Appendix C

This is the documentation for the UTS neural network, as described in chapter 8.

Program MAP_NEURAL_AUSTENITIC_UTS

Provenance of Source Code

Iqbal Shah, Thomas Sourmail and H. K. D. H. Bhadeshia,
Phase Transformations Group,
Department of Materials Science and Metallurgy,
University of Cambridge,
Cambridge, U.K.

The neural network program was produced by:

David MacKay,
Cavendish Laboratory,
University of Cambridge,
Madingley Road,
Cambridge, CB3 0HE, U.K.

Added to MAP: Sept 2002

Purpose

To estimate the UTS of austenitic stainless steels at particular temperatures as a function of chemical composition and heat treatments.

Specification

Language:	FORTRAN / C
Product form:	Source code / Executable files
Operating System:	Solaris 5.5.1 & Windows 95

Description

The modelling procedure is a purely empirical one, and is based on a neural network program called **`generate44`**, which was developed by David MacKay and is part of the **`bigback5`** program. The model is constituted of a committee of several individual neural networks. It was trained on a set of experimental data for which the "outputs" are known, and creates a kind of non-linear, multi-parameter "regression" of the outputs versus the inputs. This "regression" has already been produced and the model is delivered ready to perform predictions for steels of any desired composition (within certain specified limits). The source code for the neural network program can be downloaded from David MacKay's website; the executable files only are available from MAP.

- The program runs on a Solaris 5.1.1 UNIX operating system, Linux and Irix. The files for UNIX are separated compressed into a file called *unix_aus_uts.tar.gz.*

The .tar file contains the following files:

`READMEUTS`
: A manual containing step-by-step instructions for running the program, including a list of input variables.

`MINMAX`
: A text file containing the minimum and maximum limits of each input and output variable. This file is used to normalise and unnormalise the input and output data.

`test.dat`
: An input text file containing the input variables used for predictions, together with an example set of data.

`result_test.txt`
: Contains the results you should expect from the example set of data. To test the model is running properly on your computer, use the given 'test.dat' file to do predictions and compare the 'result' file with this file.

model.gen

This is a unix shell file containing the command steps required to run the module. It can be executed by typing **sh model.gen** at the command prompt. These shell files run all the programs necessary for normalising the input data, executing the network for each model, unnormalising the output data and combining the results of each model to produce the final *committee* result.

spec.t1

Created by **generate_spec**, which contains information about the module and the number of data items being supplied. It is read by the program **generate44**.

.generate_spec (hidden)

This executable file creates a file called **spec.t1**, required by generate44.

.randomise (hidden)

This executable file creates a file called **norm_test.in**, which contains the normalised equivalent of the input data found in test.dat. It requires the MINMAX file

.generate44

This is the executable file for the neural network program. It reads the normalised input data file, **norm_test.in** (created by normalise) , and uses the weight files in subdirectory **c**, to find a value for the output. The results are written to the temporary output file **_out**.

.gencom

This executable file combines the predictions of the different models in the committee and calculates the combined error bar. **.treatout**
This executable un-normalise the committee predictions and produces the file 'result'.

result

Contains the final un-normalised committee results for the predicted output.

SUBDIRECTORY c

_w*f

The weights files for the different models.

***.lu**

Files containing information for calculating the size of the error bars for the different models.

_c*

Files containing information about the perceived significance value for each model.

_R*

Files containing values for the noise, test error and log predictive error for each model.

SUBDIRECTORY d

outran.x

A normalised output file which was created during the building of the model. It is accessed by **generate44** via **spec.t1**.

SUBDIRECTORY outprdt

com.dat

The normalised output file containing the committee results. It is generated by **`.gencom`**.

Detailed instructions on the use of the program are given in the README file.

References

Materials Science and Technology, 18 (2002) 655

D.J.C. MacKay, 1997, *Mathematical Modelling of Weld Phenomena 3*, eds. H. Cerjak & H.K.D.H. Bhadeshia, Inst. of Materials, London, pp 359.

D.J.C MacKay's website at:
http://wol.ra.phy.cam.ac.uk/mackay/README.html#Source_code

Parameters

Input parameters

The input variables for the model are listed in the **`README or README.DOC`** file in the corresponding directory. The maximum and minimum values for each variable are given in the file **`MINMAX`**.

Output parameters

These programs give the yield strength in 'MPa'. The corresponding output files is called **`Model_RESULT.dat or Result`**. The format of the output file is:

```
Prediction      Error   Upper-limit     Lower-limit
(MPa)                   (MPa)           (MPa)
```

A more detailed description is presented in the README file.

Error Indicators

None.

Accuracy

An estimated predictive error bar is provided by the model.

Further Comments

See Neuromat Ltd for a user friendly way of making predictions.

Example

1. Download the model

Uncompress the "*unix_aus_uts.tar.gz*" file in a dedicated directory (for example: "neural").
On UNIX systems, this is done by:

- *gzip -d unix_aus_uts.tar.gz*
- *tar -xvf unix_aus_uts.tar.gz*

2. Program data

3. Running the program (making predictions)

For Solaris 5.5.1, Linux or SGI, just type:

sh model.gen

4. Results of the program (predictions)

The results are written in the "Result" or "model_result.dat" file, as described in the README file. In the present case:

Auxiliary Routines

Keywords

neural network, UTS, strength, austenitic stainless steel