Feed artificial intelligence component with experimental data

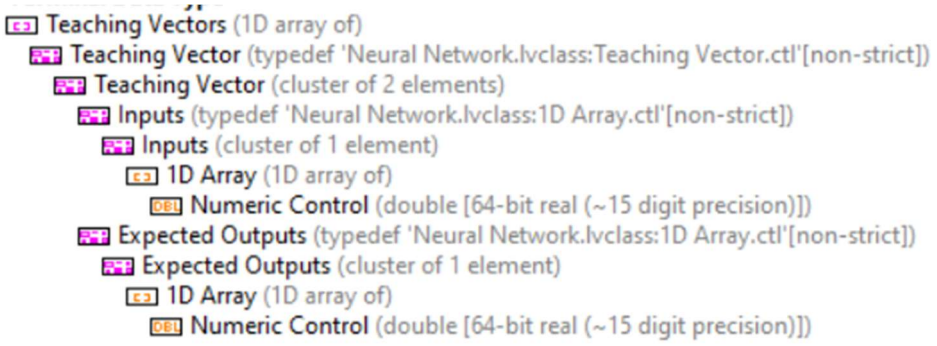
Work objective

Learning how to teach a neural network and afterwards obtain answers from it.

Organizational topics

For teaching a neural network, you have to provide it a series of data sets (vectors).
The vectors are grouped into an array.

Each vector must contain the process parameters (inputs) and the measured or processed values (outputs) from one experiment in certain conditions.
Both inputs and outputs from one vector are grouped into separate arrays.



The neural network will have to be fed with two separate series of data sets:

- a training series, containing around 70% of the available data;
- a validation series, containing around 30% of the available data.

It is advisable to start by building separate CSV files, one for training and one for validation, having a data set on each row.

For reducing the computing time, it is advisable to use not the real values of the outputs, but to group the output values in classes (e.g. like in a histogram) and to replace them with the index of the class to which each of it belongs.

The steps you will have to perform are:

- build the specific training and validation **bin** files;
- train the neural network with different configurations and select the optimal one;
- feed the selected network with new input data for obtaining output estimations.

Build the specific training and validation bin files

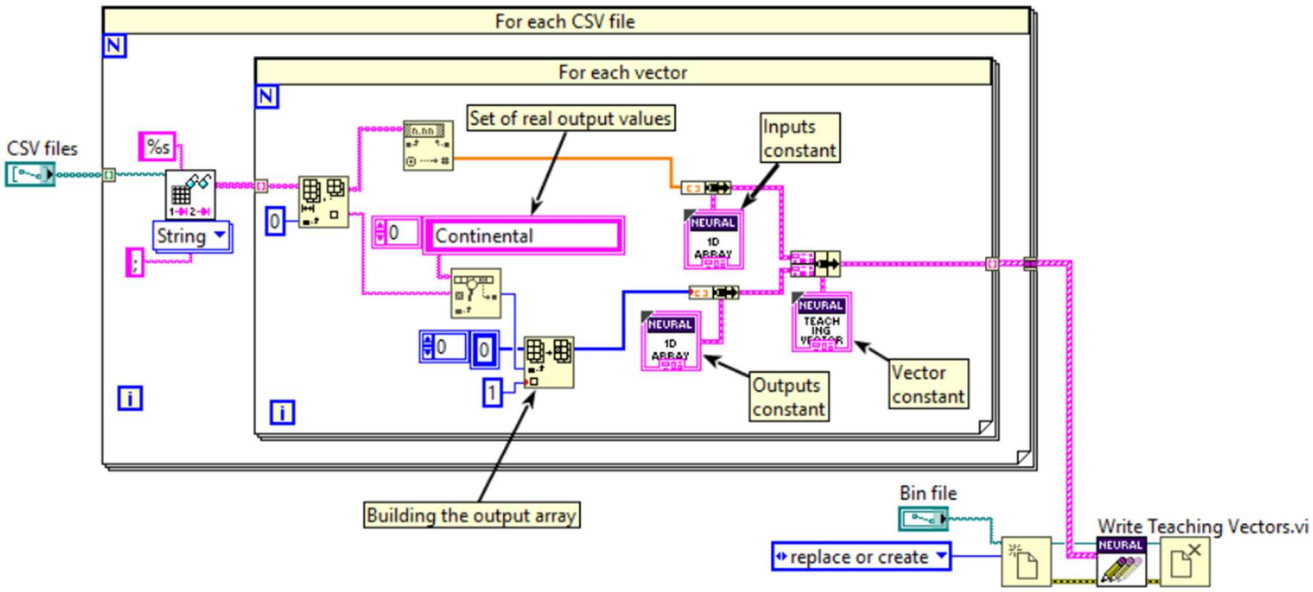
You can use an application structure similar with the one from the figure below.

Each line of the CSV file is separated in inputs and outputs, according to your specific number of variables.

Output values are replaced with class indexes, as previously recommended.

Both inputs and outputs are formatted according to the needed structures and concatenated into a vector.

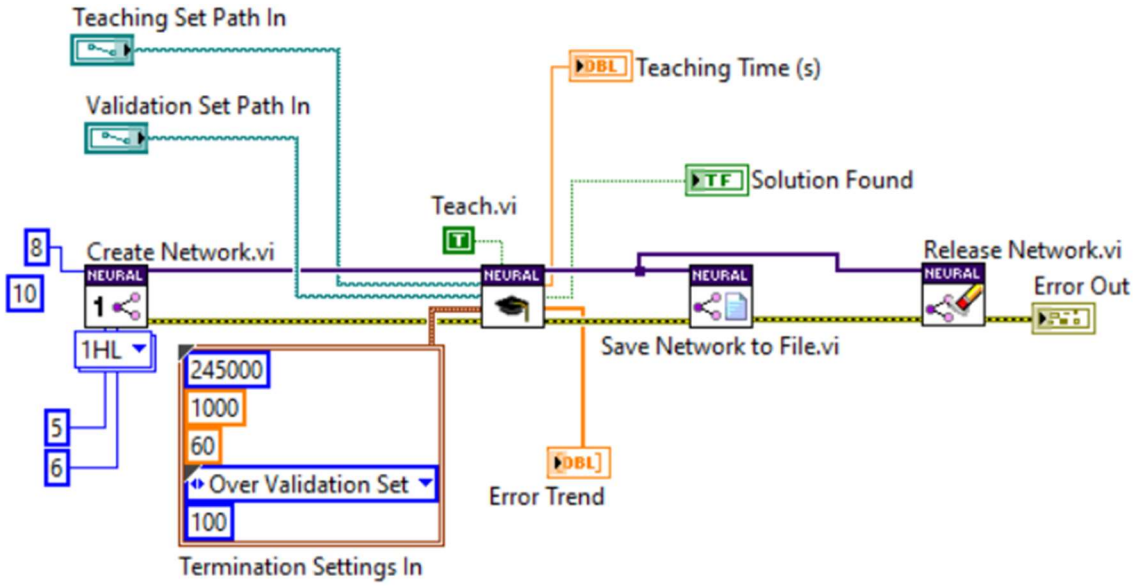
The vectors array is written into a separate **bin** file.

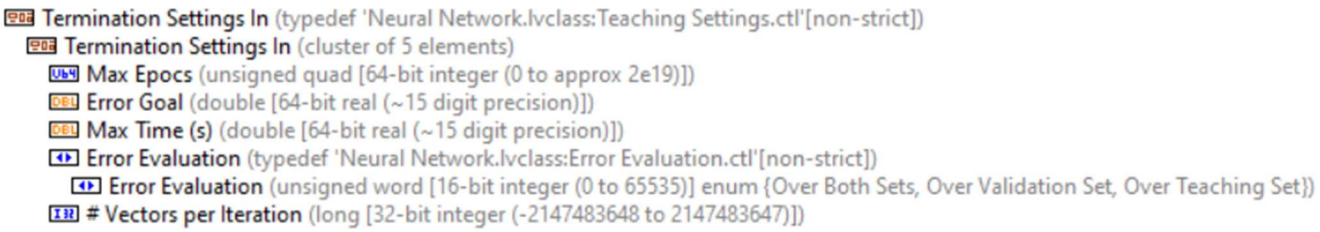


Train the neural network

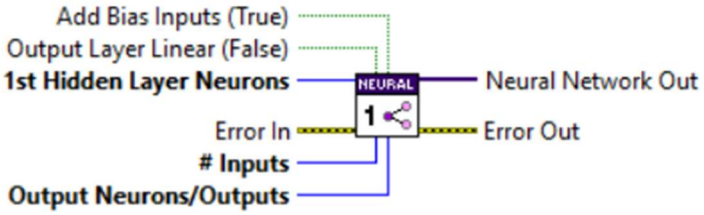
You can use an application structure similar with the one from the figure below.

The meanings of the values in the **Termination Settings** constant are detailed in the second figure below.





You can test different network configurations, changing the number of hidden layers and the numbers of neurons on each hidden layer.
For each configuration, you can change the number of epochs (iterations), the error limits or the running time.



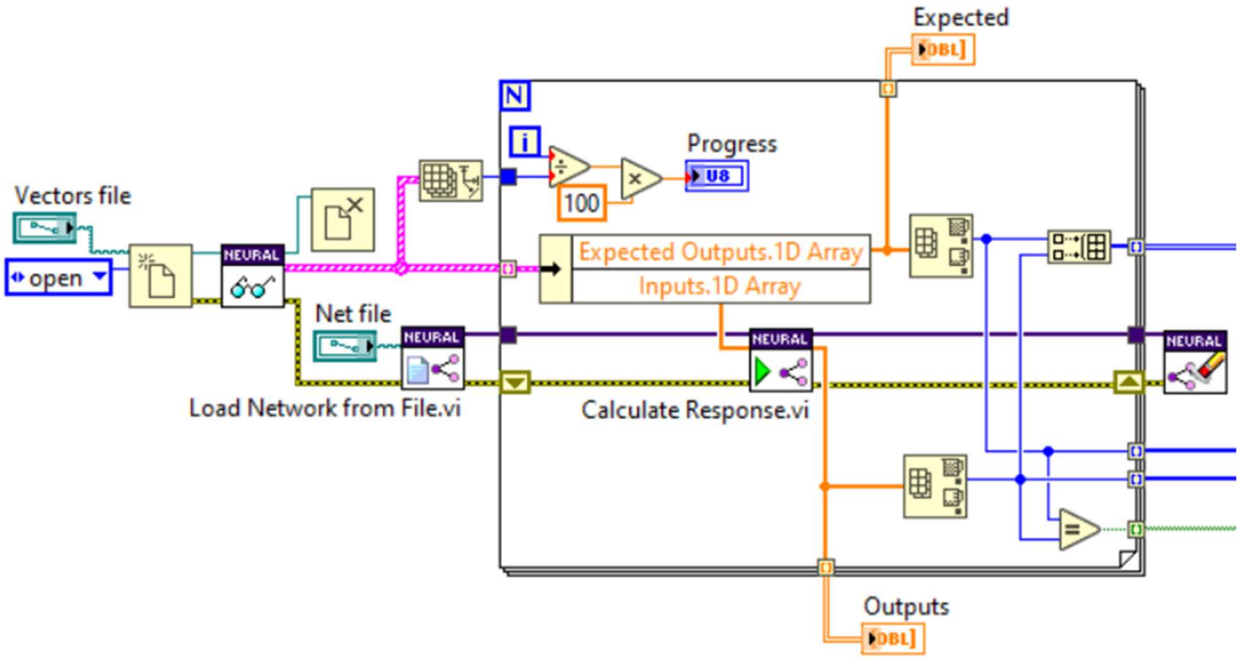
What is important is to find a network configuration which is finding a solution and is offering an as small as possible error.

Keep the file in which the chosen network was saved.

Feed the selected network

Now that you have a neural network stored in a file, you can feed it with inputs (process parameters) and the network will return an estimation of the measured or processed values.

You can use an application structure similar with the one from the figure below.



The application is a testing one, using a **Vectors file** in which also expected outputs are stored, for comparison reasons.

The application is loading the network from the file in which it was stored and then, for each data set, is calculating the response.

You can also initially use some data for which you know the expected answers and, after checking that the network is offering the correct answers, you can use it for estimating unknown data.