KAuditor User's Guide
Version 0.3.1

Marko Grönroos (magi@iki.fi)
Institute of Information Systems
Åbo Akademi University

March 26, 2001

Contents

Chapter 1

Introduction

KAuditor is an auditing tool for analyzing monthly balances of multiple accounts. The program uses several methods to make predictions of what each month's account should be, and this value can then be compared to the actual value and budgeted value. If the difference is noticeable, more attention should be directed to that month.

Following prediction methods are supported:

- *Zero delta prediction.* This method doesn't actually predict anything. Any methods that are considered useful are expected to give better predictions.
- *Monthly average prediction.* This very trivial method predicts the montly balance as the average of the balances of the same month in the previous years.
- *Average delta prediction.* This trivial method calculates average monthly changes from previous years, and makes the predictions by adding the change to the account value of the previous month.
- *Combined trivial prediction.* This method combines the monthly average, average delta, and zero delta prediction. It is useful with accounts that have special absolute balances in certain months.
- *Neural network prediction.* Uses a neural network to first learn the behavious from previous years, and then predicts the monthly accounts for the inspected year. This method has been found to yield slightly better predictions than the other methods in our test cases.

The tool has so far been tested only with very few companies, and more general applicability is not known. The predictions are by no means reliable, but they may or may not give some understanding of the account behaviour to the auditor.

The neural prediction method is outlined in, for example, following papers:

> Koskivaara, Eija. *Artificial neural network models for predicting patterns in auditing monthly balances.* TUCS Technical Report No 67. 1996
>
> Koskivaara, Eija. Artificial neural network models for predicting patterns in auditing monthly balances. *Journal of Operational Research Society*, Volume 51, Number 9. 2000

Table 1.1 compares the methods by their capabilities. The capabilities are the ability to predict absolute (fixed) balances on certain months, to eliminate adverse effects of long-term trends with relative predictions, and to utilize possible relationships between different accounts to make better predictions. The last column reports the best prediction error with our primary research data (from one company). The neural network parameters were: 10000 cycles, delta0 = 0.1, max delta = 50.0, weigth decay = 0.99992, all variables as inputs, one output at a time, four input months, network topology 48-20-20-10-1, global linear range equalization, one run (this network took hours to train, and the advantage to faster models was very small - only about 14 in error).

Prediction method	Absolute	Relative	Relationships	Error
Zero delta	-	-	-	660
Average	Yes	No	No	559
Average delta	No	Yes	No	427
Combined trivial	Yes	Yes	No	355
Neural network	Yes	Perhaps	Perhaps	310

Table 1.1: Comparison of capabilities of prediction methods.

General system requirements

KAuditor works currently under Linux, with the KDE1 desktop environment. For installation instructions, see the Chapter 3: *Administrative tasks* in page 12.

Chapter 2

Usage

2.1 General Usage

The general procedure for analyzing account data is as follows:

1. **Open an account data file** from menu *File/Open...* or *File/Open Recent*
2. (Optionally) Set training options from menu *Options/Training Settings...*
3. **Train the network with the data** from menu *Data Analysis/Train neural network*
4. (Optionally) Open budget from menu *File/Open budget...*
5. **Inspect prediction statistics** from menu *Data Analysis/Prediction statistics...*
6. **Visualize the predictions** from menu *Data Analysis/Plot data...*

These steps and other optional procedures are explaned below.

2.1.1 Notes

Training options should typically not be altered during production use of KAuditor, as they should be robust enough to be suitable for all companies and data. See below for more details.

You can open a file from the menu *File/Open*, or you can give it as a command-line parameter for KAuditor.

2.2 Preparing a data file

Before you can analyze any data, you have to load it in KAuditor. Account data is loaded from menu *File/Open*.

The data file contains a number of monthly account balances from a continuous time period. The last 12 months of the data are considered as *test data*, and all the data before that is used as *training data*. These data sets are displayed in the main window of KAuditor. See Figure 2.1.

The training data is used to predict what the account balaces in the last year *should* be. The accountant compares that prediction to the actual balances, the test data in the file. If there is a great difference in a certain month's balance, the accountant may choose to pay more attention to it.

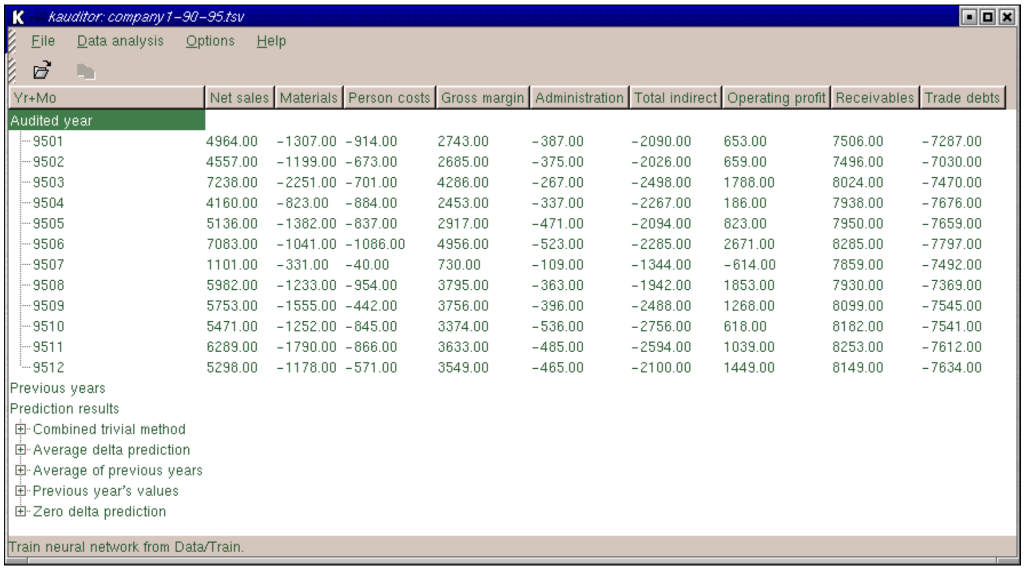


Figure 2.1: KAuditor main screen, with a data file loaded. The folder containing the inspected year, the "test data", is opened.

2.2.1 Data file format

The file format of the balance file must be exactly correct:

- The immediately first row is a **header row** that lists the names of the accounts, separated by tab (ASCII 0x08) characters. The first column of the header row is empty, followed by a single tab character.
- The rest of the rows are **data rows**. The columns are separated by a single tabulator character (ASCII 0x08), and the columns may not be empty.
- First column of each data row is a **time label**. Two first numbers of the time label are the year and two latter numbers are the month (YYMM format). The year is always given with two digits, so that 2001 is 01 (y2k problems won't occur).
- **Data columns** are monthly balances of the accounts, as listed in the header row. The decimal point *must* be a dot (`.`), and the values *must not* contain any non-numeric characters such as commas (often used as 1,000 -separator) or spaces.
- There must not be any extra empty fields in the ends of rows, or empty rows anywhere in the data file.
- The last row must have a trailing newline just like others, but no more that just that.

Below is an example from the beginning of an account data file.

```
Net sales    Materials    Person costs    Gross margin    Administration      Total indirect     Operating profit    Receivables     Trade debts
9001 4769 -974 -1041 2754 -297 -1713 1041 8184 -6891
9002 3794 -874 -717 2203 -112 -1074 1129 8612 -8334
9003 4846 -1315 -938 2593 -387 -2191 402 8613 -8153
9004 4452 -1564 -922 1966 -136 -1582 384 9204 -8071
9005 4665 -1283 -702 2680 -424 -1995 685 9492 -8704
9006 4420 -1301 -738 2381 -316 -2041 340 9818 -8541
9007 871 -242 -193 436 -92 -549 -113 9513 -8110
```

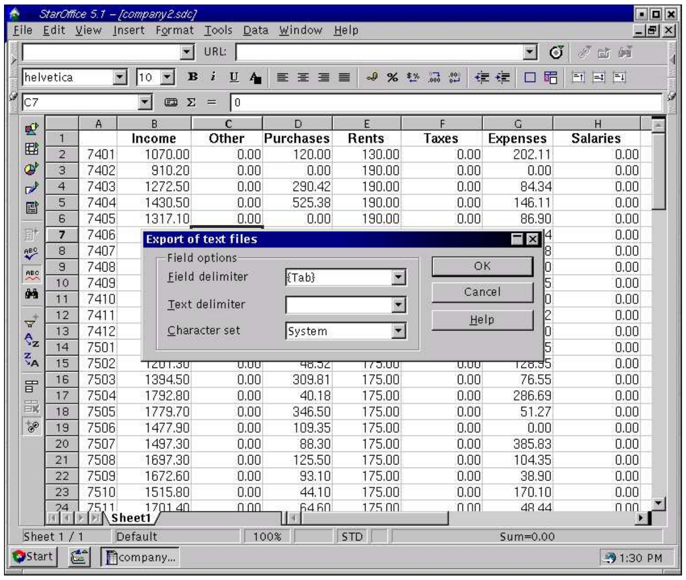


Figure 2.2: Exporting StarOffice sheet as a TSV text file.

```
9008 5561 -1451 -1046 3064 -355 -2274 790 9558 -8023
9009 3666 -1145 -799 1722 -268 -1731 -9 9624 -7983
9010 4179 -1043 -777 2359 -216 -1832 527 9615 -8091
9011 3924 -1013 -1077 1834 -295 -1305 529 9615 -8161
9012 2572 -792 -814 966 -290 -2166 -1200 8609 -7794
9101 3584 -723 -840 2021 -343 -1824 197 8866 -7693
9102 3151 -707 -568 1876 -436 -1517 359 7737 -7700
9103 3703 -825 -652 2226 -285 -1947 279 8820 -7299 ...
```

(The first row is not shown completely here.)

If the file format is not correct, unexpected behaviour may occur. The program displays an error message in case of most obvious errors. The program also checks that the order of months and years is correct.

2.2.2 Exporting a table from spreadsheets

Most spreadsheet programs support saving tables in the Tab Spaced Values (TSV) format required by KAuditor.

StarOffice

StarOffice is probably the best office software for Linux today. In addition to exporting the sheet as TSV, you can also select the relevant area from the sheet and Copy&Paste the data to a text editor (this does not currently seem to be possible with Gnumeric or KSpread).

To export a sheet as TSV, open menu *File/Save As*. Select the option "Text - txt - csv" from the file type selection box. The filename extension will change to `.txt`. A text file export dialog will appear, asking for field and text delimites. The field delimiter should be {tab}, and the text delimiter should be empty, as shown in Figure 2.2.

Gnumeric

Gnumeric supports the TSV file format. Use menu *File/Save As* and select the "Text File Export (*.csv)" option from the file format selection. Put `.tsv` as the filename extension. A dialog box appears that asks which sheets should be exported. Select the sheet, click *Add*, and then *Next*.

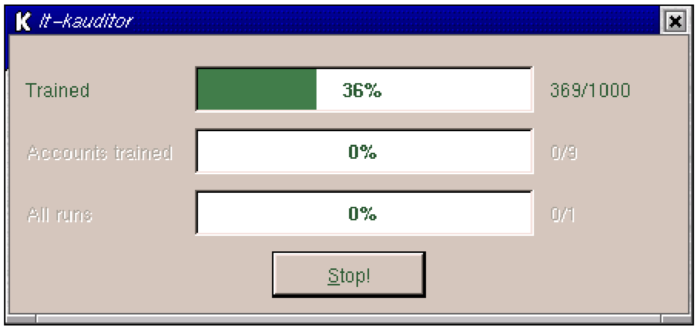


Figure 2.3: Training progress dialog.

Then give “Unix (linefeed)” as line termination, “tab” as the separator, and “Never” for quoting text.

KSpread

Gnumeric supports the TSV file format. Choose menu item *File/Save As*, select the CSV format, choose “tabulator” as the CSV delimiter, and save the file with name `datafile.tsv`.

Check the file format

If you haven’t done the exporting before, you might want to check that the file format is correct with a text editor (such as the KDE Advanced Editor or kwrite). Just remember to be careful with the tabulator characters, as they are important and should not be lost or changed as space characters.

2.2.3 Budget data files

Budget data files follow exactly the same format as the actual data file. They must have monthly balances for exactly 12 months. The column names in the first row are ignored. The number and order of the data columns must be exactly same as in the corresponding auditing file.

2.3 Trivial prediction methods

The predictions with the trivial methods are made immediately when a data file has been loaded. The predictions are displayed in the main window, as shown in Figure 2.1. You can click the folders to view the results. You can also view the error statistics and plot the results immediately.

2.4 Training the neural network predictor

The next step is to analyze the data with the neural learning algorithm. This is done from menu *Data Analysis/Train neural network*. The training can take a lot of time, even hours, depending on the training options. See Figure 2.3.

The predicted values are added to the “Neural prediction” folder in the main screen. When the data is selected, it can be copied to clipboard and pasted to another application.

If the training is stopped prematurely, all the current training is forgotten and the training cannot be resumed.

Saving and restoring a trained neural network

After a network has been trained with a particular dataset, it can be saved as a file. This is done from the menu *File/Save network*. Network file suffix should be .net. The file format is compatible with SNNS, except that requlariztion data is also attached to the neural network file.

A previously saved neural network can be restored from the menu *File/Load network*. Prediction is then done with the loaded network.

We strongly recommend that the network should only be applied to the exactly same dataset which it was originally trained with. If you really want to play with this, you should only apply a saved network to a formally equivalent dataset, i.e., one that has same number of account variables, in exactly same order. Making predictions for a company with a predictor that has been trained with another company, will very obviously give very bad results. The program will probably not give any errors if you misuse it in this way.

Notice that the training options are restored to the settings used for training the saved network.

2.5 Viewing the prediction errors

The monthly prediction errors provide the most valuable information for the auditor. They can be inspected by choosing the menu item *Data Analysis/Inspect errors*. The statistics dialog is shown in Figure 2.4.

The prediction results are chosen with control ***Prediction***. The neural network prediction is available only when the network has been trained with the data. The budget is available when it has been loaded.

The ***Display errors as*** -selection determines how the errors are calculated. It has currently four options:

1. ***Percentage of reported value***, which means simply $error = |x_{m,v} - o_{m,v}| \,/\, |o_{m,v}|$, where x is the predicted value, and o is the balance reported for the particular month and account.

2. ***Percentage of audited year's range of each account***, which means $error = 100 \cdot |x_{m,v} - o| \,/ (max_a(o_v) - min_a(o_v))$, where $max_a(o_v)$ and $min_a(o_v)$ are, respectively, the minimum and maximum values of that particular account over the audited year.

3. ***Percentage of entire range of each account***, which means $error = 100 \cdot |x_{m,v} - o| \,/ (max(o_v) - min(o_v))$, where $max(o_v)$ and $min(o_v)$ are, respectively, the minimum and maximum values of that particular account over all years.

4. ***Percentage of audited year's range of all account***s, which means $error = 100 \cdot |x - o| \,/ (max_a(o) - min_a(o))$, where $max_a(o)$ and $min_a(o)$ are , respectively, the minimum and maximum values for that account over the audited year.

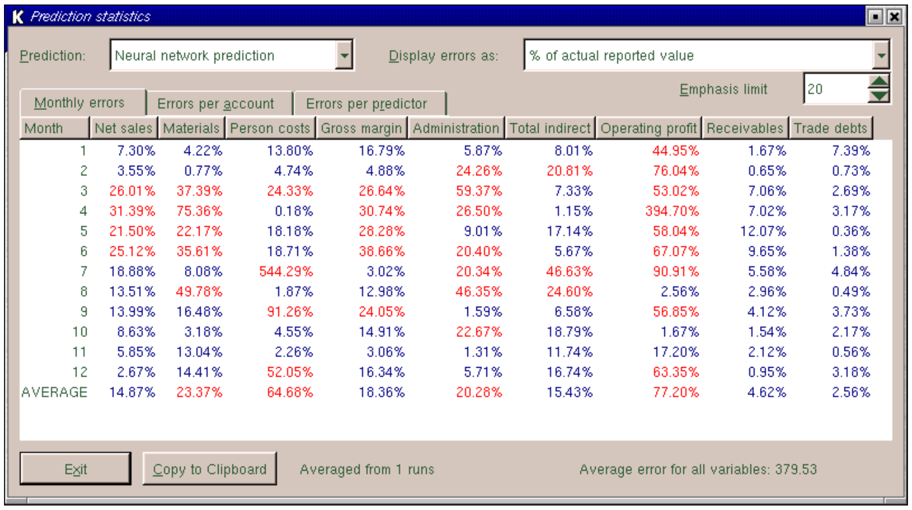
5. ***Percentage of entire range of all accounts***, which means $error = 100 \cdot |x - o| \,/ (max(o) - min(o))$, where $max(o)$ and $min(o)$ are , respectively, the minimum and maximum values for all accounts over all years.

6. ***Absolute error in currency***, which means simply error in the units of the data (for example, euros or dollars).

The ***Emphasis limit*** controls the percentage limit where the error values are emphasized with red color. Acceptable values are shown in blue.

The results are shown in three different view items. The item ***Monthly errors*** shows the monthly errors for each account. The item ***Errors per account*** shows the average errors over the 12 predicted months for each account. The item ***Errors per predictor*** shows the errors per account for all prediction methods.

The *Copy to CB* button copies the currently shown table to clipboard, so that it can be pasted to another application.



Month	Net sales	Materials	Person costs	Gross margin	Administration	Total indirect	Operating profit	Receivables	Trade debts
1	7.30%	4.22%	13.80%	16.79%	5.87%	8.01%	44.95%	1.67%	7.39%
2	3.55%	0.77%	4.74%	4.88%	24.26%	20.81%	76.04%	0.65%	0.73%
3	26.01%	37.39%	24.33%	26.64%	59.37%	7.33%	53.02%	7.06%	2.69%
4	31.39%	75.36%	0.18%	30.74%	26.50%	1.15%	394.70%	7.02%	3.17%
5	21.50%	22.17%	18.18%	28.28%	9.01%	17.14%	58.04%	12.07%	0.36%
6	25.12%	35.61%	18.71%	38.66%	20.40%	5.67%	67.07%	9.65%	1.38%
7	18.88%	8.08%	544.29%	3.02%	20.34%	46.63%	90.91%	5.58%	4.84%
8	13.51%	49.78%	1.87%	12.98%	46.35%	24.60%	2.56%	2.96%	0.49%
9	13.99%	16.48%	91.26%	24.05%	1.59%	6.58%	56.85%	4.12%	3.73%
10	8.63%	3.18%	4.55%	14.91%	22.67%	18.79%	1.67%	1.54%	2.17%
11	5.85%	13.04%	2.26%	3.06%	1.31%	11.74%	17.20%	2.12%	0.56%
12	2.67%	14.41%	52.05%	16.34%	5.71%	16.74%	63.35%	0.95%	3.18%
AVERAGE	14.87%	23.37%	64.68%	18.36%	20.28%	15.43%	77.20%	4.62%	2.56%

Figure 2.4: Prediction errors.

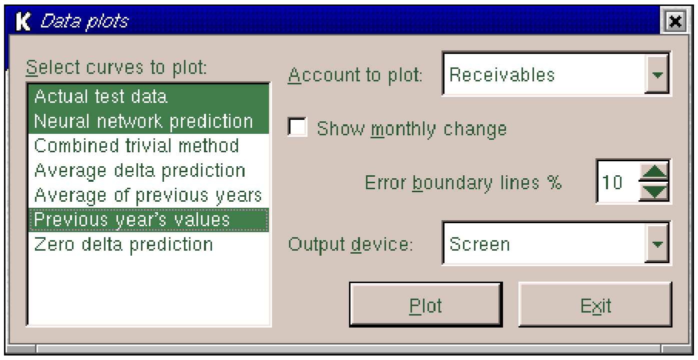


Figure 2.5: Plot dialog.

2.6 Viewing prediction graphs

The predictions can also be viewed graphically, by selecting the menu *Data Analysis/Plot predictions*.

Use the plot dialog, shown in Figure 2.5, to choose the graphs and account you want to view. Then push the ***Plot*** button to open a plot window. An example plot is shown in Figure 2.6. The plot window is updated automatically when you make changes to the selections in the plot dialog.

The neural network graph is available if the network has been trained. The budget graph is available if the budget file has been loaded.

The option ***Show monthly changes*** shows the predicted change from previous month. This may give better understanding of what the program actually does — predicts the value of next month according to the value of the previous month (and with neural network prediction this can actually be more than one month). Viewing the standard curves may easily lead to false intuition about the goodness of a prediction. For example, if the prediction is always "50 less than previous month" for every month, or any similar trivial systematic prediction, the curve may look somewhat fine, but actually it may not be a very good prediction.

The ***Error boundary lines %*** changes the width of the "accepted" region around the actual balance. Prediction values which lie outside this region are indicated with red error marks. Setting 0 disables the boundary lines and error marks.

You don't have to close the plot window, as it is updated when you make changes to the

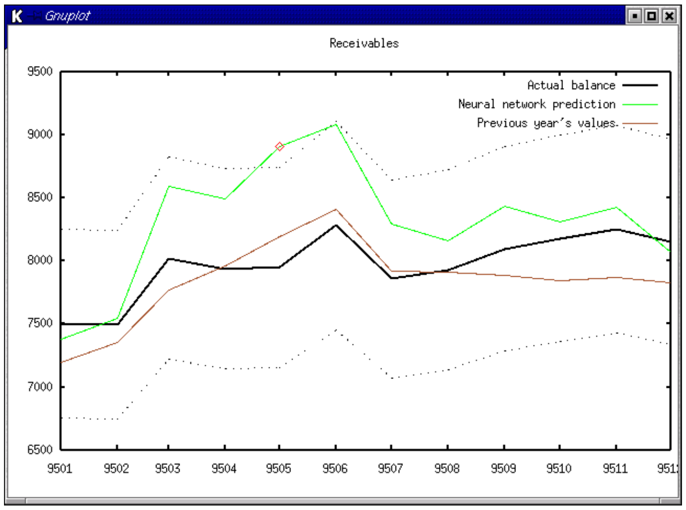


Figure 2.6: Prediction (not a very good one) plot without the *Show monthly change* option.

selections.

Exporting plots

The plots can be exported as EPS (Encapsulates PostScript) or GIF files. GIFs are useful for screen presentations because they use colors, and EPS for paper publications because they use dashed lines. Choose appropriate file type from the ***Display*** selection and push the *Plot* button to save the figure in file.

Chapter 3

Administrative tasks

3.1 Installation

3.1.1 How to obtain KAuditor

KAuditor is available as a binary package and as a source package. The source is available in Inanna (see below) CVS repository in SourceForge. See *http://inanna.sourceforge.net/* for further information about how to obtain the code from the CVS.

3.1.2 Binary installation

Requirements

- **KDE1** (K Desktop Environment) version 1.2.x. KDE 2.x will do, if their installation includes KDE1 compatibility package `kde1-compat`.
- **Qt-1.44**

Installation

You must have the KDE1 and Qt-1.44 runtime libraries installed first.

Then, download the RPM package from the website or FTP site. You can also download it with wget:

```
# wget http://www.iki.fi/magi/ohjelmointi/kauditor/kauditor-0.3-1.i386.rpm
```

Now install it with RPM. If you really have Qt-1.44 and KDE1 installed, but RPM still complains about their versions, just give the `--nodeps` option:

```
# rpm -i kauditor-0.3.1-1.i386.rpm
error: failed dependencies:
qt < 2.0 is needed by kauditor-0.3.1-1
# rpm -i --nodeps kauditor-0.3.1-1.i386.rpm
```

KAuditor is now installed and if you have KDE1, you can start the program immediately. If you have KDE2, you unfortunately need to start KAuditor from shell window and give the location of the `kde1-compat` compatibility library with:

```
$ export KDEDIR=/usr/lib/kde1-compat
$ kauditor
```

The KDEDIR path depends on where you have the KDE1 libraries installed. If you do not do this, KAuditor may not function properly, if at all.

3.1.3 Installation from sources

Installation can be done by compiling KAuditor from sources. The source RPM package is available from the website.

Compilation requires that following libraries have been installed before KAuditor:

- MagicCLib foundation class library, version 1.1 or a later compatible version. Available as a source package from *http://inanna.sourceforge.net/*
- Inanna artificial neural network library, version 0.3 or a later compatible version. Available as a source package from *http://inanna.sourceforge.net/*. MagiCLib must be compiled and installed before Inanna can be compiled.
- KDE version 1.1.2 libraries, particulary `kdesupport-1.1.2`, `kdelib-1.1.2`, and `kdecore-1.1.2` packages. These packages come with for example RedHat 6.x distributions. KDE version 2.x libraries are not compatible.
- Qt version 1.44. Later versions are not compatible with KDE 1.1.2.

MagiClib and Inanna are included in the KAuditor source RPM package, and compiled automatically.

Operating system

KAuditor has been developed and tested in Linux. It may work in other Unixes, but it is not guaranteed. It does not currently compile in Solaris.

3.2 Neural network training options

It should not be necessary for the end-user to change the neural network training options in routine operation. Many of the options are mostly for research purposes.

Changing the options to get better results for a particular case is considered *ad hoc* adaptation of rules, and violates scientific principles. The idea is to find parameters that work with certain "research" data sets, and then use them with actual work data.

The options dialog box is shown in Figure 3.1.

3.2.1 Resilient backpropagation parameters

Most important parameter is the number of ***training cycles***. Good values are between 1000 and 10000. Smaller values can give the prediction results faster, which can be an advantage sometimes. Results do suffer a little if the value is less than about 3000. It is unlikely that more training than 10000 would be needed.

The ***weight decay***, which is used by default, has been found very useful in the training, as it reduces overlearning and therefore increases the generalization ability. The optimal amount of decay is not known, and varies by data and amount of training data. Its optimal value is also affected by the network topology and the number of training cycles. We have found its correct value rather critical in our experiments. Value 0.999 (3 nines) seems to be too much, while 0.99999 (5 nines) seems to be too little. Values around 0.9999 (4 nines) and 0.99992 have given best results with our test data.

Delta0 and ***max delta*** are parameters spesific to resilient backpropagation. They are expected to be rather robust, so there shouldn't be much need to change them.

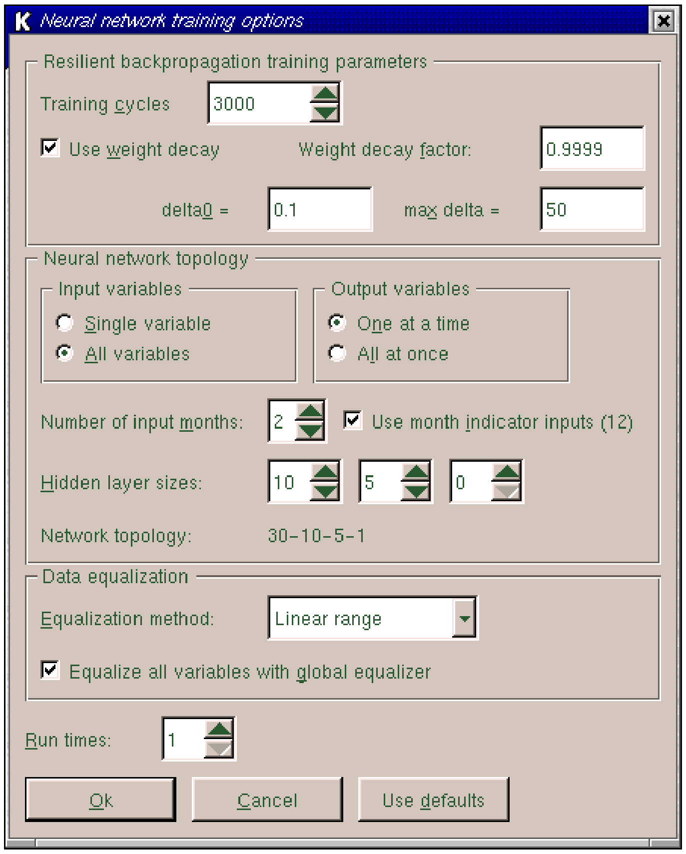


Figure 3.1: Training options dialog.

3.2.2 Neural network topology

The network topology (connection architecture or structure) used in KAuditor is a feedforward Multilayer Perceptron topology. The more spesific network topology is determined by a number of choises. The description of the resulting topology is shown in the bottom of the section, for example "21-10-5-9". The first value is the number of input units, which have different meanings. The last value is the number of output units. The values between are the sizes of the hidden layers. Each layer is fully connected to the next layer. Shortcut connections are not used.

Number of input variables

We have two options regarding the choise of variables given to the network. With the "***single variable***" option, each montly account value is predicted from the previous months values of only that account. With this option, it is possibly to predict only that single variable, and therefore there can be only one output variable. With the "***all variables***" option, each account is predicted from the values of all account in previous months.

Output variables

The prediction can be done by either ***one account at a time*** or ***all accounts at the same time***. In the former case, a separate neural network will be trained for each account. In the latter case, there is a single network with all the accounts in the output layer. It has been found that predicting just one account at a time yields better results. However, it increases the training time by many times.

Number of input months

The prediction of a month's account value can be done by giving the value from the previous month, or more previous months. The default value is two months.

Month indicator inputs

12 binary inputs are usually given to the network, which indicate the months given as inputs. They are very useful for learning, but can be disabled for experimentation purposes.

Sizes of the hidden layers

The network can have maximum of three hidden layers. If the size of a layer is set to zero, the layer is bypassed.

3.2.3 Data equalization

The data is usually equalized with linear equalization which maps the account values linearly to range [0,1]. This is currently the only equalization method supported.

It is possible to equalize the values either one account at a time or all accounts at same time (globally). If we equalize only one account at a time, the value 0 will refer to the smallest value in the particular account, and 1 to the largest. If we equalize all acounts at same time, 0 and 1 will refer to the minimum and maximum of the entire data. For unknown reason, the global equalization has been found to yield better predictions.

3.2.4 Other

The "***run times***" option determines how many times the network is initialized with random weights, trained and applied to make predictions. The predictions from the different runs are averaged, so we can expect the method to yield slightly more reliable results than with single runs. However, making multiple runs increases the training time greatly.

Chapter 4

Copyright and License