final performance measures of the learning and testing set are computed. The test error figures are also evaluated.
- If the test error is less than the minimum test error, the weights are saved and the test error will be the minimum test error.
- Otherwise, the net will be trained in a second phase and new error measures are recorded.
- The network is evaluated by calculating the mean square error (MSE) and the mean absolute deviation (MAD).

## Cross Entropy Error Function

During the learning process, the ANN goes through stages in which the reduction of the error can be extremely slow. These periods of stagnation can influence learning times. In order to resolve this problem we propose to replace the mean square error (MSE) by cross entropy error function. Simulation results using this error function show a better network performance with a shorter stagnation period.

The original MSE function for all training patterns is given by

$$E_m = \frac{1}{m} \sum_{k=1}^{m} (t_k - y_k)^2$$

where $t_k$ represents the GC target value and $y_k$ is the GC actual network value.

In the backpropagation model, we minimize the error through iterative updates of weights for all training patterns. In practice, this approach enables the network to have a good performance but slow convergence to the final outcome. Therefore, in order to accelerate the BP algorithm and instead of minimizing the squares of the differences between the actual and target values summed over the output units and all cases, we propose the following cross entropy error function to be minimized:

$$E_m = \frac{1}{m} \sum_{k=1}^{m} [t_k \ \ln y_k \ + (1 - t_k) \ln(1 - y_k \ )]$$

To minimize the error $E_m$, each weight $w_{jk}$ is updated by an amount proportional to the partial derivative of $E_m$ with respect to the weight. Using the mean square error the partial derivative of $E_m$ with respect to $w_{jk}$ is

$$\frac{\partial E_m}{\partial w_{jk}} = \sigma.(y_k - t_k) y_k (1 - y_k) \cdot z_j$$

By using the cross entropy error function, the partial derivative of $E_m$ with respect to $w_{jk}$ becomes

$$\frac{\partial E_m}{\partial w_{jk}} = \sigma \cdot (y_k - t_k) \cdot z_j$$

Thus, the error signal, propagating back from each output unit, becomes directly proportional to the difference between target value and actual value leading to a better network performance with a shorter stagnation period.

## ANN Models

### Model I

Since present and future gasoline consumption (GC) depends on previous gasoline consumption, a univariate model is implemented. ANN model I requires previous GC as data input patterns and has the layered structure shown in Figure 1. This model is a fully connected model since each input unit broadcasts its signal to each hidden unit. The GC inputs represent four lagged gasoline consumption parameters (a future value is dependent on the last four month values). All input parameters are selected following extensive testing by varying the values of the learning rate, the momentum parameter, the slope parameter and the number of input units. Parameter values yielding lowest error figures are given in Table 1. The error measures used in this study are the mean square error (MSE) and the mean absolute deviation (MAD). The corresponding prediction curve is shown in Figure 2.

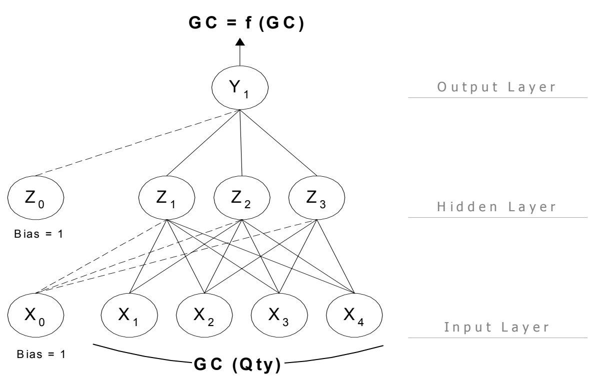


Figure 1: ANN Model I

Table 1: Model I Parameters and Errors

| ANN model I - (GC) | |
|---|---|
| Learning Rate ($\alpha$) | = 0.5 |
| Momentum ($\mu$) | = 0.5 |
| Slope ($\sigma$) | = 1 |
| No. Inputs (Q) | = 4 |
| No. Hidden units | = 3 |
| Error Function | = cross entropy |
| *Errors* | |
| MSE | = 793.7331 |
| MAD | = 24.3257 |

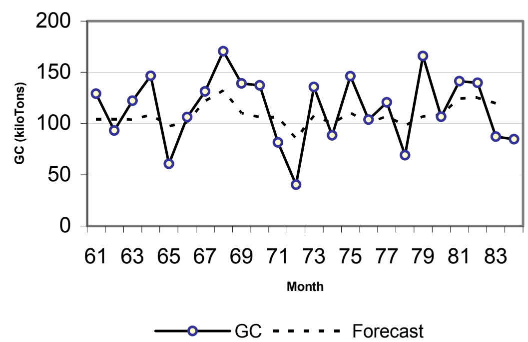


Figure 2. Actual and Forecasted Data using Model I

### Model II

The second ANN proposed model is a trivariate model which includes, in addition to previous GC data, the price of gasoline (P) and car registration (CR) time series as shown in Figure 3. Three P inputs (nodes 1-3) are fully connected to two hidden nodes (1-2), Four GC inputs (nodes 4-7) are also fully connected to two hidden nodes (3-4) and three CR inputs (nodes 8-10) are fully connected to two hidden nodes (5-6). Then all hidden neurons are fully connected to the output unit. The parameters yielding the lowest error values for this model are given in Table 2. It is important to note that the best trivariate model is achieved with four lagged GC terms, while the Price (P) and Car Registration (CR) have three lagged terms each. The results also show a significant improvement in forecasting accuracy for this trivariate model over the previous model. The predicted and the actual curve of gasoline consumption for this model are shown in Figure 4.

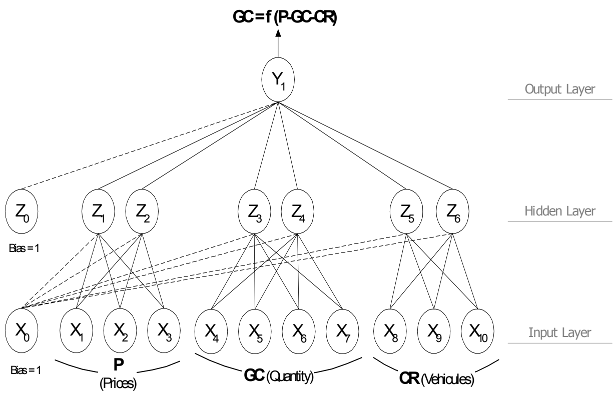


Figure 3: ANN Model II

Table 2: Model II Parameters and Errors

| ANN model II - (P-GC-CR) | |
|---|---|
| Learning Rate ($\alpha$) | = 0.5 |
| Momentum ($\mu$) | = 0.5 |
| Slope ($\sigma$) | = 1 |
| No. Inputs (P-GC-CR) | = 3 – 4 - 3 |
| No. Hidden units | = 2 - 2 - 2 |
| Error Function | = cross entropy |
| *Errors* | |
| MSE | = 129.2828 |
| MAD | = 9.509096 |

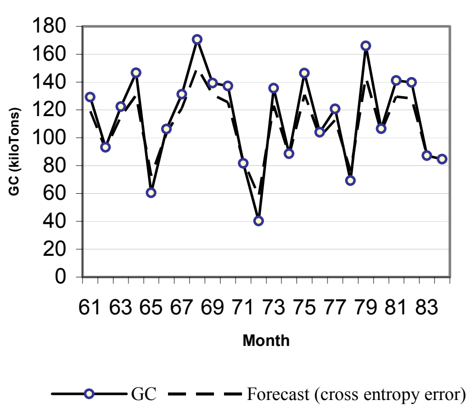


Figure 4: Actual and Forecasted Data using Model II

## Conclusion

This paper estimates gasoline consumption using artificial neural networks models. Two models are built to forecast future gasoline consumption. The first model is a univariate model with four gasoline consumption (GC) input units and is a fully connected model. The second model is a trivariate partially connected model and has past gasoline consumption, gasoline price (P) and car registration (CR) as input units. To build the models, the network is processed into three stages: the training stage, the testing stage and the evaluation stage. The cross entropy error function is used in the training stage to accelerate the bacpropagation algorithm and to reduce stagnation periods. The two models are trained, tested and evaluated using gasoline consumption, price and car registration data from Lebanon during the period extending from January 1993 to December 1999. Forecasting performance measures, namely, the mean square error (MSE) and the mean absolute deviation (MAD) are computed for both models. These measures clearly show that the trivariate model is a much better predictor than the univariate model.

## References

Darbellay, G.A.; and Slama, M. 2000. Forecasting the short-term demand for electricity. Do neural networks stand a better chance? *International Journal of Forecasting* 16: 71-83.

Kayser, H. 2000. Gasoline demand and car choice: estimating gasoline demand using household information. *Energy Economics* 22: 331-348.

Nasr, G.E.; Badr, E.A.; and Younes, M.R. 2001. Neural Networks in Forecasting Electrical Energy Consumption. In *Proceedings of the 14th International FLAIRS Conference*, 489-492. Menlo Park, Calif: AAAI Press.

Nasr, G.E.; Badr, E.A.; and Dibeh, G.L. 2000. Econometric modeling of electricity consumption in postwar Lebanon. *Energy Economics* 22 (6): 627-640.

Puller, S.; and Greening, L.A. 1999. Household adjustment to gasoline price change: an analysis using 9 years of US survey data. *Energy Economics* 21: 37-52.

Saab, S.; Badr, E.A.; and Nasr, G.E. 2001. Univariate Modeling and Forecasting of Energy Consumption: The case of Electricity in Lebanon. *Energy* 26: 1-14.

Sharda, R. 199. Neural networks for the MS/OR analyst: An application bibliography. *Interfaces* 24:116-130.

Statistical Administration. 1993-1999. Statistical Bulletin. *Lebanese Republic*.

Zhang, G.; Patuwo, E.; and Hu, M. 1998. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*.14: 36-62.