M Blatzheim et al.

Neural Network Performance Enhancement for Limited Nuclear Fusion Experiment Observations Supported by Simulations

Neural Network Performance Enhancement for Limited Nuclear Fusion Experiment Observations Supported by Simulations

Marko Blatzheim[1,2]

mabl@ipp.mpg.de

Daniel Böckenhoff[1]

dboe@ipp.mpg.de

Hauke Hölbe[1]

haho@ipp.mpg.de

Roger Labahn[2]

roger.labahn@uni-rostock.de

Thomas Sunn Pedersen[1]

tspe@ipp.mpg.de

and the W7-X Team

[1]*Max-Planck-Institute für Plasmaphysik, Wendelsteinstraße 1, 17491 Greifswald*

[2]*Institut für Mathematik, Universität Rostock, Ulmenstraße 69, 18057 Rostock*

Abstract

Artificial neural networks (NN) are computational tools that learn tasks from a large amount of data without task-specific programming. It has been shown in [1] that NNs are able to find connections between heat load pattern and plasma edge properties of the nuclear fusion experiment Wendelstein 7-X. Continuing the previous work, the impact of the data pre-processing is investigated systematically with respect to the NN performance on sparse experimental data. With the considered pre-processings, training with simulated data only does not lead to sufficient plasma parameter estimation. Also in contrast to NN training restricted to experimental data only, NNs show a significantly improved performance on the same, exclusively experimental test set when trained with simulated data supported by a very limited amount of experimental data. This is of great practical relevance because simulation and experiment data sets show discrepancies. The NNs are able to focus on pattern occurring in both, the simulated data as well as the experimental data. The results presented herein support real time plasma control which is an important step en route towards a nuclear fusion power plant.

---

1. Introduction

In machine learning, the size of the data set on which an artificial neural network (NN) is trained is crucial for a good performance. However, there are many cases where the cardinality of real (measured) data is limited. This dilemma may be resolved by complementing the available data with simulations. The underlying model for a simulation typically neglects some aspects of reality. Often it is necessary to simplify the simulation model to have numerically stable equations describing the considered process that can be solved in a reasonable amount of time. Or some underlying principles are just not fully understood so they can not be modeled. Thus, simulation and experiment often show systematic differences in certain properties. On the other hand, it can be assumed that, even with underlying simplifications, some properties with a sufficiently small discrepancy can be found.

Based on the example of the nuclear fusion research reactor Wendelstein 7-X (W7-X) [2], this work aims to find suitable NN architectures as well as input parameterizations. The focus lies on retrieving NNs that, trained on simulated data, eventually supported with a very limited number of experimental data, generalize onto experimental data. More specific, this is demonstrated by training a NN on simulated and experimental observations of W7-X heat load patterns to reconstruct the plasma edge property t. Practical application of this NN will be the support of a real-time control system, ensuring the safety of all W7-X plasma facing components.

In the next section we briefly explain the essentials regarding W7-X, heat load pattern retrieval from infra-red cameras and a proper formulation of t, followed by a definition of the data set composition. The two used NN architectures are described and the parametrizations introduced before the NN performance is presented and analyzed.
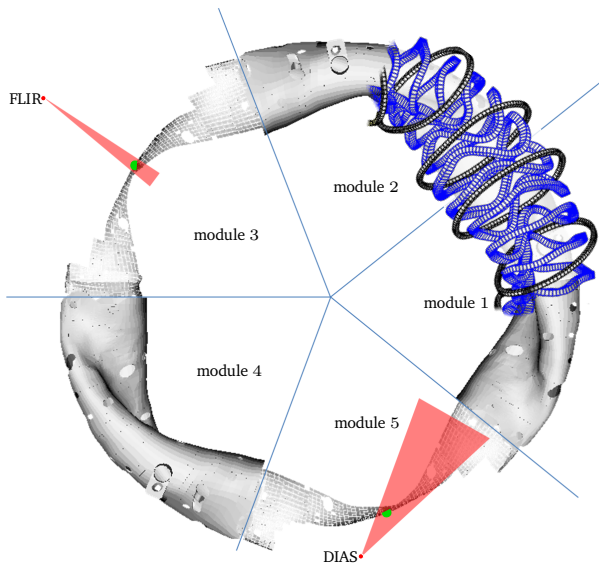
2. Methods

2.1. Physical Background

Thermal energies required for nuclear fusion imply the reactants to be in the plasma state. W7-X follows a new approach to confine the plasma with 70 modular, superconducting, electromagnetic coils. All coils are based on five non planar and two planar, unique coils (cf. red and blue coils in Figure 1b respectively) that define one submodule. Two submodules assembled in a point mirrored way (cf. Figure 1b) make up one of five symmetric modules shown in Figure 1a. The non planar coils give rise to a poloidal magnetic field component. Together with the toroidal magnetic field, generated by the non planar as well as the planar coils "A" and "B", it gives rise to a twist of the magnetic field lines which is essential for stable plasma confinement. This twist is denoted ι and can be easily modified by changing the electric current in the planar coils.

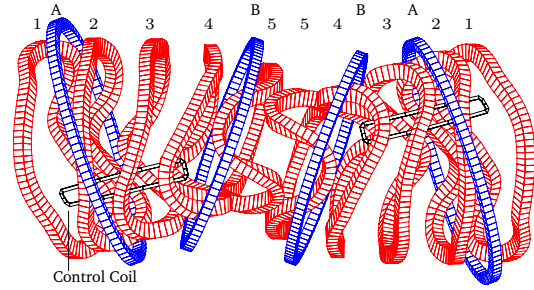The ι at the magnetic edge strongly influences the heat load pattern resulting from the contact of the edge plasma with the plasma facing components, i. e., the components limiting the plasma spatially. In a special experiment series conducted in March 2016, the normalized current of the planar coil "B", I_{B}, can be used as a substitute for ι (cf. [1, Section 2.3]). I_{B} is the target for the NNs here. Post processing the infra-red videos of the so called "DIAS" diagnostic, we retrieve the heat load on the triangulated CAD (computer-aided design) representation of the primary plasma facing component, the so called 'limiter' (cf. Figures 1a and 1c).

This heat load pattern is considered the NN input in a further processed form. The reconstruction of ι and further plasma properties in real-time will support the protection of the plasma facing components of W7-X.

For details regarding the underlying physics and data origin (experimental as well as synthetic) we refer to [1]. Figure 2 shows an exemplary heat load pattern on the limiter from W7-X module 5 for vacuum field line diffusion simulation (Figure 2a) as well as infrared data (Figure 2b) at the same magnetic configuration. The right half of the limiter is shadowed from the infrared camera view. Some elementary characteristics are similar, e. g. the maximum heat load is located at the third limiter tile. However, in detail, the structure of the infrared observation can not be covered.

(b) *Coils contained in one module with modular coils $1-5$ (red) and planar coils A, B (blue). One module is point symmetrical towards its center. Adapted after [3]*

(a) *Top-down CAD view of the W7-X inner vessel, showing sight lines of the IR camera system for the limiter setup with cutaways in modules three and five as used in the first experimental campain. On this scale and view, the limiters are small (green). One segment of the total 50 modular (blue) and 20 planar coils (black) is overlaid in modules 1 and 2.*
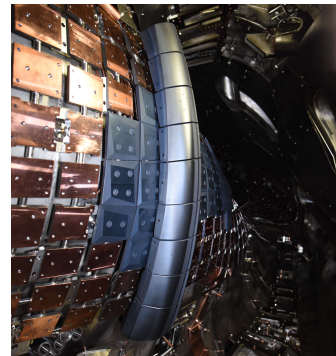
(c) *Side view of the W7-X limiter in module 5*

Figure 1: *Overview of essential parts of W7-X regarding the first experimental campain*
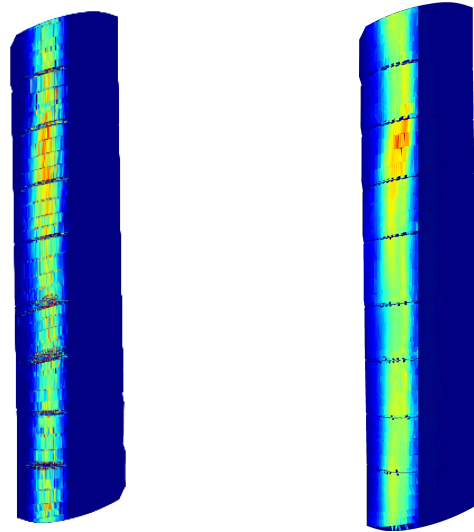
(a) *Heat load, represented by strike point density from a field line diffusion simulation*
(b) *Heat load, calculated from DIAS camera data*

Figure 2: *Front views of heat load representations on limiter in module 5 at discharge "20160309.007". The right side of the limiter is rejected since it is largely shadowed from the camera view.*

2.2. Data Sets

The investigated data sets result from experimental and simulated ι scans of W7-X (cf. [1, Section 2.3]). The simulation set $\mathbb{S}$ was created by field line diffusion in vacuum magnetic configuration with $|\mathbb{S}| = 3993$ (cf. [1, Section 2.4.2]).

The experiment set of processed infrared data $\mathbb{I}$ depends on 16 available experiments of the desired ι scan with six dissimilar I_B (cf. [1, Section 2.4.1]). Each infrared video corresponding to one experiment contributes in average 20 frames to the same value of I_B, leading to a total cardinality of $|\mathbb{I}| = 319$. A mixed set is defined as $\mathbb{M} = \mathbb{S} \cup \mathbb{I}$. To determine NN quality, the set is divided in disjoint subsets, namely training set, validation set and test set. Any test set contains 6 values of I_B from $\mathbb{I}$ and each of corresponding video frames of the experiment.

2.3. NN Architectures

Two promising NN architectures are compared that only need a feasible amount of free parameters to avoid overfitting, since the data set size is limited. The first one is mainly based on convolutional layers [4] as seen in Figure 3. Three consecutive convolutional layers are followed by two fully connected layers. The second NN architecture starts with an inception module [5] followed by pooling [6], a convolutional layer and two fully connected layers (as shown in Figure 4).

Weights are initialized randomly as recommended in [7] and biases are initialized as zero. The NNs weights and biases are iteratively improved by the adam optimizer [8] during the training process. All activation functions except for the last layer are rectified linear units (ReLU). Because the NNs are designed to solve a regression problem, the activation function of the last layer is the identity. The implementation is done in tensorflow [9].

2.4. Parametrization

The heat load is given on an unstructured triangular grid that represents the CAD structure of one limiter. This data is transformed affine from cartesian coordinates to an orthogonal ξ, η, ζ-coordinate system, where ζ points in the direction of the mean of the normals of all triangles
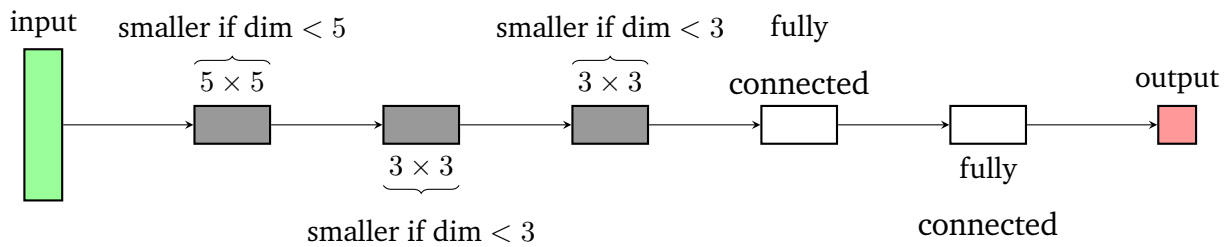
Figure 3: *Structure of NN with three convolutional layers: The grey elements represent convolutional layer with associated kernel size.*
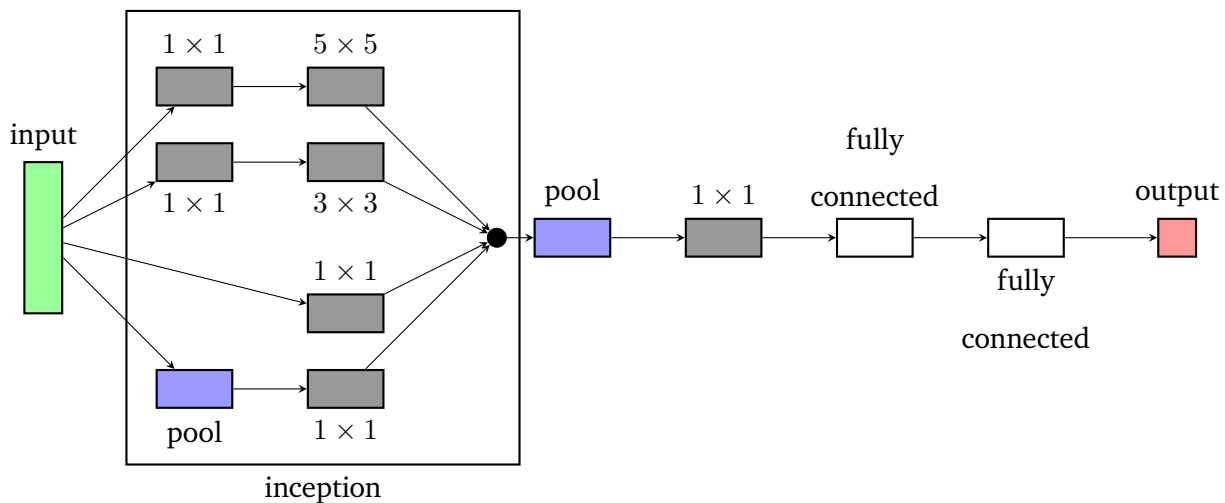

Figure 4: *Structure of NN with inception module: The grey elements represent convolutional layer with associated kernel size. The blue elements are max-pooling layer and the white elements are fully connected layer.*

Table 1: *Dimensions of the used partitionings and total number of elements*

n_ξ	2	4	9	9	15	18	27	36	72	144
n_η	1	1	1	5	6	8	10	12	15	30
$n_\xi \cdot n_\eta$	2	4	9	45	90	144	270	432	1080	4320

forming the limiter. Only triangles within a tight bounding box around the limiter are considered for the parametrization.

2.4.1. Partitioning

The ξ, η- space is then divided by various 2D grids with sizes described in Table 1 and shown for simulations and experiments in Figure 5a and 5b respectively. There are no divisions in ζ direction. The first three columns in Table 1 show one dimensional partitionings. The inception NN is applied for inputs of dimensionality of at least 5×5 only. However, for the one-dimensional inputs the convolutional kernel dimensionality is reduced from that presented in Figure 3 such that it does not exceed the input dimensionality.

2.4.2. Extracted Characteristics

For each element p of the partitionings, characteristic values can be extracted. One is the spatial mean

$$\vec{\mu}_p(\xi, \eta, \zeta) = \frac{1}{\sum_{i=1}^{n_p} A_i q_i} \sum_{i=1}^{n_p} A_i q_i \vec{\kappa}_{ip}, \tag{1}$$

with number of triangles per partitioning n_p, triangle areas A_i, triangle heat load q_i and triangle centroids $\vec{\kappa}_{ip} = (\xi_i, \eta_i, \zeta_i)_p^T$. Another statistical characteristic is the spatial standard deviation $\vec{\sigma}_p$ calculated complementary to $\vec{\mu}_p$. The direction vector $\vec{\delta}_p(\xi, \eta, \zeta)$, calculated as the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ to the covariance matrix of $\mathbf{\Xi}_p = \left(\vec{\kappa}_{1p}, \cdots, \vec{\kappa}_{ip}, \cdots, \vec{\kappa}_{n_p p} \right)^T$ weighted by $A_i q_i$, can be characteristic as well. The weighted covariance matrix is defined as

$$\boldsymbol{Cov}(\mathbf{\Xi}_p) = \left(\frac{\sum_{i=1}^{n_p} A_i q_i \left(\Xi_{ijp} - \mu_{jp} \right) \left(\Xi_{ikp} - \mu_{kp} \right)}{\left(\sum_{i=1}^{n_p} A_i q_i \right)^2} \right). \tag{2}$$

This parameter is inspired by divertor heat load patterns which show a more versatile shape [10]. The last examined parameter is the relative heat load

$$\rho_p = \frac{\hat{q}_p}{\sum_{p'=1}^{m} \hat{q}_{p'}} \tag{3}$$

with

$$\hat{q}_p = \frac{\sum_{i=1}^{n_p} A_{ip} q_{ip}}{\sum_{i=1}^{n_p} A_{ip}} \tag{4}$$

defined as the absolute heat load.

Three combinations of those parameters are studies as NN input: (μ, σ), (μ, δ) and ρ exclusively. In Figure 5 ρ is shown for simulation and experiment of the same physical condition. Note that the input dimension of the NNs is batch size $\times$ n_ξ $\times$ n_η $\times$ input channels, with input channels beeing 6 for the cases (μ, σ) as well as for (μ, δ) but 1 for ρ.

A short notation for the NN settings is defined as

$$f(\text{training set}, \text{validation set}, \text{test set})_{\text{partition, architecture}}^{\text{parametrization}}, \tag{5}$$

with an arbitrary function f applied to the NN output. For example $rmse(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})_{9\times5, inception}^{\rho}$ refers to the root mean square error of an inception NN trained as well as validated on samples from set $\mathbb{S}$ and tested with experimentally observed infrared data $\mathbb{I}$, requiering a 9×5 input of relative intensities ρ. $\mathbb{S}_{90} \subset \mathbb{S}$ and $\mathbb{S}_{10} \subset \mathbb{S}$ with $\mathbb{S}_{90} \cap \mathbb{S}_{10} = \emptyset$ refer to two disjunct sets for training and validation consisting of $90\,\%$ and $10\,\%$ of the samples within $\mathbb{S}$ respectively. Given a single term in the brackets implies that training, validation and test data sets are disjunct and subset of the same super set. If one describing parameter is omitted, the entirety of all possible parameters of that kind is referred to. So $(\mathbb{S})_{9x5}^{\rho}$ describes all NNs parametrized by ρ and partitioned into 9 times 5 parts with both of the two considered architectures. They are trained, validated and tested on subsets of $\mathbb{S}$. The set $\mathbb{I}_{\complement} = \mathbb{I} \setminus \mathbb{M}$ with $|\mathbb{I}_{\complement}| = 190$ is defined as a NN test set.
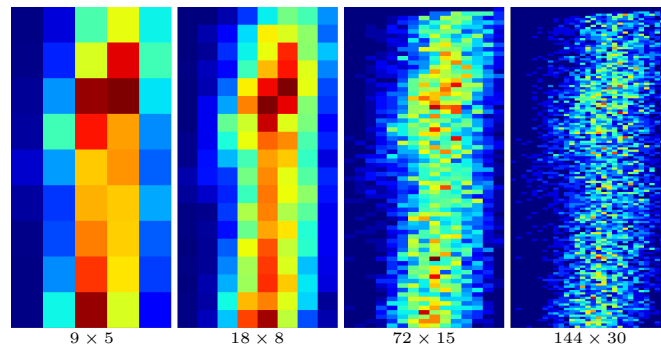
(a) *Simulation*
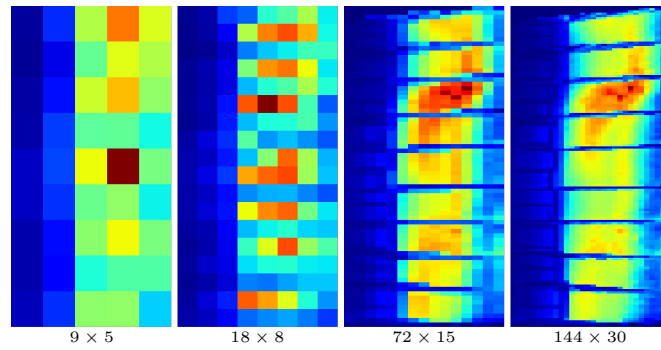
(b) *Experiment*

Figure 5: *Comparision of relative heat load for four exemplary partitions between simulation and experiment. The color scale ranges from 0 to the maximal value of each plot respectively.*

3. Results

The NN performance for all tested settings is shown in Figure 6. These two graphics depict the $rmse$ dependence on the partitionings defined in section 2.4.1. The NN performances measured in terms of the $rmse$ are divided into 10 groups representing the partitioning. Within each group the parameter and NN architecture choices are shown. To avoid false conclusions by statistical outliers, an ensemble of 27 NNs trained with the same settings has been calculated. Variations are the randomness of the weight initialization and the mini batch sampling as well as different learning rates and batch sizes. Markers and bars indicate the mean $rmse$ ($\overline{rmse}$) and the associated $95\,\%$ confidence interval for the mean respectively. The confidence interval has been calculated by bootstrapping [11]. The true possible values of I_B range between -0.05 and 0.18. To emphasize the NN reconstruction quality, $10\,\%$ of the total I_B range is marked by the dotted, green line. NNs with $rmse$ below $10\,\%$ I_B range are showing a reasonable ability in I_B targeting. The figures 7, 8, 9, and 11 depict subsets of the outcomes shown in Figure 6 to clarify the observations.

3.1. Simulation trained NNs

The results of the NNs trained on $\mathbb{S}$ are shown in Figure 6a. It can be seen, that $(\mathbb{S})^\rho$ performs well with both NN architectures for partitionings up to 36×12. The $\overline{rmse}$ is minimal at the partitioning 9×5 and increases with finer as well as coarser resolution. The NNs $(\mathbb{S})^{\mu\&\sigma}$ and the $(\mathbb{S})^{\mu\&\delta}$ perform especially good for the coarse partitionings between 2×1 and 9×5. For finer partitionings the performance decreases gradually. This can be understood contemplating the decreasing information content per section with shrinking section size.

On the basis of Figure 6a we investigate the performance of NNs characterized by $(\mathbb{S})$ vs $(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$. A good performance of the NN $(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$ would be advantageous since it would indicate applicability to new, never conducted experiments. Although it would be preferable, the evaluating indicates that this is not the case.

We observe not only that $\overline{rmse}(\mathbb{S}) \ll \overline{rmse}(\mathbb{S}, \mathbb{S}, \mathbb{I})$, but also that $\overline{rmse}(\mathbb{S}, \mathbb{S}, \mathbb{I})$ significantly exceeds the $10\,\%$ I_B range. The NNs $(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$ are specialized onto patterns of $\mathbb{S}$. Those patterns are not suitable to determine I_B from experimental data. The fundamentally different magnitude of the
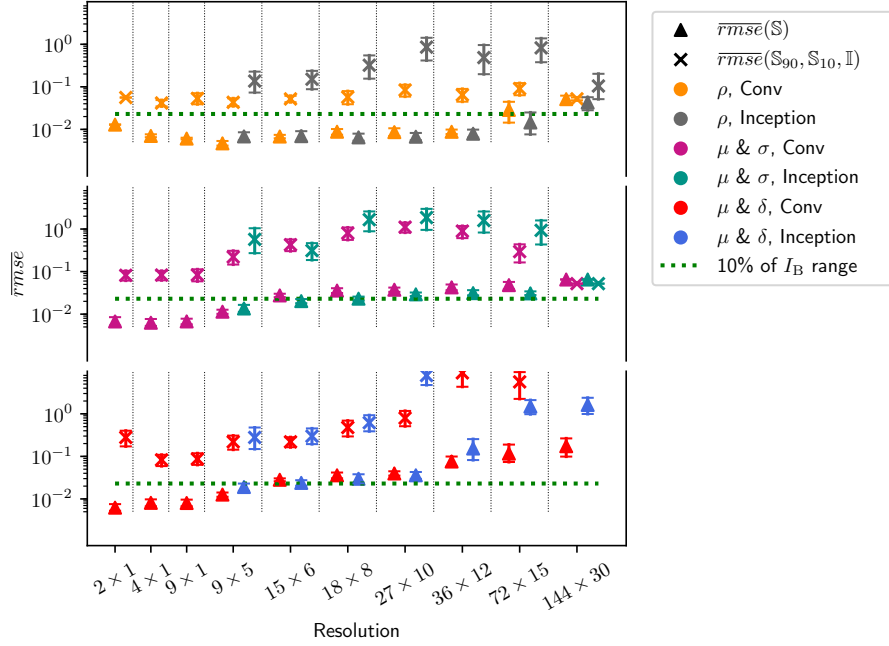
(a) *Logarithmic representation of $\overline{rmse}(\mathbb{S})$ and $\overline{rmse}(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$*
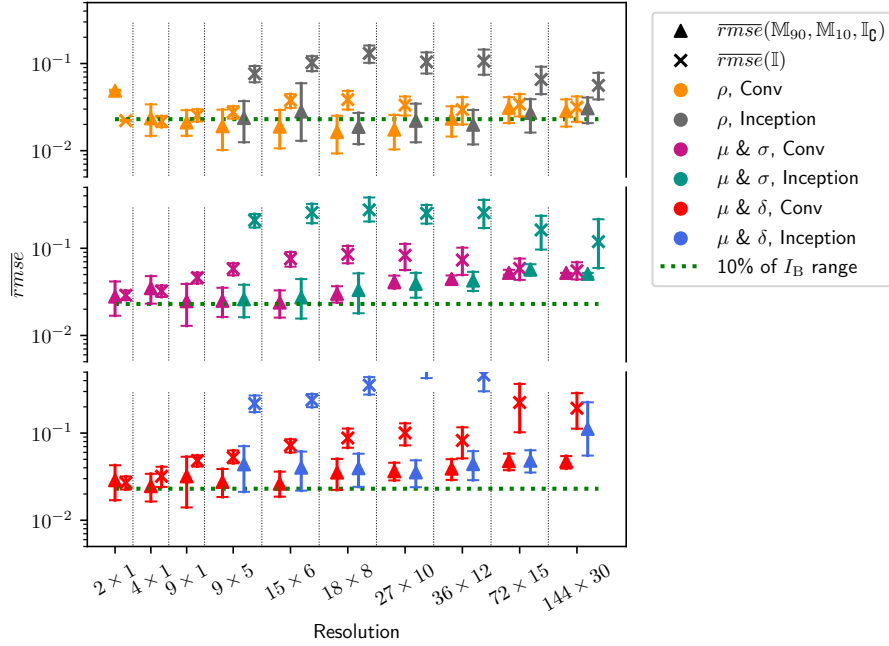
(b) *Logarithmic representation of $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\complement)$ and $\overline{rmse}(\mathbb{I})$*

Figure 6: *Average $rmse$ with 90% confidence level for various parametrizations, limiter partitionings and NN architectures. In case of smaller $rmse$ than 10% of the I_B range a sufficient reconstruction of I_B has been achieved.*

width of the $\overline{rmse}$ confidence intervals $CI(\mathbb{S},\mathbb{S},\mathbb{I})$ as compared to $CI(\mathbb{S})$ points towards the same reason.
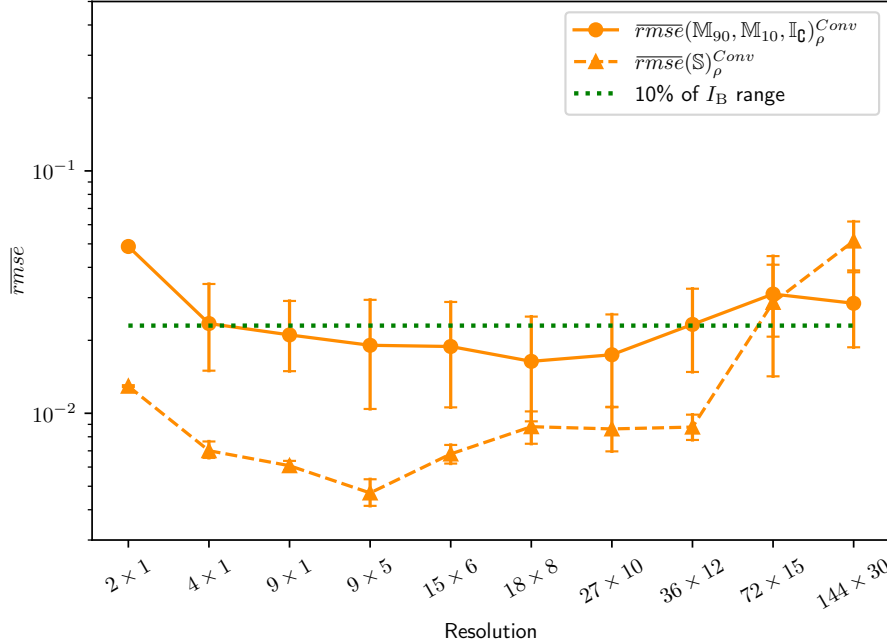

Figure 7: *The development of $\overline{rmse}(\mathbb{S})_{\rho}^{Conv}$ and $\overline{rmse}(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_{\mathbb{C}})_{\rho}^{Conv}$ are compared for the increasingly refined partitionings. Especially for coarse and extremly fine partitionings the behavior is similiar.*

Both parametrizations including μ show a decreasing performance with growing resolution. Only a marginal difference between $rmse(\mathbb{S})^{\mu \& \delta}$ and $rmse(\mathbb{S})^{\mu \& \sigma}$ can be observed. The $rmse\,(\mathbb{S})$ range seems independent of the architecture.

3.2. NN trained with simulation and experiment

Since $\mathbb{S}$ based training did not lead to sufficient NN performance for application to $\mathbb{I}$, some samples from $\mathbb{I}$ are provided during training and validation, i. e., $\mathbb{M}$ and $\mathbb{I}_{\mathbb{C}}$ as defined in Section 2.2 are used. With this procedure, we intend to force the NNs to consider patterns present in both, $\mathbb{S}$ and $\mathbb{I}$ during training. The performance of the NNs $(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_{\mathbb{C}})$ is compared to NNs trained, validated and tested with the small amount of available experimental data only, i. e., $(\mathbb{I}_{37},\mathbb{I}_{4},\mathbb{I}_{\mathbb{C}})$. Note, that the performance is tested with the same set $\mathbb{I}_{\mathbb{C}}$. Figure 6b depicts this comparison. The
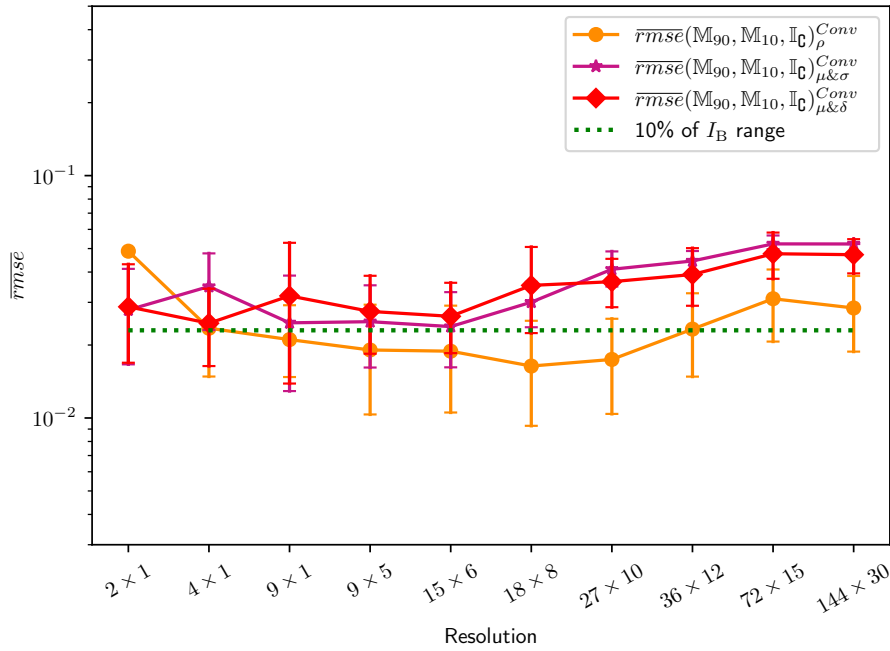
Figure 8: *The development of $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})_{\rho}^{Conv}$, $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})_{\mu\&\sigma}^{Conv}$ and $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})_{\mu\&\delta}^{Conv}$ are compared for the increasingly refined partitionings. Over all partitionings μ & σ and μ & δ behave similiar while ρ performces clearly better for fin partitionings.*
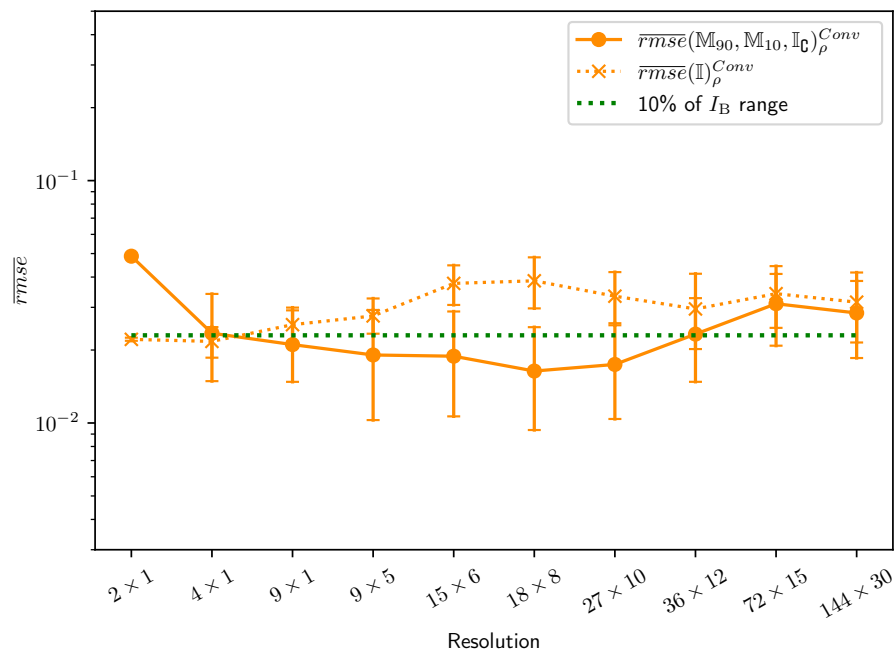
Figure 9: *The development of $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\complement)_\rho^{Conv}$ and $\overline{rmse}(\mathbb{I})_\rho^{Conv}$ are compared for the increasingly refined partitionings. Especially for partitionings between 9×5 and 27×10 the mixed training leads to a clearly better NN.*
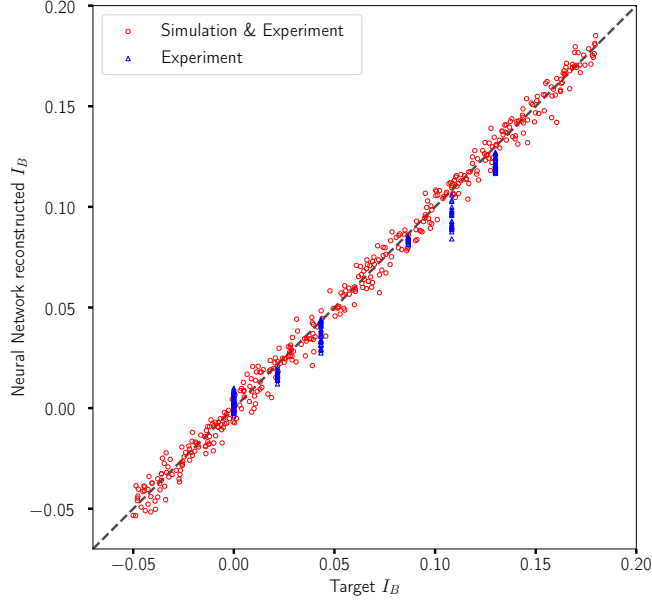
Figure 10: *For the median of $rmse(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})^{18 \times 8, Conv}_{\rho}$ the fitting performance on the validation set (red circles) and test set $\mathbb{I}_{\mathbb{C}}$ (blue triangles) is shown. On the x-axis are the target values and on the y-axis are the NN reconstructions. An ideal result would be the identity, shown by the dashed line.*

upper end of the occurring $rmse$ range is reduced by two orders of magnitude in this Figure as compared to 6a. As in the case of NNs trained with $\mathbb{S}$, the parametrizations by ρ yield the best $rmse$ for partitionings between 9×5 and 36×12 while the μ based parametrizations are best for one dimensional partitionings decreasing in performance with growing resolution as visualized in Figure 8.

For two dimensional partitionings, we observe in Figure 9 $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}}) < \overline{rmse}(\mathbb{I})$, as intended by training on $\mathbb{M}$. In the case of coarse partitioning, the opposite behavior is found, i.e., training with $\mathbb{M}$ has no advantages above direct training with $\mathbb{I}$.

The most impressive results were found for resolutions between 9×5 and 36×12, for $(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})^{\rho}_{conv}$. In order to display the good reconstruction quality, we choose a representative $rmse$, namely median $rmse$, of the NNs $(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})^{18 \times 8, Conv}_{\rho}$, as seen in Figure 10. The $rmse$ is 0.008 which is 3.5% of the I_{B} range, so the median $rmse$ is clearly below the 10% I_{B} range threshold. Changing the NN architecture to an inception model seems to slightly improve the performance for finer partitioning as seen in Figure 11.
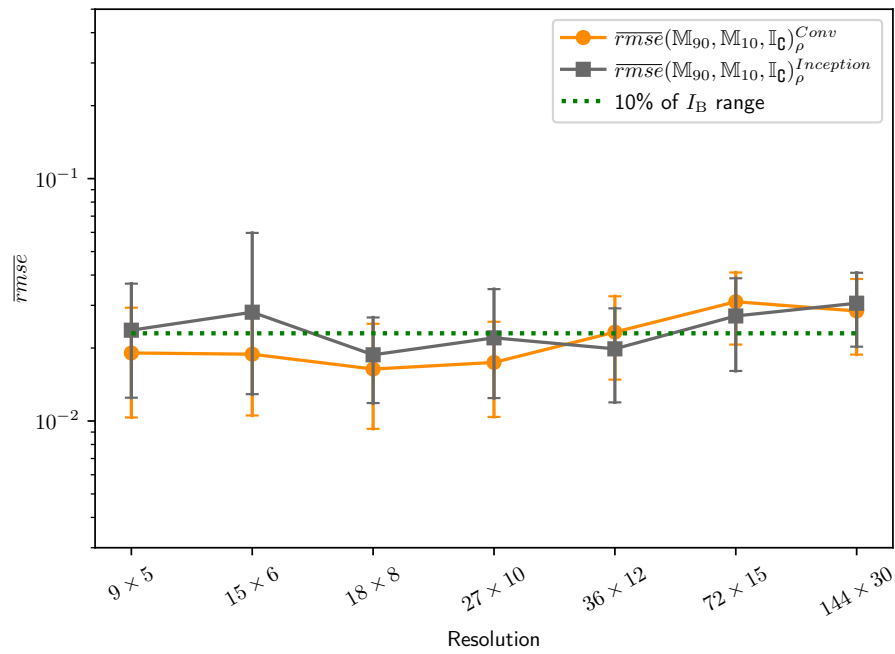
Figure 11: *The performance between the convolutional NN and the inception NN is compared by examine the $rmse(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})_{\rho}^{15 \times 6, Conv}$ with $rmse(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_{\mathbb{C}})_{\rho}^{15 \times 6, inception}$. The inception NN performs better for finer partitionings.*

The results outperform the NN in [1], where the best $rmse$ is 0.029.

4. Conclusion and Future Work

It was shown here that it is possible to reconstruct a relevant property of the W7-X magnetic edge from limiter heat load patterns. The main challenge was to deal with sparse experimental data given. Good performance for NNs trained with synthetic data created by vacuum field line diffusion simulation were found. A naive approach to apply NNs trained and validated with synthetic data to experimental data showed good performance only in a minority of cases. Mostly the patterns chosen by the NN in the training process were not present in the experimental observations. With this found, it was tried to train and validate NNs on a mixture of experimental and synthetic data. This approach resulted in convincing NN performance for certain NN input processing. Partitioning the limiter with resolutions between 9×5 and 36×12 and defining the NN input as the heat load in each part divided by the maximum heat load of all parts results in better performance compared to NNs trained, validated and tested with experimental data only. The low number of experimental results probably leads to overfitting in these nets but the added simulation data diminished these effects. This means, that we created NNs that extract relevant patterns from experimental as well as from synthetic data sets to reconstruct the magnetic edge parameter. With this systematic approach NNs were found to outperform the results found in [1].

The updated W7-X geometry with installed divertors will be the next object of interest. It is sensible to start the investigation with a parametrization based on a two dimensional partitioning of the heat load. Favoring one of the two examined NN architectures a priori and excluding the other is not possible because neither consitently outperforms the other. The reached results are satisfactory, however it remains future work to investigate other methods such as generative adversarial nets [12] to enhance the reconstruction performance when dealing with simulated and experimental data.

5. Acknowledgments

[1] D. Böckenhoff, M. Blatzheim, H. Hölbe, T. Sunn Pedersen, R. Labahn, Reconstruction of magnetic configurations in w7-x using artifical neural networks, Nuclear Fusion (2018).

[2] G. Grieger, I. Milch, Das fusionsexperiment wendelstein 7-x, Physik Journal 49 (1993) 1001–1005.

[3] T. Andreeva, Vacuum magnetic configurations of Wendelstein 7-X, Technical Report, Max-Planck-Institut für Plasmaphysik, Greifswald, 2002.

[4] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, The handbook of brain theory and neural networks 3361 (1995).

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, 2015.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25, pp. 1097–1105.

[7] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256.

[8] D. P. Kingma, J. L. Ba, Adam: a method for stochastic optimization, International Conference on Learning Representations 2015 (2015) 1–15.

[9] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, G. Brain, Tensorflow: A system for large-scale machine learning, in: OSDI, pp. 265–283.

[10] H. Renner, D. Sharma, J. Kißlinger, J. Boscary, H. Grote, R. Schneider, Physical aspects and design of the wendelstein 7-x divertor, Fusion Science and Technology 46 (2004) 318–326.

[11] T. J. Diciccio, B. Efron, Bootstrap confidence intervals, Statistical Science 11 (1996) 189–228.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks (2014).