Mimicking adaptation processes in the human brain with neural network retraining

Lori Malatesta, Amaryllis Raouzaiou, George Caridakis, Kostas Karpouzis
Image, Video and Multimedia Systems Laboratory, National Technical University of Athens,
9, Heroon Politechniou str., Zografou 15780, Greece
{lori, araouz, gcari, kkarpou}@image.ece.ntua.gr

Abstract. Human brain processes undergo cycles of adaptation in order to meet the requirements of novel conditions. In affective state recognition, brain processes tend to adapt to new subjects as well as environmental changes. By using adaptive neural network architectures and by collecting and analysing data from specific environments we present an effective approach in mimicking these processes and modelling the way the need for adaptation is detected as well as the actual adaptation. Video sequences of subjects displaying emotions are used as data for our classifier. Facial expressions and body gestures are used as system input and system output quality is monitored in order to identify when retraining is required. This architecture can be used as an automatic analyzer of human affective feedback in human computer interaction applications.

1 Introduction

The ability to detect and understand affective states and other social signals of someone with whom we are communicating is the core of social and emotional intelligence and relies upon finely tuned neural mechanisms in the brain. This kind of intelligence is a facet of human intelligence that has been argued to be indispensable and even the most important for a successful social life 2. Neuropsychological (8) and neuroimaging data (9) with humans have suggested that recognition of some distinct facial expressions engages specific neural circuits. Although various brain regions have therefore been correlated with facial expression recognition, the nature of their contributions remains unresolved. The act of seeing is so effortless that it is difficult to appreciate the sophisticated mechanisms underlying it. However, current computing technology does not account for the fact that human-human communication is always socially situated and that discussions are not just facts but part of a larger social interplay. Not all computers will need social and

emotional intelligence and none will need all of the related skills humans have. Yet, human-machine interactive systems capable of sensing stress, inattention, confusion, and heedfulness, and capable of adapting and responding to these affective states of users are likely to be perceived as more natural, efficacious, and trustworthy (see 6).
Regarding personalized expressivity, it is well known (see, for example, recent results, on emotional signs from signals, of the Humaine network of Excellence 4) that in human computer interaction, the emotional characteristics and signs of signals captured from a specific user, although adhering to some general descriptive theories and psychological models, differ, sometimes significantly, between different persons. Thus, emotion recognition is a research problem, the solution of which highly depends on individual human characteristics and way of behaviour. Emotion recognition systems are generally based on a rule base system, or on a system that has learnt to solve the problem through extensive training. In either case, if such a system is to be used in a real life experiment, it further needs to take into account, i.e., to adapt its knowledge to the specific user characteristics as well as behavioural and environmental conditions, i.e., the context of interaction.
In all cases, it is essential that systems are derived which are able to adapt their performance to environmental changes, by detecting deterioration of their performance, and refining it with data obtained by the specific environment and respective cues provided by the user or by cross-correlating different modalities. Neural networks fit well with this requirement, since adaptation is their main advantage when compared with knowledge-based systems, where updating of knowledge is a complex, generally off-line procedure. Both supervised, such as multilayered feed-forward networks, and unsupervised networks, such as SOM or k-NN based approaches can be used for this purpose. In the rest of the paper an adaptive supervised feed-forward network is described and used for human computer interaction enriched with emotion analysis capabilities, showing that it can provide an effective approach to handling of the above described problems. The basic methodology can be extended to unsupervised, clustering techniques.
Section 0 describes the adaptive network architecture, while its use in different contexts is presented in section 0. An experimental study, with emotion datasets showing, not only extreme emotions, but also intermediate real-life ones, generated in the framework of the EC IST Humaine Network of Excellence, is given in section 0, while conclusions and further work are discussed in section 0.

1.1 Neural Architectures for Emotion Recognition

Taylor and Fragopanagos describe a neural network architecture in 7 in which features, from various modalities, that correlate with the user's emotional state are fed to a hidden layer, representing the emotional content of the input message. The output is a label of this state. Attention acts as a feedback modulation onto the feature inputs, so as to amplify or inhibit the various feature inputs, as they are or are not useful for the emotional state detection. The basic architecture is thus based on a feed-forward neural network, but with the addition of a feedback layer (IMC in **Error! Reference source not found.**), modulating the activity in the inputs to the hidden layer.

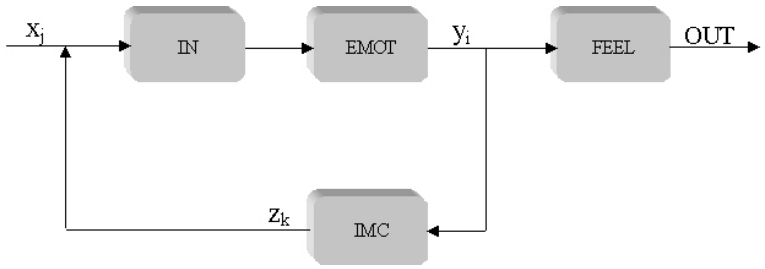

Fig. 1. Information flow in the system. IMC= inverse model controller; EMOT = hidden layer emotional state; FEEL = output state emotion classifier

Results have been presented for the success levels of the trained neural system based on a multimodal database, including time series streams of text (from an emotional dictionary), prosodic features (as determined by a prosodic speech feature extraction) and facial features (facial animation parameters). The obtained results are different for different viewers who helped to annotate the datasets. These results show high success levels on certain viewers, while lower (but still good) levels on other ones. In particular very high success was obtained using only prediction of activation values for one user who seemed to use mainly facial cues, whilst a similar, but slightly lower success level, was obtained on an annotator, who used predominantly prosodic cues. Other two annotators appeared to use cues from all modalities, and for them, the success levels were still good but not so outstanding.

This leads to the need for a further study to follow up the spread of such cue-extraction across the populace, since if this is an important component then it would be important to know how broad is this spread, as well as to develop ways to handle such a spread (such as having a battery of networks, each trained on the appropriate subset of cues). It is, thus evident that adaptation to specific users and contexts is a crucial aspect in this type of fusion. Decision-level fusion caters for integrating asynchronous but temporally correlated modalities. Here, each modality is first classified independently and the final classification is based on fusion of the outputs of the different modalities. Designing optimal strategies for decision level fusion is still an open research issue. Various approaches have been proposed, e.g. sum rule, product rule, using weights, max/min/median rule, majority vote etc. As a general rule, semantic fusion builds on individual recognizers, followed by an integration process; individual recognisers can be trained using unimodal data, which are easier to collect.

In the rest of this paper, we examine the confidence produced by each classifier, such as a feed-forward multilayer neural network, handling a single modality - focusing on facial expressions - and we derive an efficient methodology for adapting the classifier's performance, when detecting such a need, by collecting data from its specific environment. Thus, in the framework presented here, facial expression is considered as the dominant modality; this means that most of the time classification is performed using the facial features as input. In cases where the network trained with the facial data does not perform well (hence, the need to adapt arises), speech

prosody or gestures can be utilized as "fall-back" solutions, possibly providing the expected output for the adaptation process.

2 The Adaptive neural network architecture

Let us assume that we seek to classify, to one of, say, p available emotion classes ω, each input vector $\underline{x}_i$ containing the features extracted from the input signal. A neural network produces a p-dimensional output vector $\underline{y}(\underline{x}_i)$

$$\underline{y}(\underline{x}_i) = \left[p^i_{\omega_1} p^i_{\omega_2} \cdots p^i_{\omega_p} \right]^T \tag{1}$$

where $p^i_{\omega_j}$ denotes the probability that the ith input belongs to the jth class.

Let us first consider that the neural network has been initially trained to perform the classification task using a specific training set, say, $S_b = \left\{ (\underline{x}'_1, \underline{d}'_1), \cdots, (\underline{x}'_{m_b}, \underline{d}'_{m_b}) \right\}$, where vectors $\underline{x}'_i$ and $\underline{d}'_i$ with $i = 1,2,\cdots,m_b$ denote the ith input training vector and the corresponding desired output vector consisting of p elements.

Then, let $\underline{y}(\underline{x}_i)$ denote the network output when applied to a new set of inputs, and let us consider the ith input outside the training set, possibly corresponding to a new user, or to a change of the environmental conditions. Based on the above described discussion, slightly different network weights should probably be estimated in such cases, through a network adaptation procedure.

Let $\underline{w}_b$ include all weights of the network before adaptation, and $\underline{w}_a$ the new weight vector which is obtained after adaptation is performed. To perform the adaptation, a training set S_c has to be extracted from the current operational situation composed of, (one or more), say, m_c inputs; $S_c = \left\{ (\underline{x}_1, \underline{d}_1), \cdots, (\underline{x}_{m_c}, \underline{d}_{m_c}) \right\}$ where $\underline{x}_i$ and $\underline{d}_i$ with $i = 1,2,\cdots,m_c$ similarly correspond to the i-th input and desired output data used for adaptation. The adaptation algorithm that is activated, whenever such a need is detected, computes the new network weights $\underline{w}_a$, minimizing the following error criteria with respect to weights,

$$\begin{aligned} E_a &= E_{c,a} + \eta E_{f,a} \\ E_{c,a} &= \frac{1}{2} \sum_{i=1}^{m_c} \left\| \underline{z}_a(\underline{x}_i) - \underline{d}_i \right\|_2 \\ E_{f,a} &= \frac{1}{2} \sum_{i=1}^{m_b} \left\| \underline{z}_a(\underline{x}'_i) - \underline{d}'_i \right\|_2 \end{aligned} \tag{2}$$

where $E_{c,a}$ is the error performed over training set S_c ("current" knowledge), $E_{f,a}$ the corresponding error over training set S_b ("former" knowledge); $\underline{z}_a(\underline{x}_i)$ and $\underline{z}_a(\underline{x}'_i)$ are the outputs of the adapted network, corresponding to input vectors $\underline{x}_i$ and $\underline{x}'_i$ respectively, of the network consisting of weights $\underline{w}_a$. Similarly $\underline{z}_b(\underline{x}_i)$ would represent the output of the network, consisting of weights $\underline{w}_b$, when

accepting vector $\underline{x}_i$ at its input; when adapting the network for the first time $\underline{z}_b(\underline{x}_i)$ is identical to $\underline{y}(\underline{x}_i)$. Parameter η is a weighting factor accounting for the significance of the current training set compared to the former one and $\|\cdot\|_2$ denotes the L_2 -norm.

The goal of the training procedure is to minimize (2) and estimate the new network weights $\underline{w}_a$. The adopted algorithm has been proposed by the authors in 1. Let us first assume that a small perturbation of the network weights (before adaptation) $\underline{w}_b$ is enough to achieve good classification performance. Then,

$$\underline{w}_a = \underline{w}_b + \Delta\underline{w}$$

where $\Delta\underline{w}$ are small increments. This assumption leads to an analytical and tractable solution for estimating $\underline{w}_a$, since it permits linearization of the non-linear activation function of the neuron, using a first order Taylor series expansion.

Equation (2) indicates that the new network weights are estimated taking into account both the current and the previous network knowledge. To stress, however, the importance of current training data in (2), one can replace the first term by the constraint that the actual network outputs are equal to the desired ones, that is

$$z_a(\underline{x}_i) = d_i \quad i = 1,...,m_c, \quad \text{for all data in } S_c \tag{3}$$

Through linearization, solution of (3) with respect to the weight increments is equivalent to a set of linear equations

$$\underline{c} = \mathbf{A} \cdot \Delta\underline{w} \tag{4}$$

where vector $\underline{c}$ and matrix **A** are appropriately expressed in terms of the previous network weights. In particular,

$$\underline{c} = \left[d_1 \cdots d_{m_c}\right]^T - \left[z_b(\underline{x}_1) \cdots z_b(\underline{x}_{m_c})\right]^T \tag{5}$$

Moreover, minimization of the second term of (2), which expresses the effect of the new network weights over data set S_b, can be considered as minimization of the absolute difference of the error over data in S_b with respect to the previous and the current network weights. This means that the weight increments are minimally modified, with respect to the following error criterion

$$E_S = \left\| E_{f,a} - E_{f,b} \right\|_2 \tag{6}$$

with $E_{f,b}$ defined similarly to $E_{f,a}$, with $\underline{z}_a$ replaced by $\underline{z}_b$ in (2).

It can be shown 5 that (6) takes the form of

$$E_S = \frac{1}{2}(\Delta\underline{w})^T \cdot \mathbf{K}^T \cdot \mathbf{K} \cdot \Delta\underline{w} \tag{7}$$

where the elements of matrix **K** are expressed in terms of the previous network weights $\underline{w}_b$ and the training data in S_b. The error function defined by (7) is convex since it is of squared form. Thus, the weight increments can be estimated through solution of (7). The gradient projection method has been used in [6] to estimate the weight increments.

Each time the decision mechanism ascertains that adaptation is required, a new training set S_c is created, which represents the current condition. Then, new network weights are estimated taking into account both the current information (data in S_c) and the former knowledge (data in S_b). Since the set S_c has been optimized only for the current condition, it cannot be considered suitable for following or future states of the environment. This is due to the fact that data obtained from future states of the environment may be in conflict with data obtained from the current one. On the contrary, it is assumed that the training set S_b, which is in general based on extensive experimentation, is able to roughly approximate the desired network performance at any state of the environment. Consequently, in every network adaptation phase, a new training set S_c is created and the previous one is discarded, while new weights are estimated based on the current set S_c and the old one S_b, which remains constant throughout network operation.

3 Detecting the need for adaptation

The purpose of this mechanism is to detect when the output of the neural network classifier is not appropriate and consequently to activate the adaptation algorithm at those time instances when a change of the environment occurs.

Let us first assume that a network adaptation has taken place and let us focus visual inputs. Let $\underline{x}(k)$ denote the feature vector of the *k*-th image or image frame, following the time at which adaptation occurred. Index *k* is therefore reset each time adaptation takes place, with $\underline{x}(0)$ corresponding to the feature vector of the image where the adaptation of the network was accomplished. At this input, the network performance had deteriorated, i.e., the network output deviated from the desired one. Let us recall that vector $\underline{c}$ in eq. (5) expresses the difference between the desired and the actual network outputs based on weights $\underline{w}_b$ and applied to the current data set. As a result, if the norm of vector $\underline{c}$ increases, network performance deviates from the desired one and adaptation should be applied. On the contrary, if vector $\underline{c}$ takes small values, then no adaptation is required. In the following we use the difference between the output of the adapted network and of that produced by the initially trained classifier to approximate the value of $\underline{c}$. Moreover, we assume that the difference computed when processing input $\underline{x}(0)$ constitutes a good estimate of the level of improvement that can be achieved by the adaptation procedure. Let us denote by $e(0)$ this difference and let $e(k)$ denote the difference between the corresponding classifiers' outputs, when the two networks are applied to $\underline{x}(k)$. It is anticipated that the level of improvement expressed by $e(k)$ will be close to that of $e(0)$ as long as the classification results are good. This will occur when input images are similar to the ones used during the adaptation phase. An error $e(k)$, which is quite different from $e(0)$, is generally due to a change of the environment. Thus, the quantity $a(k) = |e(k) - e(0)|$ can be used for detecting the change of the

environment or equivalently the time instances where adaptation should occur. Thus, no adaptation is needed if:

$$a(k) < T \tag{8}$$

where T is a threshold which expresses the max tolerance, beyond which adaptation is required for improving the network performance.

Such an approach detects with high accuracy the adaptation time instances both in cases of abrupt and gradual changes of the operational environment since the comparison is performed between the current error difference $e(k)$ and the one obtained right after adaptation, i.e., $e(0)$. In an abrupt operational change, error $e(k)$ will not be close to $e(0)$; consequently, $a(k)$ exceeds threshold T and adaptation is activated. In case of a gradual change, error $e(k)$ will gradually deviate from $e(0)$ so that the quantity $a(k)$ gradually increases and adaptation is activated at the frame where $a(k) > T$.

Network adaptation can be instantaneously executed each time the system is put in operation by the user. Thus, the quantity $a(0)$ initially exceeds threshold T and adaptation is forced to take place.

4 Experimental Study

Our experiments aimed at investigating the practical stand of the proposed adaptation procedure. The main idea of the experimental study was to explore the performance of the adapted networks over inputs belonging to the same tune, but not used for adaptation, as well as to tunes of the same emotional quadrant as the one used for adaptation purposes.

Out of approximately 35.000 frames, belonging to 477 tunes of the SAL database 3, we selected a merely 500 frames – from all four subjects - for training a feed-forward back-propagation network referred from now as NetProm. The architecture details for NetProm are three layers consisting of 10 and 5 neurons on the first and second hidden layers respectively and 5 neurons of the output layer. The targets were formatted as a 5x1 vector for every frame so as to only one, of the 5 candidate classes, was equal to 1. So for example if the frame used for training belonged to the first quadrant the output vector would be [1 0 0 0 0]. The fifth class of the classification problem corresponds to the neutral emotional state and the other four to the four quadrants of the Whissel's wheel.

The selection of the 500 frames used for training the NetProm network was made following a prominence criterion. More specifically, for every frame, a metric was assigned denoting the distance of the values of the FAPs for that specific frame with reference to the mean values of the FAPs of the other frames belonging to the same class. This metric of FAP variance was the sorting parameter for the frames. Under the constraint that each class should be represented as equally as possible we selected the 500 most prominent frames and used it as input for training the NetProm network.

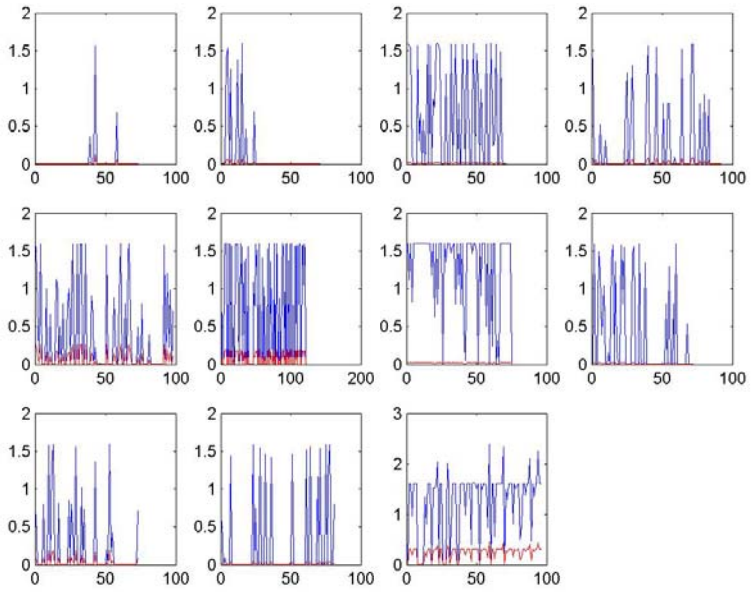

Fig. 2. MSE of NetProm (blue) and Neti (red)

With regard to the adaptation phase we selected eleven tunes - from a single subject - consisting of the largest number of frames. This selection was based on the idea that it would not make much sense selecting very short tunes, because the adaptation data would be very sparse as will be explained later. Also we made sure that no frame belonging to these eleven tunes was used for training NetProm. Each of the eleven tunes was divided into two groups of frames, the adaptation group and the testing group containing 30% and 70% of the total frames of the original tune, sorted by the prominence criterion, respectively.

NetProm was adapted using the adaptation group of the eleven tunes and produced eleven new networks Net_i, i=1..11. Each Net_i was then tested on the testing group of the respective tune and the results can be seen in **Error! Reference source not found.**. It is clear that the adaptation procedure has been beneficial and greatly reduced the MSE for every tune it was applied.

Furthermore, we tested the procedure proposed in section 4 for detecting when adaptation is necessary. In particular, we used the above derived Net_i and compared their performance with that of NetProm through criterion (8) in 11 synthetic experiments, shown in Figure 3. In the first 6 experiments and in the 9th, there was no change of the subject showing the expression. It can be verified that the value of $e(k)$, for all values of k shown in the horizontal axis, are close to the $e(0)$ value, so no need for adaptation was detected. On the contrary, the 7th, 8th, 10th and 11th experiments contained one or more frames where a different subject (the first) showed a similar expression. In most of these cases the $\alpha(k)$ value was raised due to the inappropriateness of the adapted (to the fourth subject) network to cope well with the specific characteristics of the first subject. Consequently the need for (new) adaptation was detected through usage of criterion (8).

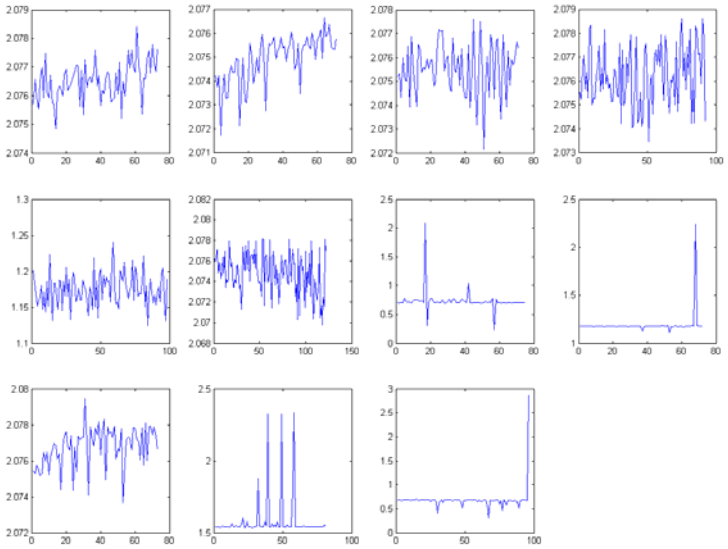

Fig. 3. Detecting the need for network adaptation using the criterion of eq.(8)

These results are very promising indicating that the proposed process can form an effective adaptation tool in expression/emotion recognition.

5 Conclusions – Future work

Recognition of facial expressions is an important part of human-computer interaction, especially since psychological research has shown that the face is a vital ingredient of human expressivity. However, in everyday HCI, emotions are usually subtle, hence difficult to pick out using a small set of universal labels; to tackle this, one needs to consider multiple modalities as a "fall-back" or reinforcement solution. In addition to this, personalized expressivity and context-dependence make generalization of learning techniques a daunting task.

In this paper we proposed an extension of a neural network adaptation procedure which caters for training from different modalities. After training and testing on a particular subject, the best-performing network is adapted using prominent samples from discourse with another subject, so as to adapt and improve its ability to generalize. Results shown here indicate that the performance of the network is improved using this approach, without the need to train a specific network for each subject, which would wipe out the nice generalization attribute of the network. Future work includes the extension of this work to include speech-related modalities, deployment on different naturalistic contexts and introduction of mechanisms to handle uncertainty in the various modalities and decide which of them would be the more robust to depend upon for co-training.

Acknowledgments: This research is partly supported by the European Commission as part of the FEELIX GROWING project (http://www.feelix-growing.org) under contract FP6 IST-045169. The views expressed in this paper are those of the authors, and not necessarily those of the consortium.

References

1. N. Doulamis, A. Doulamis and S. Kollias, On-Line Retrainable Neural Networks: Improving Performance of Neural Networks in Image Analysis Problems, IEEE Transactions on Neural Networks, vol. 11, no.1, pp.1-20, January 2000.
2. D. Goleman, Emotional Intelligence. Bantam Books, New York, NY, USA, 1995.
3. S. Ioannou, A. Raouzaiou, V. Tzouvaras, T. Mailis, K. Karpouzis, S. Kollias, Emotion recognition through facial expression analysis based on a neurofuzzy network, Neural Networks, Elsevier, Volume 18, Issue 4, May 2005, Pages 423-435
4. Humaine Network of Excellence on Emotions, http://emotion-research.net
5. D. Park, M. A. EL-Sharkawi, and R. J. Marks II, An adaptively trained neural network, IEEE Trans. Neural Networks, vol. 2, pp. 334–345, 1991.
6. R. Picard, Affective Computing, The MIT Press, Cambridge, MA, USA, 1997.
7. J. Taylor, N. Fragopanagos, The interaction of attention and emotion, Neural Networks , Volume 18, Issue 4, May 2005, pp. 353 – 369.
8. R. Adolphs, D. Tranel, H. Damasio and A. Damasio, Impaired recognition of emotion in facial expressions following bilateral damage to the human amygdale, Nature, 372, 669-672, 1994.
9. M.L Phillips, A. W. Young, C. Senior, M. Brammer, C. Andrew, A.J. Calder, E.T. Bullmore, D.I. Perrett, D. Rowland, S.C.R. Williams, J.A. Gray & A.S. David, A specific neural substrate for perceiving facial expressions of disgust, Nature 389, 495-498, 1997.