MW

Politecnico di Torino

1859

mODa14

Model-Oriented Data Analysis and Optimum Design

Davide Ronco

R&D and Innovation manager at MW Italy

PhD student at Politecnico of Torino

June 2026

Topic

- Current wheel optimization process
- Deep learning: possible integration in the optimization process
- How it works
- Conclusion

Topic

Current wheel optimization process

Deep learning: possible integration in the optimization process

How it works

Conclusion

Wheel: safety component

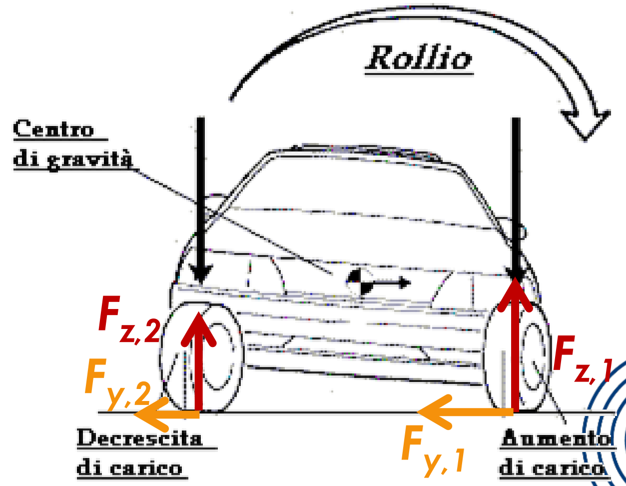
- Mature component
- Established geometries and technologies
- **Key safety component**

- Difficult to innovate
- Components already strongly optimized
- Designed following the Safe design approach

Fatigue

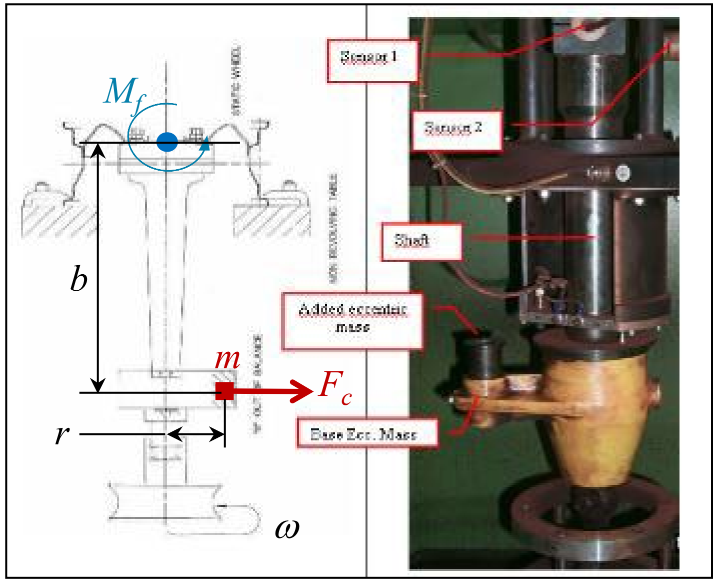
The purpose of the wheel is to sustain the weight of the vehicle and to connect it to the road via the tyre. The fatigue resistance is the key parameter to take into account in designing the wheel.

External loads radial (due to the car weight) and axial (due to the curve behaviour) coupled with rotation creates cyclic rotary loads

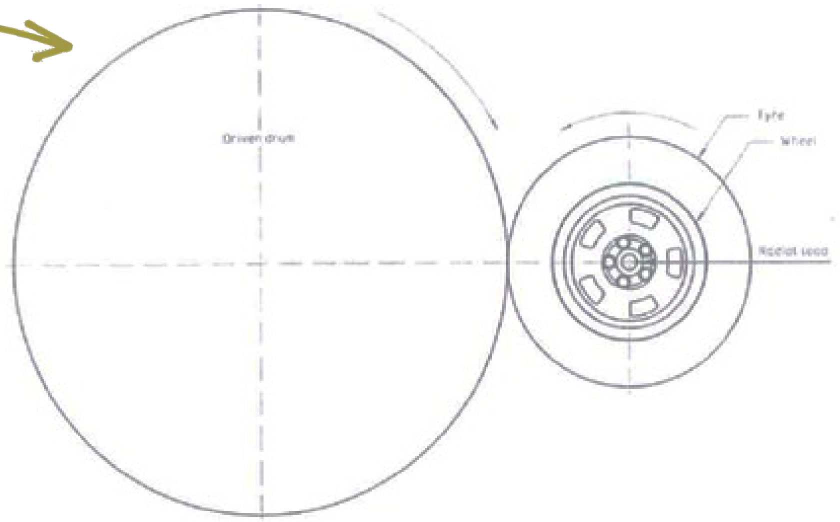


Fatigue test and requirement

Rotary bending test → Disc



Rim rolling test → Rim & welding



Parameter for test:

- Bending moment (Mb)
- Number of cycles (N)

Parameter for test:

- Radial force (Fr)
- Axial force (Fa)
- Tyre pressure (p)
- Number of cycles (N)

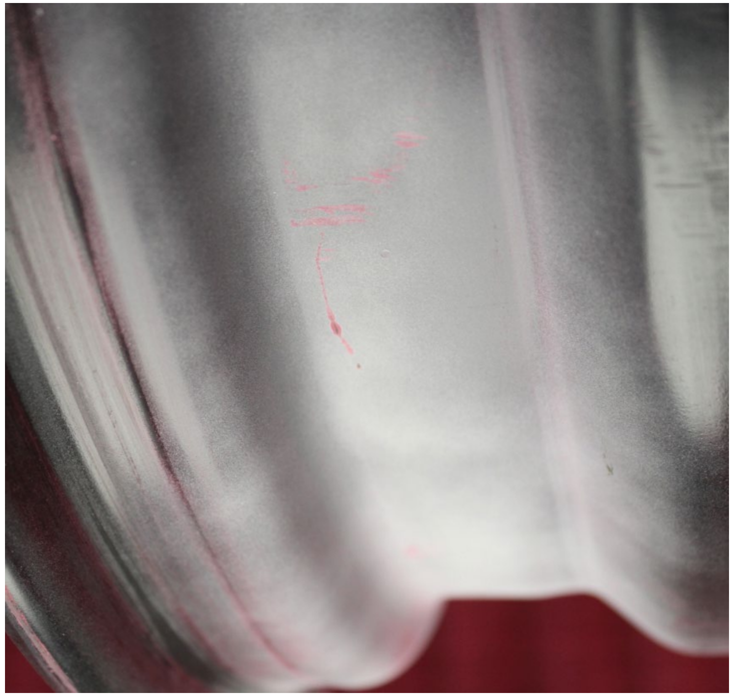
Fatigue test and requirement

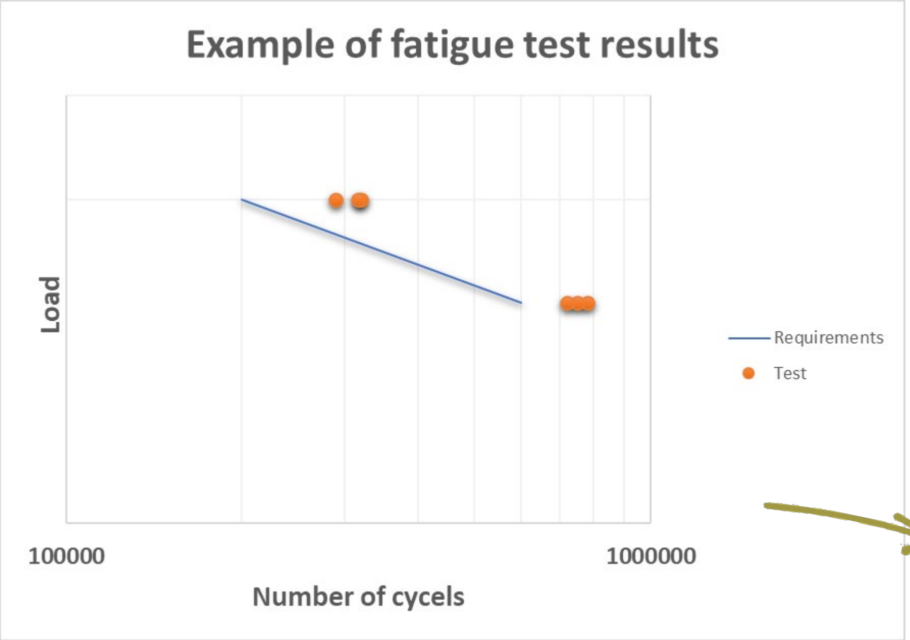
Rotary bending test → Disc

On the rotary bending test typically the crack are on the **disc**

Rim rolling test → Rim & welding

On the **rim rolling test** typically the crack are on the **rim** or on **welding**





The tests results should be higher the requirements imposed by customer.

Main goal: Predict the fatigue behaviour

Fatigue is the main aspect for a steel wheel.

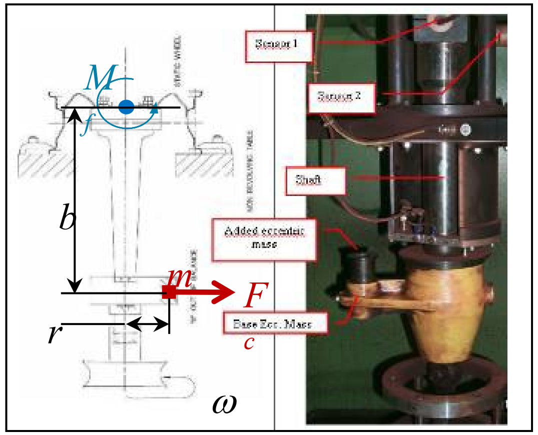
Product **not** suitable for **fast prototyping**.

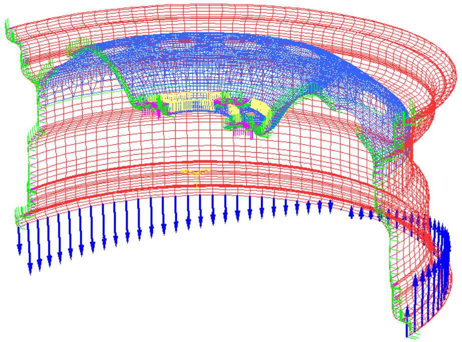
We need to predict the fatigue behaviour of steel wheel.

FEM simulation

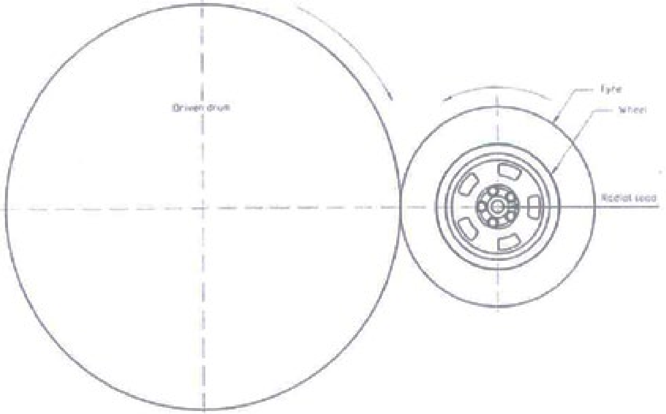
FEM simulation

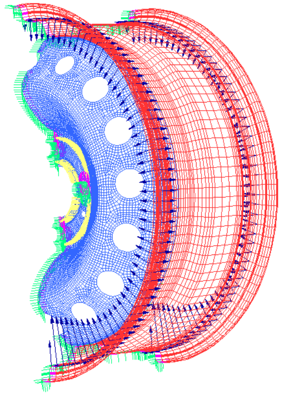
Rotary bending test

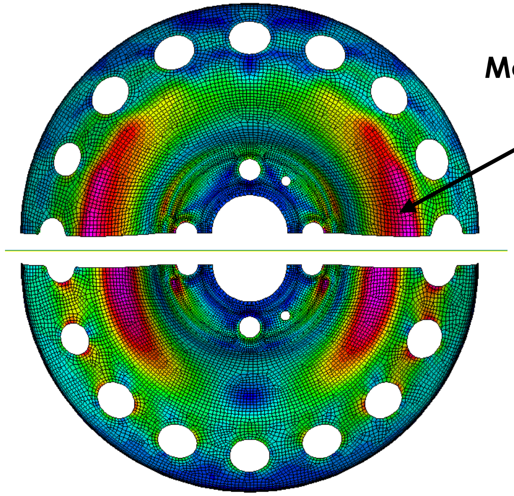




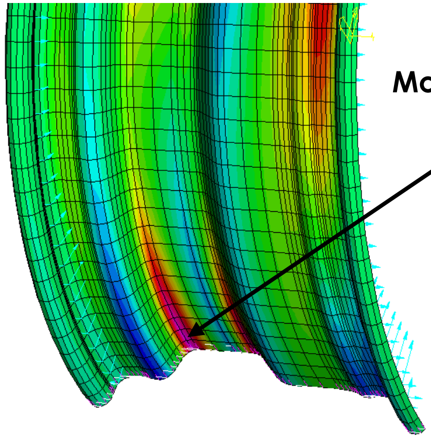
Rim rolling test



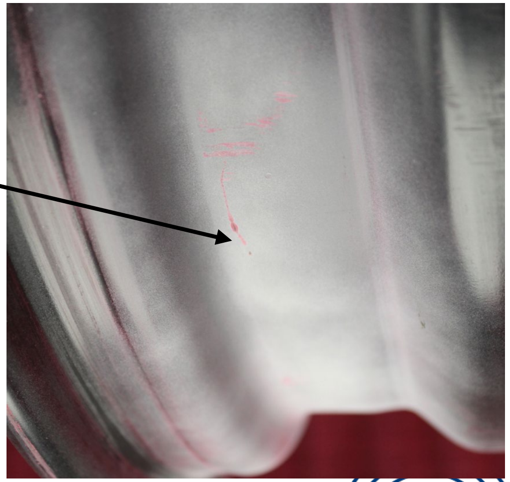




Most stressed area



Most stressed area



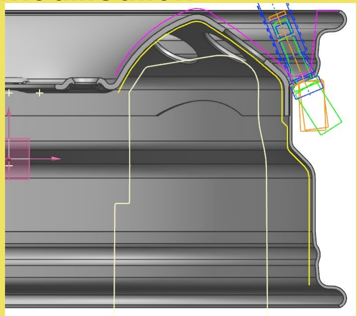
Current wheel optimization process

Optimization process: several loops (above 50 iterations) to obtain the best solution possible

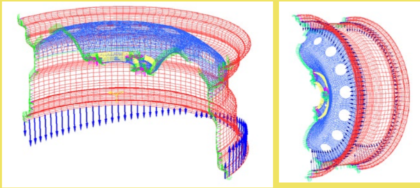
Optimization process

Input data from customer:
- Dimensions
- Target weight
- Interface (brakes, valve, wheel trim…)
- Fatigue requirements
- Stiffness requirements (static, dynamic, NVH)
- Style
- …

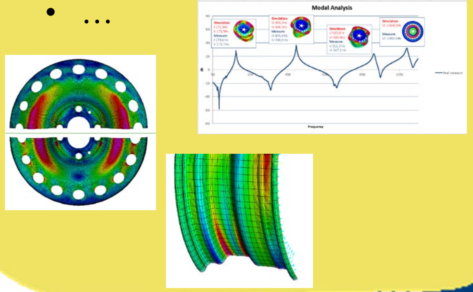
3D model generation / modification



FEM setup
- Mesh generation from 3D
- Load application
- Constraints
- Material properties



FEM simulation
- Fatigue tests
- Dynamic / stiffness
- …



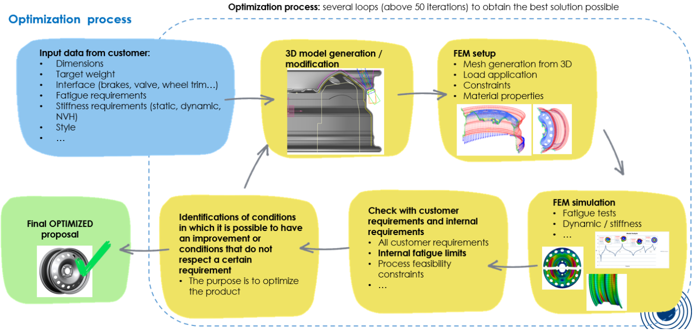
Check with customer requirements and internal requirements
- All customer requirements
- **Internal fatigue limits**
- Process feasibility constraints
- …

Identifications of conditions in which it is possible to have an improvement or conditions that do not respect a certain requirement
- The purpose is to optimize the product

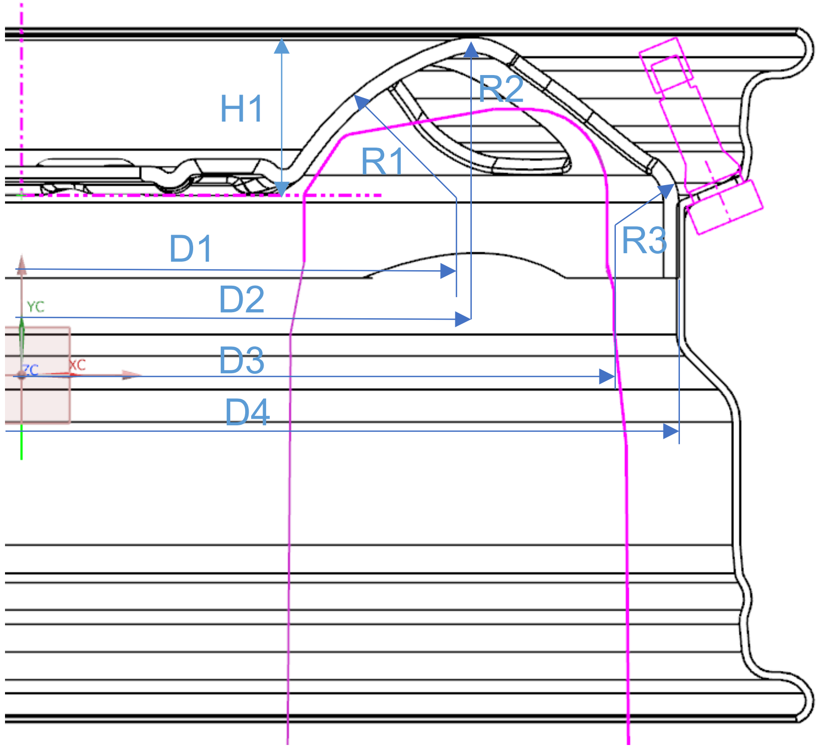
Final OPTIMIZED proposal

Current wheel optimization process

Speed-up the optimization process



In the **Optimization process**, usually are modified **few parameters per time** (sometimes only 1 parameter), in order to avoid that the impact of 1 parameter is hidden by the modification of other parameters.
The parameters are modified on 3D CAD evaluating their impact on interfaces (brakes, valve, wheel trim…).
The calculation (FEM) is performed with another software that evaluate the impact of parameters on fatigue.



- It is possible to have the full focus of the impact of the single parameter on fatigue behaviour (stress)
- It is possible to have the clear view of the direction of optimization

- Time (from 1 to 5 days to find the optimized solution)
- Possibility to investigate few parameters per time
- Difficulty to reach the most optimized solution
- Human error

Topic

Current wheel optimization process

Deep learning: possible integration in the optimization process

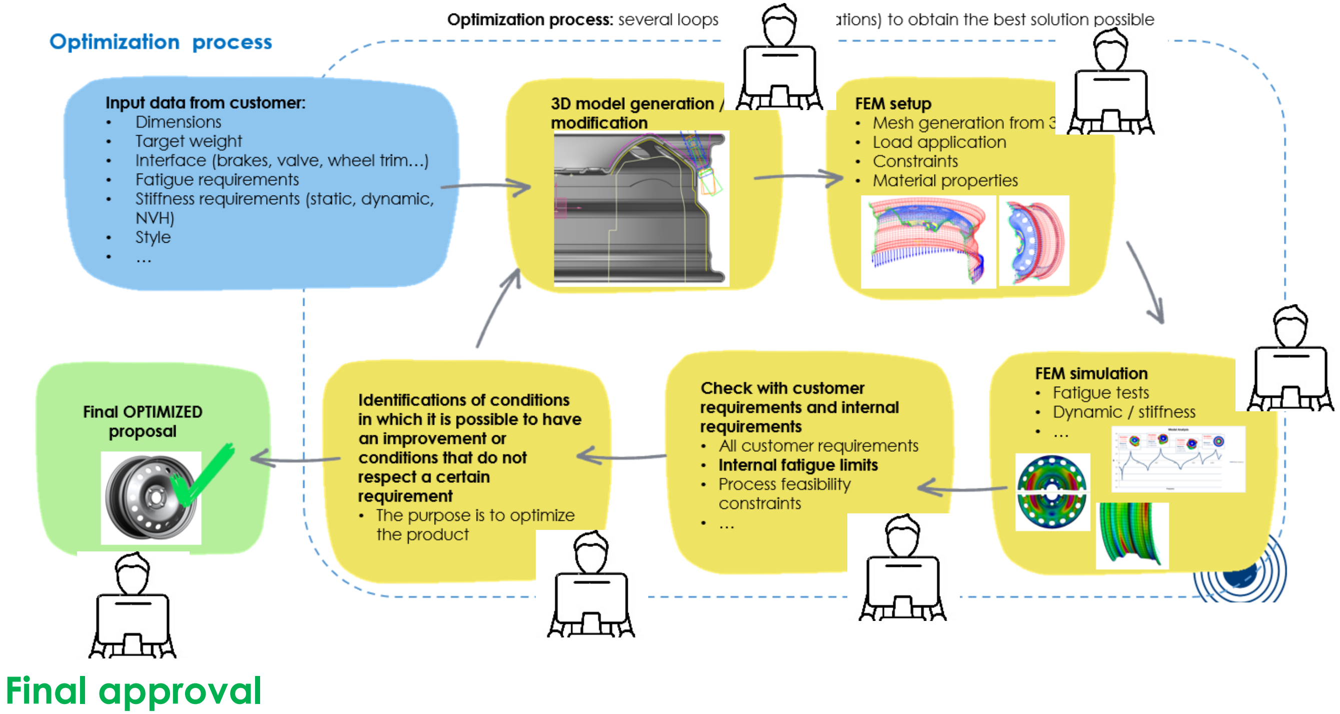
How it works

Conclusion

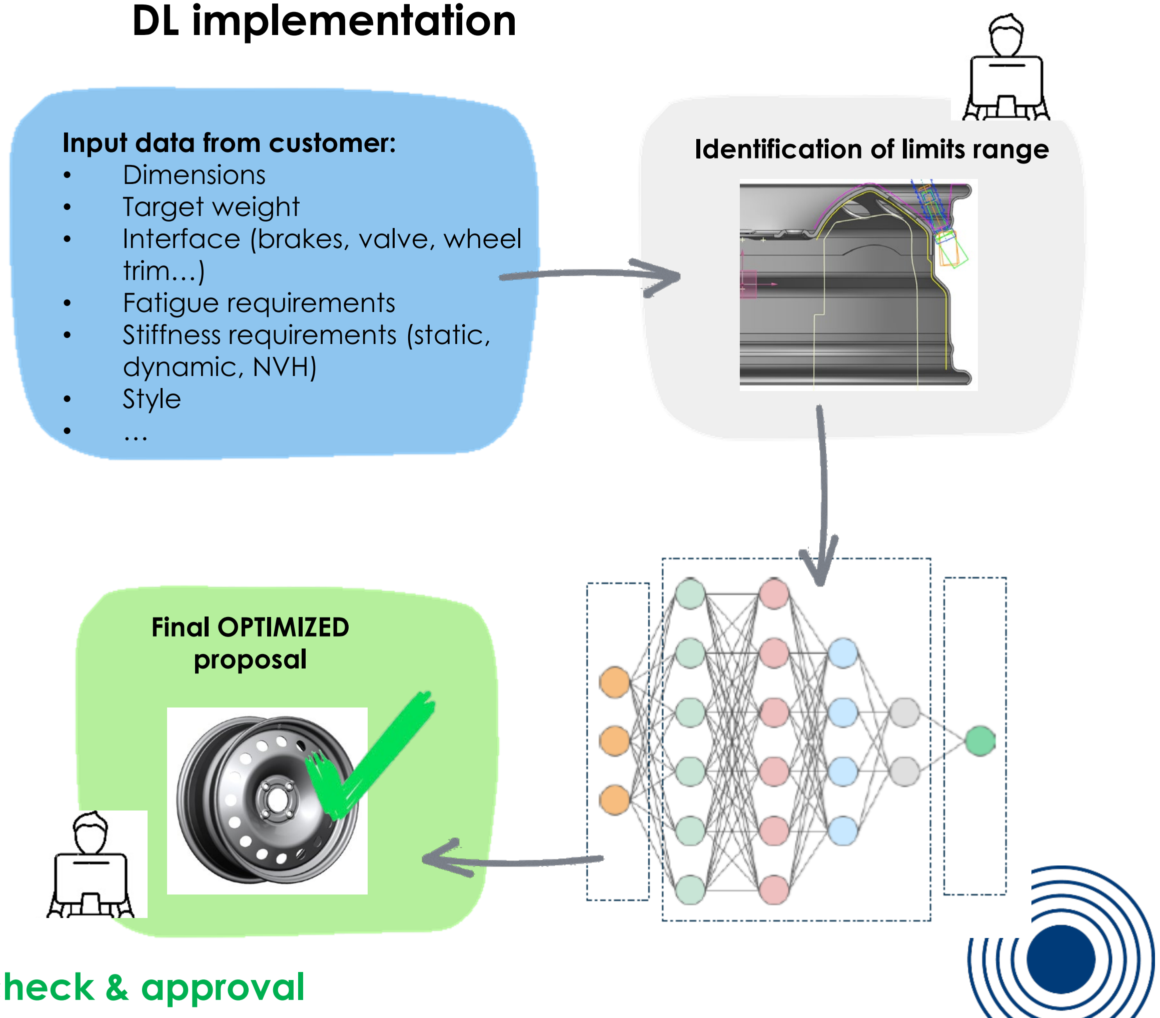
Deep learning: possible integration in the optimization process

Speed-up the optimization process

Standard optimization process



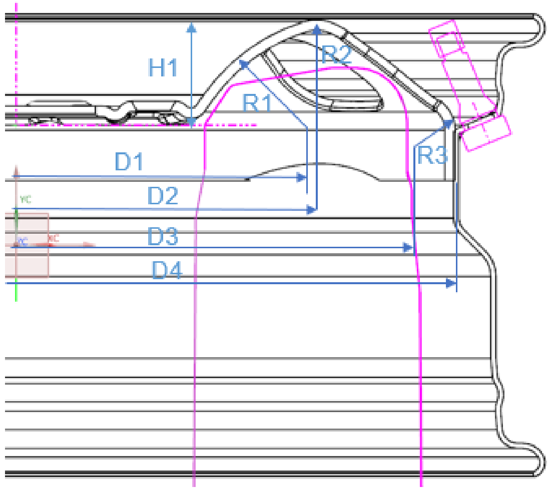
DL implementation

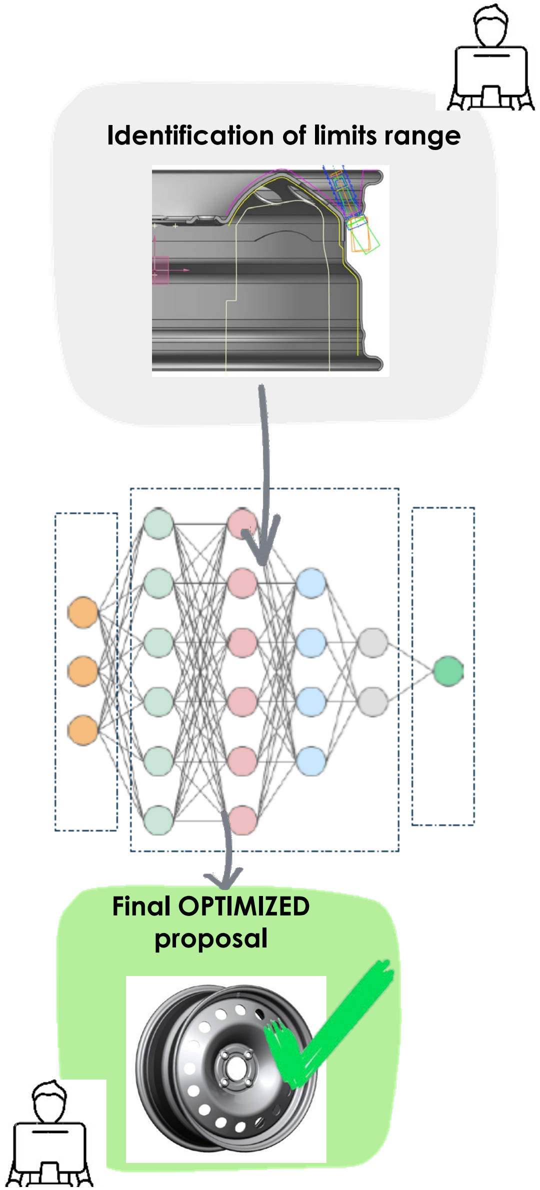


Deep learning: possible integration in the optimization process

Speed-up the optimization process

Prediction





The Operator, will provide only the range for each parameters, that take into account the interface.
E.g.:
H1= 25 – 30 mm
R1= 10 – 20 mm
….

The optimized definition/s will be identified, exploring all the possible combinations (minimizing the stress) between parameters inside the range defined by the Operator.

The Operator will check the optimized proposal and gives the approval.

Deep learning: possible integration in the optimization process

Training (Supervised Learning) → **Prevision**

Training (Supervised Learning):
- Input – output definition
- Dataset split: Training set / Test set
- Normalization
- Hyper parameter optimization FFD
- Neural network definition
- Training
- Results

Prevision:
- Optimization
- Results cpmparison

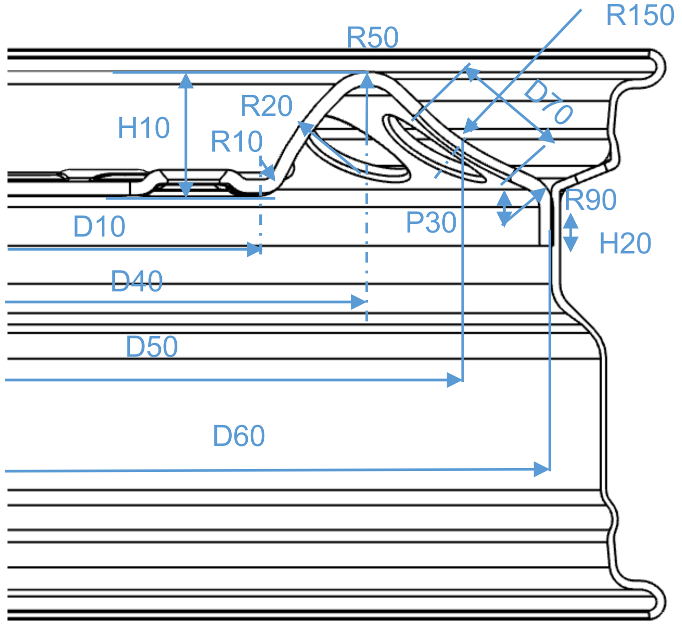
Deep learning: possible integration in the optimization process

Input – output definition

Input

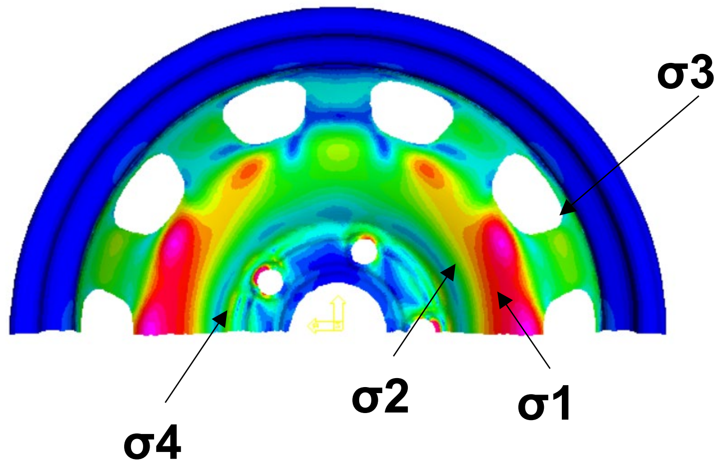
All the characteristic dimensions of the model.
At this stage of the project, the main dimension able to characterize the disc are taken into account **(15 inputs)**

- Disc thickness
- Vent hole number



Output

- Stress in the critical areas of the wheel
  - σ1, MPa: Nose area
  - σ2, MPa: Voile area
  - σ3, MPa: Vent hole area
  - σ4, MPa: Central part area
- Disc net weight, kg
- Raw material surface, mm^2

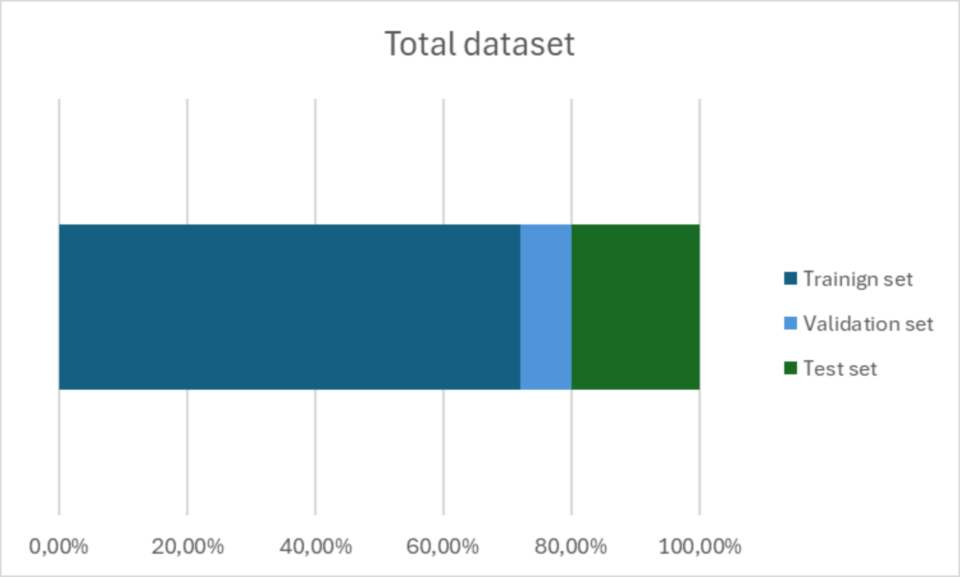


Deep learning: possible integration in the optimization process

Dataset split: Training set – Test set

The dataset (for the moment 258 samples) was splitted in:

- 80% Training set
    - 90% Pure Training
    - 10% Validation set
- 20% Test set



Normalization

The training set was normalized between 0 to1.
This permit to improve and speed up the model convergence during the training phase.
The input and output have very different scale between each others.

$$x_{norm} = (x - min(x))/(max(x) - min(x))$$

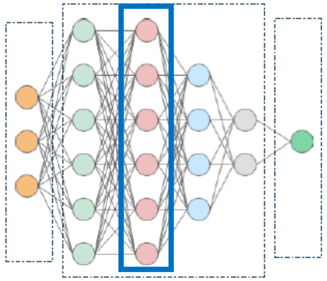
Also the standardization was tried, but the performances were better with the normalization.

Deep learning: possible integration in the optimization process

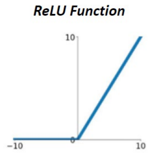
Hyper parameter optimization

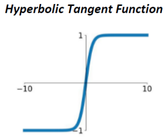
Through a Full Factorial Desing (FFD), implemented in the algorithm, it was researched the better configuration of Hyper parameters.

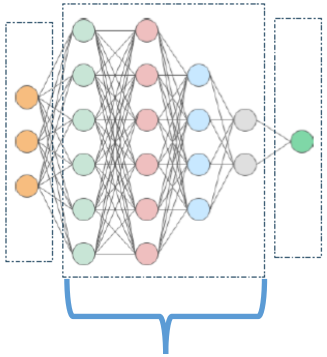
Neurons number	Batch size	Learnign rate	Activation function	Hydden layer number	Epoche	Optimizer
• Number of neurons for each hidden layer between input and output layer	• Number of samples to be used to updated the weights during each iteration	• This parameter manages the speed of the weights updating in each iteration	• ReLU and Tanh	• Number of level between Input layer and Output layer	• Represent the iteration between all the training set. It is important to avoid the Overfitting or Underfitting phenomena	• It is the algorithm that updates the weights and the bias with the scope to reduce the loss function

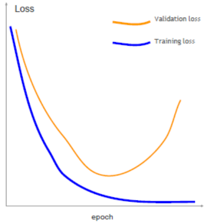


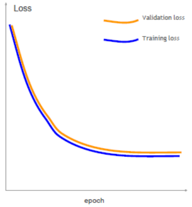
$$\theta := \theta - \eta \cdot \frac{1}{m}\sum_{i=1}^{m}\nabla_\theta \mathcal{L}(x^{(i)}, y^{(i)}; \theta)$$









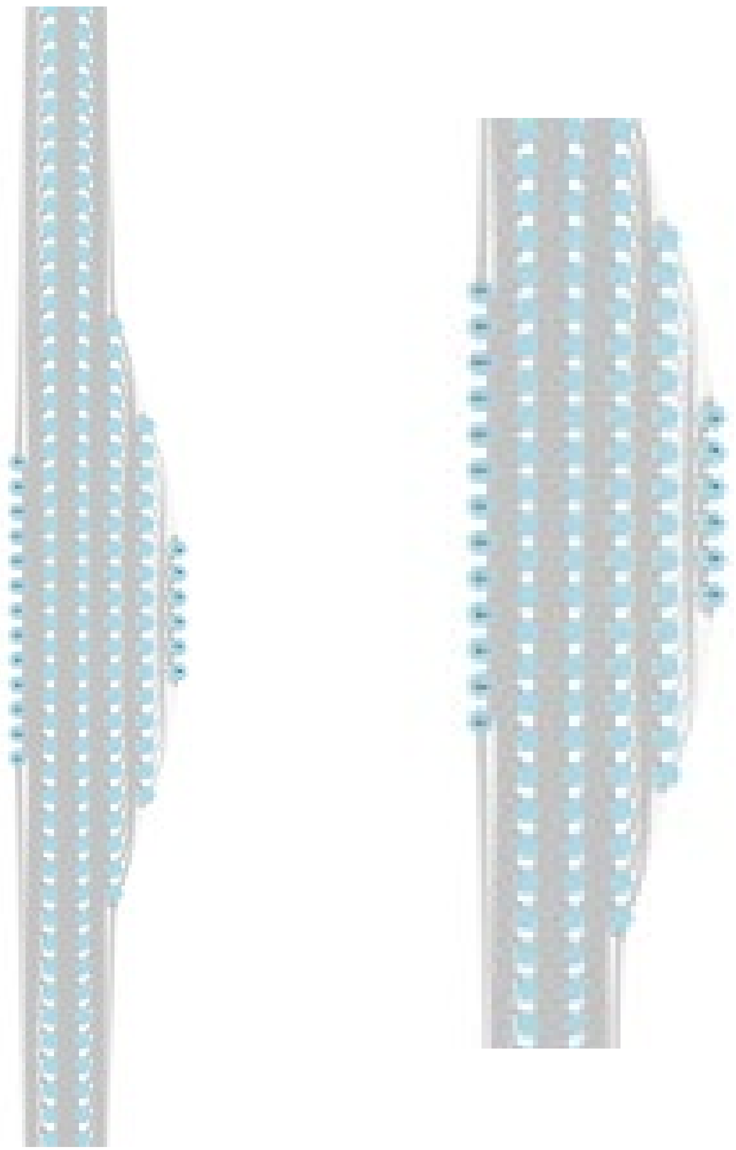


Deep learning: possible integration in the optimization process

Neural network definition

Iper-paramter	Value
Input layer	15 neurons → Input parameters
1° Hidden layer	64 neurons
2° Hidden layer	16 neurons
3° Hidden layer	16 neurons
4° Hidden layer	8 neurons
Output layer	6 neurons → Output
Learning rate	0,005
Batch size	4
Activation function	ReLu
Epochs	500
Optimizer	Adam

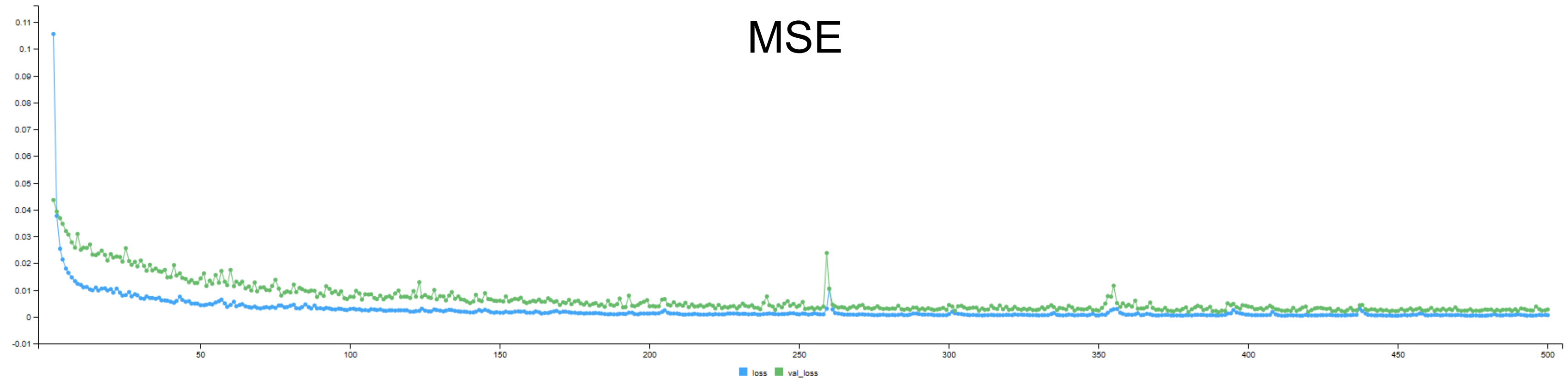
Training time: 30 min



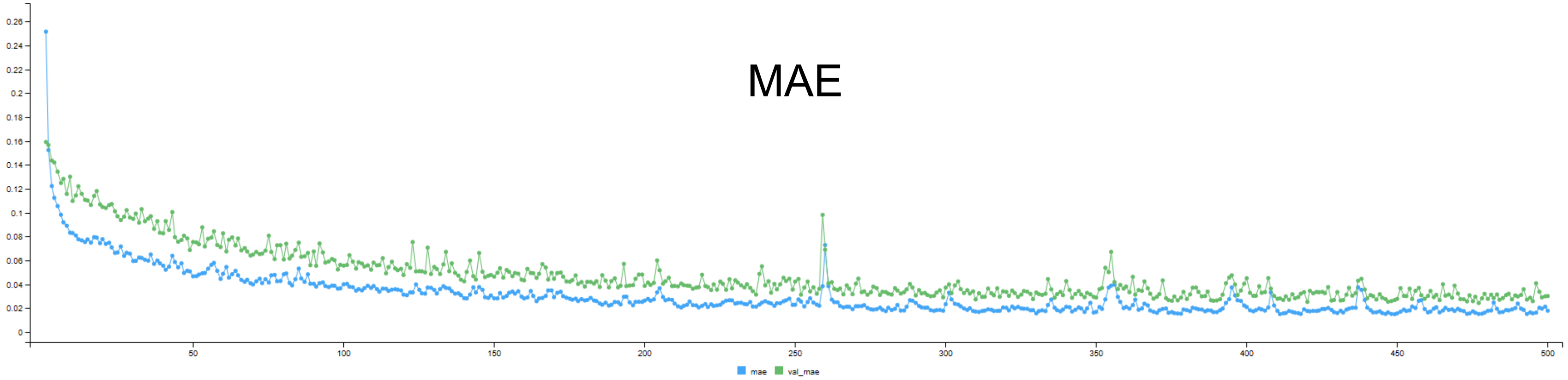
Deep learning: possible integration in the optimization process

Trining

In the graph next to it you can see the MSE loss function, and also the MAE (Mean Absolute Error), for training data (blue) and validation data (green).
Both converge to 0, a symptom of a good definition of the model (no Underfitting, no Overfitting).

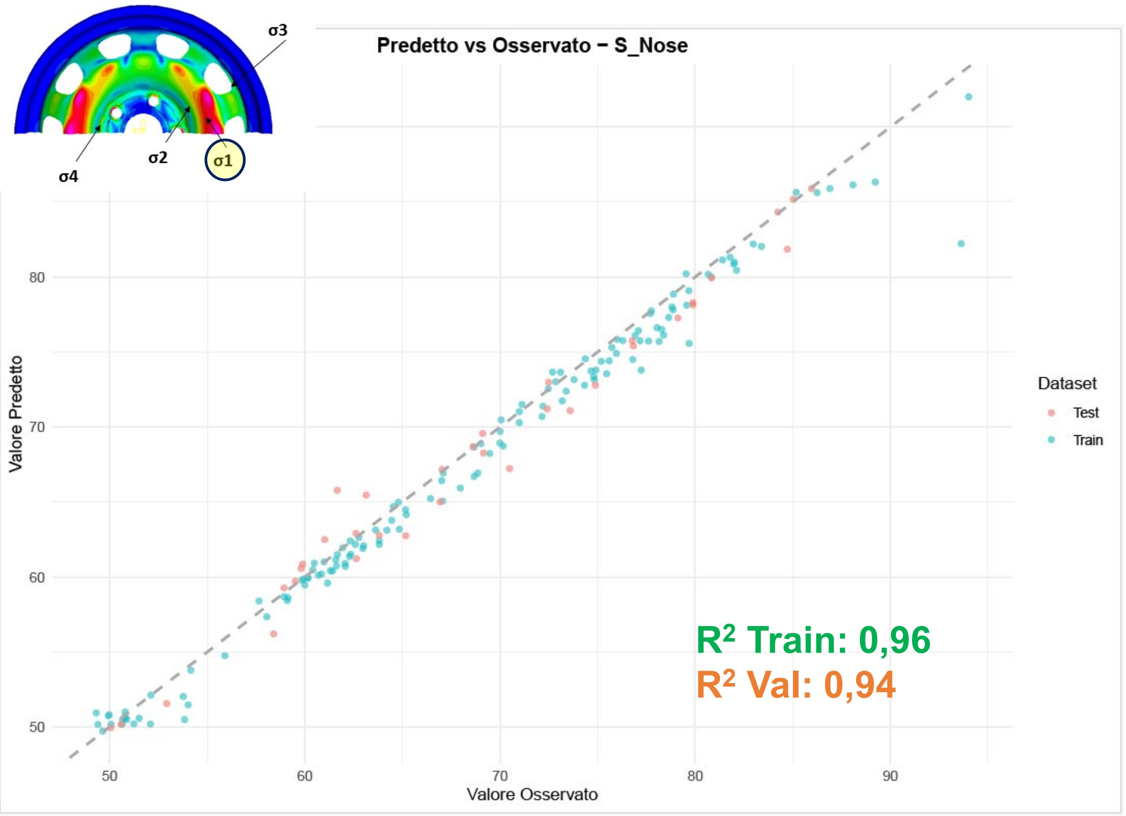


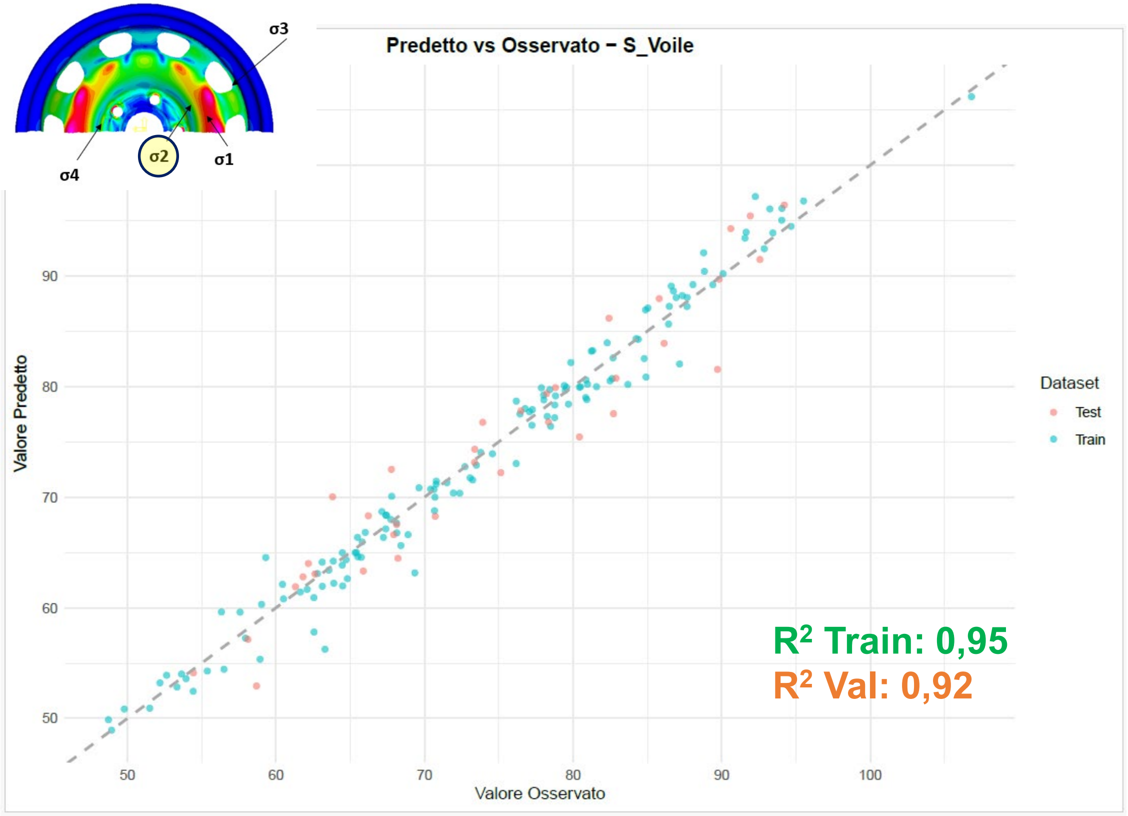
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_{i,real} - Y_{i,pred}\right)^2$$



Deep learning: possible integration in the optimization process

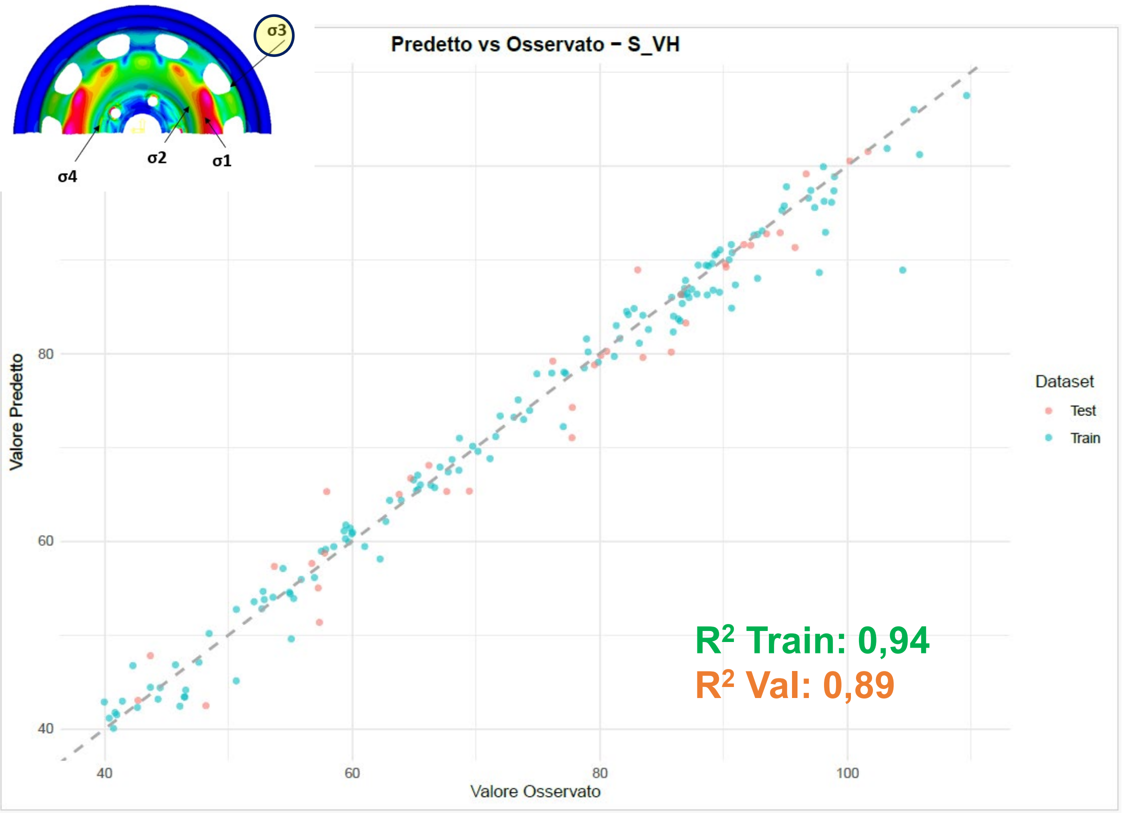
Results



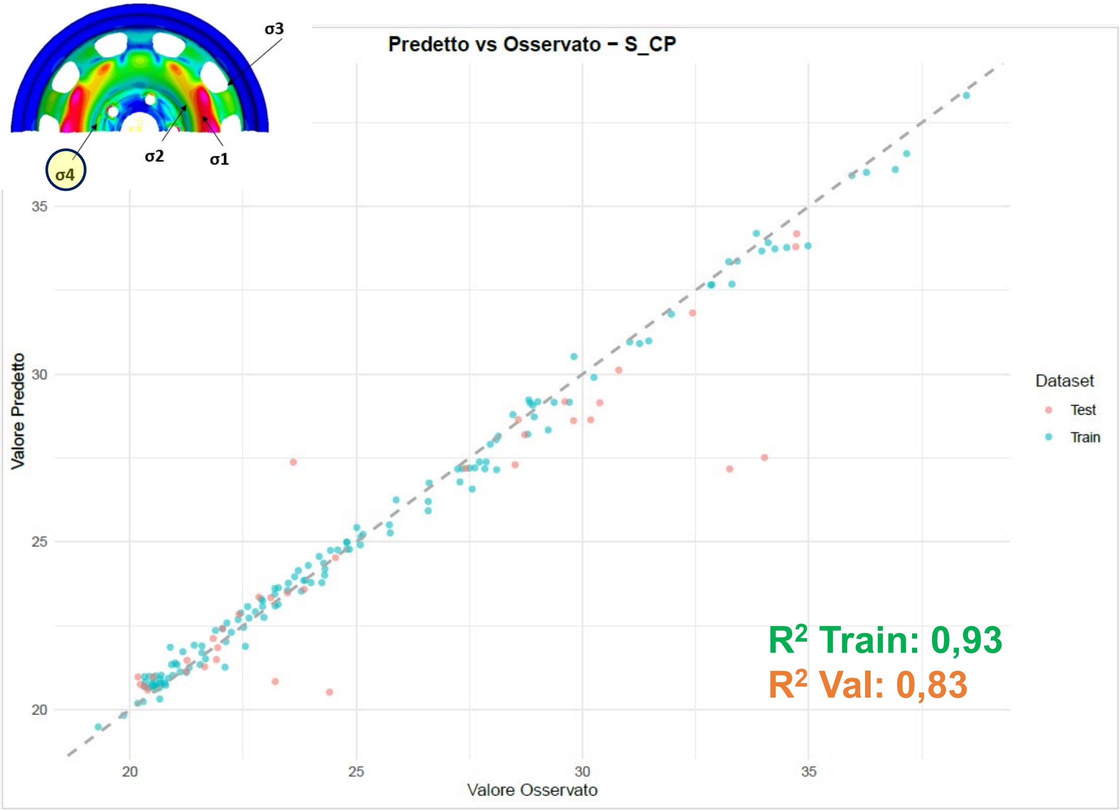


Deep learning: possible integration in the optimization process

Results



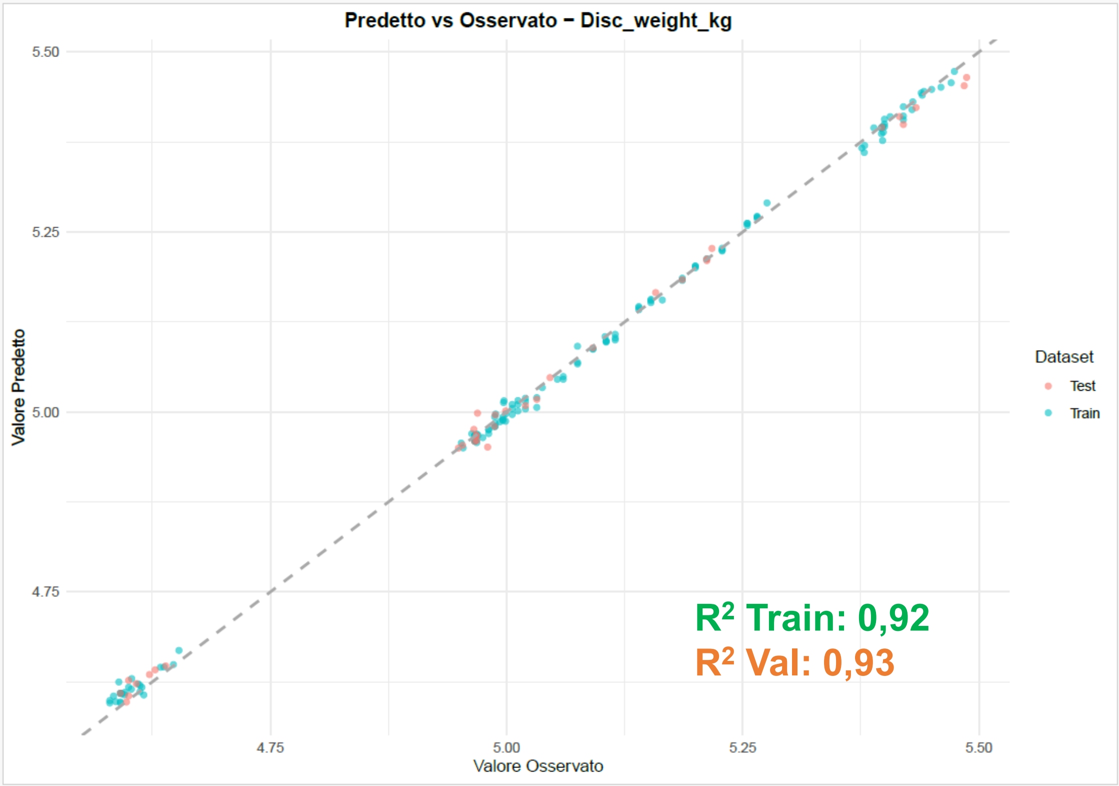
R^2 Train: 0,94
R^2 Val: 0,89

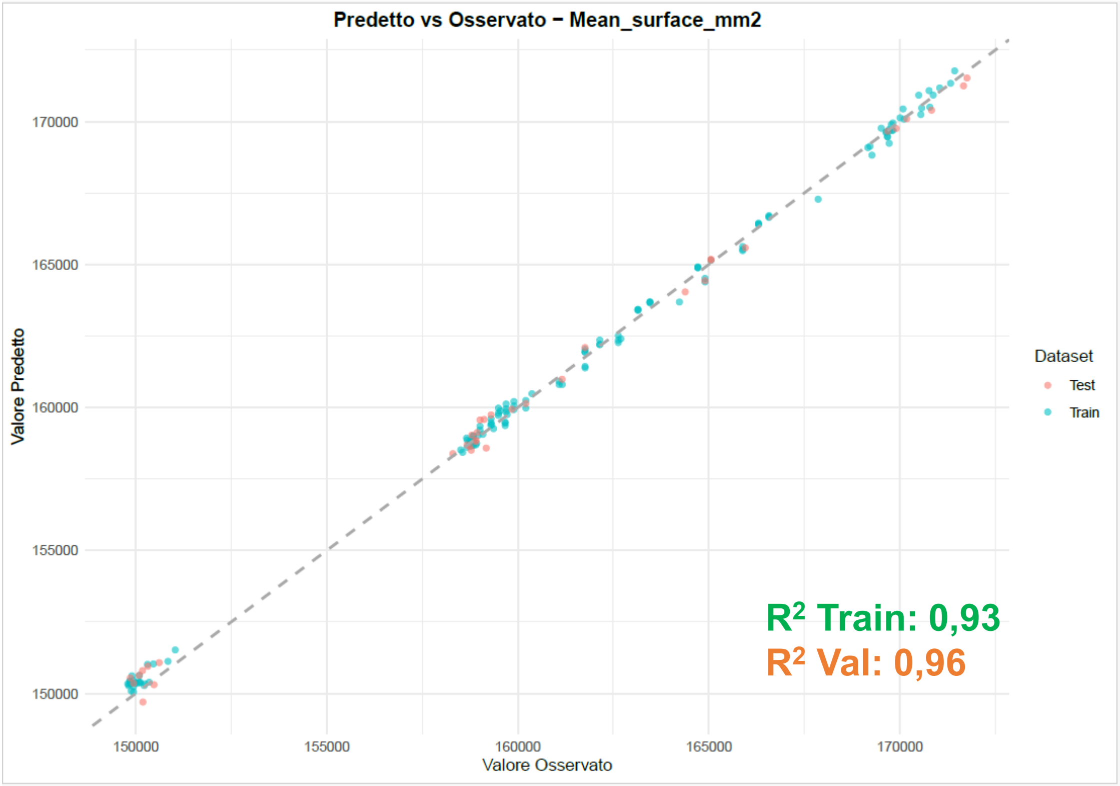


R^2 Train: 0,93
R^2 Val: 0,83

Deep learning: possible integration in the optimization process

Results





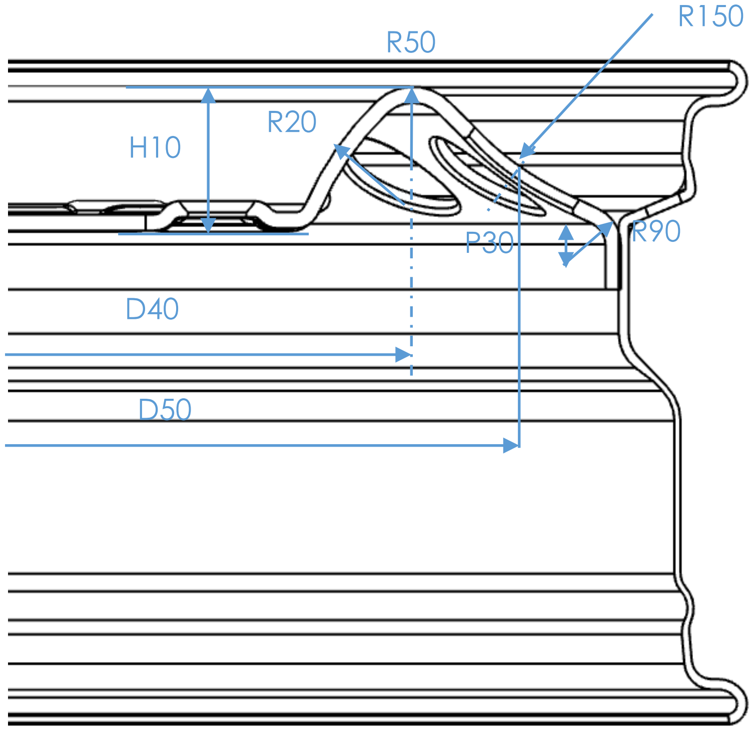
Deep learning: possible integration in the optimization process

Prevision: Steel wheel design optimization

After training the neural network, it is possible to make predictions about the tension state of the wheel disc, its weight and the amount of material needed for its manufacture.

For some dimensions (8/13 Input parameter) a minimum value, a maximum value and an incremental step have been chosen, as shown in the table.

Parameter	Min	Max	Step
R20	85	110	5
R50	10	20	1
D40	245	265	5
H10	30	50	1
R80	150	500	50
D50	325	335	1
R90	10	20	2
P30	-4	4	2
Total combinations		**571'725**	



In just over 6 minutes, the algorithm calculated the solution of over 570,000 solutions, providing (on request) the 4 solutions that minimize the tensions in the 4 areas defined as critical.

Deep learning: possible integration in the optimization process

Results comparison

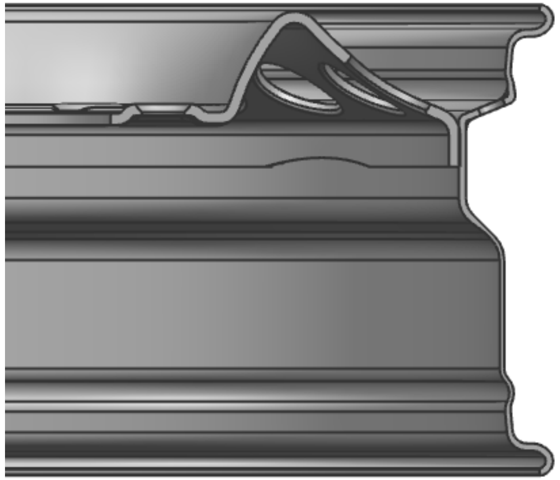
Downstream of the analysis, as a result, the algorithm returned the 4 configurations that minimize stress in the 4 critical areas.

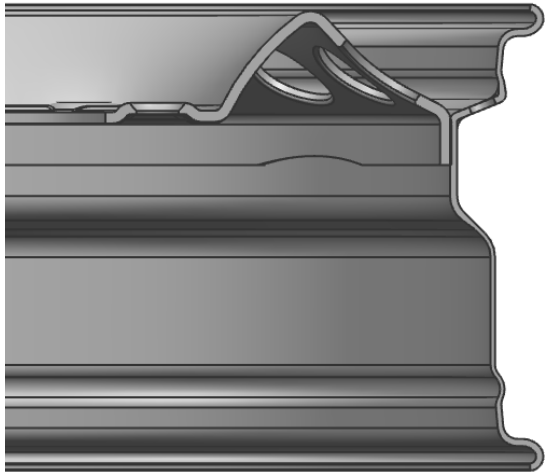
	R20	R50	D40	H10	R80	D50	R90	P30	S_Nose	S_Voile	S_VH	S_CP	Disc_weight_kg	Mean_surface_mm2
70506	109	12	245	50	150	328	12	-4	48.66285	53.23623	43.37734	24.35182	5.365451	169020.0
802434	109	10	265	50	150	328	10	0	48.96111	48.25116	42.67980	24.83352	5.390100	169584.3
403264	99	10	265	50	150	328	10	-2	49.46952	48.88618	40.62879	25.74068	5.394061	169561.9
1628394	109	16	245	43	500	328	10	4	67.12288	74.39213	71.66362	19.27807	5.169920	164065.2

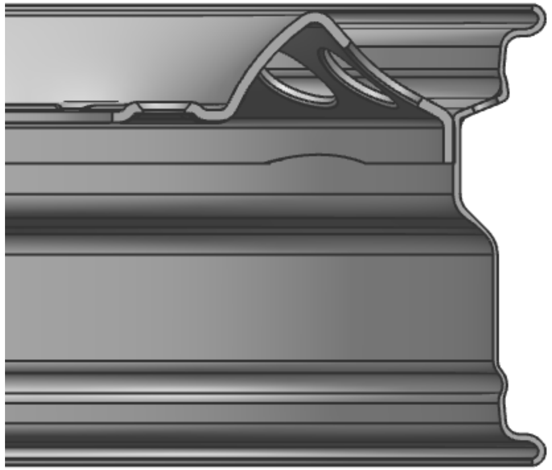
Geometrical parameters

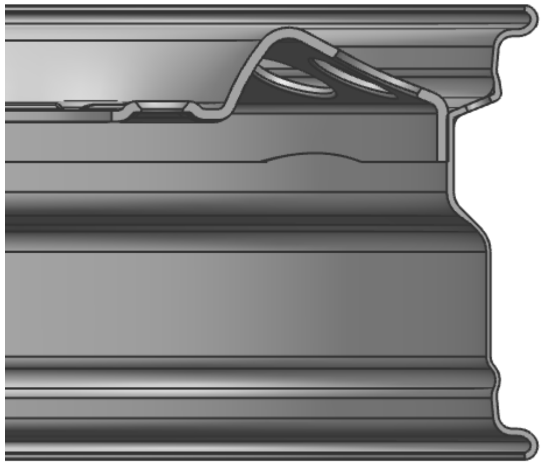
Stresses

Weigth and raw material





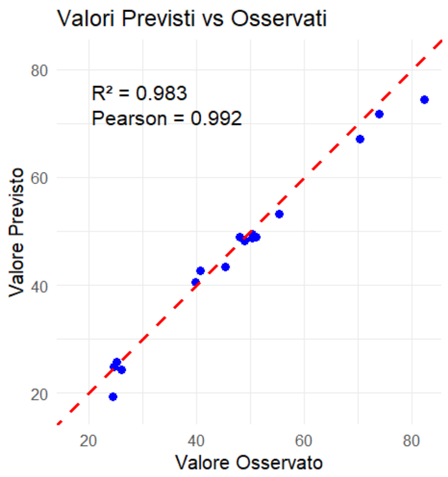


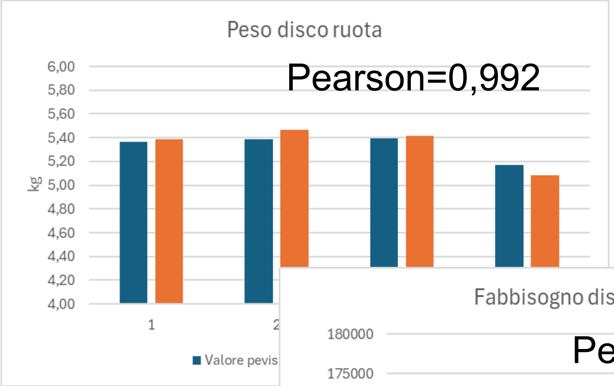


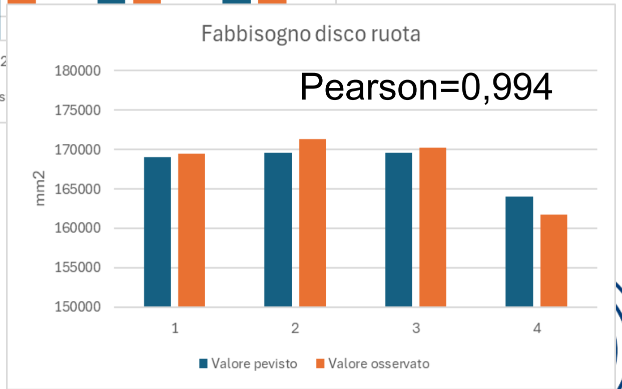
Deep learning: possible integration in the optimization process

Results comparison

	S_nose	S_Voile	S_VH	S_CP	Weight	Surface
NN	48,66	53,23	43,37	24,35	5,37	169020,00
FEM	50,39	55,38	45,39	26,12	5,39	169513,17
NN	48,96	48,25	42,67	24,83	5,39	169584,30
FEM	51,01	48,87	40,67	24,68	5,46	171361,92
NN	49,47	48,88	40,628	25,74	5,39	169561,90
FEM	50,31	47,95	39,84	25,17	5,42	170206,58
NN	67,12	74,39	71,66	19,28	5,17	164065,20
FEM	70,32	82,27	73,89	24,53	5,09	161723,35







Topic

Current wheel optimization process

Deep learning: possible integration in the optimization process

How it works

Conclusion

How it works

Bending moment definition

Definition of the bending moment at 200’000 cycles, typically used for the optimization loop.
All the stresses will be scaled to this Bending Moment.

```r
# ==============================================================
# DEFINUZIONE MOMENTO A 200'000cicli
# ==============================================================

Mf <- 4500 #Momento flettente a 200'000 cicli
```

Material definition

Definition of the disc material.
This permits to set the Stress Limits.

```r
# ==============================================================
# SCELTA MATERIALE
# ==============================================================
Material <- "MW06"

material_limits <- list(
  MW06 = c[redacted]
  MW05 = c[redacted]
  MW04 = c[redacted]
  MW01 = c[redacted]
)

# Controllo che il materiale scelto esista
if (!Material %in% names(material_limits)) {
  stop("Materiale non valido. Scegli tra: MW06, MW05, MW04")
}

limits <- material_limits[[Material]]

print(Material)
print(limits)
```

How it works

Input parameters definition

In this phase it is necessary to define for each dimensions the Min value, the Max value and the Step.
This will permit to generate all the possible combinations of all the parameters.

```
# ===============================================================
# DEFINIZIONE PARAMETRI GEOMETRICI
# ===============================================================
D_Thick_p <- seq(3.5, 5.5, by = 1)
R10_p     <- 6
D10_p     <- 176
R20_p     <- seq(85, 110, by = 5)
R50_p     <- seq(10, 20, by = 5)
D40_p     <- seq(245, 265, by = 10)
H10_p     <- seq(30, 50, by = 5)
R80_p     <- seq(150, 500, by = 50)
D50_p     <- seq(295, 365, by = 10)
R90_p     <- seq(10, 20, by = 5)
P30_p     <- seq(-4, 4, by = 4)
H20_p     <- seq(10, 20, by = 5)
D60_p     <- 390
N_VH_p    <- seq(16, 20, by = 4)
D70_p     <- seq(30.5, 40.5, by = 5)
```

How it works

Results

The program will generate 2 type of imput:

Minimum absolute configuration

- Configuration that minimize the Nose Stress
- Configuration that minimize the Voile Stress
- Configuration that minimize the V_Hole Stress
- Configuration that minimize the C_Part Stress
- Configuration that minimize the Disc weight
- Configuration that minimize the Disc blank surface

The best N solutions

The best N (of default 10) solutions that have the stresses below the Stress Limits in all the areas, and that minimize disc weight and disc blank surface.

R 4.4.3 · C:/Users/ronco/Desktop/Dottorato/Project/R/Project3/

```
46 536.3772 242.04526      4.291642       145653.0 45
47 409.2386 130.32626      4.995296       158965.6 45
48 341.0740  74.05300      5.958545       164335.8 45
49 536.5436 243.37946      4.293740       145246.2 45
50 409.5963 130.88988      5.018739       158584.0 45
 [ reached 'max' / getOption("max.print") -- omitted 839
```

```
> print(top10_best)
        Disc_thick_mm R20 R50 D40 H10 R80 D50 R90 P30 H20 D60 N_VH  D70   S_Nose  S_Voile
5920240           3.5  90  10 265  50 350 305  10   4  10 390   16 40.5 309.7388 333.7233
5920252           3.5 110  10 265  50 350 305  10   4  10 390   16 40.5 307.8734 331.6469
5920249           3.5 105  10 265  50 350 305  10   4  10 390   16 40.5 308.1132 331.8962
5920243           3.5  95  10 265  50 350 305  10   4  10 390   16 40.5 308.7985 332.6397
5920246           3.5 100  10 265  50 350 305  10   4  10 390   16 40.5 308.3531 332.1455
5759815           3.5 105  10 255  50 450 295  10   0  10 390   16 40.5 313.9729 334.8551
5759818           3.5 110  10 255  50 450 295  10   0  10 390   16 40.5 313.3429 333.9651
5766283           3.5  85  10 255  50 450 305  10   0  10 390   16 40.5 308.8117 332.8471
5611534           3.5 110  10 245  50 500 305  10  -4  10 390   16 40.5 309.9375 334.5363
5767093           3.5  85  10 255  50 500 305  10   0  10 390   16 40.5 308.7325 332.7714
            S_VH     S_CP Disc_weight_kg Mean_surface_mm2   Mf
5920240 206.6668 192.8250       3.943109         171100.5 4500
5920252 203.8066 192.4606       3.943889         170995.6 4500
5920249 204.1254 192.5723       3.944731         171035.8 4500
5920243 205.1226 192.7769       3.945473         171103.5 4500
5920246 204.4441 192.6839       3.945572         171076.0 4500
5759815 215.0480 198.4892       3.966462         168923.5 4500
5759818 213.9201 198.7972       3.969777         168885.9 4500
5766283 208.6773 190.2588       3.972011         170784.1 4500
5611534 213.5945 187.6809       3.972623         170169.5 4500
5767093 208.8702 190.0281       3.974807         170758.0 4500
```

```
      6.197475         169154.4 4500
      6.195110         168976.0 4500
      6.191007         168748.0 4500
      6.186961         168516.1 4500
      6.200791         169433.1 4500
      6.200254         169339.2 4500
      6.198191         169171.4 4500
ption("max.print") -- omitted 2138973 rows ]
```

Topic

Current wheel optimization process

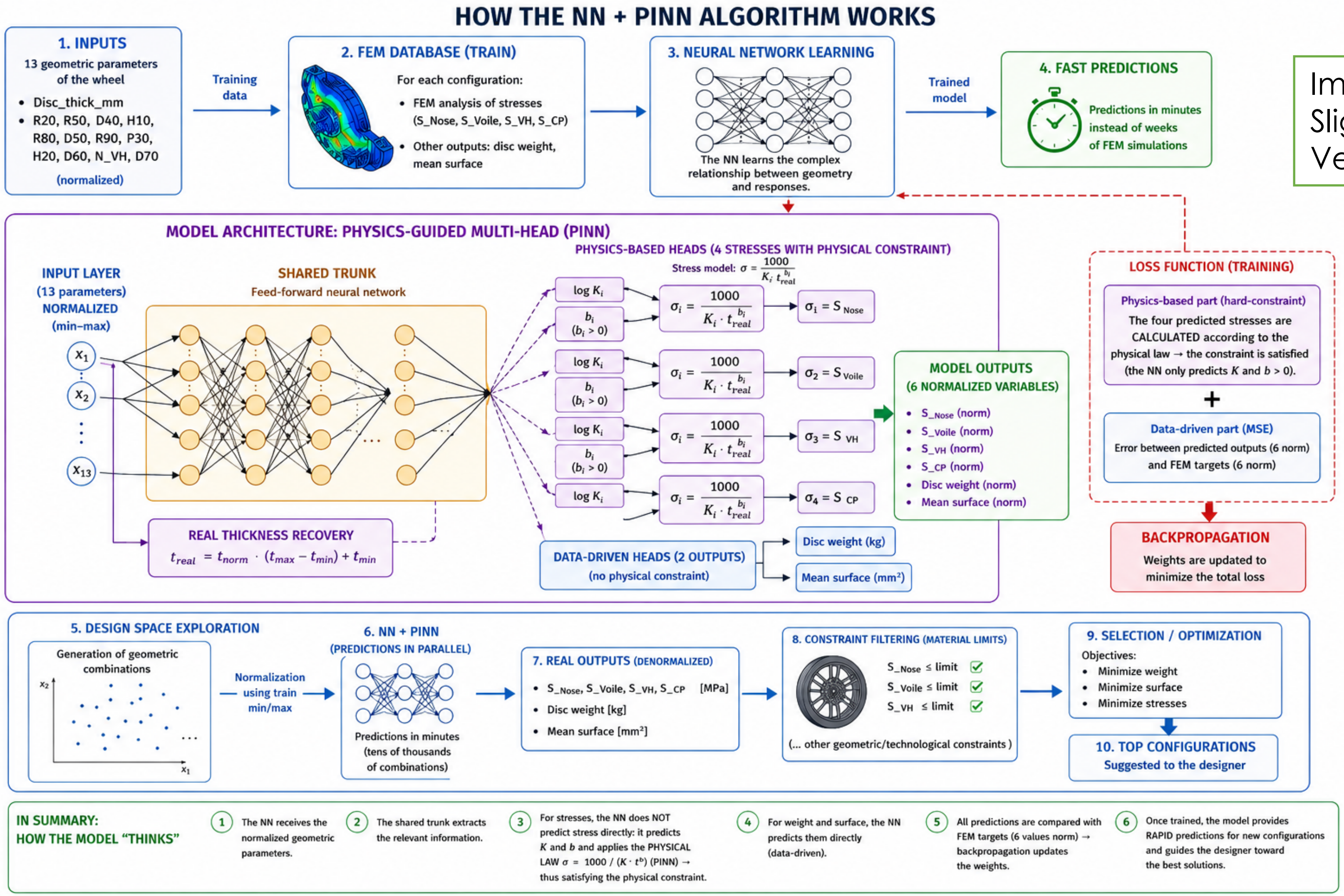
Deep learning: possible integration in the optimization process

How it works

Conclusion

How to improve

PINN (Physic Informed Neural Netweork)

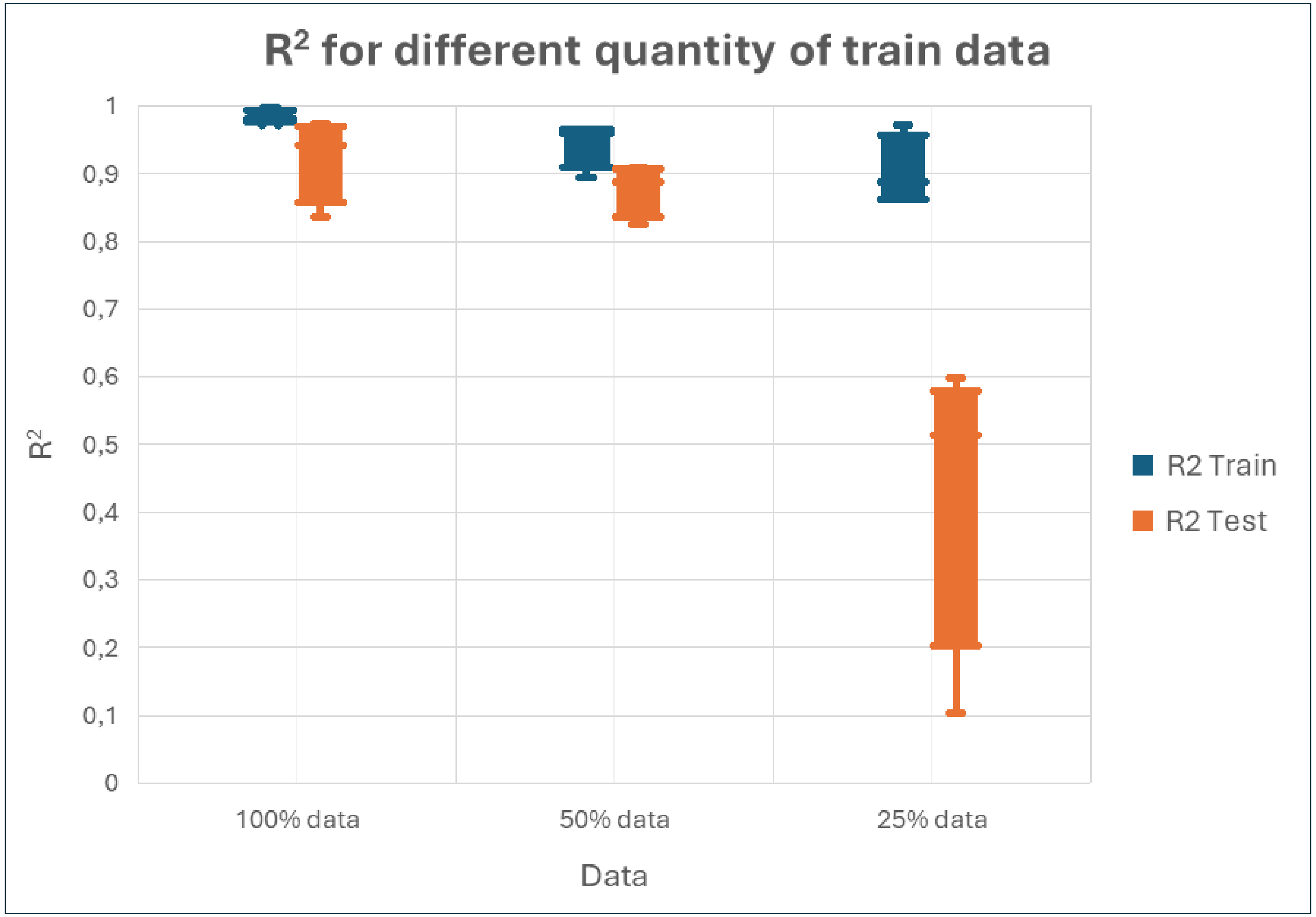


Implementation ready.
Slightly improvement: +2% on $R2$
Very high time consuming +300%.

How to improve

Increase the dataset

Increasing the training dataset, there is a huge improvement on reliability.



Conclusion

- The results in this study phase are very promising.
- The algorithm is able to autonomously select the optimal hyperparameters for the neural network training phase.
- Given the promising results, the implementation of a supervised Deep Learning algorithm could simplify and speed up the iterative optimization process of a wheel rim during the design phase.
- Once trained, the algorithm is capable of providing a vast number of solutions (10^5–10^6) in just a few minutes, making it possible to select the best one(s) according to the project requirements.

Thank you!

gruppocln.com