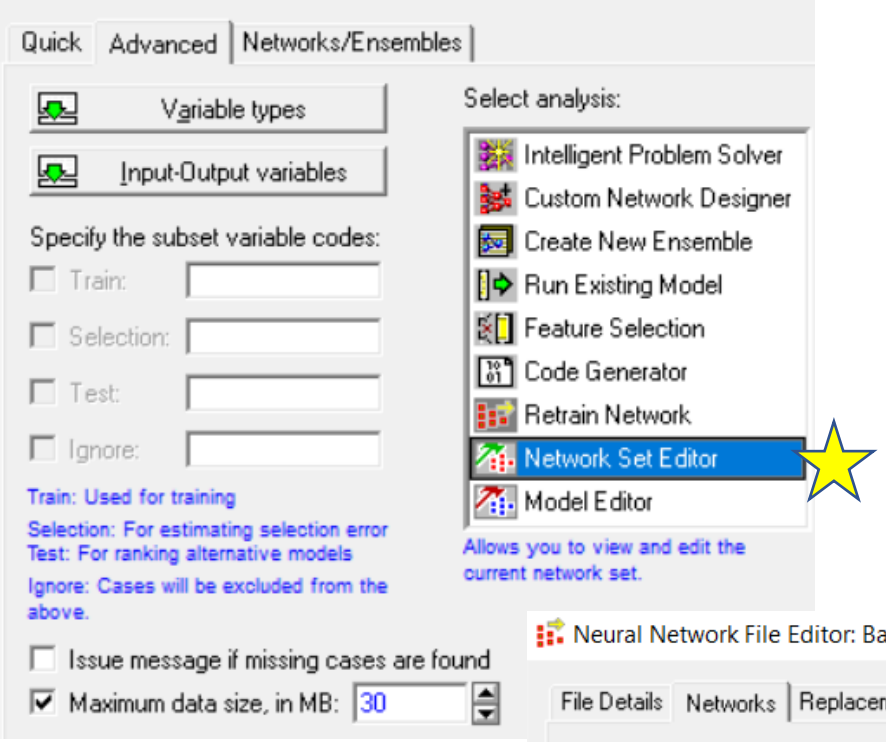
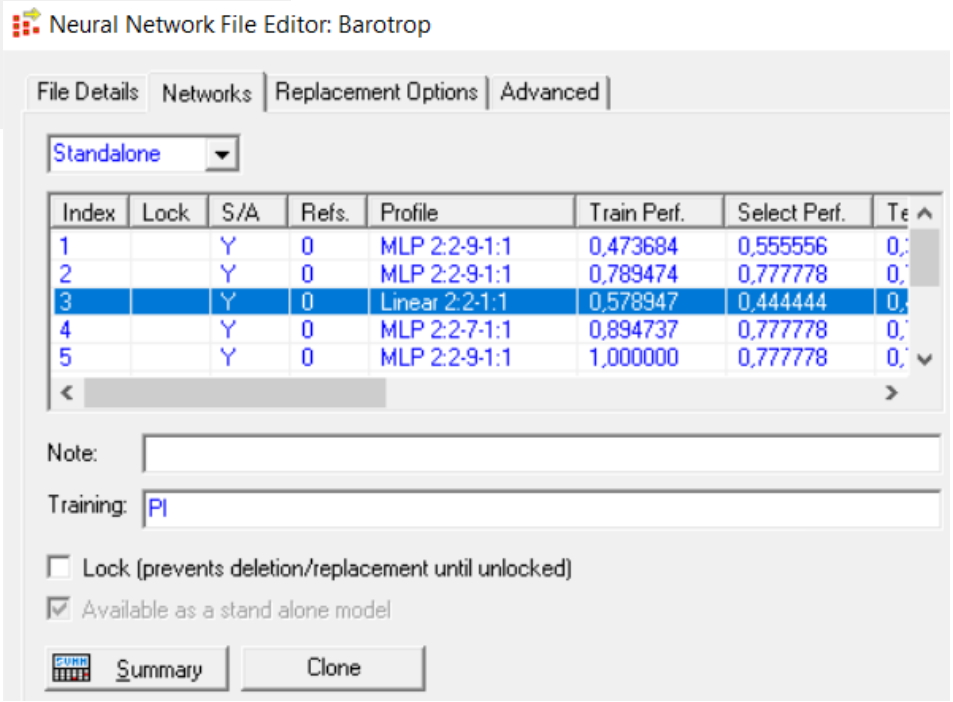
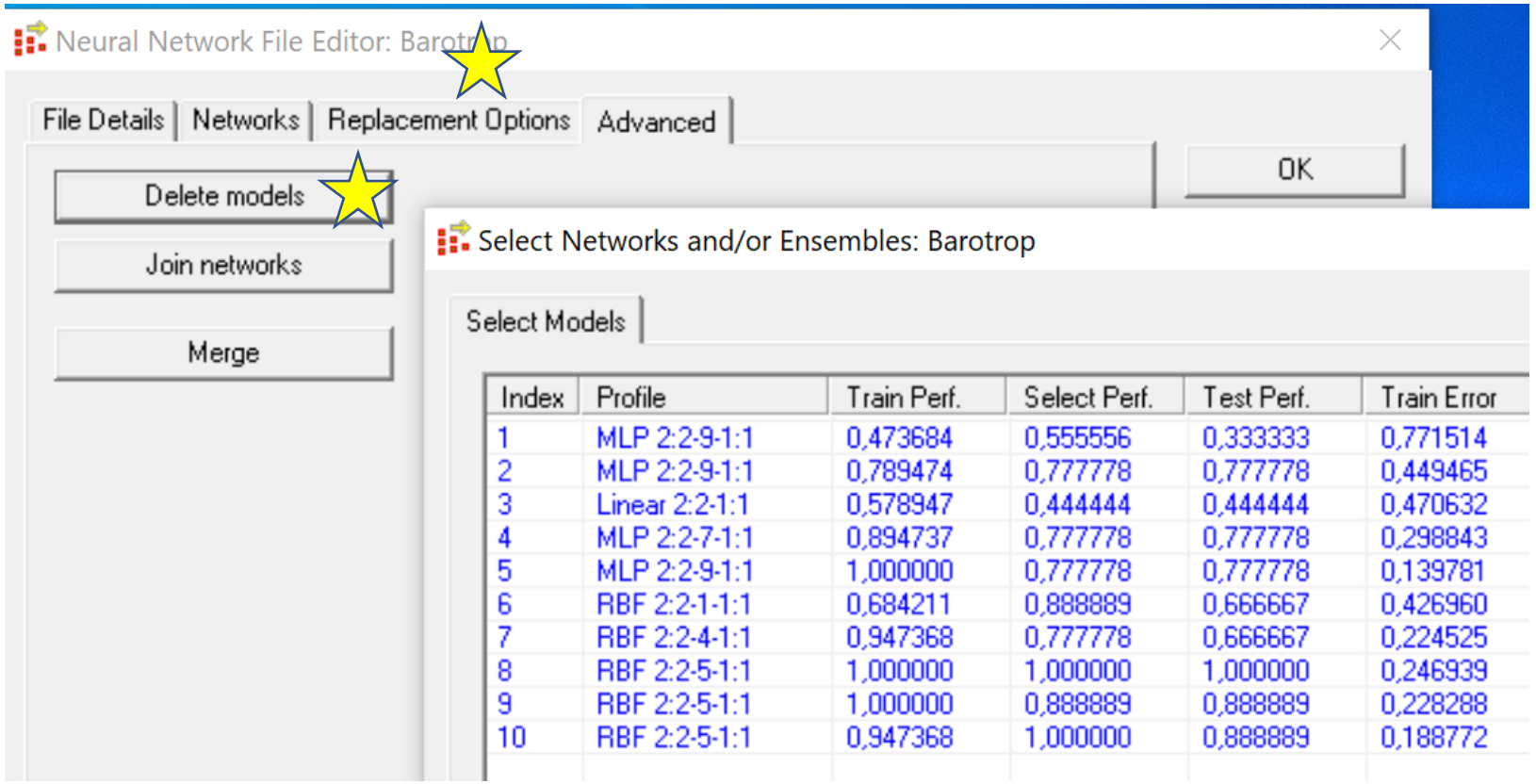




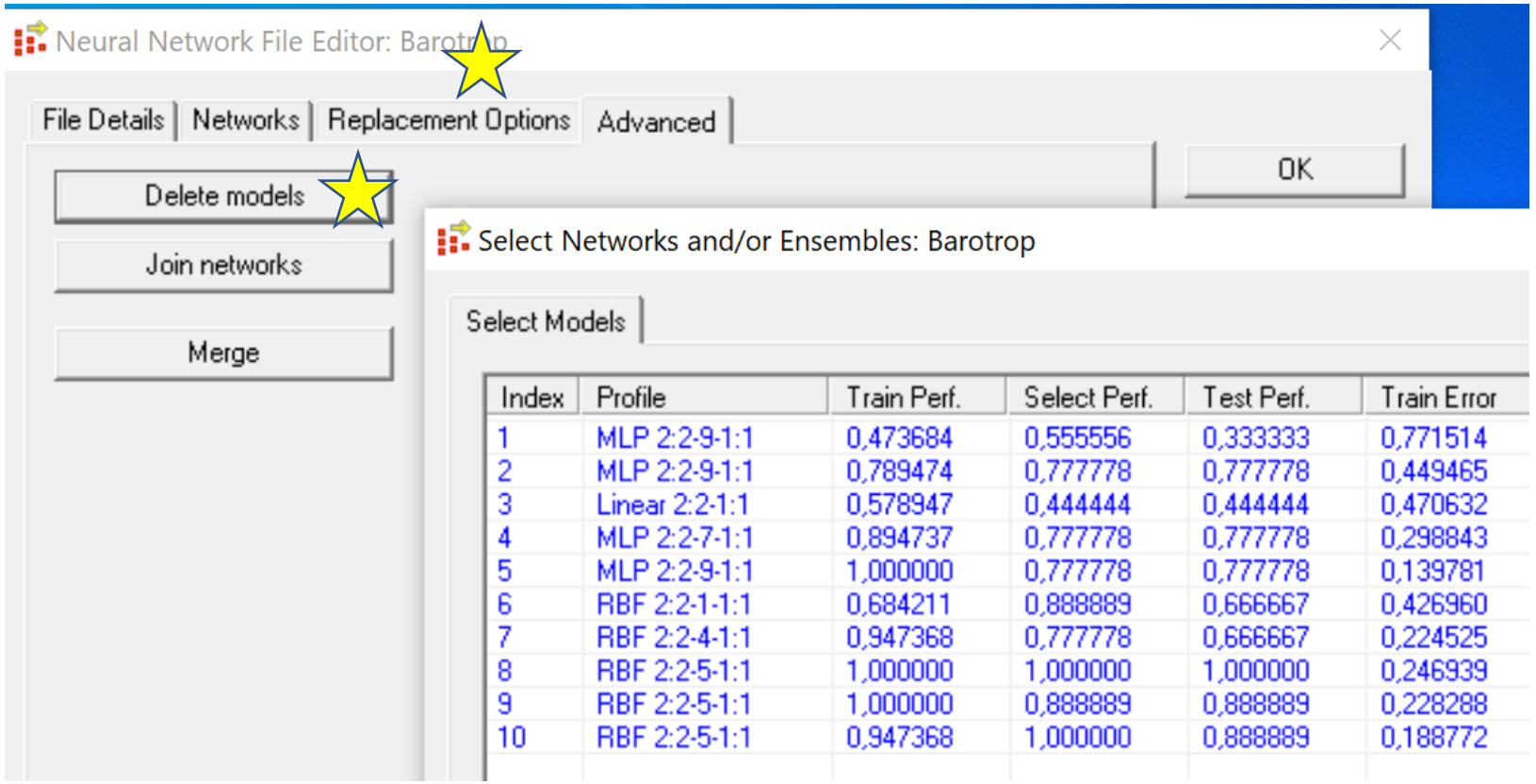
SNN Example 4: Using the Network File Editor

This dialog allows you to view, select between and organize the various neural networks and ensembles associated with a particular data set.



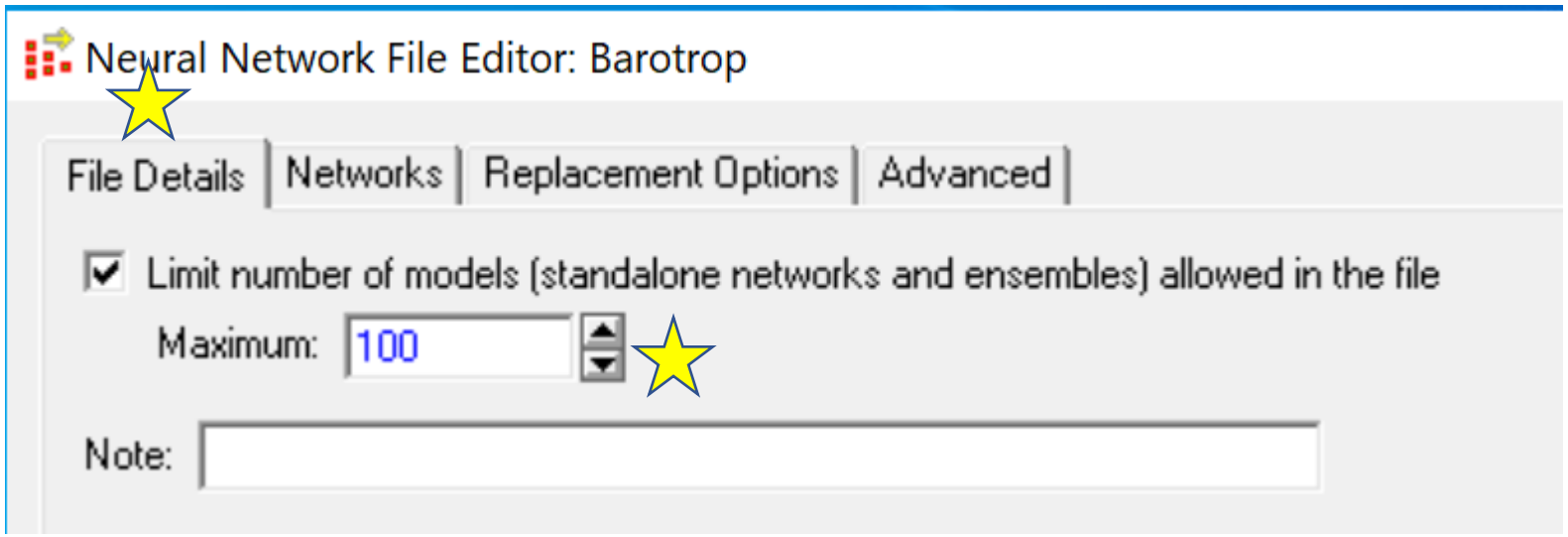
Index	Profile	Train Perf.	Select Perf.	Test Perf.	Train Error
1	MLP 2:2-9-1:1	0,473684	0,555556	0,333333	0,771514
2	MLP 2:2-9-1:1	0,789474	0,777778	0,777778	0,449465
3	Linear 2:2-1:1	0,578947	0,444444	0,444444	0,470632
4	MLP 2:2-7-1:1	0,894737	0,777778	0,777778	0,298843
5	MLP 2:2-9-1:1	1,000000	0,777778	0,777778	0,139781
6	RBF 2:2-1-1:1	0,684211	0,888889	0,666667	0,426960
7	RBF 2:2-4-1:1	0,947368	0,777778	0,666667	0,224525
8	RBF 2:2-5-1:1	1,000000	1,000000	1,000000	0,246939
9	RBF 2:2-5-1:1	1,000000	0,888889	0,888889	0,228288
10	RBF 2:2-5-1:1	0,947368	1,000000	0,888889	0,188772

Once you have studied the individual networks, you may want to delete some. For example, having established that the Linear networks are very poor in this problem domain, and that some of the smaller MLPs are also highly inferior, we may decide to remove them and not to experiment with those types again. It is easy to generate many hundreds of neural networks using *STATISTICA Neural Networks*, and the network file can rapidly become unmanageable. One alternative is to periodically purge unwanted networks.

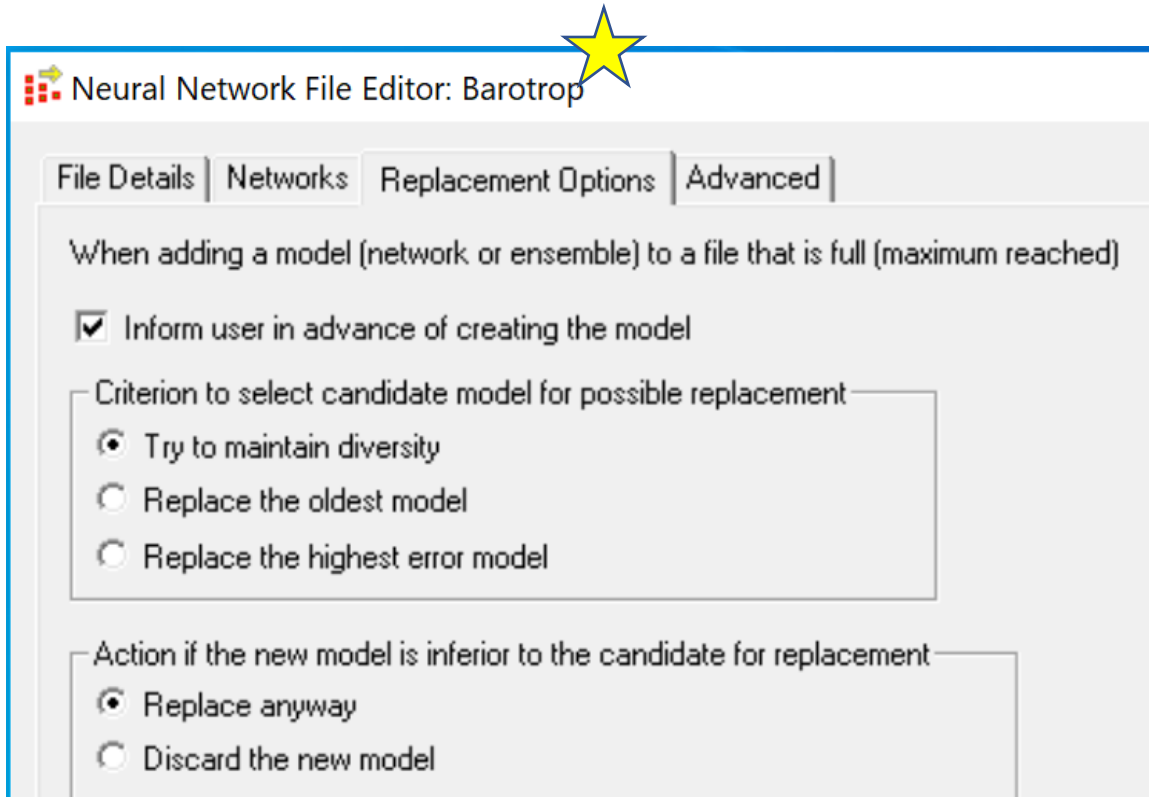


Index	Profile	Train Perf.	Select Perf.	Test Perf.	Train Error
1	MLP 2:2-9-1:1	0,473684	0,555556	0,333333	0,771514
2	MLP 2:2-9-1:1	0,789474	0,777778	0,777778	0,449465
3	Linear 2:2-1:1	0,578947	0,444444	0,444444	0,470632
4	MLP 2:2-7-1:1	0,894737	0,777778	0,777778	0,298843
5	MLP 2:2-9-1:1	1,000000	0,777778	0,777778	0,139781
6	RBF 2:2-1-1:1	0,684211	0,888889	0,666667	0,426960
7	RBF 2:2-4-1:1	0,947368	0,777778	0,666667	0,224525
8	RBF 2:2-5-1:1	1,000000	1,000000	1,000000	0,246939
9	RBF 2:2-5-1:1	1,000000	0,888889	0,888889	0,228288
10	RBF 2:2-5-1:1	0,947368	1,000000	0,888889	0,188772

For the storm data, you should observe that the performance of the Linear networks is extremely poor. This is not surprising, as we saw graphically that the two classes clearly are not linearly separable. Some of the networks are likely to have quite poor performance, and have been included in the network set as they have very few units, and demonstrate what happens if a very low complexity model is used. For example, you may find an MLP network with a single input and single hidden unit has been selected. These networks are included as we requested of the *Intelligent Problem Solver* that a diverse range of networks with different complexity versus performance trade-offs be included. However, there is likely to be at least one network with a performance of 1 (i.e. 100% correct classification). In this particular problem, RBF networks are particularly effective (they are good at modeling strongly local clusters, which this problem certainly has), and there may be several of these.



It is easy to generate many hundreds of neural networks using *STATISTICA Neural Networks*, and the network file can rapidly become unmanageable. One alternative is to periodically purge unwanted networks.



When your network file is full, *STATISTICA Neural Networks* has to decide what to do if you try to create more networks.
If the *Inform user in advance of creating model* check box is selected, and you try to create new networks when the file is already full, a warning dialog will be displayed to let you know that this might require discarding some models.
If you go ahead and create new models, *STATISTICA Neural Networks* enters a two-stage process. First, it identifies existing models that are candidates to be replaced by the newly created models. You can specify the criteria used to identify these candidates. Second, it determines whether the newly created models are actually better than the candidates. If they are, all is well and replacement goes ahead. If not, you can set options to specify either that the replacement takes place anyway, or that the new network is discarded.

The default under *Criterion to select candidate model for possible replacement* is *Try to maintain diversity*. This attempts to maintain a diverse range of network types and performance versus complexity trade-offs (the best performing network of each type is always kept, irrespective of its complexity). The default under *Action if the new network is inferior to the candidate for replacement* is *Replace anyway*.
With these settings, you should find an interesting range of different models is retained. There may be some "churning" among the less effective models, as new ones displace the worst performing, but at least newly-created models will be retained for testing