Hot-Strength of Ferritic Creep-Resistant Steels

Comparison of Neural Network and Genetic Programming
Strength of Ferritic Steels?
Neural networks

Neural networks in a Bayesian framework used to model the hot-strength of ferritic steels

Neural network: non-linear method of regression

Data fitted to a function to capture complex relations between inputs and output
Three layer neural network

- input variables
- hidden units
- bias
- output
General form of neural network

\[ y = \sum_{i} w_{i}^{(2)} h_{i} + \theta^{(2)} \]

\[ h_{i} = \tanh\left( \sum_{j} w_{ij}^{(1)} x_{j} + \theta_{i}^{(1)} \right) \]

y - output
w - weight
h - hidden unit
\( \theta \) - bias
x - inputs
i, j - subscripts
Hyperbolic tangents

- transfer function
- very flexible

- can be combined

\[ y \]

\[ f(x_j) \]
Genetic programming

Machine learning technique to optimise a population of computer programs

Program may be an expression, formula, plan, control strategy, decision tree or learning model
Genetic programming evolution cycle

Random Generation → Fitness Evaluation → Selection → Genetic Operators

Individuals → Replacement
Genetic programming functions

- arithmetic (+, -, *, /)
- elementary (exp, log, power, ...)
- trigonometric (sin, cos, tan, ...)
- genetic operators (mutation, ...)
Input variables

- Al, C, Cu, Cr, Mn, Mo, Ni, N, Si
- Austenitising time and temperature
- Tempering time and temperature
- Test temperature
<table>
<thead>
<tr>
<th>Variable</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluminium / wt%</td>
<td>0.001</td>
<td>0.04</td>
</tr>
<tr>
<td>Carbon / wt%</td>
<td>0.09</td>
<td>0.48</td>
</tr>
<tr>
<td>Copper / wt%</td>
<td>0.0001</td>
<td>0.25</td>
</tr>
<tr>
<td>Chromium / wt%</td>
<td>0.0001</td>
<td>12.38</td>
</tr>
<tr>
<td>Manganese / wt%</td>
<td>0.38</td>
<td>1.44</td>
</tr>
<tr>
<td>Molybdenum / wt%</td>
<td>0.01</td>
<td>1.05</td>
</tr>
<tr>
<td>Nickel / wt%</td>
<td>0.0001</td>
<td>0.6</td>
</tr>
<tr>
<td>Nitrogen / wt%</td>
<td>0.001</td>
<td>0.04</td>
</tr>
<tr>
<td>Silicon / wt%</td>
<td>0.18</td>
<td>0.86</td>
</tr>
<tr>
<td>Austenitising time / min</td>
<td>10</td>
<td>5400</td>
</tr>
<tr>
<td>Tempering time / min</td>
<td>30</td>
<td>660</td>
</tr>
<tr>
<td>Austenitising temperature / K</td>
<td>1143.15</td>
<td>1243.15</td>
</tr>
<tr>
<td>Tempering temperature / K</td>
<td>898.15</td>
<td>1023.15</td>
</tr>
<tr>
<td>Test temperature / K</td>
<td>293.15</td>
<td>973.15</td>
</tr>
<tr>
<td>Hot strength / MPa</td>
<td>69</td>
<td>660</td>
</tr>
</tbody>
</table>
neural network

Hot Strength / MPa

Test Temperature / K
genetic programming using trigonometric functions
genetic programming using trigonometric functions
genetic programming using all functions
Conclusions

Both methods were similar in capturing the two regimes in the decrease of hot strength.

The neural network having a plus in also capturing the slopes correctly.
ACKNOWLEDGEMENTS

- Many thanks to professor H. K. D. H. Bhadeshia
- Department of Material Science and Metallurgy
- University of Cambridge
- European Commission under the Marie Curie Early Stage Research Training Programme
Thank you for listening