Artificial Neural Networks for Time Series Prediction - a novel Approach to Inventory Management using Asymmetric Cost Functions

Sven F. Crone
University of Hamburg, Institute of Information Systems
crone@econon.uni-hamburg.de

Abstract

Artificial neural networks in time series prediction generally minimize a symmetric statistical error, such as the sum of squared errors, to model least squares predictors. However, applications in business elucidate that real forecasting problems contain non-symmetric errors. In inventory management the costs arising from over- versus underprediction are dissimilar for errors of identical magnitude, requiring an ex-post correction of the predictor through safety stocks. To reflect this, an asymmetric cost function is developed and employed as the objective function for neural network training, deriving superior forecasts and a cost efficient stock-level directly from the network output. Some experimental results are computed using a multilayer perceptron trained with different objective functions, evaluating the performance in competition to statistical forecasting methods on a white noise time series.

1. Introduction

Artificial neural networks (ANN) have found increasing consideration in forecasting theory, leading to successful applications in time series and explanatory sales forecasting [2,14,16]. In management, forecasts are a prerequisite for all decisions based upon planning. Therefore the quality of a forecast is evaluated considering its ability to enhance the quality of the resulting management decision. In inventory management, the final evaluation of a forecast must be measured by the monetary costs arising from setting suboptimal inventory levels based on imprecise predictions of future demand [13]. The costs from over- and underprediction are typically not quadratic in form and frequently non-symmetric [8].

Based upon modest research in non-quadratic error functions in ANN theory [2,11,15] and asymmetric costs in prediction theory [1,8,4], a set of asymmetric cost functions was recently proposed as objective functions for neural network training [6]. In this paper, we analyse the efficiency of a linear asymmetric cost function in inventory management decisions, training a multilayer perceptron to find a cost efficient stock-level for a stationary white noise time series directly from the data.

Following a brief introduction to neural network prediction in inventory management, Section 3 assesses statistical error measures and asymmetric cost functions for neural network training. Section 4 gives an experimental evaluation of neural networks trained with asymmetric cost functions, outperforming expert software-systems for time series prediction in Section 4. Conclusions are given in Section 5.

2. Neural network predictions for inventory management decisions

2.1. Forecasting in inventory management

In management, forecast are generated as a prerequisite of business decisions. Through decisions based on sub-optimal forecasts, costs arise to the decision maker in choosing an inefficient alternative.

In inventory management, forecasts of future demands are generated to select an efficient inventory level, balancing inventory holding costs for excessive stocks with costs of lost sales-revenue through insufficient stock [3]. Although the amount of costs will generally increase with the numerical magnitude of the forecast errors, the costs arising from over- and underprediction are frequently neither symmetric nor quadratic [8,5].

A service-level is routinely determined from strategic objectives or according to the actual costs arising from the decision, e.g. aiming to fulfil 98.5% of customer demand to balance this trade-off. Assuming Gaussian distribution, setting the inventory level to the optimum predictor will only fulfil 50% of all customer demand. Therefore, safety-stocks are calculated to reach the service level, using assumptions of the conditional distribution of the ex post forecast errors of the method applied.

For the decision of an inventory level for a single product in a single period of time the classic "newsboy"-problem is applicable. The decision rule for a service level resulting from a given cost of underprediction c_u and overprediction c_o reads

$$p_{y<}(Q^*) = \frac{c_u}{c_u + c_o} \quad , \tag{1}$$

giving the value for a lookup of k in the probability table of the valid distribution. The final stock-level s is calculated using the forecast and adding a safety stock (SS) of k standard deviations of the forecast errors:

$$s = \hat{y}_{t+h} + k\delta_e \quad . \tag{2}$$

Consequently, the precision of the forecasts directly determines the safety stocks kept, the inventory level and the inventory holding costs. Hence, forecasting methods with superior accuracy such as ANN may significantly reduce inventory holding costs [3].

2.2. Neural networks for time series forecasting

Forecasting time series with non-recurrent ANNs is generally based on modelling the network in analogy to a non-linear autoregressive AR(p) model [10,4]. At a point in time t, a one-step ahead forecast $\hat{y}_{t+1}$ is computed using n observations $y_t, y_{t-1}, \ldots, y_{t-n+1}$ from n preceding points in time *t, t-1, t-2, ..., t-n+1*, with n denoting the number of input units of the ANN. This models a time series prediction of the form

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-n+1}) \quad . \tag{3}$$

The architecture of a feed-forward multilayer perceptron (MLP) of arbitrary topology together with the resulting residuals of invalid forecasts denoted as the absolute error (*AE*) is displayed in Fig. 1.

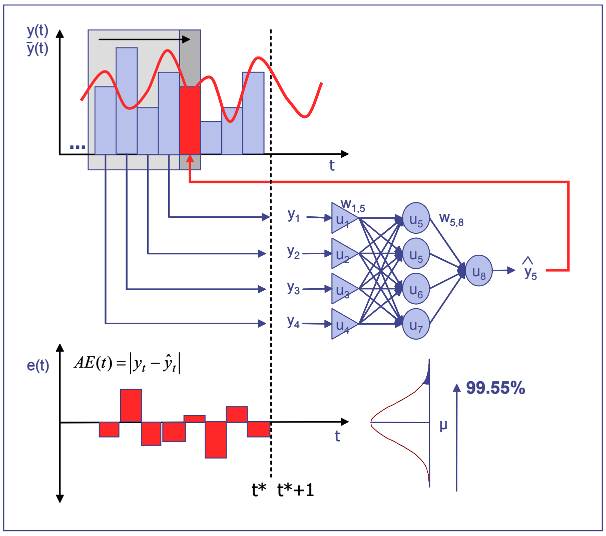

Figure 1. Neural network application to time series forecasting in inventory management, applying a (4-4-1)-MLP, using *n=4* input units for observations in *t, t-1, t-2, t-3*, 4 hidden and 1 output neuron for time period *t+1* [16].

The task of the MLP is to model the underlying generator of the data during training, so that a valid forecast is made when the trained network is subsequently presented with a new value for the input vector [2]. Therefore the objective function used for ANN training determines the resulting system behaviour and performance. The objective functions routinely employed in neural network training differ from the objective function of the underlying inventory management decision in slope, scale and ratio of asymmetry. Following, alternate objective functions are discussed to incorporate the original objective structure in ANN training.

3. Objective functions for neural network training

Supervised online-training of a MLP is the task of adjusting the weights of the links w_{ij} between units j and adjusting their thresholds to minimize the error δ_j between the actual and a desired system behaviour [11]. Gradient descent algorithms traditionally minimize the modified sum of squared errors (*SSE)* as the objective function, ever since the popular description of the backpropagation algorithm by Rumelhart, Hinton and Williams [12].

The *SSE*, as all statistical error measures, produces a value of 0 for an optimal forecast and is symmetric about $e_t = 0$, implying symmetric costs of errors in predicting future demand for inventory levels. The consistent use of the modified *SSE* in time series forecasting with ANN is motivated primarily by analytical simplicity [11] and the similarity to statistical regression problems, modelling the conditional distribution of the output variables [2] similar to most statistical forecasting methods. As neural network theory and applications consistently focus on the symmetric *SSE*-function for training, also modeling least squares predictors, the forecasts also need to be adjusted using safety stocks to attain a desired service-level.

Following, we propose an asymmetric cost function (ACF), modelling the objective function of the costs arising in the original decision problem instead of least squares predictors. These cost are often not only non-quadratic, but also non-symmetric in form. The objective function in ANN training, determining the size of the error in the output-layer, may thus be interpreted as the actual costs arising from an overprediction or an underprediction of the current pattern p in training.

Recently, we introduced a linear ACF to ANN training [6], originally developed by Granger for statistical forecasts in inventory management problems [9]. The *LINLIN* cost function *(LLC)* is linear to the left and right of *0*. The parameters a and b give the slopes of the branches for each cost function and measure the costs of error for each stock keeping unit (SKU) difference between the forecast $\hat{y}_{t+h}$ and the actual value y_{t+h}. The parameter a corresponds to an overprediction and the resulting stock-keeping costs, while *b* relates to the costs of lost sales-revenue for each underpredicted SKU. The *LLC* yields:

$$LLC(y_{t+h}, \hat{y}_{t+h}) = \begin{cases} a\left|y_{t+h} - \hat{y}_{t+h}\right| & \text{for } y_{t+h} < \hat{y}_{t+h} \\ 0 & \text{for } y_{t+h} = \hat{y}_{t+h} \\ b\left|y_{t+h} - \hat{y}_{t+h}\right| & \text{for } y_{t+h} > \hat{y}_{t+h} \end{cases} \quad (4)$$

The shape of one asymmetric *LLC,* as a valid linear approximation of a real cost function in our corresponding inventory management problem, is displayed in Fig. 2.

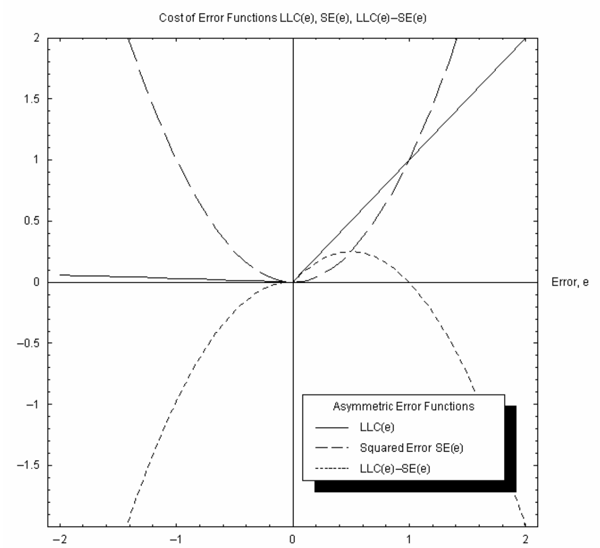

Figure 2. Empirical Asymmetric Cost Function showing cost arsing for over- and under-prediction, using *a*=$0.001 and *b*=$1.00 in comparison to the *SE*.

For $a \neq b$ these cost functions are non-symmetric about *0* and are hence called asymmetric cost functions. The degree of asymmetry is given by the ratio of *a* to *b* [4]. For $a = b = 1$ the LLC equals the statistical absolute error measure *AE*. The linear form of the ACF represents constant marginal costs arising from the business decision. Our model therefore coincides with the analysis of business decisions based on linear marginal costs and profits.

Yang, Chan and King introduce a classification-scheme for objective functions, introducing dynamic non-symmetric margins for support vector regression [17]. Applied to objective functions in ANN training it allows a classification of all symmetric statistical error functions and asymmetric cost functions previously developed. Linear, non-linear and mixed ACFs have been specified [1,4] in literature while variable or dynamic objective functions have not yet been developed for ANN-training, as shown in Table 1.

Table 1. Objective functions for neural network training

Variability	Symmetry of objective function	
	Symmetric	Non-symmetric
Fixed	*SE, AE, ACE.* *(statistical error functions)*	*LINLIN etc.* *(asymmetric cost functions)*
Variable	-	-

Asymmetric transformations of the error function alter the error surface significantly, resulting in changes of slope and creating different local and global minima. Therefore, using gradient descent algorithms, different solutions are found minimizing cost functions instead of symmetric error functions, finding a cost minimum prediction for the inventory management problem. These asymmetric cost functions may be applied in ANN training using a simple generalisation of the error-term of the back-propagation rule and its derivatives, amending only the error calculation for the weight adaptation in the output layer [6], but applying alternative training methods or global search methods to allow network training [11].

4. Simulation experiment of neural networks in inventory management

4.1. Experimental time series and objectives

Following, we conduct an experiment to evaluate the ability of a MLP to evolve a set of weights minimizing an *LLC* asymmetric cost function for a random, stationary time series. To control additional influences in the forecasting experiment we analyse a time series which is white noise of the form:

$$y_t = c + e_t \quad . \quad (5)$$

A white noise model represents a simple random model consisting of an overall level *c* and a random error component e_t which is uncorrelated in time [10].

The time series considered is derived from the popular monthly airline passenger data, introduced by Brown [3]. In [10], the monthly values from 01/1949 unto 12/1956 are detrended and deseasonalised using a X-12-ARIMA Census II decomposition, extracting only the residuals. An analysis of the autocorrelations confirms a stationary white noise model of the form *c*=124900 and no data-pattern in the residuals e_t. Considering the structure of a stationary white-noise model, lacking any systematic pattern in the residuals of e_t, it should prohibit one ANN to extract any underlying linear or nonlinear generator of the data and thus outperform competing ANNs or linear methods, ensuring an unbiased comparison of methods regardless of individual model performance. Experiencing random fluctuations, the ANN as other methods should predict the level c as the optimum predictor without overfitting to the training data.

In order to specify the underlying costs arising from the decision process we require a suitable objective function. We construct the airline decision problem as an inventory model without backordering. An airline carrier needs to allocate planes of different sizes to match passenger demand of flights. Overprediction of air flight passengers leads to inventory holding costs a for the seat while underprediction results in costs b through lost sales-revenue per seat, assuming $b>a$ and disregarding fixed costs of the decision. For our experiment, we assume a highly asymmetric cost relationship of (a=\$0.001; b=\$1.00). We generate business forecasts based upon ordinary least-squares predictors added to the safety buffers necessary to achieve the desired service level. Additionally we use asymmetric cost predictors to decide the cost efficient amount of passenger seats provided for each month, assessing the ex-post performance.

For forecasting methods using ordinary least squares predictors, such as ANN trained on *SE* or ARIMA-models, the service level is calculated, using

$$p_{y<}\left(Q^*\right)=\frac{c_u}{c_u+c_o}=\frac{1.00}{1.001}=0.9990 \quad , \tag{6}$$

to derive an optimum service level of 99.90% for our inventory problem, therefore requiring k=3.09 standard deviations assuming a Gaussian distribution of residuals.

4.2. Experimental design of the forecasting methods

A sample of $n=96$ observations is split into three consecutive datasets, using 72 observations for the training-, 12 for the validation- and 12 for the test-set, resulting in 59, 12 and 12 predictable patterns in each set. All data was scaled from a range of 90000 to 110000 to the interval [-1;1]. We consider a fully connected MLP with a topology of 13 input, 12 hidden, 1 output node without shortcuts connections to exploit all feasible yearly time-lags of a monthly series. All units use a summation as an input-function, the tangens hyperbolicus (tan h) as a semilinear activation-function and the identity function as an output-function. Additionally, 1 bias unit models the thresholds for all units in the hidden and output layer.

Two sets of networks were trained. Set ANN_{SE} was trained on minimizing the *SE*, set ANN_{LLC} was trained minimizing an asymmetric cost function with the parameters *LLC* (a=\$0.001;b=\$1.00) for (7). Each MLP was initialised and trained for five times to account for [-0.3;0.3] randomised starting weights.

Training consisted of a maximum of 10000 epochs with a validation after every epoch, applying early stopping if the validation error did not decrease for 1000 epochs. After training, the results for the best network, chosen on its performance on the validation set, and the average of five networks are computed for all data-sets. Only the test-set-data is used to measure generalisation, applying a simple hold out method for cross-validation.

As benchmarks, the Naïve1 method using last periods sales as a forecast, and results using the software Forecast Pro and Autobox were computed. Each software selects and parameterises appropriate models of exponential smoothing or ARIMA intervention models based upon statistical testing and expert knowledge. All ANNs were simulated using NeuralWorks II Professional and distinct error function tables to bias the calculated standard error.

For the predictions by ANN_{SE} and the expert software systems the final inventory level was calculated, using

$$s=\hat{y}_{t+h}+3.09\delta_e \quad . \tag{7}$$

Resulting in an ex-post correction of the ordinary least squares predictor. The ANN_{LLC} was trained to forecast the inventory level directly through the ACF.

4.3. Experimental results on asymmetric costs

Table 2 displays the results using mean error measures computed on each data-set to allow comparison between data-sets of varying length. The results are given in the

Table 2. Results on Forecasting Methods and ANNs trained on linear Asymmetric Costs and Squared Error Measures

	Error Measures						Service Level			Rank on test-set	
	MSE(e)			*MLLC(e)*			*Beta*			*MSE*	*LLC*
Naïve 1(random walk)	18.4	10.26	0.82	1.57	1.53	0.320	98.75%	98.78%	99.74%	4	11
Autobox ARIMA with pulse	1.80	2.95	0.58	0.53	0.58	0.250	99.58%	99.54%	99.80%	2	9
ForecastProXE ARIMA (0,0,0)	10.03	4.39	0.50	1.02	0.92	0.230	99.19%	99.27%	99.82%	**1**	8
ANN_{SE} trained on *SE*, best network (No.45)	10.80	4.48	0.68	0.96	0.77	0.160	99.23%	99.38%	99.88%	3	7
ANN_{SE} trained on *SE,* average of 5 nets	10.91	4.90	0.82	1.07	0.91	0.270	99.14%	99.28%	99.78%	4	10
Autobox ARIMA + safety stock k=3	18.5	21.6	18.47	0.05	0.004	0.004	99.96%	100.0%	100.0%	8	3
Forecast Pro ARIMA + safety stock k=3	123.6	81.7	81.3	0.05	0.009	0.009	99.96%	100.0%	100.0%	11	6
ANN_{SE} trained on *SE*, best net + safety st. k=3	39.9	30.3	31.6	0.19	0.01	0.006	99.85%	100.0%	100.0%	9	4
ANN_{SE} trained on *SE*, av. 5 net +safety st. k=3	42.0	32.4	33.8	0.18	0.01	0.006	99.86%	100.0%	100.0%	10	4
ANN_{LLC} trained on *LLC1*, best net (No.48)	18.30	9.05	8.47	0.40	0.17	0.003	99.68%	99.86%	100.0%	7	**1**
ANN_{LLC} trained on *LLC1*, av. of 5 networks	15.23	8.87	6.36	0.45	0.17	0.003	99.65%	99.81%	100.0%	6	**1**

form (training-set / validation-set / test-set) to allow interpretation. The descriptive performance measure of the β-service-level gives the amount of suppressed sales per dataset in relation to all demand. An asymmetric ex-post performance measure is calculated, denoting the ex post mean *LINLIN* costs (*MLLC*) resulting from a given forecast method. All methods are evaluated on and ranked by their performance for the mean *SE (MSE)* and the *MLLC*.

Various results may be drawn from the experiment. As expected, the best ANN_{SE} trained on the standard *SE* gives forecasts close to the white noise level *c* with little deviations due to limited overfitting on the residuals, as shown in Fig.3. The forecasts given by all ANNs robustly centre around *c* as displayed in the comparative performance of the average of all 5 networks trained on minimizing the *SE* in table 2..

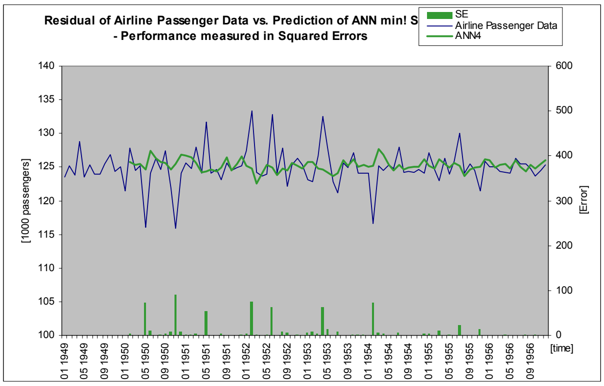

Figure 3. ANN trained on minimizing the symmetric *SE* to forecast monthly airline passengers, showing the white noise time series of ticket sales, the ANN forecast and the ex-post forecast error measured by the *SE(e)*.

In comparison, the ARIMA-(0,0,0)-model selected by the expert system software Forecast Pro predicts the constant *c* precisely, as shown in figure 4.

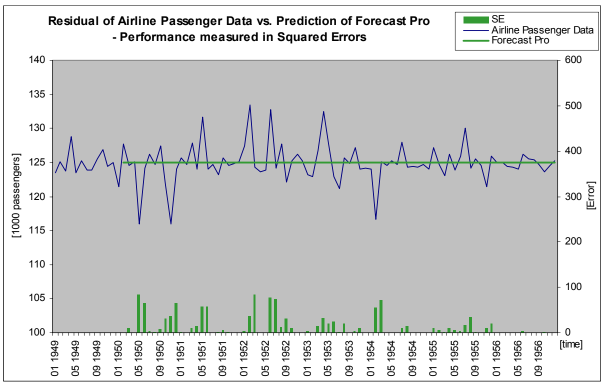

Figure 4. Forcast Pro prediction of ARIMA (0,0,0)-model, the original white noise time series of ticket sales and the ex-post forecast error measured by the *SE(e)*.

The ARIMA intervention model parameterised by the software Autobox fitted significant deviations of *c* as pulses to reduce the ex-post forecast error, thereby reducing the standard deviation of the error in figure 5.

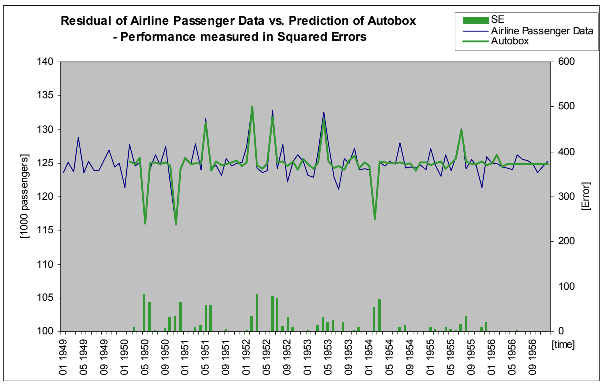

Figure 5. Autobox prediction of ARIMA intervention model, the original white noise time series of ticket sales and the ex-post forecast error measured by the *SE(e)*.

However, the ARIMA (0,0,0)-model outperforms all other methods including the ANN and the random walk model regarding the performance of the *SE* on the test-set.

The impact of intervention modelling in comparison to standard ARIMA-modelling becomes evident in the proposed inventory levels of the competing models. Although all methods avoid costly stockouts achieving a service level of 100.0%, the increased standard deviation of the ARIMA forecasting residuals including all pulses leads to significantly higher safety stocks and an increased inventory level on the test-set.

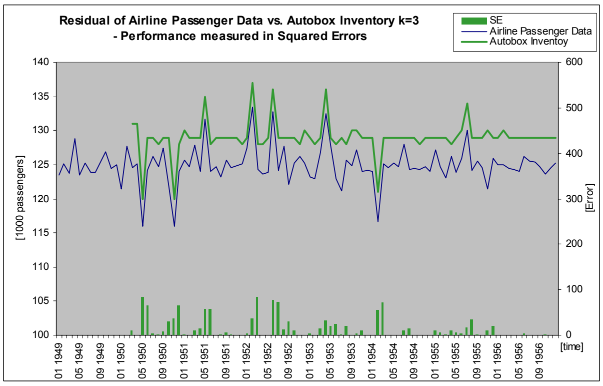

Figure 6. Autobox forecast of ARIMA intervention models including safety stocks of k=3 standard deviations and the ex-post forecast error measured by the *SE(e)*.

Consequently, the costs arising from the ARIMA inventory levels exceed that of the intervention models, as displayed in figures 6 and 7 as well as the error measures presented in table 2.

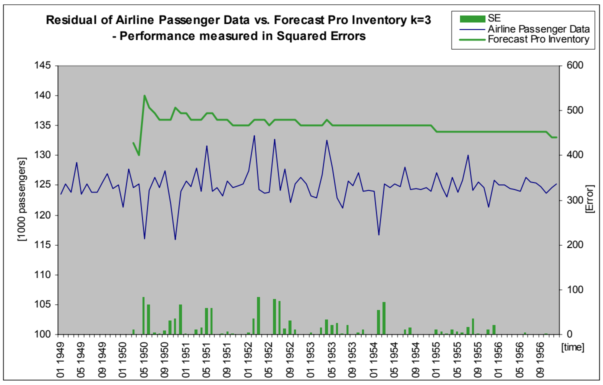

Figure 7. Forecast Pro forecast of ARIMA (0,0,0) model including safety stocks of k=3 standard deviations and the ex-post forecast error measured by the *SE(e)*.

The best ANN_{LLC} trained with the asymmetric *LINLIN* cost function *LLC,* shown in fig. 8, gives an overall superior forecast regarding the business objective, achieving the lowest mean costs on the test-data with 0.003. It exceeds all inventory methods, and clearly outperforms forecasts of ANN trained with the *SE* criteria and added safety stocks, as well as the ARIMA and ARIMA-intervention models with safety stocks selected by the software expert system.

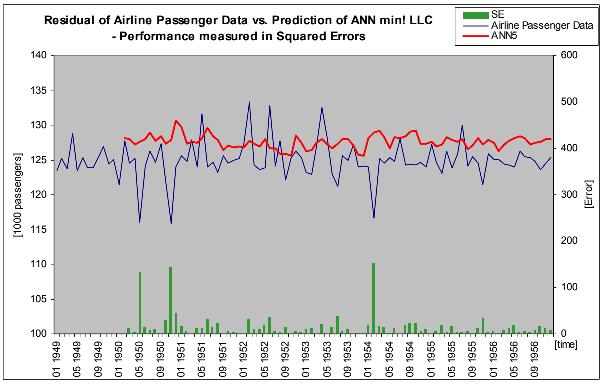

Figure 8. ANN trained on minimizing the asymmetric cost function *LLC1* to forecast monthly airline passengers, showing the actual ticket sales, the ANN forecast and the ex-post forecast error measured by the *SE(e)*.

Analysing the behaviour of the forecast based upon the asymmetry of the costs function, the neural network ANN_{LLC} raises its predictions in comparison to the ANN_{SE} trained on squared errors to achieve a cost efficient forecast of the optimum inventory level, accounting for higher costs of underpredicition versus overprediction and therefore avoiding costly stock-outs. This is also evident in the lack of stock-outs represented by the increased β -service-level of 100.00%.

Consequently, the neural network no longer predicts the expected mean *c* of the white noise function but instead produces a biased optimum predictor, as intended by Grangers original work through ex post correction of the original predictor [8]. This may be interpreted as finding a point on the conditional distribution of the optimal predictor depending on the standard-deviation.

For an inventory management problem, the network finds a cost efficient inventory level without the separate calculation of safety stocks directly from the cost relationship. This reduces the complexity of the overall management process of stock control, successfully calculating a cost efficient inventory level directly through a forecasting method using only a cost function and the data.

5. Conclusion

We have examined symmetric and asymmetric error functions as performance measures for neural network training. The restriction on using squared error measures in neural network training may be motivated by analytical simplicity, but it leads to biased results regarding the final performance of forecasting methods. Asymmetric cost functions can capture the actual decision problem directly and allow a robust minimization of relevant costs using standard multilayer perceptrons and training methods, finding optimum inventory levels. Our approach to train neural networks with asymmetric cost functions has a number of advantages. Minimizing an asymmetric cost function allows the neural network not only to forecast, but instead to reach optimal business decisions directly, taking the model building process closer towards business reality. As demonstrated, considerations of finding optimal service levels in inventory management are incorporated within the ANN training process, leading directly to the forecast of a cost minimum stock level without further computations.

However, the limitations and promises of using asymmetric cost functions with neural networks require systematic analysis. Future research may incorporate the modelling of dynamic carry-over-, spill-over-, threshold- and saturation-effects for exact asymmetric cost functions where applicable. In particular, verification on multiple time series, other network topologies and architectures is required, in order to evaluate current research results.

References

[1] G. Arminger, N. Götz, Asymmetric Loss Functions for Evaluating the Quality of Forecasts in Time Series for Goods Management Systems - Technical Report 22/1999, Dortmund, 1999

[2] J. S. Armstrong, Strategic Planning and Forecasting Fundamentals, in K. Albert (eds.), The Strategic Management Handbook, McGraw-Hill, New York, 1983, 2, 2-1 - 2-32,

[3] J. S. Armstrong, Error Measures for Generalizing About Forecasting Methods - Empirical Comparisons, International Journal of Forecasting, 1992, 8, 69-80,

[4] J. S. Armstrong, Evaluating Forecasting Methods, in J. S. Armstrong (eds.), Principles of Forecasting - A Handbook for Researchers and Practitioners, Kluwer Academic, Boston, 2001, 443-472,

[5] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995

[6] G. Box, G. Jenkins, Time Series Analysis - Forecasting and Control, Holden-Day, San Francisco, 1970

[7] R. G. Brown, Statistical Forecasting for Inventory Control, McGraw-Hill, New York, 1959

[8] P. F. Christoffersen, F. X. Diebold, Further Results on Forecasting and Model Selection under Asymmetric Loss, *Journal of applied econometrics*, 11, 1996, 5, 561-572,

[9] P. F. Christoffersen, F. X. Diebold, Optimal Prediction under General Asymmetric Loss, *Econometric Theory*, 13, 1997, 808-817, in M. P. Clements, D. F. Hendry (eds.), *Forecasting economic time series*, Cambridge University Press, Cambridge, 1998

[10] S. F. Crone, Training Artificial Neural Networks for Time Series Prediction using Asymmetric Cost Functions, *in L. Wang, C. Rajapakse, K. Fukushima, S.-Y. Lee, X. Yao (eds.), Computational Intelligence for the E-Age - Proceedings of the 9th International Conference on Neural Information Processing ICONIP'02, Nov. 18-22 2002, Singapore, 2002*, III, III-1301-III-1308,

[11] G. Dorffner, Neural Networks for Time Series Processing, *Neural Network World*, 4, 1996, 6, 447-468,

[12] C.W.J. Granger, Prediction with a Generalized Cost of Error Function, *Operational Research Society (eds.), Operational Research Quarterly*, 20, 1969, S. 199-207,

[13] C.W.J. Granger, *Forecasting in Business and Economics*, Academic Press, New York, 1980

[14] S. Makridakis, S. C. Wheelwright, R. J. Hyndman, *Forecasting - Methods and Applications*, Wiley, New York, 1998

[15] Parsian, N. S. Farsipour, Estimation of the mean of the selected population under asymmetric loss function, *O. Anderson, et al. (eds.), Metrika, International Journal for theoretical and applied statistics*, 50, 1999, 2, 89 - 107,

[16] R. D. Reed, R. J. Marks II, *Neural Smithing - Supervised Learning in Feedforward Artificial Neural Networks*, MIT Press, Camebridge, MA, 1998

[17] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors - (reprint), *in J. A. Anderson, E. Rosenfeld (eds.), Neurocomputing - Foundations of Research, MIT Press, Camebridge, MA, 1988*, 1, 696-699,

[18] J. Schwarze, Statistische Kengrößen zur Ex-Post-Beurteilung von Prognosen - (Prognosefehlermaße), *in J. Schwarze (eds.), Angewandte Prognoseverfahren, Neue Wirtschafts-Briefe, Herne, 1980*, 317-344,