

Contents

About aiNet:

Features

Menu commands:

Static menus: (what is a static menu?)

File Menu

View Menu

Options Menu

Window Menu

Help Menu

Dynamic menus: (what is a dynamic menu?)

Edit Menu (Model Vectors View)

Edit Menu (Prediction View)

Edit Menu (Error Distribution View)

Model Vectors Menu (Model Vectors View)

Prediction (Prediction View)

Error Distribution(Error Distribution View & Charts View)

Toolbar:

Toolbar commands

Registration:

All about registration

What Is a Static Menu

Static menus are always present in the menu bar and are almost always accessible. See also [dynamic menus](#).

What Is a Dynamic Menu

Dynamic menus are view-dependent. When the **Model Vector View** window is active, for example, than **Edit** menu and **Model Vectors** menu appear on the menu bar and they disappear when some other view becomes active.

See also static menus.

Features of aiNet

aiNet application is a very powerful and a very simple tool for solving the problems which are usually solved with artificial neural networks (ANN). All possible tests we had run proved that the results obtained with aiNet are at least as good as the results obtained with some other ANNs. Let us state some of aiNet's features.

1. The major attribute that distinguishes aiNet from other ANNs is the analysis speed. Since aiNet uses an algorithm, which does not require any learning phase, the answers about prediction can be obtained almost immediately.
2. There is also only one coefficient (penalty coefficient), which has a major effect on the results. If we neglect some aspects, we can claim that knowing the right value for this coefficient solves the entire problem.
3. aiNet can dynamically change the "knowledge base". This means that you can add a new data to neural network (or remove old one), add additional variables (or remove old ones) and still get answers right away -- there is no time consuming learning phase.
4. Our experiments show us that noisy data is aiNet's favorite. If the data is just noisy, aiNet will give you excellent results. When you have chaotic data (and you do not know that), than you can not obtain any solution, still aiNet will assure you that something is wrong with the data.
5. aiNet provides you a way to estimate the rate of error in your prediction. If your problem is smooth, i.e. without noise, than this error will represent estimation for the error in the predicted result. If you have noisy data, than this will represent estimation for the noise around the predicted result. This means that an error estimation behaves locally.
6. aiNet is very suitable to work with missing values in your data. In real life problems it is usually very difficult to find a perfectly assembled knowledge base -- there is always some data missing. aiNet handles missing data automatically and you need not worry about how to represent such data.
7. aiNet's graphical user interface is very simple to use. It looks like a spreadsheet application and if you are familiar with any other spreadsheet, aiNet would not present a problem for you. Almost everything is only a mouse click away, menus are simple and there are also speed buttons.
8. On-line help is there for you. If you are stuck and do not know what to do just press F1 or select Help button and aiNet will help you find the way out.
9. Several charts are also available. They represent the most natural way to estimate how good your problem data is. They tell you visually if your problem can be generalized and what kind of results you can expect.

We are 100% sure that you will like our aiNet application.

Registration (Thank you for registering aiNet!)

[What Is Your Benefit](#)

[How to Register](#)

[Disclaimer of Warranty](#)

[Registration Form](#)

We did not cripple this program in any way. This means that this is a FULL version of aiNet application. Nothing has been disabled and there are no extra files.

We think that releasing a full version is necessary and it enables you to fully evaluate aiNet before you decide to register it. This also means that we are relying totally on your honesty to register.

What Is Your Benefit

If you become a registered user, you will be able to receive free upgrades to all future shareware releases of aiNet application, till (but not including) the next major release. We will number major upgrades in whole numbers and minor in .10 or .01 increments. As we already said, the minor upgrades will be free. For the new major releases we will give you a 50% discount if you are a registered user.

You will also have a chance to have an influence on the features of the future versions, the same is not guaranteed for unregistered users.

Registered users will have 100% support via e-mail or ordinary mail. This support is for correcting bugs in the software and manuals and does not include advice on how to solve various problems using neural network. We will also inform registered users about new issues of aiNet or of our similar applications, but only if we are allowed to do so.

A 50% discount will be available to all registered users, when they will consult us about solving various problems using aiNet application. We are also available for contract work on the use of neural networks.

How to Register

If you use aiNet for more than 21 days - three weeks - you are obligated to register it by paying the registration fee. To do so, fill the registration form, sign it and mail the payment to:

**aiNet
Trubarjeva 42
SI-3000 Celje
Slovenia
Europe**

If you are paying with a credit card, than you can also send the registration form via e-mail (ainet@ikpir.fagg.uni-lj.si). Just take a registration form template (ORDER.TXT), fill and drop it into your emailer. This will speed up our response.

As you will see from the registration form we provide two different ways for the registration. The first way is faster and slightly cheaper; we will mail (or e-mail) you your registration text and code. After you receive the text and code, you will enter them into the registration dialog box. This will remove all registration-demand messages and will explicitly register aiNet to you (registration text will appear in the caption bar).

If you choose the second way, we will send you aiNet application on a diskette. You will also receive the registration text and code, which can be used for next minor upgrades. Optionally, you may order also a hardcopy of the manual.

Currently, we accept two methods of payment: credit cards (preferable) and money checks. Generally, there are no problems with credit card payment, but some problems may occur with money checks. For this reason, we prefer a credit card payment.

Disclaimer of Warranty

This software and documentation are sold "as is" and without warranties as to performance of merchantability or any other warranties whether expressed or implied. Because of the various hardware and software environments into which this program may be put, no warranty of fitness for a particular purpose is offered. You use aiNet **entirely at your own risk**, and you supply it to your customers, friends, family or acquaintances **entirely at your own risk**.

In no event shall we (aiNet) be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use aiNet application, even if we have been advised of the possibility of such damages.

If these terms are not acceptable to you, then please **DELETE** all the files from your disks **immediately and permanently**.

aiNet 1.25 REGISTRATION FORM (USE CAPITAL LETTERS, PLEASE)

Name: _____

E-MAIL Address: _____

Company/Address: _____

Registration Name: _____ (Do not make it too long!)

(This name will appear on aiNet's caption bar.)

How do you want to receive a registration:

☐ I want to receive a registered version of aiNet on a diskette.

☐ I want to receive a registration card only (includes Registration Text & Code), but no diskette. Send it to me via ☐ mail, ☐ e-mail.

Price Calculation:

Registration Fee: _____ Single user registration fee: US\$49.
(Multiple users registration fee: Contact us)

Shipping & Handling: _____ Zero, if you do not order diskettes. Otherwise:
Europe US\$4, other countries US\$10.

Commercial DLL reg: _____ US\$69. You get source code, too (Free for non-
commercial and personal use. See below!)

(Checks only!): _____ US\$15 FALUS. Bank costs for checks are about US\$20
per check.

(Cash only!): _____ BONUS DISCOUNT if you pay with the cash. BONUS
US\$14 for application registration and US\$24 BONUS
for application & commercial DLL registration.

Total Costs: _____

Credit Card Orders: Please Check

☐ MasterCard, ☐ Visa, ☐ American Express, ☐ Eurocard, ☐ Diners Club

Cardholder's Name: _____

Address: _____

Credit Card Number: _____

Expiration Date: Month _____ Year _____

Credit card payments may be sent via mail or E-Mail!

Your Signature: _____

Check Payments: Make check payable in U.S. currency to:

Do not forget the extra \$15 for bank costs. (See above - Price calculation)

aiNet, Trubarjeva 42, SI-3000 Celje, Slovenia, Europe

Cash Payments: Enclose cash into a hard paper envelope. Be sure that the cash is not visible from outside. Embed the cash with the thick dark (black) paper once or twice. If you pay with the cash the application costs US\$35 and full registration (app. & commercial DLL) costs US\$94.

Mail the envelope to: aiNet, Trubarjeva 42, SI-3000 Celje, Slovenia, Europe

aiNet DLL library personal use registration:














By signing this form you agree that aiNet DLL library will not be used for

any other purposes but your own and that any product which will use aiNet DLL library will not be sold, resold or distributed in any form to the third party. (This signature is not required if you pay Commercial DLL reg. fee.)

Your Signature:_____

The Toolbar

The Toolbar is a row of buttons at the top of the main window which represent application commands. Clicking one of the buttons is a quick alternative to choosing a command from the menu. Buttons on the toolbar activate and deactivate according to the state of the application.

<u>Button</u>	Action	Menu Equivalent
	Create a new neural net.	File <u>N</u> ew
	Locate and open a neural net	File <u>O</u> pen
	Save the file in the active window	File <u>S</u> ave
	Cut selected text to Clipboard	Edit <u>C</u> ut
	Copy selected text to Clipboard	Edit <u>C</u> opy
	Paste text from Clipboard	Edit <u>P</u> aste
	Undo previous editor action	Edit <u>U</u> ndo
	Switch to Model Vectors View	View <u>M</u> odel <u>V</u> ectors
	Switch to Prediction View	View <u>P</u> rediction
	Switch to Error Distribution View	View <u>E</u> rror <u>D</u> istribution
	Switch to Charts View	View <u>C</u> harts <u>V</u> iew
	Calculate prediction, filtration or verification	Prediction&Error Distr. <u>C</u> alculate
	Display help file contents	Help <u>C</u> ontents

File Menu

The file menu provides commands for creating new files (in our case new artificial neural networks), opening existing files, closing files, saving files and exiting the aiNet application.

<u>N</u> <u>e</u> <u>w</u>	Create a new, untitled file - neural network.
<u>O</u> <u>p</u> <u>e</u> <u>n</u>	Open an existing file.
<u>C</u> <u>l</u> <u>o</u> <u>s</u> <u>e</u> <u>N</u> <u>e</u> <u>u</u> <u>r</u> <u>a</u> <u>l</u> <u>N</u> <u>e</u> <u>t</u>	Close the current neural network - document.
<u>C</u> <u>l</u> <u>o</u> <u>s</u> <u>e</u> <u>V</u> <u>i</u> <u>e</u> <u>w</u>	Close the current view.
<u>S</u> <u>a</u> <u>v</u> <u>e</u>	Save the current neural network.
<u>S</u> <u>a</u> <u>v</u> <u>e</u> <u>A</u> <u>s</u>	Save the current neural network under a new name.
<u>S</u> <u>a</u> <u>v</u> <u>e</u> <u>P</u> <u>r</u> <u>e</u> <u>d</u> <u>i</u> <u>c</u> <u>t</u> <u>i</u> <u>o</u> <u>n</u>	Save the data in the Prediction View window.
<u>O</u> <u>p</u> <u>e</u> <u>n</u> <u>P</u> <u>r</u> <u>e</u> <u>d</u> <u>i</u> <u>c</u> <u>t</u> <u>i</u> <u>o</u> <u>n</u>	Open an existing prediction file.
<u>E</u> <u>x</u> <u>i</u> <u>t</u>	Exit the aiNet application.

New Command

Using this command you order aiNet to open a brand new neural network. Before aiNet lets you enter some data into a new neural network, a dialog box pops up. The dialog asks you about the number of model vectors and the number of variables in the model vector. (You can always add or remove some model vectors or variables later.)

After you have filled the dialog, aiNet will ask you about the file format you want to use. Select the one you like and Model Vectors View window should appear on the screen.

Open Command

You use this command when you want to load some existing neural network from the disk. aiNet will display the File Open dialog. There you select the file you want to load. If the file is loaded successfully, aiNet will automatically open Model View and display model vectors from the file.

Close Neural Net Command

Close Neural Net command will close all opened views of the current neural network and will delete the network from memory. Before aiNet does that, it will remind you to (give you the last chance) to save the network.

Close View Command

aiNet will close the active view of the current neural network. If this view is the last opened view, you will be asked to save the neural network to the disk.

Save

Save command will save the current neural network to the disk. It usually takes a moment or two for aiNet to do this job. However, if the neural network does not have a file name yet and is labeled as Untitled1 or Untitled2 ... than the Save As command will be executed.

Save As

This command saves the current neural network to the disk, but before this is done, a dialog is displayed. In the dialog you must enter a file name and a location of the file.

You can also specify a file format you want to use. You can use either binary format (AIN extension) or ASCII comma delimited format (CSV extension). If you use the CSV format, than the neural network must be denormalized first.

Save As command also changes the text in the title bar of the views -- a new file name will appear within the square brackets.

Save Prediction

This command is enabled only if Prediction View is active. Save Prediction allows you to save the prediction vectors separately from the model vectors. This is very useful if you want to have several different independent sets of prediction vectors. Predictions are always saved in the ASCII comma delimited format, but with a PRD extension.

Open Prediction

This command does the opposite from the Save Prediction command. It is enabled only if Prediction View is active. It will ask you to select prediction file (PRD extension) you want to load. After you select a file, aiNet will discard previous contents of Prediction View and will load it with the prediction vectors from the file.

Exit

Use Exit when you want to quit the aiNet application. Before aiNet closes, it will ask you to confirm your decision. If you really want to quit aiNet, you will be further asked to save the neural networks.

View Menu

This menu allows you to bring up the view you want to see. If the view has not been created yet, than aiNet creates a window for the selected view and puts it on the top. If the view window already exists and is covered with other windows, than this command puts selected view window to the top. This rule holds for all four views.

Model Vectors View

Prediction View

Error Distribution View

Charts View

Sometimes some of the views are disabled. This usually happens when the model vectors in the Model Vectors View window are not normalized. Selecting the Model Vectors|Normalize + Lock command will enable all views.

Model Vectors View

Model Vectors View is a basic view, where all model vectors are presented in a spreadsheet form. The rows represent model vectors and the columns represent variables. Usually you can not see all the model vectors and variables at once, so you must do some scrolling. This view is automatically activated when you create a new neural network or you load an old neural network from the disk. You use this view to edit model vectors, to add some new or remove some old model vectors, etc. In this view you also normalize and denormalize model vectors.

Prediction View

Prediction View is used when you want to run some predictions. This view is very similar to Model Vectors View and allows you to take similar actions. You can edit the input part of the predictions vectors and then use aiNet to calculate the output part. Prediction view is therefore the most useful view.

Error Distribution View

When aiNet calculates the prediction it usually makes some mistakes -- the predictions are not 100% correct. Thanks to the special algorithms, aiNet can besides prediction itself predict also the ratio of the error in the prediction result. Before aiNet can do that, so called filtration and verification process must run over all model vectors. Error Distribution View shows the results of the filtration and verification. Like both views above, this view also uses spreadsheet-like window to present the results. This view is **read only** view, which means that you can browse through the results only, but you can not change them.

Charts View

Charts View is an extension of Error Distribution View, because this view represents the same results as Error Distribution View, but in the graphical ways. This enables you to estimate these results visually. Two different charts are currently available. The first one is some kind of correlation chart. On the horizontal axis are output values from model vectors (correct values) and on the vertical axis are values calculated by aiNet. If the calculated and the correct value for some model vector are the same (or almost the same), then the point of this model vector lies exactly on the diagonal line. In other words this means, closer the dots are to the diagonal, less error is present in the verification (red dots) or filtration (green dots). The second chart is a bar chart. It shows the verification or filtration error of each model vector. Model vectors are enumerated on the horizontal axis and the error is shown on the vertical axis.

Options Menu

Comment Width

When you want to change the width of the comment column in Model Vectors View, than you can do that by selecting this command.

Each model vector can have comments. Although you usually do not need them, you will encounter occasions, when they will be pure gold. Imagine you are collecting model vectors from some experiment, which is time dependent. You put these model vectors into the neural network. Now you make analysis and you notice that there is something wrong with some of the model vectors. If these vectors do not have a comment, it is hard to tell when were they measured. But if they do have a comment, a date and time, then it is easy.

Fonts

In aiNet version 1.1 this command is disabled. In later versions it will allow you to use different fonts in the views.

Register

If you decided to register the aiNet application, we will send you your unique registration code. You enter your name and code in a dialog box, which is invoked by Register command. After successful registration, aiNet stops displaying registration messages.

Window Menu

The Window menu provides commands to control the position and layout of aiNet's windows.

<u>Cascade</u>	Resize and position all windows in an overlapping pattern.
<u>Tile</u>	Resize and position all windows in an non overlapping pattern.
<u>Arrange Icons</u>	Align all iconized windows along a grid.
<u>Close All</u>	Close all windows.

Help Menu

The Help menu provides access to the help system and the about dialog.

Contents Help topic contents.

Window Arrange Icons Command

The **Window|Arrange Icons** command arranges all iconized windows into rows along the bottom of the application's main window.

Window Cascade Command

The **Window|Cascade** command arranges all document windows from the top-left position of the application's main window so that the title bar of each is visible.

Window Close All Command

The **Window|Close All** command closes all document windows open in the application.

Window Tile Command

The **Window|Tile** command arranges all document windows side-by-side in a non-overlapping pattern.

Window Help table of contents

The **Help|Contents** displays the help contents page.

Edit Menu (Model Vectors View)

The Edit menu provides several commands to edit model vectors.

Add rows (model vectors)

Add columns (variables)

Insert row/column

Delete row/column

Delete empty rows (model vectors)

Undo, Cut, Copy, Paste

Add rows (model vectors)

Select this command to add some extra model vectors to Model Vector View. aiNet will open a simple dialog. The dialog will ask you about the number of rows (model vectors) you want to add. Type the number and close the dialog with the OK button. The selected number of new rows will appear at the end of Model Vector View.

Add columns (variables)

If you realized that your model needs more variables, use this command to add some. A dialog will pop up asking you for the number of new variables. All new variables will have the exclude status turned on. (Exclude means neither input or output.)

Insert row/column

This command will be available only if you have previously selected a model vector (row) or a variable (column). If a row has been selected, than a new row (model vector) will be inserted before the selected row. The same is valid for the columns; a new column (variable) will be inserted before the selected column. The new variable will have the exclude status turned on.

Delete row/column

This command will be available only if you have previously selected a model vector (row) or a variable (column). Before the selection is deleted, the aiNet asks you to confirm the command.

Delete empty rows (model vectors)

This command deletes all model vectors with no entries. It sometimes happens that you have initially created too many rows or you may have added too many rows later. Nevertheless, it would be a good idea to delete such empty rows. This will not only free some memory, but it will speed up the computation as well.

Undo, Cut, Copy, Paste

Undo works within Edit window only, which is located in the speed button bar. The same is valid for the Cut, Copy and Paste commands.

Future releases of the aiNet application will probably further implement these commands.

Edit Menu (Prediction View)

The Edit menu provides several commands to edit predict vectors.

Show local error

Hide local error

Add rows (predictions)

Insert row

Delete row

Delete empty rows (predictions)

Undo, Cut, Copy, Paste

Show local error

aiNet can besides prediction calculate also an estimation for the error (or the noise rate) for every single prediction result. Show local error command displays these estimations in the Prediction View window.

Please note that if you really want to see these estimations you must calculate error distribution first. If error distribution is not calculated, than only empty lines are displayed.

Hide local error

Does the opposite of the Show local error command. It hides error estimations in the Prediction View window.

Add rows (predictions)

This command is the same as the Add rows (model vectors) command, except that it acts in the Prediction View window.

Insert row

A row must be selected, before this command can be used. It will insert a new prediction vector before selected row.

Delete row

A row must be selected, before this command can be used. It will delete the selected row. You will have to confirm this command.

Delete empty rows (predictions)

This command is the same as Delete empty rows (model vectors) command, except that it acts in the Prediction View window.

Edit Menu (Error Distribution View)

The Edit menu in the Error Distribution View window provides just two commands:

Show difference

Show predicted value

Error Distribution View is used to show the results of filtration (FE:) and verification (VE:). These results can be presented in two ways:

- As the difference between an initial (correct) and a calculated result (**Show difference** command),
- as a predicted value - calculated result (**Show predicted value** command).

Model Vectors Menu (Model Vectors View)

This menu is available only when Model Vectors View is the active window. Here are the commands it provides:

Normalize + Lock
Denormalize + Unlock
Normalization settings

Normalize + Lock

Before you can do any computation with aiNet, you must normalize the model vectors. After the normalization is done, aiNet will enable a lot of commands, which you will need to perform calculations. This command also locks the model vectors - you can not edit them.

Denormalize + Unlock

This command reverses normalized model vectors back to their original values. It also unlocks them and thus allows you to edit them. But do not forget that when model vectors are unlocked, you can not perform any calculations.

Normalization settings

Before aiNet can perform any kind of calculation, all model vectors must be normalized. This means that some real physical dimensions will be transformed to something else.

Two normalization methods are available:

- regular.
- statistical.

How to choose the right kind of normalization? Here is the rule we use (it is not an ultimate rule):

- * Use regular normalization if the data in model vectors is already normalized in some way -- if you know for sure that all values will fall within some physical limits.
- * When you can not say for sure that all the values will be between some limits, use statistical normalization.

In most cases the normalization type does not reflect significantly on the results. Please note that **normalization type does have an influence on the optimal value of penalty coefficient**. Statistical normalization usually requires larger values for the penalty coefficient.

Regular Normalization

If you choose a regular type of normalization, than aiNet will perform the following actions during normalization:

1. Scan trough all model vectors and find the largest (maximum) and the smallest value (minimum) for each variable.
2. For each variable calculate normalization factors in such a way, that all values in model vectors will be between -1 and 1 after normalization. (The maximum will be 1 and the minimum will be -1.)
3. Normalize all model vectors. Now all values in model vectors will be between -1 and 1.

Statistical Normalization

Here are the steps that aiNet will perform during statistical normalization:

1. Run through all model vectors and calculate an average and a standard deviation for each variable respectively.
2. For each variable calculate normalization factors in such a way that values, which are exactly one standard deviation apart from the mean value, will become 1 (or -1) after normalization.
3. Normalize all model vectors. All values that are now less than $|1|$ are those within one standard deviation, all others (greater than $|1|$) are more than one standard deviation apart from the mean value.

Prediction Menu (Prediction View)

This menu is available only when Prediction View is the active window. Here are the commands it provides:

Calculate prediction

Penalty settings

Normalization settings

Local error settings

Calculate prediction

When you have entered all input values in the prediction vectors, you can calculate the output values selecting the Calculate prediction command. In a moment or two results will appear in the output variables of the Prediction View window. If the error distribution was calculated before, than the estimation of the errors will also be calculated. The error distribution will be shown, if Edit|Show local error is selected.

Penalty settings

This is the most important command and we are sure you will use it a lot. When you select this command a dialog appears on the screen. The dialog asks you for the value of penalty coefficient and also lets you select a type of the coefficient.

The penalty coefficient value

Choosing the right value of the penalty coefficient usually solves the problem.

We can give you only informative instructions for selecting the right value. Our experiences can be summarized in the following simple rules:

- * The more model vectors you have, the smaller the optimal value is.
- * The more variables you have, the larger the optimal value is.
- * The more noise your data has, the larger the optimal value should be.
- * The best thing you can do is to try a few values. Do not forget to try some stupid values also.
- * Small penalty value leads to overfitting (overtraining) and
- * Large penalty value leads to underfitting (overgeneralizing).

The penalty coefficient type

The dialog that pops up gives two possible choices for the penalty coefficient type:

- * static
- * dynamic

Please note that switching from the static to the dynamic type (or opposite) reflects also on an optimal penalty coefficient value. The dynamic type usually leads to smaller values.

Note: There is a way to obtain a good estimation for penalty coefficient based on minimization of verification error. See also: [optimal penalty coefficient](#).

Static penalty coefficient type

The static coefficient type means that penalty coefficient value will be the same all over the hyper-dimensional space. Use this type when your model vectors are approximately uniformly (regularly) distributed. If your data is noisy, than this selection is also recommended.

Dynamic penalty coefficient type

This coefficient type behaves dynamically. When prediction, filtration and verification are calculated, than actual value of penalty coefficient depends on the density of model vectors. If there are a lot of model vectors in the surroundings of prediction vector, than penalty coefficient decreases -- fits little more. However, if model vectors are rare in the surroundings of prediction vector, than penalty coefficient increases -- generalizes little more. Use the dynamic coefficient type, when model vectors are grouped in clusters and are not uniformly distributed over hyper-dimensional space.

Local error settings

Using the Local Error Settings command you control the calculation of the error distribution. You can turn the error distribution on or off by selecting between these two possibilities:

- * **Enable (Yes, calculate local error),**
- * **Disable (Do not calculate local error).**

You can also select which type of local error you want to use. There are two types:

- * **Unsigned (absolute) local error,**
- * **Signed local error.**

Use an unsigned (absolute) type if you have noise in your model vectors. When you know that your problem is smooth and without any noise, than you may try with signed local error type.

The selection of local error type will have the greatest effect on prediction process, more precisely on estimation of local error in the prediction process.

So far we have been using two terms, which are related and have the same background:

an error distribution and
a local error (or local error estimation).

Error Distribution

Error distribution is used together with the terms filtration and verification. Actually, the error distribution is the result of filtration and verification. We named it distribution, because filtration and verification calculate an error for each model vector. When we know the error for each model vector, we also know how error is distributed over all model vectors, hence the term distribution.

Local Error (or Local Error Estimation)

Local error estimation is associated with prediction. If we know the error distribution and we run the prediction process, then we can predict not only the output variables, but also the error for every single prediction. Because such an error prediction can behave locally (it is not a constant in the hyperspace), we use the term local error. Since error distribution is always noisy, this error prediction can not be very accurate, hence the term estimation.

Error Distribution Menu

This menu is available only when Error Distribution View or Charts View is the active window. Here are the commands this menu provides:

Calculate error distribution

Global error report

Penalty settings

Normalization settings

Local error settings

Calculate error distribution

With this command you engage the calculation of error distribution. In a couple of moments results will appear in the output variables of the Error Distribution View window. If there are many model vectors, the calculation of the error distribution can take a considerable amount of time.

(NOTE: When you double the number of model vectors, than the calculation will take four times longer.)

Global error report

If you want to see global error estimates (root mean square error = RMS) than select this command. A dialog will be popped up, where the total RMS error will be shown, as well as the RMS errors for single output variables.

Exclude Selected Model Vectors Dialog

As soon as you select an area in the Correlation Chart this dialog will pop up. In the window you can see indexes of individual model vectors that have fallen in the selection area. Sometimes you can see that some model vectors appear twice in the window. This can happen only if both - filtration and verification - has been turned on and does have no side effects.

Currently, you can select between two commands:

The Exclude button

This button will mark all model vectors that appear in the window as excluded. After the dialog closes down, excluded model vectors will be visibly indicated by yellow background color. Additionally, all excluded model vectors will not influence the prediction and error distribution calculations.

The Cancel button

This button will simply close the dialog and return back to the chart window.

Optimal Penalty Coefficient Dialog

This dialog is used to determine the optimal value of the penalty coefficient. The criteria (goal function) will be verification error. The search will try to minimize the verification error as much as possible. You need to specify the initial search interval and the required precision of the solution. You do that by specifying three values:

Lower search bound	... is the initial search interval lower bound
Upper search bound	... is the initial search interval upper bound
Precision	... is the precision of the final solution.

Note: The number of iterations is correlated with the interval size and precision. The larger the initial interval is, more iterations is needed before a solution at specified precision will be found. The smaller the precision value is more iterations will be needed to fulfil the stop criteria.

It can happen that verification error has no minimum in the specified initial interval. In this case the result is either the upper or the lower bound of initial interval.

If you have a lot of model vectors, than you can estimate the optimal solution by reducing the set of model vectors in the verification calculations by specifying the percentage of model vectors used in optimization. aiNet will randomly select model vectors in specified percentage. Selected model vectors will be used in verification calculation. The relationship between percentage of used model vectors and calculation speed is linear.

Full model	... select to include all model vectors in the calculation.
Sub model	... select to include a specified percentage of model vectors in the model.
percentage	... a number between 1-100.

Once you have specified initial interval and size of the model you start the optimization process by selecting the **Search** button.

After the search has been completed you can select **OK** to keep newly defined penalty coefficient and **Cancel** to restore previous settings.

New Neural Network Dialog

The dialog asks you to specify a neural network size. The size is defined by the number of model vectors (learn samples) and by the number of variables (units) in a model vector.

Example

Assume, we have a multiplication problem:

$$\mathbf{x} * \mathbf{y} = \mathbf{z}$$

A model vector would in this case have two input variables (\mathbf{x} and \mathbf{y}) and one output (\mathbf{z}). This means three variables in total.

We want to train the network on 20 model vectors. An example is below:

x	y	z
1	1	1
1	2	2
2	3	6
...		
4	5	20

The neural network size for this example would be:

20 model vectors,
3 variables

Add Some Rows or Columns Dialog

Enter a number of new columns (variables) or new rows (model vectors) which will be added (appended) to the base.

All new cells will be declared as missing values -- the cells will be empty.
The new variables will have EXCLUDE status turned on.

Input value is out of range

This is only a warning. The aiNet keeps track of the minimum and maximum for each variable. If you have entered a value, which is not in the minimum-maximum interval, aiNet asks you to confirm your entry.

This *safety* is somehow annoying in the first few entries, but later prevents you from entering accidental values.

Variable Description Dialog

This dialog enables you to change some attributes of selected variable.

Name:

Enter a new name for the variable. Do not make it too long. The new name will appear on the screen when the dialog is closed with the OK button.

Status:

Three different status types are available: input, output and exclude.

Input Select input status, when you want that this variable will be an input variable.

Output Select output, when you want that this variable will be an output variable.

Exclude Select exclude status, when you want to temporarily exclude this variable from the model vectors and prediction vectors. The aiNet then behaves as if this variable does not exist.

Discrete:

Check discrete option to specify that this variable will have discrete values only. Discrete values can be any real values, however integers are preferred. In other words, discrete type enables clustering data.

An Example ...:

Range:

Shows current minimal and maximal value of the variable.

Number of decimals:

Enables you to specify a number of decimals you want to see on the screen.

Alignment:

Allows you to select an alignment type.

Cell width:

Here you enter the number of characters (approximately) you want to be displayed on the screen

Global RMS (root mean square) Error Dialog

The Global RMS dialog box displays all global errors which aiNet calculates.

You see:

- * total RMS errors and
- * RMS errors on variables

To see an error of a specific output variable, select desired variable in the output variable list box. As you change the selection in the list box, numbers on the right change accordingly.

Registration Dialog

In this dialog you enter the registration text and code. By doing that you explicitly register the aiNet application to you or your company.

How you get the text & code. Fill in the registration order and send it to us. On the registration order you specify the exact text, you want to appear in aiNet's caption bar. When we get your order, we will calculate the registration code on the basis of the text you send to us. Then both (your text and registration code) will be send back to you.

If you have already received the registration text and code, than enter the text and code exactly as it was send back to you. If you have entered everything correctly, you can close the dialog with the OK button. The aiNet will thank you for your registration and all nagging messages will disappear.

[More about registration ...](#)

Total RMS error

The total error is used as a measure of an error in the model of the whole phenomena (problem) including all output variables. See Part 2, Chapter 3 in the User's Manual for more detail.

RMS Error on Variables

The variable RMS error is used as a measure of an error in the selected output variable. See Part 2, Chapter 3 in the User's Manual for more detail.

Discrete Option Example

Imagine, that we have an input variable, which describes printer types: dot-matrix, laser, ink-jet, daisy-wheel, ... We want that different printer types will be handled separately, which means that data associated with selected printer type, will not interfere with the data of other printer types.

Now we want to do some predictions. In the input part of the prediction vector is also the discrete printer type variable and we specify the ink-jet printer for this variable. As prediction runs, all model vectors, which have discrete variable not equal to ink-jet, are ignored. Thus, the result is calculated only on the basis of the ink-jet printer data.

IMPORTANT NOTE: DISCRETE OPTION ONLY WORKS WITH THE INPUT STATUS TURNED ON. IF YOU TURN ON THE OUTPUT STATUS, THAN DISCRETE OPTION WILL BE IGNORED.

Possible future improvements:

Now we are testing discrete option, which will enable the user to select the rate of the interference of the discrete variables. (Currently, this rate is set to zero - no interference at all.)

Copy Data to the Clipboard Dialog

As you have probably noticed, the aiNet application is not able to produce any printed output. If you want to have your results on paper, then copying the result to the Clipboard is the easiest solution. If your current view is Error Distribution View or Charts View then the copy command invokes a dialog box which lets you specify some options.

Select Output Variable:

You must select an output variable. The results of selected variable will be copied to the Clipboard.

Copy Contents:

You must also select what should be copied:

filtration will include the filtration results in the copy.

verification will include verification results in the copy.

(At least one option must be checked.)

Copy Mode:

As soon as you press one of the three buttons, the data will be send to the Clipboard. You can get your copy in three possible modes. Three buttons indicate modes:



This button will put results of selected output variable in the following table form:

Var Name (Quasi Correlation Data)		
Model	Filtration	Verification
true1	filt1	ver1
true X	filt X	ver X
true nmv	filt nmv	ver nmv

Here *true X* means the value of selected output variable taken from the model in X-th model vector, *filt X* means the filtration result of selected output variable taken from the model in X-th model vector and *ver X* stands for verification result.



This button will put calculated errors of the selected variable in the following table form:

Var Name (Error Data)		
MV Index	Filt. Error	Ver. Error
1	filt.err 1	ver.err 1
X	filt.err X	ver.err X
nmv	filt.err nmv	ver.err nmv

Here X stands for model vector index, *filt.err X* means the error of filtration result of selected output variable taken from the model in X-th model vector and *ver.err X* stands for the error of verification result.



This button will put results of filtration and verification in the following table form:

Var Name (Prediction Data)			
MV Index	Model	Filtration	Verification
1	true 1	filt 1	ver 1
X	true X	filt X	ver X
nmv	true nmv	filt nmv	ver nmv

3D - Surface Chart Dialog

This dialog generates prediction vectors necessary to create 3D - surface charts.



By selecting the **Run&Copy** button prediction vectors are generated, prediction is evaluated and results are copied into the clipboard.

You will use this command if you would like to see the 3D - surface chart. (You must use a spreadsheet application to actually see the chart.)

In the first place, however, you must select two distinct input variables. For each selected input variable you must specify the range and number of steps the variable should made within the range. **If there are more than two input variables in the model, than other (not selected) variables get their values from the first row in prediction view.** See the example below.

Is you press the **Run&Copy** button you may also select output variables. If you do not select any output variable than the aiNet application will select the first output variable automatically.

Example:

Let us say we have four input variables denoted as *Inp1*, *Inp2*, ..., *Inp4* and two output variables denoted as *Out1*, *Out2*. Let us assume that the first row in prediction view looks something like

<i>Inp1</i>	<i>Inp2</i>	<i>Inp3</i>	<i>Inp4</i>	<i>Out1</i>	<i>Out2</i>
2.1	3.2	1.6	5.1	2	2

Let us select the *Inp3* variable as the first and *Inp4* as the second input variable. Let the range of *Inp3* be from 1 to 9 in 4 steps and let the range of *Inp4* be from 3 to 6 in 3 steps. Then aiNet generates the following 20 predict vectors: $(4+1)*(3+1)=20$

<i>Inp1</i>	<i>Inp2</i>	<i>Inp3</i>	<i>Inp4</i>	<i>Out1</i>	<i>Out2</i>
2.1	3.2	1	3	?	?
2.1	3.2	3	3	?	?
2.1	3.2	5	3	?	?
2.1	3.2	7	3	?	?
2.1	3.2	9	3	?	?
2.1	3.2	1	4	?	?
2.1	3.2	3	4	?	?
2.1	3.2	5	4	?	?
2.1	3.2	7	4	?	?
2.1	3.2	9	4	?	?
2.1	3.2	1	5	?	?
2.1	3.2	3	5	?	?
2.1	3.2	5	5	?	?
2.1	3.2	7	5	?	?
2.1	3.2	9	5	?	?
2.1	3.2	1	6	?	?
2.1	3.2	3	6	?	?
2.1	3.2	5	6	?	?
2.1	3.2	7	6	?	?
2.1	3.2	9	6	?	?

Note, how all generated prediction vectors have the same values for *Inp1* and *Inp2* variables and how variables *Inp3* and *Inp4* vary according to specified ranges and steps.

Drawing the Surface Chart

If you select the **Run&Copy** button, than the aiNet application will run the prediction process and copy the results of calculation into the clipboard. Now you must switch to a spreadsheet application. Select a cell in the spreadsheet and select the paste command. The prediction results will be pasted in a form suitable for surface chart creation. Please, see your spreadsheet manual for further details.

On-Line Registration Dialog

This dialog enables you to make a safe registration using email. Namely, an information from the registration form will be encrypted and passed to your email application. Since we developed our own encryption algorithm, only we know how to decode your mail.

You can use Safe-On-Line registration if your computer can send an email. If your computer does not have an access to email, than you have to use conventional mail registration. See Registration Form.

Before you can submit the registration, you have to fill out the registration form. The form is divided into four logical topics: Personal Info, Registration Info, Credit Card Info and Notes.

Personal Info

Only basic user information is required:

Name: Your name.
Address: This is the address, which will be used to send any mail to you.
Email: Your email address, which will be used to send emails to you.

Registration Info

Registration name preferred: The name, which will appear on the caption, after you register it.
Registration fee: Application reg. fee (\$49 US) is always required. The DLL reg. fee (\$69 US) is required only if you intend to use it in commercial purposes. There is no fee for DLL if you will use it for non-commercial purposes. Note: If you pay commercial DLL fee, you also get the source code of DLL.
Disk Formats: Check the disk type that will be send to you.
Shipping & Handling: Fill only if you want to get the application on disks.
Total: Is the sum you have to pay.

Credit Card Info

Credit Card Type: Select the credit card you want to pay with.
Credit Card Number: Type the credit card number. Do not worry - the email will be encrypted.
Expiration Date: Type the credit card expiration date - month / year. Use YY or YYYY format for the year.
Card Holder's Name & Add: Type the name of CC owner or simply check the check box if the CC owner is the same as user.

Notes

Type anything you like to say. You can also leave it blank.

Submit Command

This command is enabled only if MAPI is installed on you computer. MS Exchange, MS Outlook and also some other applications use MAPI system to send email messages. (Sometimes users install MAPI and they do not configure it correctly. In this case use the Copy command.) The Submit command will construct an encrypted email and pass it to you Internet mailer, automatically. The mailer will take care for the rest of the process. (Check the outgoing mail.)

Copy Command

If MAPI is not installed on your computer or you do not use it for sending emails, than you must construct an email manually. The copy command will encrypt the message and copy it to the Clipboard. Now you can switch to your favorite email application, point to the message field and paste the contents of the Clipboard. Additionally, you have to specify the email address the mail will be sent to. In this case you have to type ainet@siol.net AINET@SIOL.NET into the TO field.

Show Mail Command

This command shows you the data that will be sent to us. You can see the encrypted form and decoded form.

