



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 0 935 212 B9**

(12) **CORRECTED EUROPEAN PATENT SPECIFICATION**

Note: Bibliography reflects the latest situation

(15) Correction information:
Corrected version no 1 (W1 B1)
Corrections, see page(s) 18

(51) Int Cl.7: **G06K 9/64**

(48) Corrigendum issued on:
10.07.2002 Bulletin 2002/28

(45) Date of publication and mention
of the grant of the patent:
07.11.2001 Bulletin 2001/45

(21) Application number: **98201910.1**

(22) Date of filing: **09.06.1998**

(54) **N-Tuple or ram based neural network classification system and method**

Klassifizierungssystem und -verfahren mit N-Tuple- oder RAM-basiertem neuronalem Netzwerk

Système et procédé de classification avec un réseau neuronal basé sur N-Tuple ou RAM

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **05.02.1998 DK 16298**

(43) Date of publication of application:
11.08.1999 Bulletin 1999/32

(73) Proprietor: **Intellix A/S**
1879 Frederiksberg C (DK)

(72) Inventors:
• **Jørgensen, Thomas Martini**
3650 Ølstykke (DK)
• **Linneberg, Christian**
2200 Copenhagen N (DK)

(74) Representative: **Hoeiberg, Susanne**
Hoeiberg ApS,
St. Kongensgade 59 B
1264 Copenhagen K (DK)

(56) References cited:
EP-A- 0 300 648 **US-A- 4 490 847**
US-A- 5 136 687

• **MEHENNI B ET AL: "APRIS: AUTOMATIC
PATTERN RECOGNITION AND INSPECTION
SYSTEM" PROCEEDINGS OF THE ANNUAL
COMPUTERS IN DESIGN, MANUFACTURING
AND PRODUCTION CONFERENCE
(COMPEURO), PARIS-EVRY, MAY 24 - 27,
1993,24 May 1993, pages 23-28, XP000463849**
CROISIER A;ISRAEL M; CHAVAND F
• **SLOBODAN RIBARIC ET AL: "PARALLEL
CHARACTER RECOGNITION SYSTEM:
THEORY, SIMULATION AND SYNTHESIS"
MICROPROCESSING AND
MICROPROGRAMMING, vol. 22, no. 5, 1
December 1988, pages 333-346, XP000038022**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 935 212 B9

Description

1. Field of the Invention

[0001] The present invention relates generally to n-tuple or RAM based neural network classification systems and, more particularly, to n-tuple or RAM based classification systems having weight vectors with element values being determined during a training process.

2. Description of the Prior Art

[0002] A known way of classifying objects or patterns represented by electric signals or binary codes and, more precisely, by vectors of signals applied to the inputs of neural network classification systems lies in the implementation of a so-called learning or training phase. This phase generally consists of the configuration of a classification network that fulfils a function of performing the envisaged classification as efficiently as possible by using one or more sets of signals, called learning or training sets, where the membership of each of these signals in one of the classes in which it is desired to classify them is known. This method is known as supervised learning or learning with a teacher.

[0003] A subclass of classification networks using supervised learning are networks using memory-based learning. Here, one of the oldest memory-based networks is the "n-tuple network" proposed by Bledsoe and Browning (Bledsoe, W.W. and Browning, I, 1959, "Pattern recognition and reading by machine", Proceedings of the Eastern Joint Computer Conference, pp. 225-232) and more recently described by Morciniec and Rohwer (Morciniec, M. and Rohwer, R., 1996, "A theoretical and experimental account of n-tuple classifier performance", Neural Comp., pp. 629-642).

[0004] One of the benefits of such a memory-based system is a very fast computation time, both during the learning phase and during classification. For the known types of n-tuple networks, which is also known as "RAM networks" or "weightless neural networks", learning may be accomplished by recording features of patterns in a random-access memory (RAM), which requires just one presentation of the training set(s) to the system.

[0005] The training procedure for a conventional RAM based neural network is described by Jørgensen (co-inventor of this invention) et al. (Jørgensen, T.M., Christensen, S. S. and Liisberg, C., 1995, "Cross-validation and information measures for RAM based neural networks", Proceedings of the Weightless Neural Network Workshop WNNW95 (Kent at Canterbury, UK) ed. D. Bisset, pp.76-81) where it is described how the RAM based neural network may be considered as comprising a number of Look Up Tables (LUTs). Each LUT may probe a subset of a binary input data vector. In the conventional scheme the bits to be used are selected at random. The sampled bit sequence is used to construct an address. This address corresponds to a specific entry (column) in the LUT. The number of rows in the LUT corresponds to the number of possible classes. For each class the output can take on the values 0 or 1. A value of 1 corresponds to a vote on that specific class. When performing a classification, an input vector is sampled, the output vectors from all LUTs are added, and subsequently a winner takes all decision is made to classify the input vector. In order to perform a simple training of the network, the output values may initially be set to 0. For each example in the training set, the following steps should then be carried out:

[0006] Present the input vector and the target class to the network, for all LUTs calculate their corresponding column entries, and set the output value of the target class to 1 in all the "active" columns.

[0007] By use of such a training strategy it may be guaranteed that each training pattern always obtains the maximum number of votes. As a result such a network makes no misclassification on the training set, but ambiguous decisions may occur. Here, the generalisation capability of the network is directly related to the number of input bits for each LUT. If a LUT samples all input bits then it will act as a pure memory device and no generalisation will be provided. As the number of input bits is reduced the generalisation is increased at an expense of an increasing number of ambiguous decisions. Furthermore, the classification and generalisation performances of a LUT are highly dependent on the actual subset of input bits probed. The purpose of an "intelligent" training procedure is thus to select the most appropriate subsets of input data.

[0008] Jørgensen et al. further describes what is named a "cross validation test" which suggests a method for selecting an optimal number of input connections to use per LUT in order to obtain a low classification error rate with a short overall computation time. In order to perform such a cross validation test it is necessary to obtain a knowledge of the actual number of training examples that have visited or addressed the cell or element corresponding to the addressed column and class. It is therefore suggested that these numbers are stored in the LUTs. It is also suggested by Jørgensen et al. how the LUTs in the network can be selected in a more optimum way by successively training new sets of LUTs and performing cross validation test on each LUT. Thus, it is known to have a RAM network in which the LUTs are selected by presenting the training set to the system several times.

[0009] In an article by Jorgensen (co-inventor of this invention) (Jorgensen, T.M. "Classification of handwritten digits using a RAM neural net architecture", February 1997, International Journal of Neural Systems, Vol. 8, No. 1, pp. 17-25 it is suggested how the class recognition of a RAM based network can be further improved by extending the traditional

RAM architecture to include what is named "inhibition". This method deals with the problem that in many situations two different classes might only differ in a few of their features. In such a case, an example outside the training set has a high risk of sharing most of its features with an incorrect class. So, in order to deal with this problem it becomes necessary to weight different features differently for a given class. Thus, a method is suggested where the network includes inhibition factors for some classes of the addressed columns. Here, a confidence measure is introduced, and the inhibition factors are calculated so that the confidence after inhibition corresponds to a desired level.

[0010] The result of the preferred inhibition scheme is that all addressed LUT cells or elements that would be set to 1 in the simple system are also set to 1 in the modified version, but in the modified version column cells being set to 1 may further comprise information of the number of times the cell has been visited by the training set. However, some of the cells containing 0's in the simple system will have their contents changed to negative values in the modified network. In other words, the conventional network is extended so that inhibition from one class to another is allowed.

[0011] In order to encode negative values into the LUT cells, it is not sufficient with one bit per cell or element as with a traditional RAM network. Thus, it is preferred to use one byte per cell with values below 128 being used to represent different negative values, whereas values above 128 are used for storing information concerning the number of training examples that have visited or addressed the cell. When classifying an object the addressed cells having values greater than or equal to 1 may then be counted as having the value 1.

[0012] By using inhibition, the cells of the LUTs are given different values which might be considered a sort of "weighting". However, it is only cells which have not been visited by the training set that are allowed to be suppressed by having their values changed from 0 to a negative value. There is no boosting of cells having positive values when performing classification of input data. Thus, very well performing LUTs or columns of LUTs might easily drown when accompanied by the remaining network.

[0013] Thus, there is a need for a RAM classification network which allows a very fast training or learning phase and subsequent classification, but which at the same time allows real weights to both boost and suppress cell values of LUT columns in order to obtain a proper generalisation ability of the sampled number of input bits based on access information of the training set. Such a RAM based classification system is provided according to the present invention.

[0014] According to a first aspect of the present invention as set out in claims 1, 21, 25 and 42 there is provided a method for training a computer classification system which can be defined by a network comprising a number of n-tuples or Look Up Tables (LUTs), with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of possible classes and further comprising a number of columns being addressed by signals or elements of sampled training input data examples, each column being defined by a vector having cells with values, said method comprising determining the column vector cell values based on one or more training sets of input data examples for different classes so that at least part of the cells comprise or point to information based on the number of times the corresponding cell address is sampled from one or more sets of training input examples, and determining weight cell values corresponding to one or more column vector cells being addressed or sampled by the training examples.

[0015] According to a second aspect of the present invention there is provided a method of determining weight cell values in a computer classification system which can be defined by a network comprising a number of n-tuples or Look Up Tables (LUTs), with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of possible classes and further comprising a number of column vectors with at least part of said column vectors having corresponding weight vectors, each column vector being addressed by signals or elements of a sampled training input data example and each column vector and weight vector having cells with values being determined based on one or more training sets of input data examples for different classes, said method comprising determining the column vector cell values based on the training set(s) of input examples so that at least part of said values comprise or point to information based on the number of times the corresponding cell address is sampled from the set(s) of training input examples, and determining weight vector cell values corresponding to one or more column vector cells.

[0016] Preferably, the weight cell values are determined based on the information of at least part of the determined column vector cell values and by use of at least part of the training set(s) of input examples. According to the present invention the training input data examples may preferably be presented to the network as input signal vectors.

[0017] It is preferred that determination of the weight cell values is performed so as to allow weighting of one or more column vectors cells of positive value and/or to allow boosting of one or more column vector cells during a classification process. Furthermore, or alternatively, the weight cell values may be determined so as to allow suppressing of one or more column vector cells during a classification process.

[0018] The present invention also provide a method wherein the determination of the weight cell values allows weighting of one or more column vector cells having a positive value (greater than 0) and one or more column vector cells having a non-positive value (lesser than or equal to 0). Preferably, the determination of the weight cells allows weighting of any column vector cell.

[0019] In order to determine or calculate the weight cell values, the determination of these values may comprise initialising one or more sets of weight cells corresponding to at least part of the column cells, and adjusting at least part of the weight cell values based on the information of at least part of the determined column cell values and by use

of at least part of the training set(s) of input examples. When determining the weight cell values it is preferred that these are arranged in weight vectors corresponding to at least part of the column vectors.

[0020] In order to determine or adjust the weight cell values according to the present invention, the column cell values should be determined. Here, it is preferred that at least part of the column cell values are determined as a function of the number of times the corresponding cell address is sampled from the set(s) of training input examples. Alternatively, the information of the column cells may be determined so that the maximum column cell value is 1, but at least part of the cells have an associated value being a function of the number of times the corresponding cell address is sampled from the training set(s) of input examples. Preferably, the column vector cell values are determined and stored in storing means before the adjustment of the weight vector cell values.

[0021] According to the present invention, a preferred way of determining the column vector cell values may comprise the training steps of

a) applying a training input data example of a known class to the classification network, thereby addressing one or more column vectors,

b) incrementing, preferably by one, the value or vote of the cells of the addressed column vector(s) corresponding to the row(s) of the known class, and

c) repeating steps (a)-(b) until all training examples have been applied to the network.

[0022] However, it should be understood that the present invention also covers embodiments where the information of the column cells is determined by alternative functions of the number of times the cell has been addressed by the input training set(s). Thus, the cell information does not need to comprise a count of all the times the cell has been addressed, but may for example comprise an indication of when the cell has been visited zero times, once, more than once, and/or twice and more than twice and so on.

[0023] So far it has been mentioned that weight cell values may be determined for one or more column cells, but in a preferred embodiment all column vectors have corresponding weight vectors.

[0024] When initialising weight cell values according to embodiments of the present invention, the initialisation may comprise setting each weight cell value to a predetermined specific cell value. These values may be different for different cells, but all weight cell values may also be set to a predetermined constant value. Such a value may be 0 or 1, but other values may be preferred.

[0025] In order to determine the weight cell values, it is preferred to adjust these values, which adjustment process may comprise one or more iteration steps. The adjustment of the weight cell values may comprise the steps of determining a global quality value based on at least part of the weight and column vector cell values, determining if the global quality value fulfils a required quality criterion, and adjusting at least part of the weight cell values until the global quality criterion is fulfilled.

[0026] The adjustment process may also include determination of a local quality value for each sampled training input example, with one or more weight cell adjustments being performed if the local quality value does not fulfil a specified or required local quality criterion for the selected input example. As an example the adjustment of the weight cell values may comprise the steps of

a) selecting an input example from the training set(s),

b) determining a local quality value corresponding to the sampled training input example, the local quality value being a function of at least part of the addressed weight and column cell values,

c) determining if the local quality value fulfils a required local quality criterion, if not, adjusting one or more of the addressed weight vector cell values if the local quality criterion is not fulfilled,

d) selecting a new input example from a predetermined number of examples of the training set(s),

e) repeating the local quality test steps (b)-(d) for all the predetermined training input examples,

f) determining a global quality value based on at least part of the weight and column vectors being addressed during the local quality test,

g) determining if the global quality value fulfils a required global quality criterion, and,

h) repeating steps (a)-(g) until the global quality criterion is fulfilled. Preferably, steps (b)-(d) of the above mentioned adjustment process may be carried out for all examples of the training set(s).

[0027] The local and/or global quality value may be defined as functions of at least part of the weight and/or column cells. Correspondingly, the global and/or the local quality criterion may also be functions of the weight and/or column cells. Thus, the quality criterion or criteria need not be a predetermined constant threshold value, but may be changed during the adjustment iteration process. However, the present invention also covers embodiments in which the quality criterion or criteria is/are given by constant threshold values.

[0028] It should be understood that when adjusting the weight cell values by use of one or more quality values each with a corresponding quality criterion, it may be preferred to stop the adjustment iteration process if a quality criterion is not fulfilled after a given number of iterations.

[0029] It should also be understood that during the adjustment process the adjusted weight cell values are preferably stored after each adjustment, and when the adjustment process includes the determination of a global quality value, the step of determination of the global quality value may further be followed by separately storing the hereby obtained weight cell values or classification system configuration values if the determined global quality value is closer to fulfil the global quality criterion than the global quality value corresponding to previously separately stored weight cell values or configuration values.

[0030] A main reason for training a classification system according to an embodiment of the present invention is to obtain a high confidence in a subsequent classification process of an input example of an unknown class.

[0031] Thus, according to a further aspect of the present invention, there is also provided a method of classifying input data examples into at least one of a plurality of classes using a computer classification system configured according to any of the above described methods of the present invention, whereby the column cell values and the corresponding weight cell values are determined for each n-tuple or LUT based on one or more training sets of input data examples, said method comprising

a) applying an input data example to be classified to the configured classification network thereby addressing column vectors and corresponding weight vectors in the set of n-tuples or LUTs,

b) selecting a class thereby addressing specific rows in the set of n-tuples or LUTs,

b) determining an output value as a function of values of addressed weight cells,

d) repeating steps (b)-(c) until an output has been determined for all classes,

d) comparing the calculated output values, and

f) selecting the class or classes having maximum output value.

[0032] When classifying an unknown input example, several functions may be used for determining the output values from the addressed weight cells. However, it is preferred that the parameters used for determining the output value includes both values of addressed weight cells and addressed column cells. Thus, as an example, the output value may be determined as a first summation of all the addressed weight cell values corresponding to column cell values greater than or equal to a predetermined value. In another preferred embodiment, the step of determining an output value comprises determining a first summation of all the addressed weight cell values corresponding to column cell values greater than or equal to a predetermined value, determining a second summation of all the addressed weight cell values, and determining the output value by dividing the first summation by the second summation. The predetermined value may preferably be set to 1.

[0033] The present invention also provides training and classification systems according to the above described methods of training and classification.

[0034] Thus, according to the present invention there is provided a system for training a computer classification system which can be defined by a network comprising a stored number of n-tuples or Look Up Tables (LUTs), with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of possible classes and further comprising a number of columns being addressed by signals or elements of sampled training input data examples, each column being defined by a vector having cells with values, said system comprising input means for receiving training input data examples of known classes, means for sampling the received input data examples and addressing column vectors in the stored set of n-tuples or LUTs, means for addressing specific rows in the set of n-tuples or LUTs, said rows corresponding to a known class, storage means for storing determined n-tuples or LUTs, means for determining column vector cell values so as to comprise or point to information based on the number of

times the corresponding cell address is sampled from the training set(s) of input examples, and means for determining weight cell values corresponding to one or more column vector cells being addressed or sampled by the training examples.

[0035] The present invention also provides a system for determining weight cell values of a classification network which can be defined by a stored number of n-tuples or Look Up Tables (LUTs), with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of the number of possible classes and further comprising a number of column vectors with at least part of said column vectors having corresponding weight vectors, each column vector being addressed by signals or elements of a sampled training input data example and each column vector and weight vector having cell values being determined during a training process based on one or more sets of training input data examples, said system comprising: input means for receiving training input data examples of known classes, means for sampling the received input data examples and addressing column vectors and corresponding weight vectors in the stored set of n-tuples or LUTs, means for addressing specific rows in the set of n-tuples or LUTs, said rows corresponding to a known class, storage means for storing determined n-tuples or LUTs, means for determining column vector cell values so as to comprise or point to information based on the number of times the corresponding cell address is sampled from the training set(s) of input examples, and means for determining weight vector cell values corresponding to one or more column vector cells.

[0036] Here, it is preferred that the means for determining the weight cell values is adapted to determine these values based on the information of at least part of the determined column vector cell values and by use of at least part of the training set(s) of input examples.

[0037] Preferably, the means for determining the weight cell values is adapted to determine these values so as to allow weighting of one or more column cells of positive value and/or to allow boosting of one or more column cells during a classification process. The determining means may furthermore, or alternatively, be adapted to determine the weight cell values so as to allow suppressing of one or more column vector cells during a classification process.

[0038] According to an embodiment of the present invention the weight determining means may be adapted to determine the weight cell values so as to allow weighting of one or more column vector cells having a positive value (greater than 0) and one or more column vector cells having a non-positive value (lesser than or equal to 0). Preferably, the means may further be adapted to determine the weight cell values so as to allow weighting of any column cell. It is also preferred that the means for determining the weight cell values is adapted to determine these values so that the weight cell values are arranged in weight vectors corresponding to at least part of the column vectors.

[0039] In order to determine the weight cell values according to a preferred embodiment of the present invention, the means for determining the weight cell values may comprise means for initialising one or more sets of weight vectors corresponding to at least part of the column vectors, and means for adjusting weight vector cell values of at least part of the weight vectors based on the information of at least part of the determined column vector cell values and by use of at least part of the training set(s) of input examples.

[0040] As already discussed above the column cell values should be determined in order to determine the weight cell values. Here, it is preferred that the means for determining the column vector cell values is adapted to determine these values as a function of the number of times the corresponding cell address is sampled from the set(s) of training input examples. Alternatively, the means for determining the column vector cell values may be adapted to determine these cell values so that the maximum value is 1, but at least part of the cells have an associated value being a function of the number of times the corresponding cell address is sampled from the training set(s) of input examples.

[0041] According to an embodiment of the present invention it is preferred that when a training input data example belonging to a known class is applied to the classification network thereby addressing one or more column vectors, the means for determining the column vector cell values is adapted to increment the value or vote of the cells of the addressed column vector(s) corresponding to the row(s) of the known class, said value preferably being incremented by one.

[0042] In order to initialise the weight cells according to an embodiment of the invention, it is preferred that the means for initialising the weight vectors is adapted to setting the weight cell values to one or more predetermined values.

[0043] For the adjustment process of the weight cells it is preferred that the means for adjusting the weight vector cell values is adapted to determine a global quality value based on at least part of the weight and column vector cell values, determine if the global quality value fulfils a required global quality criterion, and adjust at least part of the weight cell values until the global quality criterion is fulfilled.

[0044] As an example of a preferred embodiment according to the present invention, the means for adjusting the weight vector cell values may be adapted to

a) determine a local quality value corresponding to a sampled training input example, the local quality value being a function of at least part of the addressed weight and column vector cell values,

b) determine if the local quality value fulfils a required local quality criterion,

- b) adjust one or more of the addressed weight vector cell values if the local quality criterion is not fulfilled,
- c) repeat the local quality test for a predetermined number of training input exam ples,
- 5 d) determine a global quality value based on at least part of the weight and column vectors being addressed during the local quality test,
- e) determine if the global quality value fulfils a required global quality criterion, and,
- 10 f) repeat the local and the global quality test until the global quality criterion is ful-filled.

[0045] The means for adjusting the weight vector cell values may further be adapted to stop the iteration process if the global quality criterion is not fulfilled after a given number of iterations. In a preferred embodiment, the means for storing n-tuples or LUTs comprises means for storing adjusted weight cell values and separate means for storing best so far weight cell values or best so far classification system configuration values. Here, the means for adjusting the weight vector cell values may further be adapted to replace previously separately stored best so far weight cell values with obtained adjusted weight cell values if the determined global quality value is closer to fulfil the global quality criterion than the global quality value corresponding to previously separately stored best so far weight values. Thus, even if the system should not be able to fulfil the global quality criterion within a given number of iterations, the system may always comprise the "best so far" system configuration.

[0046] According to a further aspect of the present invention there is also provided a system for classifying input data examples of unknown classes into at least one of a plurality of classes, said system comprising: storage means for storing a number or set of n-tuples or Look Up Tables (LUTs) with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of the number of possible classes and further comprising a number of column vectors with corresponding weight vectors, each column vector being addressed by signals or elements of a sampled input data example and each column vector and weight vector having cells with values being determined during a training process based on one or more sets of training input data examples, said system further comprising: input means for receiving an input data example to be classified, means for sampling the received input data example and addressing columns and corresponding weight vectors in the stored set of n-tuples or LUTs, means for addressing specific rows in the set of n-tuples or LUTs, said rows corresponding to a specific class, means for determining an output value as a function of addressed weight cells, and means for comparing calculated output values corresponding to all classes and selecting the class or classes having maximum output value.

[0047] According to a preferred embodiment of the classification system of the present invention, the output determining means comprises means for producing a first summation of all the addressed weight vector cell values corresponding to a specific class and corresponding to column vector cell values greater than or equal to a predetermined value. It is also preferred that the output determining means further comprises means for producing a second summation of all the addressed weight vector cell values corresponding to a specific class, and means for determining the output value by dividing the first summation by the second summation.

[0048] It should be understood that it is preferred that the cell values of the column and weight vectors of the classification system according to the present invention are determined by use of a training system according to any of the above described systems. Accordingly, these cell values may be determined during a training process according to any of the above described methods.

[0049] For a better understanding of the present invention and in order to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings in which:

Fig. 1 shows a block diagram of a RAM classification network with Look Up Tables (LUTs),

Fig. 2 shows a detailed block diagram of a single Look Up Table (LUT) according to an embodiment of the present invention,

Fig. 3 shows a block diagram of a computer classification system according to the present invention,

Fig. 4 shows a flow chart of a learning process for LUT column cells according to an embodiment of the present invention,

Fig. 5 shows a flow chart of a learning process for weight cells according to a first embodiment of the present invention,

Fig. 6 shows a flow chart of a learning process for weight cells according to a second embodiment of the present invention, and

Fig. 7 shows a flow chart of a classification process according to the present invention.

[0050] In the following a more detailed description of the architecture and concept of a classification system according to the present invention will be given including an example of a training process of the column cells of the architecture and an example of a classification process. Furthermore, different examples of learning processes for weight cells according to embodiments of the present invention are described.

Notation

[0051] The notation used in the following description and examples is as follows:

X :	The training set.
$\bar{x}$:	An example from the training set.
N_x :	Number of examples in the training set X .
$\bar{x}_j$:	The j 'th example from a given ordering of the training set X .
y :	A specific example (possible outside the training set).
C :	Classlabel.
$C(\bar{x})$:	Class label corresponding to example $\bar{x}$ (the true class).
C_w :	Winner Class obtained by classification.
C_R :	Runner Up Class obtained by classification.
$\Lambda(\bar{x})$:	Leave-one-out cross-validation classification for example $\bar{x}$.
N_C :	Number of training classes corresponding to the maximum number of rows in a LUT.
Ω :	Set of LUTs (each LUT may contain only a subset of all possible address columns, and the different columns may register only subsets of the existing classes).
N_{LUT} :	Number of LUTs.
N_{COL} :	Number of different columns that can be addressed in a specific LUT (LUT dependent).
S_C :	The set of training examples labelled class C .
w_{iC} :	Weight for the cell addressed by the i 'th column and the C 'th class.
v_{iC} :	Entry counter for the cell addressed by the i 'th column and the C 'th class.
$a_i(\bar{y})$:	Index of the column in the i 'th LUT being addressed by example $\bar{y}$.
$\bar{v}$:	Vector containing all v_{iC} elements of the LUT network.
$\bar{w}$:	Vector containing all w_{iC} elements of the LUT network.
$Q_L(\bar{v}, \bar{w}, \bar{x}, X)$:	Local quality function.
$Q_G(\bar{v}, \bar{w}, X)$:	Global quality function.

Description of architecture and concept

[0052] In the following references are made to Fig. 1, which shows a block diagram of a RAM classification network with Look Up Tables (LUTs), and Fig. 2, which shows a detailed block diagram of a single Look Up Table (LUT) according to an embodiment of the present invention.

[0053] A RAM-net or LUT-net consists of a number of Look Up Tables (LUTs) (1.3). Let the number of LUTs be denoted N_{LUT} . An example of an input data vector $\bar{y}$ to be classified may be presented to an input module (1.1) of the LUT network. Each LUT may sample a part of the input data, where different numbers of input signals may be sampled for different LUTs (1.2) (in principle it is also possible to have one LUT sampling the whole input space). The outputs of the LUTs may be fed (1.4) to an output module (1.5) of the RAM classification network.

[0054] In Fig. 2 it is shown that for each LUT the sampled input data (2.1) of the example presented to the LUT-net may be fed into an address selecting module (2.2). The address selecting module (2.2) may from the input data calculate the address of one or more specific columns (2.3) in the LUT. As an example, let the index of the column in the i 'th LUT being addressed by an input example $\bar{y}$ be calculated as $a_i(\bar{y})$. The number of addressable columns in a specific LUT may be denoted N_{COL} , and varies in general from one LUT to another. The information stored in a specific row of a LUT may correspond to a specific class C (2.4). The maximum number of rows may then correspond to the number of classes, N_C . In a preferred embodiment, every column within a LUT contains two sets of cells. The number of cells within each set corresponds to the number of rows within the LUT. The first set of cells may be denoted column vector cells and the cell values may correspond to class specific entry counters of the column in question. The other set of cells may be denoted weight cells or weight vector cells with cell values which may correspond to weight factors, each

of which may be associated with one entry counter value or column vector cell value. The entry counter value for the cell addressed by the i 'th column and class C is denoted v_{iC} (2.5). The weight value for the cell addressed by the i 'th column and class C is denoted w_{iC} (2.6).

[0055] The v_{iC} - and w_{iC} -values of the activated LUT columns (2.7) may be fed (1.4) to the output module (1.5), where a vote number may be calculated for each class and where finally a winner-takes-all (WTA) decision may be performed.

[0056] Let $\bar{x} \in X$ denote an input data example used for training and let $\bar{y}$ denote an input data example not belonging to the training set. Let $C(\bar{x})$ denote the class to which $\bar{x}$ belongs. The class assignment given to the example $\bar{y}$ is then obtained by calculating a vote number for each class. The vote number obtained for class C is calculated as a function of the v_{iC} and w_{iC} numbers addressed by the example $\bar{y}$:

$$\text{VoteNo}(C, \bar{y}) = \text{function}(v_{a_1(\bar{y}), C}, w_{a_1(\bar{y}), C}, v_{a_2(\bar{y}), C}, w_{a_2(\bar{y}), C}, \dots, v_{a_{N_{LUT}}(\bar{y}), C}, w_{a_{N_{LUT}}(\bar{y}), C})$$

[0057] From the calculated vote numbers the winner class, C_w , can be obtained as:

$$C_w = \underset{C}{\operatorname{argmax}}(\text{VoteNo}(C, \bar{y})), \quad 1 \leq C \leq N_C.$$

[0058] An example of a sensible choice of $\text{VoteNo}(C, \bar{y})$ is the following expression:

$$\text{VoteNo}(C, \bar{y}) = \frac{\sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta_1(v_{a_i(\bar{y}), C})}{\sum_{i \in \Omega} w_{a_i(\bar{y}), C}}$$

$$= \frac{\sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta_1\left(\sum_{\bar{r} \in S_C} \delta_{a_i(\bar{r}), a_i(\bar{y})}\right)}{\sum_{i \in \Omega} w_{a_i(\bar{y}), C}},$$

where $\delta_{i,j}$ is Kroneckers delta ($\delta_{i,j} = 1$ if $i = j$ and 0 otherwise), and

$$\Theta_n(z) = \begin{cases} 1 & \text{if } z \geq n \\ 0 & \text{if } z < n \end{cases}.$$

[0059] Ω describes the set of LUTs making up the whole LUT network. S_C denotes the set of training examples labelled class C . The special case with all w_{iC} -values set to 1 gives the traditional LUT network,

$$C_w = \underset{C}{\operatorname{argmax}}\left(\sum_{i \in \Omega} \Theta_1(v_{a_i(\bar{y}), C})\right).$$

[0060] Figure 3 shows an example of a block diagram of a computer classification system according to the present invention. Here a source such as a video camera or a database provides an input data signal or signals (3.0) describing the example to be classified. These data are fed to a pre-processing module (3.1) of a type which can extract features, reduce, and transform the input data in a predetermined manner. An example of such a pre-processing module is a

FFT-board (Fast Fourier Transform). The transformed data are then fed to a classification unit (3.2) comprising a RAM network according to the present invention. The classification unit (3.2) outputs a ranked classification list which might have associated confidences. The classification unit can be implemented by using software to programme a standard Personal Computer or programming a hardware device, e.g. using programmable gate arrays combined with RAM circuits and a digital signal processor. These data can be interpreted in a post-processing device (3.3), which could be a computer module combining the obtained classifications with other relevant information. Finally the result of this interpretation is fed to an output device (3.4) such as an actuator.

Initial training of the architecture

[0061] The flow chart of Fig. 4 illustrates a one pass learning scheme or process for the determination of the column vector entry counter or cell distribution, v_{iC} -distribution (4.0), according to an embodiment of the present invention, which may be described as follows:

1. Initialise all entry counters or column vector cells by setting the cell values, $\bar{v}$, to zero and initialise the weight values, $\bar{w}$. This could be performed by setting all weight values to a constant factor, or by choosing random values from within a specific range (4.1).
2. Present the first training input example, $\bar{x}_i$, from the training set X to the network (4.2, 4.3)
3. Calculate the columns addressed for the first LUT (4.4, 4.5).
4. Add 1 to the entry counters in the rows of the addressed columns that correspond to the class label of $\bar{x}$ (increment $v_{a(\bar{x}),C(\bar{x})}$ in all LUTs) (4.6).
5. Repeat step 4 for the remaining LUTs (4.7, 4.8).
6. Repeat steps 3-5 for the remaining training input examples (4.9, 4.10). The number of training examples is denoted N_x .

Classification of an unknown input example

[0062] When the RAM network of the present invention has been trained to thereby determine values for the column cells and the weight cells whereby the LUTs may be defined, the network may be used for classifying an unknown input data example.

[0063] In a preferred example according to the present invention, the classification is performed by determining the class having a maximum vote number, VoteNo, where VoteNo is given by the expression

$$\text{VoteNo}(C, \bar{y}) = \frac{\sum_{i \in \Omega} w_{a(\bar{y}),C} \Theta_1(v_{a(\bar{y}),C})}{\sum_{i \in \Omega} w_{a(\bar{y}),C}}$$

[0064] If the denominator is zero the VoteNo can be defined to be 0.

[0065] Thus, the classification example may be described as follows by reference to Figs. 1 and 2:

- Present an unknown input example, $\bar{y}$, to the network (1.1).
- For all LUTs calculate the columns $a(\bar{y})$ addressed by $\bar{y}$ (2.3).
- For each class (corresponding to a specific row in each of the addressed columns) produce the sum (sum_1) of $w_{a(\bar{y}),C} \Theta_1(v_{a(\bar{y}),C})$. The $\Theta_1(v_{a(\bar{y}),C})$ term implies that $w_{a(\bar{y}),C}$ components only are included if $v_{a(\bar{y}),C} \geq 1$ (1.5).
- For each class (corresponding to a specific row in each of the addressed columns) produce the sum (sum_2) of $w_{a(\bar{y}),C}$ (1.5).
- Calculate the output value corresponding to class C as $\text{Out}[C] = \text{sum_1} / \text{sum_2}$ (1.5).
- Choose the class (or classes) that maximise $\text{Out}[C]$ (1.5).

[0066] Figure 7 shows a block diagram of the operation of a computer classification system in which a classification process (7.0) is performed. The system acquires one or more input signals (7.1) using e.g. an optical sensor system. The obtained input data are pre-processed (7.2) in a pre-processing module, e.g. a low-pass filter, and presented to a classification module (7.3) which according to an embodiment of the invention may be a LUT-network. The output data from the classification module is then post-processed in a post-processing module (7.4), e.g. a CRC algorithm

calculating a cyclic redundancy check sum, and the result is forwarded to an output device (7.5), which could be a monitor screen.

Weight adjustments

[0067] Usually the initially determined weight cell values will not present the optimal choice of values. Thus, according to a preferred embodiment of the present invention, an optimisation or adjustment of the weight values should be performed.

[0068] In order to select or adjust the weight values to improve the performance of the classification system, it is suggested according to an embodiment of the invention to define proper quality functions for measuring the performance of the weight values. Thus, a local quality function $Q_L(\bar{v}, \bar{w}, \bar{x}, X)$ may be defined, where $\bar{v}$ denotes a vector containing all v_{iC} elements of the LUT network, and $\bar{w}$ denotes a vector containing all w_{iC} elements of the LUT network. The local quality function may give a confidence measure of the output classification of a specific example $\bar{x}$. If the quality value does not satisfy a given criterion (possibly dynamically changed during the iterations), the weights $\bar{w}$ are adjusted to make the quality value satisfy or closer to satisfying the criterion (if possible).

[0069] Furthermore a global quality function: $Q_G(\bar{v}, \bar{w}, X)$ may be defined. The global quality function may measure the performance of the input training set as a whole.

[0070] Fig. 5 shows a flow chart for weight cell adjustment or learning according to the present invention. The flow chart of Fig. 5 illustrates a more general adjustment or learning process, which may be simplified for specific embodiments.

Example 1

[0071] The vote number function for an input example $\bar{y}$ is given as

$$\text{VoteNo}(C, \bar{y}) = \frac{\sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta_1(v_{a_i(\bar{y}), C})}{\sum_{i \in \Omega} w_{a_i(\bar{y}), C}}$$

[0072] With this definition of the VoteNo() function a leave-one-out cross-validation classification for an input example $\bar{x}$ of the training set may be calculated as:

$$\Lambda(\bar{x}) = \underset{C}{\operatorname{argmax}} \left(\frac{\sum_{i \in \Omega} w_{a_i(\bar{x}), C} \Theta_{1+\delta_{C(\bar{x}), C}}(v_{a_i(\bar{x}), C})}{\sum_{i \in \Omega} w_{a_i(\bar{x}), C}} \right).$$

[0073] This expression is actually explained above except for the factor $\Theta_{1+\delta_{C(\bar{x}), C}}(v_{a_i(\bar{x}), C})$ which is equal to $\Theta_1(v_{a_i(\bar{x}), C})$ if $C \neq C(\bar{x})$ and equal to $\Theta_2(v_{a_i(\bar{x}), C})$ if $C = C(\bar{x})$. $\Theta_2(v_{a_i(\bar{x}), C})$ is only 1 if $v_{a_i(\bar{x}), C} \geq 2$, else it is 0. This simply assures that an example cannot obtain contributions from itself when calculating the leave-one-out cross-validation.

[0074] Let the local quality function calculated for the example $\bar{x}$ be defined as:

$$Q_L(\bar{v}, \bar{w}, \bar{x}, X) = \delta_{A(\bar{x}), C(\bar{x})}.$$

[0075] Here Q_L is 0 if $\bar{x}$ generates a cross-validation error, else Q_L is 1. So if $Q_L = 0$ then weight changes are made.

[0076] Let the global quality function be defined as:

$$Q_G(\bar{v}, \bar{w}, X) = \sum_{\bar{x} \in X} \delta_{\Lambda(\bar{x}), C(\bar{x})} = \sum_{\bar{x} \in X} Q_L(\bar{v}, \bar{w}, \bar{x}, X),$$

[0077] This global quality function measures the number of examples from the training set X that would be correctly classified if they were left out of the training set as each term in the sum over the training set is 1 if $C(\bar{x}) = \Lambda(\bar{x})$ and else 0. The global quality criterion may be to satisfy $Q_G > \varepsilon N_X$, where ε is a parameter determining the fraction of training examples demanded to be correctly classified in a leave-one-out crossvalidation test.

[0078] An updating scheme for improving Q_G can be implemented by the following rules:

[0079] For all input examples $\bar{x}$ of the training set with a wrong cross-validation classification ($\Lambda(\bar{x}) \neq C(\bar{x})$) adjust the weights by:

$$w_{a_i(\bar{x}), C(\bar{x})}^{New} = \max(w_{a_i(\bar{x}), C(\bar{x})}^{Old} + k \cdot \Theta_2(v_{a_i(\bar{x}), C(\bar{x})}) - k \cdot (1 - \Theta_2(v_{a_i(\bar{x}), C(\bar{x})})), 0),$$

where k is a small constant. A feasible choice of k could be one tenth of the mean of the absolute values of the $w_{a_i(\bar{x}), C(\bar{x})}^{New}$ values.

[0080] This updating rule implies that $w_{a_i(\bar{x}), C(\bar{x})}^{New} = w_{a_i(\bar{x}), C(\bar{x})}^{Old} + k$ if $v_{a_i(\bar{x}), C(\bar{x})} \geq 2$ and that $w_{a_i(\bar{x}), C(\bar{x})}^{New} = w_{a_i(\bar{x}), C(\bar{x})}^{Old} - k$ if $v_{a_i(\bar{x}), C(\bar{x})} < 2$. The $\max()$ function ensures that the weights cannot become negative.

[0081] Referring now to Fig. 5, the weight update or adjustment steps of example 1 may be described as:

- Initialise all w_{iC} values to zero (5.1).
- Loop through all examples in the training set (5.2, 5.10, 5.3).
- Calculate the local quality value for each example (can the example be correctly classified if excluded from the training set?) (5.4, 5.5).
- If yes proceed with next example (5.10). If not increase the addressed weights of the "true" class if the corresponding column cells adds a positive contribution to the VoteNo() function (i.e. if $v_{a_i(\bar{x}), C(\bar{x})} \geq 2$) and decrease the weights of the "true" class otherwise (i.e. if $v_{a_i(\bar{x}), C(\bar{x})} < 2$) (5.6-5.9).
- Calculate the global quality value. If the quality is the highest obtained hitherto store the LUT network (5.11).
- Repeat until global quality value is satisfactory or other exit condition is fulfilled (5.12, 5.13).

Example 2

[0082] Let the vote number function for an input example $\bar{y}$ be given as

$$\text{VoteNo}(C, \bar{y}) = \frac{\sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta_1(v_{a_i(\bar{y}), C}) \Theta_2(w_{a_i(\bar{y}), C})}{\sum_{i \in \Omega} |w_{a_i(\bar{y}), C}|}$$

[0083] For the true class $C(\bar{x})$ sum the $w_{a_i(\bar{x}), C(\bar{x})}$ values for a given v -value using the function

$$\text{Hist}(l, C(\bar{x})) = \sum_{i \in \Omega} \delta_{l, v_{a_i(\bar{x}), C(\bar{x})}} w_{a_i(\bar{x}), C(\bar{x})}$$

[0084] The parameter l runs over the possible values of v_{iC} , $0 < v_{iC} \leq N_X$. A confidence $Conf$ between the winning class, C_W , and the runner-up class C_R may then be defined as:

$$Conf = \begin{cases} \text{VoteNo}(C_W, \bar{x}) \sum_{i \in \Omega} w_{a_i(\bar{x}), C_W} - \text{VoteNo}(C_R, \bar{x}) \sum_{i \in \Omega} w_{a_i(\bar{x}), C_R}, & \text{if } (C_W = C(\bar{x})) \\ 0, & \text{if } (C_W \neq C(\bar{x})) \end{cases}$$

[0085] A value m may be determined by the function:

$$m = \max \left(\arg \sum_{l=1}^n \text{Hist}(l, C(\bar{x})) \leq \text{Conf} \right)$$

[0086] The upper limit of the summation index n can vary from 1 to the maximum v_{iC} value within the $\bar{v}$ vector. The expression states that m is chosen as the largest value of n for which

$$\sum_{l=1}^n \text{Hist}(l, C(\bar{x})) \leq \text{Conf}$$

[0087] A local quality function may now be defined as:

$$Q_L(\bar{v}, \bar{w}, \bar{x}, X) = \begin{cases} m - m_{\text{thresh}} & \text{if } C_w = C(\bar{x}) \\ 0 & \text{if } C_w \neq C(\bar{x}) \end{cases}$$

where m_{thresh} is a threshold constant. If $Q_L < 0$ then the weights w_{ij} are updated to make Q_L increase, by adjusting the weights on the runner-up class, C_R :

$$w_{a_i(\bar{x}), C_R}^{\text{New}} = w_{a_i(\bar{x}), C_R}^{\text{Old}} - k_1 \cdot \Theta_1(v_{a_i(\bar{x}), C_R}) + k_2 \cdot (1 - \Theta_1(v_{a_i(\bar{x}), C_R}))$$

[0088] This updating rule implies that $w_{a_i(\bar{x}), C_R}^{\text{New}} = w_{a_i(\bar{x}), C_R}^{\text{Old}} - k_1$ if $v_{a_i(\bar{x}), C_R} \geq 1$ and that

$$w_{a_i(\bar{x}), C_R}^{\text{Old}} = w_{a_i(\bar{x}), C_R}^{\text{New}} + k_2 \text{ if } v_{a_i(\bar{x}), C_R} < 1.$$

[0089] The global quality criterion may be based on two quality functions:

$$Q_{G1}(\bar{v}, \bar{w}, X) = \sum_{\bar{x} \in X} \delta_{\Lambda(\bar{x}), C(\bar{x})}$$

and

$$Q_{G2}(\bar{v}, \bar{w}, X) = \sum_{\bar{x} \in X} \Theta_0(Q_L(\bar{v}, \bar{w}, \bar{x}, X))$$

[0090] Here $\Theta_0(Q_L)$ is 1 if $Q_L \geq 0$ and 0 if $Q_L < 0$. Q_{G1} measures the number of examples from the training set that can pass a leave one out cross-validation test and Q_{G2} measures the number of examples that can pass the local quality criterion.

[0091] These two quality functions can then be combined in to one quality function based on the following boolean expression (a true expression is given the value 1 and a false expression is given the value 0):

$$Q_G(\bar{v}, \bar{w}, X) = (Q_{G1}(\bar{v}, \bar{w}, X) > \varepsilon_1 N_X) \wedge (Q_{G2}(\bar{v}, \bar{w}, X) > \varepsilon_2 N_X)$$

[0092] Here ε_1 and ε_2 are two parameters determining the fractions of training examples demanded to pass a leave-

one-out crossvalidation test and the local quality criterion, respectively. If both of these criteria are passed the global quality criterion is passed in which case $Q_G(\bar{v}, \bar{w}, X)$ is 1, otherwise it is 0.

[0093] With reference to Fig. 5, the weight updating or adjustment steps of example 2 may be described as:

- 5 • Initialise all w_{iC} values to zero (5.1).
- Loop through all examples in the training set (5.2, 5.10, 5.3).
- Calculate the local quality value for each example (5.4) (Does the example have sufficient "support" ? (5.5)).
- If yes process next example (5.10), if not decrease the weights associated with voting for the runner-up class and increase the weights associated with cells having $v_{a_i(\bar{x}), C_R} < 1$ (5.6-5.9).
- 10 • Calculate the global quality value. If the quality is the highest obtained hitherto store the network (5.11).
- Repeat until global quality value is satisfactory or other exit condition is fulfilled (5.12, 5.13).

Example 3

15 **[0094]** Again the vote number function for an input example $\bar{y}$ is given as

$$20 \quad \text{VoteNo}(C, \bar{y}) = \frac{\sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta_1(v_{a_i(\bar{y}), C})}{\sum_{i \in \Omega} w_{a_i(\bar{y}), C}}$$

25 **[0095]** A local quality function $Q_L(\bar{v}, \bar{w}, \bar{x}, X)$ is defined as a measure of a vote confidence for an input training example $\bar{x}$. For an example $\bar{x}$ the confidence *Conf* between the true class, $C(\bar{x})$, and the runner-up class C_R may be determined as:

$$30 \quad \text{Conf}(\bar{x}) = \frac{\sum_{i \in \Omega} w_{a_i(\bar{x}), C(\bar{x})} \Theta_1(v_{a_i(\bar{x}), C(\bar{x})})}{\sum_{i \in \Omega} w_{a_i(\bar{x}), C(\bar{x})}} - \frac{\sum_{i \in \Omega} w_{a_i(\bar{x}), C_R} \Theta_1(v_{a_i(\bar{x}), C_R})}{\sum_{i \in \Omega} w_{a_i(\bar{x}), C_R}}$$

35 **[0096]** The confidence can be zero stating that the runner up class has a vote level equal to that of the true class (if one or more classes have a vote level equal to that of the true class we will define one of the classes different from the true one as the runner up class). The local quality function may now be defined as:

$$40 \quad Q_L(\bar{v}, \bar{w}, \bar{x}, X) = \text{Conf}(\bar{x}).$$

45 **[0097]** A threshold value may be determined for the calculated local quality value and if $Q_L < Q_{\text{threshold}}$ then the weights are updated to make Q_L increase. A possible value of $Q_{\text{threshold}}$ would be 0.1 stating that the difference between the vote level of the true and that of the runner up class should at least be 10% of the maximum vote level. The weights may be updated by adjusting the weights for the runner-up class, C_R :

$$50 \quad w_{a_i(\bar{x}), C_R}^{\text{New}} = w_{a_i(\bar{x}), C_R}^{\text{Old}} \left(1 - k \left[2\Theta_1(v_{a_i(\bar{x}), C_R}) - 1 \right] \right),$$

where k is a small constant, and adjusting the weights for the true class

$$55 \quad w_{a_i(\bar{x}), C(\bar{x})}^{\text{New}} = w_{a_i(\bar{x}), C(\bar{x})}^{\text{Old}} \left(1 + k \left[2\Theta_1(v_{a_i(\bar{x}), C(\bar{x})}) - 1 \right] \right).$$

The small constant k determines the relative change in the weights to be adjusted. One possible choice would be $k=0.05$.

[0098] Again the number of cross-validation errors is a possible global quality measure.

5

$$Q_G(\bar{v}, \bar{w}, X) = \sum_{\bar{x} \in X} \delta_{\Lambda(\bar{x}), C(\bar{x})}.$$

10 **[0099]** The global quality criterion may be to satisfy $Q_G > \varepsilon N_X$, where ε is a parameter determining the fraction of training examples demanded to be correctly classified in a leave-one-out crossvalidation test.

[0100] With reference to Fig. 5, the weight updating or adjustment steps of example 3 may be described as:

- Initialise all w_{iC} values to zero (5.1).
- 15 • Loop through all examples in the training set (5.2, 5.10, 5.3).
- Calculate the local quality value for each example (5.4) (Can the example be correctly classified if excluded from the training set and at the same time have sufficient "support" ? (5.5)).
- If yes process next example, if not update the weights associated with cells voting on the runner-up class and update the weights associated with cells voting for the true class (5.6-5.9) in order to increase the vote level on the true class and decrease the vote level on the runner up class.
- 20 • Calculate the global quality value. If the quality is the highest obtained hitherto store the network (5.11).
- Repeat until global quality value is satisfactory or other exit condition is fulfilled (5.12, 5.13).

Example 4

25

[0101] Again the vote number function for an input example $\bar{y}$ is defined as

30

$$\text{VoteNo}(C, \bar{y}) = \frac{\sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta(v_{a_i(\bar{y}), C})}{\sum_{i \in \Omega} w_{a_i(\bar{y}), C}}$$

35

[0102] The vote levels obtained for a training example when performing a cross-validation test is then:

40

$$\text{VoteNo}_{CV}(C, \bar{x}) = \left(\frac{\sum_{i \in \Omega} w_{a_i(\bar{x}), C} \Theta(v_{a_i(\bar{x}), C} + \delta_{C(\bar{x}), C})}{\sum_{i \in \Omega} w_{a_i(\bar{x}), C}} \right)$$

45

[0103] Again the runner-up class obtained using $\text{VoteNo}(C, \bar{y})$ may be denoted C_R (if one or more classes have a vote level equal to that of the true class we will define one of the classes different from the true one as the runner up class).

[0104] The local quality function $Q_L(\bar{v}, \bar{w}, \bar{x}, X)$ may now be defined by a Boolean expression.

50

$$Q_L(\bar{v}, \bar{w}, \bar{x}, X) = (\text{VoteNo}_{CV}(C(\bar{x}), \bar{x}) > k_1) \wedge (\text{VoteNo}_{CV}(C_R, \bar{x}) < k_2) \wedge (\Lambda(\bar{x}) = C(\bar{x}))$$

where k_1 and k_2 are two constants between 0 and 1 and $k_1 > k_2$. If all three criteria ($\text{VoteNo}_{CV}(C(\bar{x}), \bar{x}) > k_1$, $\text{VoteNo}_{CV}(C_R, \bar{x}) < k_2$, and $\Lambda(\bar{x}) = C(\bar{x})$) are satisfied then $Q_L(\bar{v}, \bar{w}, \bar{x}, X)$ is 1 otherwise it is 0. The two criteria corresponds to demanding the vote level of the true class in a leave-one-out cross-validating test to be larger than k_1 and the vote level of the runner up class to be below k_2 with level k_1 being larger than level k_2 . The $\text{VoteNo}()$ function used in this example will have value between 0 and 1 if we restrict the weights to have positive value in which case a possible choice of k

55

values are k_1 equal to 0.9 and k_2 equal to 0.6.

[0105] If the given criteria for the local quality value given by $Q_L(\bar{v}, \bar{w}, \bar{x}, X)$ is not satisfied then the weights w_{iCR} are updated to satisfy, if possible, the criteria for Q_L , by adjusting the weights on the runner-up class, C_R :

$$w_{a_i(\bar{v}), C_R}^{New} = w_{a_i(\bar{v}), C_R}^{Old} \left(1 - k_3 \left[2\Theta_1(v_{a_i(\bar{v}), C_R}) - 1 \right] \right),$$

where k_3 is a small constant, and adjusting the weights w_{iCR} on the true class:

$$w_{a_i(\bar{v}), C(\bar{v})}^{New} = \max \left(w_{a_i(\bar{v}), C(\bar{v})}^{Old} + k_4 \left[2\Theta_1(v_{a_i(\bar{v}), C(\bar{v})}) - 1 \right], 0 \right).$$

k_3 determines the relative change in the weights to be adjusted for the runner up class. One possible choice would be k_3 is 0.1 A feasible choice of k_4 could be one tenth of the mean of the absolute values of the w_{iC} values.

[0106] A suitable global quality function may be defined by the summation of the local quality values for all the training input examples:

$$Q_G(\bar{v}, \bar{w}, X) = \sum_{\bar{x} \in X} Q_L(\bar{v}, \bar{w}, \bar{x}, X).$$

[0107] The global quality criterion may be to satisfy $Q_G > \varepsilon N_x$, where ε is a parameter determining the fraction of training examples demanded to pass the local quality test.

[0108] With reference to Fig. 5, the weight updating or adjustment steps of example 4 may be described as:

- Initialise all w_{iC} values to zero (5.1).
- Loop through all examples in the training set (5.2, 5.10, 5.3).
- Calculate the local quality for each example (5.4) (Can the example be correctly classified if excluded from the training set and at the same time have sufficient vote "support" ? (5.5)).
- If yes process next example, if not update the weights associated with cells voting on the runner-up class and update the weights associated with cells voting on the true class (5.6-5.9) in order to increase the vote level on the true class and decrease the vote level on the runner up class.
- Calculate the global quality function. If the quality is the highest obtained hitherto store the network (5.11).
- Repeat until global quality value is satisfactory or other exit condition is fulfilled (5.12, 5.13).

Example 5

[0109] In this example the vote number function for an input example $\bar{y}$ is given as

$$\text{VoteNo}(C, \bar{y}) = \sum_{i \in \Omega} w_{a_i(\bar{y}), C} \Theta_1(v_{a_i(\bar{y}), C})$$

[0110] The local quality function and the threshold criteria is now defined so that the answer to the question "is Qlocal OK" will always be no. Thus, the local quality function may be defined as:

$$Q_L = \text{FALSE}$$

[0111] With these definition all training examples will be used for adjusting $w_{a_i(\bar{y}), C}$, as the answer to (5.5) will always be no.

[0112] The weight updating rule is:

$$w_{a_i(\bar{x}), C}^{New} = f_a(v_{a_i(\bar{x}), C},$$

where $f_a(z)$ is defined as:

$$f_a(z) = 1 + \Theta_a(z) = \begin{cases} 2 & \text{if } z > \alpha \\ 1 & \text{if } z \leq \alpha \end{cases}$$

and α is the iteration number.

[0113] The global quality function for the α^{th} iteration may be defined as:

$$Q_G^\alpha = \sum_{\bar{x} \in X} \delta_{\Lambda(\bar{x}, \alpha), C(\bar{x})},$$

where

$$\Lambda(\bar{x}, \alpha) = \operatorname{argmax}_C \left(\sum_{i \in \Omega} f_{a+\delta_{i, \tau_i, C}}(v_{a_i(\bar{x}), C}) \Theta_{1+\delta_{i, \tau_i, C}}(v_{a_i(\bar{x}), C}) \right).$$

[0114] With reference to Fig. 5, the weight updating or adjustment steps of example 5 may be described as:

- Initialise all w_{iC} values to zero (5.1)
- Loop through all examples in the training set (5.2, 5.10, 5.3).
- Calculate the local quality for each example (5.4) (In this example it will always be false, i.e. it will not fulfil the quality criteria).
- If $Q_L = \text{TRUE}$ (5.5) proceed with next example (it will never be true for this example), else set the addressed weights using

$$f_a(v_{a_i(\bar{x}), C})$$

which depends on the actual iteration (5.6-5.9).

- Calculate the global quality value. If the quality value is the highest obtained hitherto store the network (5.11).
- Repeat until the last iteration (5.12, 5.13).

[0115] Thus, the above described example 5 fits into the flowchart structure shown in Fig. 5. However as the answer to (5.5) is always no, the weight assignment procedure can be simplified in the present case as described below with reference to Fig. 6, which shows the flow chart of a more simplified weight cell adjustment process according to the present invention:

[0116] A number of schemes, α_{MAX} , for setting the w_{iC} values may be defined as follows (6.1, 6.6, 6.7):

Scheme α :

(6.2)

for all LUTs do:

for all i do:

for all C do:

$$w_{iC} = f_{\alpha}(v_{iC})$$

[0117] For each α scheme a global quality function for the classification performance may be calculated (6.3). One possibility for the global quality function is to calculate a cross-validation error:

$$Q_G^{\alpha} = \sum_{\bar{x} \in X} \delta_{\Lambda(\bar{x}, \alpha), C_{\bar{x}}},$$

where

$$\Lambda(\bar{x}, \alpha) = \underset{C}{\operatorname{argmax}} \left(\sum_{i \in \Omega} f_{\alpha + \delta_{\Lambda(\bar{x}), C}}(v_{a_i(\bar{x}), C}) \Theta_{1 + \delta_{\Lambda(\bar{x}), C}}(v_{a_i(\bar{x}), C}) \right).$$

[0118] The network having the best quality value Q_G^{α} may then be stored (6.4, 6.5). Here, it should be understood that another number of iterations may be selected and other suitable global quality functions may be defined.

[0119] The foregoing description of preferred exemplary embodiments of the invention has been presented for the purpose of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously many modifications and variations are possible in light of the present invention to those skilled in the art. All such modifications which retain the basic underlying principles disclosed and claimed herein are within the scope of this invention.

Claims

1. A method of training a computer classification system which can be defined by a network comprising a number of n-tuples or Look Up Tables (LUTs), with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of possible classes and further comprising a number of columns being addressed by signals or elements of sampled training input data examples, each column being defined by a vector having cells with values, said method comprising
 - determining the column vector cell values based on one or more training sets of input data examples for different classes so that at least part of the cells comprise or point to information based on the number of times the corresponding cell address is sampled from one or more sets of training input examples, and
 - determining weight cell values corresponding to one or more column vector cells being addressed or sampled by the training examples to thereby allow weighting of one or more column vectors cells of positive value during a classification process, said weight cell values being determined based on the information of at least part of the determined column vector cell values and by use of at least part of the training set(s) of input examples.
2. A method according to claim 1, wherein the weight cells are arranged in weight vectors and the determination of the weight cell values comprises
 - initialising one or more sets of weight vectors corresponding to at least part of the column vectors, and
 - adjusting weight vector cell values of at least part of the weight vectors based on the information of at least part of the determined column vector cell values and by use of at least part of the training set(s) of input examples.
3. A method according to claim 2, wherein the adjustment of the weight vector cell values comprises the steps of
 - determining a global quality value based on at least part of the weight and column vector cell values,
 - determining if the global quality value fulfils a required quality criterion, and
 - adjusting at least part of the weight cell values until the global quality criterion is fulfilled.
4. A method according to any of the claims 1-3, wherein the weight cell values are determined so as to allow boosting of one or more column vector cells during a classification process.

5. A method according to claim 4, wherein the weight cell values are determined so as to allow suppressing of one or more column vector cells during a classification process.
6. A method according to any of the claims 1-5, wherein the determination of the weight cell values allows weighting of one or more column vector cells having a positive value (greater than 0) and one or more column vector cells having a non-positive value (lesser than or equal to 0).
7. A method according to any of the claims 1-6, wherein the weight cell values are arranged in weight vectors corresponding to at least part of the column vectors.
8. A method according to any of the claims 1-7, wherein determination of the weight cells allows weighting of any column vector cell.
9. A method according to any of the claims 1-8, wherein at least part of the column cell values are determined as a function of the number of times the corresponding cell address is sampled from the set(s) of training input examples.
10. A method according to any of the claims 1-9, wherein the maximum column vector value is 1, but at least part of the values have an associated value being a function of the number of times the corresponding cell address is sampled from the training set(s) of input examples.
11. A method according to any of the claims 2-10, wherein the column vector cell values are determined and stored in storing means before the adjustment of the weight vector cell values.
12. A method according to any of the claims 1-11, wherein the determination of the column vector cell values comprises the training steps of
- a) applying a training input data example of a known class to the classification network, thereby addressing one or more column vectors,
 - b) incrementing, preferably by one, the value or vote of the cells of the addressed column vector(s) corresponding to the row(s) of the known class, and
 - c) repeating steps (a)-(b) until all training examples have been applied to the network.
13. A method according to any of the claims 7-12, wherein all column vectors have corresponding weight vectors.
14. A method according to any of the claims 2-13, wherein the initialisation of the weight vectors comprises setting all weight vector cell values to a predetermined constant value, said predetermined value preferably being 1.
15. A method according to any of the claims 2-13, wherein the initialisation of the weight vectors comprises setting each weight vector cell to a predetermined specific cell value.
16. A method according to any of the claims 2-15, wherein the adjustment of the weight cell values comprises the steps of
- a) selecting an input data example from the training set(s),
 - b) determining a local quality value corresponding to the sampled training input example, the local quality value being a function of at least part of the addressed weight and column vector cell values,
 - c) determining if the local quality value fulfils a required local quality criterion, if not, adjusting one or more of the addressed weight vector cell values if the local quality criterion is not fulfilled,
 - d) selecting a new input example from a predetermined number of examples of the training set(s),
 - e) repeating the local quality test steps (b)-(d) for all the predetermined training input examples,
 - f) determining a global quality value based on at least part of the weight and column vectors being addressed during the local quality test,
 - g) determining if the global quality value fulfils a required global quality criterion, and,
 - h) repeating steps (a)-(g) until the global quality criterion is fulfilled.
17. A method according to claim 16, wherein steps (b)-(d) are carried out for all examples of the training set(s).
18. A method according to any of the claims 3-17, wherein the global and/or the local quality criterion is changed during

the adjustment iteration process.

19. A method according to any of the claims 3-18, wherein the adjustment iteration process is stopped if the global quality criterion is not fulfilled after a given number of iterations.

20. A method according to any of the claims 16-19, wherein the adjusted weight cell values are stored after each adjustment, and wherein the determination of the global quality value further is followed by separately storing the hereby obtained weight cell values or classification system configuration values if the determined global quality value is closer to fulfil the global quality criterion than the global quality value corresponding to previously separately stored weight cell values or configuration values.

21. A method of classifying input data example into at least one of a plurality of classes using a computer classification system configured according to a method of any of the claims 1-20, whereby the column vector cell values and the corresponding weight vector cell values are determined for each n-tuple or LUT based on one or more training sets of input data examples, said method comprising

- a) applying an input data example to be classified to the configured classification network thereby addressing column vectors and corresponding weight vectors in the set of n-tuples or LUTs,
- b) selecting a class thereby addressing specific rows in the set of n-tuples or LUTs,
- c) determining an output value as a function of values of addressed weight cells,
- d) repeating steps (b)-(c) until an output has been determined for all classes,
- e) comparing the calculated output values, and
- f) selecting the class or classes having maximum output value.

22. A method according to claim 21, wherein the output value further is determined as a function of values of addressed column cells.

23. A method according to claim 22, wherein said output value is determined as a first summation of all the addressed weight vector cell values corresponding to column vector cell values greater than or equal to a predetermined value, said predetermined value preferably being 1.

24. A method according to claim 22, wherein said step of determining an output value comprises

- determining a first summation of all the addressed weight vector cell values corresponding to column vector cell values greater than or equal to a predetermined value,
- determining a second summation of all the addressed weight vector cell values, and
- determining the output value by dividing the first summation by the second summation.

25. A system for training a computer classification system which can be defined by a network comprising a stored number of n-tuples or Look Up Tables (LUTs), with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of possible classes and further comprising a number of columns being addressed by signals or elements of sampled training input data examples, each column being defined by a vector having cells with values, said system comprising

- input means for receiving training input data examples of known classes,
- means for sampling the received input data examples and addressing column vectors in the stored set of n-tuples or LUTs,
- means for addressing specific rows in the set of n-tuples or LUTs, said rows corresponding to a known class,
- storage means for storing determined n-tuples or LUTs,
- means for determining column vector cell values so as to comprise or point to information based on the number of times the corresponding cell address is sampled from the training set(s) of input examples, and
- means for determining weight cell values corresponding to one or more column vector cells being addressed or sampled by the training examples to thereby allow weighting of one or more column vectors cells of positive value during a classification process, said weight cell values being determined based on the information of at least part of the determined column vector cell values and by use of at least part of the training set(s) of input examples.

26. A system according to claim 25, wherein the means for determining the weight cell values comprises

means for initialising one or more sets of weight vectors corresponding to at least part of the column vectors,
and
means for adjusting weight vector cell values of at least part of the weight vectors based on the information
of at least part of the determined column vector cell values and by use of at least part of the training set(s) of
input examples.

27. A system according to claim 26, wherein the means for adjusting the weight vector cell values is adapted to

determine a global quality value based on at least part of the weight and column vector cell values,
determine if the global quality value fulfils a required global quality criterion, and
adjust at least part of the weight cell values until the global quality criterion is fulfilled.

28. A system according to any of the claims 25-27, wherein the means for determining the weight cell values is adapted
to determine these values so as to allow boosting of one or more column vector cells during a classification
process.

29. A system according to claim 28, wherein the means for determining the weight cell values is adapted to determine
these values so as to allow suppressing of one or more column vector cells during a classification process.

30. A system according to any of the claims 25-29, wherein the means for determining the weight cell values is adapted
to determine these values so as to allow weighting of one or more column vector cells having a positive value
(greater than 0) and one or more column vector cells having a non-positive value (lesser than or equal to 0).

31. A system according to any of the claims 25-30, wherein the means for determining the weight cell values is adapted
to determine these values so that the weight cell values are arranged in weight vectors corresponding to at least
part of the column vectors.

32. A system according to any of the claims 25-31, wherein the means for determining the weight cell values is adapted
to allow weighting of any column vector cell.

33. A system according to any of the claims 25-32, wherein the means for determining the column vector cell values
is adapted to determine these values as a function of the number of times the corresponding cell address is sampled
from the set(s) of training input examples.

34. A system according to any of the claims 25-32, wherein the means for determining the column vector cell values
is adapted to determine these values so that the maximum value is 1, but at least part of the values have an
associated value being a function of the number of times the corresponding cell address is sampled from the
training set(s) of input examples.

35. A system according to any of the claims 25-34, wherein, when a training input data example belonging to a known
class is applied to the classification network thereby addressing one or more column vectors, the means for de-
termining the column vector cell values is adapted to increment the value or vote of the cells of the addressed
column vector(s) corresponding to the row(s) of the known class, said value preferably being incremented by one.

36. A system according to any of the claims 25-35, wherein all column vectors have corresponding weight vectors.

37. A system according to any of the claims 26-36, wherein the means for initialising the weight vectors is adapted to
setting all weight vector cell values to a predetermined constant value, said predetermined value preferably being
one.

38. A system according to any of the claims 26-37, wherein the means for initialising the weight vectors is adapted to
setting each weight vector cell to a predetermined specific value.

39. A system according to any of the claims 26-38, wherein the means for adjusting the weight vector cell values is
adapted to

a) determine a local quality value corresponding to a sampled training input example, the local quality value
being a function of at least part of the addressed weight and column vector cell values,

- b) determine if the local quality value fulfils a required local quality criterion,
- c) adjust one or more of the addressed weight vector cell values if the local quality criterion is not fulfilled,
- d) repeat the local quality test for a predetermined number of training input examples,
- e) determine a global quality value based on at least part of the weight and column vectors being addressed during the local quality test,
- f) determine if the global quality value fulfils a required global quality criterion, and,
- g) repeat the local and the global quality test and associated weight adjustments until the global quality criterion is fulfilled.

40. A system according to any of the claims 26-39, wherein the means for adjusting the weight vector cell values is adapted to stop the iteration process if the global quality criterion is not fulfilled after a given number of iterations.

41. A system according to any of the claims 27-39, wherein the means for storing n-tuples or LUTs comprises means for storing adjusted weight cell values and separate means for storing best so far weight cell values, said means for adjusting the weight vector cell values further being adapted to replace previously separately stored best so far weight cell values with obtained adjusted weight cell values if the determined global quality value is closer to fulfil the global quality criterion than the global quality value corresponding to previously separately stored best so far weight values.

42. A system for classifying input data examples into at least one of a plurality of classes, said system comprising: storage means for storing a number or set of n-tuples or Look Up Tables (LUTs) with each n-tuple or LUT comprising a number of rows corresponding to at least a subset of the number of possible classes and further comprising a number of column vectors with corresponding weight vectors, each column vector being addressed by signals or elements of a sampled input data example and each column vector and weight vector having cells with values being determined during a training process based on one or more sets of training input data examples, said system further comprising:

- input means for receiving an input data example to be classified,
- means for sampling the received input data example and addressing columns and corresponding weight vectors in the stored set of n-tuples or LUTs,
- means for addressing specific rows in the set of n-tuples or LUTs, said rows corresponding to a specific class,
- means for determining an output value as a function of addressed weight cells, and
- means for comparing calculated output values corresponding to all classes and selecting the class or classes having maximum output value.

43. A system according to claim 42, wherein the cell values of the column vectors and the weight vectors have been determined by use of a training system according to any of the systems of claims 25-41.

44. A system according to claim 42, wherein the cell values of the column vectors and the weight vectors have been determined during a training process according to any of the methods of claims 1-20.

45. A system according to any of the claims 42-44, wherein the output value further is determined as a function of values of addressed column cells.

46. A system according to any of the claims 42-45, wherein the output determining means comprises means for producing a first summation of all the addressed weight vector cell values corresponding to a specific class and corresponding to column vector cell values greater than or equal to a predetermined value.

47. A system according to claim 46, wherein the output determining means further comprises means for producing a second summation of all the addressed weight vector cell values corresponding to a specific class, and means for determining the output value by dividing the first summation by the second summation.

Patentansprüche

1. Trainingsmethode für ein Computerklassifizierungssystem, welches definiert werden kann durch ein Netzwerk mit einer Anzahl von n-Tupeln oder Referenztabelle, wobei jedes n-Tupel oder jede Referenztabelle eine Anzahl von Zeilen entsprechend wenigstens einer Untermenge möglicher Klassen und des weiteren eine Anzahl von Spalten

aufweist, die durch Signale oder Elemente von Beispielen von Trainingseingabedaten angesprochen werden, wobei jede Spalte durch einen Vektor definiert wird, der Elemente mit Werten aufweist, und wobei die Methode Sätze von Eingabedatenbeispielen für verschiedene Klassen aufweist, so daß wenigstens ein Teil der Vektorelemente Informationen aufweist oder auf Informationen hinweist, die darauf basieren, wie oft das entsprechende Element von einem Satz oder mehreren Sätzen von Trainingseingabebeispielen angesprochen wurde, und wobei Gewichtselementwerte bestimmt werden, entsprechend einem oder mehreren Spaltenvektorelementen, die durch die Trainingsbeispiele angesprochen wurden, wodurch gleichzeitig die Gewichtung eines oder mehrerer Spaltenvektorelemente positiven Werts während eines Klassifizierungsvorgangs ermöglicht wird, wobei die Gewichtselementwerte bestimmt werden auf der Basis der Information von zumindest einem Teil der bestimmten Spaltenvektorelementwerte und unter Verwendung von mindestens einem Teil der Trainingssätze von Eingabebeispielen.

2. Methode nach Anspruch 1, **dadurch gekennzeichnet, daß** die Gewichtselemente in Gewichtsvektoren angeordnet sind, und die Bestimmung der Gewichtselementwerte die folgenden Schritte umfaßt:

Initialisierung eines Satzes oder mehrerer Sätze von Gewichtsvektoren entsprechend zumindest einem Teil der Spaltenvektoren, und

Einstellung der Gewichtsvektorelementwerte von zumindest einem Teil der Gewichtsvektoren auf der Basis der Information von zumindest den bestimmten Spaltenvektorelementwerten und unter Verwendung von zumindest einem Teil der Trainingssätze von Eingabebeispielen.

3. Methode nach Anspruch 2, **dadurch gekennzeichnet, daß** die Einstellung der Gewichtsvektorelementwerte die folgenden Schritte umfaßt:

Bestimmung eines globalen Qualitätswerts auf der Basis zumindest eines Teils der Gewichts- und Spaltenvektorelementwerte,

Bestimmung, ob der globale Qualitätswert ein gefordertes Qualitätskriterium erfüllt, und

Einstellung zumindest eines Teils der Gewichtselementwerte, bis das globale Qualitätskriterium erfüllt ist.

4. Methode nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet, daß** die Gewichtselementwerte so bestimmt werden, daß ein oder mehrere Spaltenvektorelemente während eines Klassifizierungsprozesses verstärkt werden können.

5. Methode nach Anspruch 4, **dadurch gekennzeichnet, daß** die Gewichtselementwerte so bestimmt werden, daß ein oder mehrere Spaltenvektorelemente während eines Klassifizierungsprozesses unterdrückt werden können.

6. Methode nach einem der Ansprüche 1 bis 5, **dadurch gekennzeichnet, daß** die Bestimmung der Gewichtselementwerte eine Gewichtung eines oder mehrerer Spaltenvektorelemente mit positivem Wert (größer als 0) und eines oder mehrerer Spaltenvektorelemente mit nicht-positivem Wert (kleiner oder gleich 0) ermöglicht.

7. Methode nach einem der Ansprüche 1 bis 6, **dadurch gekennzeichnet, daß** die Gewichtselementwerte in Gewichtsvektoren entsprechend zumindest einem Teil der Spaltenvektoren angeordnet sind.

8. Methode nach einem der Ansprüche 1 bis 7, **dadurch gekennzeichnet, daß** die Bestimmung der Gewichtselemente eine Gewichtung jedes Spaltenvektorelements ermöglicht.

9. Methode nach einem der Ansprüche 1 bis 8, **dadurch gekennzeichnet, daß** zumindest ein Teil der Spaltenelementwerte als eine Funktion davon bestimmt wird, wie oft das entsprechende Element von dem Satz / den Sätzen der Trainingseingabebeispiele angesprochen wird.

10. Methode nach einem der Ansprüche 1 bis 9, **dadurch gekennzeichnet, daß** der höchste Spaltenvektorwert 1 beträgt, aber zumindest ein Teil der Werte einen assoziierten Wert besitzt, der eine Funktion davon ist, wie oft das entsprechende Element von dem Satz / den Sätzen von Trainingseingabebeispielen angesprochen wird.

11. Methode nach einem der Ansprüche 2 bis 10, **dadurch gekennzeichnet, daß** die Spaltenvektorelementwerte

bestimmt werden und in Speichervorrichtungen gespeichert werden, bevor die Einstellung der Gewichtsvektorelementwerte stattfindet.

5 12. Methode nach einem der Ansprüche 1 bis 11, **dadurch gekennzeichnet, daß** die Bestimmung der Spaltenvektorelementwerte folgende Trainingsschritte umfaßt:

a) Eingabe eines Trainingseingabedatenbeispiels einer bekannten Klasse in das Klassifizierungsnetzwerk, wobei ein oder mehrere Spaltenvektoren angesprochen werden,

10 b) Erhöhen des Werts der Elemente des/der angesprochenen Spaltenvektors/Spaltenvektoren entsprechend der Zeile(n) der bekannten Klasse, bevorzugt um 1, und

c) Wiederholung der Schritte (a) - (b), bis alle Trainingsbeispiele in das Netzwerk eingegeben wurden.

15 13. Methode nach einem der Ansprüche 7 bis 12, **dadurch gekennzeichnet, daß** allen Spaltenvektoren entsprechende Gewichtsvektoren zugeordnet sind.

20 14. Methode nach einem der Ansprüche 2 bis 13, **dadurch gekennzeichnet, daß** die Initialisierung der Gewichtsvektoren den Schritt aufweist, alle Gewichtsvektorelementwerte auf einen vorbestimmten konstanten Wert einzustellen, der bevorzugt 1 ist.

25 15. Methode nach einem der Ansprüche 2 bis 13, **dadurch gekennzeichnet, daß** die Initialisierung der Gewichtsvektoren den Schritt aufweist, jedes Gewichtsvektorelement auf einen vorbestimmten speziellen Elementwert einzustellen.

30 16. Methode nach einem der Ansprüche 2 bis 15, **dadurch gekennzeichnet, daß** die Einstellung der Gewichtselementwerte folgende Schritte umfaßt:

a) Auswahl eines Eingabedatenbeispiels aus dem Trainingssatz / den Trainingssätzen,

35 b) Bestimmung eines lokalen Qualitätswerts entsprechend dem ausgewählten Trainingseingabebeispiel, wobei der lokale Qualitätswert eine Funktion von zumindest einem Teil der angesprochenen Gewichts- und Spaltenvektorelementwerte ist,

c) Bestimmung, ob der lokale Qualitätswert ein gefordertes lokales Qualitätskriterium erfüllt, und, wenn nicht, Einstellung eines oder mehrerer der angesprochenen Gewichtsvektorelementwerte, wenn das lokale Qualitätskriterium nicht erfüllt ist,

40 d) Auswahl eines neuen Eingabebeispiels von einer vorbestimmten Anzahl an Beispielen des Trainingssatzes / der Trainingssätze,

e) Wiederholung der lokalen Qualitätstestschritte (b) - (d) für alle vorbestimmten Trainingseingabebeispiele,

45 f) Bestimmung eines globalen Qualitätswerts auf der Basis von zumindest einem Teil der Gewichts- und Spaltenvektoren, die während des lokalen Qualitätstests angesprochen wurden,

g) Bestimmung, ob der globale Qualitätswert ein gefordertes globales Qualitätskriterium erfüllt, und

50 h) Wiederholung der Schritte (a) - (g), bis das globale Qualitätskriterium erfüllt ist.

55 17. Methode nach Anspruch 16, **dadurch gekennzeichnet, daß** die Schritte (b) - (d) für alle Beispiele des Trainingssatzes / der Trainingssätze durchgeführt werden.

18. Methode nach einem der Ansprüche 3 bis 17, **dadurch gekennzeichnet, daß** das globale und/oder das lokale Qualitätskriterium während des Einstellungsiterationsprozesses verändert wird.

19. Methode nach einem der Ansprüche 3 bis 18, **dadurch gekennzeichnet, daß** der Einstellungsiterationsprozeß angehalten wird, falls das globale Qualitätskriterium nach einer vorgegebenen Anzahl von Wiederholungen nicht

erfüllt wird.

20. Methode nach einem der Ansprüche 16 bis 19, **dadurch gekennzeichnet, daß** die eingestellten Gewichtselementwerte nach jeder Einstellung gespeichert werden, und daß nach der Bestimmung des globalen Qualitätswerts des weiteren der Schritt folgt, die dabei erzielten Gewichtselementwerte oder die Konfigurationswerte des Klassifizierungssystems getrennt voneinander abzuspeichern, wenn der bestimmte globale Qualitätswert dem globalen Qualitätskriterium näher kommt als der globale Qualitätswert entsprechend den früher getrennt gespeicherten Gewichtselementwerten oder Konfigurationswerten.

21. Methode zur Klassifizierung eines Eingabedatenbeispiels in zumindest eine von einer Mehrzahl von Klassen unter Verwendung eines Computerklassifizierungssystems, das entsprechend einer Methode nach einem der Ansprüche 1 bis 20 konfiguriert ist, wobei die Spaltenvektorelementwerte und die entsprechenden Gewichtsvektorelementwerte für jedes n-Tupel oder für jede Referenztafel auf der Basis eines Trainingsatzes oder mehrerer Trainingsätze von Eingabedatenbeispielen bestimmt werden, und wobei die Methode folgende Schritte umfaßt:

a) Eingabe eines zu klassifizierenden Eingabedatenbeispiels in das konfigurierte Klassifizierungsnetzwerk, wobei Spaltenvektoren und entsprechende Gewichtsvektoren in dem Satz von n-Tupeln oder Referenztabellen angesprochen werden,

b) Auswahl einer Klasse und dabei Ansprache spezieller Zeilen in dem Satz von n-Tupeln oder Referenztabellen,

c) Bestimmung eines Ausgabewertes als Funktion der Werte der angesprochenen Gewichtselemente,

d) Wiederholung der Schritte (b) - (c), bis ein Ausgabewert für alle Klassen bestimmt wurde,

e) Vergleich der berechneten Ausgabewerte, und

f) Auswahl der Klasse(n) mit maximalem Ausgabewert.

22. Methode nach Anspruch 21, **dadurch gekennzeichnet, daß** der Ausgabewert des weiteren als eine Funktion der Werte der angesprochenen Spaltenelemente bestimmt wird.

23. Methode nach Anspruch 22, **dadurch gekennzeichnet, daß** der Ausgabewert als eine erste Summe aller angesprochenen Gewichtsvektorelementwerte bestimmt wird, die Spaltenvektorelementwerten von größer oder gleich einem vorbestimmten Wert entsprechen, wobei der vorbestimmte Wert bevorzugt 1 ist.

24. Methode nach Anspruch 22, **dadurch gekennzeichnet, daß** der Schritt der Bestimmung eines Ausgabewertes folgende Teilschritte umfaßt:

Bestimmung einer ersten Summe aller angesprochenen Gewichtsvektorelementwerte, die Spaltenvektorelementwerten größer oder gleich einem vorbestimmten Wert entsprechen,

Bestimmung einer zweiten Summe aller angesprochenen Gewichtsvektorelementwerte, und

Bestimmung des Ausgabewerts durch Teilung der ersten Summe durch die zweite Summe.

25. System zum Trainieren eines Computerklassifizierungssystems, das durch ein Netzwerk mit einer gespeicherten Anzahl von n-Tupeln oder Referenztabellen definiert werden kann, wobei jedes n-Tupel oder jede Referenztafel eine Anzahl an Zeilen entsprechend zumindest einer Untermenge möglicher Klassen und des weiteren eine Anzahl an Spalten aufweist, die durch Signale oder Elemente von stichprobenartig ausgewählten Beispielen von Trainingseingabedaten angesprochen werden, wobei jede Spalte durch einen Vektor definiert wird, der Elemente mit Werten aufweist, und wobei das System umfaßt:

eine Eingabevorrichtung zum Empfangen von Trainingseingabedatenbeispielen bekannter Klassen,

eine Vorrichtung zur Stichprobenentnahme aus den empfangenen Eingabedatenbeispielen und zur Ansprache

von Spaltenvektoren in dem gespeicherten Satz von n-Tupeln oder Referenztabellen,

eine Vorrichtung zur Ansprache spezieller Zeilen in dem Satz von n-Tupeln oder Referenztabellen, wobei die Zeilen einer bekannten Klasse entsprechen,

eine Speichervorrichtung zum Speichern der bestimmten n-Tupel oder Referenztabellen,

eine Vorrichtung zur Bestimmung von Spaltenvektorelementwerten, die Informationen aufweisen oder auf Information hinweisen, welche darauf basieren, wie oft das entsprechende Element von dem Trainingssatz / den Trainingssätzen der Eingabebeispiele angesprochen wurde, und

eine Vorrichtung zur Bestimmung von Gewichtselementwerten entsprechend einer oder mehrerer Spaltenvektorelemente, die durch die Trainingsbeispiele angesprochen wurden, wodurch gleichzeitig eine Gewichtung einer oder mehrerer Spaltenvektorelemente positiven Werts während eines Klassifizierungsprozesses ermöglicht wird, wobei die Gewichtselementwerte bestimmt werden auf der Basis der Information von zumindest einem Teil der bestimmten Spaltenvektorelementwerte und unter Verwendung zumindest eines Teils der Trainingssätze von Eingabebeispielen.

26. System nach Anspruch 25, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Gewichtselementwerte umfaßt:

eine Vorrichtung zur Initialisierung eines Satzes oder mehrerer Sätze von Gewichtsvektoren entsprechend zumindest einem Teil der Spaltenvektoren, und

eine Vorrichtung zur Einstellung der Gewichtsvektorelementwerte von zumindest einem Teil der Gewichtsvektoren auf der Basis der Information von zumindest einem Teil der bestimmten Spaltenvektorelementwerte und unter Verwendung von zumindest einem Teil der Trainingssätze von Eingabebeispielen.

27. System nach Anspruch 26, **dadurch gekennzeichnet, daß** die Vorrichtung zur Einstellung der Gewichtsvektorelementwerte geeignet ist,

einen globalen Qualitätswert auf der Basis von zumindest einem Teil der Gewichts- und Spaltenvektorelementwerte zu bestimmen,

zu bestimmen, ob der globale Qualitätswert ein gefordertes globales Qualitätskriterium erfüllt, und

zumindest einen Teil der Gewichtselementwerte einzustellen, bis das globale Qualitätskriterium erfüllt ist.

28. System nach einem der Ansprüche 25 bis 27, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Gewichtselementwerte dazu geeignet ist, diese Werte so zu bestimmen, daß ein oder mehrere Spaltenvektorelemente während eines Klassifizierungsprozesses verstärkt werden.

29. System nach Anspruch 28, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Gewichtselementwerte dazu geeignet ist, diese Werte so zu bestimmen, daß ein oder mehrere Spaltenvektorelemente während eines Klassifizierungsprozesses unterdrückt werden.

30. System nach einem der Ansprüche 25 bis 29, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Gewichtselementwerte dazu geeignet ist, diese Werte so zu bestimmen, daß eine Gewichtung von einem oder mehreren Spaltenvektorelementen mit positivem Wert (größer als 0) und von einem oder mehreren Spaltenvektorelementen mit nicht-positivem Wert (kleiner oder gleich 0) ermöglicht wird.

31. System nach einem der Ansprüche 25 bis 30, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Gewichtselementwerte dazu geeignet ist, diese Werte so zu bestimmen, daß die Gewichtselementwerte in Gewichtsvektoren entsprechend zumindest einem Teil der Spaltenvektoren angeordnet sind.

32. System nach einem der Ansprüche 25 bis 31, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Gewichtselementwerte dazu geeignet ist, eine Gewichtung jedes Spaltenvektorelements zu ermöglichen.

33. System nach einem der Ansprüche 25 bis 32, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Spaltenvektorelementwerte dazu geeignet ist, diese Werte als eine Funktion davon zu bestimmen, wie oft das entsprechende Element von dem Satz / den Sätzen von Trainingseingabebeispielen angesprochen wird.
- 5 34. System nach einem der Ansprüche 25 bis 32, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Spaltenvektorelementwerte dazu geeignet ist, diese Werte so zu bestimmen, daß der Maximalwert 1 ist, aber zumindest ein Teil der Werte einen assoziierten Wert aufweist, der eine Funktion davon ist, wie oft das entsprechende Element von dem Satz / den Sätzen von Trainingseingabebeispielen angesprochen wird.
- 10 35. System nach einem der Ansprüche 25 bis 34, **dadurch gekennzeichnet, daß** die Vorrichtung zur Bestimmung der Spaltenvektorelementwerte dazu geeignet ist, den Wert der Elemente des/der angesprochenen Spaltenvektors/Spaltenvektoren entsprechend der Zeile(n) der bekannten Klasse, bevorzugt um 1, zu erhöhen, wenn ein Trainingseingabedatenbeispiel, das zu einer bekannten Klasse gehört, in das Klassifizierungsnetzwerk eingegeben wird, wobei ein oder mehrere Spaltenvektoren angesprochen werden.
- 15 36. System nach einem der Ansprüche 25 bis 35, **dadurch gekennzeichnet, daß** alle Spaltenvektoren entsprechende Gewichtsvektoren haben.
- 20 37. System nach einem der Ansprüche 26 bis 36, **dadurch gekennzeichnet, daß** die Vorrichtung zur Initialisierung der Gewichtsvektoren dazu geeignet ist, alle Gewichtsvektorelementwerte auf einen vorbestimmten konstanten Wert zu setzen, wobei der vorbestimmte Wert bevorzugt 1 ist.
- 25 38. System nach einem der Ansprüche 26 bis 37, **dadurch gekennzeichnet, daß** die Vorrichtung zur Initialisierung der Gewichtsvektoren dazu geeignet ist, jedes Gewichtsvektorelement auf einen vorbestimmten speziellen Wert einzustellen.
- 30 39. System nach einem der Ansprüche 26 bis 38, **dadurch gekennzeichnet, daß** die Vorrichtung zur Einstellung der Gewichtsvektorelementwerte dazu geeignet ist,
- a) einen lokalen Qualitätswert entsprechend einem als Stichprobe ausgewählten Trainingseingabebeispiel zu bestimmen, wobei der lokale Qualitätswert eine Funktion zumindest eines Teils der angesprochenen Gewichts- und Spaltenvektorelementwerte ist,
- b) zu bestimmen, ob der lokale Qualitätswert ein gefordertes lokales Qualitätskriterium erfüllt,
- 35 c) einen oder mehrere der angesprochenen Gewichtsvektorelementwerte einzustellen, wenn das lokale Qualitätskriterium nicht erfüllt wird,
- d) den lokalen Qualitätstest für eine vorbestimmte Anzahl von Trainingseingabebeispielen zu wiederholen,
- 40 e) einen globalen Qualitätswert auf der Basis von zumindest einem Teil der Gewichts- und Spaltenvektoren zu bestimmen, die während des lokalen Qualitätstests angesprochen werden,
- f) zu bestimmen, ob der globale Qualitätswert ein gefordertes globales Qualitätskriterium erfüllt, und
- 45 g) den lokalen und den globalen Qualitätstest und die damit verbundenen Gewichtungseinstellungen zu wiederholen, bis das globale Qualitätskriterium erfüllt wird.
- 50 40. System nach einem der Ansprüche 26 bis 39, **dadurch gekennzeichnet, daß** die Vorrichtung zur Einstellung der Gewichtsvektorelementwerte dazu geeignet ist, den Iterationsprozeß anzuhalten, falls das globale Qualitätskriterium nicht nach einer vorgegebenen Anzahl von Wiederholungen erfüllt wird.
- 55 41. System nach einem der Ansprüche 27 bis 39, **dadurch gekennzeichnet, daß** die Vorrichtung zur Speicherung der n-Tupel oder Referenztabelle eine Vorrichtung zur Speicherung der eingestellten Gewichtselementwerte und eine getrennte Vorrichtung zur Speicherung der bis dato besten Gewichtselementwerte aufweist, wobei die Vorrichtung zur Einstellung der Gewichtsvektorelementwerte des weiteren dazu geeignet ist, die zuvor getrennt gespeicherten bis dato besten Gewichtselementwerte durch die erhaltenen eingestellten Gewichtselementwerte zu ersetzen, wenn der bestimmte globale Qualitätswert näher an das globale Qualitätskrite-

rium herankommt als der globale Qualitätswert entsprechend den zuvor getrennt gespeicherten bis dato besten Gewichtswerten.

42. System zur Klassifizierung von Eingabedatenbeispielen in zumindest eine Mehrzahl von Klassen, das aufweist:
eine Speichervorrichtung zur Speicherung einer Anzahl oder eines Satzes von n-Tupeln oder Referenztabellen, wobei jedes n-Tupel oder jede Referenztabelle eine Anzahl von Zeilen entsprechend zumindest einer Untermenge der Anzahl möglicher Klassen und des weiteren eine Anzahl von Spaltenvektoren mit entsprechenden Gewichtsvektoren aufweist, wobei jeder Spaltenvektor durch Signale oder Elemente eines stichprobenartig ausgewählten Eingabedatenbeispiels angesprochen wird, und jeder Spaltenvektor und Gewichtsvektor Elemente mit Werten aufweist, die während eines Trainingsprozesses bestimmt werden, der auf einem Satz oder mehreren Sätzen von Trainingseingabedatenbeispielen basiert, und wobei das System des weiteren aufweist:

eine Eingabevorrichtung zur Aufnahme eines zu klassifizierenden Eingabedatenbeispiels,

eine Vorrichtung zur stichprobenartigen Auswahl des empfangenen Eingabedatenbeispiels und zum Ansprechen von Spalten- und entsprechenden Gewichtsvektoren in dem gespeicherten Satz von n-Tupeln oder Referenztabellen,

eine Vorrichtung zum Ansprechen spezieller Zeilen in dem Satz von n-Tupeln oder Referenztabellen, wobei die Zeilen einer speziellen Klasse entsprechen,

eine Vorrichtung zur Bestimmung eines Ausgabewerts als eine Funktion der angesprochenen Gewichtselemente, und

eine Vorrichtung zum Vergleichen der berechneten Ausgabewerte entsprechend allen Klassen und zur Auswahl der Klasse(n) mit dem maximalen Ausgabewert.

43. System nach Anspruch 42, **dadurch gekennzeichnet, daß** die Elementwerte der Spalten- und der Gewichtsvektoren unter Verwendung eines Trainingssystems gemäß einem der Systeme gemäß Ansprüchen 25 bis 41 bestimmt wurden.

44. System nach Anspruch 42, **dadurch gekennzeichnet, daß** die Elementwerte der Spalten- und der Gewichtsvektoren während eines Trainingsprozesses entsprechend einer der Methoden gemäß Ansprüchen 1 bis 20 bestimmt wurden.

45. System nach einem der Ansprüche 42 bis 44, **dadurch gekennzeichnet, daß** der Ausgabewert des weiteren als eine Funktion von Werten der angesprochenen Spaltenelemente bestimmt wird.

46. System nach einem der Ansprüche 42 bis 45, **dadurch gekennzeichnet, daß** die Vorrichtung zur Ausgabewertbestimmung eine Vorrichtung zur Erstellung einer ersten Summe aus allen angesprochenen Gewichtsvektorelementwerten entsprechend einer speziellen Klasse und entsprechend Spaltenvektorelementwerten größer oder gleich einem vorbestimmten Wert aufweist.

47. System nach Anspruch 46, **dadurch gekennzeichnet, daß** die Vorrichtung zur Ausgabewertbestimmung des weiteren eine Vorrichtung zur Erstellung einer zweiten Summe aus allen angesprochenen Gewichtsvektorelementwerten entsprechend einer speziellen Klasse sowie eine Vorrichtung zur Bestimmung des Ausgabewerts durch Teilung der ersten Summe durch die zweite Summe aufweist.

Revendications

1. Procédé pour former un système de classification informatique qui peut être défini par un réseau comportant de nombreux n-tuples ou de nombreuses Tables de Recherche (LUT), chaque n-tuple ou table LUT comportant de nombreuses rangées correspondant à au moins un sous-ensemble de classes possibles et comportant en outre de nombreuses colonnes adressées par des signaux ou des éléments d'exemples de données d'entrée de formation échantillonnés, chaque colonne étant définie par un vecteur ayant des cellules ayant des valeurs, ledit procédé comportant des ensembles d'exemples de données d'entrée pour différentes classes de sorte qu'au moins une partie des cellules comportent des informations ou pointent vers celles-ci sur la base du nombre de fois que l'adres-

se de cellule correspondante est échantillonnée à partir d'un ou plusieurs ensembles d'exemples d'entrée de formation, et

la détermination de valeurs de cellules de poids correspondant à une ou plusieurs cellules de vecteurs de colonnes qui sont adressées ou échantillonnées par les exemples de formation de manière à permettre la pondération d'une ou plusieurs cellules de vecteurs de colonnes de valeur positive pendant un traitement de classification, lesdites valeurs de cellules de poids étant déterminées sur la base des informations d'au moins une partie des valeurs de cellules de vecteurs de colonnes déterminées et en utilisant au moins une partie de l'ensemble ou des ensembles d'exemples d'entrée de formation.

2. Procédé selon la revendication 1, dans lequel les cellules de poids sont agencées dans des vecteurs de poids et la détermination des valeurs de cellules de poids comporte

l'initialisation d'un ou plusieurs ensembles de vecteurs de poids correspondant à au moins une partie des vecteurs de colonnes, et

l'ajustement des valeurs de cellules de vecteurs de poids d'au moins une partie des vecteurs de poids sur la base des informations d'au moins une partie des valeurs de cellules de vecteurs de colonnes déterminées et en utilisant au moins une partie de l'ensemble ou des ensembles d'exemples d'entrée de formation.

3. Procédé selon la revendication 2, dans lequel l'ajustement des valeurs de cellules de vecteurs de poids comporte les étapes consistant à

déterminer une valeur de qualité globale sur la base d'au moins une partie des valeurs de cellules de vecteurs de poids et de colonnes,

déterminer si la valeur de qualité globale satisfait à un critère de qualité requis, et

ajuster au moins une partie des valeurs de cellules de poids jusqu'à ce que le critère de qualité globale soit satisfait.

4. Procédé selon l'une quelconque des revendications 1 à 3, dans lequel les valeurs de cellules de poids sont déterminées de manière à permettre l'accentuation d'une ou plusieurs cellules de vecteurs de colonnes pendant un traitement de classification.

5. Procédé selon la revendication 4, dans lequel les valeurs de cellules de poids sont déterminées de manière à permettre la suppression d'une ou plusieurs cellules de vecteurs de colonnes pendant un traitement de classification.

6. Procédé selon l'une quelconque des revendications 1 à 5, dans lequel la détermination des valeurs de cellules de poids permet la pondération d'une ou plusieurs cellules de vecteurs de colonnes ayant une valeur positive (supérieure à 0) et d'une ou plusieurs cellules de vecteurs de colonnes ayant une valeur non-positive (inférieure ou égale à 0).

7. Procédé selon l'une quelconque des revendications 1 à 6, dans lequel les valeurs de cellules de poids sont agencées dans des vecteurs de poids correspondant à au moins une partie des vecteurs de colonnes.

8. Procédé selon l'une quelconque des revendications 1 à 7, dans lequel la détermination des cellules de poids permet la pondération d'une cellule de vecteur de colonne quelconque.

9. Procédé selon l'une quelconque des revendications 1 à 8, dans lequel au moins une partie des valeurs de cellules de colonnes sont déterminées en fonction du nombre de fois que l'adresse de cellule correspondante est échantillonnée à partir de l'ensemble ou des ensembles d'exemples d'entrée de formation.

10. Procédé selon l'une quelconque des revendications 1 à 9, dans lequel la valeur de vecteur de colonne maximale est égale à 1, mais au moins une partie des valeurs ont une valeur associée qui est fonction du nombre de fois que l'adresse de cellule correspondante est échantillonnée à partir de l'ensemble ou des ensembles d'exemples d'entrée de formation.

11. Procédé selon l'une quelconque des revendications 2 à 10, dans lequel les valeurs de cellules de vecteur de colonnes sont déterminées et mémorisées dans des moyens de mémorisation avant l'ajustement des valeurs de cellules de vecteurs de poids.

12. Procédé selon l'une quelconque des revendications 1 à 11, dans lequel la détermination des valeurs de cellules de vecteurs de colonnes comporte les étapes de formation consistant à

- a) appliquer un exemple de données d'entrée de formation d'une classe connue au réseau de classification, adressant ainsi un ou plusieurs vecteurs de colonnes,
- b) incrémenter, d'une manière préférée de un, la valeur ou le vote des cellules du vecteur ou des vecteurs de colonnes adressés correspondant à la rangée ou aux rangées de la classe connue, et
- c) répéter les étapes (a) à (b) jusqu'à ce que tous les exemples de formation aient été appliqués au réseau.

13. Procédé selon l'une quelconque des revendications 7 à 12, dans lequel tous les vecteurs de colonnes ont des vecteurs de poids correspondants.

14. Procédé selon l'une quelconque des revendications 2 à 13, dans lequel l'initialisation des vecteurs de poids comporte l'établissement de toutes les valeurs de cellules de vecteurs de poids à une valeur constante prédéterminée, ladite valeur prédéterminée étant d'une manière préférée égale à 1.

15. Procédé selon l'une quelconque des revendications 2 à 13, dans lequel l'initialisation des vecteurs de poids comporte l'établissement de chaque cellule de vecteur de poids à une valeur de cellule spécifique prédéterminée.

16. Procédé selon l'une quelconque des revendications 2 à 15, dans lequel l'ajustement des valeurs de cellules de poids comporte les étapes consistant à

- a) sélectionner un exemple de données d'entrée à partir de l'ensemble ou des ensembles de formation,
- b) déterminer une valeur de qualité locale correspondant à l'exemple d'entrée de formation échantillonné, la valeur de qualité locale étant une fonction d'au moins une partie des valeurs de cellules de vecteurs de poids et de colonnes adressées,
- c) déterminer si la valeur de qualité locale satisfait à un critère de qualité locale requis, sinon, ajuster une ou plusieurs valeurs parmi les valeurs de cellules de vecteurs de poids adressées si le critère de qualité locale n'est pas satisfait,
- d) sélectionner un nouvel exemple d'entrée à partir d'un nombre prédéterminé d'exemples de l'ensemble ou des ensembles de formation,
- e) répéter les étapes (b) à (d) de test de qualité locale pour tous les exemples d'entrée de formation prédéterminés,
- f) déterminer une valeur de qualité globale sur la base d'au moins une partie des vecteurs de poids et de colonnes qui sont adressés pendant le test de qualité locale,
- g) déterminer si la valeur de qualité globale satisfait à un critère de qualité globale requis, et,
- h) répéter les étapes (a) à (g) jusqu'à ce que le critère de qualité globale soit satisfait.

17. Procédé selon la revendication 16, dans lequel les étapes (b) à (d) sont réalisées pour tous les exemples de l'ensemble ou des ensembles de formation.

18. Procédé selon l'une quelconque des revendications 3 à 17, dans lequel le critère de qualité globale et/ou locale est modifié pendant le traitement d'itération d'ajustement.

19. Procédé selon l'une quelconque des revendications 3 à 18, dans lequel le traitement d'itération d'ajustement est arrêté si le critère de qualité globale n'est pas satisfait après un nombre donné d'itérations.

20. Procédé selon l'une quelconque des revendications 16 à 19, dans lequel les valeurs de cellules de poids ajustées sont mémorisées après chaque ajustement, et dans lequel la détermination de la valeur de qualité globale est suivie en outre par

la mémorisation d'une manière séparée des valeurs de cellules de poids ainsi obtenues ou des valeurs de configuration du système de classification si la valeur de qualité globale déterminée est plus près de satisfaire au critère de qualité globale que la valeur de qualité globale correspondant aux valeurs de cellules de poids ou aux valeurs de configuration précédemment mémorisées d'une manière séparée.

21. Procédé pour classer un exemple de données d'entrée dans au moins une classe parmi une pluralité de classes en utilisant un système de classification informatique configuré conformément à un procédé selon l'une quelconque des revendications 1 à 20, où les valeurs de cellules de vecteurs de colonnes et les valeurs de cellules de vecteurs

de poids correspondantes sont déterminées pour chaque n-tuple ou table LUT sur la base d'un ou plusieurs ensembles d'exemples de données d'entrée de formation, ledit procédé comportant

- a) l'application d'un exemple de données d'entrée destiné à être classé au réseau de classification configuré de manière à adresser des vecteurs de colonnes et des vecteurs de poids correspondants dans l'ensemble de n-tuples ou de tables LUT,
- b) la sélection d'une classe de manière à adresser des rangées spécifiques se trouvant dans l'ensemble de n-tuples ou de tables LUT,
- c) la détermination d'une valeur de sortie sous la forme d'une fonction de valeurs de cellules de poids adressées,
- d) la répétition des étapes (b) et (c) jusqu'à ce qu'une sortie ait été déterminée pour toutes les classes,
- e) la comparaison des valeurs de sortie calculées, et
- f) la sélection de la classe ou des classes ayant une valeur de sortie maximale.

22. Procédé selon la revendication 21, dans lequel la valeur de sortie est en outre déterminée sous la forme d'une fonction de valeurs de cellules de colonnes adressées.

23. Procédé selon la revendication 22, dans lequel ladite valeur de sortie est déterminée en tant que première sommation de toutes les valeurs de cellules de vecteurs de poids adressées correspondant à des valeurs de cellules de vecteurs de colonnes supérieures ou égales à une valeur prédéterminée, ladite valeur prédéterminée étant d'une manière préférée égale à 1.

24. Procédé selon la revendication 22, dans lequel ladite étape de détermination d'une valeur de sortie comporte les étapes consistant à

- déterminer une première sommation de toutes les valeurs de cellules de vecteurs de poids adressées correspondant à des valeurs de cellules de vecteurs de colonnes supérieures ou égales à une valeur prédéterminée,
- déterminer une seconde sommation de toutes les valeurs de cellules de vecteurs de poids adressées, et
- déterminer la valeur de sortie en divisant la première sommation par la seconde sommation.

25. Système pour former un système de classification informatique qui peut être défini par un réseau comportant de nombreux n-tuples ou de nombreuses Tables de Recherche (LUT) mémorisés, chaque n-tuple ou table LUT comportant de nombreuses rangées correspondant à au moins un sous-ensemble de classes possibles et comportant en outre de nombreuses colonnes adressées par des signaux ou des éléments d'exemples de données d'entrée de formation échantillonnés, chaque colonne étant définie par un vecteur ayant des cellules ayant des valeurs, ledit système comportant

- des moyens d'entrée pour recevoir des exemples de données d'entrée de formation de classes connues,
- des moyens pour échantillonner les exemples de données d'entrée reçus et adresser des vecteurs de colonnes se trouvant dans l'ensemble de n-tuples ou de tables LUT mémorisés,
- des moyens pour adresser des rangées spécifiques se trouvant dans l'ensemble de n-tuples ou de tables LUT, lesdites rangées correspondant à une classe connue,
- des moyens de mémorisation pour mémoriser des n-tuples ou des tables LUT déterminés,
- des moyens pour déterminer des valeurs de cellules de vecteurs de colonnes de manière à comporter des informations ou à pointer vers celles-ci sur la base du nombre de fois que l'adresse de cellule correspondante est échantillonnée à partir de l'ensemble ou des ensembles d'exemples d'entrée de formation, et
- des moyens pour déterminer des valeurs de cellules de poids correspondant à une ou plusieurs cellules de vecteurs de colonnes adressées ou échantillonnées par les exemples de formation de manière à permettre la pondération d'une ou plusieurs cellules de vecteurs de colonnes de valeur positive pendant un traitement de classification, lesdites valeurs de cellules de poids étant déterminées sur la base des informations d'au moins une partie des valeurs de cellules de vecteurs de colonnes déterminées et de l'utilisation d'au moins une partie de l'ensemble ou des ensembles d'exemples d'entrée de formation.

26. Système selon la revendication 25, dans lequel les moyens pour déterminer les valeurs de cellules de poids comportent

- des moyens pour initialiser un ou plusieurs ensembles de vecteurs de poids correspondant à au moins une partie des vecteurs de colonnes, et

des moyens pour ajuster des valeurs de cellules de vecteurs de poids d'au moins une partie des vecteurs de poids sur la base des informations d'au moins une partie des valeurs de cellules de vecteurs de colonnes déterminées et de l'utilisation d'au moins une partie de l'ensemble ou des ensembles d'exemples d'entrée de formation.

5

- 27.** Système selon la revendication 26, dans lequel les moyens pour ajuster les valeurs de cellules de vecteurs de poids sont adaptés pour

10

déterminer une valeur de qualité globale sur la base d'au moins une partie des valeurs de cellules de vecteurs de poids et de colonnes,
déterminer si la valeur de qualité globale satisfait à un critère de qualité globale requis, et
ajuster au moins une partie des valeurs de cellules de poids jusqu'à ce que le critère de qualité globale soit satisfait.

15

- 28.** Système selon l'une quelconque des revendications 25 à 27, dans lequel les moyens pour déterminer les valeurs de cellules de poids sont adaptés pour déterminer ces valeurs de manière à permettre l'ajout d'une ou plusieurs cellules de vecteurs de colonnes pendant un traitement de classification.

20

- 29.** Système selon la revendication 28, dans lequel les moyens pour déterminer les valeurs de cellules de poids sont adaptés pour déterminer ces valeurs de manière à permettre la suppression d'une ou plusieurs cellules de vecteurs de colonnes pendant un traitement de classification.

25

- 30.** Système selon l'une quelconque des revendications 25 à 29, dans lequel les moyens pour déterminer les valeurs de cellules de poids sont adaptés pour déterminer ces valeurs de manière à permettre la pondération d'une ou plusieurs cellules de vecteurs de colonnes ayant une valeur positive (supérieure à 0) et d'une ou plusieurs cellules de vecteurs de colonnes ayant une valeur non-positive (inférieure ou égale à 0).

30

- 31.** Système selon l'une quelconque des revendications 25 à 30, dans lequel les moyens pour déterminer les valeurs de cellules de poids sont adaptés pour déterminer ces valeurs de sorte que les valeurs de cellules de poids sont agencées dans des vecteurs de poids correspondant à au moins une partie des vecteurs de colonnes.

- 32.** Système selon l'une quelconque des revendications 25 à 31, dans lequel les moyens pour déterminer les valeurs de cellules de poids sont adaptés pour permettre la pondération d'une cellule de vecteur de colonne quelconque.

35

- 33.** Système selon l'une quelconque des revendications 25 à 32, dans lequel les moyens pour déterminer les valeurs de cellules de vecteurs de colonnes sont adaptés pour déterminer ces valeurs sous la forme d'une fonction du nombre de fois que l'adresse de cellule correspondante est échantillonnée à partir de l'ensemble ou des ensembles d'exemples d'entrée de formation.

40

- 34.** Système selon l'une quelconque des revendications 25 à 32, dans lequel les moyens pour déterminer les valeurs de cellules de vecteurs de colonnes sont adaptés pour déterminer ces valeurs de sorte que la valeur maximale est égale à 1, mais au moins une partie des valeurs ont une valeur associée qui est une fonction du nombre de fois que l'adresse de cellule correspondante est échantillonnée à partir de l'ensemble ou des ensembles d'exemples d'entrée de formation.

45

- 35.** Système selon l'une quelconque des revendications 25 à 34, dans lequel, lorsqu'un exemple de données d'entrée de formation appartenant à une classe connue est appliqué au réseau de classification en adressant ainsi un ou plusieurs vecteurs de colonnes, les moyens pour déterminer les valeurs de cellules de vecteurs de colonnes sont adaptés pour incrémenter la valeur ou le vote des cellules du vecteur ou des vecteurs de colonnes adressés correspondant à la rangée ou aux rangées de la classe connue, ladite valeur étant d'une manière préférée incrémentée de un.

50

- 36.** Système selon l'une quelconque des revendications 25 à 35, dans lequel tous les vecteurs de colonnes ont des vecteurs de poids correspondants.

55

- 37.** Système selon l'une quelconque des revendications 26 à 36, dans lequel les moyens pour initialiser les vecteurs de poids sont adaptés pour établir toutes les valeurs de cellules de vecteurs de poids à une valeur constante prédéterminée, ladite valeur prédéterminée étant d'une manière préférée égale à un.

38. Système selon l'une quelconque des revendications 26 à 37, dans lequel les moyens pour initialiser les vecteurs de poids sont adaptés pour établir chaque cellule de vecteur de poids à une valeur spécifique prédéterminée.

39. Système selon l'une quelconque des revendications 26 à 38, dans lequel les moyens pour ajuster les valeurs de cellules de vecteurs de poids sont adaptés pour

a) déterminer une valeur de qualité locale correspondant à un exemple d'entrée de formation échantillonné, la valeur de qualité locale étant une fonction d'au moins une partie des valeurs de cellules de vecteurs de poids et de colonnes adressées,

b) déterminer si la valeur de qualité locale satisfait à un critère de qualité locale requis,

c) ajuster une ou plusieurs valeurs parmi les valeurs de cellules de vecteurs de poids adressées si le critère de qualité locale n'est pas satisfait,

d) répéter le test de qualité locale pour un nombre prédéterminé d'exemples d'entrée de formation,

e) déterminer une valeur de qualité globale sur la base d'au moins une partie des vecteurs de poids et de colonnes adressés pendant le test de qualité locale,

f) déterminer si la valeur de qualité globale satisfait à un critère de qualité globale requis, et,

g) répéter le test de qualité locale et globale et les ajustements de poids associés jusqu'à ce que le critère de qualité globale soit satisfait.

40. Système selon l'une quelconque des revendications 26 à 39, dans lequel les moyens pour ajuster les valeurs de cellules de vecteurs de poids sont adaptés pour arrêter le traitement d'itération si le critère de qualité globale n'est pas satisfait après un nombre donné d'itérations.

41. Système selon l'une quelconque des revendications 27 à 39, dans lequel les moyens pour mémoriser les n-tuples ou les tables LUT comportent des moyens pour mémoriser des valeurs de cellules de poids ajustées et des moyens séparés pour mémoriser les meilleures valeurs de cellules de poids obtenues jusqu'à présent, lesdits moyens pour ajuster les valeurs de cellules de vecteurs de poids étant en outre adaptés pour

remplacer les meilleures valeurs de cellules de poids obtenues jusqu'à présent précédemment mémorisées d'une manière séparée par des valeurs de cellules de poids ajustées obtenues si la valeur de qualité globale déterminée est plus proche de satisfaire au critère de qualité globale que la valeur de qualité globale correspondant aux meilleures valeurs de poids obtenues jusqu'à présent précédemment mémorisées d'une manière séparée.

42. Système pour classer des exemples de données d'entrée dans au moins une classe parmi une pluralité de classes, ledit système comportant :

des moyens de mémorisation pour mémoriser un certain nombre ou un ensemble de n-tuples ou de Tables de Recherche (LUT), chaque n-tuple ou table LUT comportant de nombreuses rangées correspondant à au moins un sous-ensemble du nombre de classes possibles et comportant en outre de nombreux vecteurs de colonnes ayant des vecteurs de poids correspondants, chaque vecteur de colonne étant adressé par des signaux ou des éléments d'un exemple de données d'entrée échantillonné et chaque vecteur de colonne et vecteur de poids ayant des cellules ayant des valeurs déterminées pendant un traitement de formation sur la base d'un ou plusieurs ensembles d'exemples de données d'entrée de formation, ledit système comportant en outre :

des moyens d'entrée pour recevoir un exemple de données d'entrée à classer,

des moyens pour échantillonner l'exemple de données d'entrée reçu et adresser des vecteurs de colonnes et de poids correspondants dans l'ensemble de n-tuples ou de tables LUT mémorisés,

des moyens pour adresser des rangées spécifiques de l'ensemble de n-tuples ou de tables LUT, lesdites rangées correspondant à une classe spécifique,

des moyens pour déterminer une valeur de sortie sous la forme d'une fonction de cellules de poids adressées, et

des moyens pour comparer des valeurs de sortie calculées correspondant à toutes les classes et sélectionner la classe ou les classes ayant une valeur de sortie maximale.

43. Système selon la revendication 42, dans lequel les valeurs de cellules des vecteurs de colonnes et des vecteurs de poids ont été déterminées par l'utilisation d'un système de formation conformément à l'un quelconque des systèmes selon les revendications 25 à 41.

44. Système selon la revendication 42, dans lequel les valeurs de cellules des vecteurs de colonnes et des vecteurs de poids ont été déterminées pendant un traitement de formation conformément à l'un quelconque des procédés

selon les revendication 1 à 20.

45. Système selon l'une quelconque des revendications 42 à 44, dans lequel la valeur de sortie est en outre déterminée sous la forme d'une fonction de valeurs de cellules de colonnes adressées.

46. Système selon l'une quelconque des revendications 42 à 45, dans lequel les moyens de détermination de sortie comportent des moyens pour produire une première sommation de toutes les valeurs de cellules de vecteurs de poids adressées correspondant à une classe spécifique et correspondant à des valeurs de cellules de vecteurs de colonnes supérieures ou égales à une valeur prédéterminée.

47. Système selon la revendication 46, dans lequel les moyens de détermination de sortie comportent en outre des moyens pour produire une seconde sommation de toutes les valeurs de cellules de vecteurs de poids adressées correspondant à une classe spécifique, et des moyens pour déterminer la valeur de sortie en divisant la première sommation par la seconde sommation.

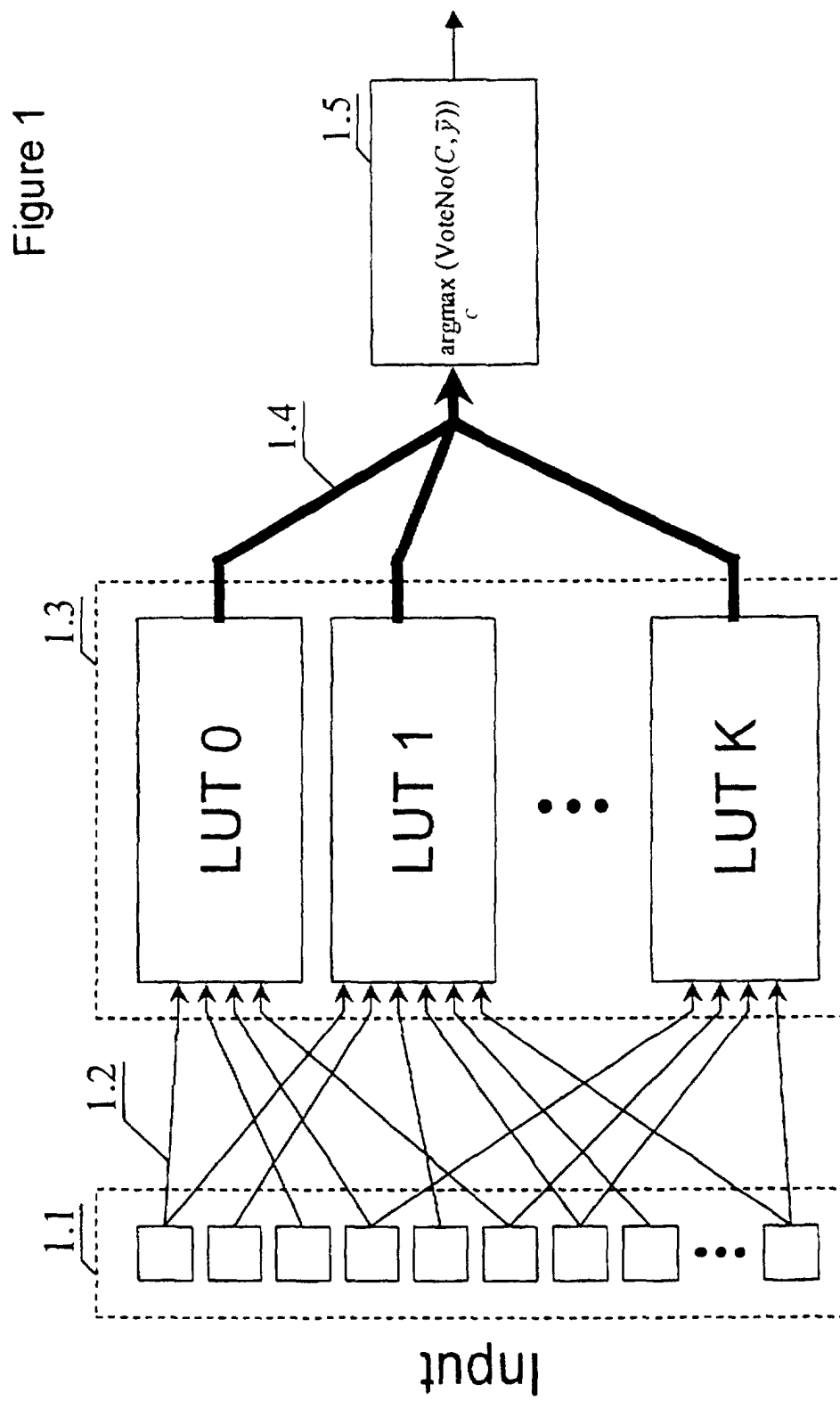
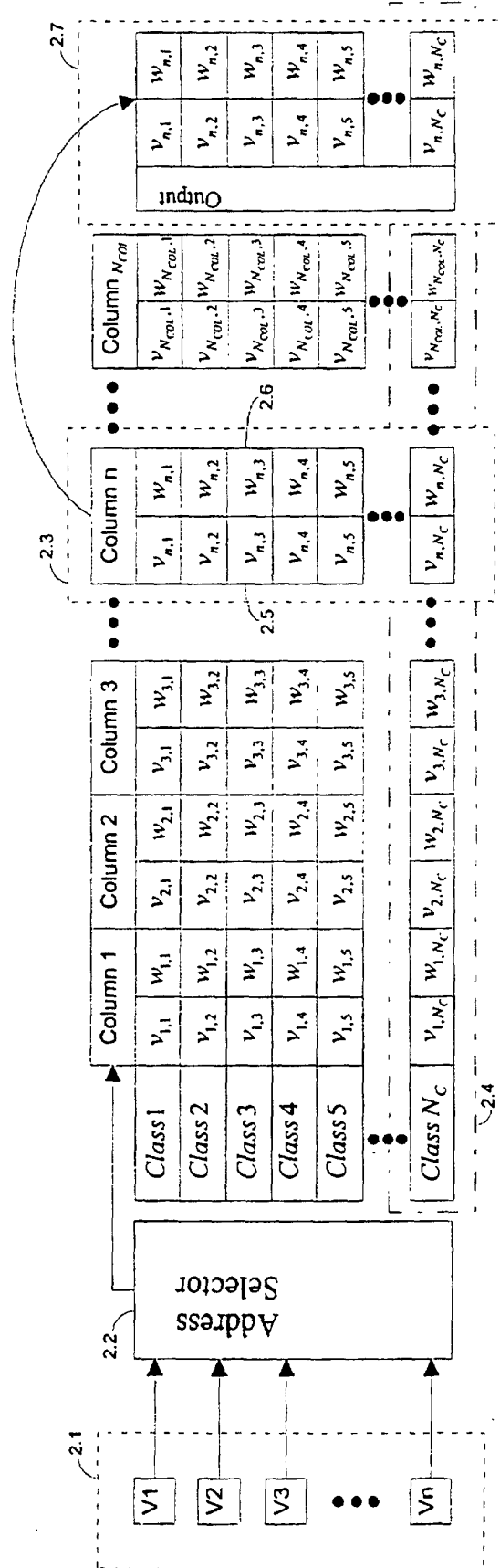


Figure 2



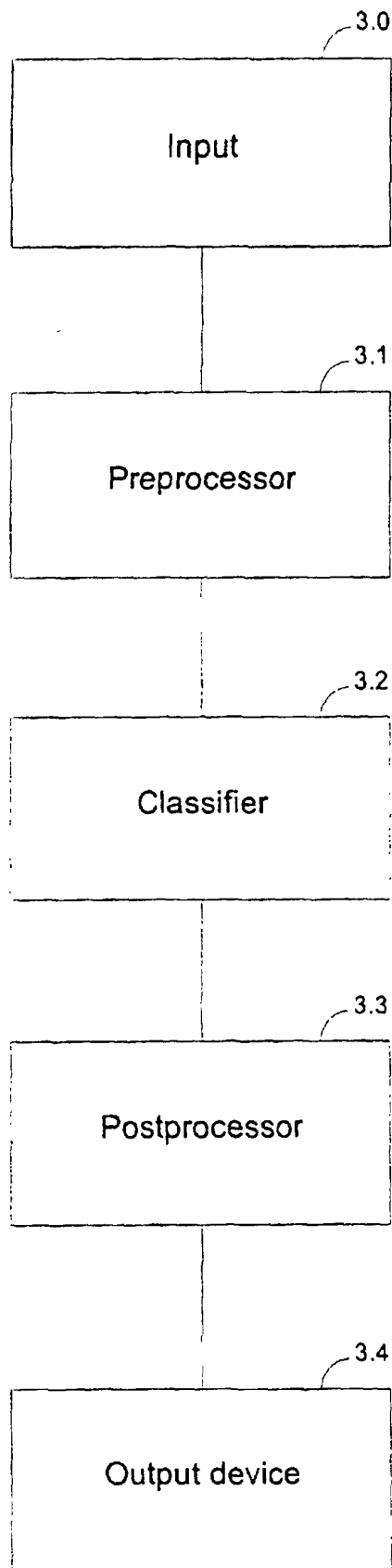


Figure 3

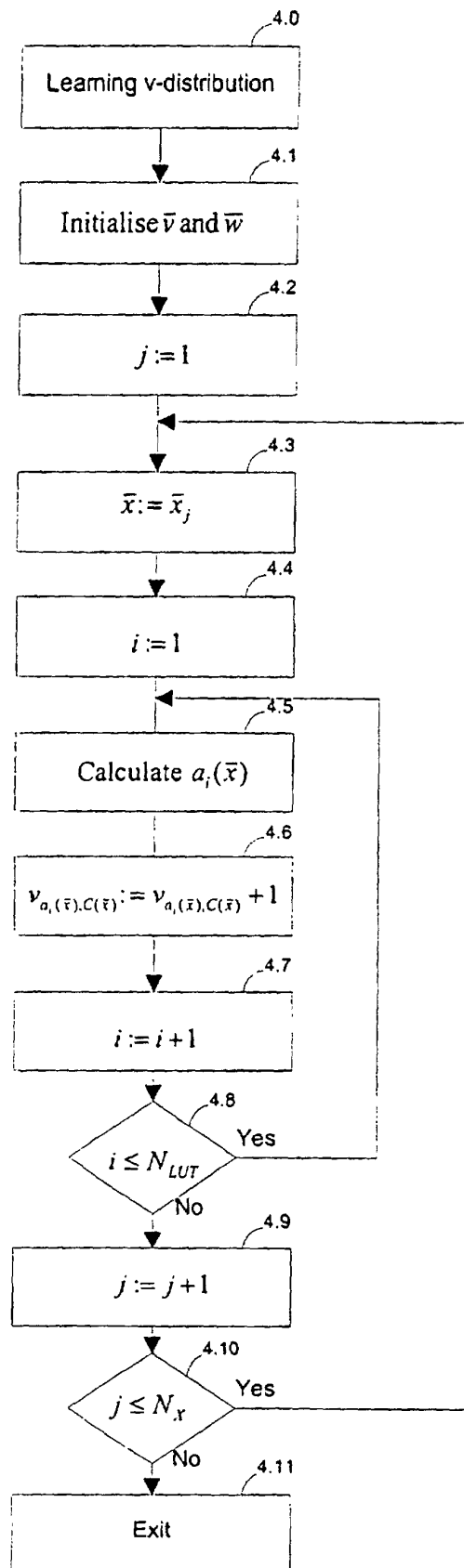


Figure 4

Figure 5

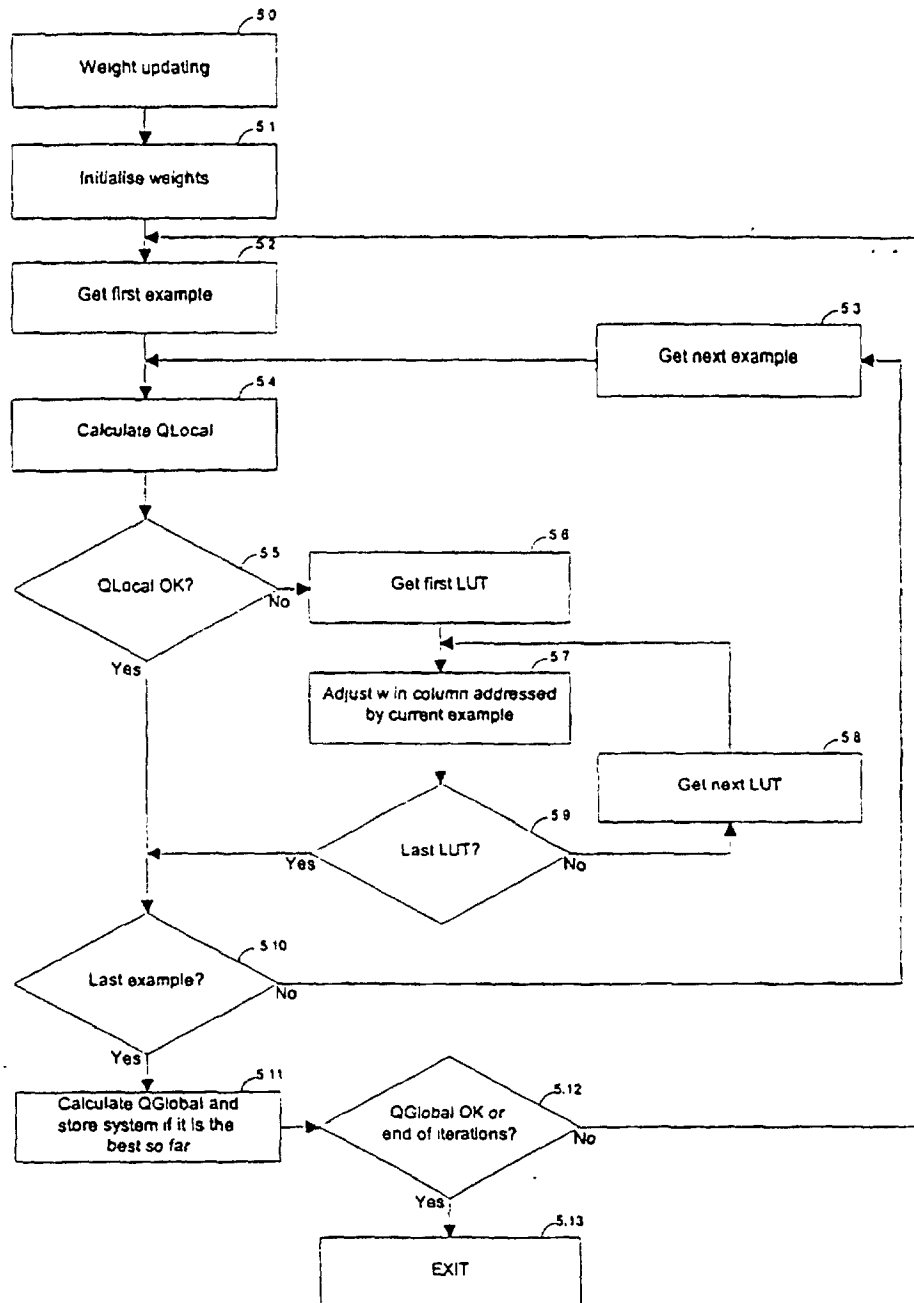
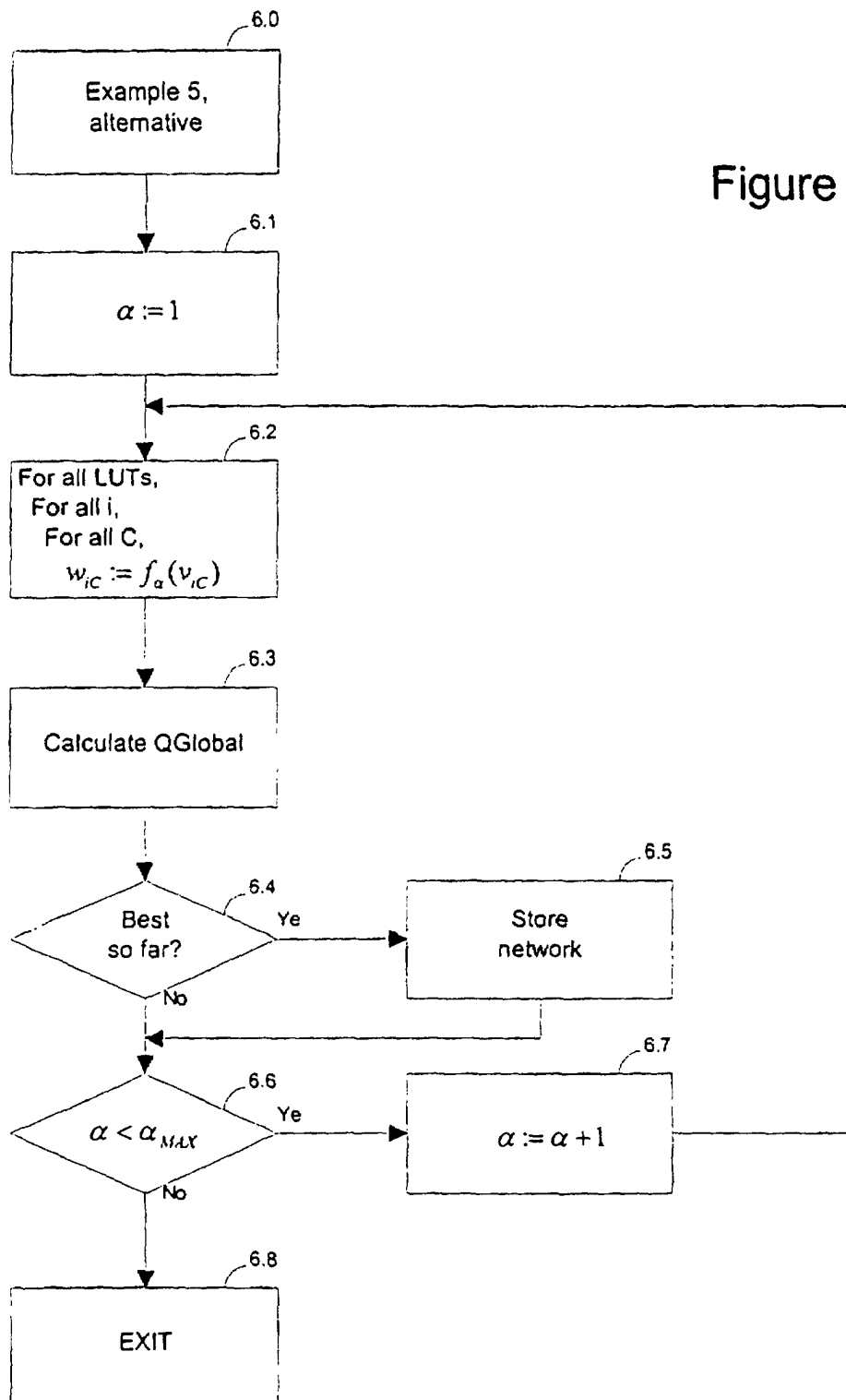


Figure 6



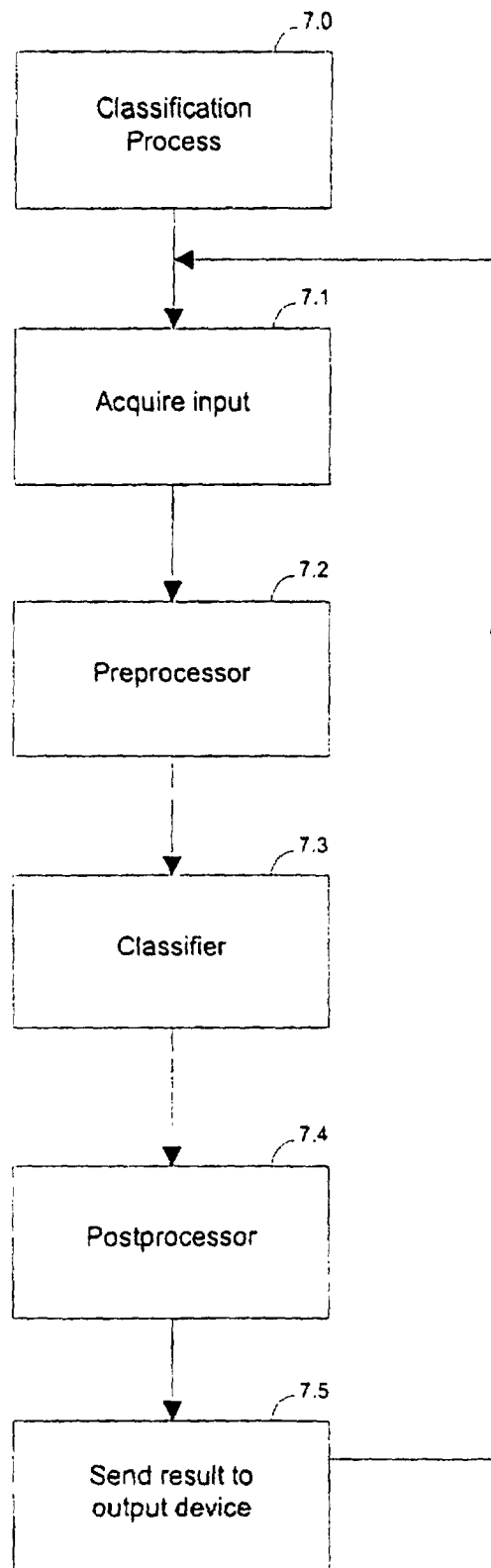


Figure 7