Evolving Connectionist Systems for On-Line Pattern Classification of Multimedia Data

[1]NIKOLA KASABOV, [1]IRENA KOPRINSKA, [2]GEORGI ILIEV
[1]Department of Information Science
University of Otago
NEW ZEALAND
[2]Department of Telecommunications
University of Sofia
BULGARIA
gli@tu-sofia.bg

Abstract: - The paper develops further the concept of evolving connectionist systems, and one particular model – evolving fuzzy neural networks, that are applied on pattern classification tasks of multimedia data. The evolving systems learn in an on-line, life-long learning mode and adapt to the new data. This mode is crucial when the system is required to adapt quickly to new data and be able to generalize immediately afterwards. These tasks are typical for processing of multimedia data, such as adaptive speech recognition and adaptive video data processing that are presented in the paper.

Key-Words: - Evolving connectionist systems, on-line learning, pattern classification, speech recognition, camera operation recognition

1 Introduction

Some pattern classification tasks are characterized by:

- small data sets;
- static data sets, that are used once to create a model, which does not change in time;
- no need for rapid on-line adaptation to continuously incoming data.

For these tasks the traditional statistical and AI techniques are well suited. Such methods for data analysis and feature extraction are: principle component analysis (PCA); correlation analysis; off-line clustering techniques such as k-means, fuzzy C-means; self-organizing maps (SOM), and many more. Many modeling techniques are applicable for such tasks, for example: statistical techniques – such as regression analysis, support vector machines; AI techniques – such as decision trees, hidden Markov models, finite automata; and neural network techniques – such as multilayer perceptrons (MLP), learning vector quantization (LVQ), fuzzy neural networks (FuNN). Some of the modeling techniques allow for extracting knowledge – e.g. rules from the models that can be used for explanation, or new knowledge discoveries.

Unfortunately, some of the pattern recognition tasks especially in the area of multimedia data processing are characterized by:

- large data sets that are updated regularly;
- a need for on-line learning and new models creation from input data streams;
- learning from data streams that change their dynamics in time.

There are only few techniques that can deal with such problems [1], [2], [3]. One computational modeling paradigm called ECOS (evolving connectionist systems) was introduced in [4] to deal with the above issues. ECOS are dynamically growing (and shrinking) modular neural networks for on-line learning and adaptation from an input-output stream of data. A particular implementation of ECOS is the Evolving Fuzzy Neural Networks [5].

The paper develops further the concept of EFuNNs and applies EFuNNs on a variety of pattern classification tasks.

2 Evolving Connectionist Systems and Evolving Fuzzy Neural Networks

Evolving connectionist systems are systems that evolve their structure and functionality over time through interaction with the environment [4]. They have some ('genetically') predefined parameters (knowledge) but they also learn and adapt as they operate. They emerge, evolve, develop, unfold through learning, and through changing their structure in organization to better represent incoming data. ECOS are multi-level, multi-modular structures where many neural network modules (denoted as NNM) are connected with inter-, and intra-connections.

Fuzzy neural networks are connectionist structures that can be interpreted in the concepts of fuzzy rules [6], [7]. EFuNN is an on-line learning fuzzy neural network [4], [8]. The EFuNN architecture and functionality are further developed here, illustrated and analyzed in details.

EFuNNs are general purpose fuzzy neural networks that have a five-layer structure (Figure 1). The input layer represents crisp input variables. The fuzzy input neurons stand for the fuzzy quantification of input variables using membership functions. The rule nodes evolve through learning and represent prototypes of data mapping between the fuzzy input and fuzzy output spaces. Each rule node is defined by two weight vectors: W1 and W2. The former is adjusted via unsupervised learning based on similarity between the fuzzy input and the prototypes already stored. W2 is updated by Least Mean Squares (LMS) algorithm to minimize the fuzzy output error. The fuzzy output neurons stand for the fuzzy quantization of the output variables while the output nodes represent the real output values.

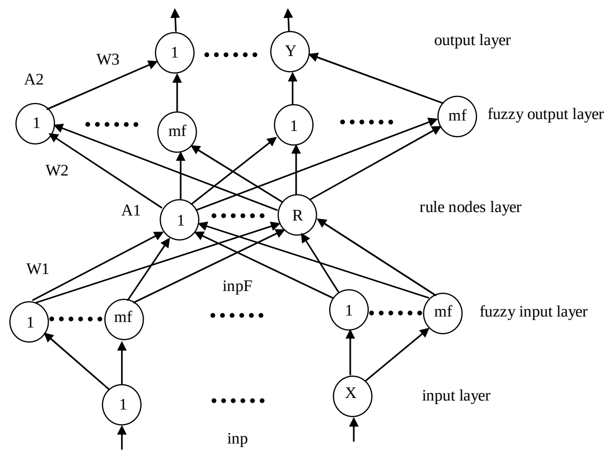

Fig.1. EFuNN's architecture.

There are several options for EFuNN growing [5], [8]. We use the 1-of-n method. In the next two sections EFuNN is applied on two tasks for multimedia data classification.

3 On-Line Adaptive Speech Recognition

The task is recognition of speaker independent pronunciations of Italian digits. The digits are from the SPK database, collected at ITC-Irst, Trento, Italy [9]. 30 speakers (15 male and 15 female) for training and other 30 (15 male and 15 female) for testing, totally 6000 training and 6000 testing examples. For full details about this database see Ref. [9]. We use 8 Mel Frequency Scaled Cepstral Coefficients (MFSCC) and log-energy as acoustic features.

Clean speech data are used for training. Noise is introduced to the clean speech test data to evaluate the behavior of the recognition systems in a noisy environment. Two different experiments are conducted. In the first instance, car noise is added to the clean speech. In the second instance office noise is introduced over the clean signal.

In Table 1 the results of four recognition systems are summarized. Namely, we have compared the performance of LVQ [10], EFuNN, Continuous Density Hidden Markov Model (CDHMM), and a recently developed model – Segmental Neural Network with Trainable Amplitude of activation functions [9] (SNN-TA) (for details about CDHMM and SNN-TA see ref. [9]). Comparing the results at SNR=0 dB, which case is of particular interest for the present application, it can be seen that SNN-TA and EFuNN models have higher WRR.

Table 1. Italian digits – word recognition rate (WRR) of four speech recognition systems: LVQ – codebook vectors – 396, training iterations – 15840, EFuNN – 3MF, rule nodes – 139.

Model	WRR(%)			
	Car noise (dB)		Office noise (dB)	
	SNR=0	SNR=18	SNR=0	SNR=18
LVQ	77.43	94.75	13.83	81.45
CDHMM	45.08	97.69	8.90	71.18
SNN-TA	85.47	96.15	79.35	87.62
EFuNN	84.98	91.57	77.59	85.69

4 On-Line Adaptive Camera Operation Recognition

Camera operations recognition is an important issue in content-based organization of video databases. At the stage of video parsing, it is essential to distinguish gradual shot transitions from the false positives due to camera operations. It is also needed at the step of video indexing and retrieval as camera operations reflect how the attention of the viewer should be directed and this information can be used for index extraction and key frame selection. With the widespread use of MPEG [11], methods for camera operations recognition that process directly the compressed stream were proposed. The majority of them are based on motion vector (MV) pattern

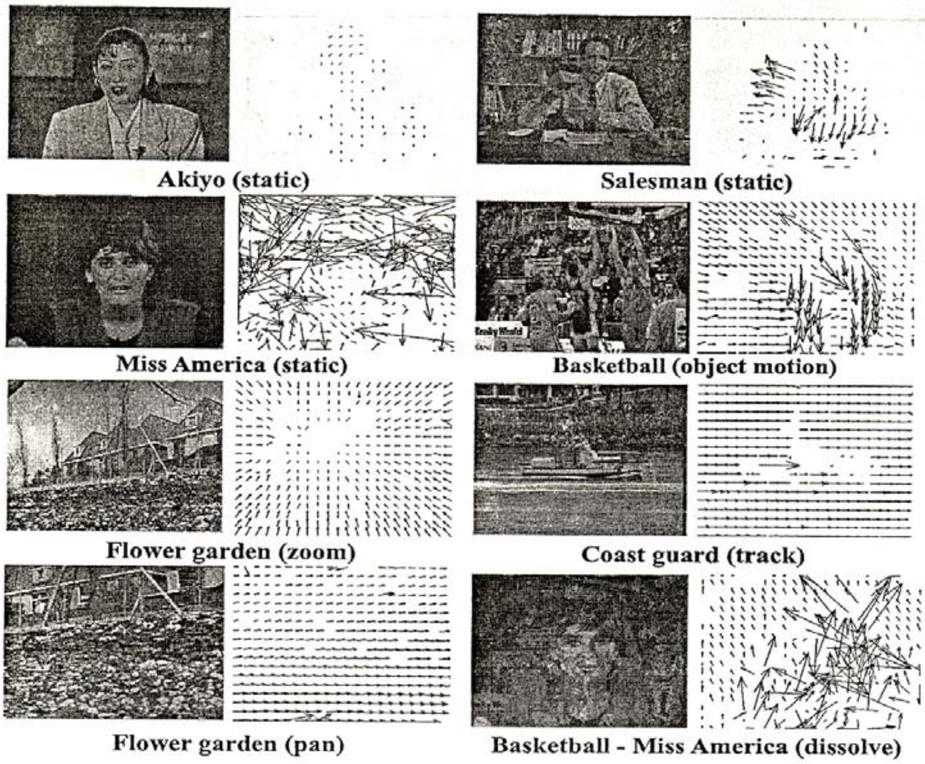

Fig.2. MV patterns corresponding to the 6 classes.

analysis. Zhang et al. [12] compute measures based on the MV direction and then use thresholds to recognize pan/tilt and zoom. The disadvantage of the manual threshold tuning is eliminated by the application of decision trees in [13] and LVQ in [14]. LVQ, though, like most of the static neural networks employing supervised learning, has predefined topology, requires multiple passes on the training set and suffers from "catastrophic forgetting", i.e. is not capable to accommodate new data without retraining on both the original and new data. In this section we study the potential of EFuNNs for camera operation recognition.

4.1 Data and Classes

Following [14] we consider six classes: static, panning, zooming, object motion tracking and dissolve. While four of these classes are camera operations, object motion and dissolve are added as they introduce false positives. Each of the classes is characterized by a specific pattern in the field of MVs of P and B frames in a MPEG encoded sequence, Fig. 2. The well-known benchmark sequences Akiyo, Salesman, Miss America, Basketball, Football, Tennis, Flower Garden, and Coastguard were used in our experiments. Several aspects of data complicate learning. As it cam be seen from Fig.2, the three static examples have rather different MV fields. While the frames of Akiyo can be viewed as ideal static images, there are occasionally sharp movements in Salesman. The MV field of Miss America is completely different as the encoder we used [11] generates MVs with random orientation for the homogeneous background. Hence, it will be advantageous to use a classification system capable incrementally to adapt to new representatives of the static class without the need to retrain the network on the originally used data.

4.2 Feature Selection

Data pre-processing and feature extraction were done as in [14]. MVs of P and B frames are extracted and smoothed by a vector median filter. Based on them, a 22-dimentional feature vector is created for each frame. The first component is a measure for how static the frame is. It is calculated as the fraction of zero MVs using both the forward and backward MV components. The forward MV area is then sub-divided in 7 vertical strips, for which the average and standart deviation of MV directions and the average of MV magnitudes are computed. In order to build the EFuNN classifier, the MV patterns of 1200 P and B frames (200 for each class), have been visually examined and manually labeled.

4.3 Experimental Results and Discussion

We run several experiments varying the number of membership functions and distance type. The best results we obtained using: 1) two fuzzy membership functions and fuzzy distance measure (labeled as EFuNN1) and 2) Eucledian distance and applying no fuzzification (labeled as EFuNN2).

For evaluation of the EFuNN classification results we use 10-fold cross validation. The EFuNN performance is compared with seven other classifiers: baseline, 1-rule, k-nearest-neighbor, naïve Bayes, decision table, decision trees and learning vector quantization.

Baseline simply predicts the majority class in the training data. It determines the baseline performance that can be used as a benchmark for the other classifiers. If a learning algorithm performs worse than Baseline, this would be an indication for serious overtraining.

1-rule was introduced by Holte [15]. It generates a one-level decision tree, expressed as a set of rules testing only attribute. Holte demonstrated that such rules give surprisingly good results and are practical alternative to more complex classifiers.

The nearest neighbor [16] algorithm stores all training examples and the unknown example is classified as belonging to the majority class of the closest k-training examples based on a distance measure. We use k=1 and Eucledian distance metric. Naïve Bayes [25] is based on the Bayes's rule of conditional probability assuming that all attributes are equally important and independent.

Decision table (DTab) [17] are built by selecting the best sub-set of attributes by measuring the table's cross-validation error for different subsets.

Decision trees (DT) [18] are the most popular and highly developed techniques for supervised learning. At each step the most informative attribute is selected and placed at the root and a branch is created for each possible attribute value. The process is repeated recursively for the branches using the instances that reached that branch until all instances at a node belong to the same class. Unseen examples are propagated from the root down the leaves and accordingly classified.

Learning vector quantization LVQ [10] creates a few prototypes for each class, adjust their positions by learning and then classified unseen examples by the nearest-neighbor principle.

For baseline, 1-rule, kNN, naïve Bayes, decision table and decision trees we used the implementations provided in WEKA [19]; for the LVQ algorithm – LVQPack [20]. The classification accuracy on training and testing sets is summarized in Table 2.

Table 2. Classification accuracy [%] on training and testing set

Accuracy [%]	Training set	Testing set
Baseline	17.121 ± 0.3	12.58 ± 2.2
1-rule	68.68 ± 0.6	62.00 ± 4.1
kNN	100 ± 0	95.32 ± 1.8
Naïve Bayes	86.07 ± 0.3	85.58 ± 2.5
DTab	91.64 ± 3.7	81.42 ± 3.7
DT	98.78 ± 0.32	92.33 ± 2.7
LVQ	85.4 ± 2.5	85.83 ± 2.2
EFuNN1	91.16 ± 0.87	86.27 ± 3.13
EFuNN2	92.58 ± 0.76	86.42 ± 3.9

The classification performance of EfuNN1 and EFuNN2 compares favorably with 1-rule, Naïve Bayes, decision table and LVQ algorithms. The nearest neighbor algorithm achieves the best results but at the highest computational price: it stores all 1080 training examples and compares each of them with the unseen example. The accuracy of DTs is with 6 % higher than those of the EFuNNs algorithms. However, as it can be seen from Table 3, the DT algorithm generates complex trees with many nodes and leaves. The classification results of EFuNN1 and EFuNN2 are comparable but the application of the fuzzification in EFuNN1 implies higher number of nodes and, hence, higher computational complexity of the training algorithm. As a result, the learning speed slows down. The LVQ network is much smaller than the EFuNN1 architectures but comparable with EFuNN2 (Table 4). It should be also noted that EFuNN requires only 1 epoch for training in contrast to the LVQ's multi pass learning that needed 1520 epochs in our case study.

Table 3. Complexity of 1-rule and DT (number of nodes and leaves) and kNN (number of stored examples)

1-rule	Nodes: 1 ± 0 Leaves: 27.2 ± 1.93
DT	Nodes: 76 ± 5.52 Leaves: 38.5 ± 2.76
kNN	Prototypes: 1080 ± 0

We also studied how EFuNN classifies the individual classes. It was found that while zoom and pan are easily identified, the recognition of object movement, tracking and dissolve is more difficult. A detailed analysis indicates that the algorithm actually has difficulties to discriminate well between these three classes. Despite the fact that the MV fields of Miss America were not typical for static videos and complicated learning, they learned incrementally and classified correctly by EFuNN.

Table 4. Complexity of the neural networks (number of hidden neurons)

LVQ	Codebook nodes: 38 ± 0
EFuNN1	Fuzzy input nodes: 44 ± 0 Rule nodes: 33.3 ± 4.32 Fuzzy output nodes: 12 ± 0
EFuNN2	Rule nodes: 43.4 ± 2.91

5 Conclusion

The paper presents a methodology for on-line, adaptive learning and classification of patterns from multimedia data. Two case studies are used – adaptive speech recognition and adaptive camera operation recognition.

The paper demonstrates that this methodology can be successfully applied to on-line pattern recognition from fast streams of multimedia data. It produces high classification accuracy, learns quickly and does not suffer from catastrophic forgetting.

References:

[1] E. Blanzieri, P. Katenkamp, Learning radial basis function networks on-line, *Proc. of Intern. Conf. On Machine Learning*, 1996, pp. 37-35.

[2] G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, D. Rosen, FuzzyARTMAP: A neural network architecture for incremental supervised learning of analog multi-dimensional maps, *IEEE Trans. of Neural Networks*, Vol. 3, No 5, 1991, pp. 698-713.

[3] J. Freeman, D. Saad, On-line learning inradial basis function networks, *Neural Computation*, Vol. 9, No 7, 1997, pp. 218-233.

[4] N. Kasabov, ECOS: A framework for evolving connectionist systems and the ECO learning paradigm, *Proc. of ICONIP'98*, Kitakyushu, Japan, Oct. 1998, pp. 1222-1235.

[5] N. Kasabov, Evolving fuzzy neural networks – algorithms, applications and biological motivation, *in Yamakawa and Matsumoto (eds), Methodologies for the Conception, Design and Application of Soft Computing*, World Scientific, 1998, pp. 271-274.

[6] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, MIT Press, 1996.

[7] N. Kasabov, N. Kim, M. Watts, A. Gray, FuNN/2 – A fuzzy neural network architecture for adaptive learning and knowledge acquisition, *Information Sciences*, 1997, pp. 155-175.

[8] N. Kasabov, Evolving connectionist and fuzzy connectionist systems – theory and applications for adaptive, on-line intelligent systems, *in Neuro-Fuzzy Techniques for Intelligent Information Processing*, Physica Verlag, 1999, pp. 111-114.

[9] E. Trentin, M. Matassoni, Robust segmental-connectionist learning for recognition of noisy speech, *Proc. of Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, May 1999, pp. 159-162.

[10] T. Kohonen, *Self-Organizing Maps*, Springer verlag, 1997.

[11] http://www.mpeg.org/MPEG/MSSG/VMPEG

[12] H. Zhang, C. Low, S. Smoliar, Video parsing and browsing using compressed data, *Multimedia Tools & Applications*, 1995, pp. 89-111.

[13] N. Patel, I. Sethi, Video shot detection and characterization for video databases, *Pattern Recognition*, 1997, pp. 583-592.

[14] I. Koprinska, S. Carrato, Video segmentation of MPEG compressed data, *Proc. of ICECS'98*, Lisbon, Portugal, 1998, pp. 243-246.

[15] R. Holte, Very simple classification rules perform well on most commonly used databases, *Machine Learning*, 1993, pp. 63-91.

[16] R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, John Wiley, 1973.

[17] R. Kohavi, The power of decision trees, *Proc. of European Conf. on Machine Learning*, 1995, pp. 252-259.

[18] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[19] I. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 2000.

[20] http://nucleus.hut.fi/nnrc