GENERIC APPROACH TO PLAUSIBILITY CHECKS FOR STRUCTURAL MECHANICS WITH DEEP LEARNING

Spruegel, Tobias; Schröppel, Tina; Wartzack, Sandro
Friedrich-Alexander-University Erlangen-Nuremberg, Germany

Abstract

The simulation of product behavior is a vital part in virtual product development, but currently there is no tool or method available that can examine the quality of FE simulations and decide automatically on whether a simulation is plausible or non-plausible. In the paper a method is presented that enables automatic plausibility checks on basis of empirical simulation datasets. Nodal simulation data is transformed to numerical arrays, of fixed size, using virtual spherical detector surfaces. Afterwards the arrays are used to train a Deep Convolutional Neural Network (AlexNet). The Neural Network can then be used for plausibility checks of FE simulations (structural mechanics). In a first application a Deep Convolutional Neural Network is trained with simulation data of a demonstrator part, the rail of speed inline skates. After the GPU training of the Neural Network, further simulations are evaluated with the net. These simulations were not part of the training data and are used to calculate the prediction quality of the Neural Network. This approach is to support development engineers during design accompanying FEA in virtual product development.

Keywords: Simulation, Computational design methods, Integrated product development, Plausibility checks, Deep learning neural network

Contact:
Tobias Spruegel
Friedrich-Alexander-University Erlangen-Nuremberg
Mechanical Engineering
Germany
spruegel@mfk.fau.de

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 1: Resource-Sensitive Design | Design Research Applications and Case Studies, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

The simulation of product behavior is a vital part in virtual product development. Especially because simulation, at an early stage of product development, can create certain business benefits: reduced product development time; increased innovation; reduced product cost; reduced development cost and improved product quality (Adams, 2008). The set-up of valid simulations requires expert knowledge, acquired skills and sufficient expertise (Fröhlich, 2005). Design engineers, who perform Finite Element Analysis (FEA) infrequently, must be assisted and their FEA results need to be checked for plausibility. According to a recent survey (Boucher, 2013), companies struggle to expand simulation knowledge to a growing pool of users.

An automatic plausibility check for FE simulations can identify non-plausible simulations, and warn the user to utilize the results cautiously or ask for expert help (Spruegel, 2015). Before a classifier can evaluate a simulation, all the relevant information from the simulation set-up and the results must be available in computer-interpretable form. Classifiers such as Artificial Neural Networks attempt to create a good representation and build a mathematical metamodel to learn these representations from large data sets. Deep Artificial Neural Networks have won many contests in recent years (Schmidhuber, 2015) and can therefore be used for classification of simulation results in the two categories, plausible or implausible simulation.

The aim of this paper is to provide an overview of current tools for plausibility checks, current Deep Learning techniques and to propose a generic approach to plausibility checks for structural mechanics with deep learning, and a first application of the method at the end of this paper. This approach is to support development engineers during design accompanying FEA in virtual product development.

The structure of the paper is as follows. In Section 2, current approaches of plausibility checks for simulation are mentioned. Followed by Section 3 with Artificial Neural Networks and Deep Learning Classification. In Section 4 the methodology of the generic approach with spherical detector surfaces is presented. A first application of the methodology can be found in Section 5. The paper closes with summary and outlook in Section 6.

2 PLAUSIBILITY CHECKS

Plausible FE simulations are apparently, likely valid. This means that plausible FE results can still be wrong, but give a good hint on whether to accept or reject a considered simulation result (Spruegel et al. 2016). In Figure 1 the bending of a steel cantilever beam under constant load can be seen.

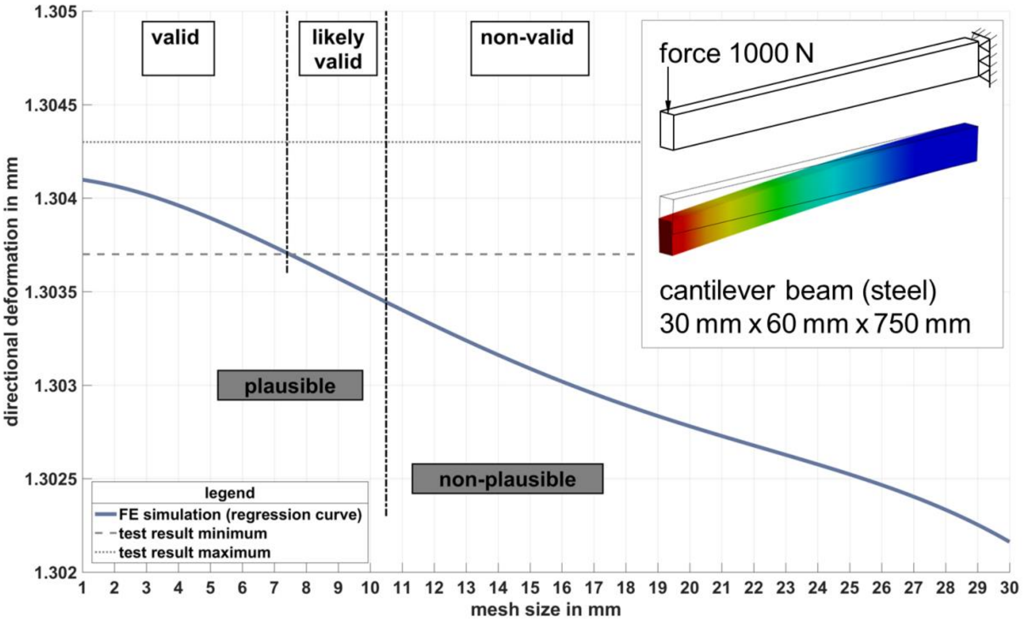

Figure 1. Qualitative example of a steel cantilever beam with plausible and non-plausible results due to coarse FE mesh size

Based on structural FE simulations the functional relationship between element mesh size and directional deformation are approximated by polynomial regression. Notional testing results are plotted exemplarily and define the categories valid, likely valid and non-valid simulation results. Additionally the terms plausible and non-plausible are augmented, which means that a non-plausible simulation is also non-valid. In this case results must be rechecked carefully or simulation experts must support the design engineer with the given simulation task.

Plausibility checks are a common tool to evaluate data or methods. The different fields of application are very diverse, as the following exemplary compilation shows:

Plausibility checks can be used to analyze electrical breakdown mechanisms in syntactic foams (Tröger, 2009) or for the measurement of uncertainties in soil analysis (Nestler, 2007). Tischler, 2013 describes common decisions and cooperative actions of multiple vehicles and the need for plausibility checks for vehicle sensor data. Braun et al., 2011 analyze the best combination of frequencies for the calculation of mean hearing loss in pure tone threshold audiometry for correlation with hearing loss for numbers in speech audiometry for plausibility checking in expertise.

First approaches for plausibility checks for simulation can be distinguished. Consequently, Qian, 2013 uses plausibility checks in dynamic simulations for a cylindrical roller bearing model in wind turbines, based on the known behaviors of bearings. Integrated behavior models for critical signs and their consistency are analyzed with plausibility checking by Ermel et al., 2011. Müller-Sommer and Straßburger, 2010 develop methods for automated pre-plausibility checks of input data for Intra-Logistic-Simulations in digital factories, entered by users. This enables the detection of obviously erroneous data. Automated plausibility checks for similar recurring mechanical parts are developed as part of an FEA assistance system (FEdelM) by Spruegel et al., 2015 and Spruegel and Wartzack, 2016. Currently there is no tool or method available that can examine FE simulations and decide automatically on whether the simulation is plausible or non-plausible.

3 DEEP LEARNING

Artificial Neural Networks (ANN) are a combination of units (artificial neurons), which are connected according to a defined scheme – therefore, they can communicate among each other. The structure of one individual unit is relatively simple, but they can solve efficiently complex tasks as a connected network (Callan, 1999). Initialized by McCulloch and Pitts, 1943, neural networks in their basic form try to model the brain of living beings with individual neurons (nerve cells).

Figure 2 shows the structure of an organic neuron. From the perikaryon (cell body), which contains the soma (cell nucleus), numerous dendrites and the axon go off. Dendrites form the connection to neighboring nerve cells, absorb stimuli, and relay these impulses to the nervous system (perikaryon and soma) (Luxem et al., 2010).

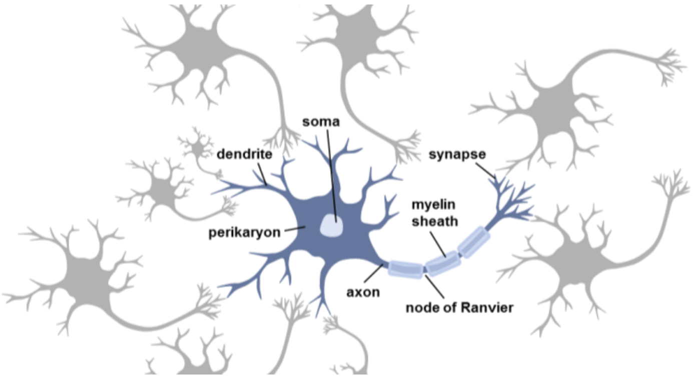

Figure 2. based on (Luxem et al., 2010)

The cell itself transmits pulses via the axon, the main exciter of the nerve cell. The axon divides itself into smaller branches and ends in the synapses. The connections between dendrites and synapses are the important contact points, via which nerve cells communicate with each other. With the splitted axon, one nerve cell can be connected with up to 100,000 other cells (Beck et al., 2016). Due to the large number of nerve cells and synapses that are linked to each other, a nerve cell usually receives its incoming signals from several upstream neurons (Adamy, 2011).

ANNs are complex mathematical models, which try to model nature's neural systems, and are a common machine learning technique (Bengio, 2009 & Widrow and Lehr, 1990). An ANN consists of artificial neurons and connections/weights between them. The artificial neurons are arranged within layers (Runkler, 2010 & Qin und Tang, 2014), and an ANN consists of at least one input and one output layer. Further, so-called hidden layers can be arranged between the input and the output layer (LeCun et al., 2015 & Schmidhuber, 2015 & Hinton, 2007).

Input values are passed to an ANN on the input layer (Figure 3, left side). The inner structure of a neuron on the hidden layer is detailed for neuron *G*. The inputs O_A to O_D are multiplied with the corresponding weight and a bias value is added to the sum. The resulting value n_G is the input for the transfer function *f*. The functional value of the transfer function is the output of the neuron and is transferred to all the neurons of the next layer.

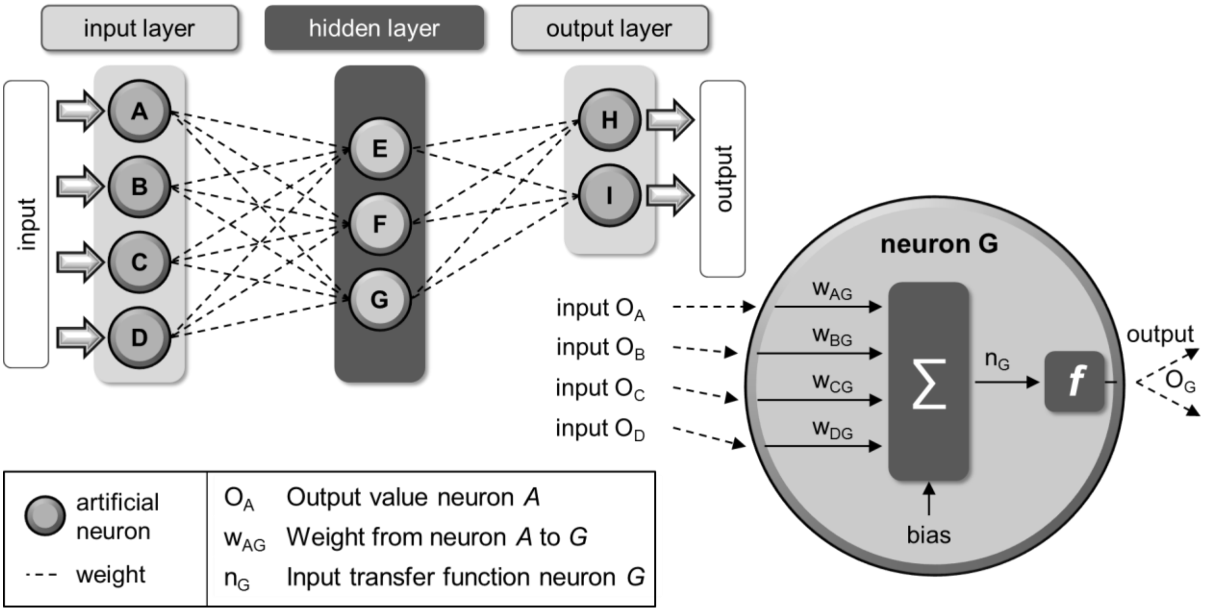

Figure 3. ANN with four inputs, one hidden layer and two outputs

Deep learning refers to methods and techniques of machine learning, which have been developed extensively since 2006 (Bengio, 2009 & LeCun et al., 2015 & Deng, 2014). The term “deep learning” is not commonly defined, but nevertheless an ANN with many hidden layers and a large number of neurons is considered “deep”. First concepts comparable to deep learning can already be found in the 1950s. Selfridge, 1958 introduced a mathematical model named Pandemonium, the basic structure and its topology can be compared roughly with today’s deep learning neural networks.

Deep learning can be applied on a variety of different tasks and performs very well. A deep convolutional network (known as AlexNet) was trained for the classification of 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into 1,000 different classes and achieved remarkable low error rates (Krizhevsky et al., 2012).

The improved convolutional neural network architecture GoogLeNet performed very well in the ImageNet ILSVRC-2014 contest for image classification (Russakovsky et al., 2015).

In summary, multi-layer ANNs, especially Deep Convolutional Networks, with a high number of neurons can be used for very diverse tasks and achieve very good results and high prediction quality. Available hardware, training time and available large annotated or large empirical datasets impose limitations for larger Neural Networks and on the implementation for further applications.

Nevertheless, Deep Neural Networks have great potential to perform very well in complicated classification tasks. Such a task is the classification of plausible and implausible FE simulation results. In the following a methodology is presented, that can transform FE simulations to a constant sized input matrix for a deep learning neural network. In an example application the presented methodology is applied and the results are presented in section 5.

4 A GENERIC APPROACH TO PLAUSIBILITY CHECKS FOR STRUCTURAL MECHANICS USING DEEP LEARNING

When FE simulations should be evaluated with deep neural networks, several challenges need to be overcome:

- Each numerical FE simulation is different, i.e. different parts with individual geometries, different number of mesh nodes, different boundary conditions such as supports and applied forces.
- All information must be available in computer-interpretable form.
- A large empirical dataset with input and target variables must be available for the training of deep learning networks (including training and validation data).
- The prediction quality of the trained network must be evaluated after the training with previously unknown data (testing data).
- Suitable network architecture and network parameters, for the given task, must be selected or created from scratch.

In Section 4.1 the conversion of individual FE meshes to a numerical array of fixed size using spherical detector surfaces is presented. Section 4.2 describes the conversion of nodal FE information to arrays of the same size. Finally a suitable Deep Learning architecture and framework is selected in Section 4.3.

4.1 Spherical Detector Surfaces

The methodology for transforming individual parts (meshed within FE software) to a constant array of numerical values uses spherical detector surfaces. Figure 4a) shows a roughly meshed block with 10-node tetrahedral elements, which are typically used to mesh solid parts of any shape. Afterwards a spherical surface with 36 by 36 pixels is spanned around the part; the part's center of gravity is the origin of the sphere and each pixel has the same surface area. After the conversion from Cartesian to spherical coordinates each node is projected to the detector surface (Figure4c) to f)). The mapping from the FE node to a certain pixel of the detector surface depends only on the azimuth and polar angle of the spherical coordinates, the radial distance has no influence on the mapping. After the mapping, the characteristic number of projected nodes per pixel can be counted and transformed into a numerical array of fixed size, in Figure 4 the result is a 36x36 array, as the sphere had 36x36 pixels . For higher resolutions and larger arrays, the angular increment of the detector pixel can be reduced optionally. Further information on the methodology of detector surfaces are described in Spruegel and Wartzack, 2015.

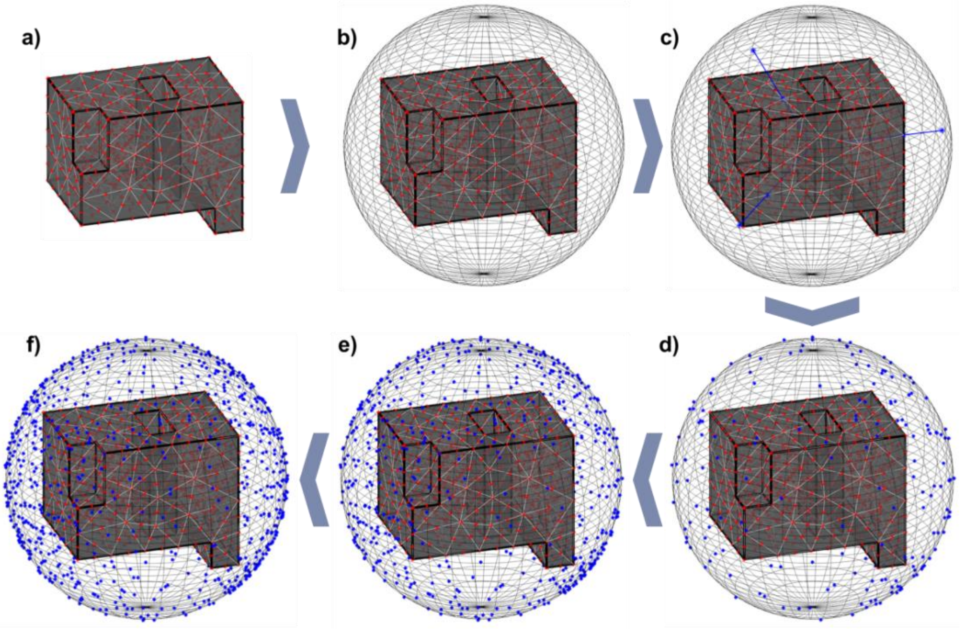

Figure 4. Conversion from FE nodes of a block to characteristic detector sphere with individual number of nodes per pixel surface

4.2 Conversion of FE simulations to identical arrays

As stated earlier, it is highly essential to transfer further information from an underlying FE simulation to the plausibility check, only the nodes are not sufficient. Therefore, a slight modification of the spherical detector surface methodology is implemented. Besides the detector array from Figure 4f), the assignment of each node to one distinct pixel is captured. This enables the transformation of any nodal value to a numerical array of fixed size. Consequently, result values and the boundary conditions of simulations can be transformed to arrays of pre-defined size. The result is an array for each of the following nodal inputs: nodes; fixed support; remote displacement; force component x-direction; force component y-direction; force component z-direction; deformation x-direction; deformation y-direction; deformation z-direction; equivalent stress; normal stress x-direction; normal stress y-direction; normal stress z-direction.

4.3 Deep Learning

For the plausibility check of FE simulations a classification tool is necessary that can deal with the following requirements:

- Highly nonlinear classification problem.
- Classification with two classes: plausible and non-plausible input data.
- Processing of a large number of inputs (several input arrays of size 100x100 values).
- Processing capabilities for large empirical datasets (several thousand FE simulations).

As the number of inputs of such an ANN is very high and the relationship between input and output is nonlinear, a large number of hidden units is required to do the classification task with ANNs. Therefore, the training of deep convolutional networks on GPUs (graphics processing units) is the method of choice. Convolutional Neural Networks (CNNs) are hierarchically structured, multi-layered artificial neural networks (Fasel, 2002). CNNs recognize basic features in the first layers and combine them in later layers (LeCun and Bengio, 1998). For the recognition of basic features receptive fields are used, which means that a set of neurons passes their output values on to a single downstream neuron which are combined again later on, such as in the retina of vertebrate animals (Bryngdahl, 1964). Typically, CNNs process complete images.

Deep learning frameworks, such as CAFFE (Jia et al., 2013); CNTK (Agarwal et al., 2014) or TensorFlow (Abadi et al., 2015), provide the capability to handle large datasets and to use GPUs for the training of the CNNs. Besides the framework, the architecture of the CNN must be defined, or available architectures such as the AlexNet (Krizhevsky et al., 2012) can be applied.

5 FIRST APPLICATION AND RESULTS

As a first application, the rail of speed inline skates, differing in geometry and applied forces, is used to show the capabilities of the described methodology for automatic plausibility checks for structural mechanics with deep learning. The CAD geometry is shown in the upper section of Figure 5. Different wheel diameters (76 mm, 80 mm and 90 mm) as well as different wall thicknesses (2 mm, 3 mm, 4 mm and 5 mm) create 12 different geometries (the extreme geometries are shown on the left and right in Figure 5). Simulation input parameters, such as the mesh size (1 mm, 1.5 mm, 2 mm, 3 mm, 4 mm, 5 mm, 10 mm, 15 mm) and the applied forces differ. The first force simulates the weight of a person on to the rail and the second force simulates the hit of an object with the first wheel. Consequently, 607 different combinations of input parameters (simulation design points) are simulated for each of the geometries. This results in a large database of 7.284 structural mechanic FE simulations. The dataset is split into sub-datasets used for training, validation and testing of the CNN. For each simulation the following results are available: deformation in x-, y- and z-direction, equivalent stress and normal stress in x-, y- and z-direction. In combination with the nodes and the boundary conditions this forms the whole dataset considered in the application.

For each simulation the above described methodology with spherical detector surfaces is applied to generate the desired uniform numerical arrays. The nodes from the FE simulation can have random orientation (rotation and translation), therefore, a Principal Component Analysis (PCA) is used.

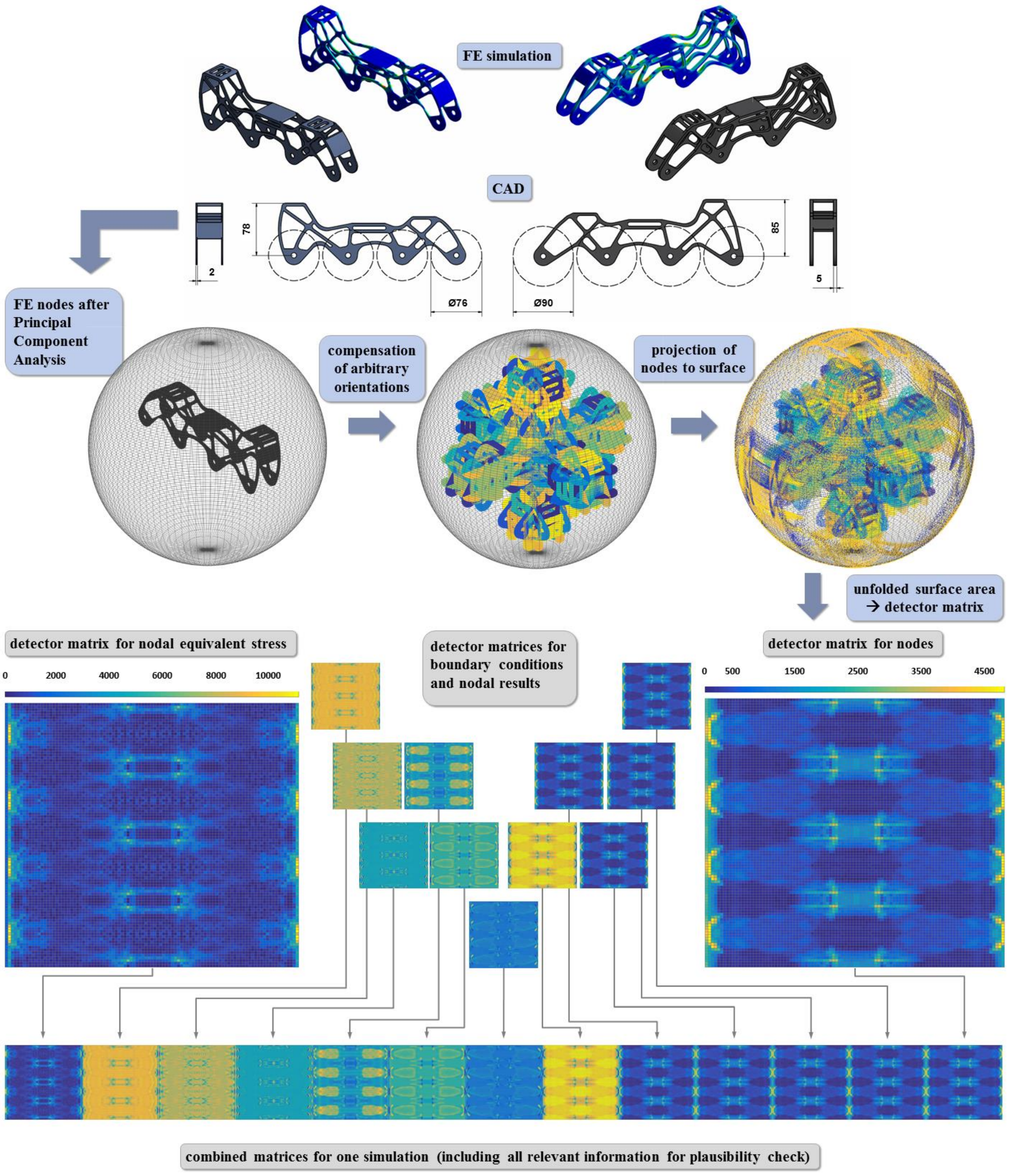
FE simulation
CAD
78
2
Ø76
Ø90
85
5
FE nodes after Principal Component Analysis
compensation of arbitrary orientations
projection of nodes to surface
unfolded surface area → detector matrix
detector matrix for nodal equivalent stress
0
2000
4000
6000
8000
10000
detector matrices for boundary conditions and nodal results
detector matrix for nodes
0
500
1500
2500
3500
4500
combined matrices for one simulation (including all relevant information for plausibility check)

Figure 5. Automatic transformation of FE simulation to a matrix array of constant size

The randomly oriented nodes are transformed to a representation in the principal component space. As the result form the PCA is not explicit (the geometry can rotate around each of the global axes by 0°, 90°, 180° or 270°). This results in four different rotations for each of the x-, y- and z-axis of the global coordinate system. In sum these are 64 ($=4^3$) possible orientations. To eliminate the inconsistent rotations, the other 64 orientations are created intentionally (see Figure 5, sphere in the middle). Afterwards the nodes are projected onto the surface and the number of nodes in each pixel is counted. This forms the uniform numerical arrays as described in the methodology (in Figure 5 the two 100x100 arrays for the nodes and the equivalent stress are shown in detail with their corresponding scale). The whole input for the plausibility check consists of 13 arrays and contains all the relevant information of one FE simulation for the plausibility check. These arrays are different for each considered geometry and the corresponding simulation input parameters. As a result they can be used to evaluate the plausibility of a simulation.

The available dataset of 7.200 simulations is split on a 80:20 basis: 5.700 data samples for training, 1.500 data samples for validation and additional 84 data samples for testing the prediction quality of the trained CNNs. A single Tesla GPU is used for training and the deep learning framework is CAFFE (Jia et al., 2013). The CNN architecture is the AlexNet and the arrays are converted to images, each numerical value is one pixel of the image. The 13 arrays are combined to one image per simulation with 100x1300 pixels. With an AlexNet crop size of 100 and batch size 500 the supervised learning on a Tesla 2075 GPU takes 21 days. The prediction quality is calculated from the entries of the confusion matrix for the unknown 84 data samples. According to Powers, (2011) the following performance characteristics can be calculated for the best CNN out of 450.000 AlexNet iterations:

Table 1: Performance characteristics calculated for the best CNN out of 450.000 AlexNet iterations

performance characteristic	value
positive predictive value (precision)	86,11 %
negative predictive value	89,58 %
true positive rate (sensitivity)	86,11 %
true negative rate (specificity)	89,58 %
accuracy	88,10 %

The results from Table 1 demonstrate that a trained AlexNet CNN in standard configuration can perform an automatic plausibility check for structural FE simulations. Further improvements in the prediction quality can be achieved by optimizing the CNN net parameters, during the training of this example the net parameters (i.e. multiplier on the global learning rate, multiplier on the global weight decay, etc.) are on default. These improvements are not included in the results in Table 1 and therefore, an improvement of the prediction quality is very likely.

6 SUMMARY AND OUTLOOK

In conclusion, a generic approach to plausibility checks for structural mechanics with deep learning was presented in this paper. The main challenges, such as the conversion of FE simulations to numerical arrays can be overcome with the method of spherical detector surfaces for nodal FE information. The arrays can be used as input deep neural networks. The training of CNNs on GPU enables the usage of networks with many layers and a large number of hidden neurons. In a first application the feasibility of the approach could be proved with high prediction quality, already for the standard AlexNet parameters. The further steps are the generation of a larger data base with many different parts and simulation parameters and the training of a CNN with modified structure to prove the generality of the approach. Especially the conversion from numerical arrays to pictures, with potential loss of information due to compression, should be eliminated with an own ANN architecture that is able to perform the classification directly on basis of the described numerical arrays.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D. Olah, C., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015), TensorFlow: *Large-scale machine learning on heterogeneous systems*, Software available from: tensorflow.org.

Adams, V. (2008), *A Designer's Guide to Simulation with Finite Element Analysis*, NAFEMS, Hamilton.

Adamy, J. (2011), *Fuzzy Logik Neuronale Netze und Evolutionäre Algorithmen*, Shaker, Aachen.

Agarwal, A., Akchurin, E., Basoglu, C., Chen, G., Cyphers, S., Droppo, J., Eversole, A., Guenter, B., Hillebrand, M., Hoens, T.R., Huang, X., Huang, Z., Ivanov, V., Kamenev, A., Kranen, P., Kuchaiev, O., Manousek, W., May, A., Mitra, B., Nano, O., Navarro, G., Orlov, A., Parthasarathi, H., Peng, B., Radmilac, M., Reznichenko, A., Seide, F., Seltzer, M.L., Slaney, M., Stolcke, A., Wang, H., Wang, Y., Yao, K., Yu, D., Zhang, Y. and Zweig, G. (in alphabetical order) (2014), "An Introduction to Computational Networks and the Computational Network Toolkit", *Microsoft Technical Report* MSR-TR-2014-112.

Beck, H., Anastasiadou, S. and Meyer zu Reckendorf, C. (2016), *Eine bebilderte Reise in die Welt der Nervenzellen*, Springer Spektrum, Berlin Heidelberg. DOI 10.1007/978-3-662-47092-3

Bengio, Y. (2009), *Learning deep architectures for AI*, Now Publishers, Hanover.

Boucher, M. (2013), *Enhance Engineering: Reduce Time and Optimize Products with Simulation Best Practices*, Research Report Aberdeen Group.

Braun, T., Dochtermann, S., Krause, E., Schmidt, M., Schorn, K. and Hempel, J.M., (2011) *"Korrelation von Tonschwellenaudiogramm und Hörverlust für Zahlen"*, Vol. 59, pp. 908–914.

Bryngdahl, O. (1964), "Größenschätzung des rezeptiven Feldzentrums der menschlichen Retina", *Pflügers Archiv - European Journal of Physiology*, Vol. 280 No.4 , pp. 362-368. DOI 10.1007/BF00362720

Callan, R. (1999), *The essence of neural networks*, Prentice Hall Europe, London.

Deng, L. (2014), "A tutorial survey of architectures, algorithms, and applications for deep learning", *APSIPA Transactions on Signal and Information Processing*, Vol. 3, pp. 1-29). DOI 10.1017/ATSIP.2013.99

Ermel, C., Gall, J., Lambers, L. and Taentzer, G. (2011), "Modeling with Plausibility Checking: Inspecting Favorable and Critical Signs for Consistency between Control Flow and Functional Behavior", *International Conference on Fundamental Approaches to Software Engineering (FASE)*, Saarbrücken, 26. March – 3. April 2011,Summary of Proceedings, Springer, Berlin, pp. 156-170. DOI 10.1007/978-3-642-19811-3_12.

Fasel, B. (2002), "Robust face analysis using Convolutional Neural Networks", *16th International Conference on Pattern Recognition (ICPR 2002)*, Quebec, 11.-15. August, IEEE, Los Alamitos, pp. 40-43.

Fröhlich, P. (2005), *FEM-Anwendungspraxis. Einstieg in die Finite Elemente Analyse*, Vieweg, Wiesbaden. DOI 10.1007/978-3-663-10053-9

Hinton, G.E. (2007), "To recognize shapes, first learn to generate images", *Progress in Brain Research*, Vol. 165, pp. 535-547. DOI 10.1016/S0079-6123(06)65034-6

Jia, J., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T. (2013), *Caffe: An open source convolutional architecture for fast feature embedding*, http://caffe.berkeleyvision.org/

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), *ImageNet Classification with Deep Convolutional Neural Networks*, University of Toronto.

LeCun, Y., Bengio Y. (1998), "Convolutional networks for images, speech, and time series", In: Arbib, M.A. (Ed.), *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge MA, pp. 255-258.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, Vol. 521, pp. 436-444. DOI 10.1038/nature14539

Luxem, J., Kühn, D., Runggaldier, K. and Knieps, M.M. (2010), *Anatomie und Physiologie. Rettungsdienst RS/RH*, Urban & Fischer, München.

McCulloch, W.S. and Pitts, W. (1943), "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, Vol. 5, No. 4, pp. 115–133.

Müller-Sommer, H. and Straßburger, S.. (2010), "Methods for the Plausibility Check of Input Data for Intra-Logistic-Simulations", *Integration Aspects of Simulation: Equipment, Organization and Personnell*, Karlsruhe, 7.-8. October 2010, *KIT Scientific Publishing*, Karlsruhe, pp. 61-68.

Nestler, A. (2007), B*estimmung der Messunsicherheit für die Verfahren und Methoden zur Bodenanalytik des Anhanges 1 der Bundes-Bodenschutz- und Altlastenverordnung*, Dissertation, Freie Universität Berlin.

Powers, D.M.W. (2011), "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation", *Journal of Machine Learning Technologies*, Vol. 2 No.11, pp. 37-63.

Qian, W. (2013), *Dynamic Simulation of Cylindrical Roller Bearings*, Dissertation, RWTH Aachen University.

Qin, Z and Tang, Y. (2014), *Uncertainty Modeling for Data Mining, A Label Semantics Approach*, Springer, Berlin Heidelberg. DOI 10.1007/978-3-642-41251-6

Runkler, T.A. (2010), *Data Mining. Methoden und Algorithmen intelligenter Datenanalyse*, Vieweg + Teubner, Wiesbaden. DOI 10.1007/978-3-8348-9353-6

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. and Fei-Fei, L. (2015), "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, Vol. 115 No.3, pp. 211-252. DOI 10.1007/s11263-015-0816-y

Schmidhuber, J. (2014), "Deep Learning in Neural Networks: An Overview", *Neural Networks*, Vol. 61, pp. 85-DOI 117. 10.1016/j.neunet.2014.09.003

Selfridge, O.G. (1958), "Pandemonium: A Paradigm for Learning", *National Physical Laboratory symposium No. 10 Mechanisation of Thought Processes*, London, 24.-27. November, Her majesty's stationary office, London, pp. 513-529.

Spruegel, T.C., Hallmann, M. and Wartzack, S. (2015), "A concept for FE plausibility checks in structural mechanics", *NAFEMS World Congress*, San Diego, 21.-24. June 2015, Summary of Proceedings, NAFEMS.

Spruegel, T.C. and Wartzack, S. (2015), "Concept and Application of Automatic Part Recognition with Artificial Neural Networks for FE Simulations", *Proceedings of the 20th International Conference on Engineering Design (ICED15)*, Vol. 6 No.2, pp. 183-193.

Spruegel, T.C. and Wartzack, S. (2016), "Das FEA-Assistenzsystem – Analysteil FEdelM", Entwerfen Entwickeln Erleben, Dresden, 30. June - 1. July 2016, TUD Press, Dresden, pp. 463-474.

Spruegel, T.C., Kestel, P. and Wartzack, S. (2016), “FEA-Assistenzsystem – Plausibilitätsprüfung für Finite-Elemente-Simulationen mittels sphärischer Detektorflächen”, *Design for X (DFX)*, 05.-06. October, Jesteburg, TuTech, Hamburg, pp. 27-38.

Tischler, K. (2013), *Informationsfusion für die cooperative Umfeldwahrnehmung vernetzter Fahrzeuge*, Dissertation, Karlsruher Institut für Technologie (KIT).

Tröger, K.A. (2009), *Analyse der elektrischen Durchschlagmechanismen in syntaktischen Schäumen*, Dissertation, RWTH Aachen University.

Widrow, B. and Lehr, M.A. (1990), “30 Years of Adaptive Neural networks: Perceptron, Madaline, and Backpropagation”, *Proceedings of the IEEE*, Vol. 78 No.9, pp. 1415-1442. DOI 10.1109/5.58323

ACKNOWLEDGMENTS

The authors would like to thank the NVIDIA Corporation and the academic GPU Grant Program for the donation of a Tesla GPU.

This research work is part of the FAU “Advanced Analytics for Production Optimization” project (E|ASY-Opt) and funded by the Bavarian program for the “Investment for growth and jobs” objective finance by the European Regional Development Fund (ERDF), 2014-2020. It is managed by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology. The authors are responsible for the content of this publication.