Learning the Inverse Solution of Laser Drilling Model

Zahra Kheirandish[1*], Christian Heinigk[1], and Wolfgang Schulz[1,2]

[1]*Nonlinear Dynamics of Laser Processing, RWTH Aachen University, Aachen, Germany*
[2]*Fraunhofer Institute for Laser Technology, Aachen, Germany*
[*]*Corresponding author's e-mail: zahra.kheirandish@nld.rwth-aachen.de*

Laser drilling cooling holes into turbine components is well established in industries. However, there are drawbacks related to setting and maintaining the appropriate manufacturing conditions, which can be formulated as an inverse problem. The inverse solution of a physical model describing long-pulse laser drilling of sheet metal is learned through an artificial algorithm. The trained feed-forward neural network predicts process parameters of desirable production outcomes. Here, the trained network predicts beam radius and pulse power to drill through getting a hole with specified conicity. The hyperparameters of the neural network are trained using algorithmic differentiation. Using a physical model improves the solvability of the inverse problem, since all trials during training belong to the applicable range of laser drilling and improves the training procedure going in a physically informed direction.

DOI: 10.2961/ilmn.2024.02.2001
Keywords: laser drilling, inverse solution, physically informed neural network, customized training loss, reduced model

1. Introduction

Nowadays, laser beam ablation is a preferred manufacturing procedure for drilling holes in an extensive range of diameters, from one micrometer to several millimeters, without any physical contact. This ablation process is operated in various manufacturing applications, such as drilling filters, turbine-blade, or injection nozzles. The dominant physical phenomena mainly depend on laser radiation's intensity and pulse duration. Considering the physical phenomena during the laser pulse, the laser-material interaction is categorized as cold ablation (fs-pulses, 10^{16} to 10^{22} Wcm^{-2}), hot ablation (ps-pulses, 10^{15} Wcm^{-2}, and melt expulsion (ns$\sim \mu$s-pulses, 10^{9} to 10^{11} Wcm^{-2}) [1]. Hence, the mathematical models should describe the propagation and absorption of laser radiation leading to ionization, evaporation, melting, and heating. However, assuming all or even most of the mentioned phenomena conduct to nonlinear, time-dependent, and complex equations that are computationally expensive.

To improve solvability, phenomenological model reduction is applied successfully for different drilling regimes. Furmanski et al. [2] used this technique to model the laser manufacturing process to optimize an equation describing the ultrashort pulse laser ablation. In another investigation, this method was enforced to designate an asymptotic model for laser drilling [3]. The asymptotic model is based on the observation that after a certain number of pulses, the bore wall has become so steep that, although it is illuminated by the laser radiation, it hits too large an area and the material can no longer be removed. The model is a forward solution for the laser drilling process, which accurately calculates the drill-hole shape. However, the inverse answer to this process is even more helpful from the manufacturing point of view since, in an existing production process, the manufacturers pursue a particular process's parameters for constructing a specific hole shape.

In general, inverse problems are hard to solve and often ill-posed. Artificial Intelligence (AI) is becoming a versatile tool to solve them. For modeling and solving inverse problems, artificial Neural Networks (ANNs) with suitable function approximation properties are given [4, 5]. These studies demonstrate tremendous potential for combining existing physical knowledge in the form of governing equations with the data-driven approach of ANNs. These prototypes not only solve the inverse problems but can also replace expensive computational simulations with faster models. Such models unleash the true potential of AI approaches by incorporating physics-based equations into a guided learning process.

In terms of abilities of ANNs, they are introduced in various fields, from medical [6, 7] and ecological studies [8] to different engineering applications like estimating elastic settlements of shallow foundations [9], prediction of combustion profile [10], speed control of the alternating current (AC) induction motor drive [11], and laser manufacturing processes (cutting [12], welding [13, 14, 15], engraving [16, 17], and additive manufacturing [18, 19, 20]). To optimize forming quality and reducing computational cost of modeling the Selective Laser Melting (SLM) process, the ANN is developed which predicts accurately the density of metal specimens in high-efficiency molding and helps in analyzing the uncertainty and sensitivity [21] [22]. Additionally, several studies have been conducted on the use of ANNs in the laser drilling process. For instance, a feed-forward neural network was established to optimize the parameters of laser drilling machining and produce optimum grooves in die-Steel [23, 24]. Another analysis worked on the prediction of the performance measures (circularity at entry and exit, taper, spatter area, and kerf geometry) and studied its variations under the extended range of laser drilling parameters [25, 26, 27]. To obtain an effective parameter optimization strategy for fabricating the silica

glass microchannels with a femtosecond laser, an ANN is established; it effectively and swiftly predicts the depth of tapered microchannels. This approach addresses the intricate nonlinear connection between laser processing parameters and machining outcomes, offering a practical means to forecast microchannel depths [28]. In addition, ANNs as surrogate model in simulating nanosecond laser percussion drilling of for Through-Silicon Vias (TSV), leades to the identification of an optimized processing region with specific energy and pulse parameters for excellent TSV results [29].

In the quoted analyses, the ANNs are used to solve the laser drilling problem forwardly. Conversely, in the current investigation, the ANN is employed to learn an approximation to the inverse solution; it is suggested to switch the inputs and outputs and predict the required process parameters for desirable production outcomes. Herein, the inputs are the top and bottom widths of a predefined borehole describing conicity, and outputs will be the laser pulse power and the beam radius. The backbone of this approach is a pre-validated phenomenological model combined with the ANN to gain the inverse solution. This approach represents a customized cost function that contains the ODE of the physical model; in fact, the ODE provides the physical-mathematical relations among variables during the backpropagation. The cost function is minimized w.r.t. the weights and biases of the ANN by adaptive moment estimation algorithm (Adam) [30] which is a first-order gradient-based optimization method. To establish the expressed method, firstly, a database is generated through the asymptotic drill model (developed by [3]). Then, this dataset is used to train and optimize the ANN, and finally, the ANN-model's performance is tested by an unseen dataset. During the training phase, the gradient of the ODE is required; in this study, the algorithmic differentiation (AD) technique is employed, which is an automatic method for computing gradients of numerical algorithms [31].

The following approach is structured as follows. The succeeding main section explains the long-pulse laser drilling model and its range of applicability. Section 3 then introduces the differential programming technique and explains the methodology. Section 4 demonstrates the operation of learning inverse solutions and closes with accomplished experimentation. Finally, Section 5 concludes the outcomes.

2. Physical modeling

2.1. The long-pulse laser drilling model

Hermanns's [3] model describes the relation between absorbed power intensity and the shape of the drilled hole of the long-pulse laser drilling process. Since the pulse duration of long laser pulses is a few orders of magnitude larger than the electron-phonon scattering time, melting is the dominant thermal process during the ablation. Based on this phenomenological fact, the material is ablated when it absorbs a specific amount of power called ablation intensity-threshold ($\tilde{I}_{\mathrm{thr}}$). This threshold value is calibrated experimentally using one single borehole diameter and a distributed intensity $I(x,z)$ measurement. During calibration, the ablation threshold ($\tilde{I}_{\mathrm{thr}}$) is determined to give the diameter of the borehole as manufactured. By this asymptotic time-independent model, one can sum up the effects of complicated phenomena during the laser drilling process (i.e., ionization, evaporation, melting, melt flow, and condensation) in the threshold value. As a result, the absorption coefficient $A(\cos(\theta_{\mathrm{inc}}))$, the projection factor $\cos(\theta_{\mathrm{inc}})$ and the threshold intensity are the phenomena governing and the asymptotic shape of the drill wall:

$$A\big(\cos(\theta_{\mathrm{inc}})\big)\cos(\theta_{\mathrm{inc}})I(x,z) = \tilde{I}_{\mathrm{thr}}. \tag{1}$$

$z_{\mathrm{l}}(x)$ and $z_{\mathrm{r}}(x)$ are defined as the z-coordinate of borehole's wall at left and right side of the borehole. Utilizing the ablation intensity-threshold criterion, the asymptotic laser drilling model assesses the slope of the right and left sides of the drilled hole as:

$$\frac{dz_{\mathrm{r,l}}}{dx} = \pm\sqrt{\left(\frac{A\big(\cos(\theta_{\mathrm{inc}})\big)I(x,z_{\mathrm{r,l}})}{\tilde{I}_{\mathrm{thr}}}\right)^2 - 1}. \tag{2}$$

Investigations of [3] presented an adequate agreement between phenomenological eq. (2) and experimental results.

At the top surface $(x, z = 0)$, the laser beam radiates at a large value for the projection factor $\cos(\theta_{\mathrm{inc}})$, and the ablation occurs as the projected intensity of absorbed energy reaches the ablation threshold of the material. Hence, the right $(x_r, 0)$ and the left $(x_l, 0)$ bounds of the ablated area (i.e., $(x_{(\mathrm{r,l}),0}, 0)$) are estimated by $I_{\mathrm{eff}}(x_{(\mathrm{r,l}),0}, 0) = \tilde{I}_{\mathrm{thr}}$. For a specific depth of the hole (i.e., Δz), the next x-coordinate of the wall is calculated as:

$$x_i = x_{i-1} + \left(\frac{dz}{dx}\right)^{-1}\Delta z. \tag{3}$$

As represented in eq. (2), the slope of the wall of borehole is proportional to the ratio of absorbed intensity and material threshold intensity. The beam incident angle θ_{inc} is obtained from inner product of the surface normal and Poynting vectors of the beam, as shown in **Fig. 1**. Consider a differential triangular element near the incident point of the beam and left side of the hole. The inclination dz/dx of the left drilled wall is given by:

$$\tan(\theta_{\mathrm{l}}) = \frac{dz}{dx}, x < 0 \tag{4}$$

where the inclination angle θ_{l} is:

$$\theta_{\mathrm{l}} = \theta_{\mathrm{zl}} + \theta_{\mathrm{inc}}. \tag{5}$$

Within the Rayleigh length of the laser beam the angle θ_{zl} between beam axis $x = 0$ and Poynting vector remains negligible small. If θ_{zl} remains smaller than 2.5° the inclination angle θ_{l} is set equal to the incident angle θ_{inc}.

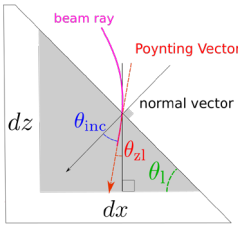

Fig. 1 Geometric relations related Poynting vector and normal vector at the drilled wall.

Gaussian distribution is supposed for the focused laser beam intensity $I(x,z)$:

$$I(x,z) = I_0 \left(\frac{w_0}{w(z)}\right)^2 e^{\left(-2\left(\frac{x}{w(z)}\right)^2\right)}, \quad (6)$$

where beam radius is:

$$w(z) = w_0\sqrt{1+\left(\frac{z}{z_R}\right)^2} \quad (7)$$

The maximum intensity I_0, is related to the laser pulse power P_l, beam waist ω_0, pulse duration τ and repetition rate ν_{rep} as $I_0 = 2P_l/(\pi w_0^2.\tau.\nu_{rep})$. The spatial shape of the beam is characterized by the focal number F, Rayleigh length z_R and the beam quality K, which are given by:

$$\begin{aligned} F &= \frac{f}{D}, \\ z_R &= \frac{\lambda}{\pi}\frac{F^2}{K}, \\ K &= \frac{\lambda}{\pi}\frac{1}{\theta_0 w_0}, 0 < K < 1. \end{aligned} \quad (8)$$

Here, D is the illuminated radius of the focusing lens with focal length f and, $\theta_0 = w_0/z_R$ and w_0 are the far-field divergence angle and the focal radius, respectively.

In the limit of metallic optical properties, where the index of refraction n and absorption κ are equal and large compared to unity ($n = \kappa >> 1$) the dielectric function $\epsilon = n^2 - \kappa^2$ tends to zero. Consequently, displacement currents become negligible. The conductive currents $\sigma/(\epsilon_0\omega) = 2n\kappa$ describe the metallic behavior. As consequence, the Fresnel formulae yield the degree of absorption A_s and A_p for s- and p-polarized light:

$$A_s(\mu) = \frac{4\iota\mu}{2+2\iota\mu+\iota^2\mu^2}, \quad (9)$$

$$A_p(\mu) = \frac{4\iota\mu}{2\mu^2+2\iota\mu+\iota^2}. \quad (10)$$

Here, $\iota = \sqrt{1/nk}$ and $\mu = \cos(\theta_{\text{inc}})$ are the material parameter and the projection factor , respectively. In the present study, a laser beam with a wavelength of 1.068 µm is radiated on AISI 304 stainless steel; thus ι equals 0.25. For non-polarized radiations, the total absorption coefficient is calculated by assuming equal portions for perpendicular and parallel beams, as below:

$$A(\mu) = 0.5A_s(\mu) + 0.5A_p(\mu). \quad (11)$$

Comparing $A(\mu)$ and $\mu \times A(\mu)$ as functions with respect to μ, it is observed that $\mu \times A(\mu)$ could be estimated by a linear relation with a constant slope G (see **Fig. 2**). Therefore, eq. (1) is simplified, as:

$$G \times \cos(\theta_{\text{inc}}) \times I = \tilde{I}_{\text{thr}}. \quad (12)$$

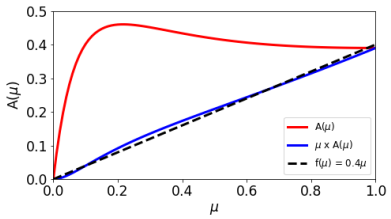

Fig. 2 Variation of absorption coefficient of AISI 304 stainless steel w.r.t. cosine of incident.

By defining the $I_{\text{thr}} = \tilde{I}_{\text{thr}}/G$, eq. (1) reads:

$$\cos(\theta_{\text{inc}}) = \frac{I_{\text{thr}}}{I}. \quad (13)$$

Finally, the first-order differential equation (i.e., eq. (2) is solved for z of the left and right side of the borehole, and consequently, the x- and z-coordinate of the drilled wall is achieved. This approach is the asymptotic model for the long-pulse laser drilling process.

2.2. Range of applicability

Each model performs for certain ranges of inputs. These adequate and usable sets of input process parameters restrict the applicable region. Some input parameters are forceful in the physical phenomenon and play more influential parts than others. Hence, rather than analyzing the model's performance under the effect of all inputs, only the dominant ones are varied, and the rest are kept constant. In this study, constant process parameters are called environment variables, and the varying ones are named features (here, the laser beam radius and pulse power), as detailed in **Table 1**.

Table 1 Process parameters before specifying applicable range of drilling through AISI 304 stainless steel.

process parameter	symbol	value
thickness	H	3 mm
beam waist	ω_0	0.01 - 0.2 mm
pulse power	P_l	10-500 W
focal position	z_0	0 mm
pulse duration	τ	0.5 ms
repetition rate	ν_{rep}	50 Hz
wavelength	λ	1068 nm
beam quality factor	K	1

The applicable range is defined based on features' combinations that lead to a drilled through-hole. The initial and applicable ranges of parameters are illustrated

in the **Fig. 3**. Here, laser beam waist's and pulse power's ranges of [10 - 200 µm]×[130 - 500 W] lead to through-holes with diameters 33 to 580 µm. Here, the initial data set contains 10000 samples in the mentioned ranges.

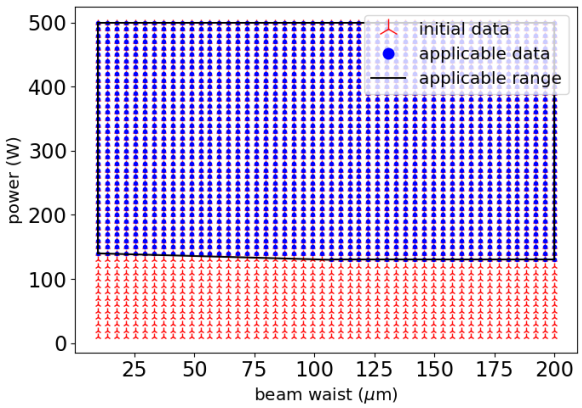

Fig. 3 Initial and applicable ranges of laser beam waist and pulse power. Other process parameters have constant values as tabulated in **Table 1**.

After extracting the applicable range of process parameters, the introduced asymptotic drill model is executed to generate the training dataset. After applying the applicable ranges of process parameter, the training dataset contains 5600 cases. Additionally, an unseen dataset is rendered to evaluate and test the final trained ANN.

3. Differential programming

3.1. Customized ANN

An ANN is a supervised knowledge processing algorithm founded on the structure of biological neural networks. Its architecture is constructed of nodes and layers whose suitable combination is different for each problem. The first and last layers are input and output, the rest are called hidden layers. Inside the ANN, information is processed in nodes with specific weights, and a threshold value dubbed bias is added to each hidden layer. Thus, when dealing with ANNs, optimizing hyperparameters is vital. Restrictions which dictated by activation functions rule fired signals of nodes. Indeed, ANNs work only like simple linear regression without activation functions; thus, nonlinear activation functions are imposed on the network to enhance performance. Routine activation functions are binary step, Sigmoid, or rectified linear unit functions. The optimum values of weights are determined via the iterative training steps, so-called backpropagation; it is established on the mathematical chain rule, which aims to minimize training loss. The ANN's performance is assessed with unseen data by completing the optimal weight and bias values.

In the current study, a customized architecture for ANN is developed to profit from the benefits of a pre-validated ODE besides the capabilities of deep neural networks. In **Fig. 4** the configuration of our customized ANN and dataflow are displayed, where $d_{i/o}^{t}$ and $d_{i/o}^{p}$ represent the target and predicted diameters (input/output) of drill-hole, respectively. $P_{1,2}$ and P_{env} stand for varying and constant parameters of ODE. In current study $P_{1,2}$ represents pulse power and beam waist and P_{env} is constant parameters. The ODE solver is embedded into the learning algorithm's architecture; in this way, the network learns to predict quantities constrained by a physical model represented by the ODE solver. In contrast to a standard ANN model, herein, the input layer contains the target value i.e., drill hole geometry. Network's last layer passes the required process parameters. In training stage, the ANN is optimized to replicate the target values by minimizing the loss function.

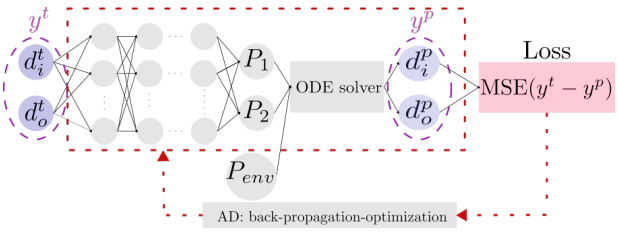

Fig. 4 Structure of present deep neural network and its data-flow. For laser drilling model.

The Learning algorithm includes the loss function and optimization benchmark. Assume that mean-square error (MSE) is the loss function:

$$l = \frac{1}{N}\sum_{i=1}^{N}(y_i^t - y_i^p)^2, \tag{14}$$

where

$$y_i^p = ODE(P_{1,2}(w,b), P_{env}). \tag{15}$$

Here, N, $y_i^{t,p}$, w, and b are batch size, target/predicted values, network's weight, and bias, respectively. To minimize the loss function, its derivatives w.r.t weights and biases are required, i.e., dl/dw and dl/db. Based on chain rule of calculus, the derivatives read:

$$\frac{dl}{dw} = \frac{-2}{N}\sum_{i=1}^{N}\frac{dy_i^p}{dw}(y_i^t - y_i^p), \tag{16}$$

$$\frac{dl}{db} = \frac{-2}{N}\sum_{i=1}^{N}\frac{dy_i^p}{db}(y_i^t - y_i^p). \tag{17}$$

Generally, in training ANNs, the gradient of the loss function is calculable effortlessly. However, as seen in eq. 16 and eq.17, by entrenching the ODE solver into the network, derivatives of the ODE solver are also required. Fortunately, algorithmic differentiation (AD) can ease this issue which is based on tracing the evaluations. In the opposite of symbolic differentiation, it can differentiate the algorithms which contain loops, recursion, and procedure calls. Indeed, it can aid in evaluating the gradient of numeric algorithms and numerical expressions by exploiting the chain rule of differential calculus.

The AD technique has two modes, i.e., forward and inverse. In the forward mode, the actual values of dependent variables are computed, and simultaneously, their derivatives are evaluated by involving the chain rule in each elementary operation. Nevertheless, when the number of independent variables is higher than the dependent ones, the forward AD does not perform efficiently, and inverse AD is advised [31]. This mode is assembled of the forward and backward phases; in the

former stage, the original function code is executed entirely, and the intermediate variables and dependencies are recorded; afterward, in the later phase, the derivatives are computed by establishing the adjoints of intermediates variables from outputs to inputs. Recently, few studies have engaged the AD technique in training neural networks, and numerical modeling [32, 33]. In the present analysis, an ANN model is built through *TensorFlow* [34] which is a well-known open-source machine learning platform. This platform benefits from the inverse AD for training ANNs which can compute the gradients of differentiable expressions.

3.2. Training optimization

In terms of the optimization strategy, gradient-based techniques like stochastic gradient descent (SGD) and adaptive moment (Adam) serve efficiently in training ANNs. They track the gradient of the entire training set downhill until it reaches the minimum. The weights and biases are corrected in each epoch, and a parameter called learning rate (α) manages the optimization procedure:

$$w^{n+1} \leftarrow w^n - \alpha \frac{dl}{dw}, \quad (18)$$

$$b^{n+1} \leftarrow b^n - \alpha \frac{dl}{db}. \quad (19)$$

The learning rate sensitizes this iterative procedure proceeds in epochs and the rate of descent. The SGD method is efficacious, but its sensitivity to learning rate is high, and when the training dataset is immense, it acts slowly, and the outcomes are noisy. Hence, the momentum method (Adam) is favored because it directs faster the gradient downturn in the relevant direction and lessens oscillations.

3.3. Training statistics

To figure out the distribution of the errors of outputs at the end of each experiment, the histograms of the mean relative errors (MRE) are assessed. The percentage of MRE is computed as:

$$MRE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i^t - y_i^p}{y_i^t} \right| \times 100. \quad (20)$$

In this way, the output's MRE is classified in error bins to facilitate the analyzing of predictions' quality.

4. Learning the inverse solution

The impacts of activation function, learning rate, batch size, and hyper-parameters on the model's performance are examined for learning the inverse solution. For simplicity, the network is anointed by the number of its nodes and layers; for instance, [3×6] gives a network with six hidden layers, and each layer has three nodes. A dense training dataset is extracted from applicable range (see 2.2) with training samples.

4.1. Numerical experiments

The behavior of three activation functions, Sigmoid (nonlinear) and ReLU and lReLU (piece-wise linear), are scanned. The mean absolute errors of training, MAE, are shown in **Fig. 5**. Network with Sigmoid function approximates MAE 100 μm while one with ReLU or lReLU makes prognoses with less than 10 μm for predicting target holes 150-600 μm.

Sigmoid function is insensitivity to input values larger than 0.7 that conducts near-zero derivatives for these values. Although this behavior is a robust tool for classifying problems, it is a burden when it comes to regression. Further, the deep neural networks with the Sigmoid function suffer from vanishing gradients; the network's parameters do not get revised due to the insignificance of variation of cost function gradients. ReLU and lReLU functions do not struggle with this effect and respond precisely to large inputs.

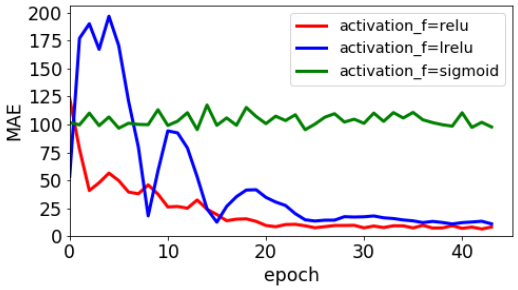

Fig. 5 Training MAE for different activation functions. [3×18] network, batch size = 128, learning rate = 0.01.

Histograms of MRE of ReLU and lReLU illustrate that more than 95% and around 80% of predictions manipulate errors less than 10%, as shown in **Fig. 6**. Hence, ReLU is picked as the preferred activation function for this problem.

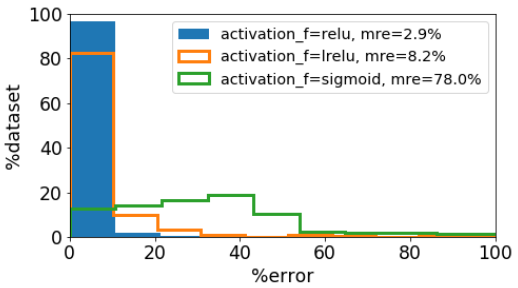

Fig. 6 Histogram of prediction relative error for different activation functions. [3×18] network, batch size = 128, learning rate = 0.01.

Learning rate controls the speed of optimization procedure. A high learning rate leads to significant changes in weights. On the one hand, these considerable evolutions can accelerate optimization. On the other hand, it can move away the training rapidly. A low learning rate provides a slow optimization pace such that the training can never converge to the optimum state.

This study uses an exponential decay function, gradually decreasing the learning rate. In this way, the network profits from a fast-learning rate at the beginning, and then it plunges exponentially to avoid overshooting. The network is examined with four initial learning rates: 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-4}. As MAEs of training plotted at **Fig. 7**, with extreme learning rates (i.e., 10^{-1} and 10^{-4}), the network does not learn due to overshooting and prolonged rate of learning, respectively. Trainings with initial

learning rates 10^{-2} and 10^{-3} conducts to MAE less than 10 µm.

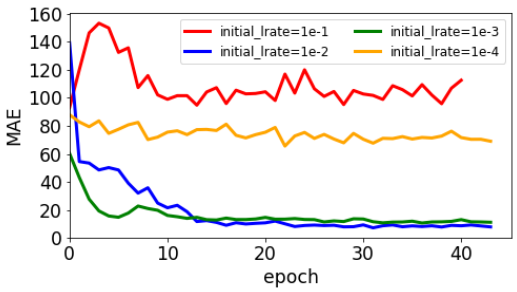

Fig. 7 Training mean absolute error for different initial learning rates. [3×18] network, batch size = 128, activation functions = ReLU.

As pictured in **Fig. 8**, initial learning rate 10^{-3} culminates MRE about 8% and 80% of predictions have relative error less than 10%. The results are enhanced enormously when the initial rate is increased one order of magnitude to 10^{-2}; more than 97% of outcomes have relative error less than 10%.

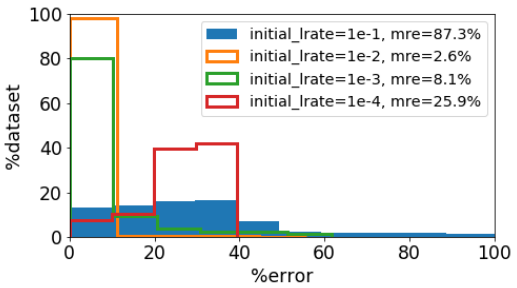

Fig. 8 Histogram of prediction error for different initial learning rate. [3×18] network, batch size = 128, activation functions = ReLU.

The optimization approaches that process the entire samples simultaneously, so-called the deterministic gradient methods, rise the training time significantly and require high computational memory. By dividing the dataset into small clusters (called the minibatch stochastic method), the learning procedure benefits from its regularizing effect, which induces noise. If an individual sample is used at a time (stochastic gradient method), generalization error is presumably the best, but in this case, a small learning rate is required to grab sustainable results [35]; it can slow down training advancement. Thus, it is crucial to find the smallest efficient batch size. As displayed in **Fig. 9**, the MAEs of training operations with very small batch sizes, i.e., 8 and 16, fluctuate intensively without noticeable improvement. By increasing the batch size, the amplitudes of instabilities reduce prominently, and the MAE represents a stable tendency.

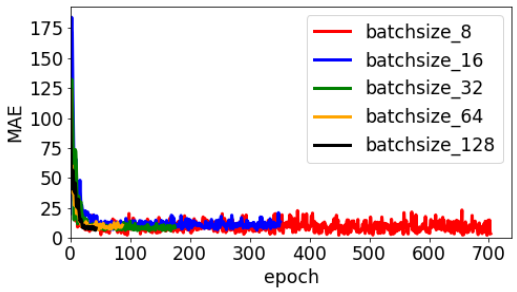

Fig. 9 Training mean absolute error for different sizes of minibatch. [3 × 18] network, learning rate = 0.01, activation functions = ReLU.

The MREs of prognoses are plotted in **Fig. 10**; they express that less oscillating training procedures can reduce the prediction's error to 3%. Here, since the MRE of 128-sample batches is small enough, i.e., 2.9%, 128-sample batch is selected to avoid an unnecessary increase in training time.

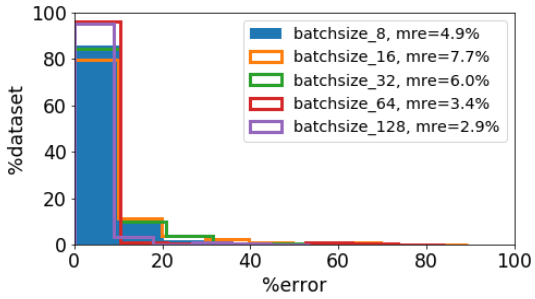

Fig. 10 Histogram of prediction error for different sizes of minibatch. [3 × 18] network, learning rate = 0.01, activation functions = ReLU.

The ANNs are found on a chain-based configuration such that each layer is a function of the previous layer. There is no fixed formula to calculate the proper number of layers and nodes; they are determined in a trial-error procedure. In this study, a uniform architecture for hidden layers is assigned; all of them have the same number of neurons. The optimum structure is obtained by deepening and widening the network and monitoring the average prediction error. The MAE of training is depicted in **Fig. 11**; in this graph, the width of the network remains constant. By increasing the depth of the network from 2 to 16 layers, the prediction accuracy is improved significantly. MAE of training by a 16-layer network decreases to 5.9 µm. By adding more layers, the training error augments gradually to 11 µm for the 24-layer network. A drastic jump is observed in training error when the network is deepened more, meaning the gradient vanishes and the network cannot learn anymore.

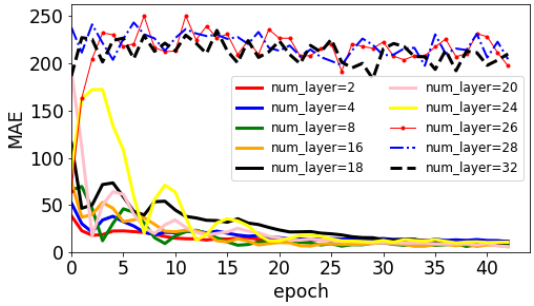

Fig. 11 Training mean absolute error for different numbers of hidden layer. learning rate = 0.01, activation functions = ReLU, batch size = 128.

Furthermore, **Fig. 12** demonstrates that for a 16-layer network, more than 97% of predictions have relative errors less than 10%, and its average relative error is about 2%. It is glimpsed that the MRE is increased gradually by stacking layers. It can be concluded that a 16-layer network is adequately deep to accomplish accurate outputs if benefiting the nonlinear behavior of the network.

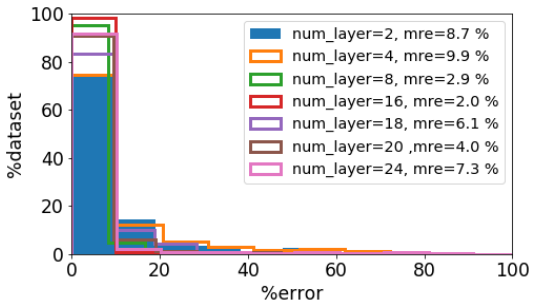

Fig. 12 Histogram of prediction error for different numbers of hidden layer. learning rate = 0.01, activation functions = ReLU, batch size = 128.

Concerning studying the effect of widening of the network, nodes are increased from 3 to 15 while the network's depth is constant. By augmenting the width, the minimum MAE of training growths broadly from 8 µm for 3-node network to 49 µm for 15-node, as depicted in **Fig. 13**. The adverse impact of increasing the network width is also understandable from analyses of relative error. As illustrated in the histogram in **Fig. 13**, with a 3-node architecture, more than 97% of prognoses have relative errors less than 10%; this value declines to 30% for the 15-node network. One reason for this inaccuracy relates to the learning rate; widened network needs a lower learning rate. As discussed before, the lower the learning rate is, the slower the training procedure is, or even it is not feasible. It is worth noticing that although the performances of a [3×2] and [5×2] network are almost the same, they differ remarkably when the network deepens to 16 hidden layers, as shown in **Fig. 14**. Therefore, and [3×16] network is selected as the final structure.

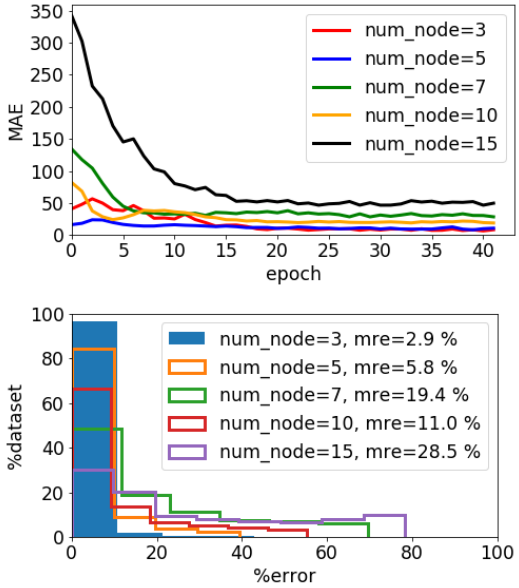

Fig. 13 (top) Training mean absolute error, (down) histogram of prediction error for different number of nodes. hidden layers = 2, learning rate = 0.01, activation functions = ReLU, batch size = 128.

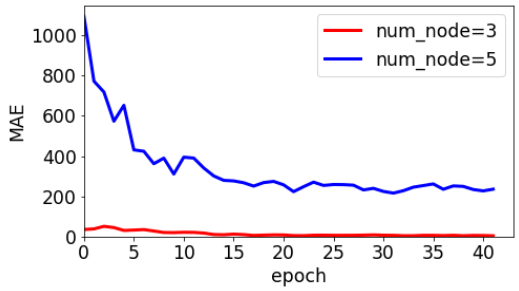

Fig. 14 Training mean absolute error for different number of nodes. learning rate = 0.01, activation functions = ReLU, batch size = 128.

4.2. Discussion

According to the above examinations, a [3×16] feed-forward-back-propagation network is practical to solve the laser drilling problem inversely. The performance of the trained network in exposing unseen dataset is depicted in **Fig. 15** and **Fig. 16** . The unseen dataset is defined once inside the training's parameter range ("unseen") and once outside it ("unseen_ext") to test the network with the extended dataset. Here, training, unseen and unseen_ext datasets contain 5600, 121, 25 samples.

The evaluations indicate that the model can anticipate about 85% of the unseen dataset with relative errors less than 10% and MRE equals 4.6%. Nonetheless, the network functions less efficiently with unseen_ext datasets such that around 65% of predictions are precise and MRE is about 9.5%.

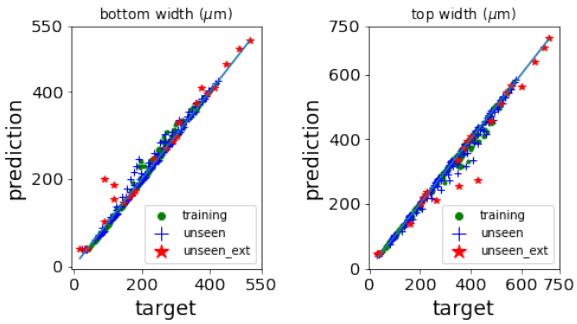

Fig. 15 Performance of model in predicting targets, top and bottom widths. The model is tested with training-, unseen- and unseen_ext-data.

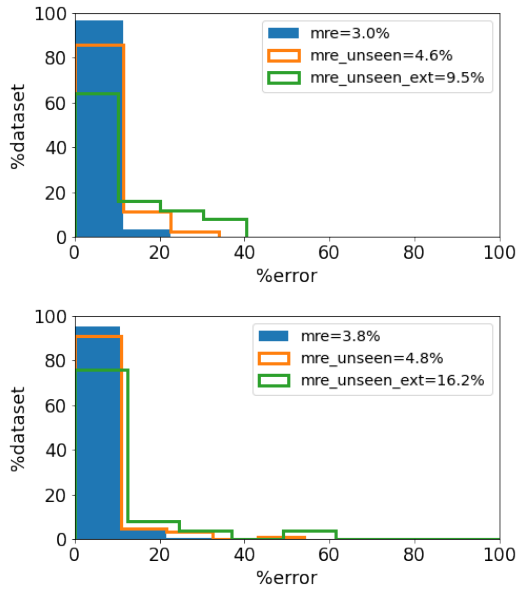

Fig. 16 Histograms of prediction errors of (up) top and (down) bottom widths for training samples, unseen and unseen_ext dataset.

In **Table 2** results of the model for five examples of laser drilling are proposed. The required beam radii (br^p) and pulse powers (pp^p) are predicted for drilling boreholes with preferred top and bottom widths, w_t^t and w_b^t respectively. To check deviation of the predictive and asymptotic models, the top and bottom widths (i.e. w_t^s and w_b^s) of the borehole are also simulated with the anticipated br^p and pp^p as tabulated in **Table 3**. The average difference between the results of predictive and simulating models is about 5 to 8 μm.

Table 2 Evaluating the model with unseen dataset.

$w_t^t(\mu m)$	$w_b^t(\mu m)$	$br^p(\mu m)$	$pp^p(W)$
35	30	14.1	59.8
90	70	29.1	91.6
140	100	45.5	126.4
260	195	90.6	221.9
310	200	100.8	243.6
450	320	155.6	359.9

Table 3 Evaluating the model with unseen dataset.

$br^p(\mu m)$	$pp^p(W)$	$w_t^s(\mu m)$	$w_b^s(\mu m)$
14.1	59.8	44.6	38.4
29.1	91.6	87.5	69.3
45.5	126.4	134.1	100.1
90.6	221.9	263.6	190.8
100.8	243.6	292.8	212.4
155.6	359.9	448.3	325.5

More precisely, the claimed range of 5 to 8 μm represents an average Absolute Error between target and predicted values. In **Table 4** and **5**, the target values and prediction results for top and bottom width of borehole are shown. The difference between predicted and target values varies between 0.1 to 17.2 μm. By calculating the mean absolute errors, we obtain 6,75 and 5,22 μm for top and bottom widths, respectively.

Table 4 Evaluating the model with unseen dataset.

$w_t^t(\mu m)$	$w_t^s(\mu m)$	Absolute Error $= \lvert y_i^t - y_i^p \rvert$
35	44.6	9.60
90	87.5	2.50
140	134.1	5.90
260	263.6	3.60
310	292.8	17.20
450	448.3	1.70
Mean Absolute Error $= \frac{1}{N}\sum_{i=1}^{N}\lvert y_i^t - y_i^p \rvert$		6.75

Table 5 Evaluating the model with unseen dataset.

$w_b^t(\mu m)$	$w_b^s(\mu m)$	Absolute Error $= \lvert y_i^t - y_i^p \rvert$
30	38.4	8.40
70	69.3	0.70
100	100.1	0.10
195	190.8	4.20
200	212.4	12.40
320	325.5	5.50
Mean Absolute Error $= \frac{1}{N}\sum_{i=1}^{N}\lvert y_i^t - y_i^p \rvert$		5.22

5. Conclusion

In this research on solving inverse problems, a neural network is established and trained that determines the laser parameters with which the desired taper of a borehole is achieved. As result, using a physical model improves the solvability of the inverse problem, since all trials during training belong to the applicable range of laser drilling and improves the training procedure going in a physically informed direction. However, the network's performance strongly depends on activation functions and structure of the ANN. Comparing Sigmoid, ReLU, and LReLU reveals that the ReLU function executes convincingly. Additionally, the hyper-parameter learning rate which controls the optimization speed has to be optimized, to

avoid prolonged learning steps by too low learning rates and avoid numerical overshooting by too high values. To relieve required computational memory and time, the training samples were separated into several mini-batches; outcomes reveal that small mini-batches lead to fluctuating training tendencies, albeit their regularization effect could improve general error. Furthermore, networks with small mini-batches require small learning rates to maintain sustainable results. Finally, the proper arrangement of layers and nodes was figured out. The trained neural network was tested with unseen data; where it completes accurately when the unseen data locates inside the training range, while the mean relative error of prediction increases extensively if unseen data are beyond this range.

This work highlights the continuing need for research in the construction of neural networks for solving inverse tasks, which should increasingly use physically informed networks and require improved methods for generating suitable network structures.

6. Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

[1] W. Schulz, U. Eppelt, and R. Poprawe: J. Laser Appl., 25, (2013) 012006.

[2] J. Furmanski, A. M. Rubenchik, M. D. Shirk, and B. C. Stuart: J. Appl. Phys., 102, (2007) 073112.

[3] T. Hermanns, "Interaktive Prozesssimulation für das industrielle Umfeld am Beispiel des Bohrens mit Laserstrahlung", (RWTH Aachen University, Aachen, 2018) p.46.

[4] C. Rackauckas, A. Edelman, K. Fischer, M. Innes, E. Saba, V. B. Shah, and W. Tebbut: Proc. AAAI Spring Symp.: MLPS 2020, (2020) 2587.

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis: J. Comput. Phys., 378, (2019) 686.

[6] L. Bottaci, P. J. Drew, J. E. Hartley, M. B. Hadfield, R. Farouk, P. W. Lee, I. M. Macintyre, G. S. Duthie, and J. R. Monson: The Lancet, 350, (1997) 469.

[7] M. Coccia: Tech. in Society, 60, (2020) 101198.

[8] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage: Ecological Inform., 40, (2017) 50.

[9] E. Díaz, V. Brotons, and R. Tomás: Soils and Found., 58, (2018) 1414.

[10] J. Cho and S. Song: Appl Therm. Eng., 181, (2020) 115974.

[11] P. Brandstetter and M. Kuchar: J. Appl Log., 24, (2017) 97.

[12] X. Ren, J. Fan, R. Pan, and K. Sun: Int. J. Adv. Manuf. Tech. , 127, (2023) 1177.

[13] K. Mathivanan and P. Plapper: Procedia CIRP, 103, (2021) 61.

[14] B. Acherjee, S. Mondal, B. Tudu, and D. Misra: App Soft Comput., 11, (2011) 2548.

[15] G. Casalino, F. Facchini, M. Mortello, and G. Mummolo: IFAC PapersOnLine, 49, (2016) 378.

[16] M. Rahimi, M. Shayganmanesh, R. Noorossana, and F. Pazhuheian: Opt. Laser Tech., 112, (2019) 65.

[17] S. Jović: Opt., 183, (2019) 766.

[18] M. A. Mahmood, A. C. Popescu, M. Oane, A. Channa, S. Mihai, C. Ristoscu, and I. N. Mihailescu: Results in Phys., 26, (2021) 104440.

[19] Z. Zhu, K. Ferreira, N. Anwer, L. Mathieu, K. Guo, and L. Qiao: Proc. CIRP, 91, (2020) 534.

[20] X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li: Eng., 5, (2019) 721.

[21] D. Chen, P. Wang, R. Pan, C. Zha,J.Fan,S. Li, and Kai Cheng: Weld. World, 66, (2022) 409.

[22] S. Chaudhry and A. Soulaimani: Proc. CIP 2022: Adv. Integr. Des. Prod. II, (2022) 155.

[23] S. K. Dhara, A. S. Kuar, and S. Mitra: Int. J. Adv. Manuf. Tech., 39, (2007) 39.

[24] G. Casalino, A. M. Losacco, A. Arnesano, F. Facchini, M. Pierangeli, and C. Bonserio: Proc. CIRP, 62, (2017) 275.

[25] S. Chatterjee, S. S. Mahapatra, V. Bharadwaj, A. Choubey, B. N. Upadhyay, and K. S. Bindra: Proc Inst. Mech. Eng., Part B: J. Eng. Manuf., 233, (2018) 1872.

[26] G. Baiocco, S. Genna, C. Leone, and N. Ucciardello: Int. J. Adv. Manuf. Tech., 114, (2021) 1685.

[27] S. Mishra and V. Yadava: Opt. Laser Tech., 48, (2013) 461.

[28] K. Liao, W. Wang, X. Mei, W. Tian, H. Yuan, M. Wang, and B. Wang : J. Intell. Manuf. , 34, (2023) 2907.

[29] D. Karnam, Y. Lo, and C. Yang: J. Mater. Res. Tech., 25, (2023) 3712.

[30] D. P. Kingma and J. Ba: Proc. ICLR, (2015) 1.

[31] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind: J. Mach. Learn. Res., 18,(2018) 1.

[32] D. Z. Huang, K. Xu, C. Farhat, and E. Darve: J. Comput. Phys., 416, (2020) 109491.

[33] E. Rollins, "Optimization of neural network feedback control systems using automatic differentiation", PhD thesis, (Massachusetts Institute of Tech., 2009) p.29.

[34] T. Developers, "TensorFlow," Zenodo, https://www.tensorflow.org.

[35] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning", (MIT Press, 2016) p.302.

(Received: June 8, 2023, Accepted: March 7, 2024)