TWNFI – Transductive Neural-Fuzzy Inference System with Weighted Data Normalization and Its Application in Medicine

Qun Song
Knowledge Engineering &
Discovery Research Institute
Auckland University of Technology

Private Bag 92006,
Auckland 1020, New Zealand
E-mail: qsong@aut.ac.nz

Nikola Kasabov
Knowledge Engineering &
Discovery Research Institute
Auckland University of Technology

Private Bag 92006,
Auckland 1020, New Zealand
E-mail: nkasabov@aut.ac.nz

Abstract: This paper introduces a novel fuzzy model – transductive neural-fuzzy inference system with weighted data normalization (TWNFI). While inductive approaches are concerned with the development of a model to approximate data in the whole problem space (induction), and consecutively – using this model to calculate the output value(s) for a new input vector (deduction), in transductive systems a local model is developed for every new input vector, based on some closest to this vector data from the training data set. The weighted data normalization method (WDN) optimizes the data normalization ranges for the input variables of a system. A steepest descent algorithm is used for training the TWNFI model. The TWNFI is illustrated on a case study example: a real medical decision support problem of estimating the level of renal functions in patients. This method can also be applied to other distance-based, prototype learning neural network or fuzzy inference models.

1. Introduction: Transductive Model and Weighted Data Normalization

Most of learning models and systems in artificial intelligence developed and implemented so far are based on *inductive* methods, where a model (a function) is derived from data representing the problem space and this model is further applied on new data. The model is usually created without taking into account any information about a particular new data vector (test data). An error is measured to estimate how well the new data fits into the model. The inductive learning and inference approach is useful when a global model ("the big picture") of the problem is needed even in its very approximate form. In contrast to the inductive learning and inference methods, transductive inference methods estimate the value of a potential model (function) only in a single point of the space (the new data vector) utilizing additional information related to this point. This approach seems to be more appropriate for clinical and medical applications of learning systems, where the focus is not on the model, but on the individual patient. Each individual data vector (e.g.: a patient in the medical area; a future time moment for predicting a time series; or a target day for predicting a stock index) may need an individual, local model that best fits the new data, rather then a global model, in which the new data is matched without taking into account any specific information about this data.

Transductive inference is concerned with the estimation of a function in single point of the space only. For every new input vector $\mathbf{x}_i$ that needs to be processed for a prognostic task, the N_i nearest neighbours, which form a sub-data set D_i, are derived from an existing data set D or/and generated from an existing model M. A new model M_i is dynamically created from these samples to approximate the function in the point $\mathbf{x}_i$ - Figure 1 and Figure 2. The system is then used to calculate the output value $\mathbf{y}_i$ for this input vector $\mathbf{x}_i$ (Figures 1 and 2).

In many neural network and fuzzy models and applications, raw (not normalized) data is used. This is appropriate when all the input variables are measured in the same units. Normalization, or standardization, is reasonable when the variables are in different units, or when the variance between them is substantial. However, a general normalization means that every variable is normalized in the same range, e.g. [0, 1] with the assumption that they all have the same importance for the output of the system.

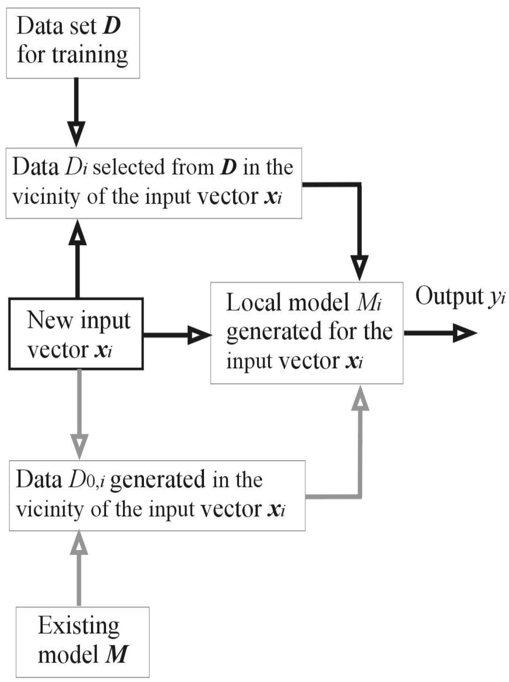
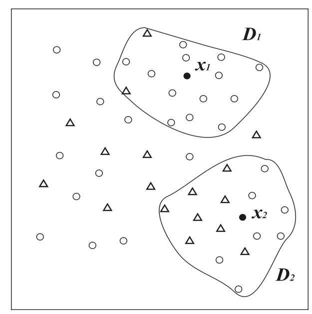


Figure 1. A block diagram of a *transductive* reasoning system. An individual model M_i is trained for every new input vector $\boldsymbol{x}_i$ with data use of samples D_i selected from a data set D, and data samples $D_{0,i}$ generated from an existing model (formula) M (if such a model is existing). Data samples in both D_i and $D_{0,i}$ are similar to the new vector $\boldsymbol{x}_i$ according to defined similarity criteria.

For many practical problems, variables have different importance and make different contribution to the output(s). Therefore, it is necessary to find an optimal normalization and assign proper importance factors to the variables. Such a method can also be used for feature selection, or for reducing the size of input vectors through keeping the most important ones [9]. This is especially applicable to a special class of neural networks or fuzzy models – the clustering based models (or also: distance-based; prototype-based) such as RBF [8] and ECOS [2, 3]. In such systems, distance between neurons or fuzzy rule nodes and input vectors are usually measured in *Euclidean distance*, so that variables with a wider range will have more influence on the learning process and vice versa.

The paper is organized as follows: Section2 presents the structure and algorithm of the TWNFI model. Sections3 illustrates the approach on a case study example. Conclusions are drawn in Section4.



● – a new data vector
○ – a sample from D
Δ – a sample from M

Figure 2. In the centre of a transductive reasoning system is the new data vector (here illustrated with two of them – $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$), surrounded by a fixed number of nearest data samples selected from the training data D and generated from an existing model M.

2. Transductive Neural Fuzzy System with Weighted Data Normalization: Structure and Learning Algorithm

TWNFI is a dynamic neural-fuzzy inference system with a local generalization, in which, the *Zadeh-Mamdani* type fuzzy inference engine [12] is used. The local generalization means that in a sub-space of the whole problem space (local area) a model is crated that performs generalization in this area. In the TWNFI model, *Gaussian* fuzzy membership functions are applied in each fuzzy rule for both the antecedent and the consequent parts. A steepest descent (BP) learning algorithm is used for optimizing the parameters of the fuzzy membership functions [6, 11]. The distance between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is measured in TWNFI in *normalized Euclidean distance* defined as follows (the values are between 0 and 1):

$$\|\boldsymbol{x}-\boldsymbol{y}\| = \frac{1}{P}\left[\sum_{j=1}^{P}|x_j - y_j|^2\right]^{\frac{1}{2}} \quad (1)$$

where: $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}^P$.

To partition the input space for creating fuzzy rules and obtaining initial values of fuzzy rules, the *ECM* (Evolving Clustering Method) is applied [4, 10] and the cluster centers and cluster radiuses are respectively taken as initial values of the centres and widths of the *Gaussian* membership functions.

For each new data vector $\boldsymbol{x}_q$, the TWNFI learning algorithm performs the following steps:

1. Normalize the training data set (the values are between 0 and 1) with the initial input variable weights.
2. Search in the training data set in the input space to find N_q training examples that are closest to $\boldsymbol{x}_q$. The value for N_q can be pre-defined based on experience, or - optimised through the application of an optimization procedure. Here we assume the former approach.
3. Calculate the distances d_i, i = 1, 2, …, N_q, between each of these data samples and $\boldsymbol{x}_q$. Calculate the vector weights $v_i = 1 - (d_i - \min(\boldsymbol{d}))$, i = 1, 2, …, N_q, $\min(\boldsymbol{d})$ is the minimum value in the distance vector $\boldsymbol{d} = [d_1, d_2, \ldots, d_{Nq}]$.
4. Use the *ECM* clustering algorithm to cluster and partition the input sub-space that consists of N_q selected training samples.
5. Create fuzzy rules and set their initial parameter values according to the *ECM* clustering procedure results; for each cluster, the cluster centre is taken as the centre of a fuzzy membership function (*Gaussian* function) and the cluster radius is taken as the width.
6. Apply the steepest descent method (back-propagation) to optimize the parameters of the fuzzy rules in the local model M_q following Eq. (6 – 13).
7. Re-normalize the training data set (the values are between 0 and 1) with the optimized variable weights.
8. Search in the training data set to find N_q samples (the same to Step 2), if the same samples are found as the last search the algorithm turns to Step 9, otherwise, Step 3.
9. Calculate the output value y_q for the input vector x_q applying fuzzy inference over the set of fuzzy rules that constitute the local model M_q.
10. End of the procedure.

The parameter optimisation procedure is described below:

Consider the system having P inputs, one output and M fuzzy rules defined initially through the ECM clustering procedure, the l-th rule has the form of:

R_l: If x_1 is F_{l1} and x_2 is F_{l2} and … x_P is F_{lP}, then y is G_l. (2)

Here, F_{lj} are fuzzy sets defined by the following *Gaussian* type membership function:

$$\text{GaussianMF} = \alpha \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] \quad (3)$$

and G_l are of a similar type as F_{lj} and are defined as:

$$GaussianMF = \exp\left[-\frac{(y-n)^2}{2\delta^2}\right] \quad (4)$$

Using the *Modified Centre Average defuzzification* procedure the output value of the system can be calculated for an input vector $\boldsymbol{x}_i = [x_1, x_2, \ldots, x_P]$ as follows:

$$f(\boldsymbol{x}_i) = \frac{\sum_{l=1}^{M}\frac{G_l}{\delta_l^2}\prod_{j=1}^{P}\alpha_{lj}\exp\left[-\frac{w_j^2(x_{ij}-m_{lj})^2}{2\sigma_{lj}^2}\right]}{\sum_{l=1}^{M}\frac{1}{\delta_l^2}\prod_{j=1}^{P}\alpha_{lj}\exp\left[-\frac{w_j^2(x_{ij}-m_{lj})^2}{2\sigma_{lj}^2}\right]} \quad (5)$$

Here, w_j are weights of the input variables.

Suppose the TWNFI is given a training input-output data pair $[\boldsymbol{x}_i, t_i]$, the system minimizes the following objective function (a weighted error function):

$$\text{E} = \frac{1}{2}v_i\left[f(x_i) - t_i\right]^2 \quad (6)$$

(v_i are defined in Step 3)

The steepest descent algorithm (BP) is used then to obtain the formulas for the optimization of the parameters G_l, δ_l, α_{lj}, m_{lj}, σ_{lj} and w_j such that the value of E from Eq. (6) is minimized:

$$G_l(k+1) = G_l(k) - \frac{\eta_G}{\delta_l^2(k)} v_i \Phi(\mathbf{x}_i)[f^{(k)}(\mathbf{x}_i) - t_i] \quad (7)$$

$$\delta_l(k+1) = \delta_l(k) - \frac{\eta_\delta v_i \Phi(\mathbf{x}_i)}{\delta_l^3(k)}[f^{(k)}(\mathbf{x}_i) - t_i][f^{(k)}(\mathbf{x}_i) - G_l(k)] \quad (8)$$

$$\alpha_{lj}(k+1) = \alpha_{lj}(k) - \frac{\eta_\alpha v_i \Phi(\mathbf{x}_i)}{\delta_l^2(k)\alpha_{lj}(k)}[f^{(k)}(\mathbf{x}_i) - t_i][G_l(k) - f^{(k)}(\mathbf{x}_i)] \quad (9)$$

$$m_{lj}(k+1) = m_{lj}(k) - \frac{\eta_m w_j^2(k) v_i \Phi(\mathbf{x}_i)}{\delta_l^2(k)\,\sigma_{lj}^2(k)}[f^{(k)}(\mathbf{x}_i) - t_i][G_l(k) - f^{(k)}(\mathbf{x}_i)][x_{ij} - m_{lj}(k)] \quad (10)$$

$$\sigma_{lj}(k+1) = \sigma_{lj}(k) - \frac{\eta_\sigma w_j^2(k) v_i \Phi(\mathbf{x}_i)}{\delta_l^2(k)\,\sigma_{lj}^3(k)}[f^{(k)}(\mathbf{x}_i) - t_i][G_l(k) - f^{(k)}(\mathbf{x}_i)][x_{ij} - m_{lj}(k)]^2 \quad (11)$$

$$w_j(k+1) = w_j(k) - \frac{\eta_w w_j(k) v_i \Phi(\mathbf{x}_i)}{\delta_l^2(k)\,\sigma_{lj}^2(k)}[f^{(k)}(\mathbf{x}_i) - t_i][f^{(k)}(\mathbf{x}_i) - G_l(k)][x_{ij} - m_{lj}(k)]^2 \quad (12)$$

here,

$$\Phi(\mathbf{x}_i) = \frac{\prod_{j=1}^{P} \alpha_{lj} \exp\left\{-\frac{w_j^2(k)[x_{ij} - m_{lj}(k)]^2}{2\sigma_{lj}^2(k)}\right\}}{\sum_{l=1}^{M} \frac{1}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left\{-\frac{w_j^2(k)[x_{ij} - m_{lj}(k)]^2}{2\sigma_{lj}^2(k)}\right\}} \quad (13)$$

where: η_G , η_δ , η_α , η_m , η_σ η_w and are learning rates for updating the parameters G_l, δ_l, α_{lj}, m_{lj}, σ_{lj} and w_j respectively.

In the TWNFI training–simulating algorithm, the following indexes are used:

- Training data samples: $i = 1, 2, \ldots, N$;
- Input variables: $j = 1, 2, \ldots, P$;
- Fuzzy rules: $l = 1, 2, \ldots, M$;
- Training epochs: $k = 1, 2, \ldots$.

3. Case Study example of applying the TWRBF for A Medical Decision Support Problem

A real data set from a medical institution is used here for experimental analysis. The data set has 447 samples, collected at hospitals in New Zealand and Australia. Each of the records includes six variables (inputs): age, gender, serum creatinine, serum albumin, race and blood urea nitrogen concentrations, and one output - the glomerular filtration rate value (GFR). All experimental results reported here are based on 10-cross validation experiments with the same model and parameters and the results are averaged. In each experiment 70% of the whole data set is randomly selected as training data and another 30% as testing data.

For comparison, several well-known methods are applied on the same problem, such as the MDRD function [5], MLP neural network [7], adaptive neural fuzzy inference system (ANFIS) [1] and dynamic evolving neural fuzzy inference system (DENFIS) [4] along with the proposed TWNFI and results are given in Table 1. The results include the number of fuzzy rules (fuzzy models), or neurons in the hidden layer (MLP), the testing RMSE (root mean square error), and the testing MAE (mean absolute error).

Two experiments with TWNFI are conducted. The first one is the TWNFI without WDN: all weights' values are set as '1' and will not be changed during the learning. Another employs the TWNFI learning algorithm described in Section 2. The experimental results illustrate that the TWNFI is efficient on such kind problem.

Table 1. The Experimental Results on GFR Data

Model	*Rules or neurons*	*RMSE*	*MAE*
MDRD	–	7.74	5.88
MLP	12	8.44	5.75
ANFIS	36	7.49	5.48
DENFIS	27	7.36	5.29
TWNFI Without WDN	6.8 (average)	7.31	5.30
TWNFI	**6.8 (average)**	**7.12**	**5.19**

4. Conclusions

This paper presents a transductive neuro-fuzzy inference with weighted data normalization method – TWNFI. The TWNFI performs a better local generalisation over new data as it develops an individual model for each data vector that takes into account the new input vector location in the space, and it is an adaptive model, in the sense that input-output pairs of data can be added to the data set continuously and immediately made available for transductive inference of local models. This type of modelling can be called "personalised", and it is promising for medical decision support systems. As the TWNFI creates a unique sub-model for each data sample, it usually needs more performing time than an inductive models, especially in the case of training and simulating on large data sets.

Further directions for research include: (1) TWNFI system parameter optimization such as optimal number of nearest neighbours; and (2) applications of the TWNFI method for other decision support systems, such as: cardio-vascular risk prognosis; biological processes modelling and prediction based on gene expression micro-array data.

Acknowledgement

The research presented in the paper is funded by the New Zealand Foundation for Research, Science and Technology under grant NERF/AUTX02-01.

References

[1] Jang, R. "ANFIS: adaptive network based fuzzy inference system", *IEEE Trans. on Syst.,Man, and Cybernetics*, vol. 23, No.3, pp 665 – 685, 1993.

[2] Kasabov, N. Evolving connectionist systems: Methods and Applications in Bioinformatics, Brain study and intelligent machines, Springer Verlag, London, New York, Heidelberg, 2002.

[3] Kasabov, N. "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge–based learning," *IEEE Trans. SMC* – part B, Cybernetics, vol.31, No.6, 902-918, December 2001.

[4] Kasabov, N. and Song, Q. "DENFIS: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction," *IEEE Trans. on Fuzzy Systems*, vol. 10, pp. 144 – 154, 2002.

[5] Levey, A.S., Bosch, J,P., Lewis, J.B., Greene, T., Rogers, N., Roth, D. for the Modification of Diet in Renal Disease Study Group, " A More Accurate Method To Estimate Glomerular Filtration Rate from Serum Creatinine: A New Prediction Equation", *Annals of Internal Medicine*, vol. 130, pp. 461 – 470, 1999.

[6] Lin, C.T. and Lee, C.S.G., *Neuro Fuzzy Systems*,. Prentice Hall, 1996.

[7] *Neural Network Toolbox User's Guide*, The Math Works Inc., ver. 4, 2001.

[8] Poggio, F. "Regularization theory, radial basis functions and networks" In: *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*. NATO ASI Series, No.136, pp. 83 – 104, 1994.

[9] Song, Q. and Kasabov, Q. "Weighted Data Normalizations and feature Selection for Evolving Connectionist Systems Proceedings", *Proc. of The Eighth Australian and New Zealand Intelligence Information Systems Conference* (ANZIIS2003), pp. 285 – 290, Sydney, Australia, December, 2003.

[10] Song, Q. and Kasabov, N. "ECM - A Novel On-line, Evolving Clustering Method and Its Applications", *Proceedings of the Fifth Biannual Conference on Artificial Neural Networks and Expert Systems (ANNES2001)*, pp. 87 – 92, Dunedin, New Zealand, November, 2001.

[11] Wang, L.X., *Adaptive Fuzzy System And Control: Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1994.

[12] Zadeh, L. A. . "Fuzzy Sets." Information and Control **8**: 338 – 353, 1965.