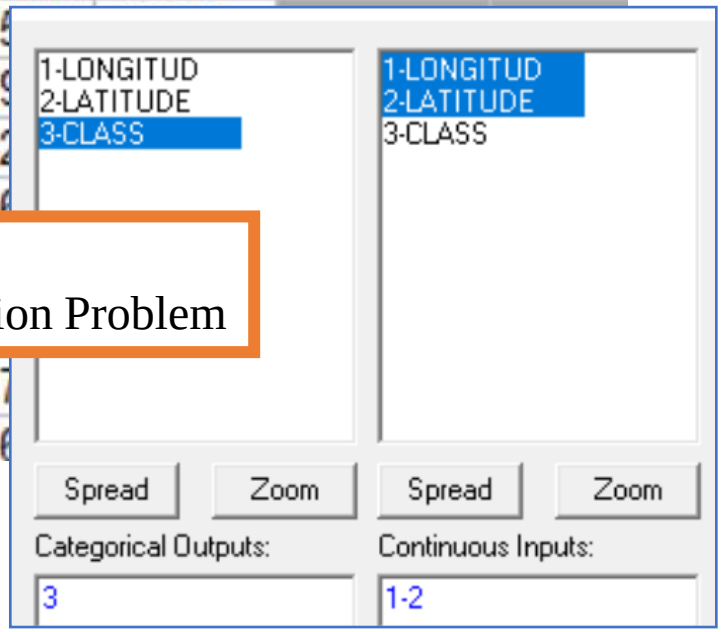
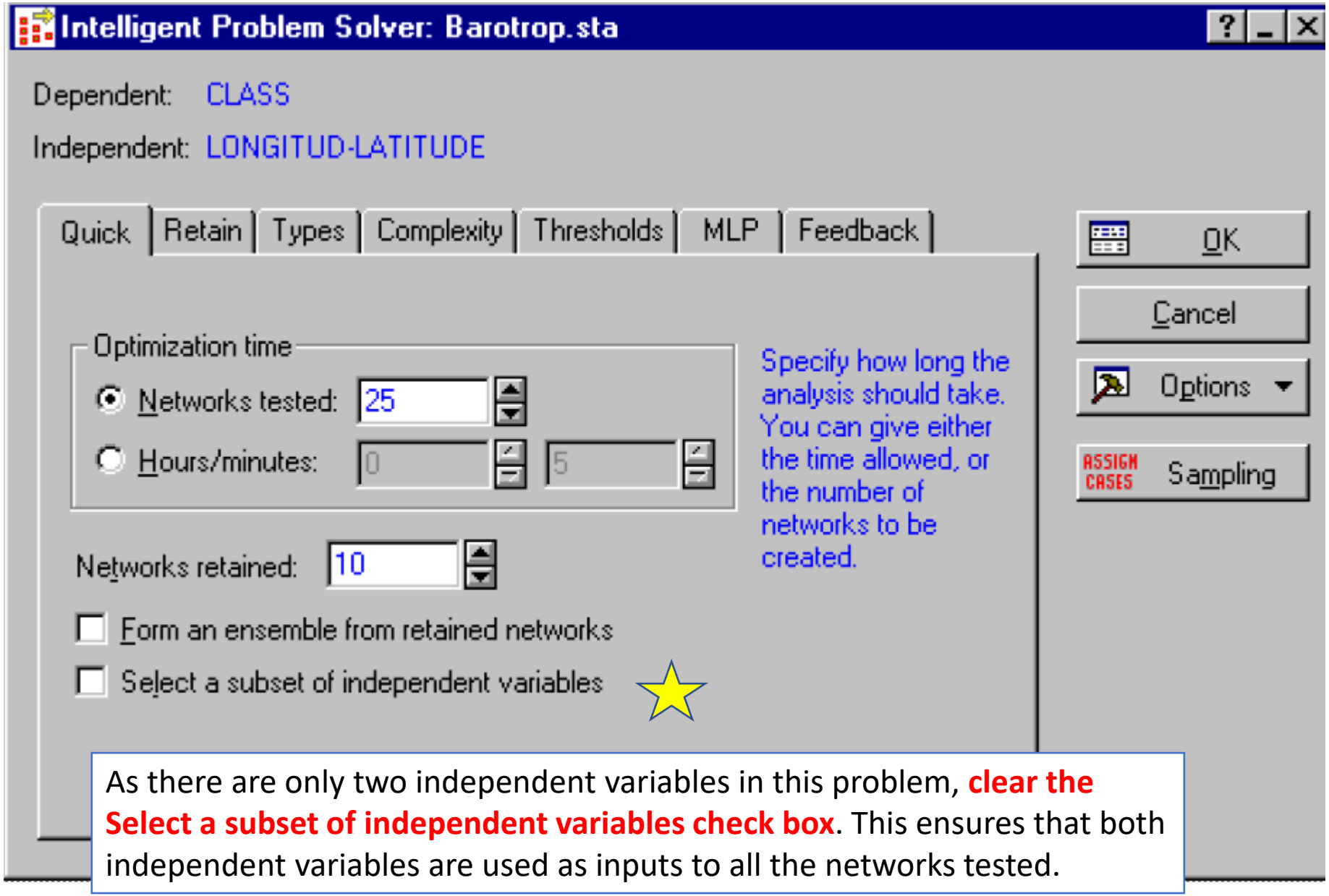
SNN Example 2: Creating a Neural Network

	Data from Elsner, Lehmiller, and Kimberlain (1996)		
	1 LONGITUD	2 LATITUDE	3 CLASS
1	59,00	17,00	BARO
2	59,50	21,00	BARO
3	60,00	12,00	BARO
4	60,50	16,00	BARO
5	61,00	13,00	BARO
6	61,00	15,00	BARO
7	61,50	17,00	BARO
8	61,50	19,00	BARO
9	62,00	14,00	BARO
10	63,00	15	
11	63,50	19	
12	64,00	12	
13	64,50	16	
14	65		
15	65		
16	65,00	17	
17	65,50	16	

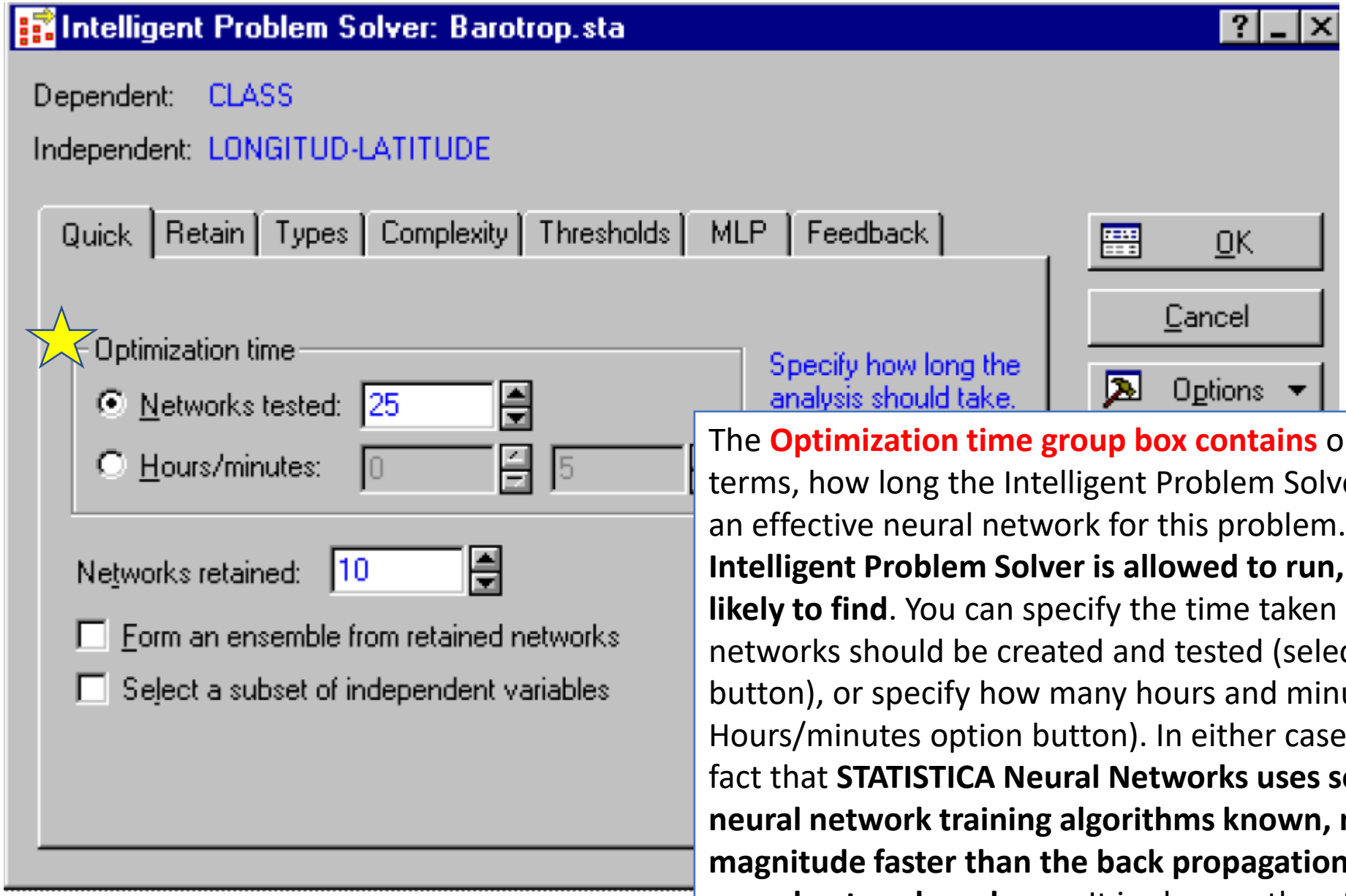


IPS:
Classification Problem

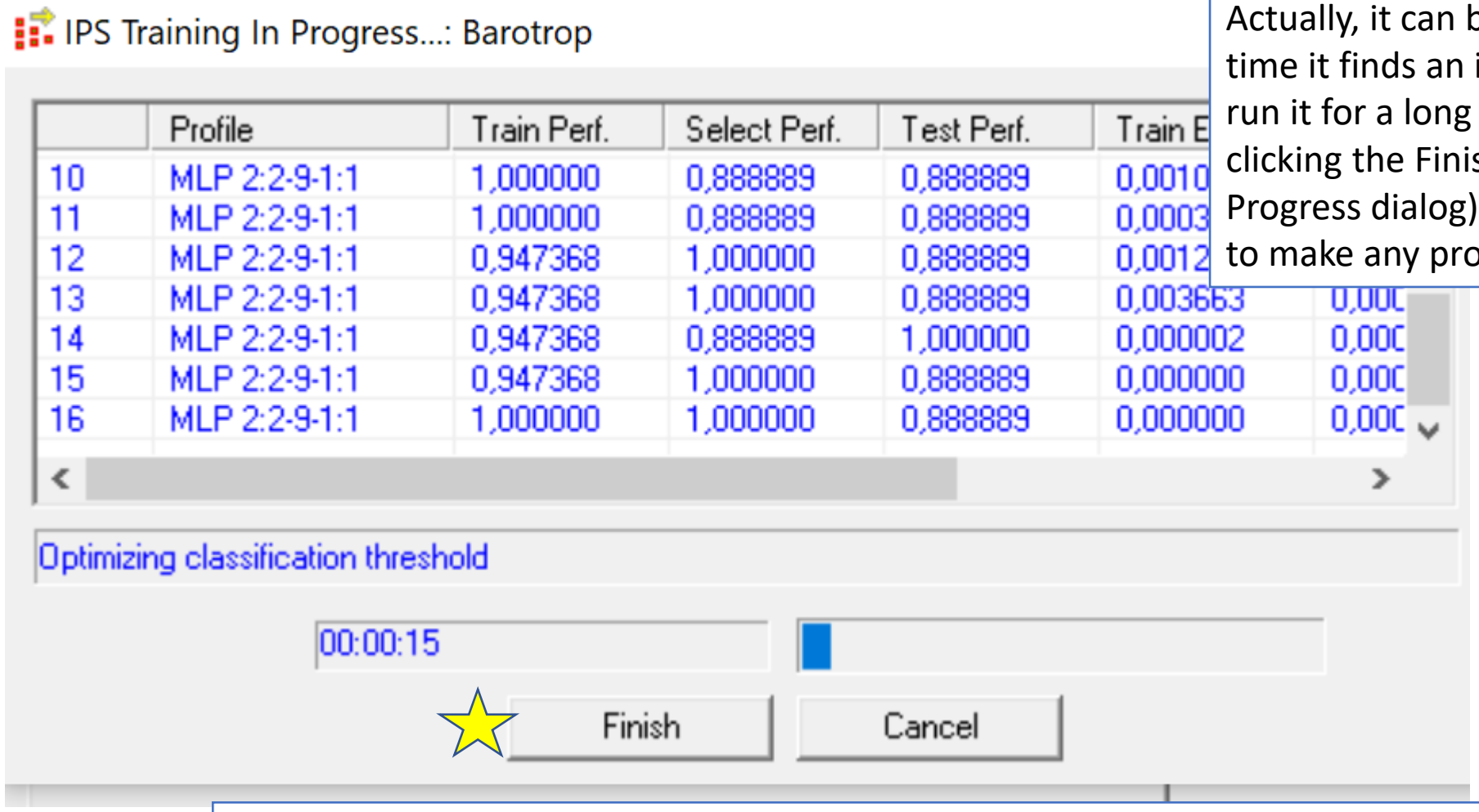
The file **Barotrop.sta** contains data on two classes of storms - *Barometric* and *Tropical*. This is a modified version of the data reported in Elsner, Lehmiller and Kimberlain (1996). The data recorded includes the type of storm and its longitude and latitude. We will attempt to predict the class of storm from its longitude and latitude. This is a simple nonlinear problem.



As there are only two independent variables in this problem, **clear the Select a subset of independent variables check box**. This ensures that both independent variables are used as inputs to all the networks tested.



The **Optimization time group box contains** options to specify, in broad terms, how long the Intelligent Problem Solver should spend trying to design an effective neural network for this problem. In general, **the longer the Intelligent Problem Solver is allowed to run, the better the solution it is likely to find**. You can specify the time taken either in terms of how many networks should be created and tested (select the Networks tested option button), or specify how many hours and minutes it should run (select the Hours/minutes option button). In either case, you can be confident in the fact that **STATISTICA Neural Networks uses some of the most advanced neural network training algorithms known, running up to two orders of magnitude faster than the back propagation algorithm employed by most neural network packages**. It is also worth noting that, although neural network training may take a long time, neural network execution (use) is extremely fast.



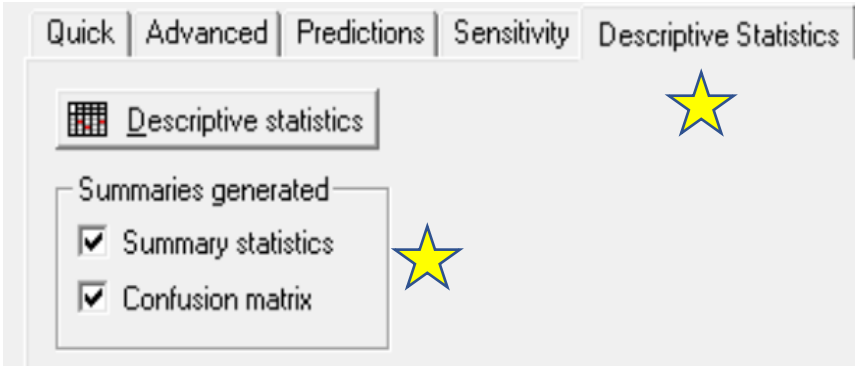
Actually, it can be configured to tell you every time it finds an improved solution, so you can run it for a long period and simply stop it (by clicking the Finish button on the IPS Training in Progress dialog) if you observe that it is failing to make any progress.

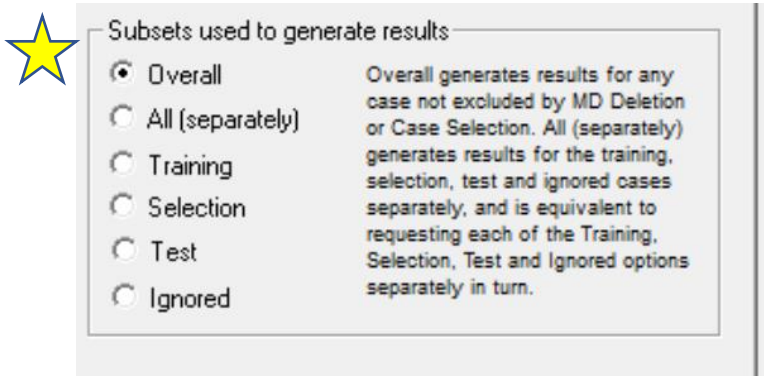
Each time an improved network is discovered, a new row is added to spreadsheet on this dialog, displaying summary details about the network. In addition, the time elapsed is displayed at the bottom of the window, as well as the percentage of the process that is complete. If you are conducting a long run and there has not been any progress for a considerable time, remember that you can click the Finish button on the Progress dialog to terminate the network testing early.

	Classification (1-10) (Barotrop)					
	CLASS.BARO.1	CLASS.TROP.1	CLASS.BARO.2	CLASS.TROP.2	CLASS.BARO.3	CLASS.TROP.3
Total	19,00000	18,00000	19,00000	18,00000	19,00000	18,00000
Correct	10,00000	9,00000	15,00000	15,00000	15,00000	15,00000
Wrong	9,00000	9,00000	4,00000	3,00000	4,00000	3,00000
Unknown	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
Correct(%)	52,63158	50,00000	78,94737	83,33333	78,94737	83,33333
Wrong(%)	47,36842	50,00000	21,05263	16,66667	21,05263	16,66667
Unknown(%)	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

The Classification spreadsheet presents overall summary details on the classification performance. There are columns for each output class as predicted by each model. For example, the column labeled CLASS.BARO.1 corresponds to the predictions of model 1 on the BARO class of the CLASS variable.

The first row says how many cases of each storm type there were in the data set. The second row records, for each class, how many are actually predicted correctly by the network, and the third how many are wrongly predicted. You are likely to see all, perhaps barring one or two, correctly predicted in this case. The fourth row records "unknown" cases, a point to which we will return in a later example using the Intelligent Problem Solver - there should be no such cases now.

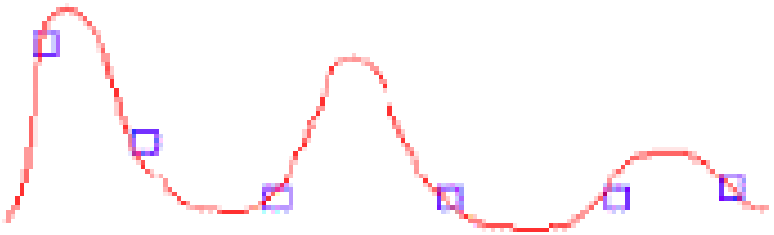




	Confusion Matrix - CLASS(1,2,3,4,5,6,7,8,9,10) (Barotrop)	
	BARO	TROP
BARO.1	9,00000	10,00000
TROP.1	10,00000	8,00000
BARO.2	14,00000	5,00000
TROP.2	5,00000	13,00000
BARO.3	18,00000	1,00000
TROP.3	1,00000	17,00000
BARO.4	19,00000	0,00000
TROP.4	0,00000	18,00000
BARO.5	19,00000	0,00000
TROP.5	0,00000	18,00000
BARO.6	19,00000	0,00000
TROP.6	0,00000	18,00000
BARO.7	19,00000	0,00000
TROP.7	0,00000	18,00000
BARO.8	19,00000	0,00000
TROP.8	0,00000	18,00000
BARO.9	19,00000	0,00000
TROP.9	0,00000	18,00000
BARO.10	19,00000	0,00000
TROP.10	0,00000	18,00000

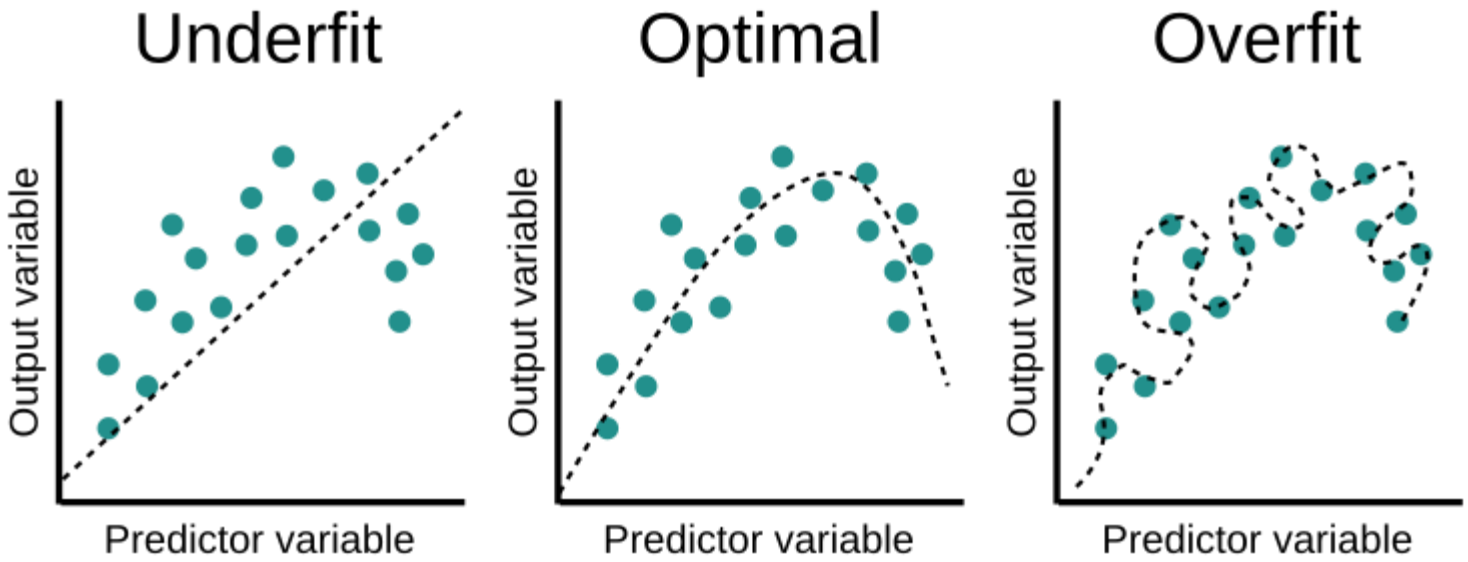
The Confusion Matrix spreadsheet gives a more detailed breakdown of misclassifications, and is particularly useful for problems with more than two output classes.

The summary details so far displayed are for all the cases in the data set. However, the cases are actually divided into a number of subsets, and it is important to consider the performance on particularly subsets.



An important limitation of neural networks (and related non-linear techniques) is that of **over-fitting**, or **over-learning**. Our objective when designing a neural network is to find a function that accurately models the unknown underlying function that relates the input variables to the output variables, and we estimate this function by fitting a function to the available data points (cases). The problem with fitting a curve to the data points is that if we choose a sufficiently eccentric function (that is, a function of high curvature) we can end up modeling the noise in the data rather than the underlying function. The figure below illustrates this problem when fitting a regression line in one dimension - a much smoother line would be better here, even if it did not pass through the data points, as it will predict results more accurately when new data is tested.

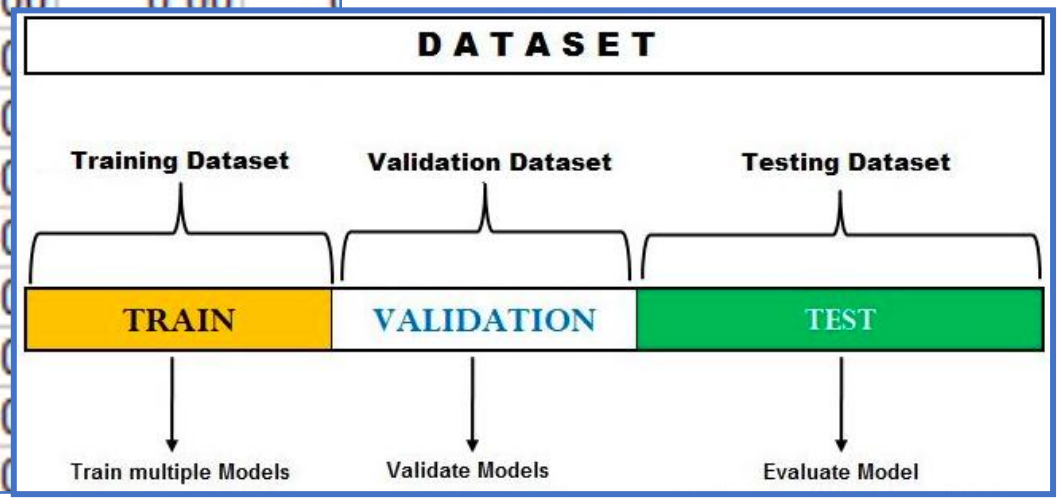
The ability to perform well on new data is called **generalization**, and is **the most desirable property of a neural network**. How then can we ensure that a network will generalize well? An important technique is to hold back some of the data, and not to use it for training the network. This data can be used to check the network's performance. This selection data is used in two ways. First, as network training progresses, the curvature of the function increases, and so we can stop training if performance on the selection set starts to deteriorate. This stops over-learning. Second, if we design a number of neural networks and want to select the best one, we cannot safely compare the performance on the training sets, as one network may have over-learned and have a deceptively good performance figure. We can, however, select the network with the best selection subset performance.



When the selection subset performance is used in this way, another problem may be introduced. If a large number of networks are tested, and the one with the best selection performance is selected, we are effectively conducting a sampling experiment, and we may end up with a network with an deceptively good selection subset performance that does not truthfully reflect its generalization ability, being biased by the sampling process. We may therefore reserve a further subset of data, the test set, which is used purely at the end of the design process to check that the selection error is not artificial. Providing that the selection and test errors are close together, we may be reasonably confident that the network will generalize successfully.

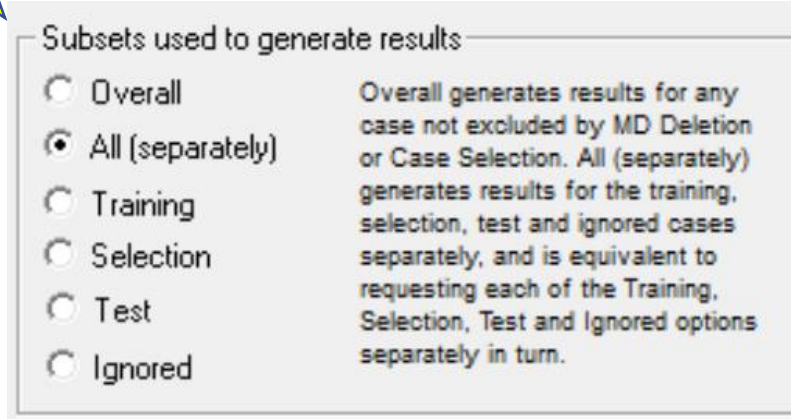
Confusion Matrix - CLASS(11,12,13,14,15,16,17,18,19,20) (Barotrop)

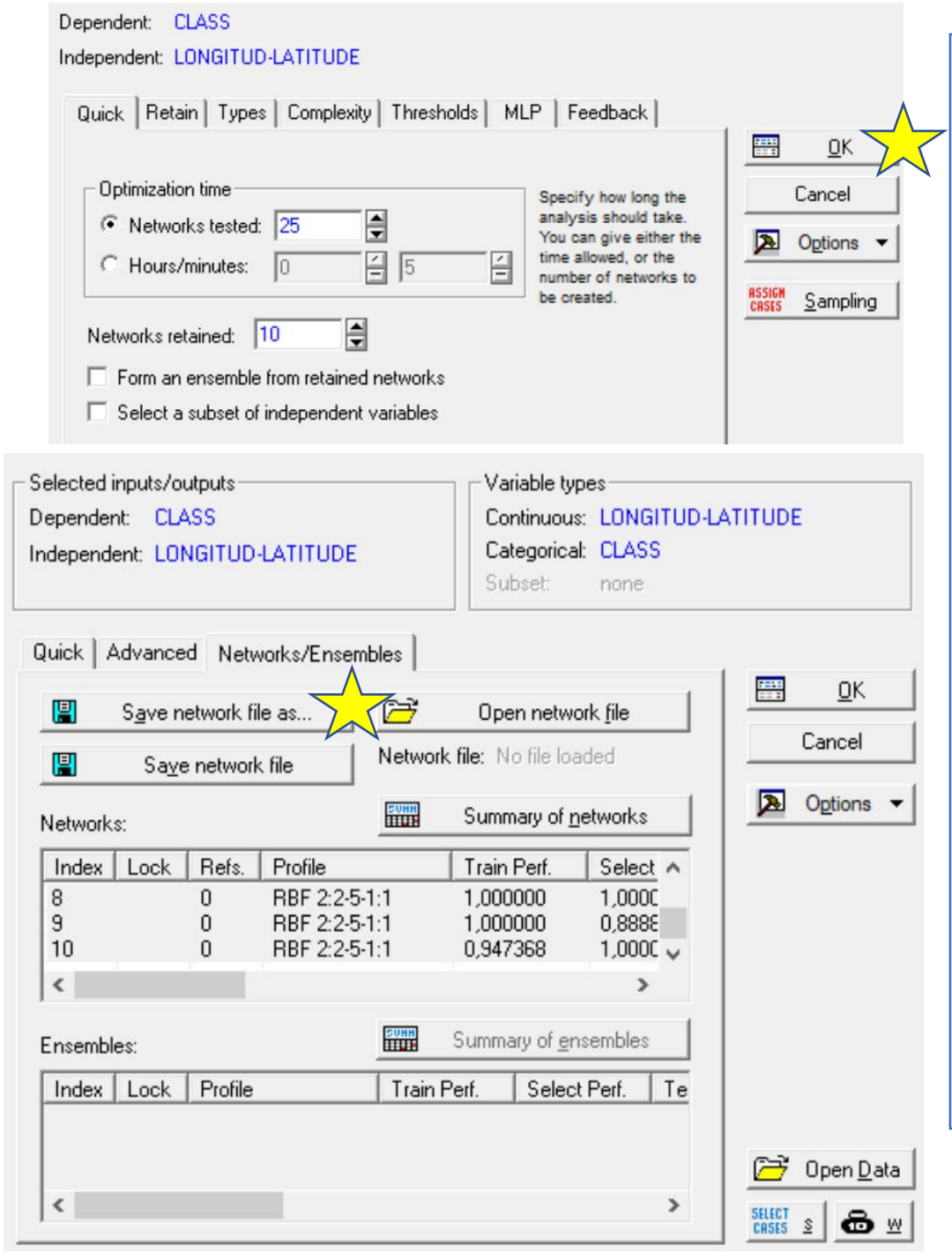
T.BARO	T.TROP	S.BARO	S.TROP	X.BARO	X.TROP	I.BARO	I.TR
4,000000	4,00000	5,000000	2,000000	2,000000	4,000000	0,00	
4,000000	7,00000	1,000000	1,000000	3,000000	0,000000	0,00	
3,000000	8,00000	5,000000	2,000000	1,000000	2,000000	0,00	
5,000000	3,00000	1,000000	1,000000	4,000000	2,0000		
6,000000	1,00000	5,000000	1,000000	4,000000	4,0000		
2,000000	10,00000	1,000000	2,000000	1,000000	0,0000		
5,000000	3,00000	4,000000	2,000000	3,000000	4,0000		
3,000000	8,00000	2,000000	1,000000	2,000000	0,0000		
6,000000	1,00000	5,000000	1,000000	5,000000	3,0000		
2,000000	10,00000	1,000000	2,000000	0,000000	1,0000		
6,000000	3,00000	6,000000	1,000000	4,000000	1,0000		



Validation ~ Selection

The resulting Classification spreadsheet is split into four sections. Columns now have titles with the prefixes T, S, X, and I corresponding to the training, selection, test, and ignored subsets respectively (we do not have any ignored cases, so the entries in the last section of the spreadsheet should be zeroes). By default, the cases are divided in the proportions 2:1:1 between the three subsets, and in this case you should have 50 training cases, 25 selection cases, and 25 test cases. You should also find that performance on the subsets is comparable (and perfect or near perfect) indicating that the neural network can indeed generalize well.





Once you are satisfied with the results generated by the *Intelligent Problem Solver*, you can complete the analysis.

Click the *OK* button on the *Results* dialog. This confirms that the analysis has been successfully completed, and commits the networks generated by the *Intelligent Problem Solver* to the network set, and returns you to the Startup Panel. If instead you stop the analysis by clicking the *Cancel* button, those networks are discarded. However, to prevent accidental lose of results you will be prompted to confirm that you want to cancel the current analysis.

Since neural networks take a considerable amount of time to train, and experimentation is generally needed to find the best performance, it is normal practice to keep copies of successful networks rather than recreating them each time the data is analyzed (which is the procedure often used for conventional statistical methods).

You can save the network set by selecting the *Networks/Ensembles* tab and clicking the *Save Network File As...* button to display a standard Save File dialog, which is used to save the networks on the current workspace to a new network file (file name extension *.snn*).