Evaluating the influence of loss function on performance of a neural network for particle 3D shape reconstruction from 2D projections

Danilo Menezes Santos[1], Alfredo Gay Neto[1]

[1]*Dept. of Structural and Geotechnical Engineering, Polytechnic School at University of São Paulo*
Av. Prof. Luciano Gualberto, 380 - Butantã, 05508-010, São Paulo, Brazil
dmsantosse@usp.br, alfredo.gay@usp.br

Abstract. In recent years, Supervised Neural Networks have been used to solve different tasks due to their ability to extract and predict patterns. This kind of algorithm has in its development a first step known as the training phase, where it tries to find the best values for the weights and biases, making the neural net predictions as close as possible to some expected outputs. In each training iteration, updating weights and biases is realized like an optimization problem. When developing a Neural Network that uses Spherical Harmonics Functions (SPH) for reconstructing the 3D shape of a particle from its 2D projections, one can choose as objective function different parameters, e.g., the volume, radius, the SPH coefficients, etc. In this paper, we discuss the performance of a Neural Network trained with different loss functions in a 3D shape reconstruction problem context.

Keywords: Morphology, Neural-Networks, Particle Shape, 3D Reconstruction.

1 Introduction

The costs of obtaining the 3D form of particles can be unfeasible when the granular medium has many particles or a small granulometry (Paixão et al [1], Yan and Su [2]). In this situation, Dynamic Imaging Analyzers (DIA) tools, e.g., QicPic and Aggregate Imaging Systems, are usually employed as alternatives to realize the morphological acquisition of grains.

The QicPic (Sympatec Gmbh [3]), is composed of a vibratory chute that accelerates the particles, making them drop in a measuring zone, equipped with a high-speed digital camera with a synchronized light source that captures the projections of particles at a rate of 500 frames per second (Witt et al [4]). Despite their efficiency and capacity to analyze a large number of particles, the QicPic only can provide information about the 2D form of particles like sphericity, Feret diameters, EQPC, convexity, and roundness, making this tool insufficient for some applications that needed on accurate 3D shape characterization of the grains, e.g., the Discrete Elements Methods.

To circumvent these limitations, many authors attempt to infer the three-dimensional format of grains by establishing correlations between the 2D and 3D shape descriptors. Through statistical methods, Yan and Su [2] estimated the average radius of ellipsoids from 2D projected images. Zhao et al [5] shows the applicability of some probability distribution functions in correlating the aspect ratio, sphericity, convexity, roundness, regularity, and roughness to cobbles and ballast particles. In Ueda [6] and Ueda et al [7], the 3D indices: sphericity, long/middle axis ratio, long/short axis ratio, volume, surface area, equivalent diameter, and long axis length were successfully estimated from measurable 2D projections, and the 3D particles shape was reconstructed through the combination of Genetic Algorithm and Spherical Harmonics Functions.

In essence, inferring the three-dimensional shape of objects from their two-dimensional data is an ill-posed problem. Understanding this, we employ Supervised Neural Networks (SNNs) to perform this task, considering that in addition to its generalization capacity, robustness, and parallelism, Durch and Diercksen [8] emphasize that this type of tool can perform the mapping of an arbitrary vector space into another vector space, correlating parameters of a given data set, being an interesting tool for ill-posed mathematical problems.

From a database with particles represented by Spherical Harmonics Functions (SPH), we trained a neural network to generate the harmonic coefficients necessary to reconstruct each particle, taking information from their projections. During the development of the neural network, it was noted that the choice of the loss function directly influences the neural network's capacity to accomplish the task of reconstruction. So, this paper shows the impact of the loss function on the final performance of neural networks.

2 Methodology

2.1 Acquisition and Reconstruction of particles

In this work, we used 30 particles taken from an example of railway ballast library developed by Moraes et al [9]. The digitized point cloud underwent pre-treatment that consisted of removing noise, positioning the point cloud coinciding with the origin, and finally scaling its bounding box through a linear transformation so that, in the end, the particle was adjusted to a cube with 2cm of edge.

As there was no uniformity between the total number of points obtained after digitization and treatment, it was decided to mathematically represent the particles employing spherical harmonic functions, a particular group of harmonic functions whose Laplacian is equal to zero, and have been used for the computational representation of various objects such as concrete aggregates and sand particles (Su and Yan [10], Liu et al [11]). For more information about the spherical harmonic functions, recommend the works of [12]. With the point cloud having its midpoint positioned at the origin, each sample point was mapped utilizing an azimuth (ϕ) and elevation (θ) angle, as seen in Fig. 1.

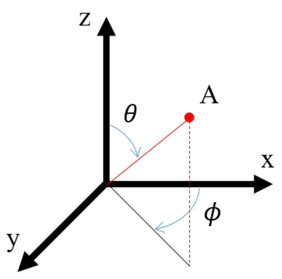


Figure 1. Angles used to locate a point in space.

Once the mapping is complete, the coordinate vector of the points $v(x, y, z)$ is expressed as a function of the azimuths and elevation angles $v(x(\phi, \theta), y(\phi, \theta), z(\phi, \theta))$. Using the least squares method and eq. (1) the harmonic coefficients required to approximate the three coordinates of the point cloud are determined.

$$\begin{aligned} x(\phi,\theta) &= \sum_{n=0}^{\infty}\sum_{m=-n}^{n} {c_x}_n^m Y_n^m(\phi,\theta) \\ y(\phi,\theta) &= \sum_{n=0}^{\infty}\sum_{m=-n}^{n} {c_y}_n^m Y_n^m(\phi,\theta) \\ z(\phi,\theta) &= \sum_{n=0}^{\infty}\sum_{m=-n}^{n} {c_z}_n^m Y_n^m(\phi,\theta) \end{aligned} \tag{1}$$

where the variables n and m are associated with Legendre Function, the ${c_x}_n^m$, ${c_y}_n^m$, ${c_z}_n^m$ are the spherical harmonics coefficients to degree n and order m, and $Y_n^m(\phi, \theta)$ the spherical harmonic at degree n. In our research, the degree adopted was equal to 16, so the number of coefficients used for reconstructing each particle was 768, encompassing 256 coefficients ${c_x}_n^m$, 256 coefficients ${c_y}_n^m$ and 256 coefficients ${c_z}_n^m$. The above equations can be expressed in a compact form by considering the existence of a matrix of generalized harmonic coefficients ($\mathbf{C}$), a set of generalized coordinates ($\mathbf{v}$), and a matrix of harmonic functions ($\mathbf{Y}$), as shown in eq. (2).

$$\mathbf{v}_{3xk} = \mathbf{C}_{3x256}\mathbf{Y}_{256xk} \tag{2}$$

where k is the number of points.

2.2 Dataset

Equation 2 is important for creating our dataset because, from it, we can rotate our particle in space and obtain different harmonic coefficients capable of reconstructing the same particle in a different orientation. Being $\mathbf{v}$ the vector with the coordinates of all points on the surface of a particle, when we apply a rotation matrix $\mathbf{R}$ we can write this transformation as indicated in eq. (3):

$$\mathbf{v}_{R,3xk} = \mathbf{R}_{3x3}\mathbf{v}_{3xk} \tag{3}$$

where $\mathbf{v}_R$ represents the new position of all points due to rotation. Substituting eq. (2) into eq. (3), and keeping the values of $\mathbf{Y}$ fixed, we obtain a new matrix of spherical harmonic coefficients $\mathbf{C}'$ able to reconstruct the original particle in actual orientation. A similar result to that obtained in eq. (4) was found by other authors such as Zhao et al [13] and Zucchelli et al [14].

$$\mathbf{v}_{R,3xk} = \mathbf{R}_{3x3}\mathbf{C}_{3x256}\mathbf{Y}_{256xk} = \mathbf{C}'_{3x256}\mathbf{Y}_{256xk} \tag{4}$$

Understanding how to get $\mathbf{C}'$, we fix the values of $\mathbf{Y}$ during the work, and for each of the 30 types of particles, we generate another 2500 particles identical to it but rotated. In this way, each original particle is represented by a set of 2500 vectors of spherical harmonic coefficients, able to describe it in a given orientation.

To produce the set of projections of the original particles, the different vectors of spherical harmonic coefficients were used to recreate 250,000 points of the particle surface, and then a new projection was generated by projecting these points onto the XY plane. Figure 2 provides an example of this procedure by showing a box aligned in distinct orientations and two projections obtained by applying this transformation.

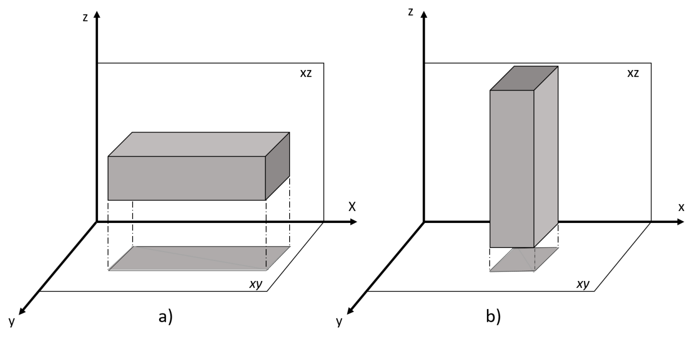


Figure 2. Extracting projections of a box, a) in the original position and b) after rotation.

Through this procedure, the dataset is composed of pairs containing the combination of spherical harmonics needed to change the orientation of a particle and the image projected onto the XY plane due to that rotation, how illustrated in Fig. 3.

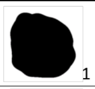
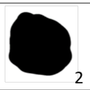
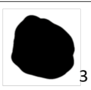
Projection	Spherical Harmonics Coefficients
 1	[-0.05511665406852835764,-0.1374319296127290146,...., 0.00042567368788293111 95, 0.0027289206424679465 95]$_{1x768}$
 2	[-0.05194697914480888440,-0.1555305058514278027,....,0.0004805450189347074000,0.002720035679467230297]$_{1x768}$
 3	[-0.04858433841995906527, -0.1730513378371226763,....,0.0005336312845970326950,0.002701046685921670507]$_{1x768}$

Figure 3. Example of dataset.

2.3 Neural Network

In recent years, the attention to Machine Learning algorithms has increased due to their ability to extract and predict patterns from datasets. According to Goodfellow et al [15], this capacity has enabled computers to solve problems that conventional computing techniques cannot. In the context of Geotechnical Science, applications like slope stability (Chakraborty and Goswami [16]), pile bearing capacity (Moayedi and Jahed Armaghami [17]), and soil parameters prediction (Wojciechowski [18]) are some examples of the use of SNNs.

Among the types of algorithms, Supervised Neural Networks (SNNs) are an assembly of interconnected layers made up of processing units called nodes or neurons. The information processing occurs by association between different layers, so when two neurons of different layers are connected, the output of one is the input of the other, and a weight is assigned to this connection. Each node has an activation function and a bias, and the outputs are produced by applying the activation function to the linear combination between all inputs, weights, and biases them.

The architecture of our SNN comprises eight layers. The first layer receives a projection of a particle through an image of 128x128 pixels. In the sequence, two convolutional layers, each with 16 filters and batch normalization, make the feature extraction, converting the input pixels into a noisy vector. The fourth layer transforms the outputs of convolutional layers into a column vector with 1024 elements. In the regression phase, a Multilayer Perceptron composed of two layers with 1024 neurons and one layer with 768 neurons processes the column vector data to produce an output vector with 768 Spherical Harmonics coefficients which has to be trained to be able to reconstruct the same particle that originated the input image. In all layers, the activation function adopted was the sigmoid. The network was implemented using Keras ([19]) environment due to its versatility, ease of manipulation, and the possibility of customizing the loss function used in training.

2.4 Training

For a neural network to be able to solve a specific problem, it is necessary to define the appropriate weights and biases. In the case of SNNs, this is realized through an adaptive process of stimulus called the training phase. This step is similar to an optimization process in that weights and biases are adjusted at each iteration to minimize an objective function that describes an error measure between the neural network's predictions and a target output. According to Ciampiconi et al [20], choosing a loss function for training depends on the problem's nature, the data available, and the type of machine learning algorithm to be optimized. For example, in a classification problem using the datasets MNIST and CIFAR10 Banerjee et al[21] found that the SM-Taylor softmax function that outperforms the original softmax. In Hwang et al [22], only changing the activation function of the last layer and loss function can develop a classification network, to detect if two particles are in contact, and a regression network to estimate the normal vector, point of contact, and depth of interpenetration.

Algorithm 1 Loss 1

```
$\mathbf{C'_{NN}} \leftarrow$ neural network output
$\mathbf{C'_{target}} \leftarrow$ desired output
$loss = 0$
for $count = 1, 2, \dots, 768$ do
    $diff = \mathbf{C'_{NN}}[count] - \mathbf{C'_{target}}[count]$
    $loss = diff^2 + loss$
end for
$loss = loss/768$
```

Three different loss functions were used during training to adjust the values of the 3,481,440 network parameters. The first loss (Table 1) function determines the mean square error between the neural network outputs and the expected harmonic coefficient values. The second loss function (Table 2) seeks to minimize the mean squared error between 2,500 points generated from the coefficients of the expected spherical harmonic functions and those generated by the neural network. In the third loss function (Table 3), the 2500 points generated are triangularized, and the polyhedral volume difference is computed.

All three network models were trained for 200 epochs with a learning rate of 0.002 and using the Adamgrad algorithm. For each family of particles, 1250 projections were used during training, and another 1,250 projections were reserved for testing the neural network's capacity.

Algorithm 2 Loss 2

```
$\mathbf{C'_{NN}} \leftarrow$ neural network output
$\mathbf{C'_{target}} \leftarrow$ desired output
$\theta_{\mathbf{list}} \leftarrow [],\phi_{\mathbf{list}} \leftarrow []$
for $i_1 = 1, 2 \ldots, 50$ do
    for $i_2 = 1, 2 \ldots, 50$ do
        Add($\frac{(2\pi-0.08)i_1}{50}$,$\theta_{\mathbf{list}}$), Add($\frac{(\pi-0.08)i_2}{50}$, $\phi_{\mathbf{list}}$)
    end for
end for
Initialize $\mathbf{Y}_{2500x256}$ like zero matrix
for $i_3 = 1, 2, 3 \ldots, 2500$ do
    $count = 1$
    for $n = 0, 1, \ldots, 16$ do
        for $m = -n, -n+1 \ldots, n-1, n$ do
            Computes $Y_n^m(\phi, \theta)$
            $\mathbf{Y}[i_3, count] \leftarrow Y_n^m(\phi, \theta)$
            $count = count + 1$
        end for
    end for
end for
reshape $\mathbf{C'_{NN}}$ to 3x256
reshape $\mathbf{C'_{target}}$ to 3x256
$\mathbf{v_{NN}} = \mathbf{C'_{NN}} \cdot \mathbf{Y}^T$
$\mathbf{v_{target}} = \mathbf{C'_{target}} \cdot \mathbf{Y}^T$
$loss = 0$
for $i_4 = 1, 2, 3 \ldots, 2500$ do
    $loss = \sqrt{(\mathbf{v_{NN}}[1, i4] - \mathbf{v_{target}}[1, i4])^2 + (\mathbf{v_{NN}}[2, i4] - \mathbf{v_{target}}[2, i4])^2 + (\mathbf{v_{NN}}[3, i4] - \mathbf{v_{target}}[3, i4])^2} + loss$
end for
$loss = loss/2500$
```

Algorithm 3 Loss 3

```
$\mathbf{C_{NN}} \leftarrow$ neural network output
$\mathbf{C_{target}} \leftarrow$ desired output
Generate 2500 points equal to Loss2
Triangularize the point cloud
Computes the volume for the Neural Network reconstruction ($Vol_{NN}$)
Computes the volume for the desired reconstruction ($Vol_{target}$) $loss = (Vol_{NN} - Vol_{target})^2$
```

3 Results

After training, the neural networks generated spherical harmonic coefficients by analyzing the projections not used when adjusting weights and biases. With these new harmonic coefficients, a set of particles was generated, and the values of their volumes and sphericity were compared with those of the particles that produced the projections. A reconstruction was considered satisfactory whenever its sphericity or volume value differed by less than 10% from the expected value. Figure 4 summarizes the number of projections classified by particle type.

As one can see, the training using the Loss 3 function failed to achieve an adequate generalization capacity, so none reconstructed particle could return adequate values of volume or sphericity. On the other hand, after training, Loss 1 and Loss 2 proved to be adequate in inferring the volume and sphericity of the particles, considering that in both cases, the number of projections classified by particle type was always greater than 50%. Loss 2 performed better in the validation database than the Loss 1 function, presenting average reconstruction values of around 80% for both descriptors.

In addition to the observation of the shape descriptors, we also sought to verify how close the particles generated by the network and the target particles are. For this, 400,000 points were analyzed, where for the same pair of elevation and azimuth, the radii coordinate obtained by the network and by the vector of real harmonic coefficients were compared whenever the greatest variation between the radii was less than 10%, it was considered that the reconstructed particle presents a shape value consistent with the expected. Figure 4 shows a summary of the number of projections considered similar to those of the original particle, as seen in Loss 2, which continued to outperform the other shape functions. It is important to note that when using this classification criterion, even Loss 1 and Loss 2 presented a particle where the number of reconstructed projections was much lower than the others, its value represented by the ball in Fig. 4.

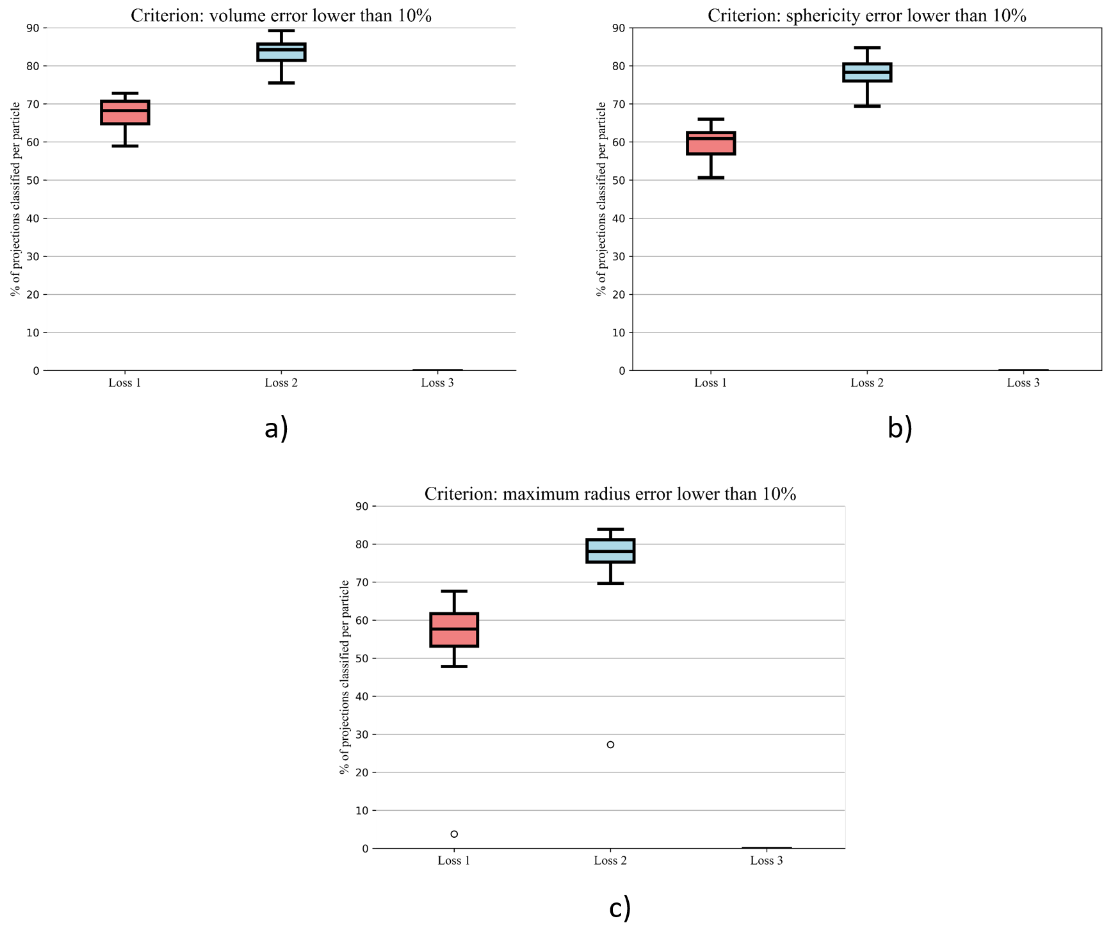


Figure 4. Boxplot particle classification per type a) volume b) sphericity and c) maximum radius error .

4 Conclusions

This paper provides a brief explanation of the development of neural networks to infer the three-dimensional shape of particles from their projections, with an essential emphasis on the analysis of the impact that the loss function has on the network's ability to accurately approximate data that were not present in the training database.

As main conclusions, we have that using volume as an objective function in training leads to an inaccurate adjustment of weights and biases. In contrast, loss functions 1 and 2 can enable the network to perform the desired task, recreating particles with sphericity, volume, and shape close to those expected.

Acknowledgements. The authors acknowledge Vale S.A. for the support through the ”EMBRAPII CICS e ESALQ - Desenvolvimento de produtos e soluções para utilização em grande escala de rejeitos de cobre e níquel” project. The second author acknowledges National Council for Scientific and Technological Development (CNPq) under the grant 304321/2021-4

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] A. Paixão, R. Resende, and E. Fortunato. Photogrammetry for digital reconstruction of railway ballast particles–a cost-efficient method. *Construction and Building Materials*, vol. 191, pp. 963–976, 2018.
[2] W. Yan and D. Su. Inferring 3d particle size and shape characteristics from projected 2d images: Lessons learned from ellipsoids. *Computers and Geotechnics*, vol. 104, pp. 281–287, 2018.
[3] Sympatec's image analysis overview and concept. `https://archive.sympatec.com/EN/ImageAnalysis/ImageAnalysis.html`, 2016.
[4] W. Witt, U. Köhler, and J. List. Current limits of particle size and shape analysis with high speed image analysis. *PARTEC 2007*, 2007.
[5] L. Zhao, S. Zhang, M. Deng, and X. Wang. Statistical analysis and comparative study of multi-scale 2d and 3d shape features for unbound granular geomaterials. *Transportation Geotechnics*, vol. 26, pp. 100377, 2021.
[6] T. Ueda, T. Oki, and S. Koyanaka. 2d-3d conversion method for assessment of multiple characteristics of particle shape and size. *Powder Technology*, vol. 343, pp. 287–295, 2019.
[7] T. Ueda. Estimation of three-dimensional particle size and shape characteristics using a modified 2d–3d conversion method employing spherical harmonic-based principal component analysis. *Powder Technology*, vol. 404, pp. 117461, 2022.
[8] W. Duch and G. H. Diercksen. Neural networks as tools to solve problems in physics and chemistry. *Computer physics communications*, vol. 82, n. 2-3, pp. 91–103, 1994.
[9] de S. T. Moraes, P. Pereira, A. G. Neto, L. L. Bernucci, R. Mota, E. Moura, and others. Cisalhamento direto de lastro ferroviário: modelo numérico e sua calibração. *Geotecnia*, , n. 155, pp. 103–117, 2022.
[10] D. Su and W. Yan. 3d characterization of general-shape sand particles using microfocus x-ray computed tomography and spherical harmonic functions, and particle regeneration using multivariate random vector. *Powder Technology*, vol. 323, pp. 8–23, 2018.
[11] X. Liu, E. Garboczi, M. Grigoriu, Y. Lu, and S. T. Erdoğan. Spherical harmonic-based random fields based on real particle 3d data: improved numerical algorithm and quantitative comparison to real particles. *Powder Technology*, vol. 207, n. 1-3, pp. 78–86, 2011.
[12] K. Atkinson and W. Han. *Spherical harmonics and approximations on the unit sphere: an introduction*, volume 2044. Springer Science & Business Media, 2012.
[13] B. Zhao, D. Wei, and J. Wang. Particle shape quantification using rotation-invariant spherical harmonic analysis. *Géotechnique Letters*, vol. 7, n. 2, pp. 190–196, 2017.
[14] M. Zucchelli, S. Deslauriers-Gauthier, and R. Deriche. A closed-form solution of rotation invariant spherical harmonic features in diffusion mri. In *Computational Diffusion MRI: International MICCAI Workshop, Granada, Spain, September 2018 22*, pp. 77–89. Springer, 2019.
[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`, 2016.
[16] A. Chakraborty and D. Goswami. Prediction of slope stability using multiple linear regression (mlr) and artificial neural network (ann). *Arabian Journal of Geosciences*, vol. 10, pp. 1–11, 2017.
[17] H. Moayedi and D. Jahed Armaghani. Optimizing an ann model with ica for estimating bearing capacity of driven pile in cohesionless soil. *Engineering with Computers*, vol. 34, pp. 347–356, 2018.
[18] M. Wojciechowski. Application of artificial neural network in soil parameter identification for deep excavation numerical model. *Computer Assisted Methods in Engineering and Science*, vol. 18, n. 4, pp. 303–311, 2017.
[19] F. Chollet and others. Keras. `https://keras.io`, 2015.
[20] L. Ciampiconi, A. Elwood, M. Leonardi, A. Mohamed, and A. Rozza. A survey and taxonomy of loss functions in machine learning. *arXiv preprint arXiv:2301.05579*, 2023.
[21] K. Banerjee, R. R. Gupta, K. Vyas, B. Mishra, and others. Exploring alternatives to softmax function. *arXiv preprint arXiv:2011.11538*, 2020.
[22] S. Hwang, J. Pan, A. A. Sunny, and L.-S. Fan. A machine learning-based particle-particle collision model for non-spherical particles with arbitrary shape. *Chemical Engineering Science*, vol. 251, pp. 117439, 2022.