Neural Planner Tutorial

Contents

Introduction

The steps in the tutorial should be followed in the order they appear. They are arranged so that you can see what to expect from a neural network, then how to train an already created network, then how to create a new network that is capable of learning from the data in a training file and then how to create the training file itself.

The tutorial uses some of the neural networks that are included in the **examples** directory. These examples are described in the help system. You can browse through them using the help browser controls after selecting the examples topic on the help contents page.

Interrogating a trained network

Select **Open Network file** from the **File** menu

***Evaluation copies only**: The first time that you load a file a dialog will be produced showing the number of days that remain for you to evaluate Neural Planner. It also shows the basic registration details. Press **Continue** when you have read it.*

Select **diggers.nnp** from the examples directory and press **OK**

The **diggers.nnp** network file will be loaded and a dialog will appear showing some network information. Press **OK** to continue.

Select **Interrogate...** from the **Action** menu.

Load the interrogating file section from **diggers.tti** in the examples directory.

A dialog will appear with the controls that can be used to interrogate the neural network. Step through the interrogation data by pressing the **Next** button. Try the other controls if you want, they are described in detail in the help file. When you have finished press the **Cancel** button.

Training a network

Select **Open Network file** from the **File** menu

Select **logic.nnp** from the examples directory and press **OK**

The **logic.nnp** network file will be loaded and a dialog will appear showing some network information. Press **OK** to continue.

Select **Forget Learning** from the **Action** menu.

Select **Learn from File...** from the **Action** menu.

Load the training file section from **logic.tti** in the examples directory.

A dialog will appear while the **logic.nnp** neural network learns the information in the **logic.tti** training file. This dialog will disappear when the learning process has finished. On a fast PC this may take less than a second but don't worry the process can be slowed down. You can now interrogate the trained network to verify that it gets the basic logical function correct.

*If you have a fast PC you can slow the learning process down by reducing the learning rate. Select **Control...** and a dialog will appear that allows most of the learning and testing parameters to be changed. Change **Learning rate** to 0.1 and **Momentum** to 0, then press **OK**. This time, if you repeat the exercise, you will be able to see the Learning dialog.*

Creating a new network

Select **Create Network** from the **Action** menu.

Load the training file section from **risk.tti** in the examples directory.

A neural network will be created that will be capable of learning the **risk.tti** training file section. Save it as **risk.nnp** in the examples directory.

Select **Smart Start** from the **Action** menu.

Load the training file section from **risk.tti** in the examples directory.

This will attempt to set the learning parameters to those that will produce the fastest learning rate. The process ends by opening the Control dialog so you can see how well it's done. It does not always get the parameters correct. For this example the learning rate should be 0.64 and the momentum should be 0.79 - you can change them if necessary.

Now you can train and interrogate the network.

Creating a new training, testing and interrogating file

Select **New Training, Testing, Interrogating file** from the **File** menu

A dialog will appear into which you can enter the input and output labels.

Type "A" into **Inputs** and press the **Add Input** button.

Type "B" into **Inputs** and press the **Add Input** button.

Type "Sum" into **Outputs** and press the **Add Output** button.

Press the **Finish** button. Save the new file as **sum.tti** in the examples directory.

A skeleton tti file has now been created. This file can be edited using any spreadsheet or word processor that can handle ASCII text files or you can use the facilities within Neural Planner as follows.

If you edit the tti file using a spreadsheet or word processor you will have to set the high and low limits manually.

Select **Open Training file section...** from the **File** menu

Load the training file section from **sum.tti** in the examples directory.

Select "dummy" in the **Case Names** list box.

Type "example 1" in the **Case Names** edit box.

In the **Labels and Values** edit box next to "A" type a number between 1 and 5.

In the **Labels and Values** edit box next to "B" type a number between 1 and 5.

In the **Labels and Values** edit box next to "Sum" type the sum of the numbers "A" and "B".

Press the **Replace** button.

Repeat this process using different case names and different numbers for about 6 examples. Each time you change the case name and enter the example numbers and sum press the **Copy** button. If you make a mistake you can delete the currently selected case by pressing the **Delete** button.

When you have finished press the **Finish** button. This will set the high and low limits automatically and save the file.
You can now create a network that will learn **sum.tti** using **Create Network**, then set the learning parameters using **Smart Start** and then train the network using **Learn from File...**

After **sum.nnp** has finished learning **sum.tti** you can interrogate the network but first you may want to enter a few cases in the interrogating file section.

Select **Open Interrogating file section...** from the **File** menu

Load the interrogating file section from **sum.tti** in the examples directory.

The process for entering interrogating cases is identical to entering the training cases but you

don't need to enter anything in the "Sum" edit box.

When you interrogate **sum.nnp** you will probably find that some of the answers are incorrect. The network has been trained to a point when it can generalise fairly well but not absolutely accurately. The results produced near to the high and low limits may be very inaccurate.

You can experiment by adding more training cases. You can also try lowering the **Target Error** in the **Control...** dialog. If the network is overtrained by lowering the target error too much it will lose its ability to generalise. The best way to control training is to set up the testing file section.

Controlling the learning process

Select **Open Testing file section...** from the **File** menu

Load the testing file section from **sum.tti** in the examples directory.

Enter some testing cases - these should be different to the training cases but within the same range.

Open the **Control...** dialog, change the **Target Error** to 0, **Cycles Per Test** to 100 and **Range** to 0

Repeat the learning process. The first testing cycle may produce a number of out of range errors. You must correct these errors before continuing. Either reduce the range of the testing cases or widen the range of the training cases.

The learning process will finish when all the testing cases are correct. If more than about 5000 cycles elapse and learning has not finished then you need to experiment with different or more training cases.

When training has finished you can interrogate the network. If the answers are still wrong then keep experimenting - you will soon get to understand the possibilities and limitations of neural networks.

Building a more complex network

The Iris Plants Database has been used many times in neural network research. You will find two text files in the examples sub-directory to help you. The **irisname.txt** file is a description of the database and **irisdata.txt** contains the data. Have a look at these two files and consider what you will need to do to set up the network.

You should have decided to use the petal and sepal dimensions for the inputs to the network and the classification for the output. It is possible to represent the three varieties of iris as a single output. You could use -1 for Setosa, 0 for Versicolour and 1 for Virginica. You can use any three numbers to represent the three varieties but a much better idea is to use three separate outputs. Each variety has it's own output with a value of 0 for false and 1 for true.

The first step is to create a suitable skeleton tti file using Neural Planner.

Select **New Training, Testing, Interrogating file** from the **File** menu

Add four input labels; petal length, petal width, sepal length and sepal width.
Add three output labels; Setosa, Versicolour and Virginica.

Press the **Finish** button. Save the new file as **iris.tti** in the examples directory.

You could now use the Neural Planner facilities to enter all the input and output values by hand but it is a lot easier to load **iris.tti** into a spreadsheet. That way you can use cut and paste from the **irisdata.txt** file to the tti file. If you decide to use a spreadsheet don't forget to save the **iris.tti** file in text format. Don't forget to set the LIMITS section - you can do this manually or using Neural Planner as follows:

Select **Open Training file section...** from the **File** menu, load **iris.tti** and then push the **Finish** button.

Once you have completed **iris.tti** you can produce a suitable network using **Create Network** and set the leaning parameters with **Smart Start**.

The iris data contains some duplicates - Neural Planner will detect these and ask you what you want to do about them. In this case it make very little difference what you do so just take the default action and let Neural Planner delete the duplicated cases.

This is a classification problem so you will need to change the **Control...** dialog to perform automatic testing. Change the **Target Error** to 0.0, the **Cycles Per Test** to 100 and the **Range** to 0. You will also need to set up the [TESTING] section of the tti file. It is possible to get a 100% classification accuracy by using the same testing and training data but this is not a good idea. What you should do is take out half of the training data and use it for testing. You must leave some training examples for each variety. If you split the data for each variety into two you should be able to get a peak classification accuracy of 96% within 5,000 cycles.