DEEP LEARNING IN SHEET-BULK METAL FORMING PART DESIGN

C. Sauer, B. Schleich and S. Wartzack

Abstract

Within the Transregional Collaborative Research Centre 73, a self-learning engineering workbench is being developed. It assists product developers in designing sheet-bulk metal formed (SBMF) parts by computing the effects of given product and process characteristics on the product properties. This contribution presents a novel approach to using deep learning methods for the properties prediction. By making use of a parameter study of 20 SBMF part designs, a metamodel is trained and used to predict the total equivalent plastic strain on local level as an indicator for part manufacturability.

Keywords: data mining, data driven design, characteristics and properties, deep learning

1. Introduction

In the light of global warming, conservation of resources and lightweight part design are key issues for the coming product generations. These issues are tightly entangled with the design of highly function-integrated parts as well as their efficient production. To achieve this, modern product developers use methods of Design for X, such as design for manufacturing, design for resource efficiency and design for function integration. However, in recent years and against the background of steadily increasing product and process complexity, established design rules and guidelines in the context of design for X proved to be insufficient (Wartzack et al., 2017).

In addition to that, the amount of data generated while designing new products also rose to a level, which leads experts to speak of it as "Big Data". Therefore, modern product developers are in a dilemma. On the one hand, the design process switched from good/bad design rules to a more complex system and first and foremost to a multiobjective optimization problem. On the other hand, designers have as much information about designs and processes as never before.

This dilemma has led to the development of so-called engineering workbenches, which support product developers in their design work by providing them with the information about the product at the right time. They are an additional source and tool for handling knowledge and information at any given time. These tools are mainly based on the adoption of Data Mining algorithms, which try to get a grip on the massive amounts of data that are produced while designing products. The employed algorithms provide product developers with predictions about the product behaviour using the data that is generated during the development, e.g. from process experts or manufacturing scientists.

Within the Transregional Collaborative Research Centre 73 (SFB/TR 73), such an engineering workbench is developed. It is supposed to assist product developers in designing parts produced by a new type of forming process called sheet-bulk metal forming (SBMF), which combines bulk-forming operations with sheet metals to create lightweight parts. With reduction of mass and heavily loaded functional elements, SBMF both exceeds the limits of existing technologies and provides a high

throughput in terms of the number of produced workpieces (Hetzner et al., 2011). Against the background of resource efficiency, this is a strong argument for the further development of SBMF.

Concerning the development of engineering workbenches, new data mining algorithms prove themselves as suitable for the use in such tools. In particular, neural networks and deep learning are developed as new prediction algorithms. However, their training requires vast amounts of data, which should ideally be in the form of structured tables (Witten et al., 2017). By the existence of such data in SBMF, neural networks, especially deep learning, show themselves valuable in the development of the engineering workbench for sheet-bulk metal forming. This contribution explores the possibility of the adoption of such algorithms in current generation engineering workbenches. The novelty of the contribution can be found in the utilization of deep learning methods for the manufacturability analysis on local part level considering new forming technologies.

The paper is structured as follows. In the next section, a brief background regarding sheet-bulk metal forming, knowledge acquisition through data mining, the engineering workbench SLASSY, current methods for knowledge representation, and deep learning is given. After this, the possibilities for deep learning in sheet-bulk metal forming are explored. Based thereon, the revision of the existing engineering workbench for SBMF is elaborated. Finally, a conclusion and an outlook are given.

2. Background and related work

Before highlighting the possibilities for deep learning in the innovative forming technology sheet-bulk metal forming as well as elaborating the adoption of the existing engineering design workbench SLASSY, this section is to provide the reader with all relevant background information.

2.1. Sheet-bulk metal forming

The overall objective of SBMF is the development of a new forming process technology to manufacture heavily loaded functional elements on sheet metal parts with tight geometrical tolerances (see SBMF parts in Figure 1). For this purpose, the sheet operations are extended by bulk-forming operations within sheet thickness. In general, sheet-bulk metal formed parts consist of two basic elements: The primary design elements (PDE) that essentially represent sheet parts (manufactured by sheet operations) and the secondary design elements (SDE), which provide the high function integration (Breitsprecher and Wartzack, 2013). By using this manufacturing technology, product developers are empowered to create locking tooth or deep drawn parts with pockets or sheet metal parts with functional elements (see Figure 1).
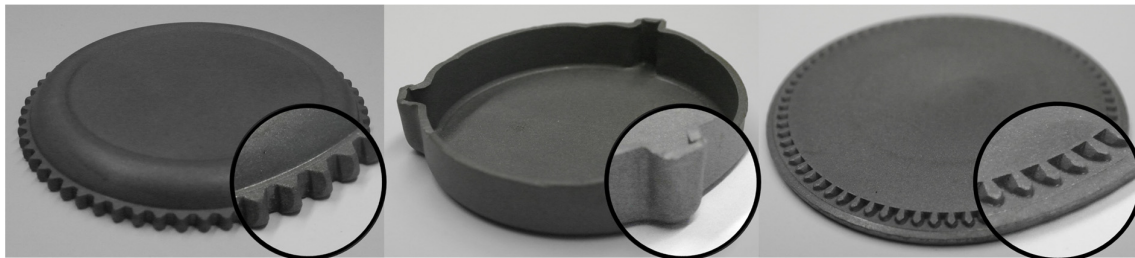

Figure 1. Examples of sheet-bulk metal formed parts

For designing parts, product developers need to know the constraints of the underlying manufacturing technology. Therefore, the established design approach is to follow published guidelines or to talk directly to the manufacturing experts about a specific design. In the context of product development for sheet-bulk metal forming, there are no design rules available yet. Engineers or scientists (experts) are not able to give specific guidelines or design rules for now. Furthermore, important criteria are often conflicting. We can summarize the following aspects:

- Manufacturing technologies being developed do neither have any experts nor good/bad rules (no explicit manufacturing knowledge)
- experts criteria for evaluation of the manufacturability of a design are vague and could be conflicting (implicit manufacturing knowledge)

One key aspect of the research in this field is to tackle these problems by developing an efficient self-learning engineering workbench, which can support the decisions of product developers in designing new parts.

2.2. Acquisition of knowledge through data mining

In order to support the "Design for sheet-bulk metal forming" approach, relevant and existing manufacturing knowledge is required and needs to be acquired. In this context, knowledge acquisition methods can be distinguished as direct and indirect knowledge acquisition methods, which both make use of experts who can either enter their knowledge in expert systems (direct knowledge acquisition) or are interviewed by knowledge engineers (indirect knowledge acquisition) who in turn do the work of entering and characterising relevant knowledge (Spur and Krause, 1997). However, for new manufacturing technologies, those experts are simply not existent.

In contrast to that, a third knowledge acquisition technology can be found as automatic knowledge acquisition, which aims at "generating knowledge from data" using data mining methods. Such data is mainly created by experimental and numerical parameter studies during the development process for new forming technologies such as sheet-bulk metal forming.

To support such parameter studies, process developers make use of the Design of Experiments methodology, which ensures a better exploration of the available parameter space, thus producing a solid database for further investigations (Siebertz et al., 2010). The parameter space is setup up by all product features that are defined by the product developers. After setting up a design of experiment, the parameter studies are carried out. The results of those studies are stored in a database (e.g. Microsoft Excel spreadsheets), this marks the point for manufacturing experts to evaluate the data and further develop an understanding of the new manufacturing technology.

The self-learning process (see Figure 2) is the last step of discovering knowledge in the database. After the creation of the database from parameter studies, the self-learning process implemented in an engineering workbench creates metamodels. Those models describe the underlying data by introducing a relation between the input values (in this case; product features) and the output values (in this case; product properties). As Breitsprecher and Wartzack (2012) point out, the term self-learning is used because every time the database is updated, a new metamodel creation is initialised. Moreover, different types of metamodels compete against each other and the one, which describes the data best, is used for the prediction of product properties. Hence, the metamodels used for description are always up to date and use all currently available data (Breitsprecher and Wartzack, 2012).
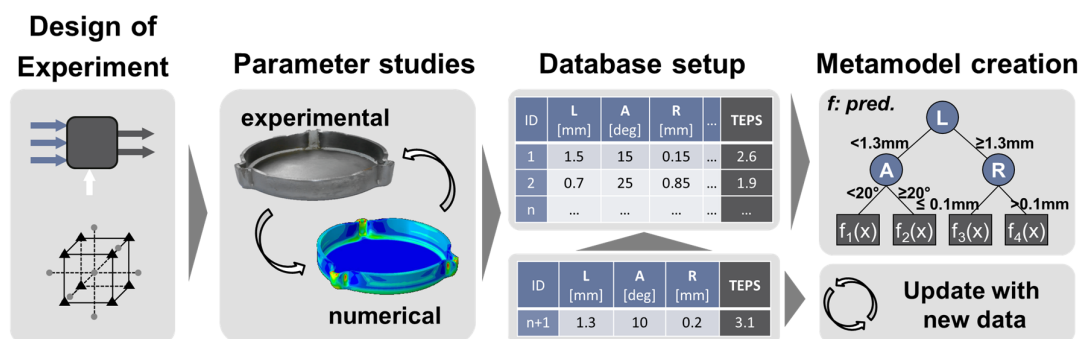

Figure 2. Knowledge discovery process (Wartzack et al., 2017)

2.3. A self-learning engineering workbench - SLASSY

As mentioned before, the objective of the research project is to develop an engineering workbench to support product developers in sheet-bulk metal forming. This engineering workbench is called SLASSY (German acronym for self-learning assistance system). It supports product developers in the design process of sheet-bulk metal formed parts. It consists of two main modules. One module dedicated to the synthesis of new part designs, the other dedicated to analyse product properties of the current part design.

Figure 3 shows the user interface of SLASSY's synthesis tool. Product developers can configure their sheet-bulk metal formed parts and can set all available product features. By making use of the previously described data mining algorithms, metamodels run an analysis of the current part design and calculate product properties based on the current part design. Metamodels are created from the database, which holds all the data from the parameter studies. The whole analysis process is shown in Figure 4.
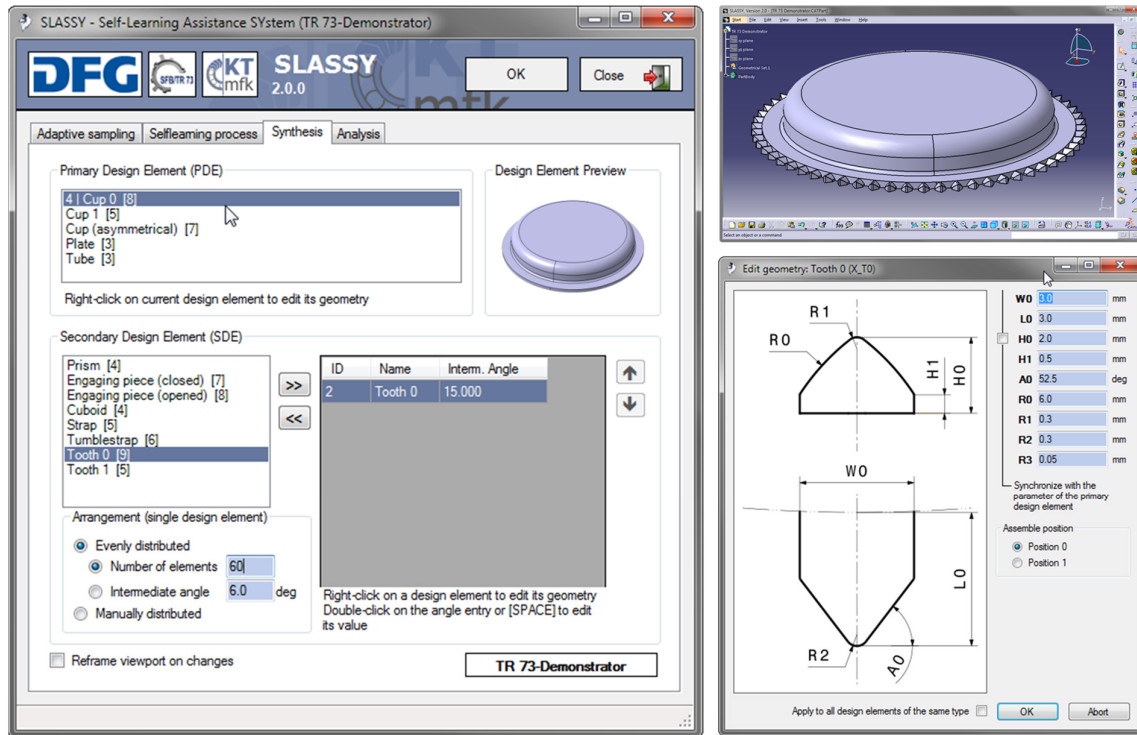

Figure 3. Design synthesis tool of SLASSY

The design analysis tool from Figure 4 makes use of the current part design (set by the product developer) in the design synthesis tool. Moreover, the knowledge base, which consists of all available data from the parameter studies, is tapped by making use of the generated meta-models. Together both parts are used to predict product properties within the design analysis tool (Wartzack et al., 2017). Product developers are able to see the output for the relevant product properties. Beforehand, the relevant product characteristics to be analysed were given by manufacturing experts. This process can happen directly or indirectly via the knowledge engineer.
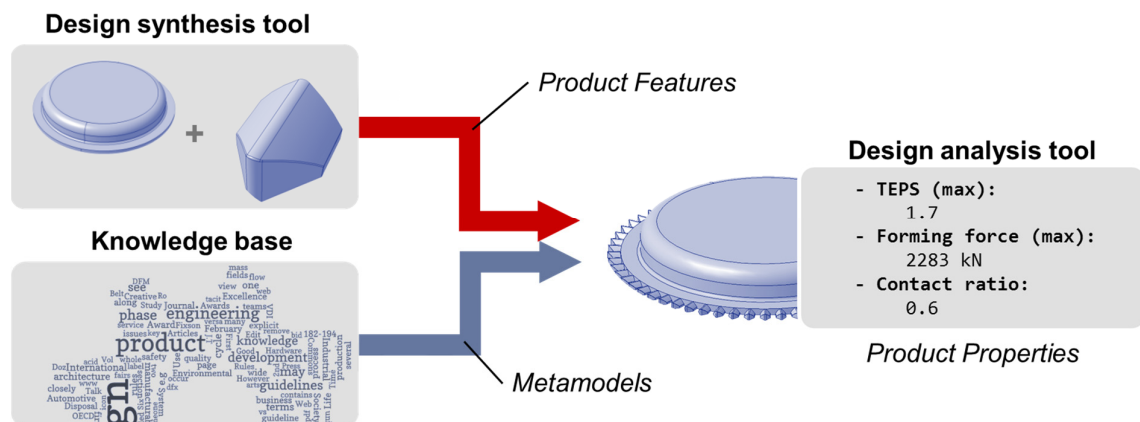

Figure 4. Design analysis tool of SLASSY

2.4. Current models for knowledge representation

The key models for knowledge representation in the engineering workbench are meta-models. Those meta-models can be characterised by their specific mathematical formulation. Figure 5 shows the currently used mathematical constructs for meta-models.
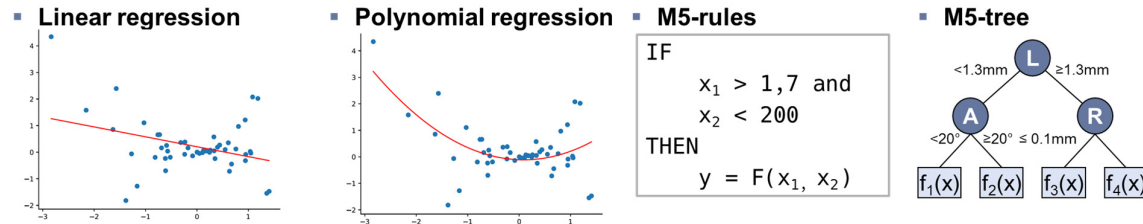

Figure 5. Current metamodels implemented in SLASSY

Two of the four used metamodel types make use of mathematical regression. On the one hand linear regression, which uses a straight line to interpolate between data points in the database. On the other hand polynomial regression that approximates data points by a polynomic curve of arbitrary order.

The other two metamodel types are based on an algorithm called M5 algorithm. Rule-based as well as tree based metamodels make use of piecewise defined linear models. The M5-algorithm supports the definition of distinct pieces where linear models are set up and provides the utilization for the prediction from rule or tree based models (Quinlan, 1992).

Current metamodels only support the prediction of product properties for the whole part level, as can be seen from Figure 4. For the future, product developers and manufacturing experts need predictions with a much higher level of detail. Therefore, the resolution of the used data needs to be denser and more data is necessary to generate metamodels. Furthermore, those models must be capable of predicting product properties on local level. Current generation metamodels are not able to handle those amounts of data (Sauer et al., 2017). Research shows that we need to look for other data mining algorithms to generate metamodels that are able to predict product properties for every position (local level) on the current product design.

2.5. A deep learning approach to metamodels

As mentioned above, current metamodels like linear regression or M5-trees cannot handle the amounts of data needed for predicting part properties on local level. In order to overcome this shortcoming, a current trend in machine learning is trying to employ neural networks for the purpose of prediction models. Neural Networks are complex mathematical constructs that are modelled according to the human brain. They consist of an arbitrary number of neurons. Those neurons are arranged in an again arbitrary number of layers. First, the input layer, which represents the inputs of any neural network. Secondly, there can follow one or more hidden layer, which do not have any input from the outside but are only connected to neurons from the input layer. In these layers, highly nonlinear information is processed and forwarded to the next layer. After one or more hidden layers, an output layer follows, which represents the prediction of the neural network.

As Hoffmann (1993) points out there are types of neural networks, which do not have any hidden layers. Those types are simply called perceptrons. On the left side in Figure 6, an exemplary structure of a neural network is described graphically. When the neurons of each layer are fully connected to the next layer in the neural network, we speak of a feed forward type of neural network (Hoffmann, 1993).

From the graphical structure of a neural network, one can always derive the mathematical formulation of the network. This can be formalized as a large matrix of functions, which is graphically described in Figure 6 on the right hand side. This fact shows why predictions from neural networks run much faster on parallelized hardware. If we look deeper in one neuron of a neural network, we get the graphical description from Figure 7.
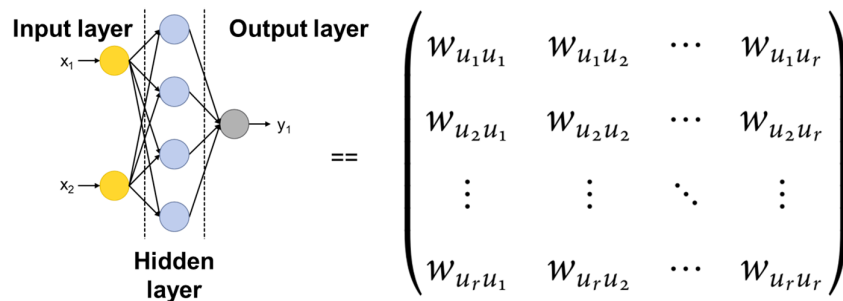
Figure 6. A simple neural network with one hidden layer

$$== \begin{pmatrix} w_{u_1 u_1} & w_{u_1 u_2} & \cdots & w_{u_1 u_r} \\ w_{u_2 u_1} & w_{u_2 u_2} & \cdots & w_{u_2 u_r} \\ \vdots & \vdots & \ddots & \vdots \\ w_{u_r u_1} & w_{u_r u_2} & \cdots & w_{u_r u_r} \end{pmatrix}$$

As described by Witten et al. (2017) we see inputs from the left hand side into the Netinput function, which processes all previously mentioned inputs. Moreover, one can define it from the outside. In most cases, this is a simple weighted summation function. The value of the Netinput function propagates further to the activation function. This is the part where the neuron "fires", similarly to the neuron in the human brain. The activation function upswings when the processed values pass an arbitrary threshold value. After this is done, external factors, such as the node bias, can manipulate the propagated values. Lastly, the Output function processes its input and returns the final output of the neuron (Witten et al., 2017).
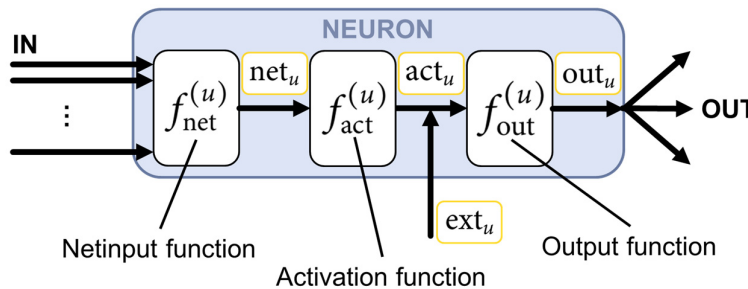

Figure 7. Internals of a neuron in neural networks

Coined by Goodfellow et al. (2016), Deep Learning extends the ideas of neural networks. To be more precise, it defines special types of networks as Deep Learning networks. For this contribution, we speak of deep learning when neural networks with two or more hidden layers are used. On the other hand, deep learning means that vast amounts of data are processed within the training and prediction processes inside the neural networks (Goodfellow et al., 2016). This is exactly the case for the efforts made in this contribution.

Moreover, the computational power of high performance computing systems is on a continuous rise. With the adoption of graphical processing units as accelerators, the parallel deep learning methods also increase in speed and accuracy (LeCun et al., 2015).

The datasets that are used for deep learning have special requirements. First, they need to be as big as possible. For neural networks, one needs much more data to train the metamodel, than for relatively simple interpolation tasks (Kruse et al., 2015). As a starting point, hundred times the number of used input neurons is a minimum requirement for creating significant metamodels.

Second, the dataset needs to have a structure, which allows to derive an input-output relation, i.e. a structured dataset based on which the deep learning neural network can be trained. In many cases, especially when neural networks are used to classify data, it needs to be structured first. Third, the dataset must support batch processing. An ideal dataset is a structured table in a machine-readable format, e.g. Excel spreadsheets or comma separated values.

In order to optimize not only the dataset used for deep learning, but also to increase the prediction quality of the used metamodels, one can adopt the ideas of Zoph and Le (2016). This approach tries to optimize the neural network structure to represent the input-output-correlation with more precision with respect to a predefined performance value, e.g. standard deviation between ground truth and predicted value (Zoph and Le, 2016).

SYSTEMS ENGINEERING AND DESIGN

3. Deep learning in sheet-bulk metal forming

For sheet-bulk metal forming, we employ above-mentioned concepts to create a metamodel for the prediction of local level product properties. Based on a parameter study of a locking teeth model we use deep learning to create a metamodel for the prediction of the TEPS, an acronym for total equivalent plastic strain. The total equivalent plastic strain is one indicator for the formability and thus for the manufacturability of the current part design. It is used to explain the maximum strain a given part design can produce. However, it is not the whole story for describing the manufacturability. This is always a multiobjective system, where many different goals need to be respected.

By using all available local level data of the parameter study, we create a metamodel that is able to predict local level product properties like the TEPS, for current part designs proposed by the product developer. Figure 8 shows this approach in detail.
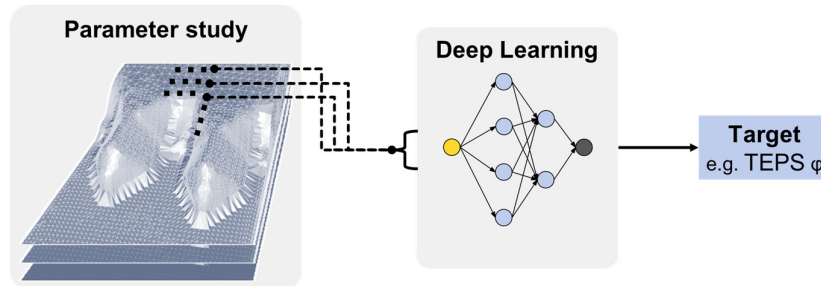

Figure 8. Approach for prediction of product properties on local level (Sauer et al., 2017)

In the process of creating the metamodel, we use the whole parameter study as the database. We train our deep learning model based on this data. For evaluation purposes, we split the data in training and test data. Based on the test data, we evaluate the quality of the prediction. For this purpose, the R^2-Coefficient of Prognosis (R^2-COP) is used (Most and Will, 2008). Furthermore, we calculate the standard deviation for the test data.

3.1. Dataset

The process of creating the dataset used for this contribution can be described as the following steps. At first, we need to identify product properties that are relevant to the considered output value. On the left hand side in Figure 9, one can see two chosen properties of a primary design element, formed by sheet-bulk metal forming. In this case, we use the height and breadth of a locking tooth. Both properties are then sampled in a Latin Hypercube Sampling (LHS) to create a trial design. This trial design defines the design of experiment. As McKay et al. (1979) point out, one characteristic of Latin Hypercube Sampling is that, with respect to a Gaussian distribution every row and each column of the design space gets a trial point. The design space can be defined as all possible product property configurations a product developer can define.
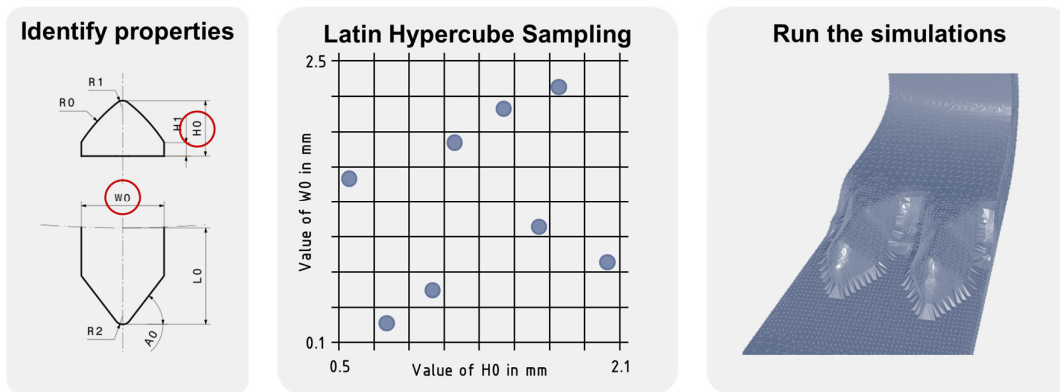

Figure 9. Latin hypercube sampling in design of experiments

After setting up the design of experiments, the simulations need to be carried out by using the finite element method. Every design of the created trial design, the so-called variant of a parameter study needs to be evaluated to create the complete parameter study. In this case, the parameter study consists of 20 different variants of a locking teeth design. The created data is arranged in the human and machine-readable comma separated values format. Those 20 different variants were previously run inside a forming simulation. The different variants of the parameter study were set up by using Latin Hypercube Sampling.

For every variant, the complete node data of the mesh is stored for each node separately. On top of that, every timestep of the iteration also creates a csv-file for all available nodes in the mesh. The used FEM meshes have around 10,000 data points. At each node, we extract the corresponding total equivalent plastic strain value for further analysis. Furthermore, every node has a distinct node ID to identify corresponding nodes between the time steps of the simulations. Table 1 shows the obtained data structure that is created by the forming simulation and later used for the training of metamodels.

Table 1. Resulting data structure for sheet-bulk metal forming parameter study

	Node ID	X-coordinate	Y-coordinate	Z-coordinate	TEPS
Type	*Integer*	*Floating point number*			

It can be seen, that the first column holds the specific node id number to identify each node. The next three columns hold the coordinates of each node based on the coordinate system used inside the forming simulation. In the last column, the desired output, in this case the TEPS is stored. By using this structure, it is easier to run the steps of the training process of the metamodel. In addition, we have directly modelled the input-output-correlation inside the dataset.

3.2. Results

After splitting the data using a train-test-split and training the metamodel with the split training data, we calculate the overall quality of prediction based on the following equation.

$$R^2 CoP = \left(\frac{\sum_{i=1}^{n}(y_p - \bar{y}_p)(y_t - \bar{y}_t)}{(n-1)\sigma_p \sigma_t} \right)^2 \tag{1}$$

Predicted values are arranged in a vector called y_p. Its mean value is calculated and described by the barred y_p. The true target values of the test samples are stored in the vector y_t. Following the same scheme as the predicted values, barred y_t represents the mean of the true target values. On the other hand, the standard deviations of prediction and the true target are described by the sigma variables. The total count of data samples is represented by the variable n. The main advantage of the calculated performance value is its scalability to the size of the data, by employing the total number of samples and calculating their performance with respect to their root mean squared error (Most and Will, 2008). Based on this measure, we calculate the R^2-COP to obtain the prediction quality for the deep learning metamodels for local level prediction. For all variants inside the parameter study, the resulting R^2-COP value does not decrease below 85%. This means that a previously unused position is predicted correctly with a chance of 85%. Moreover, the resulting root mean squared error for the total equivalent plastic strain value are around 0.2-0.3.

For training our models, we shuffled the dataset and used a 20-80 test-train data split. During the training, we tried to avoid overfitting by adopting two methods. Overfitting describes an effect where the model is only adapted to the used training data in a way, where it is not able to predict previously unknown data. On the one hand, carefully watching the training and test accuracy to detect when the training accuracy converges and stop the training process. If the training accuracy converges and training continues, the model is more likely to be overfitted. Preventing this is key in creating good prediction models for previously unknown data points. The second method to prevent this from happening in the process of determining the neural network structure, one can apply dropout techniques introduced to

obtain prediction models with good performance in previously unused new data (Srivastava et al., 2014). As a conclusion, it can be seen, that the overall prediction quality is very good and shows potential for a local level description of target values. This led us to the conclusion of adopting it in the general workflow using the SLASSY engineering workbench.

4. Extending the self-learning engineering workbench SLASSY

The existing workflow in SLASSY can be described as the following steps. First, the product developer used the synthesis part to create a new design. During this, developers can define all necessary geometric values for a given design of a primary design element. Those elements provide the functional parts of the sheet-bulk metal formed part. After that, a global analysis can be made. Previously product developers got only one value for their current design. This means that they get no feedback where on the design the predicted value arises. This means, they had to guess when they moved back from the analysis to the synthesis of a better design. By now extending this workflow through adding the local level metamodel tool, they now can evaluate the desired product properties on the local level of the current part design. Henceforth this increases the functionality of the "design for manufacturing"-engineering workbench SLASSY. Taking into account the possibilities provided by the above mentioned methods like deep learning and local level predictions. In this context, the new workflow extends the previous one, inasmuch as allowing a local level prediction of product properties. So far, product developers had the possibility to get a prediction for their whole part designs inside the analysis part of SLASSY. Now they are able to get local level product properties by employing the following new workflow.

Product developers, after their initial synthesis of the design, can now use the analysis tool of the engineering workbench to make use of the local metamodel tool. Implemented in the workflow, as shown in Figure 10 on the right hand side, is a second lane to generate local metamodels from the database. Based from the above-mentioned parameter studies, one monolithic database holds all available data for the training of the local metamodels. In the event of a completely new part design, the algorithm selects the closest possible metamodel for the current new design.
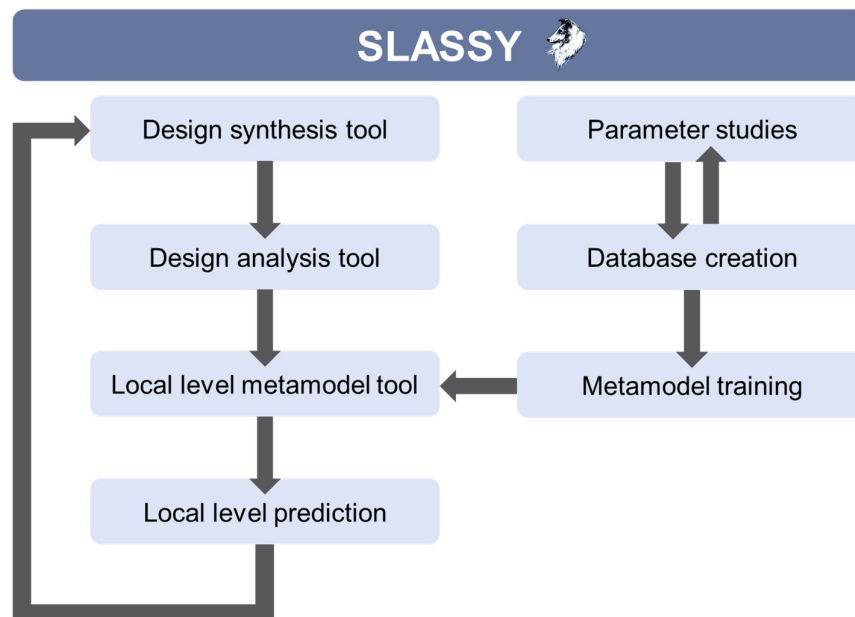

Figure 10. Workflow for the product developer

The prediction always provides its R^2-COP value and the corresponding mean standard deviation, to give the product developers a feel for the prediction quality. Based on the resulting prediction values, the product developers can now make assumptions about the resulting product properties and decide to optimise the design with further use of local metamodels.

Figure 11 shows the interactive user interface that is created by the local metamodel tool. By directly interacting with the graphical user interface (GUI), the product developers can see the resulting values in the prediction. Moreover, the model itself can be colorized in form of a heat map for clearly displaying possible critical areas, which need more attention based on their local product properties. Based on the heat map approach, product developers get a feedback regarding the bigger picture inside their current product design and directly see the distribution of the product properties in local part areas.

Figure 11 shows the resulting dialogue component. This is an important building block for the engineering workbench. As mentioned by Altenkrüger and Büttner (1992), the visual feedback and the possibility to modify the design directly is an important part of engineering workbenches or expert systems. The presented approach implements those ideas. The user can select the different locations of interest and the workbench provides him with a prediction for the currently selected point inside of the engineering workbench. Given this information, product developers can go back to the synthesis part of SLASSY and create a new part design.
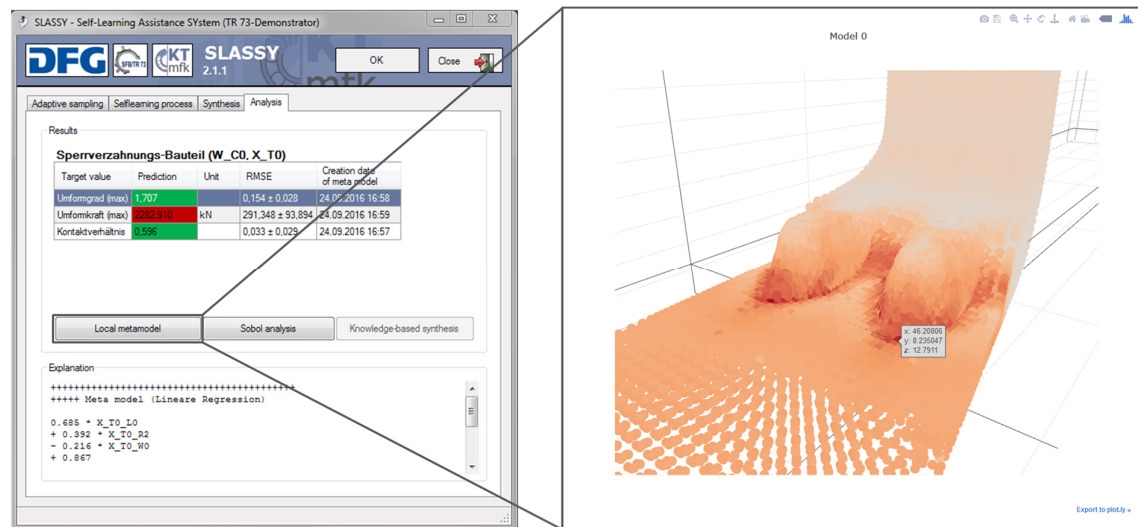

Figure 11. Realisation of dialogue component in the engineering workbench

5. Resume and outlook

The increasing need for resource efficiency and lightweight design puts product developers under immense pressure. In this context, the need for parts manufactured by manufacturing technologies with steadily increasing complexity dictates the use of engineering workbenches, which enable product developers a quick and reliable assessing of the effects of design changes on the manufacturability of their designs. Besides more classical expert systems like the ones described by Boothroyd (1996) which focus on providing practical knowledge at the time of designing parts. The self-learning engineering workbench SLASSY provides prediction methods for product properties based on the current design. Previous works in this domain focused on the prediction of whole part level (global) product properties, the presented approach makes it possible to predict product properties on local function element level. Thus, the presented work overcomes the shortcomings of existing works, which were limited to the prediction of product properties in sheet-bulk metal forming for the whole part. As mentioned above, this was a logical consequence of the used level of detail in the database and the used methods for generating metamodels for prediction.

By employing the whole database approach for training deep learning neural networks, we try to create prediction models for the local level. This is made possible by a very high level of detail in the data on the one hand. On the other hand, new data mining methods are employed. Especially neural networks show their suitability for this. In the proposed approach, they are extended by adopting deep learning for their structure design. By using hidden layers and adjusting the properties of the neurons, we create neural networks capable of predicting the total equivalent plastic strain of sheet-bulk metal formed parts

with respect to the R^2-COP value and the mean standard deviation. Both performance values undermine the suitability of this approach for the prediction of product properties on local level, e.g. the primary design elements that later provide the key functionality of the designed sheet-bulk metal formed parts. The existing workflow inside the engineering workbench SLASSY is extended by a local metamodel tool. This tool comprises a dialogue component and thus supports the design process in predicting local level product properties. For the further development of the new manufacturing technology sheet-bulk metal forming, this prediction represents a key element to support the further research of manufacturing scientists and production engineers.

Beside this, the presented methods show potential for applications in other issues of the project's research, e.g. tribology. Hetzner tried to improve sheet-bulk metal forming processes by local adjustment of tribological parameters (Hetzner et al., 2011). It is conceivable that the local metamodels can be applied in different types of disciplines. One strong requirement is however the availability of different data with a relatively high level of detail. For rapid prototyping purposes, this can be created by numerical parameter studies, which are later validated by mechanical real world analysis.

Acknowledgement

References

Altenkrüger, D. and Büttner, W. (1992), *Wissensbasierte Systeme: Architektur, Entwicklung, Echtzeitanwendungen - Eine praxisgerechte Einführung*, Vieweg+Teubner Verlag, Wiesbaden. https://doi.org/10.1007/978-3-322-86811-4

Boothroyd, G. (1996), "Design for Manufacture and Assembly: The Boothroyd-Dewhurst Experience", In: Huang, G.Q. (Ed.), *Design for X*, Springer, Dordrecht, pp. 19-40. https://doi.org/10.1007/978-94-011-3985-4_2

Breitsprecher, T. and Wartzack, S. (2012), "Architecture and realization of a self-learning engineering assistance system for the use within sheet-bulk metal forming", *Proceedings of NordDesign 2012, the 9th NordDesign Conference, Aalborg, Denmark, August 22-24, 2012*, The Design Society, Glasgow.

Breitsprecher, T. and Wartzack, S. (2013), "A classification system for secondary design features for the use within sheet-bulk metal forming", *Proceedings of the 19th International Conference on Engineering Design (ICED '13), Seoul, South Korea, August 19-22, 2013*, The Design Society, Glasgow, pp. 39–48.

Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep learning*, The MIT press, Cambridge, MA, USA.

Hetzner, H., Koch, J., Tremmel, S., Wartzack, S. and Merklein, M. (2011), "Improved sheet bulk metal forming processes by local adjustment of tribological properties", *Journal of Manufacturing Science and Engineering*, Vol. 133 No. 6, pp. 061011. https://doi.org/10.1115/1.4005313

Hoffmann, N. (1993), *Kleines Handbuch Neuronale Netze - Anwendungsorientiertes Wissen zum Lernen und Nachschlagen*, Vieweg+Teubner Verlag, Wiesbaden. https://doi.org/10.1007/978-3-322-91565-8

Kruse, R., Borgelt, C., Braune, C., Klawonn, F., Moewes, C. and Steinbrecher M. (2015), *Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*, Springer, Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-10904-2

LeCun, Y., Bengio, Y. and Hinton, G. (2015), *Deep Learning, Nature*, Vol. 521, pp. 436-444. https://doi.org/10.1038/nature14539

McKay, M.D., Beckman, R.J. and Conover, W.J. (1979), "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code", *Technometrics*, Vol. 21 No. 2, pp. 239-245. https://doi.org/10.2307/1268522

Most, T. and Will, J. (2008), "Metamodel of Optimal Prognosis - an automatic approach for variable reduction and optimal metamodel selection", *Proceedings of the Weimar Optimization and Stochastic Days 5.0, Weimar, Germany, November 20-21, 2008*, Dynardo Software and Engineering GmbH.

Quinlan, J.R. (1992), "Learning with continuous classes", *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, Hobart, Tasmania, Australia, November 16-18, 1992*, World Scientific, Singapore, pp. 343-348.

Sauer, C., Küstner, C., Schleich, B. and Wartzack, S. (2017), "Einsatz von Deep Learning zur ortsaufgelösten Beschreibung von Bauteileigenschaften", *Design for X: Beiträge zum 28. DfX-Symposium, Bamberg, Germany, October 4-5, 2017*, TuTech Verlag, Hamburg, pp. 49-60.

Siebertz, K., van Bebber, D. and Hochkirchen, T. (2010), *Statistische Versuchsplanung - Design of Experiments (DoE)*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-05493-8

Spur, G. and Krause, F.-L. (1997), *Das virtuelle Produkt: Management der CAD-Technik*, Carl Hanser Verlag, München, Wien.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), "Dropout: a simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, Vol. 15 No. 1, pp. 1929-1958.

Wartzack, S., Sauer, C. and Küstner, C. (2017), "What does Design for Production mean? - From Design Guidelines to Self-learning Engineering Workbenches", *Proceedings of the 11th International Workshop on Integrated Design Engineering (IDE Workshop), Magdeburg*, Germany, April 5-7, 2017, pp. 93-102.

Witten, I.H., Frank, E., Hall, M.A. and Pal, C. (2017), *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, Burlington, MA.

Zoph, B. and Le, Q.V. (2016), *Neural architecture search with reinforcement learning*. [online] arXiv. Available at: https://arxiv.org/abs/1611.01578

Christopher Sauer, M.Sc.

Friedrich-Alexander Universität Erlangen-Nürnberg, Chair of Engineering Design (KTmfk)

Martensstraße 9, 91058 Erlangen, Germany

Email: sauer@mfk.fau.de