Plasma Mode Classification using 2D-Convolutional Neural Networks

Bruno Rodriguez, Simone Roznowicz, Antoine Salaün

I. Abstract

In a nuclear fusion experiment, plasma changes of confinement state over time. Between states low (L), dithering (D), and high (H), the physical behavior of the plasma changes. Thus, the plasma state must be classified using four measurements: PD, FIR, DML, and IP.

Previous projects (*Classification of tokamak plasma confinement states with convolutional recurrent neural networks* [1] and *Plasma confinement mode classification using a sequence-to-sequence neural network with attention* [3]) reach a satisfying classification accuracy using deep-learning. In particular, LSTM neural networks are especially useful for this task.

In this project, in collaboration with Alessandro Pau, LSTM was avoided in order to explore different horizons. Namely, the key idea is to shape the dataset as 2D-images (four features over a window of time) that will be fed to a 2D-convolutional neural network (CNN). Moreover, a special scope was put on data analysis in order to identify patterns in the dataset.

With an accuracy of 87.83%, the results are satisfying. However, the real interest of this paper is in the validation of theses four hypotheses:

1) PCA demonstrates that unsupervised clustering should be efficient on this dataset.
2) 2D-images are a very efficient way of feeding the data to the NN.
3) The FIR measurement is not useful to the classification.
4) The Fourier transform of PD carries rich information.

II. Introduction

A. Objectives

The project focuses on the classification of plasma confinement states: for any given data point, plasma is classified in High(H), Low(L), or Dithering(D) mode. The objective is to distinguish each mode according to four measurements in the tokamak: PD, FIR, DML, and IP.

In order to accomplish the target, the following aspects were deepened: Data Analysis, PCA decomposition, Feature engineering, and Implementation of a 2D-Convolutional Neural Network.

This project is following the tracks of F. Matos et al 2020 Nucl. Fusion 60 036022 [1]. However, it aims at bringing a new method for plasma classification with the use of 2D-convolutions on images made of the four features over a fixed time-window.

The project answers the question "Does this technique help improve the classification of the plasma-state ?". For this reason, it was decided not to use Long short-term memory, namely LSTM (even though it is very efficient), since its usefulness for this problem has already been proven.

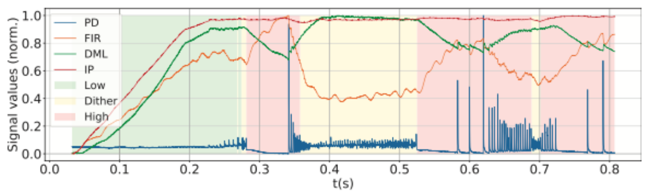

Fig. 1. Classification of the plasma state on TCV shot 32195 from F. Matos et al 2020 Nucl. Fusion 60 036022 [1]

B. Physics background

To use plasma fusion for energy production, the plasma must be in high-mode state. Thus, detecting the plasma-state is a priority. Plasma physics proved that the four measurements (PD, FIR, DML, and IP) provide enough information to characterize the plasma-state. However, until now, only an expert eye can do it. It is imprecise and time-consuming. Therefore, it makes sense to automatize the process. One must be aware that two experts may manually classify each data point inconsistently. This shows that the classification is not trivial (especially in transitioning phases). Moreover, as pulses are becoming longer, in the near-term future live classification of the plasma-state will be needed.

When a pulse experiment starts, the plasma is in low-mode (L). The input energy increases and the plasma can spontaneously transition to high-mode (H). The dithering mode (D) is a transition state between L and H and is characterized by instabilities. Its classification may be sometimes challenging.

The four measurements are :

1) PD - Photodiode signal. It measures the visible radiation emitted by the plasma at 653.3 nm
2) FIR - Interferometer signal. It measures the electron density in the plasma at 14 different spots in the TCV
3) DML - Diamagnetic Loop signal. It measures the toroidal magnetic flux of the plasma
4) IP - Plasma Current signal. It measures the total plasma electric current

C. Steps that were followed

The project has some main areas of focus, which are briefly summarized here:

1) Cleansing of the dataset: outlier removal and normalization.
2) Data visualization and feature engineering: the analyses were done either for the whole dataset or pulse-wise.
3) Creation of the model: splitting of the dataset according to each pulse and further splitting of every obtained matrix into images. The obtained samples are passed to a CNN. As a result, for each sample, a probability vector of 3 entries is obtained, each corresponding to the prediction probability of the sample being made of L, D, or H.

III. Data Analysis

A. *Data preprocessing*

Our original data set is made of 7 features: time, IP, PD, FIR, WP, LDH and pulse and contains 2615471 samples. Moreover, our dataset is composed of 172 pulses or experiments. Pulses last from one to four milliseconds and the data has been harmonized at a sampling frequency of 10'000 Hz. At first, the dataset was properly cleaned by removing those samples that do not contain useful information to determine the state of plasma. This leads to a processed data set of 2558179 samples.

The dataset is then divided into 172 experiments. Hence, each experiment may be led under different circumstances, and the trend of the plasma evolution is, therefore, peculiar for each experiment. Afterwards, nested sub matrices (20 consecutive rows x 4 features) were created for each experiment. Each of these 20×4 sub matrices represents a sample (which can be seen as a picture) to be fed to the neural network.

Subsequently, we randomly shuffle our data samples, and split them into training and testing samples according to a fixed splitting ratio.

B. *Distribution of the data*

Analyzing the distribution of the data, for every given feature, may be beneficial in developing a more performing model. In fact, a deeper knowledge in the domain of nuclear fusion physics may lead to infer some useful information in the dataset. For this reason, we plotted the distribution of each feature in the dataset in 2, allowing the interpretation to further physics studies.

C. *PCA*

To gain a visual understanding of the data, we computed the singular value decomposition SVD or PCA of the data matrix in 2 and 3 dimensions. As observed in Fig.3, this resulted in a clear appearance of clusters; specifically, the D mode is between the L and H mode, as it represents the transition state. Although we also computed the PCA for a fixed experiment, Fig.3 shows the PCA of the whole data set once filtered. We did not use the time, pulse and LDH features to compute the actual PCA.

Now, let us recall that the SVD decomposition of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ reads: $\mathbf{A} = \mathbf{U\Sigma V}^T$; in our case m is the number of samples, n is the number of features and $m >> n$.

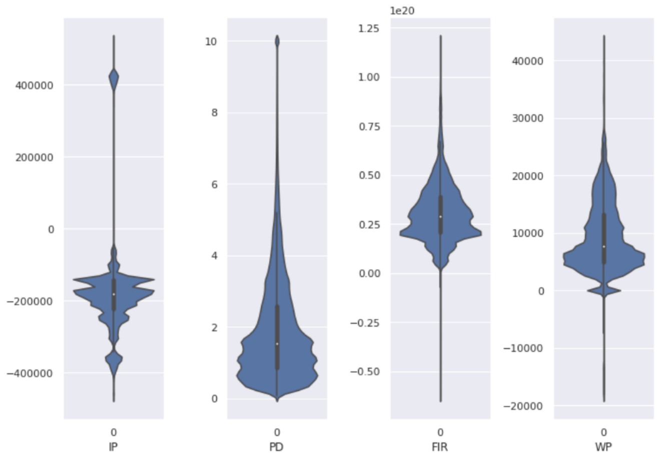

Fig. 2. 3D Principal component analysis visualization, computation made on the whole dataset

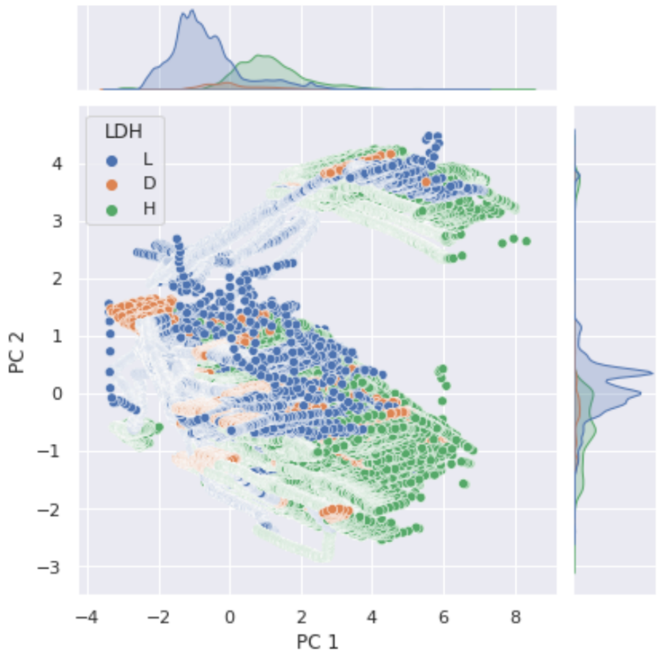

Fig. 3. PCA in 3D on the whole dataset

As described below, m is very large and varies from one experiment to another and $n = 4$. One of the most significant usage of the SVD is dimensionality reduction; this is our main goal to compute the SVD for each experiment. Let us assume that the matrix $\mathbf{A}$ corresponds to a fixed experiment with m samples and $n = 4$ features; then we compute its SVD and compute $\mathbf{U}^T\mathbf{A}$; this to rotate $\mathbf{A}$ in such a manner we have a diagonal matrix. Nevertheless, this strategy does work due to the size of $\mathbf{A}$, from which is significantly expensive to compute the full matrix $\mathbf{U}$. If we had computed the reduced SVD for each data set, we would have ended up with data sets of size (4, 4) for every single experiment; this happens since our data sets are of rank 4. That is $\mathbf{U}^T_{4\times m} \times \mathbf{A}_{m\times 4} \in \mathbb{R}^{4\times 4}$.

D. *Feature engineering*

The Fourier Transform is an additional feature that was introduced to the benefit of an analysis in the domain of frequencies. Hence, the PD measurements oscillates with frequent spikes in the H and D modes. Moreover, the frequencies of the spikes are different in H and D mode. A change of frequency may indicate a transition from H to D or from D to H. The Fourier transform of PD will be rich in information for the classifier. It is computed, pulse-wise, by averaging over frequencies the spectrogram of PD. The spectrogram is computed on a moving window of length 512 (i.e. 51.2 ms) that moves by jumps of 64 steps.

The final data set is divided into training and testing set using the following features: IP, PD, WP and Fourier transform of PD. The reason why is explained in next sections. Consequently, our train data and test sets are composed of 57000 and 6000 single-channels samples of size (40, 4), respectively.

IV. Model

A. *The neural network*

Following F. Matos et al (2020) [1], a convolutional neural network (5 layers) is implemented. The architecture of the convolutional neuronal network (CNN) that we use to classify states is shown in 4. In the input of our model, we have N single-channel pictures of size (40, 4). The first part of the CNN is composed of 3 consecutive convolutional layers whose mission is to extract the most relevant features; in the output of the convolution layers, we have N 32-channel pictures of size (11×2). These are afterwards flatten and passed through a set of 5 hidden layers, which corresponds to the second part of our model: its purpose is to extract high-level features from the data coming from the convolutions. We apply the softmax function to the output of the hidden layers to have values between 0 and 1. We design the CNN using the Pytorch framework, use cross entropy as loss function and train with the Adam optimizer using $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and learning rate = $1 \cdot 10^{-3}$. We justify the choice of components of our CNN in next sections.

B. *Measure of model accuracy*

Our approach is to have probabilities as outputs. Assume that $\mathbf{v} \in \mathbb{R}^3$ with $0 \leq v_i \leq 1$ corresponds to the output of the CNN. This holds since we use soft max in the output. Each v_i represents the likelihood of the $\mathbf{v}$ being L, D or H. Then to measure the accuracy of the model, we count how many samples are classified correctly, that is

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1} \left(\arg \max_{i \in [1,3]} \text{target}_i = \arg \max_{i \in [1,3]} v_i \right) \tag{1}$$

where $\mathbf{1}(\cdot)$ is the indicator function, target_i is the true label of the i-th sample and N is the number of samples in either the training or testing set. This means that we select the index of the output with the highest likelihood and compare to the index of the true label.

V. Results and Discussion

A. *Approach justification*

In [1], a CCN using 1D convolutions is proposed and discussed. [1] was taken as inspiration for this work. However, it was decided to implement 2D convolutions since this led to more flexibility in terms of data manipulation such as image filtering or creating multiple channels for a given sample. The image approach proposed here tries to capture how an L, D or H state looks like, independently of the experiment.

Additionally, it should be remarked that several architectures were tried and the one shown in 4 obtained the highest accuracy. It is worth pointing out that the data set itself plays a crucial role. That is, first IP, PD, FIR, WP were used as features for training, achieving an accuracy close to 50% for train and test. However, it was realized that the order of magnitude of the FIR feature (10^{18}) is substantially different from the orders of magnitude of the other features. Therefore, it was believed that the proposed model was paying too much attention to FIR; thus, FIR was removed from the data sets.

The features were augmented by polynomial expansion of each feature, which led to poor accuracy, and added extra samples to the data set by shuffling rows-columns for all samples of the H state; However, none of these strategies worked since these were performed including the FIR feature. In addition, it should be pointed out that by adding more hidden layers to the network and using a few neurons in such layers, the accuracy is improved. One may believe that this happens because, by using more layers, the model is capable of capturing more complex relations between the data. Isummarizes those parameters of the whole model and data set that have more importance on the accuracy.

B. *Window size*

The 2D-images are generated for the four features over a window of time. The choice of the window size has consequences on the results. Thus, it was decided to test the window size effect on test error in an early version of the model.

Choosing a lower window size lets us create more images and thus augment the size of the dataset. Smaller window size thus require more computing power. On the other side, bigger images contain more information. It was noticed that this richer data however demands more epochs to train the model. Good fits can be found for windows smaller than 50 samples.

C. *Results*

The obtained results are summarized in the following table I.

Win. size	learn. rate	Data ratio	Optimizer	Drop Out	Train error	Test error
10	10^{-4}	0.5	SGD	0.5	38.87%	38.51%
20	10^{-4}	0.5	ADAMW	0.5	26.87%	26.30%
20	10^{-4}	0.5	ADAM	0.5	25.66%	25.05%
20	10^{-3}	0.5	ADAM	0.5	21.26%	20.53%
40	10^{-3}	0.1	ADAM	0.1	**11.52%**	**12.17%**

TABLE I
MODEL ACCURACY FOR DIFFERENT PARAMETERS

To get the best train and test accuracy, we trained the model varying some parameters as seen in I; likewise, the *window size*, the *learning rate*, the *data splitting ratio* and the *optimizer*.

Larger windows size led to richer data and thus a smaller train and test error. It was also noticed that the learning rate plays an important role on the training as expected. In earlier versions of the model, only 50% of the data was used for testing. However, by lowering this ratio, the training could be improved and the test error dropped.

Furthermore, different optimizers were tried but the best fit was found with ADAM. Finally, dropout helps learning by randomly turning off and on some neurons in the hidden layers. However, the probability value used in dropout should not be too high. This model is optimal for a dropout probability of 0.1. In the end, the optimal model described in 4 has a train error of 11.52% and a test error of 12.17%.

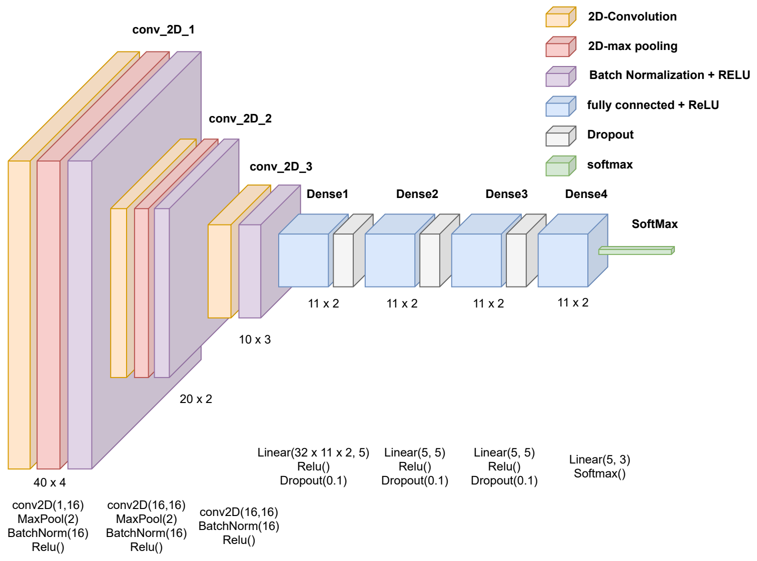
2D-Convolution
2D-max pooling
Batch Normalization + RELU
fully connected + ReLU
Dropout
softmax
conv_2D_1
conv_2D_2
conv_2D_3
Dense1
Dense2
Dense3
Dense4
SoftMax
11 x 2
11 x 2
11 x 2
11 x 2
10 x 3
20 x 2
40 x 4
Linear(32 x 11 x 2, 5) Relu() Dropout(0.1)
Linear(5, 5) Relu() Dropout(0.1)
Linear(5, 5) Relu() Dropout(0.1)
Linear(5, 3) Softmax()
conv2D(1,16) MaxPool(2) BatchNorm(16) Relu()
conv2D(16,16) MaxPool(2) BatchNorm(16) Relu()
conv2D(16,16) BatchNorm(16) Relu()

Fig. 4. Neural Network

VI. CONCLUSION

This model is a proof of concept: even without implementing LSTM, 2D-convolutions are an efficient way of classifying plasma-state data. Notwithstanding time constraints, the focus on data visualization and feature engineering, we managed to reach appreciable results. 87.83% accuracy was reached with a light model that can be trained in a few seconds on a regular desktop laptop.

The next step would be to combine this network with a LSTM similar to the one described in F. Matos et al (2020) [1]. It is reasonable to believe that it could improve the classification accuracy even more. More specifically, the identification process of the network may take advantage of it. Intuitively, as the Dithering state represents the transition between L and H or vice versa, it is clear that it is a time-dependent phenomenon. Thus, adding long and short memory to the network could help establish whether or not a transition state is actually occurring.

REFERENCES

[1] F. Matos et al (2020) *Classification of tokamak plasma confinement states with convolutional recurrent neural networks*, Nucl. Fusion 60 036022

[2] S. Coda et al (2019) *Physics research on the TCV tokamak facility: from conventional to alternative scenarios and beyond*, Nucl. Fusion 59 112023

[3] F. Matos et al (2021) *Plasma confinement mode classification using a sequence-to-sequence neural network with attention*, Nucl. Fusion 61 046019