POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Optimal feature rescaling in machine learning based on neural networks

Laurea Magistrale in
Automation and Control Engineering - Ingegneria dell'Automazione

Author: Federico Maria Vitrò

Advisor: Prof. Lorenzo Mario Fagiano

Co-advisor: Ing. Marco Leonesio

Academic year: 2021-2022

1. Introduction and goals

The advent of Industry 4.0 has radically changed the way companies manufacture, improve and distribute their products. Nowadays the intertwining between automation and data exchange is becoming a common practice for the manufacturers and in a world where resources are getting scarcer day by day we all need to do more with less. In particular, artificial intelligence and machine learning are playing an increasingly central role in this landscape where the physical and virtual worlds are fused together. Similar trends are observed in many other industrial sectors. As of today, neural networks are the basis of many of the most successful algorithms in machine learning. These networks, which try to emulate the human brain, are used to solve many prediction problems, e.g. to predict stock market prices, the effect of an actuation in robotics, etc. and for this reason they have been taken into consideration during this thesis.
Given a regression problem and a Feed Forward Neural Network (FFNN), this thesis focused on increasing the neural network performances via problem variables rescaling. This technique consists of multiplying the dataset features with a set of rescaling parameters which were chosen in an optimal way by solving a global optimization problem. For this reason, this technique was called Optimal Feature Rescaling (OFR). Throughout the thesis a metaheuristic approach (Genetic Algorithm) was used in order to find the best parameters. Typically, the loss function of a global optimization problem is non-convex. Therefore, searching for a better minimum via feature rescaling is an interesting and not trivial topic.
OFR method was then applied in a real scenario to verify its effectiveness: the roundness prediction in a centerless grinding machining operation. Centerless grinding is a machining process characterized by a high degree of difficulty in predicting the quality of the machined parts based on process parameters. The classical machine learning-based approach, even enhanced by physics-based features embedding some apriori knowledge, achieved non-satisfactory prediction performances. As the quality of the worked part mainly depends on the choice of process parameters, e.g. work piece height, feed velocity, etc., which are inhomogeneous in terms of absolute numerical values, an "Optimal Feature

Rescaling" was investigated in order to make the roundness of the worked piece more predictable.

2. Proposed method: Optimal Feature Rescaling

Feature rescaling is one of the most critical parts of the pre-processing phase in machine learning. In fact, it can make the difference between weak and strong machine learning models. In feature rescaling the problem variables are differently transformed depending on the method applied. Usually, these techniques are used to:

- make the features comparable;
- make a few algorithms converge faster, e.g. training processes of neural networks.

In this work, feature rescaling was proposed in order to improve FFNN performances. The idea behind this thesis is to find a set of rescaling parameters $\mathbf{w} = \{w_1, ..., w_M\}$ that can be used to rescale the M input variables of a regression problem. Given a dataset $\mathbf{X} \in \mathbb{R}^{N \times M}$ (where N is the number of samples) and a vector of rescaling parameters $\mathbf{w} \in \mathbb{R}^M$, the rescaled dataset $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times M}$ is obtained as:

$$\tilde{\mathbf{X}} = \mathbf{w}\mathbf{X} =$$

$$= \left[w_1 \cdot \begin{pmatrix} x_{11} \\ \vdots \\ x_{N1} \end{pmatrix}, \quad \cdots, \quad w_M \cdot \begin{pmatrix} x_{N1} \\ \vdots \\ x_{NM} \end{pmatrix} \right]$$

and $\tilde{\boldsymbol{x}}_i = (w_i x_{1i}, ..., w_i x_{Ni})^T$, $i = 1, ..., M$ is the *i-th* rescaled feature of the problem.
In order to improve the prediction capabilities of the model (in this study, a FFNN), a vector of optimal parameters $\mathbf{w}^*$ has to be chosen, hence the name "Optimal Feature Rescaling". Therefore, an optimization problem was defined. Since the analytical form of the cost function is not known in advance a *global optimization* approach was followed.

2.1. Considered cost function

In order to set a global optimization problem a loss function has to be defined. Since the target model for this thesis is a FFNN, the considered objective function takes as inputs the vector of parameters (the optimization problem *decision variables*) and the training/validation sets, performs the feature rescaling using the input parameter vector, then trains a FFNN using the training set and returns the Root Mean Squared Error (RMSE) calculated on the validation data:

$$\text{RMSE}_{validation} =$$
$$= f(\mathbf{w}, \mathbf{X}_{train}, \mathbf{y}_{train}, \mathbf{X}_{val}, \mathbf{y}_{val})$$

In this scenario, the analytical form of the objective function is not known beforehand. But, since the considered cost function, i.e. the training of the neural network, turns out to be *non-convex*, it is possible to find a better minimum point and therefore obtain different performances following different scalings. If the cost function was convex, then its convexity would remain unchanged even through a rescaling, and it would not be possible to reach a better minimum point:

affine input transformation. if $f : \Omega \to \mathbb{R}$ is convex (where $\Omega \subseteq \mathbb{R}^n$), then also $\tilde{f}(\boldsymbol{x}) = f(A\boldsymbol{x} + \mathbf{b})$ is convex on the domain $\tilde{\Omega} = \{\boldsymbol{x} \in \mathbb{R} : A\boldsymbol{x} + b \in \Omega\}$, with $A \in \mathbb{R}^{n \times m}, \mathbf{b} \in \mathbb{R}^m$ [3].

Once the optimization problem is set, a solver should be used in order to get to a solution. Thus, during the thesis a *Genetic Algorithm* (GA) was implemented.

2.2. Genetic Algorithm

Throughout the thesis, the GA was initialized with 20 individuals sampled randomly from a uniform distribution (in log-scale) and it was executed for 100 iterations, i.e. 2020 function evaluations. The GA individuals, i.e. the parameter vectors that compose the initial population, were sampled from a uniform distribution between $[-3, 3]$ in logarithmic scale. They were then converted in decimal scale, i.e. $w \in [-10^3, 10^3]$, before being applied to the dataset features.

2.3. Fixing the FFNN input layer weights

An interesting aspect of the FFNN is that, in the first layer, it implicitly performs an input rescaling, multiplying the dataset features by the input layer weights [1]. If a_j^l is the j-th input to the layer l, and β_{jk}^l is the weight from the k-th

neuron in the $(l-1)$-th layer to the j-th neuron in the l-th layer, such that:

$$a_j^l = f(\sum_k \beta_{jk}^l a_j^{l-1} + b_j^l) \quad (1)$$

where b_j^l is the j-th bias to the layer l and $f(\cdot)$ is the activation function of the considered layer.
Extending this concept by exploiting the OFR technique, it can be possible to set the first layer weights in order to optimally rescale the neural network inputs. If w_k is the k-th rescaling parameter, it can be incorporated into the k-th weight of the first layer in order to enable the network to perform input OFR without any data preprocessing. In this case the weight from the k-th neuron in the first layer to the j-th neuron in the second layer, becomes:

$$\tilde{\beta}_{jk}^1 := \beta_{jk}^1 w_k$$

while the Equation 1 changes into:

$$a_j^l = f(\sum_k \tilde{\beta}_{jk}^1 a_j^0 + b_j^1) =$$

$$= f(\sum_k \beta_{jk}^1 w_k a_j^0 + b_j^1)$$

3. Case study

In order to test the OFR method, it was applied in a real scenario: *roundness prediction for a centerless grinding machining process*. Centerless grinding is a machining operation where material is removed from the workpiece using an abrasive wheel. The work part is not clamped during the machining process, making it prone to inherent instability, and therefore the overall quality of the final piece is difficult to predict on the basis of the chosen process parameters.

3.1. Data preparation

In centerless grinding, each operation can be defined by 12 parameters which are listed in Table 1.

x_0: w.p. height	x_6: grin. wheel diam.
x_1: blade angle	x_7: contr. wheel diam.
x_2: feed vel.	x_8: contr. wheel vel.
x_3: tot. diam. removal	x_9: grin. spec. energy
x_4: w.p. length	x_{10}: edge force comp.
x_5: w.p. diameter	x_{11}: grit stiff.

Table 1: Centerless grinding process parameters [4].

There are 9 other parameters that are characteristic of the grinding machine and not of the process. For this reason, they will be considered as fixed for this study.
The model of the centerless grinding process, the so-called *high fidelity model* [2], was then used in order to obtain a dataset for the regression problem: $\mathcal{D} := \{y_i, \mathbf{x}_i\}, i = 1, 2, ..., N_s$, where $y_i \in \mathbb{R}^+$ is the continuous value of the roundness of the final piece that measures the process performances of the i-th sampled grinding operation (i.e. the target variable), $\mathbf{x}_i \in \mathbb{R}^{12}$ is the corresponding vector of process parameters (i.e. the input variables), and $N_s = 4069$ is the number of samples.
The feature set (that consisted of the 12 aforementioned parameters) was further expanded by using the output of the so-called *low fidelity model* [4], which is a simplified version of the (more complicated) high fidelity one.
The dataset thus obtained was divided in 3 subsets by applying a *hold-out procedure*: training (70%), validation (15%) and test (15%) sets. In order to reserve the majority of the data for the FFNN training, only a small portion of them was held out for the test phase. For this reason, during the thesis a *10-folds cross-validation procedure* was used in order to test the performances of the FFNN using as a metric the *Coefficient of determination* (R^2). Because of the dimensions of the test set, it was only used in cases where the cross-validation procedure was not a viable option.

3.2. Test 1: OFR application

Once the dataset was defined, the OFR approach was tested by training a FFNN on an optimally rescaled set of data. The tested FFNN was composed by:

- an input layer with 128 neurons and *relu* activation function;
- 3 hidden layers with 256 neurons each and *relu* activation functions;
- an output layer with a single neuron with *linear* activation function.

Mean Absolute Error (MAE) was used as loss metric for the training and validation phases, while an optimizer implementing the *Adam* algorithm was used for the optimization phase. In the end, 500 *epochs* were considered during the training phase and the neural network was

trained with all the training data in each one of them.
The FFNN in exam was trained on two different datasets and the performances thus obtained were used in order to define the baselines for the test. The neural network was tested on:

1. the raw data, i.e. data without feature rescaling (Base1);
2. the standardized data (Base2).

The two baselines were then compared to the performances obtained training the FFNN on:

1. the data rescaled by applying OFR on the raw ones (OFR1);
2. the data rescaled by applying OFR on the standardized ones (OFR2).

The obtained results are reported in Table 2.

Test	R^2 (Train)	CV Mean	CV Std
t1 OFR2	0.844512	0.561060	0.117817
t1 OFR1	0.822084	0.475483	0.092552
t1 Base2	0.990043	0.369556	0.337064
t1 Base1	0.515098	0.285996	0.168344

Table 2: Results of test 1 (t1), ordered w.r.t. "CV Mean".

The performance improvement obtained by applying OFR to a standardized dataset is far greater than all of the other cases, guaranteeing an increment of 95.69% compared to using of raw data, of 51.44% compared to using the standardized data, and 17.70% compared to the performances obtained by applying OFR on the raw data, demonstrating the fact that a previous standardization can be useful even when an OFR procedure is carried out. Although these results are promising, the test makes it clear that OFR cannot compensate the effect of a simple standardization, making the latter a desired step.
Nevertheless, analyzing the discrepancy between the performances on the train set and the ones obtained with cross-validation, it can be deduced that the model is affected by the overfitting problem, so the *Early Stopping* (ES) technique was applied in test 2.

3.3. Test 2: OFR application with ES

In order to counteract the overfitting problem, an ES approach was followed. The number of epochs for the FFNN training was increased to 10000, while the ES patience was set to 100. The goal of this choice is to make sure that the ending of the training phase is reached before the end of the epochs. In this way, it is guaranteed that the FFNN training stops when the validation error begins to rise.
The results obtained in this case are reported in Table 3.

Test	R^2 (Train)	CV Mean	CV Std
t2 OFR2	0.818443	0.527066	0.137574
t2 OFR1	0.888022	0.488795	0.111678
t2 Base2	0.808376	0.456640	0.116470
t2 Base1	0.655322	0.357649	0.080880

Table 3: Results of test 2 (t2), ordered w.r.t. "CV Mean".

The improvement achieved by using the OFR technique can be seen from Table 3. However, the performances of the FFNN are still rather poor despite the implementation of ES. This probably depends on several factors:

1. the FFNN is too complex for the considered problem;
2. the parameters that were set for the ES were too permissive;
3. the numerosity of the data;
4. the noise on the data.

The prediction performances of the network would surely benefit from a *model selection* procedure, where several models are tested on a validation set in order to choose the best among them.

3.4. Test 3: OFR effects on a simplified FFNN with ES

In order to test the efficacy of the OFR method with a model not subjected to the overfitting problem a simpler neural network was tested. The FFNN described in Section 3.2 was simplified by drastically reducing the number of layers and neurons. The resulting model is composed by:

- an input layer with 13 neurons and *relu* activation function;
- 1 hidden layer with 100 neurons and *relu* activation function;

- an output layer with a single neuron with *linear* activation function.

Furthermore, the objective function for the optimization program was changed, reflecting the new considered model.
The ES patience was incremented to 200, in order to gain more training time for the FFNN.
The results obtained in this case are reported in Table 4.

Test	R^2 (Train)	CV Mean	CV Std
t3 OFR2	0.615985	0.595238	0.098966
t3 Base2	0.629364	0.540791	0.147672
t3 OFR1	0.674978	0.533133	0.147843
t3 Base1	0.335624	0.237633	0.464777

Table 4: Results of test 3 (t3), ordered w.r.t. "CV Mean".

As it can be seen from Table 4 the networks trained on the optimally rescaled datasets achieve better performances compared to the baselines. Furthermore, the overfitting problem was almost completely solved, significantly reducing the gap between the training error and the cross-validated performances. Although the performances obtained by applying OFR to the standardized dataset were better compared to the baselines, it is evident that even with the simplification of the neural network the performances achieved by applying a common standardization to the data are almost equal (greater in this case) to the ones obtained by optimally rescaling the raw data. Therefore, the usefulness of a standardization procedure prior to the application of the OFR technique has been confirmed.

3.5. Final comparisons

In conclusion, the best performances obtained in the three tests are reported in Table 5.

Test	R^2 (Train)	CV Mean	CV Std
t3 OFR2	0.615985	0.595238	0.098966
t1 OFR2	0.808739	0.559659	0.133694
t2 OFR2	0.818443	0.527066	0.137574

Table 5: Final comparisons.

From Table 5 it can be seen that the simplified neural network described in Section 3.4 performs better than the more complex ones and, at the same time, it is not affected by the overfitting problem. Performing a model selection procedure before the application of the OFR technique, as demonstrated by the results obtained with the simplified network, would guarantee even better results.

3.6. Fixing the input layer weights of the FFNN

By exploiting the feature of neural networks to apply the rescaling of the variables within the input level, it was possible to test the particular application of the OFR technique described in Section 2.3.
Table 6 shows the performances obtained by FFNN (Section 3.2) by carrying out the optimal rescaling outside and inside the network, i.e. modifying the first level of the neural network.

Test	RMSE (Test)	R^2 (Train)	R^2 (Test)
Inside	5.092644	0.82803	0.247903
Outside	5.092644	0.82803	0.247903

Table 6: Performances obtained by fixing with OFR the FFNN input layer.

Observing the Table 6, it can be seen that the performances obtained by applying the optimal rescaling outside the neural network coincide with those obtained by using the unscaled data, but by modifying the weights of the first level of the network.
Thanks to the non-convexity of the training problem it is always possible to pass from the neural network original weights to the ones optimally selected. For this reason, the global optimization can be used to "globalize" the training phase of the FFNN and, in particular, the initial values of its first layer.

4. Conclusions

In this thesis, a new method to improve the performances of a Feed Forward Neural Network in regression problems was defined and tested. This method was called Optimal Feature Rescaling (OFR) and consists in multiplying the variables of the problem, i.e. the features of the

input dataset, by an optimal parameter set obtained by solving a global optimization problem. Since the training of the neural network is non-convex, the performances of the network should have improved, demonstrating the effectiveness of this approach. The Optimal Features Rescaling approach was then applied to a real scenario to test its efficacy: the roundness prediction for a centerless grinding machining process. Because the workpiece is not fixed during the grinding process, the roundness prediction is a difficult task and represents the perfect scenario on which to test the OFR technique. The results demonstrated the OFR efficacy, highlighting how, through a model selection process, it is possible to achieve satisfactory results. Moreover, tests have shown that it is possible to apply the OFR within the network, incorporating the optimal parameters in the first level weights of the neural network.
Several aspects of the method can be further analyzed to understand its efficacy in the machine learning and deep learning fields, e.g. the OFR method could be tested on more complex problems (i.e. with more variables), on techniques with less computational requirements (like k-Nearest Neighbor), it can be tested after an accurate model selection phase, etc.

References

[1] Charu C. Aggarwal. *Neural Network and Deep Learning*. Springer International Publishing AG, part of Springer Nature, 2018.

[2] Qi Cui, Kei Cheng, and Hui Ding. An analytical investigation on the workpiece roundness generation and its perfection strategies in centreless grinding. 2014.

[3] Lorenzo Fagiano. Constrained numerical optimization for estimation and control. 2020.

[4] Marco Leonesio and Lorenzo Fagiano. A semi-supervised physics-informed classifier for centerless grinding operations. 2022.