FEATURE SPACE MAPPING NETWORK FOR CLASSIFICATION

Włodzisław Duch and Rafał Adamczak[1]
Department of Computer Methods,
Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland

The Feature Space Mapping (FSM) network is based on multidimensional separable localized or delocalized functions centered around data clusters. The network learns by memorizing the noisy training data in the input (feature) space and classifies new data by searching for the nearby memory traces. Preliminary results for several classification problems are given.

1. INTRODUCTION

The development of the Feature Space Mapping (FSM) system has been motivated by our conviction that neurodynamical models of the brain are very difficult, it is not clear how to model larger groups of neurons in a realistic way and simple neural networks are not well suited for cognitive modeling. Mind arises from a complex dynamics of the modular and hierarchical brain structures. Approximations to this dynamics lead to a set of concepts [1] helpful in description of the mind, such as the concepts of mind events taking place in feature spaces, called at the highest level of hierarchy "conceptual spaces" or "mind spaces". Among many other aspects mind models should be capable of recognition, classification and reasoning. Cognitive modeling may not always be faithful to neurobiology, but tries to preserve essential functions simplifying the underlying structures and dynamics. Neurofuzzy networks are natural implementations of such models.

In this paper we present one particular realization of general cognitive modeling ideas [1] and apply the resulting neurofuzzy network to classification problems. We will assume that the incoming signals $\mathbf{I}(t)$ are subject to preprocessing (accomplished in the brain by various topographic maps and population coding mechanisms) that defines features of internal representations $\mathbf{X}_i(t)$ suitable for classification. A coordinate system based on these features $\{\mathbf{X}_i\}$ defines a multidimensional feature space. Since these features may be of different types (for example, coming from different sensory modalities) partial classifications are performed in local feature spaces while the final classification is performed in the space that is higher in the hierarchy, called here "the mind space". The system learns by creating and modifying mind (or memory) objects in this space. They are described using mind (or memory) function $M(\mathbf{X})$ as the 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.

---

[1] e-mail: duch@phys.uni.torun.pl, WWW: http://www.phys.uni.torun.pl,
archive ftp.phys.uni..torun.pl/pub/kmk/papers

The questions to be solved are: how to select optimal functions representing the data, how to introduce fuzziness of the data and how to adapt the resulting network.

2. NETWORK STRUCTURE AND ADAPTATION RULES

Feature detectors react to specific localized features of the incoming data, therefore objects in the mind space are modeled via localized functions rather then unbounded, sigmoidal functions that are most frequently used in neural models. Gaussians are the most common choice of localized functions, leading to the RBF type of networks [3] based on the regularization theory. Unbounded functions are also sometimes useful. From the point of view of efficiency and flexibility of the system good choice is provided by the bi-radial functions [2]:

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

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

where σ is a typical sigmoidal function or its approximation, D_{ki} -D_{ki} position of its left boundary and D_{ki}+D_{ki} its right boundary. The steepness of these boundaries may be determined by adaptive parameters T_{ki} used in computation of sigmoids while the degree of localization along dimension i by the parameter l_i. Factorable products of sigmoidal functions (2) are computed by single nodes of the FSM system. Factorability of the functions realized by the network nodes is important for efficiency. Linear combinations of these functions are capable of describing arbitrary mind objects, localized (l_i=1) as well as delocalized (l_i≠1). For maximum flexibility one could consider additional rotation and rescaling parameters **R** but the number of internal adaptive parameters would then be rather large. Products of sigmoidal pairs can approximate gaussian functions used in the Radial Basis Function (RBF) networks [3] but our tests indicate that biradial functions are better choice in some difficult classification problems [4].

LEARNING or creation and modification of objects in the local feature spaces and the final mind space proceeds using a constructive algorithm similar to the Resource Allocating Network (RAN) [5] 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 $\mathbf{D}_k$+$\mathbf{D}_k$ requires a new node with $\mathbf{D}_{N+1}$= **X** - $\mathbf{D}_{min}$/2, where $\mathbf{D}_{min}$ 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. Initially the admissible errors and minimal distances are set to a rather large values to locate the main data clusters and in the next step higher resolution representation is created.

Although mathematical treatment based on the complexity optimized data clustering [6], such as the entropy-constrained clustering, may lead to a more accurate representation of

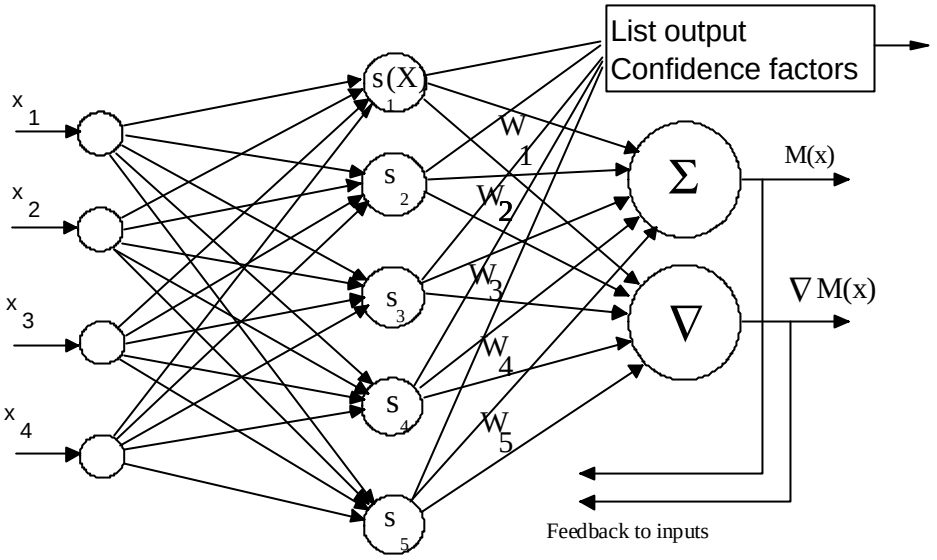

Fig. 1. Single Feature Space Mapping network module.

the data in the feature spaces it is also computationally more demanding. We plan to explore such methods at a later stage of FSM development, together with network pruning techniques based on the spontaneous memory decay. Another direction that will be explored is the use of maximum likelihood estimates instead of the simple quadratic error minimization. The conditional probability $p(Y/X)$ of obtaining the desired answer Y for input X is

$$p(Y/X) = \sum_i c_i p(Y|X, s_i) \tag{3}$$

where c_i are confidence factors estimated by the network for each node and $p(Y/X,s_i)$ are conditional probabilities of obtaining the desired answer Y for input X when the node s_i is selected. In many cases the associative Gaussian mixture model for these probabilities may be assumed

$$p(Y|X, s_i) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\|Y - s_i(X)\|^2} \tag{4}$$

and maximization of conditional likelihood performed in respect to the adaptive parameters of the nodes s and the weights W. For binary classification Bernoulli mixture model instead of Gaussian mixture may is better [7].

RECOGNITION. The weighted sum (1) computing the value of the M function is realized by the output node (Fig. 1). In contrast to most classification systems with separate outputs for each class FSM network has only three outputs: the value of the M function, the gradient of the M function and the alphanumerical output for the results of classification. Inputs $\mathbf{X}$ 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 feature space. The confidence factors for each of the internal nodes are computed as:

$$p_k = \frac{[W_k\, s(\mathbf{X}; \mathbf{D}_k; \mathbf{D}_k)]^2}{\Sigma_i [W_i\, s(\mathbf{X}; \mathbf{D}_i; \mathbf{D}_i)]^2}; \quad \sum_k p_k = 1 \tag{5}$$

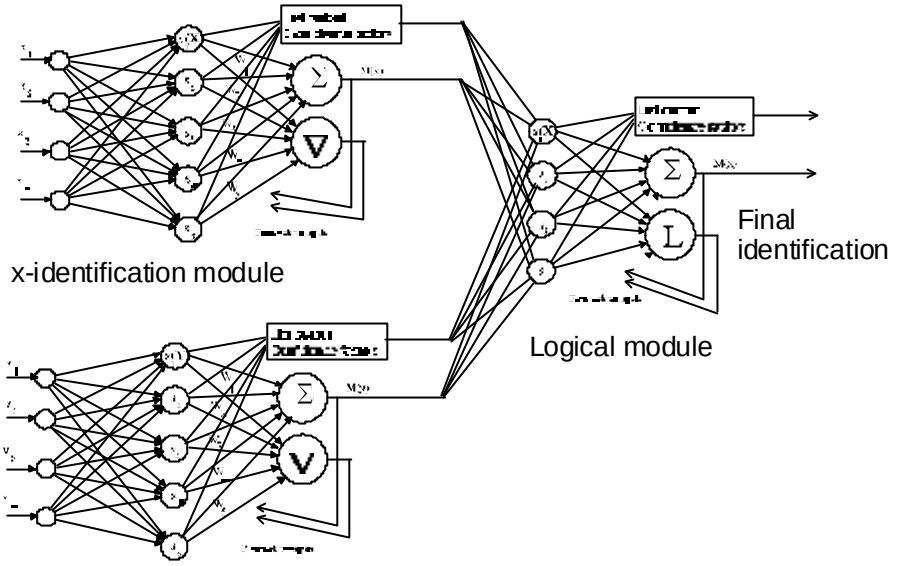

Fig. 2. Two recognition and one reasoning module of the FSM system. Recognition modules specialize in classification of different type of data and contribute to the final classification performed by the reasoning module.

i.e. they are equal to the value of renormalized contribution of the node. Another choice is given by the softmax transformation:

$$p_k = \frac{\exp\left[W_k\ s(\mathbf{X};\mathbf{D}_k;\mathbf{D}_k)\right]}{\Sigma_j \exp\left[W_j\ s(\mathbf{X};\mathbf{D}_j;\mathbf{D}_j)\right]};\quad \sum_k p_k = 1 \tag{6}$$

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 $\mathbf{X}$ (in contrast to the backpropagation techniques we are interested only in local maxima). 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 $\mathbf{X}$ 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 $\mathbf{X}$. 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 non-zero parameters $\mathbf{D}$ controlling the fuzziness of the representation of data are temporarily increased (with suitable shift of the $\mathbf{D}$ centers). This step is analogous to overgeneralization 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 overgeneralization 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 $\mathbf{D}_k - \boldsymbol{D}_{ki}$ and $\mathbf{D}_k + \mathbf{D}_k$ boundaries to determine objects that are nearest neighbors of **X**.

REASONING. So far the system described performed only classification tasks and the mind space was identical to a simple feature space. For complex categorization or classification tasks this may be insufficient and the ability to reason and draw conclusions from various sources of information is desired. For example, for classification of chemical molecules chemists use several types of spectroscopic data as well as information about the color of chemical substances, molecular weight or solvability. In FSM approach each of these different sources of information is analyzed by a separate module, with explicit representation of the training data in local feature spaces and final classification performed using the reasoning module. Each of these modules performs preliminary classification giving one or more results with some confidence levels. The reasoning module integrates all these results in the mind space 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 [8]. Network implementation of a typical expert system production rules:

if (FACT 1. **and**. FACT2. **or**. FACT3...) **than** (FACT4) (7)

or more general fuzzy rules

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

is quite natural in the FSM system [2]. Each of the terms in parenthesis is realized 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]. Finally it is useful to visualize relations among the multidimensional mind objects. This is done using Kohonen feature maps and multidimensional scaling techniques [9].

3. RESULTS.

The FSM system is still under development so the results reported below should be considered as preliminary. All classifications reported here were based on a single module of FSM (simple feature spaces). The two-spiral benchmark (two classes, 196 training points) is quickly and accurately solved with less than 100 biradial nodes. Two standard data sets for classification were obtained from ftp://ftp.ics.uci.edu/pub/machine-learning-databases. The first is the well known Iris data set. Since during learning vectors were presented randomly all trials were repeated ten times and the average accuracy of classification is reported.

The Iris data file is composed of 150 vectors distributed in three classes called iris-setosa, iris-versicolor and iris-virginica. Each vector has 4 features: sepal length and width and petal length and width (all in cm). This data set was divided into training and test vectors. An average percent of correct classifications was 90.5%. The ionosphere data has 200 training and 150 test vectors, each vector with 34 attributes. Vectors are divided in two classes: good and bad. We have obtained 92.5% accuracy during recognition. This result is slightly better

than obtained by the nearest neighbor method and by the non-linear perceptron, similar too RBF but worse than 96% accuracy obtained by backpropagation network, indicating that it is worthwhile to try also the delocalized biradial functions with $l_i \neq 1$.

The data for classification of galaxies were obtained from the authors [10]. This is a large data set (more than 5000 galaxies). The network was trained on 1700 ESO-LV galaxies, and tested on the remaining 3517 galaxies. Each vector had 32 features (for details see [10]). The training set was divided in two classes: Early_type and Late_type galaxies. Backpropagation algorithm gave for this data file 90% agreement with human experts classifying galaxies [10]. We have obtained similar accuracy at the 88% level (with much faster training times), again indicating that it is worthwhile to use delocalized functions with $l_i \neq 1$.

Acknowledgment: Support by the Polish Committee for Scientific Research, grant 8T11F 00308, is gratefully acknowledged.

REFERENCES

[1] W. Duch, *A solution to the fundamental problems of cognitive sciences*, UMK-KMK-TR 1/94 (1994); W. Duch, *On simplifying brain functions,* this volume.

[2] W. Duch, G.H.F. Diercksen, *Feature Space Mapping as a universal adaptive system.* Computer Physics Communications **87** (1995) 341-371; W. Duch, Neural Network World **4** (1994) 645-654

[3] T. Poggio and F. Girosi, *Networks for approximation and learning*. Proc. of the IEEE 78 (1990) 1481

[4] N. Jankowski and W. Duch, this volume; W. Duch, R. Adamczak, N. Jankowski and A. Naud, *Feature Space Mapping: a neurofuzzy network for system identification,* Proc. of Engineering Applications of Neural Networks, Helsinki 1995, pp. 221-224

[5] J. Platt, *A resource-allocating network for function interpolation.* Neural Comput. 3 (1991) 213

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

[7] M. Jordan, R. A. Jacobs, *Hierarchical Mixtures of Experts and the EM Algorithm,* Neural Computations 6 (1994) 181-214

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

[9] W. Duch, Open Systems and Information Dynamics **2** (1995) 295-302; A. Naud and W. Duch, this volume.

[10] O. Lahav, A. Naim, L.Sodre Jr. and M. C. Storrie-Lombardi, *Neural computation as a tool for galaxy classification: methods and examples,* Institute of Astronomy, Cambridge, Technical report CB3 OHA (1995)