Neonatal Facial Pain Detection Using NNSOA and LSVM

S. Brahnam[1]**, L. Nanni**[2]**, and R. Sexton**[1]
[1]Computer Information Systems, Missouri State University, Springfield, Missouri, USA
[2]DEIS, IEIIT-CNR, Università di Bologna, Bologna, Italy

Abstract - *We report classification experiments using the pilot Infant COPE database of neonatal facial expressions. Two sets of DCT coeffiecents were used to train a neural network simultaneous algorithm (NNSOA) and a linear support vector machine (LSVM) to classify neonatal expressions into the two categories of pain and nonpain. In the first set (VAR) only 80 of the coefficients with the highest variance were included. In the second set (SFFS), 15 DCT coefficients were selected by applying Sequential Forward Floating Selection (SFFS) [1]. We found that NNSOA+VAR produced the best classification score of 95.38% accuracy, but with no statistical difference compared with the DCT sets. However, NNSOA using the DCT coefficients outperformed with statistical significance previous experiments reported in [2] that used PCA components. It is surmised that NNSOA, an algorithm that eliminates unnecessary weights, is more stable than LSVM and may be better than SFFS at identifying relevant features.*

Keywords: Pain Detection, Facial Expression Recognition, Face Classification, SFFS, NNSOA.

1 Introduction

In this paper, we report experiments using a neural network simultaneous algorithm (NNSOA) and a linear support vector machine (LSVM) to detect pain in the facial expressions of neonates. NNSOA is a global search procedure that searches from one population of neural network solutions to another, focusing on the area that provides the current best solution, while continuously sampling the total parameter space. NNSOA is a slight modification of a genetic algorithm (GA) used in previous neural network studies (see [3]). NNSOA takes advantage of the GA's ability to simultaneously search multiple points (or solutions) at one time, unlike gradient search techniques, such as backpropagation, that are able to search for only one solution at a time.

Because NNSOA uses a genetic algorithm for the search procedure, it is not limited to differentiable functions, as is the case with gradient search techniques. Thus, NNSOA can have objective functions that add a penalty for the number of nonzero weights in a solution. NNSOA is able to eliminate unneeded weights in a solution by intermittently exchanging solution weights with hard zeros and then evaluating whether that substitution helped or hindered the network's ability to predict using normal GA operations.

NNSOA was shown in [4] to outperformed backpropagation on eleven standard problems, ranging from simple linear functions to complex time series and multidimensional production functions and a real-world problem (predicting energy consumption in a building taken from Prechelt's [5]. As far as face classification is concerned, NNSOA has only been applied to the neonatal pain detection problem (see [6] and [2]). It has successfully handled several other medical problems, however, including diagnosing breast lumps and diabetes and predicting heart disease [7]. For additional details regarding NNSOA, see [2].

In our earlier infant COPE experiments, NNSOA used 70 PCA components as inputs. It classified the facial expressions into the classes of pain and nonpain with 100% accuracy using protocol A and 95.38% using protocol B [2].

Protocol A presents the best-case scenario by assuming that samples of individual subjects are available to personalize the classifier, as is the case with most commercial speech recognition software. This protocol would be most suitable for home applications, where the software would be trained to detect the expressions of pain in individual infants. In protocol A, multiple but different samples of neonatal pain and nonpain facial expressions from all subjects are used in the testing and training sets.

Protocol B presents the worst-case scenario by assuming that the classifier will be trained on one set of subjects and then applied out of the box to an unknown set of future newborns. In this evaluation protocol, images are divided by subject, and the testing set contains images of subjects that are not used in the training set. Protocol B is more realistic for medical uses, as hospital stays for most newborns are between three and four days. This protocol is also the most difficult to classify.

Various support vector machines, especially LSVM, have produced classification scores using PCA components that compare well with NNSOA. In [8] and [2], we performed comprehensive investigations of SVM using five kernels: linear, RBF, and polynomials of degree 2, 3, and 4. Using protocol A, SVM with polynomial of degree 3 produced the best results of 88.00% accuracy. Using protocol B, LSVM produced the best results with 82.39% accuracy.

The experiments presented in this paper compare NNSOA and LSVM using protocol B. Inputs to the systems are DCT coefficients that were selected based on the variance and on employing SFFS [1].

The remainder of this paper is outlined as follows. In section 2, we briefly describe the Infant COPE database and study design. In section 3, we provide a brief outline of NNSOA. Details of our experimental design are provided in section 4, and the results are reported in section 5. Finally, in section 6, we conclude this paper by discussing directions for further research.

2 Infant COPE database design

The pilot Infant COPE (Classification Of Pain Expressions) database, which we developed, was used in all the classification experiments reported in this paper. This database contains 204 photographs of 26 Caucasian neonates (13 boys and 13 girls) ranging in age from 18 hours to 3 days old.

Photographs were taken of the infants at baseline rest and while experiencing several noxious stimuli: bodily disturbance, an air stimulus on the nose, friction on the external lateral surface of the heel, and the pain of a heel stick. The goal of the Infant COPE study design was to obtain a representative and challenging set of facial images for classification experiments. Figure 1 illustrates some of the challenges presented by the images. This figure compares crying images that were triggered by a nonpain stimulus with crying images that were triggered by the pain stimulus.

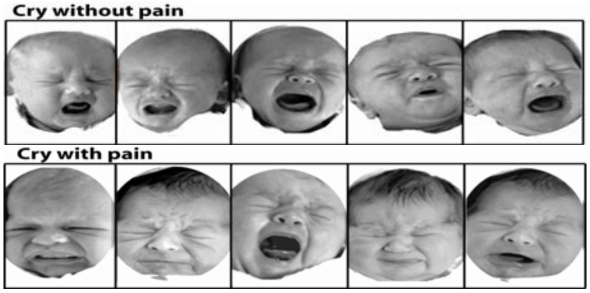

Figure 1. Illustration of challenging Infant COPE facial images.

For the pain detection problem, all images of the stimuli are divided into the two categories of pain (60 images) and nonpain (140 images).

For a detailed description of the database and study design, see [6].

3 NNSOA

As noted in the introduction, gradient search techniques, such as backpropagation, used for searching for optimal weights in a neural network (NN) solution, do not have the ability to zero out weights that have no value in a solution. The search must find a solution that has values for these unneeded weights. This works well for training data but is likely to introduce additional errors in the estimates when these solutions are applied to out-of-sample data. A solution found by the NNSOA completely eliminates this possibility of additional error because the unneeded weights are set to a hard zero. This ability to zero out weights in a solution provides additional feature reduction since those inputs that are of no value to a solution are basically removed from the solution. In addition, those inputs that offer the most value can easily be isolated by examining the weights of the NN solution.

Another advantage offered by NNSOA is that it finds the appropriate NN architectures by searching for the correct number of hidden nodes in a solution. This is done by starting a network with only one hidden node. After a user specified number of generations (MAXHID), the best solution out of the population of solutions is saved, and an additional hidden node is added to the architecture and trained for another MAXHID generation. The previous best solution is included in this additional training by replacing one of the randomly initialized solutions with the best solution found in the previous architecture. Since adding an additional node to the architecture increases the number of weights in the solutions equal to the number of inputs plus one, the additional weights for this best solution are set to hard zeros. The process of adding an additional hidden node after every MAXHID generation continues until the current best solution is worse than the previous architecture's best solution. At this point, the number of hidden nodes is set to the number of hidden nodes in the previous architecture, and the NN continues with the training process for a user defined number of generations. Once the MAXGEN number of generations has been reached, training is complete.

Below we offer an outline of NNSOA operations.

3.1 Outline of NNSOA

NNSOA classification can be broken down into the following operations: initialization, evaluation, reproduction, crossover, mutation 1, mutation 2, convergence enhancement, and termination.

Initialization. A population of 12 solutions is created by drawing random real values from a uniform distribution [-1, 1] for input weights. The output weights are determined by ordinary least squares (OLS).

Evaluation. Each member of the current population is evaluated by an objective function based on its sum-of-squared-error (SSE) value in order to assign each solution a probability for being redrawn in the next generation. To search for a parsimonious solution, a penalty value is added to the SSE for each nonzero weight (or active connection). The penalty for keeping an additional weight varies during the search and is equal to the current value of the root mean squared error (RMSE). This means that the penalty for keeping additional weights is high at the beginning of the training process when errors are high. As the optimization process gets closer to the final solution, errors decrease and the penalty value becomes smaller. Based on the objective function, each of the 12 solutions in the population is evaluated. The probability of being drawn in the next generation is calculated by dividing the distance of the current solution's objective value from the worst objective value in the generation by the sum of all distances in the current generation.

Reproduction. A mating pool of 12 solutions is created by selecting solutions from the current population based on their assigned probability. This is done by selecting a random number in the range of 0 to 1 and comparing it to the cumulative probability of the current solution. When it is found that the random value is less than the current solution's cumulative probability, the current string is drawn for the next generation. This is repeated until the entire new generation is drawn. It should be noted that a given solution can be drawn more than once or not at all, depending on its assigned probability.

Crossover. Once reproduction occurs, providing a combination of solutions from the previous generation, the 12 solutions are then randomly paired so that 6 pairs are produced. A point is randomly selected for each pair. New solutions are produced by switching the weights above the randomly generated point. In this fashion, 12 new solutions are generated for the next generation.

Mutation 1. For each weight in a population of solutions, a random number is drawn; if the random value is less than 0.05, the weight is replaced by a value randomly drawn from the entire weight space. By doing this, the entire weight space is globally searched, thus enhancing the algorithm's ability to find global solutions.

Mutation 2. For each weight in the population of solutions, a random number is drawn; if the random value is less than 0.05, the weight is replaced by a hard zero. As a result of doing this, unneeded weights are identified as the search continues for the optimum solution. After this operation is performed, this new generation of 12 solutions begins again with evaluation, and the cycle continues until it reaches 70% of the maximum set of generations.

Convergence Enhancement. Once 70% of the maximum set of generations has been completed, the best solution replaces all the strings in the current generation. The weights of these 12 identical solutions are then modified by adding a small random value to each weight. These random values decrease to an arbitrarily small number as the number of generations increases to its set maximum number.

Termination. The algorithm terminates on a user specified number of generations.

4 Experimental Design

This section describes experimental results using NNSOA and LSVM to classify the 204 images in the Infant COPE database into the binary categories of pain and nonpain. Input into NNSOA and LSVM were the pixel values of the images transformed using the DCT transform and two feature reduction techniques. As illustrated in figure 2, the experimental procedures can be divided into the following stages: image preprocessing, feature transformation and selection, and classification.

In the preprocessing stage, images were normalized. Images were rotated and scaled so that the faces were approximately equal in size and the eyes intersected the same horizontal axis. The original images, size 3008 x 2000 pixels, were reduced to 100 x 120 pixels and cropped. The rows of pixels within the images were then concatenated to form an input vector of dimension 12000 with entries ranging in value between 0 and 255.

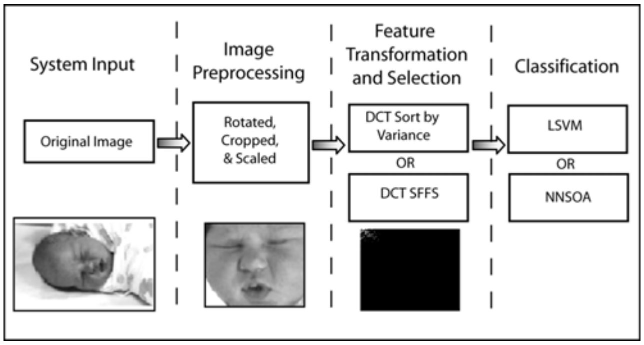

Figure 2. LSVM and NNSOA Classification Systems.

In the feature transformation and selection stage, the DCT transform was used to reduce the dimensionality of the raw input vectors. Two variable reduction techniques were employed. In the first, DCT coefficients were sorted by variance, as in [9], and feature reduction was accomplished by performing a series of LSVM experiments that determined the top 80 coefficients were optimal for the classification task. In the experiments reported in this section, we label this feature set VAR. In the second technique for variable reduction, Sequential Forward Floating Selection (SFFS) [1], was used to select a feature set of only 15 DCT coefficients. The SFFS selection criterion function was the minimum error of LSVM classification. This feature set, which we label *SFFS*, contained the following DCT coefficients: 4, 804, 903, 403, 1006, 402, 2, 13, 1401, 1204, 109, 215, 603, 1010, and 707. SFFS was implemented using the MATLAB toolbox PRTools 3.1.7 [10]. Both sets of DCT coefficients were normalized to [0, 1].

In the classification stage, LSVM and NNSOA experiments were performed with both sets of DCT inputs. We label the classifier/input combinations LSVM+VAR, LSVM+SFFS, NNSOA+VAR, and NNSOA+SFFS. Following protocol B, images are divided by subject, and the testing set contains subjects that are not used in the training set. Thus, a total of 26 experiments were performed by each of the four classifier combinations. That is, for each of the 26 subjects in the database, the set of facial images for that subject formed the testing set, and the facial images of the remaining 25 subjects formed the training set. The classification scores for each experiment were computed by averaging the number of correct classifications made.

5 Results

Tables 1 and 2 present the classification scores and 95% confidence intervals for the four classifier combinations. Examining these tables, we see that NNSOA+VAR has the highest classification rate of 95.38% accuracy, with a 95% confidence interval of ± 2.81%. NNSOA+VAR also has the lowest standard deviation of 6.97%. It is, therefore, the most stable method of classification.

An unexpected result is the slightly lower classification score for NNSOA+SFFS compared with the NNSOA+VAR classification score. Recent criticisms have been levied against SFFS. In [11], for instance, experiments were conducted that showed SFFS does not necessarily yield the best subsets. It may be the case that NNSOA is superior to SSFS in isolating relevant DTC coefficients. This may be due to NNSOA's mechanism of eliminating unneeded inputs. However, new studies will need to be conducted to evaluate whether this is the case, since, as can be seen in Table 2, there is no statistical difference in performance between NNSOA+VAR and NNSOA+SFFS. In fact, using DCT inputs there is no statistical difference in the classification performance of any of the four classifier combinations.

Table 1. Average and all 26 subject classification scores.

Subj.	LSVM+VAR	LSVM+SFFS	NNSOA+VAR	NNSOA+SFFS
1	88.89%	77.78%	88.89%	100.00%
2	100.00%	100.00%	100.00%	100.0%0
3	100.00%	87.50%	87.50%	100.00%
4	80.00%	100.00%	100.00%	100.00%
5	66.67%	75.00%	75.00%	83.33%
6	85.71%	85.71%	85.71%	85.71%
7	88.89%	77.78%	88.89%	77.78%
8	100.00%	88.89%	88.89%	100.00%
9	50.00%	66.67%	100.00%	66.67%
10	100.00%	90.00%	100.00%	90.00%
11	100.00%	100.00%	100.00%	100.00%
12	100.00%	75.00%	87.50%	75.00%
13	90.00%	90.00%	90.00%	90.00%
14	100.00%	100.00%	100.00%	100.00%
15	83.33%	100.00%	100.00%	100.00%
16	83.33%	91.67%	100.00%	100.00%
17	100.00%	100.00%	100.00%	100.00%
18	100.00%	100.00%	100.00%	100.00%
19	85.71%	100.00%	100.00%	100.00%
20	75.00%	100.00%	100.00%	100.00%
21	100.00%	100.00%	100.00%	100.00%
22	75.00%	62.50%	87.50%	87.50%
23	83.33%	100.00%	100.00%	100.00%
24	100.00%	100.00%	100.00%	100.00%
25	40.00%	100.00%	100.00%	100.00%
26	100.00%	50.00%	100.00%	66.67%
Ave.	**87.53%**	**89.17%**	**95.38%**	**93.14%**

In table 3, we compare LSVM and NNSOA using DCT with PCA components. In our previous experiments [2, 8], the first 70 PCA coefficients were used. PCA coefficients are ordered, with each one accounting for the most variation among a set of faces. In [2], which used the same experimental protocol used in this study, NNSOA+PCA had the highest classification rate of 90.20% accuracy and LSVM+PCA had a classification rate of 82.35%. Using the DCT coefficients, NNSOA has outperformed LSVM+PCA with statistical significance.

More detailed discussions of the NNSO and LSVM parameters used in the DCT experiments are provided below.

Table 2. Standard Deviation of DCT Experiment Scores and 95% Confidence Intervals.

Method	95% Confidence Interval	Standard Deviation
LSVM+VAR	87.53% ± 6.47%	16.01%
LSVM+SFFS	89.17% ± 5.69%	14.09%
NNSOA+VAR	95.38% ± 2.81%	6.97%
NNSOA+SFFS	93.18% ± 4.40%	10.90%
LSVM+PCA	82.35% ± 6.20%	15.34%
NNSOA+PCA	90.20% ± 4.16%	10.30%

The 95% Confidence Intervals were computed using t distribution ($\bar{x} \pm t_{\alpha/2}\, s/\sqrt{n}$).

Table 3. Standard Deviation of PCA Experiment Scores and 95% Confidence Intervals

Method	95% Confidence Interval	Standard Deviation
LSVM+PCA	82.35% ± 6.20%	15.34%
NNSOA+PCA	90.20% ± 4.16%	10.30%

As reported in [2].

5.1 NNSOA experiments

A total of 26 separate NNs were trained and tested for the two DCT coefficient sets. For all experiments, MAXHID and MAXGEN were set to 100 and 5,000 respectively. The objective function used in the experiments is as follows:

$$Min\,\{E = \sum \left(O_i - \hat{O}_i\right)^2 + C\sqrt{\frac{\sum_{i=1}^{N}\left(O_i - \hat{O}_i\right)^2_i}{N}}\;\} \quad (1)$$

where N is the number of observations in the dataset, O is the observed value of the dependent variable, $\hat{O}$ is the NN estimate, and C is the number of nonzero weights in the network.

For these experiments, the output weights were found by using ordinary least squares, i.e., by regressing the outputs from the hidden nodes onto the real outputs. In this way only values for the input weights were searched.

The average number of hidden nodes for the 26 NNSOA+VAR networks was 3.31. Since there were 80 inputs + 1 bias, each additional hidden node added 81 weights to the architecture, making the average number of weights overall 268.11 (81 × 3.31). However, since the NNSOA eliminated unneeded weights by turning most of them to zero, the actual average real valued weights was 27.04. The reduction of weights on average was over 90%.

In the NNSOA+SFFS data, the inputs were already reduced, thus the number of hidden nodes was set to 15. Since there were 15 inputs + 1 bias, each additional hidden node added 16 inputs to the solution, making the number of weights in the solution 256 (16 × 16). Because NNSOA eliminates weights in a solution by zeroing out the weights that are not useful for prediction, the average number of weights across the 26 NNSOA+SFFS NNs was much smaller. On average, only 19.5 of the average total of 256 possible weights were found to be nonzero.

An added advantage of eliminating unneeded weights is the identification of relevant input variables. The input variables that had all zero weight connections were not used in producing estimates. As a result, the average number of inputs that were actually used in NNSOA+VAR prediction across the 26 networks was reduced from 80 to an average of only 21.88.

In NNSOA+SFFS, all 15 DCT coefficients selected by SFFS were found relevant for classification. Thus none were eliminated by NNSOA.

All NNSOA experiments were conducted on a 1.5 GHz machine, using the Windows XP operating system. The core code of the NNSOA program was written in FORTRAN, with Visual Basic used for the interface.

5.2 LSVM experiments

LSVM performed the 26 experiments defined by evaluation protocol B using the regularization parameter, C=1, and the bandwidth parameter, $\gamma = 1$. These values were determined using a grid search. All LSVM experiments were processed in the MATLAB environment using the OSU SVM Classifier MATLAB Toolbox developed by Ohio State University.

6 Conclusion

Determining which algorithm, NNSOA and LSVM, is best for this classification task is not a simple matter. In our discussion of results and comparison of our experiments using DCT components with our earlier experiments using PCA components, NNSOA appears to have the classification edge. However, since LSVM is comparable in its classification rates and several SVM MATLAB packages are freely available, LSVM may be the algorithm of choice.

It may prove possible to combine LSVM with NNSOA to obtain even better classification scores. In our opinion, these two classification methods offer complementary information about the patterns to be classified, and it is well known in the literature that classifier ensembles that enforce diversity fare better than ones that do not [12].

In future studies we shall investigate ensemble schemes using NNSOA and LSVM, as well as other classifier combinations. We also plan to conduct studies investigating NNSOA as a feature selector as it appears to select fewer, yet perhaps more essential, components for classification.

4 References

[1] P. Pudil, J. Novovicova, and J. Kittler, “Floating search methods in feature selection,” Pattern Recognition Letters, vol. 5, no. 11, pp. 1119-1125, 1994.

[2] S. Brahnam, C. Chuang, R. Sexton, M. Slack, “Machine assessment of neonatal facial expressions of acute pain,” Decision Support Systems, vol. 43, pp. 1247-1254, 2007.

[3] R. Sexton, R. Dorsey, and J. Johnson, “Toward a global optimum for neural networks: A comparison of the genetic algorithm and backpropagation,” Decision Support Systems, vol. 22, pp. 171-185, 1998.

[4] R. Sexton, R. Dorsey, and N. Sikander, “Simultaneous optimization of neural network function and architecture algorithm,” Decision Support Systems, vol. 36, pp. 283-296, 2004.

[5] L. Prechelt, A study of experimental evaluations of neural network learning algorithm: Current research practice, Technical Report 19/94, Fakultat fur Informatik, Universitat Karlsruhe, D-76128 Karsruhe, Germany, 1994.

[6] S. Brahnam, L. Nanni, and R. Sexton, "Introduction to neonatal facial pain detection using common and advanced face classification techniques," Computational Intelligence In Healthcare, Jain Lakhmi, ed., New York: Springer-Verlag, 2007.

[7] R. Sexton, and R. Dorsey, “Reliable classification using neural networks: A genetic algorithm and backpropagation comparison,” Decision Support Systems, vol. 30, pp. 11-22, 2000.

[8] S. Brahnam, C. Chuang, F. Shih, M. Slack, “Machine recognition and representation of neonate facial displays of acute pain,” International Journal of Artificial Intelligence in Medicine (AIIM), vol. 36, no. 3, pp. 211-222, 2006.

[9] Z. Pan, A. Rust, and H. Bolouri, "Image redundancy reduction for neural network classification using discrete cosine transforms." pp. 149-154, 2000.

[10] F. van der Heijden, R. Duin, D. de Ridder, D. Tax, Classification, parameter estimation, and state estimation: An engineering approach using MATLAB, Chichester, UK: John Wiley & Sons, Ltd, 2004.

[11] J. Reunanen, “Overfitting in making comparisons between variable selection methods,” Journal of Machine Learning Research, vol. 3, pp. 1371-1382, 2003.

[12] K. Pun, and Y. Moon, "Recent advances in ear biometrics." pp. 164-169, 2004.