Neural Processing Letters (2007) 25:157–169
DOI 10.1007/s11063-007-9035-z

iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data☆

ANTONIO MUÑOZ SAN ROQUE[1,⋆], CARLOS MATÉ[1], JAVIER ARROYO[2] and ÁNGEL SARABIA[1]
[1]*Instituto de Investigación Tecnológica (IIT), Escuela Técnica Superior de Ingeniería (ICAI), Universidad Pontificia Comillas, Alberto Aguilera 25, 28015 Madrid, Spain. E-mails: amunoz@upcomillas.es, cmate@upcomillas.es, asarabia@upcomillas.es*
[2]*Departamento de Ingenieńa del Software e Inteligencie Artificial, Universidad Complutense, Profesor García-Santesmases s/n, 28040 Madrid, Spain. E-mail: javier.arroyo@tdi.ucm.es*

Abstract. Interval-valued data offer a valuable way of representing the available information in complex problems where uncertainty, inaccuracy or variability must be taken into account. In addition, the combination of Interval Analysis with soft-computing methods, such as neural networks, have shown their potential to satisfy the requirements of the decision support systems when tackling complex situations. This paper proposes and analyzes a new model of Multilayer Perceptron based on interval arithmetic that facilitates handling input and output interval data, but where weights and biases are single-valued and not interval-valued. Two applications are considered. The first one shows an interval-valued function approximation model and the second one evaluates the prediction intervals of crisp models fed with interval-valued input data. The approximation capabilities of the proposed model are illustrated by means of its application to the forecasting of daily electricity price intervals. Finally, further research issues are discussed.

Key words. feed-forward neural network, function approximation, interval analysis, interval data, interval neural networks, symbolic data analysis, time series forecasting

Abbreviations: iMLP – interval Multilayer Perceptron; INN – Interval Neural Network; MAPE – Mean Absolute Percentage Error; MLP – Multilayer Perceptron

1. Introduction

1.1. ARTIFICIAL NEURAL NETWORKS, INTERVALS AND DECISION SCIENCES

Artificial neural networks have found increasing consideration in management science, leading to successful applications in various domains, including business and operations research (see e.g. [1–3]), forecasting (see e.g. [4–6]), and data mining (see e.g. [7,8]).

In decision support systems, the Multilayer Perceptron (MLP) is one of the most popular neural network models (see e.g. [2,9]). This is due to the fact that its architecture is very clear and the algorithm is parsimonious. Successful application of

☆ Research funded by Universidad Pontificia Comillas.
⋆ Author for correspondence.

MLP to complex problems such as pattern classification, forecasting, regression, and nonlinear systems modelling is due to the mathematical properties of feedforward neural networks and to the computational intensive methodology [10].

On the other hand, decision support systems are usually applied to situations where inaccuracy, uncertainty or variability must be taken into account to faithfully represent the real world. Unfortunately, classical data sets, where a set of items is described by variables that map each item to a single (crisp) value, cannot reflect these nuances. In these cases, other kinds of variables, such as interval variables, are required. Intervals allow the representation of more general situations such as the inaccuracy of the measurement instrument, the bounds of the set of possible values of the item, the range of variation of a variable through time or through a set of sub-items, and so on. Two different domains deal with interval data: Symbolic Data Analysis and Interval Analysis. They will be briefly introduced below.

1.2. SYMBOLIC DATA ANALYSIS

As Ward et al. point up [11], nowadays, data sets increasingly suffer from the problem of scale, either in terms of the number of variables or the number of records. It is often desirable to reduce the size of the data maintaining their essential features as much as possible. This reduction can be performed by manually pruning the data set basing on some domain knowledge, or via sampling, or by dimensionality reduction methods such as principal component analysis and multidimensional scaling, or by aggregation/summarization methods, such as clustering or partitioning.

Symbolic Data Analysis, a new paradigm related to Statistics, Data Mining and Computer Sciences, addresses this problem. It offers a comprehensive approach that consists of summarization of the data set by means of symbolic variables (e.g., interval variables), resulting in a smaller and more manageable data set which preserves the essential information, and its subsequent analysis by means of symbolic methods. Symbolic methods include descriptive statistics, principal component analysis, clustering, and discrimination techniques. Bock and Diday [12] present an excellent review of the field along with illustrative examples mainly from official statistics. However, Billard and Diday [13] draw attention to the enormous need of new methodologies for symbolic data.

In symbolic data, individuals are described by symbolic variables such as lists of categorical or quantitative values with or without associated weights, intervals, histograms and frequency distributions. These variables, in contrast to the classical approach, where only one single number or category is allowed, own a great potential in order to characterize complex real-life situations (e.g., time-varying patterns, class descriptions, uncertain or inaccurate data, and so on) and to summarize massive data sets in an efficient way. As mentioned on above, in this paper, we will focus on intervals.

1.3. INTERVAL ANALYSIS

Interval analysis is a field introduced by R. E. Moore [14] which assumes that, in the real world, observations and estimations are usually incomplete or uncertain and, consequently, they do not represent the real data exactly. According to this field, if precision is needed, data must be represented as intervals enclosing the real quantities. In addition, errors in numeric computations are usually enlarged due to rounding or truncating processes. Interval analysis provides methods to control errors in numeric computations dealing with intervals. Since the 1960s, it has been an active focus on research (see [15] for a review of this area). Fundamentals used in this paper are described below.

Intervals will be denoted by an uppercase letter, e.g. A, while real numbers will be denoted by a lowercase letters, e.g. a. In addition, vectors will be denoted by boldfaced letters, e.g. $\mathbf{A}, \mathbf{a}$, and $\mathbb{IR}$ will represent the set of all intervals in the real line. An interval can be represented by its lower and upper limits as $A=[a^L, a^U]$, or, equivalently, by its midpoint and radius as $[A]=\langle a^C, a^R\rangle$, where $a^C=(a^L+a^U)/2$ and $a^R=(a^U-a^L)/2$.

The basis of interval computations is interval arithmetic. A preliminary form of interval arithmetic appears in [16], but modern interval arithmetic was proposed by Moore. Let A and B be two intervals and $\square$ be an arithmetic operator, then $A\square B$ is the smallest interval which contains $a\square b$ $\forall a\in A$ and $\forall b\in B$. More precisely, the addition, the subtraction and the multiplication are defined by:

$$A+B=[a^L+b^L, a^U+b^U], \tag{1}$$

$$A-B=[a^L-b^U, a^U-b^L], \tag{2}$$

and

$$A\cdot B=[\min\{a^L\cdot b^L, a^L\cdot b^U, a^U\cdot b^L, a^U\cdot b^U\}, \tag{3a}$$

$$\max\{a^L\cdot b^L, a^L\cdot b^U, a^U\cdot b^L, a^U\cdot b^U\}], \tag{3b}$$

respectively.

Finally, if f is an increasing function, then the interval output is given by

$$f(A)=[f(a^L), f(a^U)]. \tag{4}$$

The sigmoid and the hyperbolic tangent functions, which are standard activation functions in MLP, are strictly increasing functions (i.e. they satisfy the monotonic condition).

1.4. INTERVALS AND MULTILAYER PERCEPTRONS

According to [17], a neural network is called interval neural network (INN) if at least one of its input, output or weight sets are interval valued; in this subsection, we will review the previous INN models, specifying which sets are represented as intervals.

Ishibuchi et al. [18] propose an INN where weights, biases and output are interval valued, but input data are crisp (i.e. single-valued). They also show an application of their model to fuzzy regression analysis. Baker and Patil [19] prove the universal approximation theorem for this kind of INN. Due to the complexity of the learning algorithms, Ishibuchichi et al. [18] also propose a simplified architecture of their INN where the weights and biases to hidden units are restricted to single values, which seems to fit well to the training data in the example shown.

Simoff [20] proposes an INN where inputs, weights, biases and output are interval valued. Simoff analyzes the properties of the model and comments on the explosion of interval uncertainty that results from repeated operations on interval values. However, he does not propose a learning algorithm for this model.

Beheshti et al. [17] propose a three-layer perceptron where inputs, weights, biases and outputs are intervals, and show how to obtain optimal weights and biases for a given training data set by means of interval computational algorithms.

Drago and Ridella [21] propose a one-layer perceptron based on interval arithmetic with interval weights, where input data are classical (crisp) and output data are categorical. Their perceptron allows the detection of uncertainty regions in classification tasks. Rossi and Conan-Guez [22] do not propose a new kind of INN, instead they propose several approaches that allow intervals being the inputs and the outputs of a classical MLP. For example, they propose training a MLP with the lower interval bounds and the upper interval bounds. In another approach, Rossi and Conan-Guez [22] consider that within intervals, values are uniformly distributed, and they propose training a classical MLP with a set of data sampled from each interval in the original dataset. Both approaches serve well to work with artificial data but do not enable modelling complex real life situations.

Patiño-Escarcina et al. [23] propose a one layer perceptron for classification tasks, where inputs, weights and biases are represented by intervals. The activation function is a binary function for interval data and the output is represented in binary form.

In this paper, we propose and analyze an INN that will be called interval Multi-Layer Perceptron (iMLP) for dealing with interval-valued inputs and outputs, but where weights and biases are not interval-valued but single-valued. The iMLP will allow us to deal with uncertainty, inaccuracy or variability in datasets (that will be represented by intervals), but using a kind of architecture very similar to the classical MLP, which simplifies its learning procedure, retaining its ability to approximate non-linear interval functions.

2. Structure of the iMLP

The proposed Interval Multilayer Perceptron is basically a MLP (see [9,24]) operating on interval-valued input and output data, so both models share the same

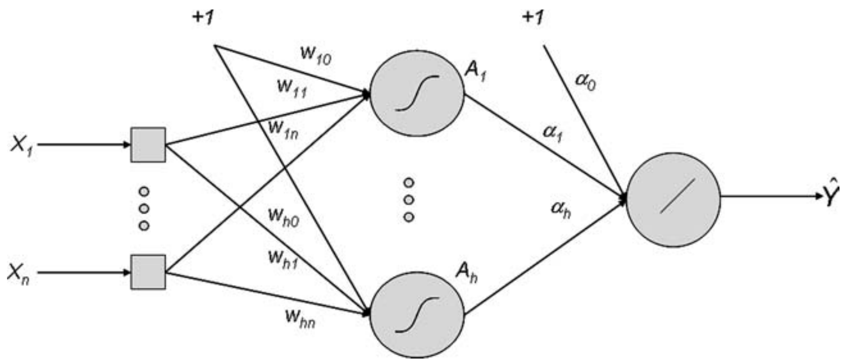

Figure 1. Structure of the iMLP.

structure but use different transfer functions. As in the case of the MLP, an iMLP with n inputs and m outputs is comprised of: an input layer with n input buffer units, one or more hidden layers with a non-fixed number of nonlinear hidden units and one output layer with m linear or nonlinear output units. Without loss of generality, we will restrict ourselves to one hidden layer with h hidden units and one output ($m = 1$). The iMLP described below is comprised of two layers of adaptive weights (a hidden layer and an output layer). This architecture is shown in Figure 1.

Considering n interval-valued inputs $X_i = \langle x_i^C, x_i^R \rangle = [x_i^C - x_i^R, x_i^C + x_i^R]$, with $i = 1, \dots, n$, the output of the jth hidden unit is obtained by forming a weighted linear combination of the n interval inputs and the bias. As the weights of the proposed structure are crisp and not intervals, this linear combination results in a new interval given by:

$$S_j = w_{j0} + \sum_{i=1}^{n} w_{ji} X_i = \left\langle w_{j0} + \sum_{i=1}^{n} w_{ji} x_i^C, \sum_{i=1}^{n} |w_{ji}| x_i^R \right\rangle. \tag{5}$$

The activation of the hidden unit j is then obtained by transforming the interval S_j using a nonlinear activation function $g(\cdot)$:

$$A_j = g(S_j). \tag{6}$$

In this study, the tanh function is used as an activation function in the hidden layer. As the activation function is monotonic, this transformation yields to a new interval which can be calculated as:

$$A_j = \tanh(S_j) = [\tanh(s_j^C - s_j^R), \tanh(s_j^C + s_j^R)] \tag{7a}$$

$$= \left\langle \frac{\tanh(s_j^C - s_j^R) + \tanh(s_j^C + s_j^R)}{2}, \right. \tag{7b}$$

$$\left. \frac{\tanh(s_j^C + s_j^R) - \tanh(s_j^C - s_j^R)}{2} \right\rangle. \tag{7c}$$

Finally, the output of the network, $\hat{Y}$, is obtained by transforming the activations of the hidden units using a second layer of processing units. In the case of a single output and a linear activation function with crisp weights, the estimated output interval is obtained as a linear combination of the activations of the hidden layer and the bias:

$$\hat{Y} = \sum_{j=1}^{h} \alpha_j A_j + \alpha_0 = \left\langle \sum_{j=1}^{h} \alpha_j a_j^C + \alpha_0, \sum_{j=1}^{h} |\alpha_j| a_j^R \right\rangle. \tag{8}$$

The resulting model can be used in two ways:

1. As an interval-valued function approximation model, whose crisp weights can be adjusted with a supervised learning procedure by minimizing an error function of the form:

$$E = \frac{1}{p} \sum_{t=1}^{p} d(Y(t), \hat{Y}(t)) + \lambda \Phi(\hat{f}), \tag{9}$$

   where $d(Y(t), \hat{Y}(t))$ is a measure of the discrepancy between the desired and the estimated output intervals for the tth training sample, denoted by $Y(t)$ and $\hat{Y}(t)$, respectively; and $\lambda \Phi(\hat{f})$ is a regularization term [25] of the estimated function $\hat{f}(\mathbf{X})$: $\mathbf{X} \to Y$.
2. As an instrument to evaluate the prediction interval of a pre-adjusted crisp MLP model subject to uncertainty on its input variables, without simulating input values. In this context, the output range is obtained in a straightforward manner by evaluating an iMLP with the same structure and weights of the pre-adjusted crisp MLP model, but using interval-valued inputs for characterizing the input uncertainty.

3. Interval-valued Function Approximation with iMLP

Let $\{(\mathbf{X}(1), Y(1)), (\mathbf{X}(2), Y(2)), \ldots, (\mathbf{X}(p), Y(p))\}$ be p training samples which consists of pairs of input–output intervals, where $\mathbf{X}(i)$ is an interval input vector in $\mathbb{IR}^n$, and $Y(i)$ is its corresponding interval output value in $\mathbb{IR}$. We consider that these pairs are generated according to an unknown continuous function that maps a crisp input vector $\mathbf{x} \in \mathbb{R}^n$ to a crisp output value $y \in \mathbb{R}$, that is,

$$f(\mathbf{x}) : \mathbf{x} \to y. \tag{10}$$

This function f is subject to output noise ε, so that in the absence of noise $\forall \mathbf{x} \in \mathbf{X}(i)$, then $(f(\mathbf{x}) + \varepsilon) \in Y(i)$. The function approximation task [25] consists of finding an estimate, say $\hat{f}(\mathbf{x})$, of the unknown function $f(\mathbf{x})$ which is supposed to be smooth.

3.1. COST FUNCTION

As mentioned earlier, this problem can be solved by selecting an approximating function $\hat{f}(\mathbf{x}, \mathbf{w})$ which depends continuously on $\mathbf{x}$ and $\mathbf{w}$, and optimizing the parameters $\mathbf{w}$ by minimizing an error function of the form given in Equation (9).

In this paper, a weighted Euclidean distance function for a pair of intervals A and B has been used. It is defined as:

$$d(A, B) = \beta(a^C - b^C)^2 + (1-\beta)(a^R - b^R)^2. \tag{11}$$

The parameter $\beta \in [0, 1]$ facilitates giving more importance to the prediction of the output centres or to the prediction of the radii. For $\beta = 1$ learning concentrates on the prediction of the output interval centre and no importance is given to the prediction of its radius. For $\beta = 0.5$ both predictions (centres and radii) have the same weights in the objective function.

3.2. LEARNING ALGORITHM

For the minimization of the cost function, a low-memory Quasi Newton method [26] with random initial weights has been applied. Second order methods require the calculation of the gradient of the cost function with respect to the adaptive weights ($w's$ and $\alpha's$). These derivatives can be calculated in an effective way by applying a backpropagation procedure, similar to the BP algorithm proposed in [24] for the standard MLP.

The derivatives of the proposed cost function (the regularization term is not considered for simplicity) with respect to the output layer weights are given by:

$$\frac{\partial E}{\partial \alpha_j} = \frac{2}{p} \sum_{t=1}^{p} \left[\left(\beta(\hat{y}(t)^C - y(t)^C) \frac{\partial \hat{y}(t)^C}{\partial \alpha_j} \right. \right. \tag{12a}$$

$$\left. \left. + (1-\beta)(\hat{y}(t)^R - y(t)^R) \frac{\partial \hat{y}(t)^R}{\partial \alpha_j} \right) \right], \tag{12b}$$

where

$$\frac{\partial \hat{y}(t)^C}{\partial \alpha_j} = \begin{cases} 1, & \text{for } j = 0; \\ a_j(t)^C, & \text{for } j > 0. \end{cases} \tag{13}$$

$$\frac{\partial \hat{y}(t)^R}{\partial \alpha_j} = \begin{cases} 0, & \text{for } j = 0; \\ \operatorname{sgn}(\alpha_j) a_j(t)^R, & \text{for } j > 0. \end{cases} \tag{14}$$

The derivatives of the cost function with respect to the hidden layer weights can be expressed as:

$$\frac{\partial E}{\partial w_{ji}} = \frac{2}{p}\sum_{t=1}^{p}\Bigg[\Bigg(\beta(\hat{y}(t)^C - y(t)^C)\frac{\partial \hat{y}(t)^C}{\partial w_{ji}} \tag{15a}$$

$$+(1-\beta)(\hat{y}(t)^R - y(t)^R)\frac{\partial \hat{y}(t)^R}{\partial w_{ji}}\Bigg)\Bigg] \tag{15b}$$

$$= \frac{2}{p}\sum_{t=1}^{p}\Bigg[\Bigg(\beta(\hat{y}(t)^C - y(t)^C)\frac{\partial \hat{y}(t)^C}{\partial a_j(t)^C}\frac{\partial a_j(t)^C}{\partial w_{ji}} \tag{15c}$$

$$+(1-\beta)(\hat{y}(t)^R - y(t)^R)\frac{\partial \hat{y}(t)^R}{\partial a_j(t)^R}\frac{\partial a_j(t)^R}{\partial w_{ji}}\Bigg)\Bigg], \tag{15d}$$

where

$$\frac{\partial \hat{y}(t)^C}{\partial a_j(t)^C} = \alpha_j \tag{16}$$

$$\frac{\partial a_j(t)^C}{\partial w_{ji}} = \frac{\tanh'(s_j(t)^C + s_j(t)^R)(x_i(t)^C + \mathrm{sgn}(w_{ji})x_i(t)^R)}{2} \tag{17a}$$

$$+\frac{\tanh'(s_j(t)^C - s_j(t)^R)(x_i(t)^C - \mathrm{sgn}(w_{ji})x_i(t)^R)}{2}, \tag{17b}$$

and

$$\frac{\partial \hat{y}(t)^R}{\partial a_j(t)^R} = |\alpha_j| \tag{18}$$

$$\frac{\partial a_j(t)^R}{\partial w_{ji}} = \frac{\tanh'(s_j(t)^C + s_j(t)^R)(x_i(t)^C + \mathrm{sgn}(w_{ji})x_i(t)^R)}{2} \tag{19a}$$

$$-\frac{\tanh'(s_j(t)^C - s_j(t)^R)(x_i(t)^C - \mathrm{sgn}(w_{ji})x_i(t)^R)}{2}. \tag{19b}$$

Standard cross-validation techniques [9] are applied in order to prevent over-fitting.

4. Numerical Example

One of the main sources of interval-valued data is the summarization of high frequency sampled data. For example, intradaily stock prices are often summarized and analyzed in terms of daily ranges, giving rise to intervals.

The Spanish electricity market (see www.omel.es for more details) is organized as a sequence of sessions where producers, distributors and resellers, qualified

Table 1. Explanatory variables applied to the model.

Input	Description
$E(t)$	Total amount of traded energy
$N(t)$	Nuclear generation
$C(t)$	Coal generation
$F(t)$	Fuel generation
$G(t)$	Gas generation
$H(t)$	Hydro generation
$P(t-1)$	Electricity price for the previous day
$P(t-7)$	Electricity price one week before

consumers and external agents perform electricity transactions. The main part of this energy is traded on the daily market. The purpose of the daily market, as an integral part of the electricity power production market, is to handle electricity transactions for the following day by the presentation of electricity sale and purchase bids by market participants. The clearance of this market establishes an hourly clearing price which is paid to generators.

In this illustrative example, the daily electricity price interval is modelled as a function of the total amount of traded energy and the production covered by different technologies: nuclear, coal, fuel, gas and hydro (see Table 1 for a summary of the variables considered). Regular and seasonal autoregressive components of the time series are modelled by including delayed electricity prices (orders 1 and 7) as input variables. All these inputs are also treated as daily intervals.

An iMLP with 10 neurons in the hidden layer has been trained with 14 months of data as training set (Jul-1-2003 to Aug-31-2004) and validated with 3 months (Sep-1-2004 to Nov-30-2004). The same weight, $\beta = 0.5$, has been assigned in the cost function to the prediction of centres and radii. Figure 2 shows the real and estimated prices for the training set, where a Mean Absolute Percentage Error (MAPE) of 8.86% has been obtained for the center and 25.83% for its radius.

In the case of the validation set (see Figure 3), the MAPE has reached a value of 11.38% for the price centre and 24.54% for the radius. The similarity of these measures with the above in the training set confirms the generalization capability of the proposed model.

Table 2 compares the performance of the iMLP with a naïve reference model that proposes the observed price interval one week before as daily price interval forecast.

These results confirm the predictability of the mean daily price as a function of the generation mix and its associated costs (similar error rates have been published in the literature: see [27] and [28] for more elaborate reference models), accompanied by a high degree of random volatility affected by market participants' strategies.

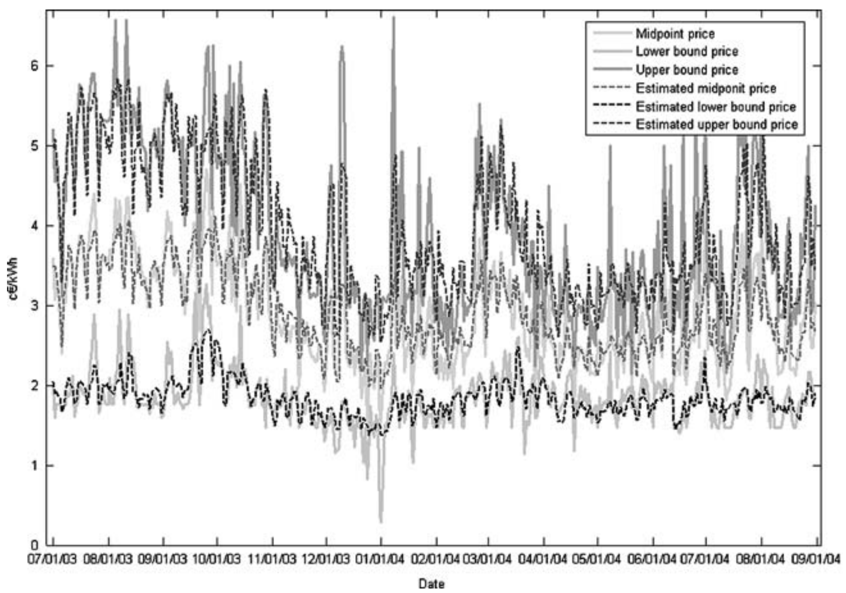

Figure 2. Training set estimation with iMLP(8,10,1).

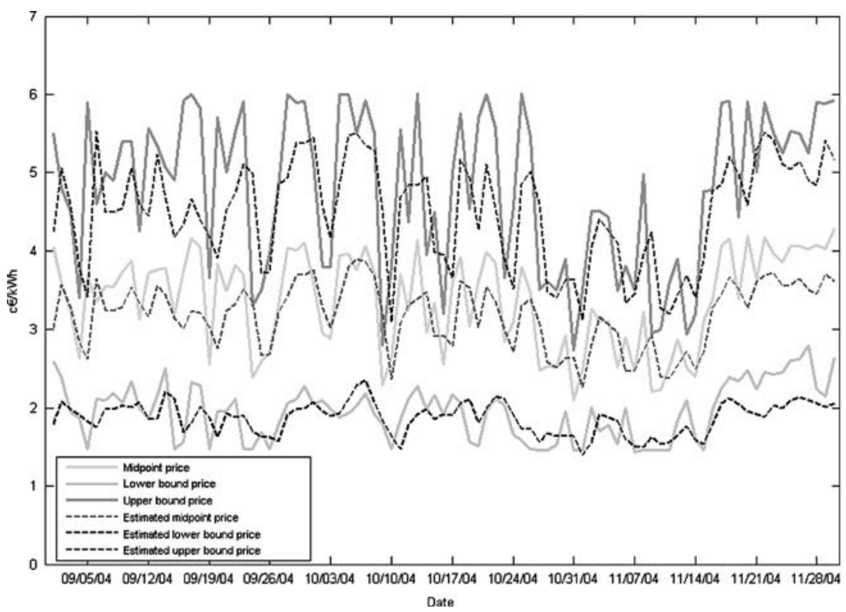

Figure 3. Validation set estimation with iMLP(8,10,1).

Table 2. Comparison of the forecasting performance between the iMLP and the naive model with stationality.

	Training set MAPE		Validation set MAPE	
	Centre (%)	Radius (%)	Centre (%)	Radius (%)
Naive model	15.85	33.82	17.86	33.44
iMLP	8.86	25.83	11.38	24.54

5. Conclusions

In this paper, a new model of MLP is proposed in order to handle interval-valued data. The proposed model has the architecture of a standard MLP with single-valued weights and bias, but its transfer function has been modified in order to operate with interval-valued inputs and outputs. The resulting model maps an input vector of intervals to an interval output by means of interval arithmetic.

Two applications of the iMLP have been considered. First, as an interval-valued function approximation model; and second, as a model that facilitates the evaluation of the prediction intervals of crisp MLP fed with interval-valued input data.

In the function approximation case, the parameters of the model are optimized by minimizing a cost function defined in terms of discrepancies between estimated and desired output intervals. The proposed cost function is a weighted sum of midpoints and radii squared estimation errors. The averaging weights in the error function allow tuning the importance assigned to midpoints and to radii according to each specific context. A backpropagation rule for the computation of the gradient has also been derived. The computation of the prediction intervals of a crisp MLP subject to input uncertainty is a straightforward result of the proposed architecture: once the crisp MLP has been trained with single-valued inputs and outputs, its response to interval inputs can be obtained directly by evaluating an iMLP with the same weights.

The first application has been illustrated in this paper by applying the iMLP to forecast daily electricity prices intervals as a function of the generation mix. The second one remains as ongoing work. Other extensions of the present work would include:

- The application of sensitivity analysis in order to quantify the effect of input midpoints and radii on the outputs of the iMLP.
- The application of iMLP to fuzzy regression analysis.
- Proposing MLPs models based on the iMLP for other types of symbolic data such as histograms, boxplots or distributions.

References

1. Smith, K. A.: Applications of neural networks. *Comp. Oper. Res.*, **32**(10) (2005), 2495–2497

2. Smith, K. A. and Gupta J. N. D.: Neural networks in business: techniques and applications for the operations researcher. *Comp. Oper. Res.*, **27**(11–12) (2000), 1023–1044.
3. Wong, B. K., Lai, V. S. and Lam, J.: A bibliography of neural networks business applications research: 1994–1998. *Comp. Oper. Res.*, **27**(11–12) (2000), 1045–1076.
4. Perez, M.: Artificial neural networks and bankruptcy forecasting: A state of the art. *Neural Comput. Appl.*, **15**(2) (2006), 154–163.
5. Zhang, G., Patuwo, B. E. and Hu, M. Y.: Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.*, **14**(1) (1998), 35–62.
6. Zhang, G. P.: An investigation of neural networks for linear time-series forecasting. *Computers and Operations Research*, **28**(12) (2001), 1183–1202.
7. Craven, M. W. and Shavlik, J. W.: Using neural networks for data mining. *Future Gener. Comp. Syst.*, **13**(2–3) (1997), 211–229.
8. Pal, S. K., Talwar, V. and Mitra, P.: Web mining in soft computing framework: relevance, state of the art and future directions. *IEEE Trans. on Neural Networks* **13**(5) (2002), 1163–1177.
9. Bishop, C. M.: *Neural Networks for Pattern Recognition*. Oxford University Press: Oxford, 1995.
10. Fine, T. L.: *Feedforward Neural Network Methodology*. Springer-Verlag: New York, 1999.
11. Ward, M., Peng, W. and Wang, X.: Hierarchical visual data mining for large-scale data. *Computational Statistics*, **19** (2004), 147–158.
12. Bock, H. H. and Diday, E.(eds.): *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag: Berlin, 2000.
13. Billard, L. and Diday, E.: From the statistics of data to the statistics of knowledge: symbolic data analysis. *J. Am. Stat. Asso.* **98**(462) (2003), 470–487.
14. Moore, R. E.: *Interval Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1966.
15. Kearfott, B.: Interval computations: Introduction, uses, and resources. *Euromath Bull.* **2**(1) (1996), 95–112.
16. Young, R. C.: The algebra of many-valued quantities. *Mathematische Annalen*, **104** (1931), 260–290.
17. Beheshti, M., Berrached, R., de Korvin, A., Hu, C. and Sirisaengtaksin,O.: On interval weighted three-layer neural networks. In: *Proceedings of the 31st Annual Simulation Symposium*, pp. 188–195, Boston, MA, USA, 1998.
18. Ishibuchi, H., Tanaka, H. and Okada, H.: An architecture of neural networks with interval weights and its application to fuzzy regression analysis. *Fuzzy Sets Syst.* **57** (1993), 27–39.
19. Baker, M. R. and Patil, R.: Universal approximation theorem for interval neural networks, *Reliab. Comput.* **4** (1998), 235–239.
20. Simoff, S. J.: Handling uncertainty in neural networks: An interval approach. In: *Proceedings of IEEE International Conference on Neural Networks*, pp. 606–610, Washington, DC, USA, 1996.
21. Drago, G. P. and Ridella, S.: Possibility and necessity pattern classification using an interval arithmetic perceptron. *Neural Comput. Appl.*, **8**(1) (1999), 40–52.
22. Rossi, F. and Conan-Guez, B.: Multilayer perceptron on interval data. In: K. Jajuga, A. Sokolowski and H. H. Bock (eds.), *Classification, Clustering, and Data Analysis (IFCS 2002)*, pp. 427–434, Cracow, Poland, 2002.
23. Patiño-Escarcina, R., Callejas Bedregal, B. and Lyra, A.: Interval computing in neural networks: One layer interval neural networks. In: G. Das and V. P. Gulati (eds.), *Proceedings of the 7th International Conference on Information Technology, CIT 2004*, pp. 68–75, Hyderabad, India, 2004.

24. Rumelhart, D. E., Hinton, G. E. and Williams, R. J.: Learning internal representations by error propagation. In: D. E. Rumelhart and J. M. McClelland (eds.), *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, pp. 318–362, MIT Press: Cambridge, MA, USA, 1986.
25. Girosi, F., Jones, M. and Poggio, T.: Regularization theory and neural networks architectures. *Neural Comput.* **7**(2) (1995), 219–269.
26. Luenberger, D. G.: *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley 1984.
27. Bunn, D. W. (ed.): *Modelling Prices in Competitive Electricity Markets*. John Wiley & Sons: Chichester, UK, 2004.
28. Mateo González, A., Muñoz San Roque, A. and García González, J.: Modeling and forecasting electricity prices with input/output hidden Markov models. *IEEE Trans. on Power Syst.* **20**(1) (2005), 13–24.