**Subject:** **What are the population, sample, training set, design set, validation set, and test set?**

It is rarely useful to have a NN simply memorize a set of data, since memorization can be done much more efficiently by numerous algorithms for table look-up. Typically, you want the NN to be able to perform accurately on new data, that is, to [generalize.](ftp://ftp.sas.com/pub/neural/FAQ3.html#A_generalize)

There seems to be no term in the NN literature for the set of all cases that you want to be able to generalize to. Statisticians call this set the "population". Tsypkin (1971) called it the "grand truth distribution," but this term has never caught on.

Neither is there a consistent term in the NN literature for the set of cases that are available for training and evaluating an NN. Statisticians call this set the "sample". The sample is usually a subset of the population.

(Neurobiologists mean something entirely different by "population," apparently some collection of neurons, but I have never found out the exact meaning. I am going to continue to use "population" in the statistical sense until NN researchers reach a consensus on some other terms for "population" and "sample"; I suspect this will never happen.)

In NN methodology, the sample is often subdivided into "training", "validation", and "test" sets. The distinctions among these subsets are crucial, but the terms "validation" and "test" sets are often confused. Bishop (1995), an indispensable reference on neural networks, provides the following explanation (p. 372):

Since our goal is to find the network having the best performance on new data, the simplest approach to the comparison of different networks is to evaluate the [error function](ftp://ftp.sas.com/pub/neural/FAQ2.html#A_functions_error) using data which is independent of that used for training. Various networks are trained by minimization of an appropriate error function defined with respect to a *training* data set. The performance of the networks is then compared by evaluating the error function using an independent *validation* set, and the network having the smallest error with respect to the validation set is selected. This approach is called the *hold out method.* Since this procedure can itself lead to some [overfitting](ftp://ftp.sas.com/pub/neural/FAQ3.html#A_over) to the validation set, the performance of the selected network should be confirmed by measuring its performance on a third independent set of data called a *test* set.

And there is no book in the NN literature more authoritative than Ripley (1996), from which the following definitions are taken (p.354):

Training set:

A set of examples used for learning, that is to fit the parameters [i.e., weights] of the classifier.

Validation set:

A set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.

Test set:

A set of examples used only to assess the performance [generalization] of a fully-specified classifier.

The literature on machine learning often reverses the meaning of "validation" and "test" sets. This is the most blatant example of the terminological confusion that pervades artificial intelligence research.

The crucial point is that a test set, by the standard definition in the NN literature, is *never* used to choose among two or more networks, so that the error on the test set provides an unbiased estimate of the generalization error (assuming that the test set is representative of the population, etc.). Any data set that is used to choose the best of two or more networks is, by definition, a validation set, and the error of the chosen network on the validation set is optimistically biased.

There is a problem with the usual distinction between training and validation sets. Some training approaches, such as [early stopping,](ftp://ftp.sas.com/pub/neural/FAQ3.html#A_stop) require a validation set, so in a sense, the validation set is used for training. Other approaches, such as maximum likelihood, do not inherently require a validation set. So the "training" set for maximum likelihood might encompass both the "training" and "validation" sets for early stopping. Greg Heath has suggested the term "design" set be used for cases that are used solely to adjust the weights in a network, while "training" set be used to encompass both design and validation sets. There is considerable merit to this suggestion, but it has not yet been widely adopted.

But things can get more complicated. Suppose you want to train nets with 5 ,10, and 20 hidden units using maximum likelihood, and you want to train nets with 20 and 50 hidden units using early stopping. You also want to use a validation set to choose the best of these various networks. Should you use the same validation set for early stopping that you use for the final network choice, or should you use two separate validation sets? That is, you could divide the sample into 3 subsets, say A, B, C and proceed as follows:

* Do maximum likelihood using A.
* Do early stopping with A to adjust the weights and B to decide when to stop (this makes B a validation set).
* Choose among all 3 nets trained by maximum likelihood and the 2 nets trained by early stopping based on the error computed on B (the validation set).
* Estimate the generalization error of the chosen network using C (the test set).

Or you could divide the sample into 4 subsets, say A, B, C, and D and proceed as follows:

* Do maximum likelihood using A and B combined.
* Do early stopping with A to adjust the weights and B to decide when to stop (this makes B a validation set with respect to early stopping).
* Choose among all 3 nets trained by maximum likelihood and the 2 nets trained by early stopping based on the error computed on C (this makes C a second validation set).
* Estimate the generalization error of the chosen network using D (the test set).

Or, with the same 4 subsets, you could take a third approach:

* Do maximum likelihood using A.
* Choose among the 3 nets trained by maximum likelihood based on the error computed on B (the first validation set)
* Do early stopping with A to adjust the weights and B (the first validation set) to decide when to stop.
* Choose among the best net trained by maximum likelihood and the 2 nets trained by early stopping based on the error computed on C (the second validation set).
* Estimate the generalization error of the chosen network using D (the test set).

You could argue that the first approach is biased towards choosing a net trained by early stopping. Early stopping involves a choice among a potentially large number of networks, and therefore provides more opportunity for overfitting the validation set than does the choice among only 3 networks trained by maximum likelihood. Hence if you make the final choice of networks using the same validation set (B) that was used for early stopping, you give an unfair advantage to early stopping. If you are writing an article to compare various training methods, this bias could be a serious flaw. But if you are using NNs for some practical application, this bias might not matter at all, since you obtain an honest estimate of generalization error using C.

You could also argue that the second and third approaches are too wasteful in their use of data. This objection could be important if your sample contains 100 cases, but will probably be of little concern if your sample contains 100,000,000 cases. For small samples, there are other methods that make more efficient use of data; see ["What are cross-validation and bootstrapping?"](ftp://ftp.sas.com/pub/neural/FAQ3.html#A_cross)