ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 139-144

Evolutionary Optimization of Neural Networks for Face Detection

Stefan Wiegand† Christian Igel† Uwe Handmann‡

†Institut für Neuroinformatik, Ruhr-Universität Bochum,
44780 Bochum, Germany

‡Viisage Technology AG, 44801 Bochum, Germany

Abstract. For face recognition from video streams speed and accuracy are vital aspects. The first decision whether a preprocessed image region represents a human face or not is often made by a neural network, e.g., in the Viisage-FaceFINDER® video surveillance system. We describe the optimization of such a network by a hybrid algorithm combining evolutionary computation and gradient-based learning. The evolved solutions perform considerably faster than an expert-designed architecture without loss of accuracy.

1 Introduction

Real-time face recognition is a challenging task requiring fast and accurate image classifiers. The Viisage-FaceFINDER® video surveillance system [9], formerly known as ZN-SmartEye®, automatically identifies people by their faces in a three step process: first, regions that contain a face are detected, then specific face models are calculated, and finally these models are looked up in a database. In the detection step, different biologically motivated cues are used to cluster the given images into regions of high and low significance, cf. [3] for a similar approach. The clusters of high significance are then classified as either containing an upright frontal face or not by a task specific feed forward neural network. As stated in a recent survey "The advantage of using neural networks for face detection is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one drawback is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance" [10]. In the following, we therefore address the task of optimizing the face detection network of Viisage-FaceFINDER®. We apply a hybrid algorithm, which uses

This work was supported by the German Federal Ministry of Education and Research (BMBF) under grants LOKI 01 IB 001 and MORPHA 01 IL 902.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 139-144

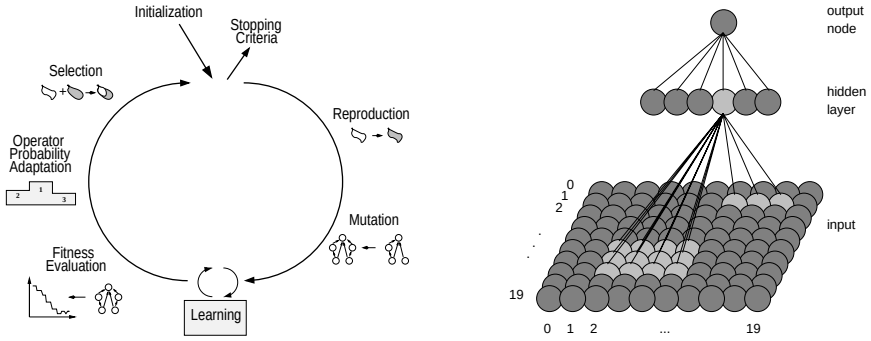


Figure 1: Left, the hybrid evolutionary algorithm. Right, visualization of the neural input dimension and the receptive field connectivity.

recent developments from evolutionary and gradient-based optimization, to the adaptation of the architecture and the weights of the face detection network in order to speed up calculation time and to increase classification performance. In the following section we outline the hybrid optimization algorithm and in section 3 we report on experimental results. Finally, we draw conclusions and give an outlook.

2 Evolutionary Network Optimization

In the efficient and hardware-friendly implementation of the face detection neural network within Viisage-FaceFINDER® the speed of the classification scales approximately linearly with the number of hidden neurons and not with the number of connections. With every hidden neuron that is saved detection costs are reduced by approximately one percentage point. Hence, the goal of the optimization is to reduce the number of hidden nodes of the detection network under the constraint that the classification error does not increase. We tolerate an increase in the number of connections as long the number of neurons decreases. Note that "the smaller the network the better the generalization" does not necessarily hold, see [1] and references therein. We initialize our optimization algorithm and compare our results with the expert-designed architecture of [8], the *reference topology*. This network has been tailored to the face detection task and has become a standard reference for neural network based face detection [10].

Evolutionary algorithms are an established method for the optimization of the topology of neural networks, see [11] for an overview. The optimization loop of our hybrid evolutionary algorithm is shown in Fig. 1 (left). Its basic scheme might be regarded as a canonical evolutionary network optimization method using direct encoding, nested learning, and Lamarckian inheritance. However, there are some special features described in the following. We initialize the parent population with 25 individuals that all represent the *reference topology*, a 400-52-1 network structure, but with different random weight initializations.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 139-144

name	pattern	faces	usage
D_{train}	3000	1500	network training and fitness evaluation
D_{val}	1400	700	cross-validation of learning and fitness evaluation
D_{test}	2000	1000	final network choice
D_{extern}	2200	1100	external evaluation

Figure 2: Left, the input to the face detection network are preprocessed 20×20 pixel grayscale images showing either frontal, upright (positive) face and (negative) nonface examples. The preprocessing comprises rescaling, lightning correction, and histogram equalization. Right, for optimization and evaluation we have partitioned the available patterns into 4 disjoint data sets.

The 400 inputs correspond to the pixels of the image patterns, cf. Fig. 1 (right) and Fig. 2 (left). No hidden neuron is fully connected to the input but to certain receptive fields, see below. The total number of connections amounts to 2905. We partitioned all the available data into 4 sets D_{train}, D_{val}, D_{test}, and D_{extern}, see Fig. 2 (right).

Each parent creates one child per generation by reproduction. The offspring is then mutated by elemental variation operators. These are chosen randomly for each offspring from a set of operators and are applied sequentially. The process of choosing and applying an operator is repeated $1 + x$ times, where x is an individual realization of a Poisson distributed random number with mean 1. There are 5 basic operators: *add-connection*, *delete-connection*, *add-node*, *delete-node*, and *jog-weights* [5]. The elemental deletion operators are based on the *magnitude based pruning* heuristic, which assigns a higher probability to the deletion of small weights. In addition to the 5 basic operators, there are 3 task-specific mutations inspired by the concept of "receptive fields", i.e., dimensions of the input space that correspond to rectangular regions of the input image, cf. Fig. 1 (right). The RF-operators *add-RF-connection*, *delete-RF-connection*, and *add-RF-node* behave as their basic counterparts, but act on groups of connections. They consider the topology of the image plane by taking into account that "isolated" processing of pixels is rarely useful for object detection.

Let $\text{MSE}_a(D)$ and $\text{CE}_a(D)$ be the mean-squared error and the classification error in percent on data set D of the neural network represented by individual a and let $n_{\text{hidden}}(a)$ and $n_{\text{weights}}(a)$ be the corresponding number of hidden neurons and weights, respectively. The weights of every newly generated offspring a are adapted by gradient-based optimization ("learning","training") of $\text{MSE}_a(D_{\text{train}})$. An improved version of the Rprop algorithm is used for at most 100 iterations of training, see [7, 4]. Training can stop earlier based on the *generalization loss* criterion GL_α as described in [6]. The latter is computed on D_{val} for $\alpha = 5$. Finally, the weight configuration with the smallest $\text{MSE}_a(D_{\text{train}}) + \text{MSE}_a(D_{\text{val}})$ encountered during training is regarded as the outcome of the training process and stored in the genome of the individual a.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 139-144

Then the fitness $\Phi(a)$ of a is given by the linear aggregation

$$\begin{aligned}\Phi(a) &= \gamma_{\mathrm{CE}} \cdot \mathrm{CE}_a(D_{\mathrm{train}} \cup D_{\mathrm{val}}) + \mathrm{MSE}_a(D_{\mathrm{train}} \cup D_{\mathrm{val}}) \\ &+ \gamma_{\mathrm{hidden}} \cdot n_{\mathrm{hidden}}(a) + \gamma_{\mathrm{weights}} \cdot n_{\mathrm{weights}}(a) \ .\end{aligned}$$

supposed to be minimized. The weighting factors are chosen such that typically $\gamma_{\mathrm{CE}} \cdot \mathrm{CE}_a(D_{\mathrm{train}} \cup D_{\mathrm{val}}) \gg \gamma_{\mathrm{hidden}} \cdot n_{\mathrm{hidden}}(a) \approx \gamma_{\mathrm{weights}} \cdot n_{\mathrm{weights}}(a) \gg \mathrm{MSE}_a(D_{\mathrm{train}} \cup D_{\mathrm{val}})$ holds. EP-style tournament selection with 5 opponents is applied to determine the parents for the next generation [5].

A key concept in evolutionary computation is strategy adaptation, i.e., the automatic adjustment of the search strategy during the optimization process. Not all operators might be necessary at all stages of evolution and questions such as when fine-tuning becomes more important than operating on receptive fields cannot be answered in advance. Hence, the application probabilities of the 8 variation operators are adapted using the method described in [5], which is based on the heuristic that recent beneficial modifications are likely to be also beneficial in the following generations.

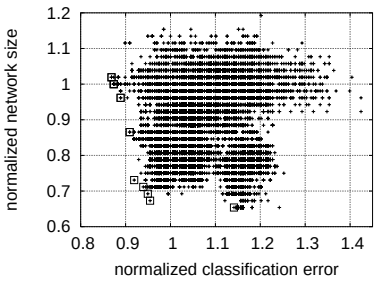
3 Experimental Evaluation

We want to quantify the benefits of hybrid optimization of neural network classifiers and not the performance of the complete Viisage-FaceFINDER® including preprocessing and face recognition. For comparison we trained the *reference topology* 100 times for 2000 iterations using the improved Rprop learning procedure on D_{train}. From all trials and all iterations we selected the network a_{ref} with the smallest classification error on $D_{\mathrm{val}} \cup D_{\mathrm{test}}$, see below. In the following, all results are normalized by the performance of a_{ref}. For example, the normalized classification error of an evolutionary optimized network a is given by $\mathrm{CE}'_a(D) = \mathrm{CE}_a(D)/\mathrm{CE}_{a_{\mathrm{ref}}}(D)$ and the normalized number of hidden neurons by $n'_{\mathrm{hidden}}(a) = n_{\mathrm{hidden}}(a)/52$.

We started 10 trials of the described evolutionary algorithm for 200 generations (i.e., 5025 fitness evaluations per trial). For each evolved network we calculated the value $\mathrm{CE}'(D_{\mathrm{test}})$. Although cross-validation is applied when training the neural networks, the evolutionary optimization may lead to overfitting, in our case it overfits the patterns from $D_{\mathrm{train}} \cup D_{\mathrm{val}}$. Hence, we additionally introduced the data set D_{test} to finally choose models that generalize well. That is, we use D_{test} for some kind of cross-validation of the evolutionary process.[1] For the evolved networks, the classification errors and the corresponding numbers of hidden neurons are depicted in Fig. 3 (left) as coordinates in a plane. Since we are optimizing two goals—size and accuracy—we determine the Pareto set with respect to these two objectives. The Pareto set contains all networks with the property that no other network has occurred that is better w.r.t. one objective and not worse w.r.t. the other. Therefore the Pareto set

[1] When we selected the reference network a_{ref}, we decided in a similar way as we would have picked a solution from the evolved architectures, but taking also D_{val} into account. This is reasonable, since D_{val} was not applied during network training.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 139-144



n'_{hidden} ($n_{\text{hidden}}, n_{\text{weights}}$)	error $\text{CE}'(D_{\text{test}})$	error $\text{CE}'(D_{\text{extern}})$
0.65 (34, 2303)	1.14	1.11
0.67 (35, 4436)	0.95	0.91
0.69 (36, 4477)	0.95	0.94
0.71 (37, 4473)	0.94	0.91
0.73 (38, 4543)	0.92	0.92
0.87 (45, 3110)	0.91	0.92
0.96 (50, 5447)	0.89	0.89
1.00 (52, 5756)	0.90	0.90
1.02 (53, 5860)	0.87	0.91

Figure 3: Evolved solutions. The left plot shows the two objectives, the normalized classification error $\text{CE}'(D_{\text{test}})$ and the normalized number of hidden neurons n'_{hidden}, for all networks of all trials and generations except the first five. The networks not dominated, i.e., those constituting the Pareto set, are highlighted. The right table shows size and performance measures for Pareto optimal solutions

can be considered as the collection of optimal trade-offs between the two objectives our algorithm has evolved. The numbers of hidden neurons of the Pareto optimal networks are reduced by 27-35 %. A generalization performance test on a fourth data set D_{extern}, which is independent from all data used for optimization and final network choice, demonstrates that most of our considerably smaller networks perform as least as good as the expert-designed architecture, see Fig. 3 (right).

4 Conclusion and Outlook

The proposed hybrid evolutionary algorithm successfully tackles the problem of reducing the number of hidden neurons of the face detection network without loss of detection accuracy. The speed of classification whether an image region corresponds to a face or not could be improved by approximately 30 %. By speeding up classification, the rate of complete scans of video-stream images can be increased leading to a more accurate recognition and tracking of persons. Note that almost all of the networks in Fig. 3 (right) have more weights than the initial one, but fewer hidden nodes. Such solutions can not be found by a pure pruning algorithm. Of course, the suggested algorithm can be adapted to the automatic construction of neural networks for any classification task.

Even if the classification error on a fixed additional data set that is not considered for adapting the weights (neither for training nor for early stopping) is—additionally or solely—used in the fitness calculation, the evolved networks would tend to overfit to the data that are responsible for their selection. Therefore we introduced additional data sets to reliably measure generalization performance. Our way of improving the generalization by cross-validation of both, learning and evolution, is an improvement over other methods. Nonetheless, the problem of evolving good generalizing neural networks needs further investigation.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 139-144

Although the results are satisfying, one can think of further enhancements of the described algorithm. It took some time to find a suitable balance in $\Phi(a)$ between the competing objectives of reducing the number of hidden neurons and reducing the classification error. Thus, more advanced methods for evolutionary multi-objective optimization would be a promising extension [2].

References

[1] R. Caruana, S. Lawrence, and C. L. Giles. Overfitting in neural networks: Backpropagation, Conjugate Gradient, and Early Stopping. In *Advances in Neural Information Processing Systems*, volume 13, pages 402–408, Denver, Colorado, 2001. MIT Press.

[2] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[3] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen. An image processing system for driver assistance. *Image and Vision Computing*, 18(5):367–376, 2000.

[4] C. Igel and M. Hüsken. Empirical evaluation of the improved Rprop learning algorithm. *Neurocomputing*, 50(C):105–123, 2003.

[5] C. Igel and M. Kreutz. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks. *Neurocomputing*, 55(1–2):347–361, 2003.

[6] L. Prechelt. Early stopping – but when? In G. B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *LNCS*, chapter 2, pages 57–69. Springer-Verlag, 1999.

[7] M. Riedmiller. Advanced supervised learning in multi-layer perceptrons – From backpropagation to adaptive learning algorithms. *Computer Standards and Interfaces*, 16(5):265–278, 1994.

[8] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, 20(1):23–38, 1998.

[9] Viisage Technology AG. http://www.viisage.com

[10] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[11] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.