UNIVERSITÄT
KOBLENZ · LANDAU
Institut für Wirtschafts- und Verwaltungsinformatik

FB 4
Informatik

A parallel algorithm for selecting activation functions of an artificial network

Oleg V. Kryuchin
Alexander A. Arzamastsev
Klaus G. Troitzsch

Nr. 12/2011

Arbeitsberichte aus dem Fachbereich Informatik

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346
ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Grimm

Kontaktdaten der Verfasser

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch
Institut für Wirtschafts- und Verwaltungsinformatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
Email : kryuchov@gmail.com, arz_sci@mail.ru, kgt@uni-koblenz.de

A parallel algorithm for selecting activation functions of an artificial network

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch

Abstract

This paper describes a parallel algorithm for selecting activation functions of an artificial network. For checking the efficiency of this algorithm a count of multiplicative and additive operations is used.

Keywords: artificial neural networks, parallel algorithms, computer clusters, activation functions of neurons.

1 Introduction

As is well-known, the training of artificial neural networks (ANNs) is aimed at the minimization of inaccuracy. For this minimization it is necessary to search for apropriate values of weight coefficients and for appropriate neuron activation functions. Nowadays a lot of algorithms of searching for weight coefficients have been developed [1, 2, 3] but there are no effective methods of searching for appropriate neuron activation functions. One of the reasons for this is the long computing time necessary for the combination of the algorithms of searching for neuron activation functions and of searching for weight coefficients. The solution to this problem is the usage of computer clusters. Computer clusters are groups of linked computers, working together closely thus in many respects forming a single computer [4]. The aim of this paper is to develop algorithms of searching for neuron activation functions and to realize them for computer clusters.

2 Neuron components

An artificial neuron is a mathematical function conceived as a crude model or abstraction of biological neurons. Artificial neurons are the constitutive units in an artificial neural network. Depending on the specific model used it has different names, such as a semi-linear unit, an Nv-neuron, a binary neuron, a linear threshold function or a McCulloc-Pits (MCP) neuron. The artificial neuron receives one or more inputs (representing one or more dendrites) and sums them to produce an output (synapse). Usually the sums of each node are weighted and the sum is passed through a non-linear function known as an activation function or a transfer function. The transfer functions usually have a sigmoid shape, but they may also take the form of other non-linear functions, piecewise linear functions or step functions. They are also often monotonically increasing, continuous, differentiable and bounded [5, 6].

Thus an universal mathematics neuron consists of two parts. These are an adder and an activation function (Figure 1). The adder calculates the sum of products of input signal values (which are output values of previous layer neurons and called y_0, y_1, ..., y_{N_x-1}) and weight coefficients (which are called w_0, w_1, ..., w_{N_x-1}). Then it adds the values of two impulses (adjugate weight coefficients). One of these impulses (which is called "the internal impulse" (p_I)) is fixed when weight coefficients are trained and the second impulse (which is called "the external impulse" (p_E)) is updated together with them [7]. After this accumulation the result is multiplied by the weight coefficient and the result of the product is the argument of the activation function. The activation function value is the neuron value. So the i-th neuron output value can be calculated by formula (1) [8, 9].

$$y_i = f_i\left(c_{fi}x_i\right) = f_i\left(c_{fi}\left(\sum_{j=0}^{N_{xi}-1} y_j w_{ji} + p_{Ei} + p_{Ii}\right)\right) \tag{1}$$

where $w_{i,j}$ is the value of the weight coefficient which links the i-th and the j-th neurons, c_{fi} is the coefficient of the i-th neuron activation function, f_i is the i-th neuron activation function, N_{xi} is the i-th neuron inputs number [9].

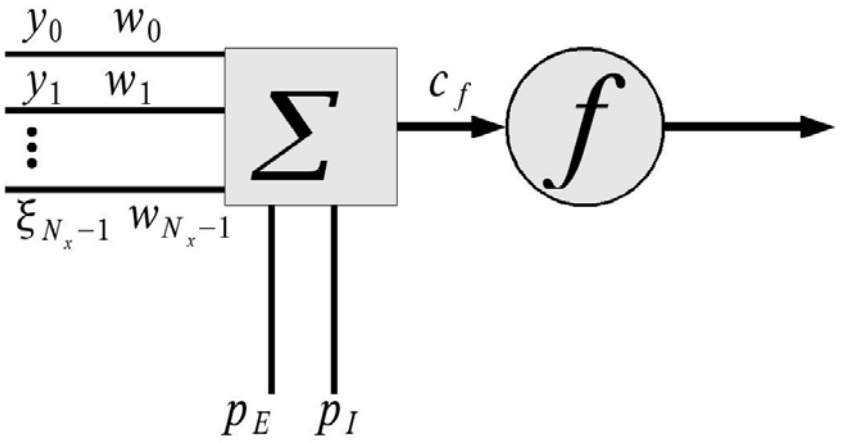

Figure 1: The mathematical neuron model.

Usually the following few activation function flavors are used:

- linear ($f = x$);
- square ($f = x^2$);
- solid ($f = x^3$);
- function of fourth degree ($f = x^4$);
- sine ($f = sin(x)$);
- cosine ($f = cos(x)$);
- tangent ($f = tan(x)$);

- cotangent ($f = ctg(x)$);
- sigmoid ($f = \frac{1}{1+x}$, $f = \frac{1}{1+e^x}$, $f = \frac{1}{1+e^{-x}}$, $f = \frac{1}{1+e^{\{x\}}}$);
- threshold function.

The universal threshold function consists of five elements. These are the threshold value ($\hat{h}_t$), up and down default values ($\hat{y}_1$ and $\hat{y}_0$) and up and down flags showing the necessity of changing the input value to the default value (b_{C1} and b_{C0}).

$$y = \begin{cases} \hat{y}_0, & b_{C0} = 1, \quad x < \hat{h}_t; \\ x, & b_{C0} = 0, \quad x < \hat{h}_t; \\ \hat{y}_1, & b_{C1} = 1, \quad x \geqslant h_t; \\ x, & b_{C1} = 0, \quad x \geqslant \hat{h}_t; \end{cases} \tag{2}$$

where $\hat{y}_1$ and $\hat{y}_0$ are up and down default values, b_{C1} and b_{C0} are up and down flags of changing the current value to the default, and $\hat{h}_t$ is the threshold value.

The function value is set to the input value in two cases. The first case is if the input value is less than the threshold value ($y < \hat{h}_t$) and the down flag is equal to zero ($b_{C0} = 0$). The second case is if the input value is equal or grater than the threshold value ($y \geqslant \hat{h}_t$) and the up flag is equal to zero ($b_{C1} = 0$). In all other cases the neuron output value is the up or down default value ($\hat{y}_1$ or $\hat{y}_0$), respectively [10].

3 The selection of activation function components

As mentioned above a neuron which has the non-threshold activation function consists of three components which can be selected. These are the activation function coefficient, the internal impulse and the activation function type. If the neuron activation function is threshold (2) then this neuron consists of 8 selectable components. These are

- the coefficient of the activation function (c_f),
- the impulse (p_I),
- the threshold value ($\hat{h}_t$),
- the up and down default values ($\hat{y}_1$ and $\hat{y}_0$) and
- the flags of the changing (b_{C1} and b_{C0}).

Most of these values are real numbers, so it is possible to define the range of these parameters with χ_d and χ_u and the step (χ_s). Thus it is possible to calculate values of selected components by formulas (3)-(7) [10].

Sometimes it is not necessary to select all elements, so it is difficult to say how many conditions (enumeration iterations) the selected neuron has (for example, if it selects the coefficient and the internal impulse and their values lie in the band $[-1; 1]$ and the step of values is 0.5 then the neuron has 25 conditions).

$$c_f = \chi_d(c_f) + i\chi_s(c_f), \quad i = 0, 1, \ldots, \left[\frac{\chi_u(c_f) - \chi_d(c_f)}{\chi_c(s_f)}\right] + 1 \tag{3}$$

$$p_I = \chi_d(p_I) + i\chi_s(p_I), \quad i = 0, 1, \ldots, \left[\frac{\chi_u(p_I) - \chi_d(p_I)}{\chi_s(p_I)}\right] + 1 \tag{4}$$

$$\hat{y}_0 = \chi_d(\hat{y}_0) + i\chi_s(\hat{y}_0), \quad i = 0, 1, .., \left[\frac{\chi_u(\hat{y}_0) - \chi_d(\hat{y}_0)}{\chi_s(\hat{y}_0)}\right] + 1 \tag{5}$$

$$\hat{y}_1 = \chi_d(\hat{y}_1) + i\chi_s(\hat{y}_1), \quad i = 0, 1, .., \left[\frac{\chi_u(\hat{y}_1) - \chi_d(\hat{y}_1)}{\chi_s(\hat{y}_1)}\right] + 1 \tag{6}$$

$$\hat{h}_t = \chi_d(\hat{h}_t) + i\chi_s(\hat{h}_t), \quad i = 0, 1, .., \left[\frac{\chi_u(\hat{h}_t) - \chi_d(\hat{h}_t)}{\chi_s(\hat{h}_t)}\right] + 1 \tag{7}$$

Figures 2 and 3 show the scheme of the algorithm changing the neuron activation functions. Here is a short description:

- The block "Input enumeration parameters" means the input options showing elements which must be iterated and the input limits and steps of these elements values ($\chi_d(c_f)$, $\chi_u(c_f)$, $\chi_s(c_f)$ for the activation function coefficient, $\chi_d(p_I)$, $\chi_u(p_I)$, $\chi_s(p_I)$ for the impulse, $\chi_d(\hat{h}_t)$, $\chi_u(\hat{h}_t)$, $\chi_s(\hat{h}_t)$ for threshold etc.).
- The block "Setting anterior values of enumerating parameters" means that iterating parameters are set by the down values.
- The block "Is activation function coefficient selected" means checking the activation function coefficient (c_f). The result of this block is "No" if it is necessary to iterate this coefficient and its value is less than the up limit value ($c_f < \chi_u(c_f)$). If this is so then the coefficient is increased by the step value (as it is shown by formula (8)).
$$c_f = c_f + \chi_s(c_f) \tag{8}$$
- The block "Is impulse selected" means checking the impulse (p_I). This block results in "No" if it is necessary to iterate the impulse and its value is less than the up value ($p_I < \chi_u(p_I)$). If this is so then the impulse value is increased by the step value (as shown by formula (9)), and the activation function coefficient value (if it is necessary to iterate it) is set by formula (10).
$$p_I = p_I + \chi_s(p_I) \tag{9}$$
$$c_f = \chi_d(c_f) \tag{10}$$
- The block "Is activation function the threshold" means checking the activation function type. This block results in "Yes" if the activation function is the threshold else its index is checked. And in this case if the type of activation function is last (the result of the block "Is activation function type selecting finish" is "Yes") then the iteration ends. Otherwise the type is changed (is set to the next type) and the coefficient and the impulse are set by formulas (10)–(11).
$$p_I = \chi_d(p_I) \tag{11}$$

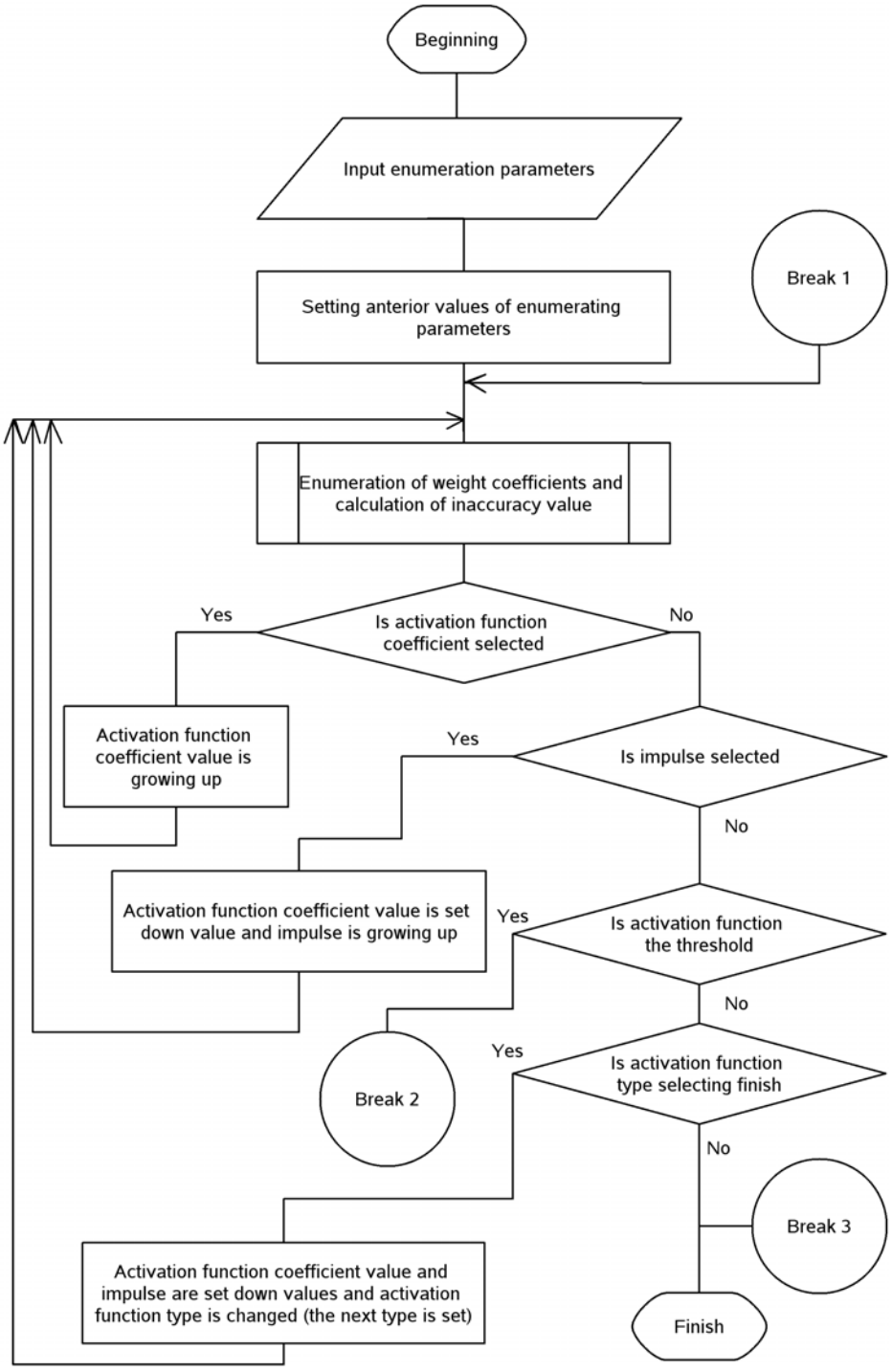

Figure 2: The scheme of the changing one neuron element (part 1).

- The block "Is threshold value selected" means checking the threshold value for finishing the iteration. The result of this block is "No" if it is necessary to iterate the threshold value ($\hat{h}_t$) and it is less than the up limit ($\hat{h}_t < \chi_u(\hat{h}_t)$. If this is so then the new threshold value is calculated by formula (12) and the activation function coefficient and the impulse (if it is necessary to iterate them) are set by

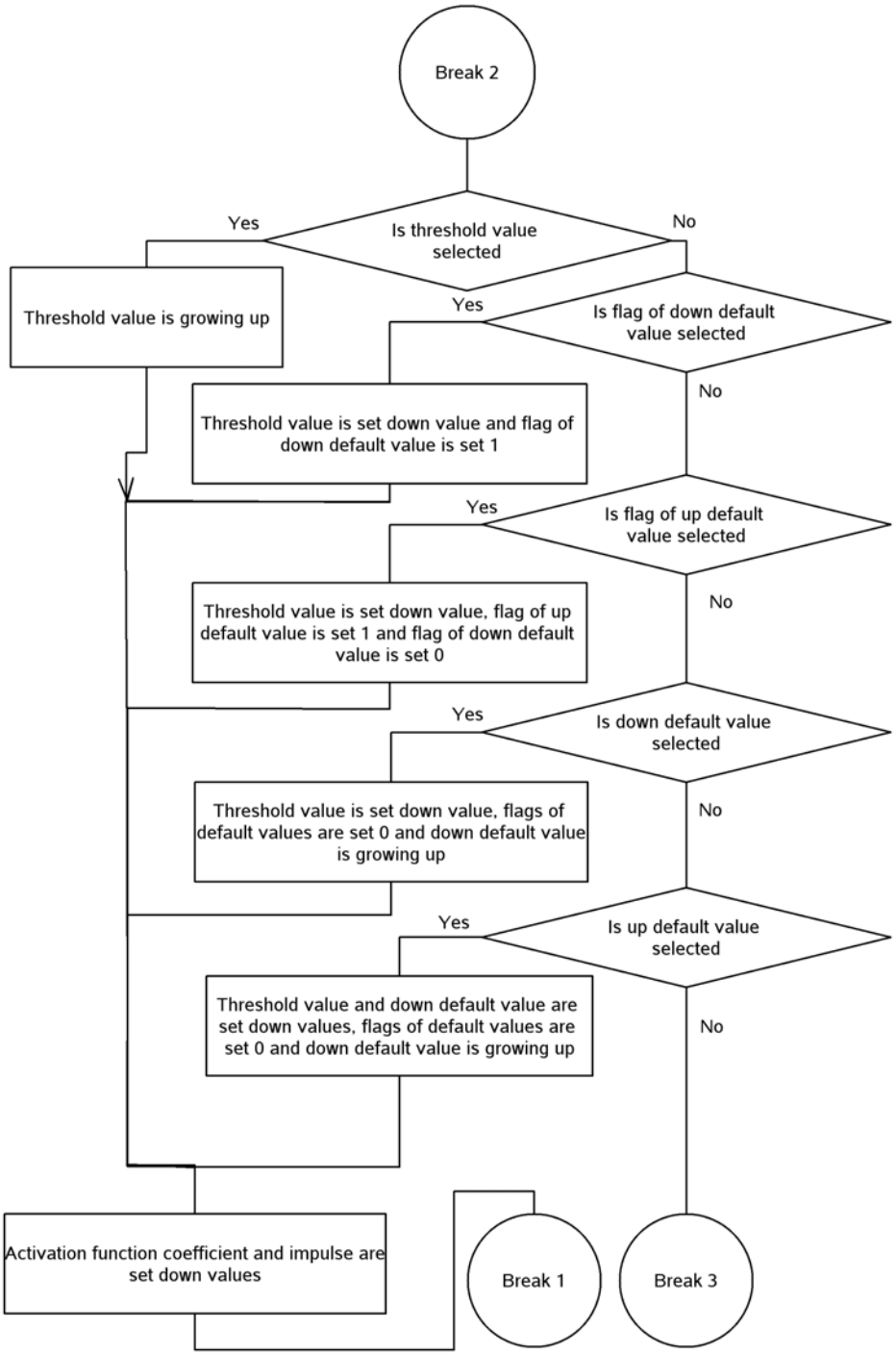

Figure 3: The scheme of the changing one neuron element (part 2).

formulas (10)–(11).

$$\hat{h}_t = \hat{h}_t + \chi_s(\hat{h}_t) \tag{12}$$

- The block "Is flag of down default value selected" means checking the down default value flag (b_{C0}) for finishing the selection. The result of this block is "No" if it is necessary to select this flag and its current value is equal to 0 ($b_{C0} = 0$). If this is so then its new value is set to 1 and the activation function coefficient, the

threshold value and the impulse (if is necessary to enumerate them) are set by formulas (10)–(12).

- The block "Is flag of up default value selected" means checking the up default value flag (b_{C1}) for finishing the selection. The result of this block is "No" if it is necessary to select it and its current value is equal 0 ($b_{C1} = 0$). If this is so then its new value is set to 1 and the new down default value flag is set to 0 (as is shown by formulas (13)–(14)). The activation function coefficient value, the threshold value and the impulse (if is necessary to iterate them) are set by formulas (10)–(12).

$$b_{C0} = 0 \quad (13)$$
$$b_{C1} = 1 \quad (14)$$

- The block "Is down default value selected" means checking the down default value ($\hat{y}_0$) for finishing the selection. The result of this block is "No" if it is necessary to iterate the down default value and its current value is less than the up limit ($\hat{y}_0 < \chi_u(\hat{y}_0)$). If this is so then the down default value is increased by formula (15) and the default values flags (if it is necessary to select them) are set to 0 (by formula (13) for the down flag and by formula (16) for the up flag). The activation function coefficient value, the impulse value and the threshold value (is it necessary to iterate them) are set by formula (10)-(12).

$$\hat{y}_0 = \hat{y}_0 + \chi_s(\hat{y}_0) \quad (15)$$
$$b_{C1} = 0 \quad (16)$$

- The block "Is up default value selected" means checking the up default value ($\hat{y}_1$) for finishing the selection. The result of this block is "No" if it is necessary to iterate it and its current value is less than the up limit ($\hat{y}_1 < \chi_u(\hat{y}_1)$). If this is so then the up default value is increased by formula (17), default values flags (if it is necessary to select them) are set by formulas (13) and (16) and the down default value (if it is necessary to iterate it) is set by formula (18). The activation function coefficient value, the impulse value and the threshold value (is it necessary to iterate them) are set by formula (10)-(12).

$$\hat{y}_1 = \hat{y}_1 + \chi_s(\hat{y}_1) \quad (17)$$
$$\hat{y}_0 = \chi_d(\hat{y}_0) \quad (18)$$

4 Iterating all neuron conditions

For successfully training an ANN it is necessary to search for vector values of conditions of neurons (neuron activation functions) $\vec{\mu}$ and weight coefficients $\vec{w}$ which minimize the inaccuracy value. It is possibly to do this by iterating all activation function combinations and by searching for values of weight coefficients for each neuron condition (activation function) group. It is necessary to consider that each neuron can have a unique group of possible activation functions. The group of possible activation functions of the i-th neuron is written as $\vec{\nu}_i$, and the size of this vector (which characterizes the

activation functions combinations number) is written as $N_\mu(\vec{\nu}_i)$. Thus the i-th neuron activation function in the I-th iteration is calculated by formula (19).

$$\mu_i^{(I)} = \nu_{i,0} + \left(\left[\frac{I}{\hat{\varrho}(i)} \right] \ mod \ N_\mu(\vec{\nu}_i) \right) \tag{19}$$

where $\varrho(i)$ is calculated by formula (20).

$$\hat{\varrho}(i) = \begin{cases} 1, & i = l_\mu - 1; \\ \prod\limits_{j=i+1}^{\hat{l}_\mu - 1} N_\mu(\vec{\nu}_j), & i < l_\mu - 1; \end{cases} \tag{20}$$

For each group of activation functions the algorithm searches for weight coefficients and calculates the target function value $\varepsilon(\vec{w}, \vec{\mu})$ such that after the iteration over all variations of combinations of activation functions an array of inaccuracy values is created. The minimal value of this array means that the condition of the ANN (activation functions values and weight coefficients) which calculated this inaccuracy value is optimal.

The total number of iterations of the activation functions is calculated by formula (21).

$$I_\mu = \prod_{i=0}^{l_\mu - 1} N_\mu(\vec{\nu}_i) \tag{21}$$

5 A parallel algorithm for iterating activation functions

If the parallel algorithm uses n processors then each processor can execute a certain part of activation function iterations. So the lead (zero) processor executes $\hat{J}_\mu$ iterations and the non-lead (nonzero) processors execute J_μ iterations. The values of J_μ and $\hat{J}_\mu$ are calculated by formulas (22)-(23).

$$J_\mu = \begin{cases} \left[\frac{I_\mu}{n} \right], & I_\mu \ mod \ n = 0; \\ \left[\frac{I_\mu}{n-1} \right], & I_\mu \ mod \ n \neq 0; \end{cases} \tag{22}$$

$$\hat{J}_\mu = \begin{cases} J_\mu, & I_\mu \ mod \ n = 0; \\ I_\mu - J_\mu(n-1), & I_\mu \ mod \ n \neq 0; \end{cases} \tag{23}$$

Thus the algorithm which is executed in the lead processor consists of few steps:

- forming $n - 1$ structures for non-lead processors;
- sending structures to other processors;
- enumerating all possible combinations which are allowed for the current processor, selecting the optimal activation function combination and calculating the inaccuracy value;

- receiving the best activation function vector and the inaccuracy value which is connected with this vector;
- selecting the minimum inaccuracy value.

The algorithm in the non-zero processors is different:

- receiving the ANN structure;
- enumerating all combination values which are possible for the current processor, selecting optimal activation function values and linking the inaccuracy value to the array of optimal activation function values;
- sending values of the selected activation functions and the inaccuracy value to the lead processor.

6 Number of multiplication operations executed by serial and parallel algorithms

The number of multiplicative operations which are executed by the serial algorithm for the activation functions selection is calculated by formula (24).

$$z_\mu = z_w I_\mu + 2\sigma I_\mu \tag{24}$$

where σ is the coefficient of the reduction of additive operations to multiplicative operations (its value is calculated by formula (25)) and z_w is the number of operations which are executed for the searching weight coefficients.

$$\sigma = \frac{t_a}{t_m} \tag{25}$$

Here t_a is the time expense for one additive operation and t_m is the time expense for one multiplicative operation.

For the calculation of the number of operations which are executed by the parallel algorithm it is necessary to analyse the steps taken by this algorithm:

1. the initialization;
2. sending data from the lead processor to the other processors;
3. the enumeration of activation function combinations (which are enumerated in the current processor);
4. sending data from all nonzero processors to the lead;
5. selecting the optimal structure by the zero processor;

Table 1: Numbers of operations in steps of the parallel enumeration of activation functions.

Step	**The lead processor**	**The non-lead (the k-th) processor**
1	$2\sigma\hat{l}_\mu$	
2	$\hat{l}_\mu(n-1)(1+2\sigma)$	$\hat{l}_\mu(k+2k\sigma+2\sigma+1)+\gamma(\hat{l}_\mu,v)$
3	$z_w\hat{J}_\mu+2\sigma\hat{J}_\mu$	$z_wJ_\mu+2\sigma J_\mu$
4	$n(l_w+1)(1+2\sigma)+\gamma(\hat{l}_\mu+1)$	$(\hat{l}_\mu+1)(1+2\sigma)$
5	$2\sigma n$	

The total number of selected elements of the ANN can be calculated by formula (26).

$$\hat{l}_\mu=\sum_{i=0}^{l_\mu-1}N_\mu(\nu_i)+l_w \tag{26}$$

where l_w is the number of network weights.

In the first step in the zero processor it executes $2\hat{l}_\mu$ additive operations ($\hat{l}_\mu$ operations for the cycle organisation and $\hat{l}_\mu$ operations for the setting).

In the second step in the lead processor it executes $\hat{l}_\mu(n-1)$ multiplicative and $2\hat{l}_\mu(n-1)$ additive operations and in other processors it executes $\hat{l}_\mu$ multiplicative and $2\hat{l}_\mu$ additive operations. But the non-lead processors cannot begin to execute these operations (of the receiving) before the data are sent by the zero processor, so non-zero processors have to wait. Thus the numbers of operations of the first two steps are calculated by formulas (27)–(29) (the number in the first step in the lead processor is calculated by formula (27), the number in the second step in the lead processor is calculated by formula (28) and the number in the second step in the k-th non-lead processor is calculated by formula (29)).

$$C_{\mu0}^{(1)} = 2\sigma\hat{l}_\mu \tag{27}$$
$$C_{\mu0}^{(2)} = \hat{l}_\mu(n-1)(1+2\sigma) \tag{28}$$
$$C_{\mu k}^{(2)} = \hat{l}_\mu(k+2k\sigma+2\sigma+1)+\gamma(\hat{l}_\mu,v) \tag{29}$$

The number of operations in the third step is analogical to the number of operations in the serial algorithm.

In the fourth step the lead processor has to receive best weight coefficients and the inaccuracy value connected with these weight coefficients from each processor, so it executes $(n-1)(l_\mu-1)$ multiplicative and $2(n-1)(l_\mu-1)$ additive operations and the non-lead processors execute $l_\mu+1$ multiplicative and $2l_\mu+2$ additive operations. And the lead processor waits for the sending.

In the fifth step the lead processor executes $2n$ additive operations. If we reduce the additive operations to multiplicative ones then it is possible to get values which are shown in table 1.

Before receiving weight coefficients and the inaccuracy value the lead processor executes three steps (the total number of operations is written as $\hat{C}_{\mu 0}$) and the other processors execute four steps (ihe total number of operations is written as $\hat{C}_{\mu k}$).

So the enumeration of parallel activation functions needs Z_μ operations (the value is calculated by formula (30)).

$$\begin{aligned} Z_\mu &= \max_{k=1..n-1}\left(\max\left(\hat{C}_{\mu 0}+k(\hat{l}_\mu+1)(2\sigma+1), \hat{C}_{\mu k}+\gamma(\hat{l}_\mu+1,v)\right)\right)+ \\ &+(\hat{l}_\mu+1)(1+2\sigma)+2\sigma n \end{aligned} \tag{30}$$

where the values of $\hat{C}_{\mu 0}$ and $\hat{C}_{\mu k}$ are calculated by formulas (31)-(32).

$$\hat{C}_{\mu 0} = 2\sigma\hat{l}_\mu + \hat{l}_\mu(n-1)(1+2\sigma) + z_w\hat{J}_\mu + 2\sigma\hat{J}_\mu \tag{31}$$

$$\begin{aligned} \hat{C}_{\mu k} &= \hat{l}_\mu(k+2k\sigma+\sigma+1)+\gamma(\hat{l}_\mu,v)+z_w J_\mu + \\ &+2\sigma J_\mu + (\hat{l}_\mu+1)(1+2\sigma) \end{aligned} \tag{32}$$

Thus the efficiency coefficient of the parallel algorithm can be calculated by formula (33).

$$\alpha_\nu(Z) = \frac{z_\mu}{nZ_\mu} = \frac{z_w I_\mu + 2\sigma I_\mu}{nZ_\mu} \tag{33}$$

7 Conclusion

The algorithm of training ANNs in a parallel manner which includes the neuron activation functions selection was developed. This algorithm does not need a high rate of interconnections, so a simple local network can be used in the computer cluster frame.

References

[1] Fahlman S.E. Faster-learning variations on back-propagation: An empirical study. In 1988 Connectionist Models Summer School (San Mateo, CA, 1988), T. J. S. G. E. Hinton and D. S. Touretzky, Eds., Morgan Kaufmann.

[2] Veith A.C., Holmes G.A. A modified quickprop algorithm // Neural Computation, Vol. 3, 1991. – P. 310-311.

[3] Riedmiller M., Brawn H. RPROP – a fast adaptive learning algorithms. Technacal Report, Karlsruhe: University Karlsruhe, 1992.

[4] Bader D., Pennington R. Cluster Computing: Applications // Georgia Tech College of Computing, 1996.

[5] McCulloch W., Pitts W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 1943 – P. 115-133.

[6] Samardak A.S., Nogaret A., Janson N.B., Balanov A.G., Farrer I., Ritchie D.A. Noise-Controlled Signal Transmission in a Multithread Semiconductor Neuron // Phys.Rev.Lett. 102 (2009) 226802.

[7] Rumelhart D. E., Hinton G. E., Williams R. J. 1986. Learning internal reprentations by error propagation. In Parallel distributed processing, vol. 1, Pp. 318-62. Cambridge, MA: MIT Press.

[8] Kryuchin O.V., Arzamastsev A.A., Troitzsch K.G. A universal simulator based on artificial neural networks for computer clusters // Fachbereich Informatik Nr. 2/2011 // [Electronic resource]. `http://www.uni-koblenz.de/f̃b4reports/2011/2011_02_Arbeit-sberichte.pdf`

[9] Крючин О.В., Арзамасцев А.А. Сравнение эффективности последовательных и параллельных алгоритмов обучения искусственных нейронных сете на кластерных вычислительных системах // Вестн. Тамб. ун-та. Сер. Естеств. и техн. науки, 2010 – Т. 15, Вып. 6. – С. 372-375. *(Kryuchin O.V., Arzamastsev A.A. Comparison of effectiveness of consecutive and parallel algorithms of teaching of artificial neural networks on cluster computation systems // Tambov University Report. Series: Natural and Technical Sciences. Tambov, 2010 – Vol. 15, Issue 6 – P. 372-375.)*

[10] Крючин О.В. Нейросетевой симулятор с автоматическим подбором активационных фнкций // Информационные технологии є7. Москва. 2010. С. 47-52. *(Kryuchin O.V. Neural network simulator with automatic selection of activation functions // Information technologies є7. Moscow. 2010. C. 47-52.)*

Glossary

$C_{\mu 0}^{(\iota)}$	the multiplicative operations number executing in the ι-th step of the parallel algorithm of the selecting neurons activation functions in the lead processor, 12
$C_{\mu k}^{(\iota)}$	the multiplicative operations number executing in the ι-th step of the parallel algorithm of the selecting neurons activation functions in the k-th nonzero processor, 12
I_μ	the total iterations number of the activation functions enumeration by the serial algorithm, 10
J_μ	the total iterations number of the activation functions enumeration by the parallel algorithm in the nonzero processors, 10
$N_\mu(\vec{\nu}_i)$	the size of the possible activation functions group of the i-th neuron $\vec{\nu}_i$, 9
N_x	the neuron input signals number, 2
Z_μ	the number of multiplicative operations executing by the parallel algorithm for the activation functions selection, 13
$\alpha_\nu(Z)$	the efficiency coefficient of the parallel algorithm of the selecting neurons activation functions, 13
χ_d	the down limit of the element enumeration, 4
χ_s	the step of the element enumeration, 4
χ_u	the up limit of the element enumeration, 4
$\gamma(m, v)$	the number of multiplicative operations executing for sending m elements with interconnect rate v, 13
$\hat{y}_0$	the down default value of a neuron threshold activation function, 3
$\hat{y}_1$	the up default value of a neuron threshold activation function, 3
σ	the coefficient of the reduction additive operations to multiplicative operations, 11
ε	the inaccuracy value (value of target ANN function), 9
$\vec{\mu}$	the activation functions vector, 9
$\vec{\nu}_i$	the possible activation functions group of the i-th neuron, 9
$\vec{w}$	the weights coefficients vector, 2
$\vec{y}$	output neuron values, 2

b_{C0} the flag showing the necessity of the changing the current neuron value to the down default value, 3

b_{C1} the flag showing the necessity of the changing the current neuron value to the up default value, 3

l_μ the size of the activation functions vector $\vec{\mu}$, 9

l_w the weights coefficients vector $\vec{w}$ size, 11

n the number of processors which are used for the training of an ANN, 10

p_E the external neuron impulse (adjugate weight coefficients), 2

p_I the internal neuron impulse (adjugate weight coefficients), 2

t_a the time expense for one additive operation, 11

t_m the time expense for one multiplicative operation, 11

v the interconnect rate, 13

z_μ the number of multiplicative operations executing by the serial algorithm for the activation functions selection, 11

z_μ the number of multiplicative operations executing for the weights coefficients selection, 11

$\hat{C}_{\mu 0}$ the multiplicative operations number executing in first three steps of the parallel algorithm of the selecting neurons activation functions in the lead processor, 12

$\hat{C}_{\mu k}$ the multiplicative operations number executing in first three steps of the parallel algorithm of the selecting neurons activation functions in the k-th nonzero processor, 12

$\hat{J}_\mu$ the total iterations number of the activation functions enumeration by the parallel algorithm in the lead processor, 10

$\hat{h}_t$ the threshold value of a neuron threshold activation function, 3

$\hat{l}_w$ the total selected elements number, 11

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(http://www.uni-koblenz-landau.de/koblenz/fb4/publications/Reports/arbeitsberichte)

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, A parallel algorithm for selecting activation functions of an artificial network, Arbeitsberichte aus dem Fachbereich Informatik 12/2011

Katharina Bräunlich, Rüdiger Grimm, Andreas Kasten, Sven Vowé, Nico Jahn, Der neue Personalausweis zur Authentifizierung von Wählern bei Onlinewahlen, Arbeitsberichte aus dem Fachbereich Informatik 11/2011

Daniel Eißing, Ansgar Scherp, Steffen Staab, Formal Integration of Individual Knowledge Work and Organizational Knowledge Work with the Core Ontology *strukt*, Arbeitsberichte aus dem Fachbereich Informatik 10/2011

Bernhard Reinert, Martin Schumann, Stefan Müller, Combined Non-Linear Pose Estimation from Points and Lines, Arbeitsberichte aus dem Fachbereich Informatik 9/2011

Tina Walber, Ansgar Scherp, Steffen Staab, Towards the Understanding of Image Semantics by Gaze-based Tag-to-Region Assignments, Arbeitsberichte aus dem Fachbereich Informatik 8/2011

Alexander Kleinen, Ansgar Scherp, Steffen Staab, Mobile Facets – Faceted Search and Exploration of Open Social Media Data on a Touchscreen Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 7/2011

Anna Lantsberg, Klaus G. Troitzsch, Towards A Methodology of Developing Models of E-Service Quality Assessment in Healthcare, Arbeitsberichte aus dem Fachbereich Informatik 6/2011

Ansgar Scherp, Carsten Saathoff, Thomas Franz, Steffen Staab, Designing Core Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 5/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, The prediction of currency exchange rates using artificial neural networks, Arbeitsberichte aus dem Fachbereich Informatik 4/2011

Klaus G. Troitzsch, Anna Lantsberg, Requirements for Health Care Related Websites in Russia: Results from an Analysis of American, British and German Examples, Arbeitsberichte aus dem Fachbereich Informatik 3/2011

Klaus G. Troitzsch, Oleg Kryuchin, Alexander Arzamastsev, A universal simulator based on artificial neural networks for computer clusters, Arbeitsberichte aus dem Fachbereich Informatik 2/2011

Klaus G. Troitzsch, Natalia Zenkova, Alexander Arzamastsev, Development of a technology of designing intelligent information systems for the estimation of social objects, Arbeitsberichte aus dem Fachbereich Informatik 1/2011

Kurt Lautenbach, A Petri Net Approach for Propagating Probabilities and Mass Functions, Arbeitsberichte aus dem Fachbereich Informatik 13/2010

Claudia Schon, Linkless Normal Form for ALC Concepts, Arbeitsberichte aus dem Fachbereich Informatik 12/2010

Alexander Hug, Informatik hautnah erleben, Arbeitsberichte aus dem Fachbereich Informatik 11/2010

Marc Santos, Harald F.O. von Kortzfleisch, Shared Annotation Model – Ein Datenmodell für kollaborative Annotationen, Arbeitsberichte aus dem Fachbereich Informatik 10/2010

Gerd Gröner, Steffen Staab, Categorization and Recognition of Ontology Refactoring Pattern, Arbeitsberichte aus dem Fachbereich Informatik 9/2010

Daniel Eißing, Ansgar Scherp, Carsten Saathoff, Integration of Existing Multimedia Metadata Formats and Metadata Standards in the M3O, Arbeitsberichte aus dem Fachbereich Informatik 8/2010

Stefan Scheglmann, Ansgar Scherp, Steffen Staab, Model-driven Generation of APIs for OWL-based Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 7/2010

Daniel Schmeiß, Ansgar Scherp, Steffen Staab, Integrated Mobile Visualization and Interaction of Events and POIs, Arbeitsberichte aus dem Fachbereich Informatik 6/2010

Rüdiger Grimm, Daniel Pähler, E-Mail-Forensik – IP-Adressen und ihre Zuordnung zu Internet-Teilnehmern und ihren Standorten, Arbeitsberichte aus dem Fachbereich Informatik 5/2010

Christoph Ringelstein, Steffen Staab, PAPEL: Syntax and Semantics for Provenance-Aware Policy Definition, Arbeitsberichte aus dem Fachbereich Informatik 4/2010

Nadine Lindermann, Sylvia Valcárcel, Harald F.O. von Kortzfleisch, Ein Stufenmodell für kollaborative offene Innovationsprozesse in Netzwerken kleiner und mittlerer Unternehmen mit Web 2.0, Arbeitsberichte aus dem Fachbereich Informatik 3/2010

Maria Wimmer, Dagmar Lück-Schneider, Uwe Brinkhoff, Erich Schweighofer, Siegfried Kaiser, Andreas Wieber, Fachtagung Verwaltungsinformatik FTVI Fachtagung Rechtsinformatik FTRI 2010, Arbeitsberichte aus dem Fachbereich Informatik 2/2010

Max Braun, Ansgar Scherp, Steffen Staab, Collaborative Creation of Semantic Points of Interest as Linked Data on the Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 1/2010

Marc Santos, Einsatz von „Shared In-situ Problem Solving“ Annotationen in kollaborativen Lern- und Arbeitsszenarien, Arbeitsberichte aus dem Fachbereich Informatik 20/2009

Carsten Saathoff, Ansgar Scherp, Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology, Arbeitsberichte aus dem Fachbereich Informatik 19/2009

Christoph Kahle, Mario Schaarschmidt, Harald F.O. von Kortzfleisch, Open Innovation: Kundenintegration am Beispiel von IPTV, Arbeitsberichte aus dem Fachbereich Informatik 18/2009

Dietrich Paulus, Lutz Priese, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereicht Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anfordeungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereicht Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese,Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extented Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aud dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description:"E-KRHyper", Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Informationsystems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen", Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe"
(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005