Neural Networks: Tricks of the Trade

R. C. Dimitriu

1 Data

The first thing necessary to make a reliable neural network model is good quality data which are physically meaningful. It is also necessary to optimise the number of input variables. An over–ambitious set will limit the data available for analysis. A pragmatic approach ensures sufficient data and variables to capture the complexity of the problem.

Fig. 1a and b show the model–perceived noise (σ) for a database with 22 and 26 input parameters respectively. The noise is noticeably smaller because of the increase in the number of inputs which contribute to the output.

Figure 2 illustrates how the uniformity with which data are distributed may not be the same for all inputs. In the case of nickel, it is likely that predictions will be associated with large uncertainties in the concentration range 1–2 wt%.

The maximum, minimum and standard deviations of each input and output should be checked before commencing the modelling, to ensure that these limits are meaningful. Fig. 3 shows such data representing steels. The first row is the carbon concentration. Suppose that the maximum carbon concentration is listed as 10 instead of 0.48, then a metallurgist would be able to see this as unphysical, and probably a consequence of a compiling error in the database.

Care should be exercises in selecting the kind of data used in developing the neural network. An assessment should be made of the quality of the data, for example, the accuracy and reliability of the experiments used to generate the data. One way of studying the behaviour of the data is to identify *outliers* after creating the neural network model Fig. 4. Outliers are those points whose 95% confidence limits are so far away from expectation that they are unlikely. Such data should be tracked back to their origin to see if there is a mistake in the process of collecting and compiling the database. If no errors are found then some reasoning has to be applied to ensure the the model construction is correct.

Before the training begins we have the possibility to fix the noise level. Fixing the noise level is an option if we know that we have a noisy database we can increase the noise level because if we let the implicit value that means the neural network will overtrain the data base until it reaches the implicit noise level which is not a good thing. If we a good database we might reduce the noise level and in that case we force the model to search for more complex solutions and that means a better prediction.

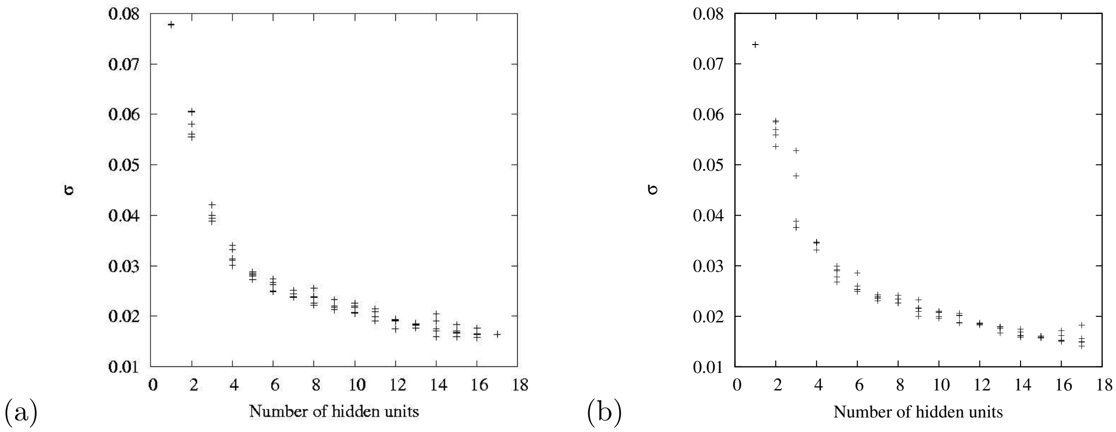

Figure 1: (a) Perceived level of noise from a neural network model with 22 input parameters. (b) As for (a) but with 26 inputs.

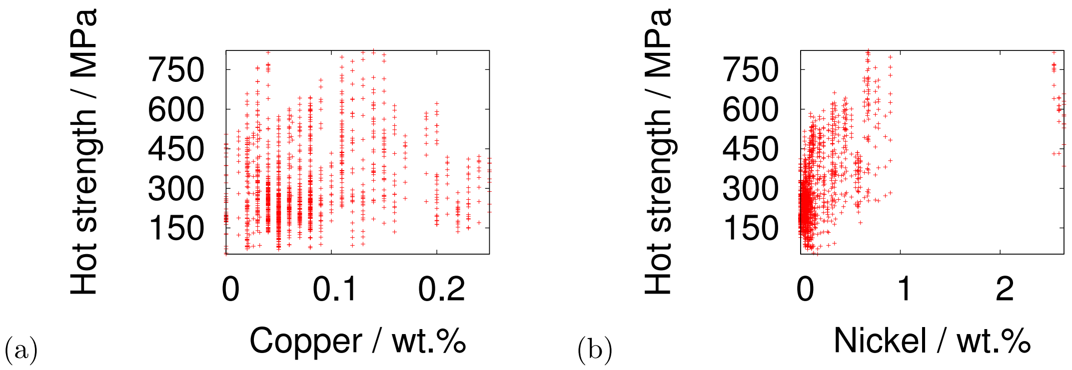

Figure 2: The uniformity in the distribution of data.

*****
Below are the minima and maxima for your database, please check them carefully as they reveal easily important errors.
*****

Min	Max	MinNo	MaxNo	Average	StDev
0.0400	0.4800	800	1079	0.1514	0.0627
0.0300	0.8600	1096	476	0.3624	0.1430
0.3500	1.7300	1039	820	0.6091	0.2402
0.0050	0.0290	1049	29	0.0149	0.0056
0.0010	0.1000	1019	839	0.0101	0.0085
0.0001	2.6400	323	800	0.2216	0.4075
0.0001	12.9000	1200	47	3.6030	4.1140
0.0050	2.0300	1192	1086	0.6944	0.3855
0.0001	0.2500	1106	403	0.0754	0.0541
0.0001	1.7400	83	1106	0.0903	0.3209
0.0001	0.3200	83	810	0.0643	0.1085
0.0001	0.1500	83	47	0.0041	0.0183
0.0010	0.2800	1039	820	0.0146	0.0339
0.0030	0.0590	1152	1049	0.0136	0.0113
0.0000	0.0100	1	800	0.0002	0.0011
0.0001	2.1600	1	810	0.0463	0.3094
0.0001	0.0900	1	1082	0.0046	0.0172
10.0000	1680.0000	197	1368	177.1775	330.5588
30.0000	1800.0000	1039	1323	260.9455	395.2508
1143.1500	1373.1500	530	1106	1218.0482	54.4109
873.1500	1143.1500	1079	1296	972.3664	50.4263
373.1500	1073.1500	1	1116	688.9318	178.4568
-1.6296	1.0906	39	1127	0.1613	0.4429

*****
Above are the minima and maxima for your database, please check them carefully as they reveal easily important errors.
*****

Figure 3: The minimum-maximum table from neural network

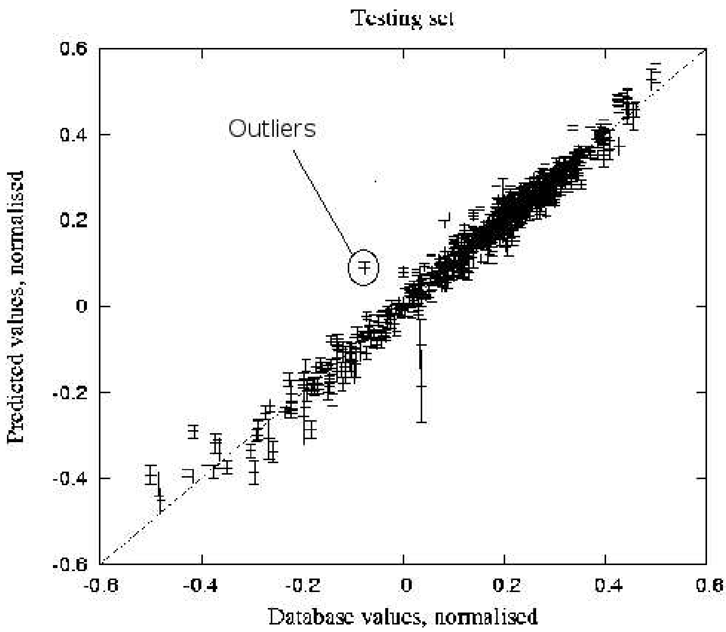

Figure 4: Example of an outlier

2 Definition of Inputs

The input parameters can be inserted in the database in raw form, for example the chemical composition of a steel, or in a functional form, for example the free–energy of transformation calculated from the composition. However, in the latter case, both the raw data and the functional form should be included as inputs to avoid biasing the model.

Suppose that the time period of heat–treatment is an input, then it would also be reasonable to include another parameter, the logarithm of time ln{time}, since the latter form is expected from kinetic theory.

Temperature is expected from theory to act via the function

$$\exp\left\{-\frac{Q}{RT}\right\} \tag{1}$$

where R is the universal gas constant, T is the temperature in Kelvin and Q is the activation energy. This formula is describes a dependence of the output upon an activation energy, and is used because it incorporates the physical relationship based on scientific understanding.

In order to allow a comparison between the influences of different variables, for example, the sensitivity of the output to the inputs, it is useful to normalise each variable between $0 - -1$ or -0.5–0.5. Otherwise it becomes difficult to compare, for example, the effect of carbon which varies in concentration from 0–0.1 wt% with chromium which can reach values in excess of 13 wt%.

If a combination of two or more existing variables has a particular significance it can be added to the database. For example, the following could be used as an input from kinetic theory, a function of time and temperature:

$$\text{time} \times \exp\left\{-\frac{Q}{RT}\right\} \tag{2}$$

3 Definition of Outputs

A neural network is a mathematical function and can predict unphysical values of output. For example, the strength of a material cannot be negative and yet a mathematical function is capable of making such a prediction. To avoid this, the output, in this case strength, should be defined in such a way that it cannot be negative:

$$\text{function of strength} = \ln\left\{-\ln\left(1 - \frac{x_{max} - x}{x_{max} - x_{min}}\right)\right\} \tag{3}$$

where x is the strength and x_{max} and x_{min} are values that are set by the user.