# FORSCHUNGSBERICHTE KÜNSTLICHE INTELLIGENZ

## Semilinear Predictability Minimization Produces Orientation Sensitive Edge Detectors

Jürgen Schmidhuber, Bernhard Foltin

Report FKI-201-94

December 1994

# T U M

TECHNISCHE UNIVERSITÄT MÜNCHEN
Institut für Informatik (H2), D-80290 München, Germany

ISSN 0941-6358

# Forschungsberichte Künstliche Intelligenz

**ISSN 0941-6358**

**Institut für Informatik**
**Technische Universität München**

Die Forschungsberichte Künstliche Intelligenz enthalten vornehmlich Vorab-Veröffentlichungen, spezialisierte Einzelergebnisse und ergänzende Materialien, die seit 1988 in der KI / Kognitionsgruppe am Lehrstuhl Prof. Brauer bzw. 1988-1993 in der KI / Intellektik Gruppe am Lehrstuhl Prof. Jessen entstanden. Im Interesse einer späteren Veröffentlichung wird gebeten, die Forschungsberichte nicht zu vervielfältigen. Alle Rechte und die Verantwortung für den Inhalt des Berichts liegen bei den Autoren, die für kritische Hinweise dankbar sind.

Eine Zusammenstellung aller derzeit lieferbaren FKI-Berichte und einzelne Exemplare aus dieser Reihe können Sie bei folgender Adresse anfordern oder über ftp beziehen:

"FKI"
Institut für Informatik (H2)
Technische Universität München
D-80290 München
Germany

Phone: +49 - 89 - 2105 2406
Telex: tumue d 05-22854
Fax: +49 - 89 - 2105 - 8207
e-mail: fki@informatik.tu-muenchen.de

The "Forschungsberichte Künstliche Intelligenz" series includes primarily preliminary publications, specialized partial results, and supplementary material, written by the members of the AI / Cognition Group at the chair of Prof. Brauer (since 1988) as well as the "Intellektik" Group at the chair of Prof. Jessen (1988-1993). In the interest of a subsequent final publication these reports should not be copied. All rights and the responsibility for the contents of the report are with the authors, who would appreciate critical comments.

You can obtain a list of all available FKI-reports as well as specific papers by writing to the adress below or via ftp:

FTP:
machine: flop.informatik.tu-muenchen.de
or 131.159.8.35
login: anonymous
directory: pub/fki

# SEMILINEAR PREDICTABILITY MINIMIZATION PRODUCES ORIENTATION SENSITIVE EDGE DETECTORS

## Technical Report FKI-201-94

Jürgen Schmidhuber* Bernhard Foltin
Fakultät für Informatik
Technische Universität München
80290 München, Germany

December 24, 1994

**Abstract**

Static real world images are processed by a computationally simple and biologically plausible version of the recent predictability minimization algorithm for unsupervised redundancy reduction. Without a teacher and without any significant pre-processing, the system automatically learns to generate orientation sensitive edge detectors in the first (semilinear) layer.

## 1 INTRODUCTION

Redundancy reduction is widely regarded as an important goal of unsupervised learning. See e.g. [2, 1, 4, 14]. But how to achieve this goal in a massively parallel, local, and biologically plausible way? The simple approach in this paper is based on the principle of **predictability minimization** [12]. A feedforward network with $n$ output units (or code units) sees redundant input patterns. Its goal is to respond with informative but less redundant output patterns (ideally creating a binary factorial code [2] of the input ensemble). The central idea of predictability minimization is: **For each code unit, there is a predictor network that tries to predict it from the remaining $n-1$ code units. But each code unit tries to become as unpredictable as possible.** The only way it can do so is by representing environmental properties that are statistically independent from environmental properties represented by other code units. Predictors and code units co-evolve by fighting each other.

So far, predictability minimization has been tested on artificial data only [6, 13, 14]. Here we study the question: what happens if we apply a biologically plausible, computationally simple, entirely local, and highly parallel variant of predictability minimization to real world images? Can we obtain feature detectors reminiscent of those observed in early visual processing stages of biological systems? Does predictability minimization offer a plausible alternative to previous parallel methods for unsupervised feature detection, e.g. [7, 5, 3, 9]? The following section presents details and results in the context of an application.

*schmidhu@informatik.tu-muenchen.de http://papa.informatik.tu-muenchen.de/mitarbeiter/schmidhu.html

## 2 APPLICATION: IMAGE PROCESSING

Predictability minimization is applied to static black and white images of driving cars (see figure 1). Each image is divided into 566 × 702 square pixels. Each pixel can take on 16 different grey levels represented as integers between 0 and 15.

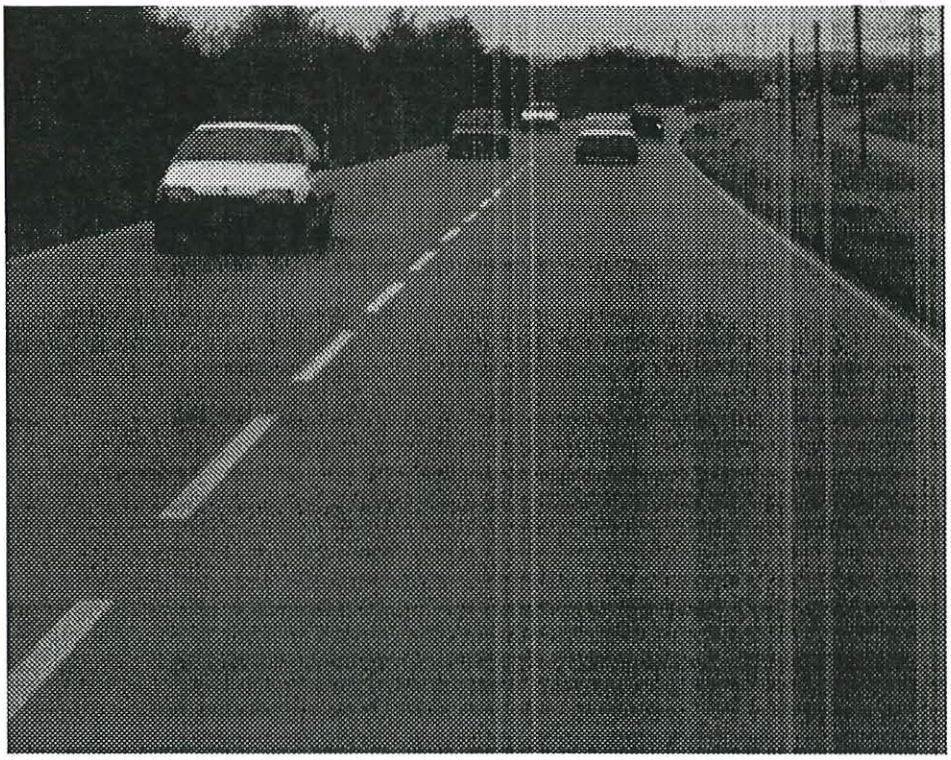

Figure 1: *A typical image from the image data base.*

**Input generation.** There is a circular "input area". Its diameter is 64 pixel widths. There are 32 output units or code units. For each code unit, there is a "bias input unit" with constant activation 1.0, and a circular receptive field of 81 evenly distributed additional input units. The diameter of each receptive field is 20 pixel widths. Receptive fields partly overlap. The positions of code units and receptive fields relative to the input area are fixed. See figure 2. The rotation of the input area is chosen randomly. Its position is chosen randomly within the boundaries of the image. If the position of an input unit is inside the input area, then its activation is the average grey level value of the closest pixel and the four adjacent pixels (see figure 3). Otherwise its activation is zero.

**Input processing.** In response to a given external input pattern, the $i$-th code unit produces an output value $y_i = f(\sum_j w_{ij} x_{ij}) \in [0, 1]$, where $f(x) = \frac{1}{1+e^{-x}}$, $x_{ij}$ is the activation of the $j$-th input unit of the $i$-th code unit, and $w_{ij}$ is the weight on the connection between the $i$-th code unit and its $j$-th input unit (before training, all weights are randomly initialized). The semilinearity is potentially important: successive stages of the system can be used for arbitrary *non-linear* input transformations, while successive stages of linear systems cannot.

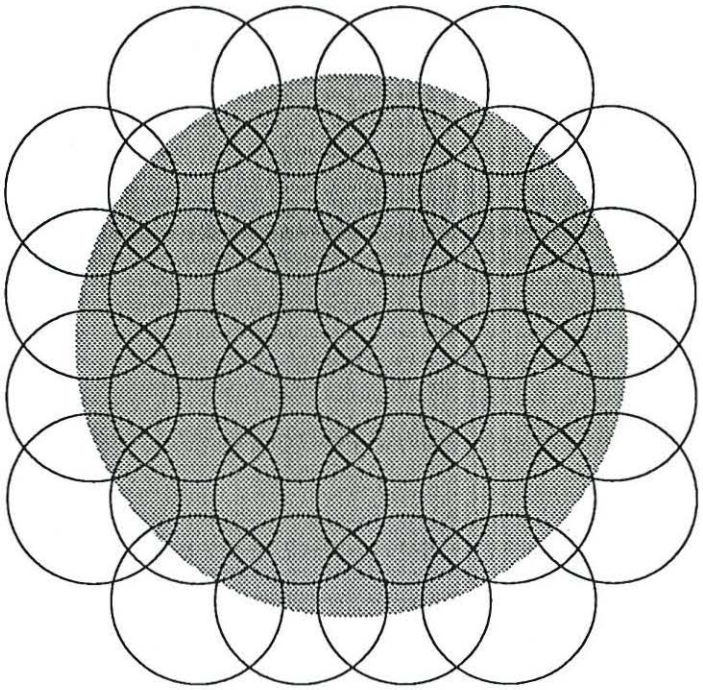

Figure 2: *Small circles represent partly overlapping receptive fields of code units. Their positions are shown relative to the input area (grey). See figure 3 for details of a receptive field.*

**On-line predictability minimization and learning.** For each code unit, there is a semi-linear predictor network that tries to predict it from the remaining code units. $P_i = f(\sum_{k:k\neq i} v_{ik} y_k)$ is the output of the predictor network for code unit $i$ in response to $\{y_k,\ k \neq i\}$, where $v_{ik}$ is the weight on the connection from the $k$-th code unit. Using the delta-rule and on-line learning with learning rate $\eta_P$, the predictor adjusts its weights to decrease

$$(P_i - y_i)^2.$$

Over time, the predictor tends to (semilinearily) approximate the conditional expectation $E(y_i \mid \{y_k, k \neq i\})$. But, simultaneously, the code units try to **maximize** the same (!) objective function the predictors try to minimize: using the inverse delta-rule and on-line learning with learning rate $\eta_C << \eta_P$, the $i$-th code unit adjusts its weights to **increase** $(P_i - y_i)^2$. This can be done in an entirely local manner (since this encourages near-binary code unit activations, the predictor actually tends to approximate the conditional probability $P(y_i \ \textit{close to}\ 1 \mid \{y_k, k \neq i\})$). Predictors and code units try to achieve conflicting goals, thus fighting each other.

**Heuristic simplifications.** To add biological plausibility, the on-line procedure above simplifies the more general method[1] presented in [12]. Heuristic simplifications are: (1) No error signals are propagated through the predictor input units down into the code network. (2) We focus on semilinear networks as opposed to general non-linear ones. (3) Predictors and code units learn simultaneously and in parallel. Also, note that each code unit sees only part of the total input.

**Performance measure.** To measure information throughput, learning is occasionally switched off. Then the number $N$ of pairwise different output patterns in response to 5000 randomly generated input

[1] P. Dayan, R. Zemel and A. Pouget gave some justification of the general method (personal communication, 1992, see also [13, 14]): They observed that maximizing $\sum_i (P_i - y_i)^2$ is equivalent to maximizing $\sum_i VAR(y_i) - \sum_i (P_i - \bar{y_i})^2$ (this expression is a special case of one given in [12]). With binary units, maximization of the first term implies local maximization of information throughput. Maximization of the second (negative) term enforces statistical independence of the code units (assuming perfect predictions), thus encouraging **global** maximization of information throughput.

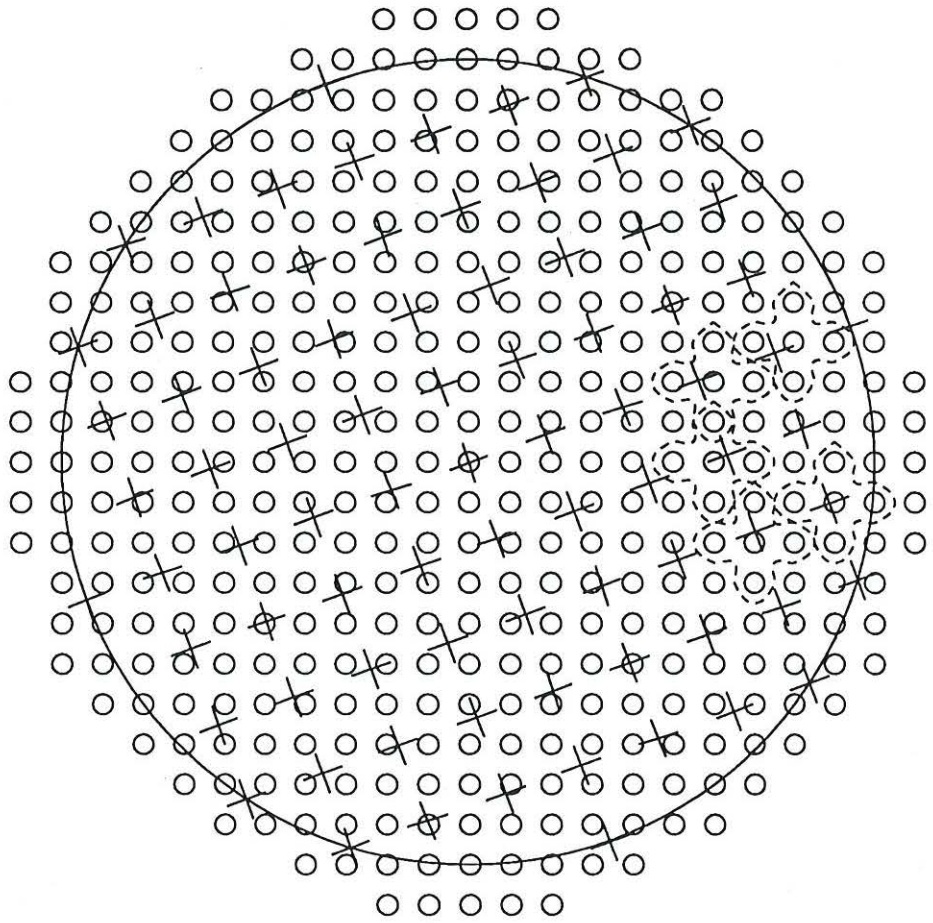

Figure 3: *Details of a receptive field inside the input area. Small circles indicate pixel positions. Crosses indicate positions of rotated input units. The activation of each input unit is the average grey level value of the closest pixel and the four adjacent pixels (indicated by dotted lines).*

patterns is determined (the activation of each output unit is taken to be 0 if below 0.05, 1 if above 0.95, and 0.5 otherwise). The **success rate** is defined by $\frac{N}{5000}$.

**Results.** Figure 4 plots success rate against number of training pattern presentations. Results are shown for various pairs of predictor learning rates $\eta_P$ and code unit learning rates $\eta_C$. For instance, with $\eta_P$ close to 1.0 and $\eta_C$ being one or two orders of magnitude smaller, high success rates are obtained. Although the learning rates do have an influence on learning speed, the basic shapes of the learning curves are similar.

**Edge detectors.** In all cases, it was found that the system creates orientation sensitive edge detectors in an unsupervised manner. Weights corresponding to a typical receptive field (after 5000 pattern presentations) are shown in figure 5. The connections are divided into two groups, one with inhibitory connections, the other one with excitatory connections. Both groups are separated by a "fuzzy" axis through the center of the receptive field. Its rotation angle determines the alignment of the edge leading to maximal response. In general, receptive fields of different code units exhibit different rotation angles. See figure 6.

**Previous work.** We do not claim that predictability minimization is the only parallel method (as opposed to sequential methods, e.g. [10]) that can lead to orientation sensitive edge detectors. For instance, Miller [9] reports the emergence of orientation sensitive cells, but unlike our approach, his approach involves additional prewired input processing. In case of Gaussian input distributions, Linsker's

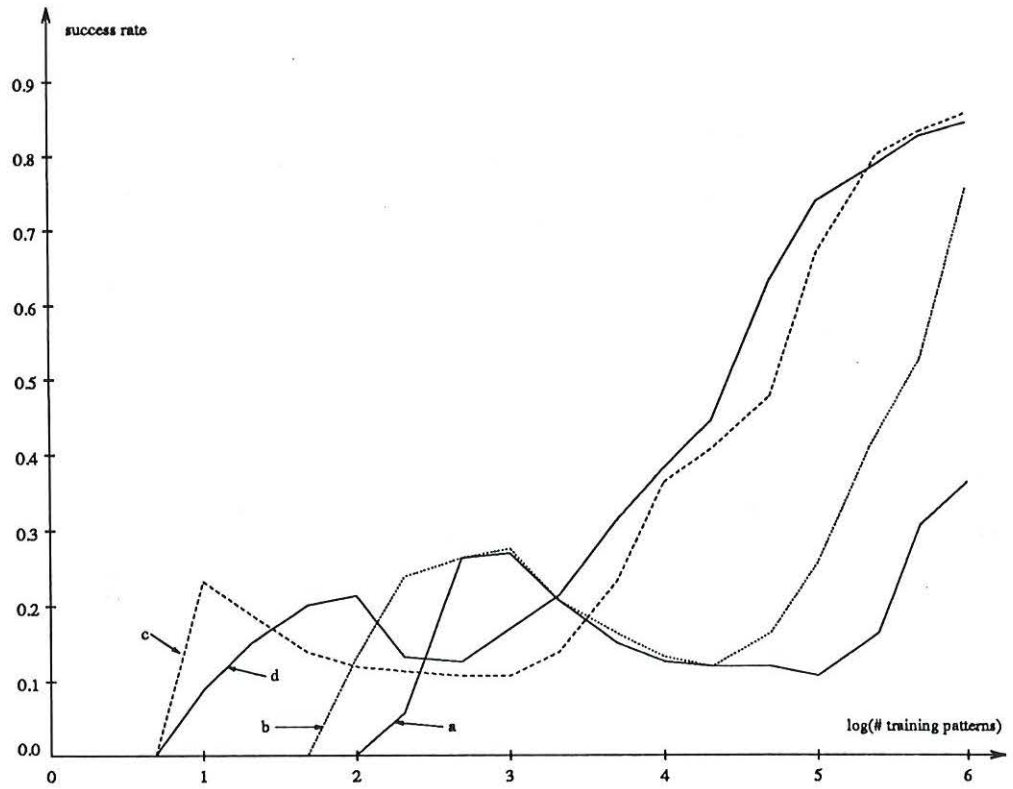


Figure 4: *Success rate plotted against number of training pattern presentations (logarithmic scale). Results are shown for various pairs of predictor learning rates $\eta_P$ and code unit learning rates $\eta_C$. a : $\eta_P = 0.001$, $\eta_C = 0.00004$. b : $\eta_P = 0.01$, $\eta_C = 0.00011$. c : $\eta_P = 0.1$, $\eta_C = 0.005$. d : $\eta_P = 1.0$, $\eta_C = 0.0042$.*

*linear* approach [7] also generates certain kinds of orientation sensitive fields (see also [8]). This holds for more structured input data as well (Linsker, personal communication, 1994). In case of multiple code units, however, Linsker has to compute the derivatives of determinants of covariance matrices, which is biologically implausible. Also, our **semilinear** system appears to have additional potential: successive semilinear stages of our system can be used for arbitrary *non-linear* input transformations, while successive stages of linear systems cannot. Thus, our approach represents an interesting (and simple) alternative. Finally, it is conceivable that Földiák's system [5], Rubner and Tavan's system [11], and Deco and Parra's system [3], might come up with similar edge detectors when applied to real world images. Unlike these approaches (and unlike other similar systems), however, our feedforward net does neither require time consuming settling phases (due to recurrent connections) nor analytic computations of the weight vectors.

**Future research.** We implemented a hierarchy of processing stages, each consisting of code modules and predictors as above. Each stage computes the input to the next stage. Preliminary tests led to feature detectors causing high information throughput. However, the corresponding receptive fields did not exhibit any obvious structure (like the one observed in the first layer). We would like to test the system on large data sets of real world scenes. We expect that this will lead to successively more complex and more specialized feature detectors, hopefully qualitatively related to those observed in biological systems. Unfortunately, however, our current hardware equipment does not permit large scale applications of this kind.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 0.04875 | | | | | |
| | | 0.10272 | 0.07864 | 0.07505 | 0.06264 | 0.03566 | 0.01298 | -0.01175 | |
| | 0.09053 | 0.10269 | 0.08987 | 0.07983 | 0.06118 | 0.02269 | -0.00485 | -0.01780 | -0.03294 |
| | 0.07959 | 0.09131 | 0.08823 | 0.08569 | 0.04267 | 0.00984 | -0.01782 | -0.03361 | -0.04932 |
| | 0.06121 | 0.06666 | 0.07216 | 0.07074 | 0.02815 | -0.01164 | -0.03140 | -0.04119 | -0.05353 |
| 0.04832 | 0.04673 | 0.04110 | 0.03959 | 0.04797 | 0.01408 | -0.02210 | -0.04571 | -0.05508 | -0.07060 -0.07994 |
| | 0.02704 | 0.02191 | 0.02710 | 0.02253 | -0.00241 | -0.03354 | -0.06114 | -0.07843 | -0.09356 |
| | 0.01660 | 0.01455 | 0.00860 | -0.01033 | -0.02982 | -0.05985 | -0.08058 | -0.10864 | -0.12072 |
| | 0.00467 | -0.00012 | -0.01743 | -0.04166 | -0.06492 | -0.08730 | -0.10855 | -0.12776 | -0.12863 |
| | | -0.01927 | -0.03761 | -0.05792 | -0.09105 | -0.11465 | -0.12351 | -0.12495 | |
| | | | | -0.09194 | | | | | |

Figure 5: *Weights corresponding to a typical receptive field (after 5000 pattern presentations). The connections are divided into two groups, one with inhibitory connections, the other one with excitatory connections. Both groups are separated by a "fuzzy" axis through the center of the receptive field. Its rotation angle determines the alignment of the edge leading to maximal response.*

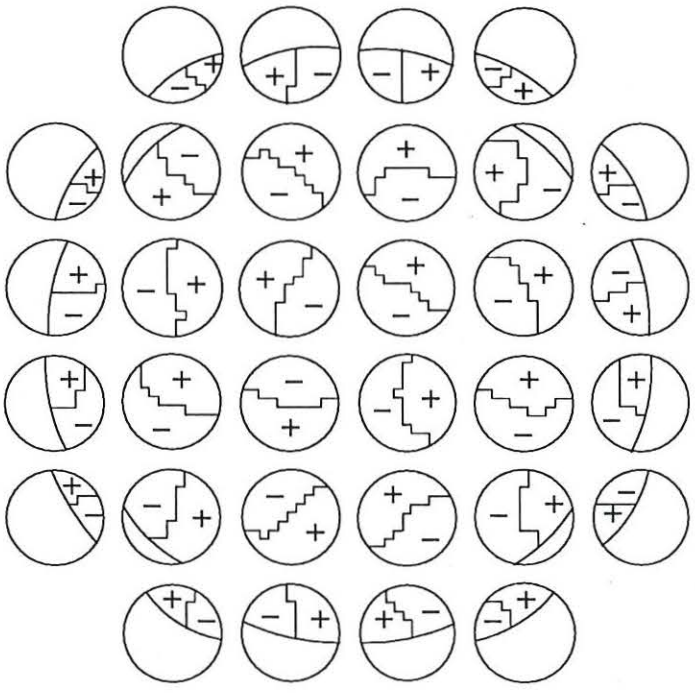

Figure 6: *For all receptive fields, typical post training boundaries between positive and negative weights are shown. Distances between field centers are "blown up" to avoid confusion caused by overlaps.*

# 3 ACKNOWLEDGMENTS

We are grateful to Ralph Linsker, Gustavo Deco, and Peter Dayan, for valuable comments on earlier drafts of this paper.

# References

[1] J. J. Atick, Z. Li, and A. N. Redlich. Understanding retinal color coding from first prinicples. *Neural Computation*, 4:559–572, 1992.

[2] H. B. Barlow, T. P. Kaushal, and G. J. Mitchison. Finding minimum entropy codes. *Neural Computation*, 1(3):412–423, 1989.

[3] G. Deco and L. Parra. Nonlinear features extraction by redundancy reduction with stochastic neural networks. *Submitted to Biological Cybernetics*, 1993.

[4] D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[5] P. Földiák. Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, 64:165–170, 1990.

[6] S. Lindstädt. Comparison of two unsupervised neural network models for redundancy reduction. In M. C. Mozer, P. Smolensky, D. S. Touretzky, J. L. Elman, and A. S. Weigend, editors, *Proc. of the 1993 Connectionist Models Summer School*, pages 308–315. Hillsdale, NJ: Erlbaum Associates, 1993.

[7] R. Linsker. From basic network principles to neural architecture: Emergence of orientation-selective cells. *Proc. Natl. Acad.Sci. USA*, 83, 1986.

[8] D. J. C. MacKay and K. D. Miller. Analysis of Linsker's simulation of Hebbian rules. *Neural Computation*, 2:173–187, 1990.

[9] K. D. Miller. A model for the development of simple cell receptive fields and the ordered arrangement of orientation columns through activity-dependent competition between on- and off-center inputs. *Journal of Neuroscience*, 14(1):409–441, 1994.

[10] J. Rubner and K. Schulten. Development of feature detectors by self-organization: A network model. *Biological Cybernetics*, 62:193–199, 1990.

[11] J. Rubner and P. Tavan. A self-organization network for principal-component analysis. *Europhysics Letters*, 10:693–698, 1989.

[12] J. H. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[13] J. H. Schmidhuber. Netzwerkarchitekturen, Zielfunktionen und Kettenregel. Habilitationsschrift, Institut für Informatik, Technische Universität München, 1993.

[14] J. H. Schmidhuber. Neural predictors for detecting and removing redundant information. In H. Cruse, J. Dean, and H. Ritter, editors, *Adaptive Behavior and Learning*, number 7. Research Group "Prerational Intelligence", Center for Interdisciplinary Research, Universität Bielefeld, 1994.

FKI-201-94 Jürgen Schmidhuber, Bernhard Foltin: Semilinear Predictability Minimization Produces Orientation Sensitive Edge Detectors

ISSN 0941-6358