Neural information processing in real-world face-recognition applications

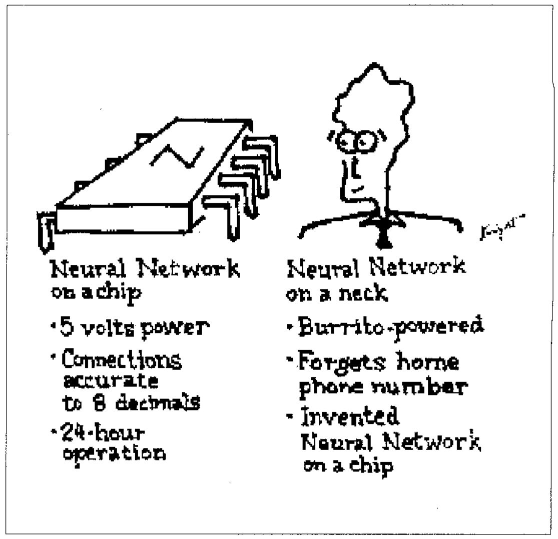
Wolfgang Konen, ZN GmbH

What have neural systems to do with face recognition? Well, we *humans* are some of the most complex neural systems around, and *we* perform face recognition with admirable ease. We recognize thousands of faces learned during our lifetimes. Our visual performance is very robust against changes in a variety of factors—facial expression, head posture or size, illumination, background, and facial aging or partial occlusion of a face. Yet, we know only very little about *how* the brain actually solves its task.

One of the challenges of neural information processing is to achieve, at least partially, a similar performance on systems for automated visual face recognition. This task is certainly not as simple as "Gather some visual input data, throw them into a backpropagation neural network, and hope for the best." On specialized tasks such as gender discrimination from faces, classical neural networks, such as the well-known backpropagation approaches, are quite successful, but these approaches clearly address only part of the problem.[1] In the full face-recognition problem, the complexity of the changeable factors listed earlier is simply too high for a net to solve without a priori knowledge.

Biological neural architectures have taught us several important lessons. The first comes from biologically inspired visual preprocessing in the form of filters that are localized both spatially and in the frequency domain (for example, wavelets, Gabor functions, and Laplace filters). This is not the sole invention of neuroinformatics, but its usefulness has been underscored by the existence of cells with similar behavior in the human visual cortex.

Another connection between today's most successful automated face recognizers and neural information processing reveals itself on a slightly more subtle level. The "Neural Decade"—the period from 1980 on, when so many neural-network applications emerged—has influenced our way of thinking about and assessing problem solutions. We no longer try to build hierarchical rule bases or measure facial attributes such as chin-to-nose distance. Instead, as neural systems would do, we assess the visual feature information more coherently and with more massive parallelism. This is also true for today's two most successful face-algorithms classes: Turk's and Pentland's eigenfaces, in which a face decomposes into sets of facial features that tend to occur together, in parallel, and von der Malsburg's facial graphs,[2] which bind Gabor features and their spatial relations together in a flexible planar graph that can adapt itself globally and locally to best fit into a given facial picture. (von der Marlsburg's research group at the University of Southern California used flexible graphs to win second place in a recent US government face-recognition contest.) For both algorithms, we can establish analogies to special neural architectures (principal component analysis networks and dynamic link architectures).



Commercial applications of the new technology don't care about the underlying architecture or paradigms. The architecture simply needs to work all the time and be easy to use. That was also the essence of ZN's product design plan when we began developing our ZN-Face access-control system. ZN-Face relies on von der Malsburg's graph matching, which is robust enough to deal with the low-quality pictures encountered outside the laboratory when developing automated image acquisition from real-world scenes. (In this way, of course, the underlying neural system's robustness is essential, because otherwise we could not have fulfilled the works-all-the-time requirement.)

At ZN, we developed the complete hardware and software setup for the biometric access-control device. We optimized and adapted the algorithms to the specific verification task—that is, "Is the person in question identical to the cardholder?"—and tested it in the hardware setup, leading to a 99.5% performance verification rate. As with most face recognizers, ZN-Face requires the cooperation of users, who must orient their heads toward the camera during picture acquisition ($\pm 15°$). Today's algorithms can only partially solve the challenge of generalizing from, for example, a half-profile view to a frontal view.

The development of ZN-Face demonstrated the importance of going to market quickly. That's the only way to get the necessary feedback about what components really matter to customers and about what parts to drop in simplifying the user interface. Having a complete hardware prototype also proved critical in selling the idea to potential consumers. Other technical challenges involved further increasing the robustness of the algorithms: even with noisy data from low-cost cameras, the face position and size must be estimated reliably and the verification must still be safe. Overall, the product development cycle was quite fast—one year to the first pilot installation in February 1995. It led to one of the world's first permanent face-recognition systems for commercial use in a non-research institution (a bank's computer center), which was installed in September 1995 and is still running.

In recent years, the emergence of new and reliable verification tools has spurred considerable commercial interest in face recognizers. Face-recognition applications such as for automated teller machines, electronic cash, and safe deposit boxes will probably join access-control applications in the near future. Tools to help witnesses in criminal investigations are also emerging. Witnesses typically must look through thousands of facial images for a match. These tools will sort out the definitely dissimilar faces from the database, helping witnesses to focus their attention on the more likely images. In a recent study, the ZN-Face algorithm also proved capable of comparing facial line drawings with facial gray-level images.[3]

Will successful face recognizers always be neural? Not necessarily. As we have seen, the narrowly defined neural-network para-

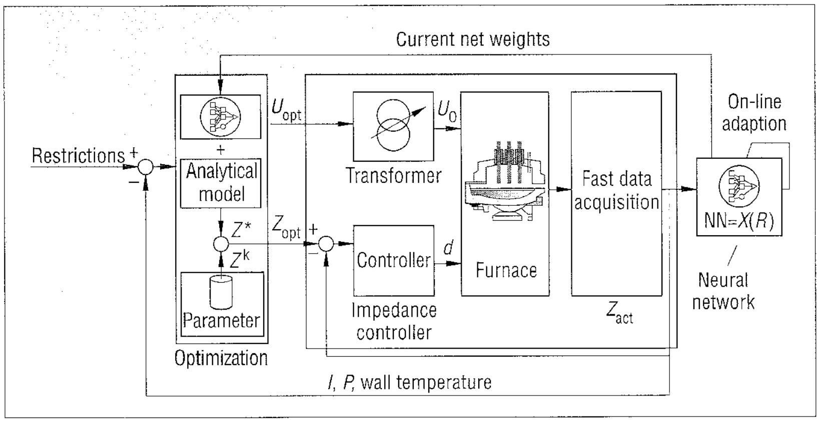


Figure 2. This hybrid model optimizes the operation of an electric arc furnace under constantly changing conditions.

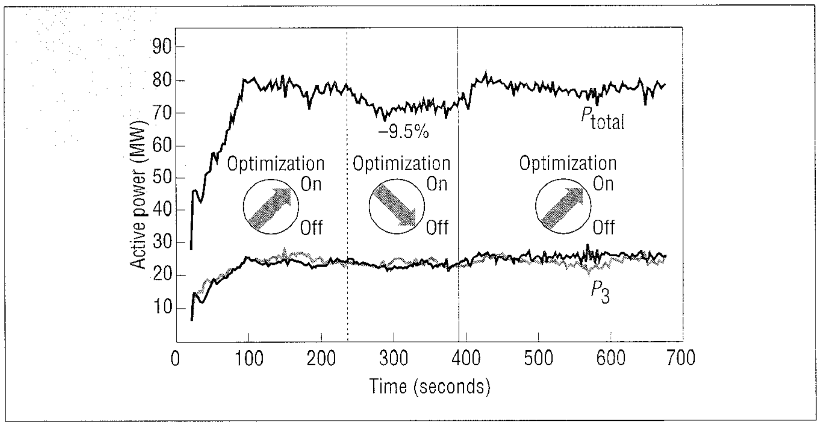


Figure 3. Power curve using the optimization technique at Krupp Nirosta. Neural network-derived optimization increases the active power of the melting process, while satisfying all boundary conditions.

digm "Learn classes from (many) examples" does not fit the face-recognition problem well where we want to learn a new class (a new face) from a few or even a single view. Perhaps a successful face-recognition algorithm will show new ways to learn from single examples and thus influence neural networks. Biological neural networks are still far superior for face recognition—especially for dynamic aspects such as facial motion sequences—and studying how they solve that task will be a worthwhile endeavor for some time to come.

References

1. B.A. Golomb, D.T. Lawrence, and T.J. Sejnowski, "Sexnet: A Neural Network Identifies Sex from Human Faces," in *Advances in Neural Information Processing Systems 3*, D.S. Touretzky, ed., Morgan Kaufman, San Francisco, 1991.
2. M. Lades et al., "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. Computers*, Vol. 42, No. 3, 1993, pp. 300–311.
3. W. Konen, "Comparing Facial Line Drawings with Gray-Level Images," *Proc. Int'l Conf. Artificial Neural Networks*, Springer, Heidelberg, Germany, in press.

Neural networks for steel manufacturing

Martin Schlang, Thomas Poppe, and Otto Gramchow, Siemens AG

For several years, the Industrial and Building Systems Group at Siemens has successfully used neural networks for second-level process automation in basic industries. Worldwide, Siemens currently has more than 20 neural-network applications running in a dozen plants, 24 hours a day. For example, its metallurgy plants use neural-network control applications in their electric arc furnaces (EAFs) and hot rolling mills to predict such factors as rolling force, rolling-stock temperature, natural spread, and short-stroke or electrode positions.

Several aspects of neural networks contribute to their usefulness for the steel industry. First, they speed the development of new applications. In the past, steelmakers had to develop and program special analytical models, a laborious and time-consuming process. Neural networks are simple mathematical structures that gather knowledge by learning from examples, which a computer can do automatically. Besides being so much quicker and easier, neural models also often achieve better performance than do analytical models in practical applications.

Second, neural networks can handle highly nonlinear problems, making them vastly superior to classical linear approaches. Finally, neural network are able to adapt on line. Steel plants experience daily process fluctuations, called the *drift* or *daily condition* by steelworkers, which can wreak havoc for static analytical models. Mathematically, this drift represents some missing input to the network. A special neural topology developed at Siemens can adapt on line to these long-term fluctuations.

Applying our solutions to real-world technical processes at Siemens required that we surmount several challenges, which involved extensive engineering effort. In particular, we needed to

- Improve the control system without discarding existing solutions.
- Cooperate intensively with experts to analyze the importance of the various process inputs.
- Develop user-friendly software and algorithms with few control parameters for the application engineers.
- Develop robust, efficient learning algorithms capable of on-line learning.
- Design additional learning strategies such as initial learning when no data are available before the production starts or when the plant hardware changes suddenly.
- Pre- and postprocess robust data and filter it for the very noisy data typically encountered in steel plants.
- Install data-acquisition systems with fast databases for on-line storage of enormous amounts of data.

Most of our neural solutions involve second-level control, which means the neural networks serve to predict the setpoints of the basic control (Level 1). This also helps guarantee stability in the steel industry's rough environment.