

Granular Data Regression with Neural Networks

Mario G.C.A. Cimino¹, Beatrice Lazzerini¹,
Francesco Marcelloni¹, and Witold Pedrycz²

¹ Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica,
Telecomunicazioni, University of Pisa, Largo Lucio Lazzarino 1 - 56122 Pisa, Italy
`{m.cimino,b.lazzerini,f.marcelloni}@iet.unipi.it`

² Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, Alberta, Canada, T6G 2G7
`pedrycz@ece.ualberta.ca`

Abstract. Granular data offer an interesting vehicle of representing the available information in problems where uncertainty, inaccuracy, variability or, in general, subjectivity have to be taken into account. In this paper, we deal with a particular type of information granules, namely interval-valued data. We propose a multilayer perceptron (MLP) to model interval-valued input-output mappings. The proposed MLP comes with interval-valued weights and biases, and is trained using a genetic algorithm designed to fit data with different levels of granularity. The modeling capabilities of the proposed MLP are illustrated by means of its application to both synthetic and real world datasets.

Keywords: Granular computing, Information granules, Neurocomputing, Interval analysis, Symbolic data analysis, Function approximation.

1 Introduction

Human capabilities are based on the ability of processing non-numeric information clumps (*granules*) rather than individual numeric values [1]. Information granules can be regarded as collections of objects that exhibit some similarity in terms of their properties of functional appearance [2]. There are a number of formal models of information granules including sets, rough sets, fuzzy sets, and shadowed sets to name a few options. In [3] the authors claim that the implementation of granules in terms of interval-valued data is the easiest to comprehend and express by a domain expert, and the simplest to process when there is a great variability of granule sizes.

The objective of this study is to propose a neural architecture to process information granules consisting of interval-valued data. Interval-valued data arise in several practical situations, such as recording monthly interval temperatures at meteorological stations, daily interval stock prices, inaccuracy of the measurement instruments, range of variation of a variable through time. In the proposed model, each operation performed in the network is based on interval arithmetic and this allows creating mappings at different levels of granularity. Since the

level of granularity is problem-oriented and user-dependent, it is a parameter of our neural architecture.

The first conceptualization of neural networks for processing granular data was introduced by Pedrycz and Vukovich in [3]. Here, several design approaches are discussed, together with a number of architectures of granular neural networks and associated training methods. Also, the authors tackle a number of fundamental issues of these networks, such as specificity of information granules, learning complexity, and generalization capabilities. Neural architectures based on interval arithmetic have been proposed in [2,4,5,8,9,10]. In particular, the model developed in [4] uses a standard multilayer perceptron (MLP) with numeric weights and biases, and a neuron transfer function able to operate with interval-valued inputs and outputs. Here, the training process uses an error function based on a weighted Euclidean distance between intervals, and a Quasi Newton method for the minimization of the error function. More robust minimization methods such as genetic algorithms and evolutionary strategies have been also proposed [5].

In its most general architecture proposed in the literature, an MLP that processes interval-valued data is characterized by weights and biases expressed in terms of intervals, and maps an interval-valued input vector to an interval-valued output. However, very often, in the design of the training methods some simplifying assumptions are made, e.g. input, weights and biases may be real numbers, or the error function between intervals is not compliant with the rules of the interval arithmetic.

This paper proposes a new genetic-based learning method for a general interval-valued neural architecture. We also show the effectiveness of this method by using three interval-valued datasets.

2 Interval Arithmetic: Some Definitions

We employ a basic implementation of granules in terms of conventional interval-valued data. An interval-valued variable \ddot{X} is defined as:

$$\ddot{X} = [\underline{x}, \overline{x}] \in \mathbb{IR}, \quad \underline{x}, \overline{x} \in \mathbb{R} . \quad (1)$$

where \mathbb{IR} is the set of all closed intervals in the real line, and \underline{x} and \overline{x} are the boundaries of the intervals. An F -dimensional granule is then represented by a vector of interval-valued variables as follows:

$$\ddot{\mathbf{X}} = [\ddot{X}_1, \dots, \ddot{X}_F] \in \mathbb{IR}^F, \quad \ddot{X}_i \in \mathbb{IR} . \quad (2)$$

Sometimes an interval variable is expressed in terms of its midpoint \dot{x} and half-width \hat{x} , as follows [5]:

$$\hat{\ddot{X}} = \langle \dot{x}, \hat{x} \rangle \in \mathbb{IR}, \quad \dot{x}, \hat{x} \in \mathbb{R}, \quad \dot{x} = (\overline{x} + \underline{x})/2, \quad \hat{x} = (\overline{x} - \underline{x})/2 . \quad (3)$$

Table 1 summarizes some basic operations of interval arithmetic that have been used in this study. The interested reader can find a detailed discussion in [6,7].

3 The Adopted Interval-Valued Neural Architecture

Let $f : \mathbb{IR}^F \rightarrow \mathbb{IR}$ be an F -dimensional interval-valued regression model:

$$f(\ddot{\mathbf{X}}) = f(\ddot{X}_1, \dots, \ddot{X}_F) = \ddot{Y}. \quad (4)$$

Table 1. Some basic interval arithmetic operations used in interval-valued MLP

Operation	Implementation
Addition	$[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$
Multiplication	$[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] = [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}]$
Function evaluation	$F([\underline{x}, \overline{x}]) = [F(\underline{x}), F(\overline{x})]$, F monotonically increasing
Real distance	$\text{dist}([\underline{x}, \overline{x}], [\underline{y}, \overline{y}]) = \max\{ \underline{x} - \underline{y} , \overline{x} - \overline{y} \}$
Absolute value	$ \underline{x}, \overline{x} = \text{dist}([\underline{x}, \overline{x}], [0, 0]) = \max\{ \underline{x} , \overline{x} \}$

Fig. 1 shows the MLP we adopt to deal with the regression problem modeled by (4). This architecture has been already proposed by some authors (for instance, in [5]). The novelty of our approach concerns the training process, which allows an effective and efficient sensitivity analysis (i.e., to quantify the effect of input variability on the outputs). The hidden layer comprises N nonlinear hidden units and the output layer consists of one linear output unit. The activation of each hidden unit j is obtained as sum between the weighted linear combination, with weights $\ddot{\Omega}_{i,j}$, $i = 1, \dots, F$, $j = 1, \dots, N$, of the F interval-valued inputs \ddot{X} and the bias $\ddot{\Omega}_{0,j}$. Since both weights and biases are intervals, this linear combination results in a new interval. The output of each hidden unit is then obtained by transforming its activation interval using a hyperbolic tangent (sigmoid) function. Since the function is monotonic, this transformation yields a new interval [5]. Finally, the output of the network, \ddot{Y} , is obtained as the sum between the weighted linear combination, with weights $\ddot{\Omega}_j$, $j = 1, \dots, N$, of the outputs of the hidden layer, and the bias $\ddot{\Omega}_0$. The overall processing method is based on the fundamental arithmetic operations on \mathbb{IR} shown in Table 1. The resulting model can be used in two ways [4]: (i) as a granular function approximation model, whose granular weights can be adjusted through supervised learning by minimizing an error function; (ii) as an instrument to evaluate the prediction

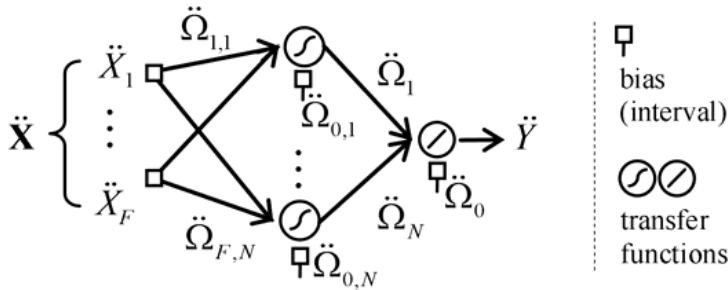


Fig. 1. The proposed architecture of MLP for interval-valued data

of a pre-adjusted MLP model subject to variable uncertainty associated with its input variables. Such input uncertainty can be characterized using interval inputs of different lengths.

4 The Training of the Interval-Valued MLP

As pointed out in [5], the width of the predicted output regions for an interval MLP is affected by the width of the weight intervals. Wide widths cause the propagation of large ranges of intermediate values through the network, thus generating wide output intervals. This is known as “bound explosion” effect. To control this effect, we adopt the following procedure. Let $\{\ddot{\mathbf{X}}, \ddot{\mathbf{Y}}\} \in \mathbb{IR}^{F+1}$ be a set of T input-output interval-valued samples, represented as midpoint and half-width. First, we use the midpoints of T to train a conventional MLP which has the same structure as the interval-valued MLP to be developed. We adopt the Levenberg-Marquadt method. In this way, we form a reference model, which solves the regression problem for reference points of the interval-valued data. When we tackle the regression problem for the interval-valued data, we expect that the interval weights and biases contain the numerical weights and biases ($\dot{\omega}_{i,j}^{(init)}$ and $\dot{\omega}_j^{(init)}$) of the reference model, respectively. Further, we expect that the widths of these intervals are constrained by the level of granularity of the mapping that is determined by the problem and by the desired resolution with which the user is interested in observing the data.

The first requirement is satisfied by enforcing the following relationship:

$$\dot{\omega}_{i,j}^{(init)} \in \ddot{\Omega}_{i,j} \text{ and } \dot{\omega}_j^{(init)} \in \ddot{\Omega}_j \quad \forall i, j. \quad (5)$$

As regards the second requirement, we enforce that the half-widths of weights and biases are bounded by an interval-valued percentage of the initial values:

$$\hat{\omega}_{i,j} \in |\dot{\omega}_{i,j}^{(init)}| \cdot \ddot{G} \text{ and } \hat{\omega}_j \in |\dot{\omega}_j^{(init)}| \cdot \ddot{G} \quad \forall i, j. \quad (6)$$

where $\ddot{G} = [\underline{g}, \overline{g}] \in \mathbb{IR}^+$, with $\underline{g}, \overline{g} \in \mathbb{R}^+$, is a granularity interval expressed in percentages which allows to adapt the granularity of the mapping to the granularity level of the information. The choice of \ddot{G} depends on the specific performance index used to assess the quality of the model. For instance, in our case we used the network error.

To learn the interval-valued weights and biases, standard error back propagation is likely to give poor results [5]. Indeed, the network prediction error surface is expected to be very nonlinear with several local minima. A global search method is much more desirable. Genetic algorithms (GAs) and evolutionary strategies are, in general, effective examples of such methods. Thus, we decided to adopt a GA. Fig. 2 shows the chromosome coding used in the GA. The initial population is randomly generated by satisfying the constraints in formulas (5) and (6). Chromosomes are selected for mating by a fitness proportional strategy. We apply the classical two-point crossover operator, with a user-defined crossover probability P_c . The mutation operator is controlled by a mutation probability defined as γ_m/L , where γ_m is a user-defined mutation coefficient and $L = 2N(F+2) + 2$ is the chromosome length. We randomly choose

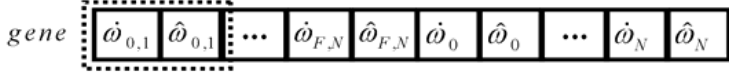


Fig. 2. The chromosome coding

a user specified percentage P_R of the genes that undergo mutation. Then, we replace the current values of each selected gene by randomly extracting two values in the intervals $[\dot{\omega}_{i,j}^{(init)} \cdot \underline{g}, |\dot{\omega}_{i,j}^{(init)}| \cdot \overline{g}]$ and $[\dot{\omega}_{i,j}^{(init)} - \hat{\omega}_{i,j}, \dot{\omega}_{i,j}^{(init)} + \hat{\omega}_{i,j}]$, respectively. The first interval is directly related to the definition of \ddot{G} . Once provided $\hat{\omega}_{i,j}$, from formulas given by (6) we derive that the maximum distance from $\dot{\omega}_{i,j}^{(init)}$ can be $\dot{\omega}_{i,j}^{(init)} \pm \hat{\omega}_{i,j}$.

As regards the fitness function, unlike the network error functions proposed in the literature that implicitly assume an isomorphism between \mathbb{IR} and \mathbb{R}^2 , we adopt the following error function directly derived from the interval arithmetic operations shown in Table 1:

$$E = \frac{1}{T} \sum_{i=1}^T E_i, \quad E_i = \text{dist}(\ddot{Y}_i, \ddot{Y}'_i) \in \mathbb{R}^+. \quad (7)$$

where \ddot{Y}_i and \ddot{Y}'_i are the desired and network outputs. The algorithm stops if a maximum number N_G of generations is reached or if the best fitness of the population is lower than a prefixed fitness threshold τ .

5 Experimental Results

A variety of works have been developed in the field of interval-valued data. Unfortunately there is still a lack of significant benchmark datasets for interval-valued data regression. In this section, we discuss the application of our interval-valued neural architecture to one real world and two synthetic datasets. In all experiments, the data are normalized between 0 and 1 (by subtracting the minimum value and dividing the data by the difference between the maximum and the minimum values). The population of the GA consists of 20 chromosomes. The parameters P_c , γ_m , P_R , and τ have been set to 0.4, 0.7, 10%, and 0.001, respectively. We used a value of N_G equals to 500 except for the experiment in section 5.3 where we adopted a value equals to 1000.

5.1 The Salary Dataset

The Salary dataset [10] shown in Fig. 3.a consists of 30 interval-valued samples which represent the range of salaries by years of experience for American males with degree in 1989. The original data samples are not granular, and subject to significant sampling error. First, fuzzy information granules have been generated via FCM clustering. Hence, an alpha-cut of 0.05 has been applied to the resulting fuzzy partition. Finally, interval-valued data have been derived considering, for each alpha-cut, the smallest containing rectangle. We adopted a 10-fold cross-validation: for each trial, the training and the test sets consist of the randomly

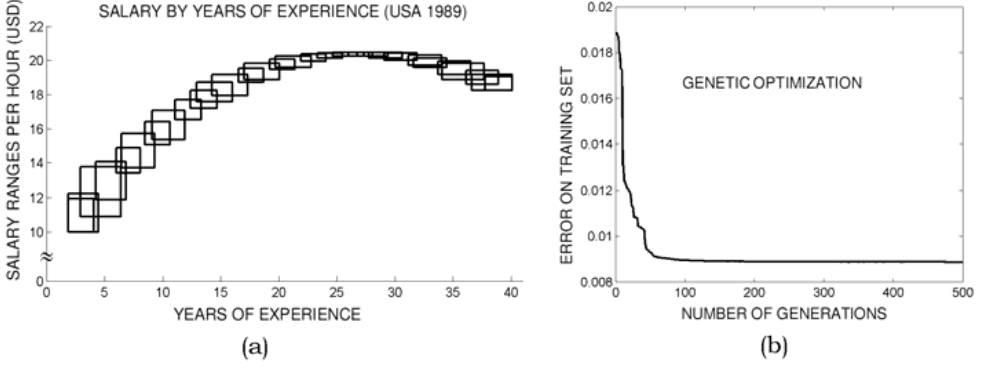


Fig. 3. (a) The Salary dataset. (b) Training error versus generations in a trial.

extracted 90% and 10% of the original data, respectively. The granularity interval used to observe the mapping is $\ddot{G} = [0, 4]\%$. The network has been equipped with 15 hidden neurons. The mean values \pm the standard deviations of the error on training and test set are, respectively, 0.007 ± 0.0051 and 0.018 ± 0.028 . We can observe a good balance between the values of errors for the training and test sets. This confirms sound generalization capabilities of the network.

Fig. 3.b shows the error of the best chromosome of each generation versus the number of generations in a sample trial. We observe that the error gets stable around 100 generations.

5.2 The Peak Dataset

The Peak dataset shown in Fig. 4.a consists of 189 synthetic interval-valued samples. Again, we adopted a 10-fold cross-validation. The granularity interval used to observe the mapping is $\ddot{G} = [0.4, 4]\%$. The network has been equipped with 30 hidden neurons. The mean values \pm the standard deviations of the

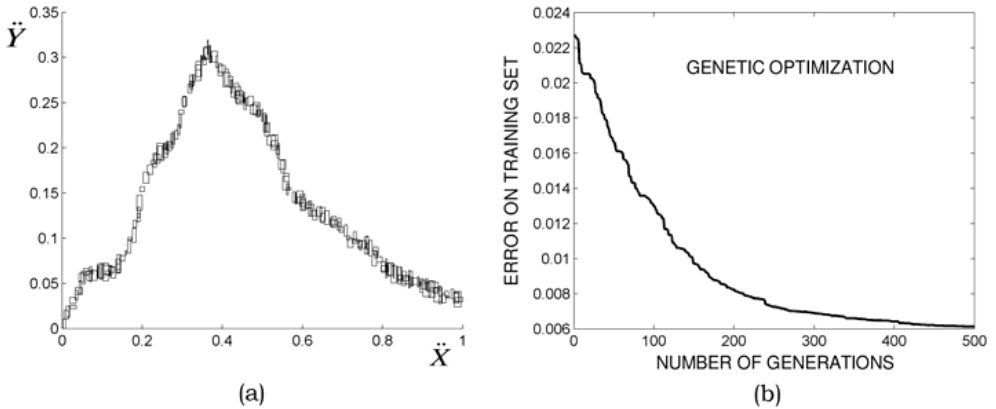


Fig. 4. (a) The Peak dataset. (b) Training error versus generations in a trial.

error on training and test set are, respectively, 0.0064 ± 0.0014 and 0.0085 ± 0.0043 . We can observe that the error in the test set is very close to the error in the training set, thus pointing out the good generalization capabilities of the network. Fig. 4.b shows the error of the best chromosome of each generation versus the number of generations in a sample trial. We observe that the error gets stable around 500 generations.

5.3 The Wave Dataset

The Wave dataset shown in Fig. 5.a consists of 400 synthetic interval-valued samples in the three-dimensional space. The network has been equipped with 15 hidden neurons. We use this dataset for analyzing the differences between the mappings with different granularity intervals. Fig. 5.b shows the error of the best chromosome of each generation versus the number of generations when

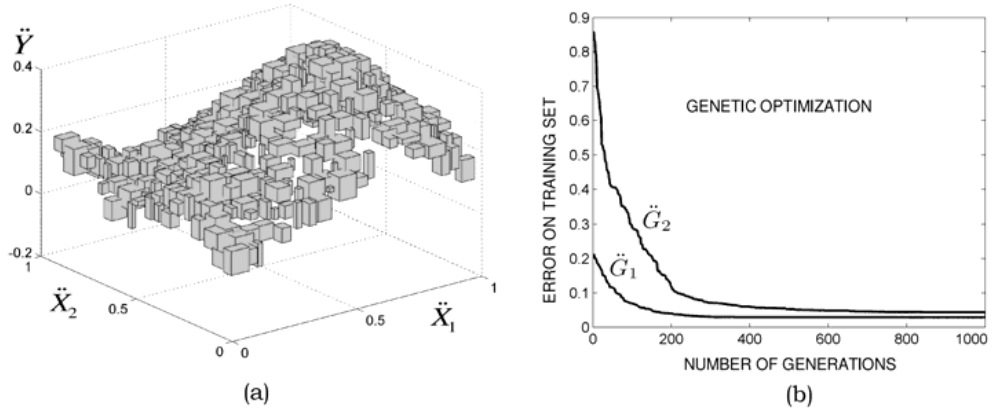


Fig. 5. (a) The Wave dataset. (b) Training error versus generations with $\ddot{G}_1 = [0.2, 10]\%$ and $\ddot{G}_2 = [10, 50]\%$.

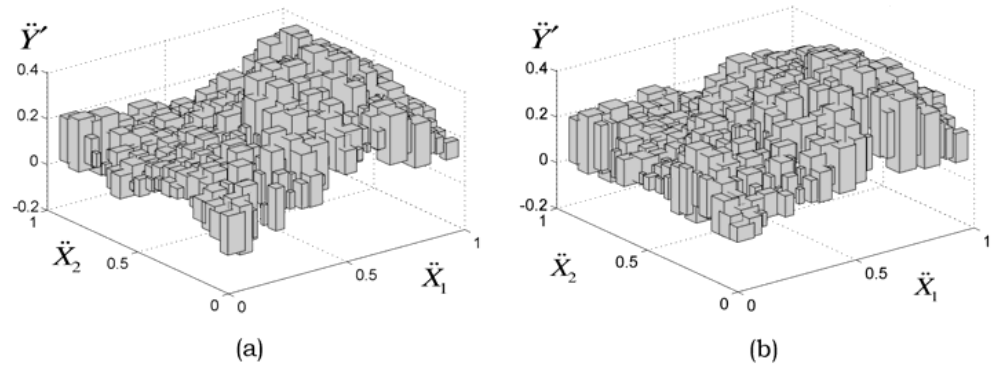


Fig. 6. Models generated by the network. (a) $\ddot{G}_1 = [0.2, 10]\%$. (b) $\ddot{G}_2 = [10, 50]\%$.

the network is trained with the granularity intervals $\ddot{G}_1 = [0.2, 10]\%$ and $\ddot{G}_2 = [10, 50]\%$.

Fig. 6.a and Fig. 6.b show the models generated by the network in the two cases, respectively. We can observe how the model generated by using \ddot{G}_2 is coarser than the model generated using \ddot{G}_1 . Further, in the former, the error gets stable around 500 generations against the 300 of the latter.

6 Conclusions

We have proposed a new genetic-based learning method for a general interval-valued neural architecture. The originality of the approach concerns the training process which allows a valuable sensitivity analysis. We have quantified the effectiveness of the approach in terms of generalization capabilities and sensitivity by using three interval-valued datasets.

References

1. Pedrycz, W.: Knowledge-Based Clustering: From Data to Information Granules. John Wiley and Sons, Hoboken (2005)
2. Nandedkar, A.V., Biswas, P.K.: A Reflex Fuzzy Min Max Neural Network for Granular Data Classification. In: 18th Int. Conference on Pattern Recognition, vol. 2, pp. 650–653 (2006)
3. Pedrycz, W., Vukovich, G.: Granular Neural Networks. *Neurocomp* 36, 205–224 (2001)
4. Muñoz San Roque, A., Maté, C., Arroyo, J., Sarabia, Á.: iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data. *Neural Processing Letters* 25, 157–169 (2007)
5. Chetwynd, D., Worden, K., Manson, G.: An application of interval-valued neural networks to a regression problem. *Proceedings of the Royal Society A* 462, 3097–3114 (2006)
6. Kearfott, R.B.: Interval computations-Introduction, uses, and resources. *Euro-math. Bull.* 2, 95–112 (1996)
7. Alefeld, G., Claudio, D.: The basic properties of interval arithmetic, its software realizations and some applications. *Computers & Structures* 67, 3–8 (1998)
8. Ishibuchi, H., Tanaka, H., Okada, H.: Architecture of neural networks with interval weights and its application to fuzzy regression analysis. *Fuzzy Sets and Systems* 57, 27–39 (1993)
9. Baker, M.R.: Universal Approximation Theorem for Interval Neural Network. *Reliable Computing* 4, 235–239 (1998)
10. Patio-Escarcina, R.E., Callejas Bedregal, B.R., Lyra, A.: Interval Computing in Neural Networks: One Layer Interval Neural Networks. *Intelligent Information Technology* 3356, 68–75 (2005)
11. Murphy, K.M., Welch, F.: The Structure of Wages. *Quarterly Journal of Economics* 1, 285–326 (1992)