EANN95, Helsinki 21-23.08.1995

Włodzisław Duch, Norbert Jankowski, Antoine Naud and Rafał Adamczak
Department of Computer Methods, Nicholas Copernicus University
Grudziądzka 5, 87-100 Toruń, Poland
e-mail: duch@phys.uni.torun.pl

Feature Space Mapping: a neurofuzzy network for system identification

Our original motivation for the development of the Feature Space Mapping (FSM) system came from cognitive modelling. Mind arises from complex dynamics of the brain. Approximations to this dynamics lead to a set of concepts [1] helpful in description of the mind, such as the “inner space” concept, called also the “conceptual space” or the “mind space”. Among many other aspects mind models should be capable of recognition, classification and reasoning. Cognitive modelling is not always faithful to neurobiology, but a natural implementation of such models has a neural network form.

In this paper we present one particular realisation of general ideas related to cognitive modelling [1] leading to a neurofuzzy network useful for system identification. The low level cognitive processes accomplished by various topographic maps in the brain define features of internal representations $\mathbf{X}_i(t)$ for the incoming signals $\mathbf{I}(t)$. These features may be of many types: shapes derived from the analogue sensory signals, numbers, linguistic variables. A coordinate system based on these features $\{\mathbf{X}_i\}$ defines a multidimensional feature space, called here “the mind space” since axes of some of the coordinate systems we use represent confidence factors and partial classification results, not only simple features. The system learns by creating and modifying “mind objects” in this space. They are described using “mind function” $M(\mathbf{X})$ as a fuzzy areas in the mind space where the function has non-zero values. Local maxima of the mind function are prototypical representations of the (fuzzy) training data.

Feature detectors react to specific localised features of the incoming data, therefore objects in the mind space are modelled via localised functions rather then unbounded, sigmoidal functions that are most frequently used in neural models. In some cases unbounded functions are also useful. From the point of view of efficiency and flexibility of the system good choice is provided by [2]:

$$M(\mathbf{X};\mathbf{D},\Delta) = \sum_{k=1}^{K} W_k\, s(\mathbf{X};\mathbf{D}_k,\Delta_k) \tag{1}$$

$$s(\mathbf{X};\mathbf{D}_k,\Delta_k) = \prod_{i=1}^{d} s_i(X_i;D_{ki},\Delta_{ki}) = \prod_{i=1}^{d} \sigma(X_i - D_{ki} + \Delta_{ki})(1-\sigma(X_i - D_{ki} - \Delta_{ki})) \tag{2}$$

where σ is a typical sigmoidal function or its approximation, $D_{ki}-\Delta_{ki}$ position of its left boundary and $D_{ki}+\Delta_{ki}$ its right boundary. The steepness of these boundaries is determined by adaptive parameters T_{ki} used in computation of sigmoids. Factorable products of sigmoidal functions (2) are computed by single nodes of the FSM system. Factorability of the functions realised by the network nodes is important for efficiency. Linear combinations of these functions are capable of describing arbitrary mind objects, localised as well as unbounded (for large Δ_{ki} they behave in finite regions like sigmoidal functions). For maximum flexibility one could consider additional rotation and rescaling parameters $\mathbf{R}$. Using functions (2) with the input vector $\mathbf{RX}=\Sigma_j R_{ij} X_j$ leads to the representation of almost arbitrary convex densities using just one single node. The number of internal adaptive parameters is $\mathbf{R}$ rather large therefore we rely on the linear combination (1) for representation of complex densities. Products of sigmoidal pairs can approximate gaussian functions used in the Radial Basis Function (RBF) networks [3] but our tests indicate that sigmoidal product functions lead to much faster convergence in difficult classification problems. In Fig. 1 a comparison of the convergence using RBF and products of sigmoidal functions (2) in the well-known two-spiral benchmark problem (two dimensions, two classes, 97 data points in each class) is presented. In this case 100 gaussian nodes were used by the RBF network and the same number of nodes computing products of pairs of sigmoidal functions.

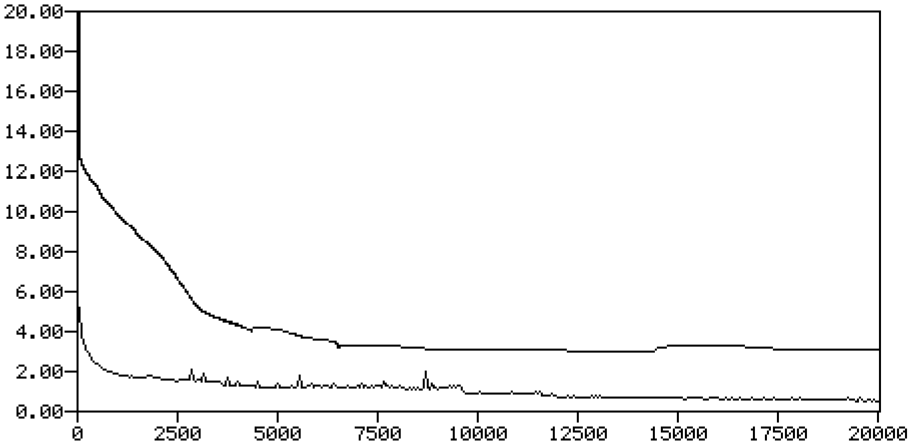

Fig. 1 Convergence of the RBF (upper curve) and products of sigmoidal pair functions (lower curve) on the two-spiral benchmark classification problem, with 194 data points for classification and 100 network nodes.

RECOGNITION. The weighted sum (1) computing the value of the *M* function is realised by the output node (Fig. 2). In contrast to most classification systems with separate outputs for each class FSM network has three outputs only: the value of the *M* function, the gradient of the *M* function and the alphanumerical output for the results of classification. Inputs **X** include not only values of all input features but also class index since class labels are represented in the same way as other features, as an additional dimension in the mind space. The confidence factors for each of the internal nodes are computed as:

$$p_k = \frac{[W_k\, s(\mathbf{X};\mathbf{D}_k;\Delta_k)]^2}{\Sigma_j\,[W_j\, s(\mathbf{X};\mathbf{D}_j;\Delta_j)]^2};\qquad \sum_k p_k = 1 \tag{3}$$

i.e. they are equal to the value of renormalized contribution of the node. After the recognition process is completed each node with the confidence factor larger than a fixed threshold writes the value of its confidence factor together with the message identifying the node (including the class this node is handling). Gradient method is used to find local maxima of the *M* function nearest to the input vector **X**. The input vector is changed along the gradient in a few large steps leading to $\mathbf{X}_{\max}$ corresponding to a local maximum of the *M* function. Calculation of gradients is done analytically at the same cost as calculation of the *M* function. Confidence factors should be adjusted for the value of the $M(\mathbf{X})$.

For queries **X** with large value of $M(\mathbf{X})$ the system is reliable and the vector $\mathbf{X}_{\max}$ at the local maximum is not too different from the input vector **X**. For small values of $M(\mathbf{X})$ local maximum may be far from the input vector and the answer may not be reliable. If the gradient is very small but the value of the *M* function is nonzero parameters Δ controlling the fuzziness of the representation of data are temporarily increased (with suitable shift of the **D** centres). This step is analogous to overgeneralisation of the knowledge used sometimes in the initial steps of the reasoning or classification. If the value of the *M* function is close to zero the system responds with “unknown data” answer but may be forced to make a guess using the overgeneralisation mechanism. This mechanism does not work if the mind objects lie in the space orthogonal to the input data, for example, if the system is trained with $X_1 = 0$ only, all factors $s_1(X_1\,;)$ may be removed from the node processing functions or the system will make $s_1(X_1\,;)$ factors sharply concentrated around zero. Instead of increasing fuzziness it is safer to find the nearest $\mathbf{D}_k - \Delta_{ki}$ and $\mathbf{D}_k + \Delta_k$ boundaries to determine objects that are nearest neighbours of **X**.

In some applications (for example in classification of chemical spectra) the input data may not correspond directly to any class known to the system, it may rather be a complex mixture of the data corresponding to two (or more) known classes (known spectra of molecular fragments). In such case FSM system will try to perform deconvolution of the input data into a sum of the mind objects that are near **X**.

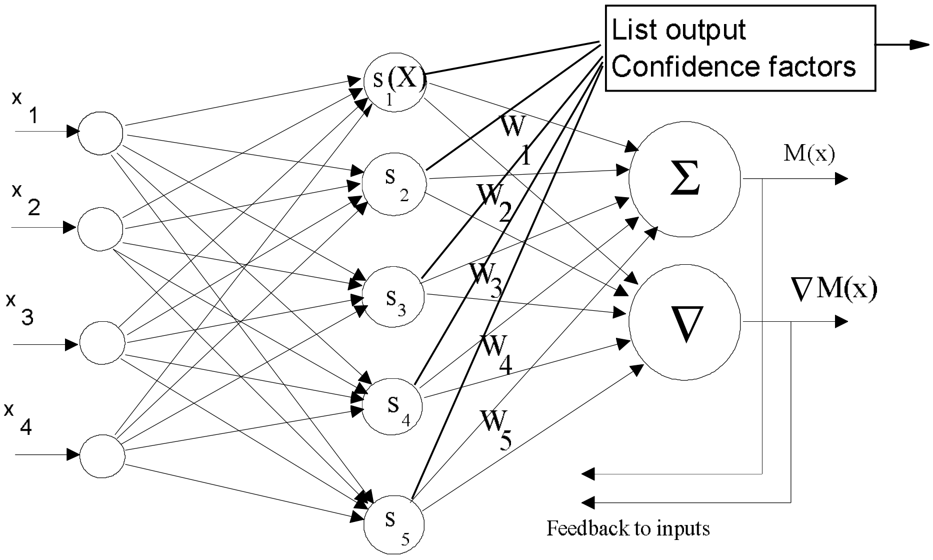

Fig. 2. Feature Space Mapping network module. The whole system is composed from several such modules specializing in classification of different type of data and contributing to the final system identification performed by the reasoning module.

If some values of the input data are not given - in particular the class value is usually not given in the recognition step since it should be determined - the system will make a classification in the subspace of the total mind space corresponding to the values given, find the $\mathbf{X}_{\max}$ in this subspace and perform the search for the unknown values X_u. The search is performed using only those nodes that contributed to classification, and consists of separate one-dimensional searches for the maximum value of the $s_u(X_u)$. The most probable classification is determined by taking into account the *a priori* probabilities of the relevant classes computed in the learning phase by the hidden nodes counting the number of the training data vectors belonging to a given node.

LEARNING or creation and modification of objects in the mind space proceeds using a constructive algorithm similar to the Resource Allocating Network (RAN) [4] introduced for function interpolation with the RBF type of networks [3]. A training vector **X** (including class label) that leads to the "unknown data" response and is sufficiently far from the limits of the nearest object of a proper class extending between $\mathbf{D}_k$ and $\mathbf{D}_k+\Delta_k$ requires a new node with $\mathbf{D}_{N+1}= \mathbf{X} -\Delta_{min}/2$, where Δ_{min} is the initial small spread of the function, related to the resolution of the data. If the data is not sufficiently far from the nearest mind object of the same class as **X** the object is modified by moving its boundaries in the direction of **X**. If the data **X** is misclassified the boundaries of the object it was assigned to are slightly shifted away from **X**, the nearest object of the proper class is found and shifted towards **X**. Changes are small to avoid disastrous consequences of errors in the training data.

Although mathematical treatment based on the complexity optimised data clustering [5], such as the entropy-constrained clustering, may lead to a more accurate representation of the data in the mind space it is also computationally more demanding and for the applications we have considered so far it should not be necessary. However, there is no reason why we could not introduce such methods at a later stage of FSM development.

REASONING. So far the system described performed only classification tasks and the mind space was identical to a simple feature space. For system identification this may be insufficient and the ability to reason and draw conclusions from various sources of information is desired. We would like to describe all cognitive processes in the mind space representation. In the more complex systems there are various sources of information. For example, for identification of chemical molecules chemists use several types of spectroscopic data as well as information about the colour of chemical substances, molecular weight or solvability. In our approach each of these different sources of information is analysed by a separate module of the FSM type, with explicit representation of the data in their local mind spaces. Each of these modules performs preliminary classification giving one or more results with some confidence levels. The last module integrates all these results and makes final classification. It has almost the same structure as other modules, except for taking into account the confidence factors in reasoning based on fuzzy evidential logic. It may be treated as a fuzzy logic version of the blackboard architecture used in artificial intelligence expert systems such as CRYSALIS system for crystallographical data analysis [6]. Network implementation of a typical expert system production rules: **if** (FACT 1. **and**. FACT2. **or**. FACT3...) **than** (FACT4) or more general fuzzy rules

$$\text{IF}\left(\left(x_1 \in X_1^{(1)} \wedge \ldots x_N \in X_N^{(1)}\right) \vee \left(x_1 \in X_1^{(2)} \wedge \ldots x_N \in X_N^{(2)}\right) \vee (\ldots)\right) \text{ THEN} (y_1 \in Y_1 \wedge \ldots \wedge y_M \in Y_M)$$

is easy. Each of the terms in parenthesis is realised by one of the nodes and confidence factors are treated as additional dimensions in the mind space, i.e. they appear as variables in each group. Search is done by gradient techniques or one-dimensional searches focusing on single variable, with the depth of search equal to the number of unknown features [2].

Characteristics of the FSM system include: symbolic interpretation, neural realisation; direct modelling of the mind objects with multidimensional function densities; full control over associations and generalisations of the system; supervised and unsupervised learning; fine tuning of object representations for pattern recognition; learning from examples and from general laws; adding and removing mind objects (network nodes); formation of hierarchies of categories and metaconcepts; multi-scale approach (focusing and defocusing) to concentration on relevant parts of the mind space; hierarchical modular construction with the reasoning module at the highest level; and the scaling of the complexity of the system linear with the number of objects, making FSM ideal for parallel processing.

Simplifications afforded by formation of **topographical maps** may also be exploited. Since it is hard to imagine relations among mind objects in more than three dimensions the system maps all objects into two-dimensional space trying to preserve metrical relations by minimising the measure of the topography preservation [7]

$$D_1(\mathbf{r}) = \sum_{i>j}^{k} (R_{ij} - r_{ij})^2$$

where R_{ij} are distances between mind objects i and j in the original mind space and r_{ij} are distances in the two-dimensional target space. This minimisation (subject to additional conditions due to the lower number of degrees of freedom in the two-dimensional space) performed using simulated annealing gives more accurate visualisation of the data than Kohonen networks or other self-organising unsupervised learning networks and is by itself useful for preliminary classification.

Summarising, FSM is an universal neurofuzzy system, based on the concept of the mind space inspired by cognitive modelling, useful in many neural network and expert system types of applications. At present we are developing the software primarily for identification of chemical molecules, stellar spectra and identification of personality types from the analysis of psychometric data.

REFERENCES

[1] W. Duch, *A solution to the fundamental problems of cognitive sciences*, UMK-KMK-TR 1/94 (1994)

[2] W. Duch and G.H.F. Diercksen, *Feature Space Mapping as a Universal Adaptive System*, Comp. Phys. Commun. (in print, May 1995 issue)

[3] T. Poggio, F. Girosi, Proc. of the IEEE **78** (1990) 1481

[4] J. Platt, Neural Computation **3** (1991) 213

[5] J. Buhmann, H. Kühnel, Neural Computation **5** (1993) 75-88

[6] R.E. Engelmore, A. Terry , IJCAI **6** (1979) 250-256

[7] W. Duch, Open Systems and Information Dynamics **2** (1995) 295-302