Predicting octane number using nuclear magnetic resonance spectroscopy and artificial neural networks

Abdul Gani Abdul Jameel [*a], Vincent Van Oudenhoven [a,b] Abdul-Hamid Emwas [c], and S. Mani Sarathy [a*]

[a] *King Abdullah University of Science and Technology (KAUST), Clean Combustion Research Center (CCRC), Thuwal 23955-6900, Saudi Arabia*

[b] *Department of Computer and Electrical Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1*

[c] *King Abdullah University of Science and Technology (KAUST), Imaging and Characterization Core Laboratory, Thuwal 23955-6900, Saudi Arabia*

*Corresponding author:

Abdul Gani Abdul Jameel <abdulgani.abduljameel@kaust.edu.sa>

S. Mani Sarathy <mani.sarathy@kaust.edu.sa

ANN Detailed Methodology

Each node has an associated weight that is directly proportional to the influence of that particular feature on the final output of the network. These weights are typically initialized randomly. For this study feed forward neural networks [1] were used. The data moves in a single direction: from the input layer to the output. During the "training" phase, where the dataset of interest is "fed" in to the network, the weights are adjusted through a method known as "backpropagation [2]"; the act of propagating information back through the network, such as the gradient of the loss function of interest, such as mean absolute error, root mean squared (RMS) error, and so forth. Each weight is adjusted respectively. After a certain number of iterations, also known as epochs, of the above process, a local minima will have been found for the specific ANN and dataset. The final model consists of the original architecture (the topology of the model) in addition to a matrix of optimized weights. A wide range of (linear) combinations of the original features is not sufficient to represent nonlinear functions. Rather, an "activation function" is needed to provide a saturation limit such as the sigmoid function or the 'rectifier [3]', which is presently the most popular function for deep ANNs. It is this combination that allows for Artificial Neural Networks to represent complex and non-linear relationships between the given inputs and outputs.

Multiple feed forward architectures were tested, some of which are presented in table S1 in the form of units in each layer separated by a dash, where the first represents the input layer. It must be noted that these models were not tuned in the same fashion as the final models, as the regularization coefficients and loss function were not optimized.

Table S1

Sample of tested ANN architectures

Architecture	Mean error (RON)			Mean error (MON)		
	absolute	RMS	percentage	absolute	RMS	percentage
9-20-1	3.9	5.1	4.4	4.6	5.7	5.7
9-20-40-1	2.9	4.1	3.3	3.6	4.6	4.5
9-80-160-1	2.4	3.8	2.8	2.8	3.9	3.4
9-200-400-1	2.5	4.0	3.0	2.6	4.1	3.0
9-160-80-40-1	2.9	4.6	3.2	3.4	4.9	4.1
9-80-160-80-1	2.8	4.2	3.1	2.7	4.6	3.3
9-20-40-80-160-1	3.1	3.9	3.4	3.4	4.8	4.1

Overall, models with two hidden layers and a high number of nodes resulted in the lowest overall error metrics and best generalization performance. The most likely explanation for this result is that 'deeper models', models with more hidden layers, have too many features for the used dataset, and therefore 'overfit' the training data. Therefore, the network is effectively 'memorizing' the individual data points, as opposed to finding a general pattern, hence performing poorly on the test set. This is subject to improvement with a larger dataset.

A common method for combating overfitting is known as "regularization". There are many different methods, although they all boil down to the penalization of solutions that are "complex". L1 [4] regularization penalizes solutions with a relatively high number of non-zero weights – solutions that depend on a lot of features, while L2 [4] regularization penalizes solutions with large weight coefficients – solutions that depend heavily on single features. Another method of interest known as 'dropout [5]' drops arbitrary nodes in the network during each epoch, forcing the network not to rely heavily on specific nodes thus reducing the likelihood of over fitting.

Apart from the architecture, several other parameters were varied, including the regularization type, the optimizer, the loss function, and the number of training epochs. L1, L2, and 'dropout' were tested as regularization methods. Dropout lead to the highest stability and the lowest overall error metrics. 'Adam [6]' an extension of stochastic gradient descent, was

settled upon as the optimizer. The optimal loss function differed for the two models: for RON it was found to be mean squared error, while for MON it was mean absolute error. Lastly, 5000 epochs were chosen due to empirical evidence. The summaries of the developed ANN models for RON and MON are reported in Table S2 and Table S3.

Table S2: RON ANN Model Summary

Layer (type)	Output Shape	Param #	Connected to
dense_370 (Dense)	(None, 341)	3410	dense_input_1[0][0]
dropout_247 (Dropout)	(None, 341)	0	dense_370[0][0]
dense_371 (Dense)	(None, 603)	206226	dropout_247[0][0]
dropout_248 (Dropout)	(None, 603)	0	dense_371[0][0]
dense_372 (Dense)	(None, 1)	604	dropout_248[0][0]

Total parameters: 210240

Table S3: MON ANN Model Summary

Layer (type)	Output Shape	Param #	Connected to
dense_298 (Dense)	(None, 541)	5410	dense_input_3[0][0]
dropout_199 (Dropout)	(None, 541)	0	dense_298[0][0]
dense_299 (Dense)	(None, 314)	170188	dropout_199[0][0]
dropout_200 (Dropout)	(None, 314)	0	dense_299[0][0]
dense_300 (Dense)	(None, 1)	315	dropout_200[0][0]

Total parameters: 175913

(1) Bebis, G.; Georgiopoulos, M. *IEEE Potentials* **1994**, *13* (4), 27–31.

(2) Hecht-Nielsen. In *International Joint Conference on Neural Networks*; IEEE, 1989; Vol. 1, pp 593–605 vol.1.

(3) Glorot, X.; Bordes, A.; Bengio, Y. In *AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*; 2011; Vol. 15, pp 315–323.

(4) Ng, A. Y. In *Twenty-first international conference on Machine learning - ICML '04*; ACM Press: New York, New York, USA, 2004; p 78.

(5) N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Y. B. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

(6) Kingma, D. P.; Ba, J. **2014**.