DETERMINATION OF THE OPTIMAL VALUE OF THE 0TH ORDER TIKHONOV REGULARIZATION PARAMETER λ ON FAST-ION LOSS DETECTOR DATA USING DEEP LEARNING

Bo Simmendefeldt Schmidt, s113086

ABSTRACT

The objective of this paper is to investigate the capability of a deep neural network to determine the optimal value of the 0th order Tikhonov regularization parameter λ on fast-ion loss detector data. This is achieved by training a feed-forward neural network (FFNN) and a convolution neural network (CNN) on synthetically generated data simulating real-world measurements at the ASDEX Upgrade fusion reactor. Specifically, three datasets of increasing complexity are generated. The networks are trained on the three datasets and optimized with respect to several relevant parameters including the specific architecture, the pooling method, the filter size, and dropout. The architectures are then compared to the VGG16 network architecture. VGG16 outperforms the CNN and FFNN on all datasets by a small margin. The optimized CNN outperforms the FFNN. For a straight line fitted to the λ predictions as a function of the true λ's, both VGG16 and the CNN achieve a slope close to 1 for dataset 1. This indicates an almost perfect predictive capability in this domain. A CNN with the architecture as described in the present paper or with the VGG16 network architecture is recommended for determining the optimal value of λ for future 0th order Tikhonov regularization on fast-ion loss detector data.

Index Terms— FFNN, CNN, Tikhonov, regularization, pinhole, scintillator, noise, optimization

1. INTRODUCTION

Fast ions play an important role in key aspects of magnetically confined fusion plasmas. Their confinement is crucial for the optimal operation of these fusion devices since fast-ion losses may lead to irreversible damage to the plasma-facing components. One way to investigate the mechanisms leading to fast-ion losses is through using fast-ion loss detectors (FILDs). Scintillator-based FILDs consist of a scintillator plate mounted in a probe which is placed near the plasma in the far scrape-off layer. The escaping fast ions reach a pinhole in the probe head and pass through a 3D collimator that filters the ion trajectories. This allows a direct measurement of the Larmor radius and pitch angle of the fast ions[1]. See Fig. 1 for an illustration.

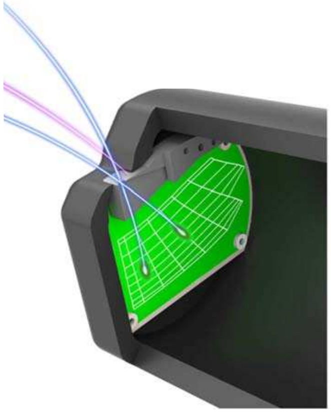


Fig. 1. A typical FILD probe head (dark gray) and the scintillator (green). Fast-ion orbits shown in red (blocked by the collimator) and blue (not blocked by the collimator). The strike map is overlayed on the scintillator. From [1].

To obtain useful information about the fast ions, the inverse problem needs to be solved: Let $M, N \in \mathbb{N}$ where M corresponds to the resolution of the scintillator and N the resolution of the measurements at the pinhole. Then, given a measurement of the velocity-space distribution at the scintillator S of dimension $M \times 1$, the undistorted velocity-space distribution of the absolute flux of fast-ion losses reaching the FILD head probe F of dimension $N \times 1$ needs to be obtained. The connection between the velocity-space distribution at the pinhole and the scintillator is described by the matrix equation

$$WF = S, \tag{1}$$

where W is the weight function matrix of dimension $M \times N$. W contains the probabilistic relationship between F and S. The unknown F is the velocity distribution of the fast-ion flux at the pinhole. Both the velocity-space distribution measured at the scintillator S and the weight function W are known.

The solution for F is mathematically an ill-posed problem: If $M > N$ the system is under-determined, and if $M < N$ the system is over-determined. In this case $M > N$ and so the system is under-determined. The 0th order Tikhonov regularization method can be used to find F in the ill-posed problem. In general, the Tikhonov regularization methods solve a minimization problem which can be expressed as:

$$\arg\min_F \left\| \begin{pmatrix} W \\ \lambda L \end{pmatrix} F - \begin{pmatrix} S \\ 0 \end{pmatrix} \right\|_2 \quad (2)$$

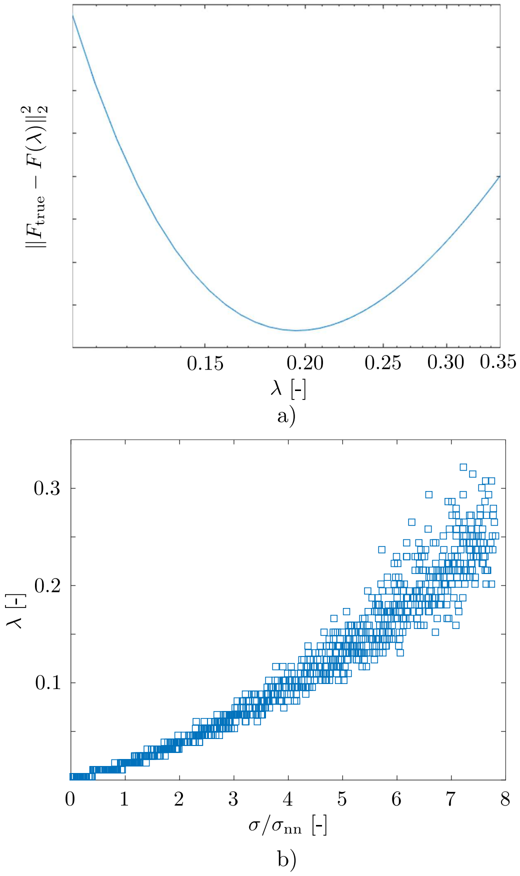
where W is a matrix composed of weight functions, S is the measurement matrix and F is the sought solution. The upper row minimizes the two-norm residual of $WF = S$, while the lower row penalizes large values of the two-norm of λLF. The definition of the L matrix can then be made based on the properties of the solution F that needs to be penalized. The regularization parameter λ controls the balance between the strength of the regularization condition and the goodness-of-fit to the data. Therefore, an optimal value for λ must be found.

2. DATA

A deep neural network is to be trained on synthetically produced FILD data where an optimal λ is known. Using non-negative Tikhonov regularization for a relevant range of λ's, the function $f(\lambda) := \|F_{\text{true}} - F(\lambda)\|_2^2$ has a parabolic shape with a clear minimum for the optimal choice of λ, where $F(\lambda)$ is the reconstruction of the pinhole found by solving the inverse problem using Tikhonov regularization for the specific λ. See Fig. 2a for an example of the previously mentioned parabola and its corresponding minimum. The best choice for λ occurs at the minimum value. The λ's for each measurement are determined in this way in the process of generating the training data.

The synthetic data is produced by choosing a physically relevant F matrix and performing the product $WF = S$. The neural network will be applied to real-world data and therefore has to determine an appropriate λ from noisy measurements. A large number of noisy measurements are generated by adding different levels of noise to S. The noise is assumed to be normally distributed. The noise level can therefore be represented by the standard deviation σ of the normal distribution. Let σ_{nn} denote the standard deviation of the measurement with no noise added. Then $\sigma/\sigma_{\text{nn}}$ can be used as a measure of the noise-level added to the synthetic data. See Fig. 2b for a plot of λ as a function of $\sigma/\sigma_{\text{nn}}$ for dataset 1. The $\sigma/\sigma_{\text{nn}}$ measure is not a standard measure for the noise level added to the data but is used here.

Three different datasets of increasing complexity are generated. Each dataset will be referenced as 'dataset 1', 'dataset 2', and 'dataset 3'. Each dataset contains a total of 1000 measurements. Dataset 1 consists of a single Gaussian blob in a region of pitch and gyroradius space that has previously been observed to contain signals in real-world ASDEX Upgrade data. The noise added to dataset 1 ranges from no noise to a very high level of noise to test the predictive capabilities of the deep neural networks in both ranges. See Fig. 3 for illustrations of the data with close to no noise and a large amount of noise with the signal hidden underneath. Dataset 2 consists of two Gaussian blobs with a noise range that can be expected in a real-world measurement. Dataset 3 consists of nine Gaussian blobs and thus contains data in most of the pitch-gyroradius space with a similar noise level as dataset 2. See Fig. 4 and 5 for an illustration of measurements from dataset 2 and dataset 3, respectively.



a)

b)

Fig. 2. a) The curve $f(\lambda) = \|F_{\text{true}} - F(\lambda)\|_2^2$ for a single scintillator measurement indicating the optimal choice of λ at the minimum of the parabola. b) The optimal choice of λ as a function of the standard deviation $\sigma/\sigma_{\text{nn}}$ in the Gaussian distribution used to generate the noise for all 1000 generated scintillator measurements. The subscript 'nn' indicates *no noise*, so the x-axis are values of the noise levels compared to the overall standard deviation of the noise-free scintillator measurement σ_{nn}. Note that for large noise levels of $\sigma/\sigma_{\text{nn}} \gtrsim 3$, the method used to determine the correct λ as used in the training dataset begins to identify different values of λ with the range of λ increasing as the noise increases.

Observe that for values of $\sigma/\sigma_{\text{nn}} \gtrsim 3$, the parabola method for determining λ returns different values of λ for the same noise level for *different* measurements (one square corresponds to one measurement), see Fig. 2b. I.e., as σ increases, an increasingly greater range of λ's are chosen as

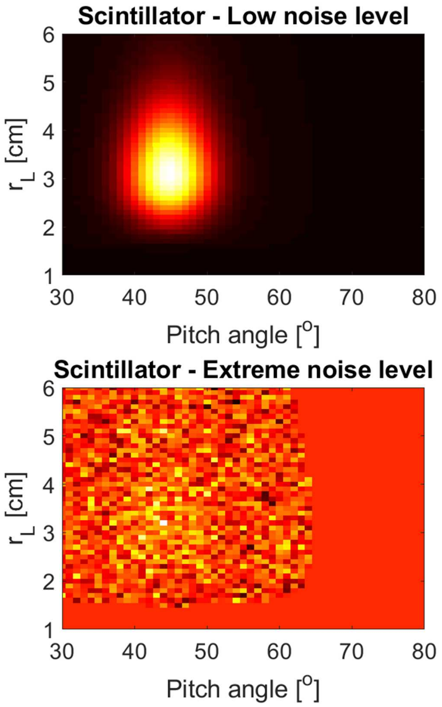


Fig. 3. Measurements from dataset 1 with a low level of noise (top) and an extremely high level of noise (bottom).

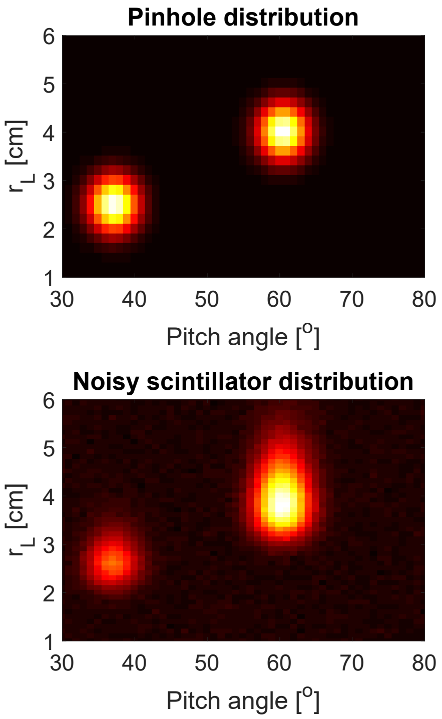


Fig. 4. The true fast-ion velocity distribution at the pinhole (top) and as measured on the scintillator (bottom). Measurements from dataset 2.

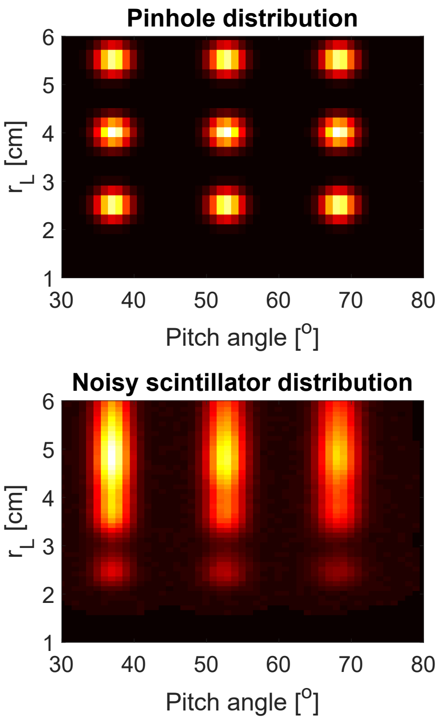


Fig. 5. The true fast-ion velocity distribution at the pinhole (top) and as measured on the scintillator (bottom). Measurements from dataset 3.

the optimal regularization parameter for the same value of σ. This indicates that the optimal λ parameter in Tikhonov regularization is not unique for a given noise level. This is due to the randomness of the noise: for each matrix entry, the noise value is determined by a random draw from a normal distribution. If large enough values of noise are added (when σ is large) in different places from measurement to measurement, it leads to different optimal regularization parameters. It is clear that this effect will be more pronounced for larger σ's. For low values of σ, the numerical value of the noise added to each matrix entry is relatively small and so the random nature of adding the noise does not change the measurement significantly, and so a smaller range of λ's are determined to be the optimal choice.

Note that this effect of having more than one optimal λ for the same level of noise does not *a priori* negatively effect the training of the neural network: The network has no information about the noise level but is only given the measurement data as input and the optimal choice of λ as a training target. It is then up to the network to learn the patterns between different levels of noise and the inherent randomness of the noise and the optimal choice of λ. It is possible that a greater number than 1000 measurements is needed for the network to find the connections between the noise levels and optimal choice of λ.

3. METHODS

An FFNN and a CNN are trained on data where the optimal λ values are known. The networks are then tested by predicting λ's for datasets where the true λ's are known. The predictions are then plotted as data points $(\lambda_{\text{true}}, \lambda_{\text{pred}})$ with the true λ's as x-values and the predicted λ's as y-values. A linear regression is performed on the resulting data points. The correlation coefficient R^2 and the slope α of the line are then used as performance measures. Both R^2 and α should numerically both be near 1, where R^2 indicates the amount that the variance of the predicted λ's is explained by the variance of the true λ's and α indicates how well the numerical values of the predicted λ's correspond to the numerical values of the true λ's. A slope of 1 and a correlation coefficient of 1 shows a one-to-one correspondence between predicted λ's and true λ's, which corresponds to 100% accuracy of the neural network. A slope $\alpha < 1$ shows that the network predicts values of λ that are below the true λ's.

Data measurements as obtained from a FILD are given as a matrix where the ith row and jth column correspond to a specific value of the gyroradius and pitch angle, respectively. The matrix dimensions in this case are 81×91. The matrix is transformed to a 1D vector of length 7371 such that Tikhonov regularization can be applied to the equation $WF = S$ and the optimal F obtained. Therefore, it can be argued that both inputs should be able to be used by the neural network: the entire 81×91 matrix and the transformed 1D vector. *A priori* it is therefore possible that both an FFNN type network and a CNN type network make good predictions for λ. This turns out to be true.

Several parameters are varied and their impact on the predictive performance of the deep neural networks measured through R^2 and α. These parameters include

- Network architecture
- Number of nodes
- CNN kernel size
- Max-pooling
- Stride length
- Dropout
- Padding

Network architecture is tested since the number and type of layers for optimal predictive capability is *a priori* unknown. The obtained optimal architecture is compared to the VGG16 CNN architecture, which won the 2014 ILSVR (ImageNet) competition[2]. Similarly, the optimal number of nodes is unknown. The number of nodes is expected to be high for the FFNN since it needs to handle 1D data with 7371 attributes. Max-pooling is used and avg-pooling is discarded since max-pooling works as a noise reduction and a dimensionality reduction mechanism, which is exactly what is needed in this noisy and high-dimensional case. Similarly, the size of the pooling matrix could be important and the stride length. Finally, dropout is used as optimization mechanism for each individual node.

4. RESULTS

4.1. Form of Data Input

The FFNN was used to test if both forms of data input were appropriate. First, the single vector input of 7371 attributes was investigated. The 'Adam' optimizer was used for all testing as it has been found to work well in practice and outperforms other adaptive techniques. ReLU was used as activation function for every layer except the output layer, which was not given an activation function. A batch size of 20 and 10 epochs was used. The initial architecture consisted of a single dense input and output layer. Increasing the number of nodes in the first layer above 256 did not further improve the network. Stacking more dense layers improves the network up until a correlation coefficient of about 85% and a slope around 0.5. The FFNN predicts values of λ much smaller than the actual values. This is a very poor model and does not improve further by adjusting the parameters mentioned in the previous section. The results for the FFNN with the 81×91 matrix as input are much better as described below, and so the matrix input was used for the remainder of the training and testing.

4.2. FFNN

To determine a good architecture for the network it was tested how many dense layers and the number of nodes in each layer that result in improvements in correlation coefficient and slope and compared to a baseline model. This testing was done on all of the datasets, but the same performance trends were evident across all datasets. For simplicity, the number of nodes in each layer was chosen as 2^n for $n = 0, 1, 2, \ldots$ counting from the output layer. The performance of the network is best for 11 layers with $R^2 = 90.12\%$ and $\alpha = 0.9655$, see Table 1. This is clearly a better model than baseline. Note that the R^2 are comparable for most tests, but the accuracy of the prediction (as given by α) is best for 11 layers. The model architecture is shown in Fig. 6a. Changing activation function from ReLU to Tanh does not change the correlation coefficient or the slope. Implementing dropout of varying percentages after one to all of the layers decreases the value of the slope, see Table 2 for implementation of 10% dropout layers. Therefore it was decided not to use dropout at all. The λ predictions vs. true λ for dataset 1 are shown in Fig. 7a.

The FFNN performs significantly worse on dataset 2 and 3. The correlation coefficient is $R^2 = 9.285\%$ and $\alpha = 0.5674$ for dataset 2 and $R^2 = 57.02\%$ and $\alpha = 0.7856$ for

Archit.	BL	6 layers	10 layers	11 layers	12 layers
R^2	1	0.8478	0.8966	0.9012	0.9012
α	0	0.8382	0.8606	0.9655	0.8660

Table 1. FFNN performance as a function of the number of layers. The best performance is achieved for 11 layers corresponding to an input layer with 1024 nodes. Archit. = "architecture", BL = "baseline".

Dropout	0 layers	1 layer	2 layers	4 layers	6 layers
R^2	0.9012	0.9110	0.9045	0.9221	0.9314
α	0.9655	0.8658	0.8652	0.8156	0.7053

Table 2. Number of dropout layers with 10% dropout implemented after 0-6 dense layers in the FFNN. Note that R^2 improves slightly but α decreases significantly with more dropout layers.

dataset 3. The reason for the very large performance drop for dataset 2 is unknown and needs to be further investigated.

4.3. CNN

The 'Adam' optimizer was used for all testing. A batch size of 20 and 10 epochs was initially used. The initial architecture consisted of a single convolution layer and a dense output layer. Similar testing strategies as was used to find the FFNN architecture were used to find the optimal CNN architecture. The architecture was then tested for different kernels, padding, max-pooling sizes, stride length, dropout implementation. See Table 3 and 4 for the specific performance evaluations for stride length and padding. The final architecture is illustrated in Fig. 6b. The optimal CNN has a (2, 2) kernel size, no padding, a (2, 2) max-pooling size, a stride length of (1, 1), and no dropout. See Table 5 for the performance evaluation for the CNN on dataset 1, 2, and 3.

Stride length	(1, 1)	(2, 1)	(1, 2)	(2, 2)
R^2	0.9523	0.9270	0.9334	0.9385
α	0.9624	0.9077	0.9185	0.9342

Table 3. CNN performance as a function of the stride length.

Padding	None ("valid")	Keep size ("same")
R^2	0.9523	0.9436
α	0.9624	0.9323

Table 4. CNN performance as a function of the padding.

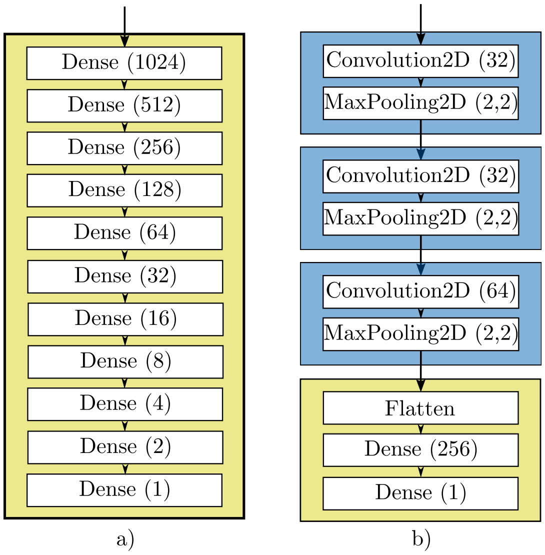


Fig. 6. The architectures for a) FFNN and b) CNN.

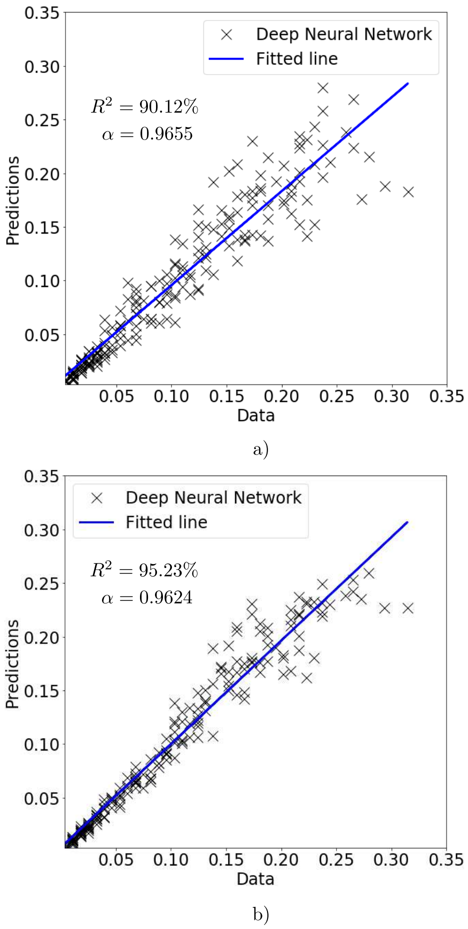


Fig. 7. Predictions of λ vs. the true λ values for a) FFNN and b) CNN on dataset 1. α indicates the slope of the fitted line.

FFNN	Dataset 1	Dataset 2	Dataset 3
R^2	0.9012	0.09285	0.5702
α	0.9655	0.5674	0.7856
CNN	Dataset 1	Dataset 2	Dataset 3
R^2	0.9460	0.3891	0.3215
α	0.9608	0.9587	0.8814
VGG16	Dataset 1	Dataset 2	Dataset 3
R^2	0.9503	0.7452	0.6986
α	0.9830	0.6343	0.7731

Table 5. Performance of the FFNN, CNN, and VGG16 networks.

4.4. VGG16

The VGG16 architecture as given in [2] seems to outperform the above-mentioned networks on both R^2 and α on all three datasets. Note that the VGG16 is not pretrained and only trained on the data from this project.

The VGG16 architecture differs from the CNN above in several ways. It contains four convolution blocks containing two and three convolution layers in succession before applying a max-pooling layer. The max-pooling layer has a stride of (2, 2). Padding is added such that the size of the matrices remains the same. The kernel size of the convolution layers are (3, 3). Two dense layers are implemented before the final output layer. The number of nodes in both convolution layers and dense layers are larger than used in the CNN above.

VGG16 is only marginally better than the above-mentioned CNN. See Table 5.

4.5. Real-World Data

The CNN was chosen as the network to use on real-world data. The CNN was trained on all three datasets and saved and subsequently applied to FILD data shot 34559 from ASDEX Upgrade. The shot lasts 3 seconds with 50 measurements per second resulting in 150 measurements. The CNN predicts an optimal λ for each of the 150 measurements. Tikhonov regularization is then applied with the given λs and 150 pinhole reconstructions calculated. These are all combined to create the movie 'movie_asdex.avi'. Previous attempts at such reconstruction comprise using a single λ value for all measurements. However, with this approach each frame of the movie (corresponding to one measurement) is optimized with respect to the optimal λ. The new movie can be compared to the movie created using the single λ value. The older movie can be found as 'movie_asdex_old.avi'. One particular improvement seems to be that there is much less noise using the new technique and more precise reconstructions of the measurements. This indicates that using the CNN to determine λ works and is an improvement of earlier methods.

5. DISCUSSION AND SUMMARY

This paper investigates the predictive capability of deep neural networks to predict the optimal choice of 0th order Tikhonov regularization parameter λ on three datasets of fast-ion loss detector data. The optimal network architectures for a feed-forward neural network and a convolution neural network are determined by how close the correlation coefficient R^2 and the slope α of the line fitted to the data points of the predicted λ values as a function of the true λ values are to 1 numerically. The FFNN and CNN are then compared to the VGG16 network. The performance results are summarized in Table 5. VGG16 is seen to outperform both the FFNN and the CNN by a small margin.

All networks perform well on dataset 1. This is a positive finding for at least two reasons: 1) Dataset 1 contains both measurements with very little noise and measurements with a very large amount of noise. However, the networks have learned to predict the correct λ's even for measurements with a lot of noise. 2) Dataset 1 closely resembles true measurement data. This indicates that the techniques can be applied to real-world data with good results.

The networks perform worse on dataset 2 and 3. This is expected since the datasets are of higher complexity. It is not clear why the FFNN has such a large performance drop from dataset 1 to dataset 2 and 3. It is possible that the higher complexity is too much for the network. This needs to be investigated further. One way could be to simply let the network train on more data.

The predictive capabilities of the networks are worse for measurements with a large amount of noise. This is illustrated in Fig. 7 where the spread of the data increases with increasing λ (where higher λ correspond to higher levels of noise). A natural next step is to train the networks on more data in this noisier region to see if this can help improve the network's predictive capabilities for these noise levels.

Finally, the CNN was trained on dataset 1-3 and subsequently tested on 150 real-world data measurements from ASDEX Upgrade. The measurements originate from a single shot lasting 3 seconds with 50 measurements per second. The CNN returns a λ for each measurement and the Tikhonov regularization was then performed to find the velocity distribution of the fast-ion flux at the pinhole. The 150 measurements are combined into a movie that show improvements over previous analyses by containing less noise and having more precise reconstructions of the signals.

Future work includes training the network on more data to improve the predictive capability on noisy data as well as attempting to improve the predictive capability on datasets of higher complexity.

Future usage includes applying the networks described in this paper to determine the optimal 0th order Tikhonov regularization parameter λ on fast-ion loss detector data.

6. REFERENCES

[1] Galdon-Quiroga, J. et al. (2018). Velocity-space sensitivity and tomography of scintillator-based fast-ion loss detectors. *Plasma Physics and Controlled Fusion*, 60(10), p.105005.

[2] Medium. (2019). *Step by step VGG16 implementation in Keras for beginners*. [online] Available at: `https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c` [Accessed 29 Dec. 2019].

Github

Link to Github repository: `https://github.com/BoSchmidt/02456_deep_learning_project`