SNN Example 9: Case Subsets

We have already mentioned that the data set is divided into three subsets: the **training, selection, and test cases**.

To reiterate, the neural networks are trained using the training subset only. The selection subset is used to keep an independent check on the performance of the networks during training, with deterioration in the selection error indicating over-learning. If over-learning occurs, the *Intelligent Problem Solver (IPS)* stops training the network and restores it to the state with **minimum selection error**.

The selection error is also used by the *IPS* to select between the available networks. However, if a large number of networks is tested, a random sampling effect can kick in, and you may get a network with a good selection error that is not actually indicative of good generalization capabilities. Therefore, a third subset (the test subset) is maintained, and you can visually inspect performance after training. Providing that **selection and test errors are reasonably close together, the network is likely to generalize well**.

By default, the *Intelligent Problem Solver* randomly assigns the available cases in the proportions **2:1:1** between the training, selection, and test subsets. Each time the *IPS* is run, a different random assignment of cases is made, and then used for all the networks created by that run of the *IPS*.

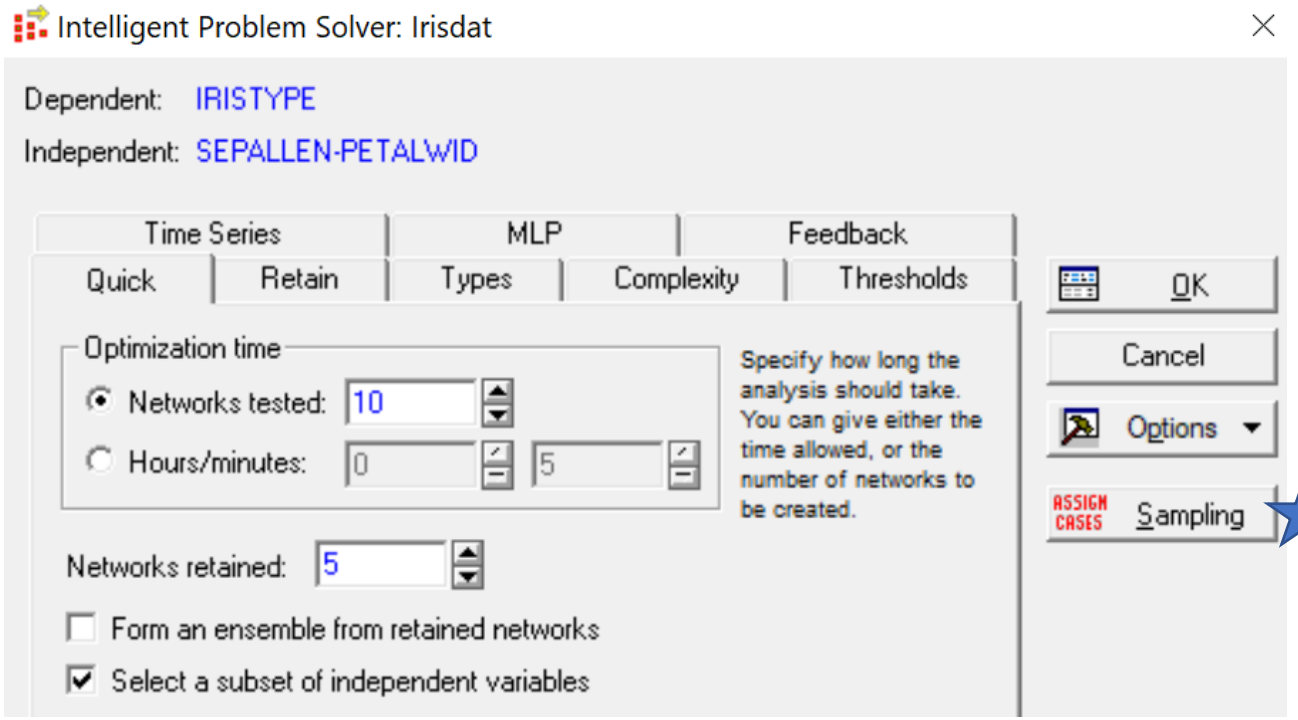
You can specify that the cases be reassigned randomly for each network created, or that the same case division be used as was used to train a preexisting network. You can also specify a subset variable (on the Advanced tab of the Startup Panel) that explicitly lays out which cases should be assigned to which subset. You may want to exercise these options for a number of reasons:

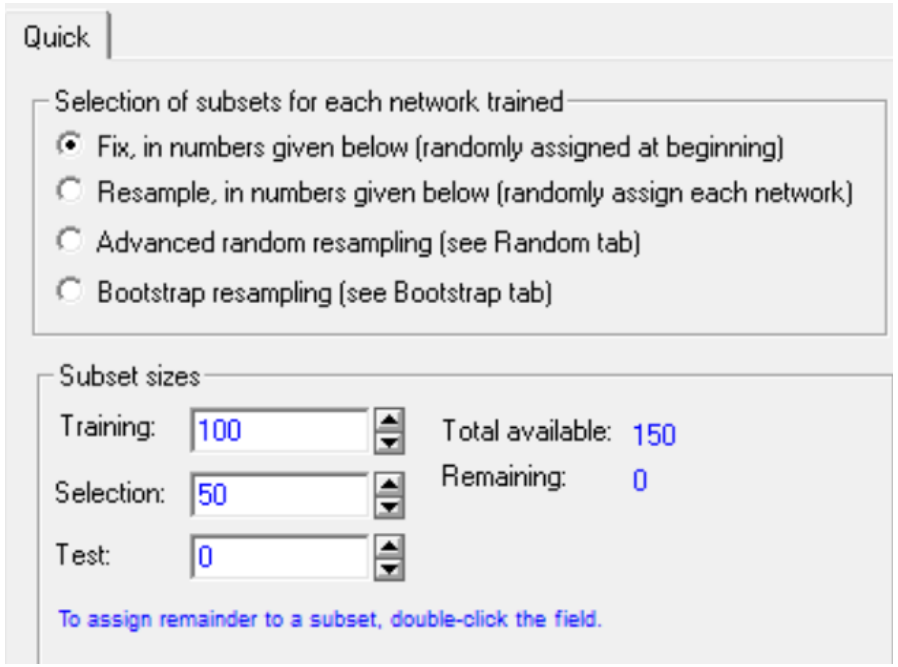
1. Reassigning cases randomly for each network can reduce sampling bias if you choose to form the networks created into an ensemble and average across their predictions.

2. Conversely, ensuring that several runs use the same subset division makes it easier to compare results. This is particularly helpful if you are trying to experimentally determine the best setting for some design parameter, such as the number of hidden units, rather than actually seeking a final solution.

3. If over a number of tests you find that the test and selection results are reasonably consistent, you can be fairly confident that the network will generalize effectively. In this case, you may decide to assign all the cases to training and selection, so as to produce a more accurate solution. You might also (although with a greater sense of caution) decide to reduce the number of selection cases, reassigning to the training subset, if the training and selection errors are consistent.

If you reassign the subsets, you should be careful in comparing the performance of networks in the network set trained with different subsets, as the performance figures are not directly comparable.





If you reassign the subsets, you should be careful in comparing the performance of networks in the network set trained with different subsets, as the performance figures are not directly comparable.

Change the number of *Training* cases to 100, and the number of *Selection* cases to 50. Then, double click on *Test* to reset that field to zero and to balance the remainder. *There are sufficiently few dubious cases in the Iris problem that the maintenance of a test set is not necessary*. Click *OK* to confirm the new case division.

Having established a good idea of the expected error rate, it is safe to run the *IPS* for a limited period without using a test set. However, as the networks habitually overfit the Iris data, you should not remove the selection subset.