Neuromelt model for estimating mold flux melting behaviour

M. Vargas Hernandez, C. Mapelli, J. Cho, N. Kölbl, I. Marschall, M. Alloni, R. Carli

The existing models and methods used to determine the melting temperature of the mold powders used in the continuous casting process remain inaccurate in the case of equations reported in current literature or consider for commercial software only an equilibrium state. In this work, a new approach has been implemented using neural networks, which will act as a "black box" to predict the melting temperature of a mold powder with a given composition within an acceptable range. The proposed neural network will be working as a regression neural model whose inputs will be the composition of each of the chemical species contained within the powder. A database provided by a research net comprising multiple countries' research institutes will be fed as a training set for network learning. Such data comes from experimental measurements performed mainly through the high-temperature microscope test. The correct implementation and training of the network should provide a new alternative to develop new products and verify existing products' melting properties. In future models, further considerations should be made towards a better understanding of these phenomena, which should consider factors such as the formation of mineral phases, interaction among some specific components of the powder, or even the parameters used at the time of experimental measurement.

KEYWORDS: NEURAL NETWORK, MOLD POWDER, LIQUIDUS TEMPERATURE

INTRODUCTION

Mold powders perform the decisive roles to achieve both the sound quality and productivity of continuous casting process: improving lubrication, controlling mold heat transfer, absorbing non-metallic inclusions, and preventing reoxidation. Therefore, the melting behavior of casting powders is always the most important factor for these performances, as it governs refining ability of mold slags on mold top surface and then the infiltration of mold slag film at the meniscus. In particular, maintaining the adequate melting behavior of casting powders is essential to develop the successful casting powders suitable for high speed casting of steels by balancing the infiltration and consumption of slag film.[1,2] Without a doubt, this can only be done on the basis of the fundamental understanding and quantification of the casting powder melting. However, little is known about the exact mechanisms and sequences that take places during the melting process of the mold powders. [3]. Such unknown nature of the process renders the task of determining parameters such as the liquidus temperature of the powder

M. Vargas Hernandez, C. Mapelli
Politecnico di Milano, Italy

J. Cho
Pohang University of Science and Technology, South Korea

N. Kölbl, I. Marschall
Montanuniversitat, Leoben, Austria

M. Alloni, R. Carli
Prosimet, Italy

(T_{liq} here used as synonym of melting temperature) particularly difficult since a different composition of the mold powder results in a different T_{liq}. The importance of having a reliable model that could predict in an accurate manner the T_{liq} of the powder according to its composition lies in the fact that it would prove useful for the designer or the user to know beforehand what would be the melting behavior of a certain powder composition to verify that a certain formulation complies with working parameters.

Even though there are several established methods for the estimation of the melting temperature of the mold powders according to their chemical composition, most of these methods prove to be still quite uncertain to some degree, which still places experimental measurements as the most reliable method for determining T_{liq} of a mold powder. However, the current measurement methods still prove to be relatively expensive and time consuming. [4] For this reason, a new approach has been taken in order to create a model that could provide a reliable prediction of the T_{liq} according to its chemical composition without the immediate need of understating the complex nature of the melting behavior of the mold powder. This new approach consists on the implementation of a neural network for the prediction of data, in this case, the melting temperature of a mold powder of a given chemical composition. The network will work as a "black box" that, after being properly trained with experimental data, will yield an approximation of T_{liq} of the powder even without a fully understandable relationship between an input (mold powder chemical composition) and an output (liquidus temperature of the mold powder). This work will focus on analyzing the performance of the preliminary model and its parameters given a relatively small amount of training data.

EXPERIMENTAL

Samples and measurements

First, for the neural network to be implemented, a set of data was needed, also known as training data. This data comes exclusively from sample measurements taken from a wide variety of commercial mold powders, all of them having a different chemical composition. The measurement data had to be provided in a joint effort by Prosimet in Italy, the Pohang University of Science and Technology in South Korea and the Montanuniversitat in Leoben, Austria. Collected data were measured with different experimental methods from lab to lab but considered as a coherent data set. The data set consists of 284 measurements. Even though this measurement data set batch could be considered small, it could still provide an accurate preliminary approximation that could justify working in the future with larger data sets in the case of obtaining a positive result with this batch of data.

In Prosimet for example, melting temperatures were measured using a hot stage microscope MISURA by Expert System solutions equipped with software MISURA 3.32HSM. In measuring melting temperature, samples of industrial casting powders were ground in an agate mortar. The powder obtained was pressed in a cylindrical piece 2 mm x 1 mm, then inserted into the instrument oven and heated at a rate of 10°C/min. The height of the cylinder (expressed in percent of starting value) was registered during heating to find the sample's sintering, melting and flowing temperatures.

Tab.1 - Chemical composition of one of the samples used.

Product:	Powder A
Component	**Concentration [%]**
Ctot.	8.34
Cfree	6.27
SiO_2	37.43
Fe_2O_3	1.73
Al_2O_3	5.97
CaO	19.75
CaF_2	12.43
Na_2O+K_2O	2.17
MnO	6.24
S	0
B_2O_3	0.09
Li_2O	0
ZrO_2	0
TiO_2	0.08
P_2O_5	0.12
MgO	0.062
Tliq[ºC]	1185

Data setup

The composition of the mold powders was established to be given in terms of 16 different chemical species as usually given for commercial powders. An example of one of the samples is given in Tab. 1

The following considerations were made:

- It was decided to group Na_2O and K_2O as a single component (referred to as "alkali") given their similar behavior and function in the powder as fluxes and network breakers.
- All volatile components (lost during melting) were not considered for further calculations. The amount of C regulates the melting rate, but certainly not the liquidus temperature of the powder.

Reducing the number of inputs for the network is beneficial for the simplicity of the model, especially when a considerably small amount of data is being used to train the neural network. Furthermore, the data is normalized.

In summary, the network is set to be working with 13 inputs from 284 measurements.

Neural network setup

The neural network trained for this exercise was created by using the Machine Learning Toolbox available in Matlab 2019. Two separate simulations were created for each data set. The first simulation (Simulation A) was implemented with the purpose of having a wider control on parameters of the network such as the number of neurons in the hidden layer, number of hidden layers and the data division The synthetic structure of the network is shown in Fig.1. The second simulation (Simulation B) was created in order to test other remaining parameters such as the learning algorithm of the network. The performance of the network was measured by calculating the root mean square error (RMSE) between the predicted values of the network and the actual measured data. Finally, the optimized network was tested with two measurements not used to train the network.

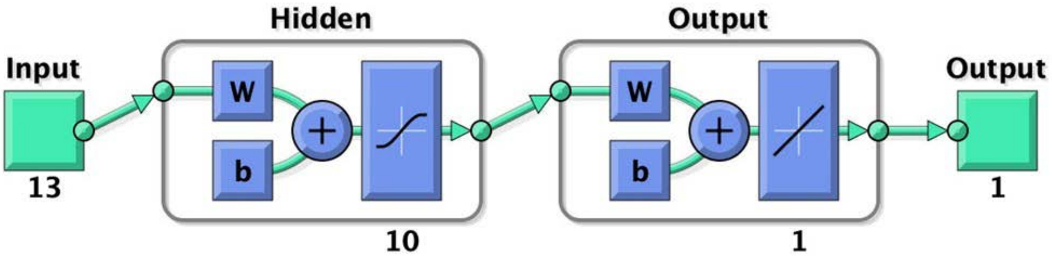

Fig.1 - The structure of the network of Simulation A consists of an input layer with13 inputs (chemical composition), a hidden layer consisting of a variable number of neurons with adjustable weights and biases and an output layer consisting of a single output (liquidus temperature).

RESULTS

Data analysis

To justify the implementation of a neural network in this case, it must be verified that there's no clear relationship between the inputs and the output. After plotting each and every single one of the components against the liquidus temperature, only the $N_2O + K_2O$ component showed some degree of correlation as Fig.2b

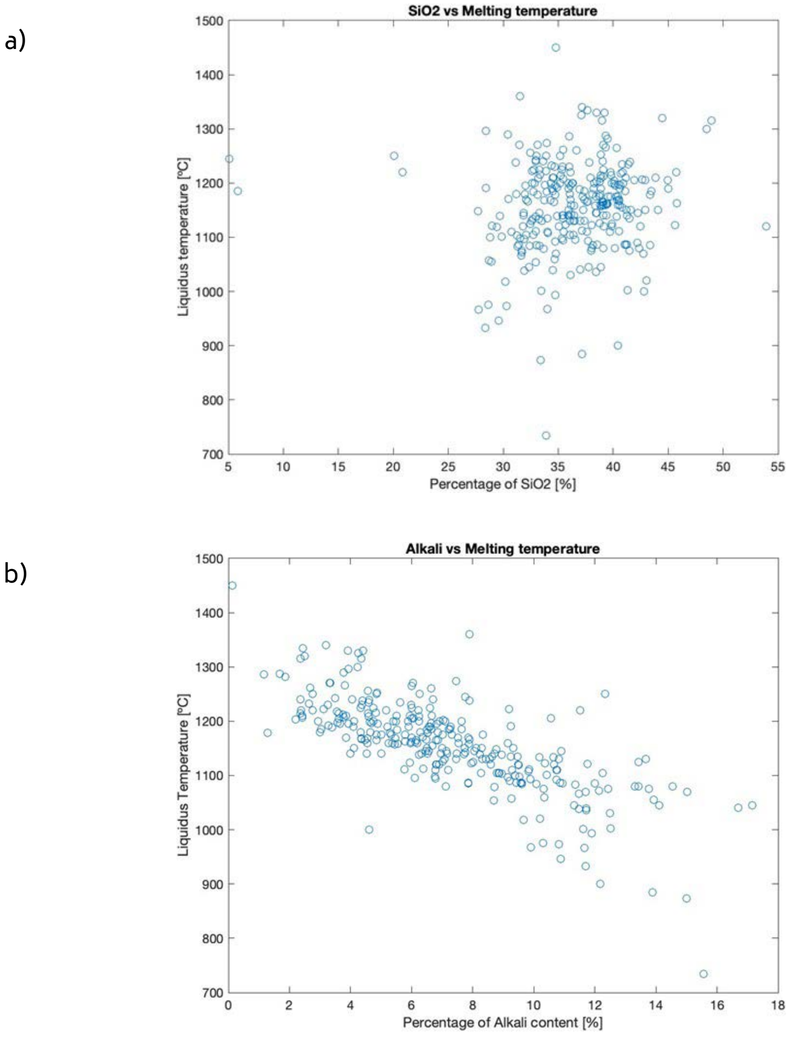

Fig.2 - Out of the 13 components considered for the analysis, 12 showed no correlation at all when comparing its concentration with the melting temperature of the powder. A clear example is the is the SiO_2 , shown at a). However, as shown in b), there's a slight negative correlation between the alkali content and the liquidus temperature.

Neural network parameter optimization

For the Simulation A, the parameters that were tested were the number of neurons in the hidden layer, the possibility of using more than one hidden layer and the data division. As an initial approach, a shallow neural network with a single hidden layer consisting of 5 neurons was used. The data division was established to be 70% for the training set and the remaining 30% for the validation set. Surprisingly, this first approach resulted to be very promising. The performance of a neural network can be measured by calculating the root mean square error between the predicted data of the network and the measured data set that the network was trained with. Using a script that could vary the size of the hidden layer from 1 to 60 neurons, it was found that a minimum RMSE could be obtained when the size of the hidden layer was around 9 and 11 neurons. Implementing a network with fewer neurons would have meant that the network was underfitted, too simple for the data the network is trying to predict. On the other hand, a model with an excessive number of neurons in the hidden layer would have resulted in an overfitted network, too complex for the data in hand [5]. In this case, for the optimal number of neurons, a RMSE of 28.05 °C was obtained for the training set and of 30.04°C for the validation set.

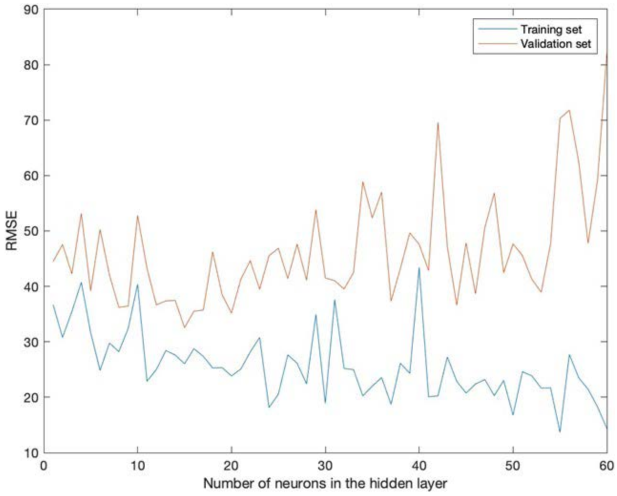

Fig.3 - Optimization of a shallow neural network going from 1 to 60 neurons. There is a range located approximately at 9 11 neurons where the RMSE between the training set and the validation set seem to have the best agreement between them. After this, the RMSE of the validation set continues to increase steadily.

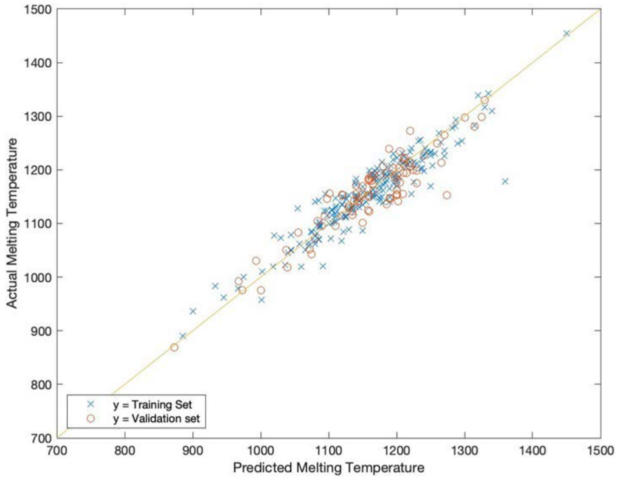

Fig.4 - Comparison between the predicted values given by the network and the actual measured values .

It should be noted that even though in Fig.3 there's a clear tendency for the RMSE to be in an ever-decreasing trend as the number of neurons in the hidden layer grows, the data set that should be considered for the actual performance of the network is the validation set. The training set is used to adjust the biases and weights of the network whereas the validation set tests how well the network generalizes and stops the training of the network once there's no further improvement. A network with a high RMSE in the validation set won't be reliable when trying to predict an output from new data. [6]

Networks using more than one hidden layer were also tested, but the results were, in the best cases, marginally better or showed no improvement at all. It could be inferred that a network using more than one hidden layer would result in a network that is too complex for the data we are trying to predict.

In the case of Simulation B, the only parameter that was tested was the learning algorithm. The LevenbergMarquardt yielded the best results. This learning algorithm results useful when working with small data sets and when a very accurate training is required. However, this algorithm starts performing poorly when the network becomes noticeably complex. [7] The Scaled Conjugated Gradient algorithm also yielded similar results as shown in Fig.6. This learning algorithm required more time for its operation but, according to the Fitting App, required less memory. Moreover, since Matlab's Neural Network Fitting App was used in this case, other parameters were inevitably change. Specifically, the data was divided in a slightly different way. The network adjusted the validation set to a 15% while adding a test set that used 15% of the data. The training set remained the same. Given the particularly small amount of data the network is working with, it is not convenient to use a test set, which usually serves the purpose of testing the network with completely new data. Nevertheless, the network itself resulted to be arguably accurate as Fig.5 shows, despite the limited amount of data available for the exercise. Having previously optimized the other parameters for Simulation A, Simulation B used the same amount of neurons, and hidden layers than Simulation A.

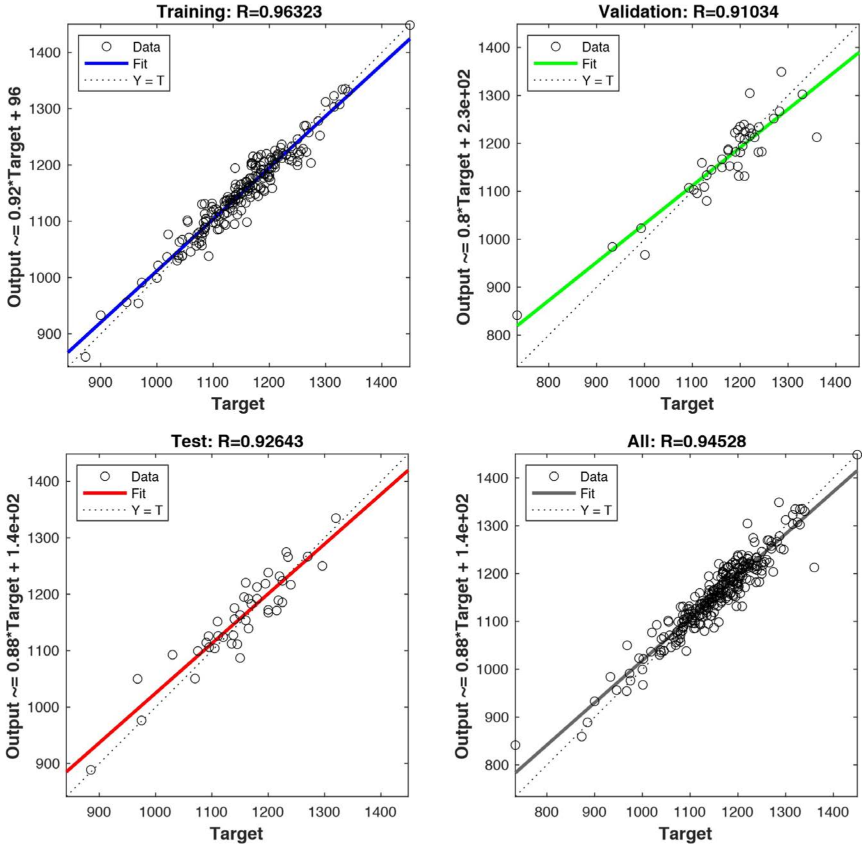

Fig.5 - For the Neural Network Fitting App it was possible to obtain a strong correlation among all data sets using the Levenberg-Marquardt learning algorithm.

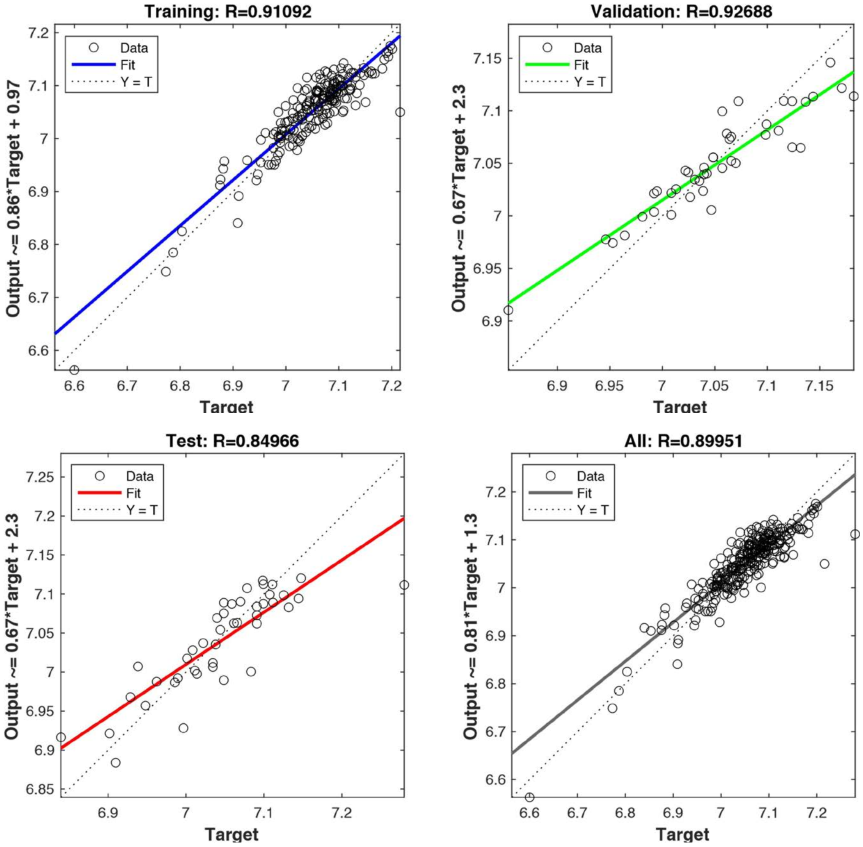

Fig.6 - Using the Scaled Conjugated Gradient learning algorithm, a similar set of results were obtained using less memory than the Levenberg-Marquardt learning algorithm.

Finally, from Simulation B, a script was derived for actual data prediction. This script requires the user to write the chemical composition of the mold powder as an input and the script will try to predict the T_{liq} of the powder as an output. Prosimet provided two new measurements that weren't used for training the network with the purpose of testing the accuracy of the network. As shown in Fig.6 the script derived from the network could predict the melting temperature of two different mold powders with an error of approximately 5 °C in both cases.

Tab.

Tab.2 -Chemical composition of two mold powder samples used as new data. The table shows a comparison between the measured.

Product:	Powder A	Powder B
Component	**Concentration [%]**	
SiO_2	39.26	37.45
Fe_2O_3	0.43	1.01
Al_2O_3	1.58	14.59
CaO	28.31	24.001
CaF_2	18.48	14.74
MgO	0.88	0.67
Na_2O+K_2O	9.009	6.92
MnO	1.88	0.013
B_2O_3	0	0
Li_2O	0	0
ZrO_2	0	0
TiO_2	0.022	0.28
P_2O_5	0.011	0.027
	0.08	0.08
Tliq[°C]	1120	1180
Predicted Tliq[°C]	1125.8	1185.5

DISCUSSION

Even though the implemented neural network was working with a relatively small data batch, it was still possible to render it accurate enough by adjusting the parameters for an adequate operation of the network. It is worth mentioning that there are other factors that have yet to be considered for the network. For example, the computing power is a determining factor for the network. Different systems will yield different results and even the same equipment will yield a different network efficiency with each iteration. [8]. Furthermore, the analysis of the network itself already shows it to be slightly more accurate than the existing empirical equations, which have uncertainty degrees of +- 30 °C in the best cases. Evidently, the first improvement that should be considered for this network is the amount of data used for the training of the network.

Once the network has been improved and optimized to be even more accurate, it would be possible to extract some data from the network itself for further analysis of the melting process of the mold powders. For instance, given that the network assigns weights and biases to each of the inputs, we can deduct from the network which chemical species have a larger influence for determining the liquidus temperature of a certain powder. Along with other phenomena known to happen during the melting process, like the formation of certain mineral phases [3], new theories and possibly new considerations could arise for the manufacturing of these powders.

CONCLUSIONS

The mechanisms and sequences of the melting of mold powders used for continuous casting are still not well understood. Because of this, scientists and engineers still have to rely on experimental measurements, which result to be expensive and time consuming. The available solutions and approaches offer a certain degree of certainty which can be vastly improved. The implementation of a neural network for the prediction of the liquidus temperature of a mold based solely on its composition proves to be an interesting alternative, since with a limited amount of measurements, a surprisingly accurate approximation could be done. For the proposed preliminary network, the parameters had to be optimized to work with a small amount of observations, yet the largest improvement that can be done for the network is to train it with a lot more data. The success in the implementation of an accurate network for the prediction of the melting temperature of the mold powders could mean a new perspective for the understanding of such phenomena, which will result in the improvement of the product itself and the casting process

REFERENCES

[1] 1. Kromhout J, Melzer S, Zinngrebe E, Kamperman A, Boom R. Mould Powder Requirements for Highspeed Casting. Steel Research Int. 2008;79:143–148.

[2] Yang J, Chen D, Qin F, Long M, Duan H. Melting and Flowing Behavior of Mold Flux in a Continuous Casting Billet Mold for Ultra-High Speed. Metals. 2020;10:1165.

[3] Marschall, I., Kölbl, N., Harmuth, H. et al. Identification of secondary raw materials in mold powders and their melting behavior. J. Iron Steel Res. Int. 26, 403–411 (2019).

[4] Mills K.C., Däcker CÅ. The Casting Powders Book. [Internet], Springer; 2017 [cited 2021 Jul 28]. 535 p. Available from https://doi.org/10.1007/978-3-319-53616-3_9

[5] MathWorks. Improve Shallow Neural Network Generalization and Avoid Overfitting (R2021a) [Internet]; 2021 [cited 2021 Jul 28]. Available from: https://it.mathworks.com/help/deeplearning/ug/improve-neural-network-generalization-and-avoidoverfitting.html

[6] Brownlee, Jason. What is the Difference Between Test and Validation Datasets? [Internet]; 2017 [updated 2020 Aug 14; cited 2021 Jul 27]. Available from:
https://machinelearningmastery.com/difference-test-validation-datasets/

[7] MathWorks. Choose a Multilayer Neural Network Training Function (R2021a) [Internet] ; 2021 [cited 2021 Jul 28]. Available from: https://it.mathworks.com/help/deeplearning/ug/choose-a-multilayerneural-network-training-function.html

[8] Dinkelbach H,Vitay J, Beuth F, Hamker F. Comparison of GPU- and CPU-implementations of meanfiring rate neural networks on parallel hardware. Network [Internet]. 2012 [cited 2021 Jul 28]; 23.
Available from: https://www.researchgate.net/publication/233392650_Comparison_of_GPU_and_CPU-implementations_of_mean-firing_rate_neural_networks_on_parallel_hardware DOI:
10.3109/0954898X.2012.739292