Universität Ulm
Fakultät für Informatik

On the Complexity of Consistency Problems for Neurons with Binary Weights

Michael Schmitt
Universität Ulm

Nr. 94-01
Ulmer Informatik-Berichte
Januar 1994

On the Complexity of Consistency Problems for Neurons with Binary Weights

Michael Schmitt

Abteilung Neuroinformatik
Universität Ulm
D–89069 Ulm, Germany
mschmitt@neuro.informatik.uni-ulm.de

Abstract

We inquire into the complexity of training a neuron with binary weights when the training examples are Boolean and required to have bounded coincidence and heaviness. Coincidence of an example set is defined as the maximum inner product of two elements, heaviness of an example set is the maximum Hamming-weight of an element. We use both as parameters to define classes of restricted consistency problems and ask for which values they are NP-complete or solvable in polynomial time.

The consistency problem is shown to be NP-complete when the example sets are allowed to have coincidence at least 1 and heaviness at least 4. On the other hand, we give linear-time algorithms for solving consistency problems with coincidence 0 or heaviness at most 3. Moreover, these results remain valid when the threshold of the neuron is bounded by a constant of value at least 2, whereas consistency can be decided in linear time for neurons with threshold at most 1.

We also study maximum consistency problems and obtain NP-completeness for example sets of coincidence at least 1 and heaviness at least 2, whereas we again find linear-time algorithms for the complementary cases. The same is shown to be true for neurons with bounded thresholds the bound being at least 1. On the other hand, maximum consistency can be decided in linear time for a neuron with threshold 0.

> ... to divide each of the difficulties that I was examining into as many parts as might be possible and necessary in order best to solve it.
> ... to conduct my thoughts in an orderly way, beginning with the simplest objects and the easiest to know, in order to climb gradually, as by degrees, as far as the knowledge of the most complex, and even supposing some order among those objects which do not precede each other naturally.
>
> —RENÉ DESCARTES (1637) *Discourse on the Method of Properly Conducting One's Reason and of Seeking the Truth in the Sciences*, translation by F. E. Sutcliffe

1 Introduction

From a computational point of view, learning in neural networks can be regarded as calculating weights that minimize the error of the net being trained with respect to a given set of examples. In formal treatments it is convenient—and in applications often sufficient—to appropriate the hypothesis that input and output values of neurons are restricted to a finite set. Consequently, learning can always be accomplished in finite time by exhaustively searching through the set of network functions.

A useful tool for investigating whether there are more clever learning algorithms is the so-called consistency problem. Stated as a decision problem and associated with a network architecture, it constitutes a non-constructive paradigm of learning to the effect that just the existence of weights producing no error has to be confirmed or rejected. The consistency problem, however, is geared to the situation of one's knowing in advance that the architecture being adapted has a weight configuration agreeing with the whole set of examples. Often the best one can hope for is finding a network with an error as small as possible. This optimization problem finds expression as decision problem in the so-called maximum consistency problem where an instance consists of an example set and a natural number which is to be a lower bound on the number of examples learned correctly. The complexity of consistency and maximum consistency problems for networks computing Boolean functions has been investigated by several theorists. Diverse constraints on architectures and weights have lead to NP-completeness results on the one hand and polynomial-time algorithms on the other [2, 3, 7, 12, 13, 15].

The most restrictive architecture we can think of is a network consisting of a single unit known as McCulloch-Pitts neuron [16]. It has been introduced by McCulloch and Pitts to model the logical behavior of nerve cells and is based on the hypothesis of an all-or-none character of neural activity. The model can be simplified further if we extend the hypothesis to the synaptic weights restricting them to binary values $\{0, 1\}$. The threshold is then allowed to take on values from 0 to n where n is the number of inputs to the neuron. Thus, we obtain a plain computing device that still embodies the threshold principle of neurons and has a non-trivial consistency problem: it has been proved to be NP-complete by Pitt and Valiant [20].

In this report we take a closer look at consistency problems for this simple neuron. Where previous work has mainly concentrated on architectural and weight constraints leaving the examples out of consideration, we are here concerned with the question whether the consistency problem becomes easier when we put constraints on the examples. To that means we introduce two measures for the complexity of example sets: coincidence and heaviness. The former counts the number of components in which two examples have a one in common and takes their maximum over all pairs of examples; the latter counts the number of ones in each example, i.e. its Hamming-weight, and takes the maximum as well. We use both to restrict the possible input space and ask to what extent the degree of simplicity of the permitted example sets has influence on the complexity of the consistency problem.

This is done by splitting the single problem into an infinite class of problems parameterized by coincidence and heaviness of their instances. Our main result is a dichotomy theorem for this problem class. In particular, we show that the consistency problem for neurons with binary weights is NP-complete when the example sets are allowed to have coincidence at least 1 and heaviness at least 4. On the other hand, if the instances are restricted to coincidence 0 or heaviness at most 3 then the problem becomes solvable in polynomial time. The latter is in fact possible in linear time on a random access machine.

We also ask what happens when the threshold of the neuron is bounded from above by a constant. It turns out that NP-completeness is still preserved when the constant is at least 2, whereas linear time is sufficient to decide consistency with a neuron having threshold at most 1.

Concerning the maximum consistency problem we raise the same questions and obtain similar but slightly more restrictive answers. We get NP-completeness for neurons with unbounded threshold with example sets of coincidence 1 and heaviness 2. Again, the complementary cases can be solved in linear time. The statements remain valid for neurons with bounded threshold when the bound is at least 1. On the other hand, maximum consistency with a neuron having threshold 0 can be decided in linear time.

In the following section we give definitions for the basic concepts and state the main theorems. Section 3 is devoted to NP-completeness results. For the problems the complexity of which is left open in that section we present linear-time algorithms in Section 4. Finally, in Section 5 we report on some further work on neurons with binary weights in theory and applications, and conclude with some suggestions and open questions.

2 Definitions and Main Results

2.1 Neurons

Let $\mathcal{B} = (\mathcal{B}_n)_{n\geq 0}$ denote the set of Boolean functions that can be computed by *McCulloch-Pitts neurons with binary weights.* In other words, a function $f : \{0,1\}^n \rightarrow \{0,1\}$ satisfies $f \in \mathcal{B}_n$ if and only if there exist weights $w_1, \ldots, w_n \in \{0,1\}$ and a threshold $t \in \{0, \ldots, n\}$ such that for all $x \in \{0,1\}^n$

$$f(x) = 1 \iff w_1x_1 + \cdots + w_nx_n \geq t.$$

By $\mathcal{B}^{t\leq k}$ we denote the set of functions computed by neurons with binary weights and threshold $t \in \{0, \ldots, k\}$. Thus, $\mathcal{B}_n = \mathcal{B}_n^{t\leq n}$.

In the above definition, a synapse can have a positive strength or no strength at all. In their original model, McCulloch and Pitts also defined inhibitory synapses that, when active, absolutely prevent the activity of the neuron regardless of the sum of the excitatory synapses [16]. Later, the name McCulloch-Pitts neuron was commonly used to refer to a neuron with arbitrary real weights [5].

2.2 Examples

An *example* is a pair (x, a) where x is supposed to be an input value and a to be the corresponding output value for the function being learned. In the context of this report, (x, a) is an example for a Boolean function, i.e. $x \in \{0,1\}^n$ and $y \in \{0,1\}$. Therefore, sets of examples are finite sets of pairs.

Given an example set $S = \{(x^{(1)}, a^{(1)}), \ldots, (x^{(m)}, a^{(m)})\}$, we denote by $\text{pos}(S)$ the set of *positive* examples, and by $\text{neg}(S)$ the set of *negative* examples where

$$\begin{aligned} \text{pos}(S) &= \{x \mid (x,1) \in S\}, \\ \text{neg}(S) &= \{x \mid (x,0) \in S\}. \end{aligned}$$

For the sake of generality, we allow example sets to contain contradictions. The set S is said to be *consistent* if $\text{pos}(S) \cap \text{neg}(S) = \emptyset$. The *domain* of S is defined as $\text{dom}(S) = \text{pos}(S) \cup \text{neg}(S)$.

We consider sets of examples to be instances of decision problems. To narrow down the set of possible instances we introduce two measures. *Coincidence* $\zeta(S)$ and *heaviness* $\eta(S)$ of an example set S are defined as

$$\begin{aligned} \zeta(S) &= \max\{x \cdot y \mid (x,a),(y,b) \in S,\ (x,a) \neq (y,b)\}, \\ \eta(S) &= \max\{x \cdot x \mid (x,a) \in S\}. \end{aligned}$$

where $x \cdot y$ denotes the inner product. In case $|S| \leq 1$ we define $\zeta(S) = 0$, in case $S = \emptyset$ we let $\eta(S) = 0$.

Thus, $\zeta(S)$ is the maximum number of components in which two elements of the domain have a one in common, and $\eta(S)$ is the maximum Hamming-weight, or number of ones, of the elements. Bounded heaviness is a stronger restriction than bounded coincidence because of $\zeta(S) \leq \eta(S)$.

2.3 Consistency Problems

A function f is said to be *consistent with* an example set S if $f(x) = a$ holds for all $(x, a) \in S$. The notion of consistency of an example set with a function must clearly be distinguished from consistency of example sets alone as defined in 2.2. But there might be no confusion because the latter is a unary predicate whereas the former is binary.

The *consistency problem for a set of functions* $\mathcal{F}$ is the problem to decide if a given set of examples S has a function $f \in \mathcal{F}$ that is consistent with S. Alternative terms frequently used in the literature are training problem [7], loading problem [13], and fitting problem [17]. The name consistency problem often occurs in the context of Valiant's pac-learning paradigm [4, 8].

The consistency problem for $\mathcal{B}$ was proved to be NP-complete by Pitt and Valiant [20, Section 5]. Now we make use of coincidence and heaviness to define the classes of consistency problems we are concerned with. We state them in the style of Garey and Johnson [10].

Let C and H be arbitrary natural numbers. The classes of consistency problems for $\mathcal{B}$ with bounded coincidence and heaviness are defined as follows. Let

$\mathcal{B}$-CONSISTENCY WITH COINCIDENCE C
INSTANCE: Example set S with $\zeta(S) \leq C$.
QUESTION: Is there a function $f \in \mathcal{B}$ that is consistent with S?

and

$\mathcal{B}$-CONSISTENCY WITH HEAVINESS H
INSTANCE: Example set S with $\eta(S) \leq H$.
QUESTION: Is there a function $f \in \mathcal{B}$ that is consistent with S?

Finally, the problem $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE C AND HEAVINESS H has only sets with $\zeta(S) \leq C$ and $\eta(S) \leq H$ as instances. For $\mathcal{B}^{t\leq k}$, the problems are defined in an analogous way.

The *maximum consistency problem for a set of functions* $\mathcal{F}$ has as instance a set of examples S and a natural number K. The question is whether there is a subset of S with at least K elements and a function $f \in \mathcal{F}$ consistent with that subset. Amaldi [2] has shown that the problem is NP-complete for single neurons with threshold $t = 0$ and arbitrary weights. The constraint $t = 0$ was removed by Höffgen and Simon [12].

For natural numbers C and H we define maximum consistency problems for $\mathcal{B}$ as follows. Let

MAXIMUM $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE C
INSTANCE: Example set S with $\zeta(S) \leq C$, natural number K.
QUESTION: Is there a subset $S' \subseteq S$ with $|S'| \geq K$ and a function $f \in \mathcal{B}$ that is consistent with S'?

and

MAXIMUM $\mathcal{B}$-CONSISTENCY WITH HEAVINESS H
INSTANCE: Example set S with $\eta(S) \leq H$, natural number K.
QUESTION: Is there a subset $S' \subseteq S$ with $|S'| \geq K$ and a function $f \in \mathcal{B}$ that is consistent with S'?

Again, we combine both to MAXIMUM $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE C AND HEAVINESS H. Finally, we consider all problems with $\mathcal{B}^{t\leq k}$ in place of $\mathcal{B}$.

An algorithm that solves the consistency problem with heaviness H can be used to solve consistency problems with heaviness H' where $H' \leq H$. Thus, consistency problems with bounded heaviness form a class of increasingly harder decision problems. From the definition also follows that NP-completeness of the consistency problem with heaviness H implies NP-completeness of the consistency problem with heaviness H'' where $H'' \geq H$. It is not hard to see that the consistency problem with heaviness

1 has a polynomial-time algorithm. Therefore, interesting questions are whether there are larger constants with polynomial-time solvable consistency problem, and whether there is a constant with NP-complete consistency problem. In the following we shall answer both questions in the affirmative. In particular, we shall prove a dichotomy theorem stating that each problem is NP-complete or solvable in linear time on a random access machine.

Similar considerations can be made with respect to coincidence. But here we have to expect that algorithms for consistency problems with bounded coincidence have to be stronger. In fact, an algorithm that solves the consistency problems with coincidence C will also solve the consistency problem with heaviness C because the instances of the latter problem are a subset of the instances of the former. The results will confirm our expectation. We shall prove that the constant where NP-completeness occurs at the first time with respect to coincidence is smaller than the constant where it occurs with respect to heaviness.

All we have said in the previous two paragraphs about the consistency problem can be repeated with respect to the maximum consistency problem. In general, maximum consistency cannot be easier than consistency because we can reduce the latter to the former letting $K = |S|$. Our results demonstrate that there are indeed NP-complete maximum consistency problems that can be solved in linear time if we require zero error.

The vaguely formulated statements of this subsection are to be made more precise in the following.

2.4 Main Results: Dichotomy Theorems for Consistency Problems

Here we formulate four so-called dichotomy theorems the proofs of which are distributed over Sections 3 and 4. The statements concerning solvability in linear time are based on a random access machine[1] with uniform cost criterion.

Theorem 1 $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE C AND HEAVINESS H *is* NP-*complete whenever* $C \geq 1$ *and* $H \geq 4$. *On the other hand,* $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 0 *is solvable in linear time, as well as* $\mathcal{B}$-CONSISTENCY WITH HEAVINESS H *for all* $H \leq 3$.

We obtain a similar result for neurons with bounded threshold if the threshold is allowed to be at least 2. If the threshold is required to be at most 1 then we get solvability in linear time regardless of coincidence and heaviness.

Theorem 2 *For all* $k \geq 2$, *Theorem 1 holds for* $\mathcal{B}^{t\leq k}$ *in place of* $\mathcal{B}$. *On the other hand,* $\mathcal{B}^{t\leq 1}$-CONSISTENCY WITH COINCIDENCE C *and* $\mathcal{B}^{t\leq 1}$-CONSISTENCY WITH HEAVINESS H *are solvable in linear time for all natural numbers* C *and* H.

Theorem 1 follows from Theorem 6, Corollary 11, and Theorem 12 below. Theorem 2 will be proved by Corollary 7, Theorem 10, and Corollary 13.

The next two theorems are concerned with maximum consistency, at first regarding neurons with arbitrary threshold.

[1] see e.g. [1] for a definition and its relationship to other models of computation

Theorem 3 MAXIMUM $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE C AND HEAVINESS H *is* NP*-complete whenever* $C \geq 1$ *and* $H \geq 2$. *On the other hand,* MAXIMUM $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 0 *is solvable in linear time, as well as* MAXIMUM $\mathcal{B}$-CONSISTENCY WITH HEAVINESS H *for all* $H \leq 1$.

At last, we have the statement corresponding to Theorem 2 above.

Theorem 4 *For all* $k \geq 1$, *Theorem 3 holds for* $\mathcal{B}^{t \leq k}$ *in place of* $\mathcal{B}$. *On the other hand,* MAXIMUM $\mathcal{B}^{t \leq 0}$-CONSISTENCY WITH COINCIDENCE C *and* MAXIMUM $\mathcal{B}^{t \leq 0}$-CONSISTENCY WITH HEAVINESS H *are solvable in linear time for all natural numbers* C *and* H.

Theorem 3 is covered by Theorem 8 and Theorem 14, whereas Theorem 4 is consequence of Corollary 9, Corollary 15, and Lemma 16.

Dichotomy properties are also known for other problem classes, the most famous among them might be k-SAT, the problem of deciding whether a set of clauses with k literals per clause has a satisfying assignment [10, p. 259]. A more general result has been shown by Schaefer for GENERALIZED SATISFIABILITY revealing a dichotomy as well [10, 21]. Finally, we mention GRAPH-K-COLORABILITY [10, p. 191]. One should have in mind that all these dichotomies are proper only if P $\neq$ NP.

3 NP-Complete Consistency Problems

To show NP-completeness of the unrestricted consistency problem for $\mathcal{B}$, Pitt and Valiant defined a reduction from ZERO-ONE INTEGER PROGRAMMING [20]. It turns out that the example sets used therein have coincidence at least $(n-1)/2$. So, nothing can be inferred from their proof concerning constant coincidence and heaviness. We obtain the result by reducing a variant of ONE-IN-THREE 3SAT which we call ALMOST DISJOINT POSITIVE 1-IN-3SAT.

> ALMOST DISJOINT POSITIVE 1-IN-3SAT
> INSTANCE: Set U of variables, collection of subsets $\mathcal{C}$ of U such that each subset $c \in \mathcal{C}$ has $|c| = 3$ and each pair of subsets $c, d \in \mathcal{C}, c \neq d$ satisfies $|c \cap d| \leq 1$.
> QUESTION: Is there a truth assignment $\beta : U \rightarrow \{0,1\}$ such that each subset in $\mathcal{C}$ has exactly one true variable?

POSITIVE 1-IN-3SAT is already known to be NP-complete [10, p. 259]. We show that it remains NP-complete under the requirement of almost disjointness.

Lemma 5 ALMOST DISJOINT POSITIVE 1-IN-3SAT *is* NP*-complete.*

Proof. We give a reduction from POSITIVE 1-IN-3SAT. Let c and d be subsets violating the condition of almost disjointness. Without loss of generality let $c = \{u_1, u_2, u_3\}$ and $d = \{u_2, u_3, u_4\}$. According to the statement of the problem, every satisfying assignment has

$$\beta(u_1) + \beta(u_2) + \beta(u_3) = \beta(u_2) + \beta(u_3) + \beta(u_4),$$

and $\beta(u_1) = \beta(u_4)$ follows. Therefore, one of the variables u_1 and u_4 is redundant, and we may delete one of the clauses c and d. Thus proceeding with all pairs of subsets having two variables in common, we obtain a collection of almost disjoint subsets that has a 1-IN-3SAT assignment if and only if the original collection has such an assignment. $\square$

The following theorem covers the first statement of Theorem 1.

Theorem 6 $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 1 AND HEAVINESS 4 *is* NP-*complete.*

Proof. The proof is by reduction from ALMOST DISJOINT POSITIVE 1-IN-3SAT. Let (U, C) be an instance of the latter where $U = \{u_1, \ldots, u_n\}$. We define a set of examples S where $\mathrm{dom}(S) \subseteq \{0,1\}^{4n+2}$. For every $V \subseteq U$ let $\overline{1}_V$ denote the element $x \in \{0,1\}^n$ where $x_i = 1$ iff $u_i \in V$. We also write $\overline{0}$ instead of $\overline{1}_\emptyset$. The components $1, \ldots, 4n+2$ are partitionend into six blocks, the first four comprising n components each, the last two consisting both of a single component.

For each variable $u_i \in U$ we define four examples

$$(\overline{1}_{\{u_i\}}, \overline{1}_{\{u_i\}}, \overline{0}, \overline{0}, 0, 0) \in \mathrm{neg}(S) \tag{1}$$
$$(\overline{0}, \overline{0}, \overline{1}_{\{u_i\}}, \overline{1}_{\{u_i\}}, 0, 0) \in \mathrm{neg}(S) \tag{2}$$
$$(\overline{1}_{\{u_i\}}, \overline{0}, \overline{0}, \overline{1}_{\{u_i\}}, 1, 0) \in \mathrm{pos}(S) \tag{3}$$
$$(\overline{0}, \overline{1}_{\{u_i\}}, \overline{1}_{\{u_i\}}, \overline{0}, 1, 0) \in \mathrm{pos}(S). \tag{4}$$

Each clause $c \in C$ is represented by two examples

$$(\overline{1}_c, \overline{0}, \overline{0}, \overline{0}, 0, 1) \in \mathrm{pos}(S) \tag{5}$$
$$(\overline{0}, \overline{1}_c, \overline{0}, \overline{0}, 0, 0) \in \mathrm{pos}(S). \tag{6}$$

Finally, there are three examples

$$(\overline{0}, \overline{0}, \overline{0}, \overline{0}, 1, 1) \in \mathrm{pos}(S) \tag{7}$$
$$(\overline{0}, \overline{0}, \overline{0}, \overline{0}, 1, 0) \in \mathrm{neg}(S) \tag{8}$$
$$(\overline{0}, \overline{0}, \overline{0}, \overline{0}, 0, 1) \in \mathrm{neg}(S). \tag{9}$$

Obviously, $\zeta(S) = 1$, $\eta(S) = 4$, and S can be computed from (U, C) in polynomial time. We show that C has a 1-IN-3SAT assignment if and only if there exists a function in $\mathcal{B}_{4n+2}$ that is consistent with S.

For the only-if part, let $\beta : U \to \{0,1\}$ be an assignment such that each $c \in C$ has exactly one true variable. We define the weight vector $(w_1, \ldots, w_{4n+2}; t)$ by

$$(\overline{1}_{\beta^{-1}(1)}, \overline{1}_{\beta^{-1}(0)}, \overline{1}_{\beta^{-1}(1)} \overline{1}_{\beta^{-1}(0)}, 1, 1; 2),$$

where $\beta^{-1}(a) = \{x \mid \beta(x) = a\}$. It can easily be verified that the function represented thereby is consistent with S.

Conversely, let $(w_1, \ldots, w_{4n+2}; t)$ be consistent with S. From (7), (8), and (9) we obtain

$$\begin{aligned} w_{4n+1} + w_{4n+2} &\geq t \\ w_{4n+1} &< t \\ w_{4n+2} &< t, \end{aligned}$$

from which $w_{4n+1} > 0$ and $w_{4n+2} > 0$ follow. Thus we have

$$w_{4n+1} = w_{4n+2} = 1 \text{ and } t = 2.$$

From (1) to (4) for every $i \in \{1, \ldots, n\}$

$$\begin{array}{lclcll} w_i & + & w_{n+i} & & & < 2 \\ w_{2n+i} & + & w_{3n+i} & & & < 2 \\ w_i & + & w_{3n+i} & + & 1 & \geq 2 \\ w_{n+i} & + & w_{2n+i} & + & 1 & \geq 2 \end{array}$$

can be derived. These inequalities imply the identities

$$\begin{array}{lclcl} & & w_i & = & w_{2n+i} \\ w_{2n+i} & + & w_{3n+i} & = & 1 \\ w_i & + & w_{n+i} & = & 1. \end{array} \tag{10}$$

Let $\beta : U \to \{0, 1\}$ be defined by

$$\beta(u_i) = w_i \quad \text{for } i \in \{1, \ldots, n\}.$$

Then by virtue of (5), (6), and (10), we obtain for each clause $c = \{u_j, u_k, u_l\} \in \mathcal{C}$

$$\beta(u_j) + \beta(u_k) + \beta(u_l) = 1.$$

Consequently, β is a 1-IN-3SAT assignment for $\mathcal{C}$. □

The proof shows that the reduction can be defined whenever the threshold is allowed to take on the value 2. Thus, NP-completeness for $\mathcal{B}^{t \leq k}$ where $k \geq 2$ follows as a corollary.

Corollary 7 *For all $k \geq 2$, $\mathcal{B}^{t \leq k}$-*CONSISTENCY WITH COINCIDENCE 1 AND HEAVINESS 4 *is* NP*-complete.*

In the remainder of the section we address maximum consistency problems.

Theorem 8 MAXIMUM $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 1 AND HEAVINESS 2 *is* NP*-complete.*

Proof. The proof is an adaptation of the proof of Theorem 3.1 in Höffgen and Simon [12]. They have established NP-completeness of the maximum consistency problem for neurons with arbitrary weights. Their result remains valid if the weights are to be from the set $\{-1, 1\}$. It turns out that a slight modification of their construction is sufficient for our purpose. The reduction is from VERTEX COVER [10, p. 190]. Let $(V,E),K$ be an instance of the latter where $V=\{v_1,\dots,v_n\}$. The set of examples S where $\mathrm{dom}(S)\subseteq\{0,1\}^{2n}$ is constructed as follows.

For each $v_i\in V$ we define a vertex example

$$(\overline{1}_{\{v_i\}},\overline{1}_{\{v_i\}})\ \in\ \mathrm{neg}(S).$$

For each $e\in E$ we define two edge examples

$$\begin{aligned}(\overline{1}_e,\overline{0})\ &\in\ \mathrm{pos}(S),\\ (\overline{0},\overline{1}_e)\ &\in\ \mathrm{pos}(S).\end{aligned}$$

Obviously, $\zeta(S)=1$, $\eta(S)=2$, and S can be computed in polynomial time. We claim that (V,E) has a vertex cover of size at most K if and only if there exists a function in $\mathcal{B}_{2n}$ that is consistent with at least $|S|-K$ examples.

Let V' be a vertex cover with at most K elements. We define the weight vector $(w_1,\dots,w_{2n};t)$ as

$$t=1\quad\text{and}\quad w_i=w_{n+i}=\begin{cases}1 & \text{if } v_i\in V'\\ 0 & \text{otherwise}\end{cases}\quad\text{for } i=1,\dots,n.$$

If $v_i\in V'$ then the corresponding vertex example is classified wrongly. The rest of the vertex examples and all edge examples are mapped correctly. The latter holds because V' is a vertex cover. Thus, the vector is consistent with at least $|S|-K$ elements. The remainder of the proof proceeds just like in Höffgen and Simon. We include it for completeness.

Let $(w_1,\dots,w_{2n};t)$ be a weight vector consistent with a subset of S of size at least $|S|-K$. Construct a set of vertices V' as follows. For each vertex example wrongly classified put the corresponding vertex into V'; for each edge example wrongly classified choose arbitrarily one of the vertices the edge is incident with and put it into V'. Consequently, V' contains at most K elements. To show that it is a vertex cover let $\{v_i,v_j\}$ be an edge where $\{v_i,v_j\}\cap V'=\emptyset$. But then the weight vector is consistent with the two vertex examples corresponding to v_i and v_j, as well as with the two edge examples corresponding to $\{v_i,v_j\}$. Thus we have

$$\begin{aligned}w_i+w_{n+i}+w_j+w_{n+j}\ &<\ 2t\\ w_i+w_j+w_{n+i}+w_{n+j}\ &\geq\ 2t\end{aligned}$$

and, by contradiction, V' is a vertex cover. □

As can be seen from the proof, a threshold of value 1 is always sufficient for the reduction. Thus, we obtain the counterpart of Corollary 7 above.

Corollary 9 *For all* $k\geq 1$, MAXIMUM $\mathcal{B}^{t\leq k}$-CONSISTENCY WITH COINCIDENCE 1 AND HEAVINESS 2 *is* NP-*complete.*

4 Consistency Problems Solvable in Linear Time

In the following, we show that all consistency problems not yet considered above have polynomial-time algorithms. Consequently, the minimum values of coincidence and heaviness required in the NP-completeness proof of Theorem 6 are optimal if P $\neq$ NP. Furthermore, the algorithms presented below to witness polynomiality are in fact linearly time-bounded on a random access machine and therefore optimal for this model of computation. Thus, we not only obtain a complete characterization of the complexity of consistency problems for $\mathcal{B}$ and $\mathcal{B}^{t\leq k}$ with bounded coincidence and heaviness, but also the striking result that, with respect to a random access machine, the problem classes solely consist of elements of the highest and—if we disregard sublinear time—lowest complexity in NP.

We start this section from where we have ended the previous one considering neurons with bounded threshold, now bounding the threshold by 1. The following result is valid for sets of examples with arbitrary coincidence and heaviness.

Theorem 10 $\mathcal{B}^{t\leq 1}$-CONSISTENCY *is solvable in linear time.*

Proof. We claim the following algorithm to compute a weight vector with threshold $t \leq 1$ consistent with a given set of examples S for all S that have such a weight vector.

> **Algorithm** $\mathcal{B}^{t\leq 1}$-CONSISTENCY(S):
> **if** neg(S) $= \emptyset$ **then**
> $t := 0$;
>   **for** $i := 1, \ldots, n$ **do**
> $w_i := 0$
>   **endfor**
> **else**
> $t := 1$;
>   **for** $i := 1, \ldots, n$ **do**
> $w_i := 0 \iff \exists y \in \text{neg}(S)[y_i = 1]$
>   **endfor**
> **endif.**

Clearly, the algorithm runs in linear time on a random access machine. The procedure is based on the observation that t certainly has to be 1 if neg(S) $\neq \emptyset$, but then we must ensure $w \cdot x = 0$ for all $x \in$ neg(S). □

If a set of examples S with $\zeta(S) = 0$ has a consistent function from $\mathcal{B}$ then it also has a consistent function from $\mathcal{B}^{t\leq 1}$. Therefore, we can employ Algorithm $\mathcal{B}^{t\leq 1}$-CONSISTENCY to decide $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 0.

Corollary 11 $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 0 *is solvable in linear time.*

Up to here we have covered all statements of Theorem 1 except the last one which is concern of the following.

Theorem 12 $\mathcal{B}$-CONSISTENCY WITH HEAVINESS 3 *is solvable in linear time.*

Proof. Consider the following decision algorithm.

```
Algorithm B-CONSISTENCY(S):
begin
I.   (w; t) := B^{t≤1}-CONSISTENCY(S);
     if (w; t) is consistent with S then
       output("yes");
       stop
     endif;
II.  t := 3;
     for i := 1, ..., n do
       w_i := 1  <=>  ∃x ∈ pos(S)[x_i = 1]
     endfor;
     if (w; t) is consistent with S then
       output("yes");
       stop
     endif;
III. (* let C be the set of clauses over the set of variables W = {w_1, ..., w_n}
     constructed as follows: *)
     1. C := ∅;
     2. for all x ∈ dom(S) do
            I := {i | x_i = 1};
            (a) if |I| = 3 then
                    {j, k, l} := I;
                    if x ∈ pos(S) then
                      C := C ∪ { {w_j, w_k}, {w_j, w_l}, {w_k, w_l} }
                    elsif x ∈ neg(S) then
                      C := C ∪ { {¬w_j, ¬w_k}, {¬w_j, ¬w_l}, {¬w_k, ¬w_l} }
                    endif
                endif;
            (b) if |I| = 2 then
                    {j, k} := I;
                    if x ∈ pos(S) then
                      C := C ∪ { {w_j}, {w_k} }
                    elsif x ∈ neg(S) then
                      C := C ∪ { {¬w_j, ¬w_k} }
                    endif
                endif;
            (c) if |I| = 1 then
                    {j} := I;
                    if x ∈ pos(S) then
                      C := C ∪ { {w_j}, {¬w_j} }
                    endif
                endif
        endfor;
     3. if there is a truth assignment β : W → {0, 1} such
```

that each clause in C has at least one true literal **then**
output("yes")
else
output("no")
endif
end.

Part I, II, and III of the algorithm make a distinction between the possible values for the threshold, namely $t \leq 1$, $t \geq 3$, and $t = 2$ respectively. Part I employs Algorithm $\mathcal{B}^{t\leq 1}$-CONSISTENCY defined in the proof of Theorem 10. By virtue of this, the output is correct if S has a consistent function from $\mathcal{B}^{t\leq 1}$. Part II tries to find a solution with $t \geq 3$, and thus without loss of generality with $t = 3$ because S satisfies $\eta(S) \leq 3$. In this case we have to ensure $w \cdot x = 3$ for all $x \in \text{pos}(S)$ what is done by the statement. At the beginning of part III we know that the threshold has to be 2. Then a set of clauses is constructed that is satisfiable if and only if there is a weight vector with $t = 2$ consistent with S. Satisfiability is verified by statement III.3.

Finally, we have to make sure of the claimed running-time. Parts I and II can easily be seen to run in linear time when we use Theorem 10. The construction of the set of clauses in part III is possible in linear time as well. Each clause contains at most two literals. Thus, we are solving a 2-SAT problem in statement III.3. A linear-time algorithm for 2-SAT has been outlined by Even, Itai, and Shamir [9]. Aspvall, Plass, and Tarjan [6] give a full description of an algorithm that even decides satisfiability of quantified 2-SAT formulas in linear time and uses the strong components algorithm by Tarjan [1, 24]. All in all, Algorithm $\mathcal{B}$-CONSISTENCY runs in linear time. □

Herewith, Theorem 1 is proved. Obviously, Algorithm $\mathcal{B}$-CONSISTENCY can be used to solve the consistency problem for $\mathcal{B}^{t\leq k}$ with heaviness 3 for all $k \geq 2$. (If $k = 2$ then part II is redundant.) Furthermore, for all $k \geq 2$, $\mathcal{B}^{t\leq k}$-CONSISTENCY WITH COINCIDENCE 0 is equivalent to $\mathcal{B}^{t\leq 1}$-CONSISTENCY. Thus, by virtue of Theorem 10 and 12, we can formulate the following corollary which completes the proof of Theorem 2.

Corollary 13 *For all $k \geq 2$,* $\mathcal{B}^{t\leq k}$-CONSISTENCY WITH COINCIDENCE 0 *as well as* $\mathcal{B}^{t\leq k}$-CONSISTENCY WITH HEAVINESS 3 *is solvable in linear time.*

We now turn to maximum consistency problems, starting with the counterpart to Corollary 11 and Theorem 12.

Theorem 14 MAXIMUM $\mathcal{B}$-CONSISTENCY WITH COINCIDENCE 0 *as well as* MAXIMUM $\mathcal{B}$-CONSISTENCY WITH HEAVINESS 1 *is solvable in linear time.*

Proof. We give an informal description of the algorithm. Let

$$t = 1 \quad \text{and} \quad w_i = \begin{cases} 1 & \text{if } \exists x \in \text{pos}(S)[x_i = 1] \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \ldots, n.$$

If $\zeta(S) = 0$ then all examples are classified correctly unless $0^n \in \text{pos}(S)$ where we have consistency with $|S| - 1$ examples. The latter can be improved only if $\text{neg}(S) = \emptyset$ when we let $t = 0$.

If $\eta(S) = 1$ then we proceed the same way, but we have to take into account that S may be inconsistent due to contradictory examples $(x, 0)$ and $(x, 1)$. Thus, we append an additional step where we count the examples the generated weight vector is consistent with, and compare the number with the bound K. $\square$

Herewith, Theorem 3 is proved. In the previous algorithm, a weight vector with threshold at most 1 is generated as solution. This implies the following statement.

Corollary 15 *For all $k \geq 1$,* MAXIMUM $\mathcal{B}^{t\leq k}$-CONSISTENCY WITH COINCIDENCE 0 *as well as* MAXIMUM $\mathcal{B}^{t\leq k}$-CONSISTENCY WITH HEAVINESS 1 *is solvable in linear time.*

Being left with maximum consistency for $\mathcal{B}^{t\leq 0}$, we are about to complete the proof of Theorem 4.

Lemma 16 MAXIMUM $\mathcal{B}^{t\leq 0}$-CONSISTENCY *is solvable in linear time.*

Proof. If the threshold is equal to 0 then everything is classified as positive. Thus, consistency with at least K examples in S is equivalent to $\text{pos}(S) \geq K$. $\square$

5 Concluding Remarks

Though neurons with binary weights constitute a fairly simple model of neural computation, they have been subject of experimental and theoretical research almost from the very beginning of the artificial neural network era. The learning matrices of Steinbuch consisted of conditioned connections that, in their simplest version, were able to attain two possible conductance values during a learning phase [22, 23]. Willshaw *et al.* used neurons with binary weights as building blocks for networks being able to perform pattern association [27] and generalization tasks [26]. Information storage capacity for Hebbian learning in these models has been calculated by Palm [18] (see also the review [19] where binary synapses are compared to arbitrary synapses and arbitrary learning rules). Venkatesh examined algorithms for finding binary weights of a single neuron consistent with an example set in a distribution dependent manner [25]. An inconsistent algorithm for pac-learning such a neuron under certain distributions has been presented by Golea and Marchand [11].

The previous short list of references is by no means intended to be complete, let alone unbiased. At least, we consider it sufficient to give evidence for the perseverating interest in simple neurons.

With our work, we aimed to investigate sufficient and necessary conditions for the existence of efficient learning algorithms. The approach was to introduce criteria for the complexity of example sets to define simpler problem instances. As results we obtained exact characterizations of the dividing lines between polynomial-time solvable and NP-complete consistency problems as well as maximum consistency problems. Figure 1 is to illustrate the dichotomies of the problem classes.

For further investigations there seem to emerge two lines of thought, one concerning the architecture, the other concerning the problem instances. With respect to the latter, we do not claim our choosing of restrictions to be the only one that makes

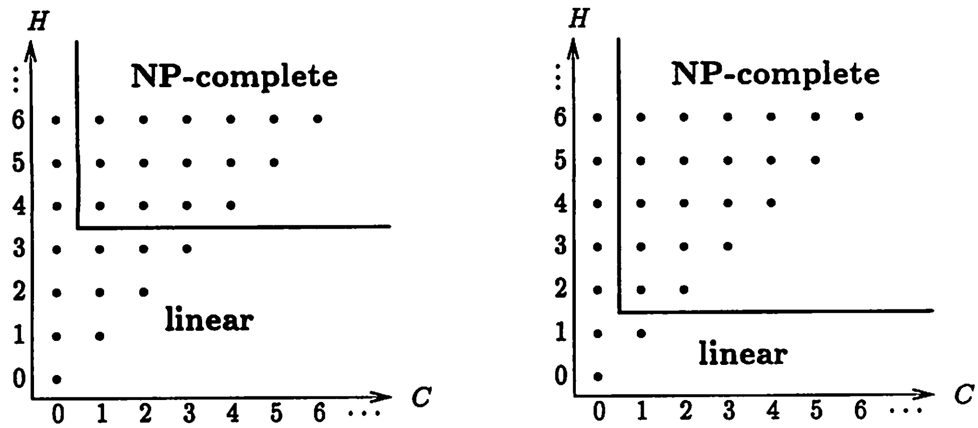

Figure 1: On the left, consistency problems with coincidence C and heaviness H for architectures $\mathcal{B}$ and $\mathcal{B}^{t\leq k}$ where $k \geq 2$ (Theorem 1 and 2); on the right, the corresponding maximum consistency problems for $\mathcal{B}$ and $\mathcal{B}^{t\leq k}$ where $k \geq 1$ (Theorem 3 and 4). The term *linear* refers to time complexity on a random access machine with uniform measure.

sense[2]. One has to look for further characterizations, for example, by taking into account the label, i.e. the second component, of the examples. With respect to the latter, one conclusion can already be drawn from our results. All assertions remain valid if the example sets are required to be monotonous, that is, if $x \leq y$ implies $a \leq b$ for all $(x, a), (y, b) \in S$. This is true because all functions represented by the neurons here considered are monotonous.

As far as the architecture is concerned, it is worth considering whether the results can be extended to neurons with fixed threshold $\mathcal{B}^{t=k}$. We are confident that the methods here provided are strong enough to show NP-completeness where coincidence is kept low while heaviness obviously increases with k.

Another way starts from our constraining the weights to binary values $\{0, 1\}$. It might not be difficult to show that, at least for the architecture with unbounded threshold, any pair of values can be chosen without changing the results. Now the question arises, to what degree, and if at all, coincidence and heaviness have to be modified when we permit sets of larger cardinality. We conjecture that the discrepancy between consistency and maximum consistency evident from Figure 1 increases. This is due to the following observation. It is not hard to see from the proof of Theorem 8 that maximum consistency with coincidence 1 and heaviness 2 is NP-complete even if the weights are allowed to take on arbitrary values. Consistency for neurons with unconstrained weights, however, can be decided in polynomial time using algorithms for Linear Programming [14]. Therefore, with respect to arbitrary weights there is no dichotomy for consistency problems, whereas the dichotomy for maximum consistency coincides with the right-hand side of Figure 1.

[2]In any case, this is not justified until after the results.

References

[1] A. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] E. Amaldi, *On the complexity of training Perceptrons*, in Artificial Neural Networks, T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, eds., Elsevier Science Publishers B.V. (North-Holland), 1991, pp. 55–60.

[3] E. Amaldi and V. Kann, *The complexity and approximability of finding maximum feasible subsystems of linear relations*, Tech. Rep. ORWP 93/11, École Polytechnique Fédérale de Lausanne, Département de Mathématiques, August 1993.

[4] M. Anthony and N. Biggs, *Computational Learning Theory*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, 1992.

[5] M. A. Arbib, *Brains, Machines, and Mathematics*, Springer-Verlag, New York, second ed., 1987.

[6] B. Aspvall, M. F. Plass, and R. E. Tarjan, *A linear-time algorithm for testing the truth of certain quantified Boolean formulas*, Information Processing Letters, 8 (1979), pp. 121–123.

[7] A. L. Blum and R. L. Rivest, *Training a 3-node neural network is NP-complete*, Neural Networks, 5 (1992), pp. 117–127.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, *Learnability and the Vapnik-Chervonenkis dimension*, Journal of the Association for Computing Machinery, 36 (1989), pp. 929–965.

[9] S. Even, A. Itai, and A. Shamir, *On the complexity of timetable and multicommodity flow problems*, SIAM Journal on Computing, 5 (1976), pp. 691–703.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

[11] M. Golea and M. Marchand, *On learning perceptrons with binary weights*, Neural Computation, 5 (1993), pp. 767–782.

[12] K.-U. Höffgen and H.-U. Simon, *Robust trainability of single neurons*, in Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, ACM Press, New York, 1992, pp. 428–439.

[13] J. S. Judd, *Neural Network Design and the Complexity of Learning*, The MIT Press, Cambridge, MA, 1990.

[14] H. Karloff, *Linear Programming*, Birkhäuser, Boston, 1991.

[15] J.-H. Lin and J. S. Vitter, *Complexity results on learning by neural nets*, Machine Learning, 6 (1991), pp. 211–230.

[16] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics, 5 (1943), pp. 115–133.

[17] B. K. Natarajan, *Machine Learning: A Theoretical Approach*, Morgan Kaufmann, San Mateo, CA, 1991.

[18] G. Palm, *On associative memory*, Biological Cybernetics, 36 (1980), pp. 19–31.

[19] G. Palm, *Memory capacities of local rules for synaptic modification: A comparative review*, Concepts in Neuroscience, 1 (1991), pp. 97–128.

[20] L. Pitt and L. G. Valiant, *Computational limitations on learning from examples*, Journal of the Association for Computing Machinery, 35 (1988), pp. 965–984.

[21] T. J. Schaefer, *The complexity of satisfiability problems*, in Proceedings of the 10th Annual ACM Symposium on Theory of Computing, 1978, pp. 216–226.

[22] K. Steinbuch, *Die Lernmatrix*, Kybernetik, 1 (1961), pp. 36–45.

[23] K. Steinbuch and U. A. W. Piske, *Learning matrices and their applications*, IEEE Transactions on Electronic Computers, 12 (1963), pp. 846–862.

[24] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM Journal on Computing, 1 (1972), pp. 146–160.

[25] S. S. Venkatesh, *On learning binary weights for majority functions*, in Proceedings of the Fourth Annual Workshop on Computational Learning Theory, L. G. Valiant and M. K. Warmuth, eds., Morgan Kaufmann, San Mateo, CA, 1991, pp. 257–266.

[26] D. J. Willshaw, *A simple network capable of inductive generalization*, Proc. R. Soc. Lond. B, 182 (1972), pp. 233–247.

[27] D. J. Willshaw, O. Buneman, and H. C. Longuet-Higgins, *Non-holographic associative memory*, Nature, 222 (1969), pp. 960–962.

Liste der bisher erschienenen Ulmer Informatik-Berichte

Einige davon sind per FTP von ftp.informatik.uni-ulm.de erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm

Some of them are available by FTP from ftp.informatik.uni-ulm.de
Reports marked with * are out of print

91-01 KER-I KO, P. ORPONEN, U. SCHÖNING, O. WATANABE
Instance Complexity

91-02* K. GLADITZ, H. FASSBENDER, H. VOGLER
Compiler-Based Implementation of Syntax-Directed Functional Programming

91-03 ALFONS GESER
Relative Termination

91-04* J. KÖBLER, U. SCHÖNING, J. TORAN
Graph Isomorphism is low for PP

91-05 JOHANNES KÖBLER, THOMAS THIERAUF
Complexity Restricted Advice Functions

91-06 UWE SCHÖNING
Recent Highlights in Structural Complexity Theory

91-07 F. GREEN, J. KÖBLER, J. TORAN
The Power of Middle Bit

91-08* V.ARVIND, Y. HAN, L. HAMACHANDRA, J. KÖBLER, A. LOZANO, M. MUNDHENK, A. OGIWARA, U. SCHÖNING, R. SILVESTRI, T. THIERAUF
Reductions for Sets of Low Information Content

92-01 VIKRAMAN ARVIND, JOHANNES KÖBLER, MARTIN MUNDHENK
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

92-02* THOMAS NOLL, HEIKO VOGLER
Top-down Parsing with Simulataneous Evaluationof Noncircular Attribute Grammars

92-03 FAKULTÄT FÜR INFORMATIK
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

92-04 V. ARVIND, J. KÖBLER, M. MUNDHENK
Lowness and the Complexity of Sparse and Tally Descriptions

92-05 JOHANNES KÖBLER
Locating P/poly Optimally in the Extended Low Hierarchy

92-06 ARMIN KÜHNEMANN, HEIKO VOGLER
Synthesized and inherited functions -a new computational model for syntax-directed semantics

92-07 HEINZ FASSBENDER, HEIKO VOGLER
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

92-08 UWE SCHÖNING
On Random Reductions from Sparse Sets to Tally Sets

92-09 HERMANN VON HASSELN, LAURA MARTIGNON
Consistency in Stochastic Network

92-10 MICHAEL SCHMITT
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11 JOHANNES KÖBLER, SEINOSUKE TODA
On the Power of Generalized MOD-Classes

92-12 V. ARVIND, J. KÖBLER, M. MUNDHENK
Reliable Reductions, High Sets and Low Sets

92-13 ALFONS GESER
On a monotonic semantic path ordering

92-14 JOOST ENGELFRIET, HEIKO VOGLER
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01 ALFRED LUPPER, KONRAD FROITZHEIM
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

93-02 M.H. SCHOLL, C. LAASCH, C. RICH, H.-J. SCHEK, M. TRESCH
The COCOON Object Model

93-03 THOMAS THIERAUF, SEINOSUKE TODA, OSAMU WATANABE
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04 JIN-YI CAI, FREDERIC GREEN, THOMAS THIERAUF
On the Correlation of Symmetric Functions

93-05 K.KUHN, M.REICHERT, M. NATHE, T. BEUTER, C. HEINLEIN, P. DADAM
A Conceptual Approach to an Open Hospital Information System

93-06 KLAUS GASSNER
Rechnerunterstützung für die konzeptuelle Modellierung

93-07 ULLRICH KESSLER, PETER DADAM
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01 MICHAEL SCHMITT
On the Complexity of Consistency Problems for Neurons with Binary Weights

Ulmer Informatik-Berichte

ISSN 0939-5091

Herausgeber: Fakultät für Informatik

Universität Ulm, Oberer Eselsberg, D-89069 Ulm