CCFE
CULHAM CENTRE FOR FUSION ENERGY

UK Atomic Energy Authority

UKAEA-CCFE-PR(20)123

Vignesh Gopakumar, D. Samaddar

Image Mapping the Temporal Evolution of Edge Characteristics using Neural Networks

Image Mapping the Temporal Evolution of Edge Characteristics using Neural Networks

Vignesh Gopakumar, D. Samaddar

This is a preprint of a paper submitted for publication in
Machine Learning : Science and Technology

Image Mapping the Temporal Evolution of Edge Characteristics in Tokamaks using Neural Networks

Vignesh Gopakumar[1] [2]

[1]University of Stuttgart, Stuttgart

[2]Culham Centre for Fusion Energy, Oxford

E-mail: vignesh.gopakumar@ukaea.uk

D. Samaddar[1]

[1]Culham Centre for Fusion Energy, Oxford

E-mail: debasmita.samaddar@ukaea.uk

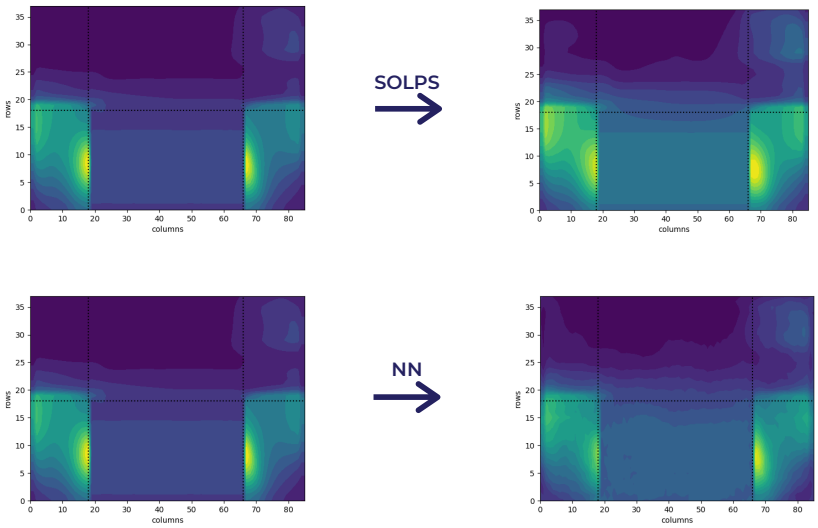


Figure 1: Mapping the temporal evolution of electron density. The set of images on top depicts the evolution from the initial state to the final state as solved by the SOLPS framework. Similarly the set in the bottom, represents the evolution characterised by our novel FCN approach.

Abstract. We propose a method for data-driven modelling of the temporal evolution of the plasma and neutral characteristics at the edge of a Tokamak using neural networks. Our method proposes a novel fully convolutional network that is capable of predicting the time evolution of the physics parameters along the two-dimensional representation of the poloidal cross section. More specifically, we target the evolution

of the temperatures and parallel velocities of the electrons, ions and neutral particles at the edge. The central challenge in this context is in modelling together the different physics principles encapsulated in the evolution of plasma and the neutrals. We demonstrate that the inherent differences in non-linear behaviour can be addressed by forking the network to process the plasma and neutral information individually before integrating as a holistic system. Our approach takes into account the spatial dependencies of the physics parameters across the grid while performing the temporal mappings, ensuring that the physics dependencies are factored in and not lost to the black-box. Having used the conventional edge plasma-neutral solver code SOLPS to build the synthetic dataset, our method demonstrates a computational gain of over 5 orders of magnitude without a considerable compromise on accuracy.

1. Introduction

Simulating magnetically confined fusion plasma in a tokamak is considered to be extremely challenging particularly at the edge of the device. Both perpendicular and parallel velocities (with respect to the magnetic field) are of significant importance in this region, also referred to as the scrape off layer (SOL). In addition to that, the interaction with the machine's plasma facing components introduces a lot of impurities into the plasma. Processes such as charge exchange and recombination contribute to the presence of neutral particles in the plasma that play a significant role in the dynamics – much of which is almost absent at the plasma core [1]. The additional complexities of the physics in the SOL lead to large differences in characteristic timescales in simulations of the plasma at the edge and the core [2]. Modelling the entire plasma within the tokamak (Integrated Tokamak Modeling (ITM) [3]) thus becomes particularly challenging. We suggest a novel approach that attempts to eradicate this problem by utilising the function approximation capabilities of neural networks to model the evolution of both the plasma and neutral behaviour at the edge.

We have demonstrated the impact of our approach on the SOLPS (Scrape Off Layer Plasma Simulator) framework, a comprehensive code suite that aims to solve the edge plasma physics [4]. By utilising synthetic data generated by the SOLPS code, we have shown how our model is capable of mapping the evolution of both the edge plasma and the neutral parameters. We model two interlinked physical systems characterised by two different nonlinear behaviours with a computational gain of over 5 orders of magnitude compared to the traditional numerical SOLPS approach.

2. Concept

Neural Networks have been consistently proven to perform as 'Universal Function Approximators' over the years, performing exceptionally well in modeling hefty nonlinear functions [5] [6]. The temporal evolution of plasma and neutrals within a specified

spatial grid is described through highly complex nonlinear transformations [7]. The charged species of the plasma and the neutrals are modelled differently (as desscribed in section 3). A time instant of the plasma and neutral evolution is characterised by various physics profiles spread across the poloidal cross-section. Each point in this grid represents the value of a physical parameter at that point in space. The evolution of this parameter in each of the grid points across time is represented by a strong nonlinear function that is dependent on itself as well as with the nonlinearities of the physical parameters encapsulated in the neighboring grid points. To capture the two different physics models at the edge described as nonlinearities with local relational dependencies, we have designed a Fully Convolutional Neural Network with forked inputs and outputs.

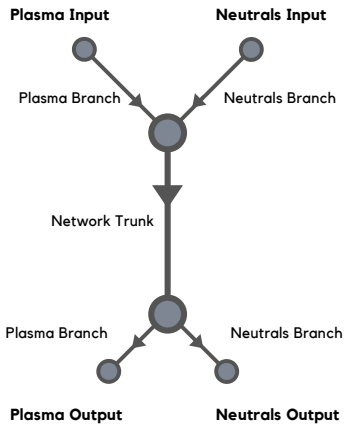


Figure 2: Structure designed to model the plasma and neutral evolution using Fully Convolutional Neural Networks.

In order to account for differing physics from which the plasma and neutral grids emerge from we designed a the network to have bifurcated inputs and outputs as shown in figure 2. At the input, grids characterising the plasma was fed into one branch, while those of the neutrals were fed into the other. Each branch is composed of the same number of layers, performing convolutions and pooling activities on the input grids. The convoluted plasma and neutral grids would be later merged at the end of the branches and fed to the trunk of the network, where the interlinked holistic system, the plasma along with its neutral interactions will be modelled. At the output, the trunk, branches out into two again, one for the plasma grids and the neutral grids. By having branched inputs and outpus, the network becomes capable of modelling both the plasma and neutral behaviour while allowing for information exchange between them.

3. SOLPS

The Scrape Off Layer Plasma Simulator (SOLPS) is a state of the art code framework used to model the edge transport and study the plasma surface interactions.The SOLPS-ITER version which has been used in this research, is the basis of design of the divertor in ITER, as well as in simulating detachment and studying the plasma-wall interactions [8] [9] [10]. Edge modeling in SOLPS is performed by two sub codes within the framework : B2.5 and EIRENE [11] .

Being a multi-fluid transport code, B2.5 solves the behaviour of multi-species plasma along a 2-dimensional geometry. It models the plasma evolution by numerically solving the Braginskii equations in magnetically aligned curvilinear coordinates [12] [7]. Conservation equations of mass, momentum and energy are taken into account within the code package [11]. Modeling the plasma behaviour using B2.5 contributes consiserably to the wall clock time as it implemented as a serial code within the SOLPS framework [13]. EIRENE is a 3-dimensional Monte Carlo transport code used to solve the evolution of neutrals by solving a multi-species set of coupled Boltzmann equations utilising a Bayesian method [14].

As the plasma and neutral evolution are modelled by solving two sets of equations encapsulating different physics principles, both B2.5 and EIRENE are run independently. Between the two code packages information characterising the plasma and neutral state is exchanged after a predetermined number of iterations within the simulation [15] [16].

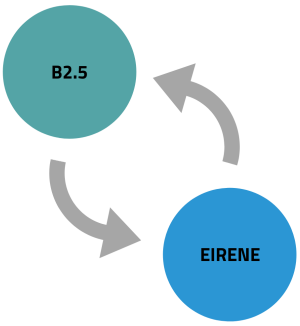


Figure 3: Information regarding Plasma and Neutral states being exchanged between the code.

3.1. Physics Parameters

Within the SOLPS framework, we have identified 7 fundamental physics parameters characterising the plasma and neutral state at any instant in time. The network models

the behaviour of plasma and neutrals by mapping the evolution of these 7 parameters.

(i) Neutral Density
(ii) Ion Density
(iii) Electron Density
(iv) Ion Temperature
(v) Electron Temperature
(vi) Neutral Parallel Velocity
(vii) Ion Parallel Velocity

Each parameter is represented by a profile in discretised grids across the poloidal cross section of the Tokamak focusing only on the edge regions as shown in figure 4. We have chosen 5 grids to characterise the plasma state and 2 grids to characterise the state of the neutrals.

3.2. Grid Representation

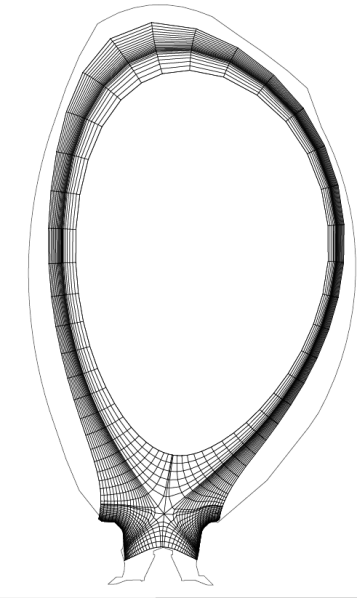
The domain of interest within the SOLPS framework is the outer edges of the poloidal cross-section. For both the neutrals and plasma, the physical domain space under observation is that shown in figure 4. The domain is discretized into 3268 grid cells of varying size, whose compact image is shown in figure 4a. Spatial coverage for each cell is determined so as to ensure that the physics parameters stay constant within each cell [7]. The meshgrid is composed of 4 parts as show in figure 4b, with each part categorising the grid areas into different operational domains within a Tokamak with single null geometry [7].

To perform numerical simulation and model the time evolution of the plasma and the neutrals, the discretised grid shown in figure 4 is mapped onto quadrangular grids. The DivGeo, CARRE, and Triageom packages within the SOLPS framework would rearrange the poloidal meshgrid into a rectangular meshgrid as shown in figure 5 [7].

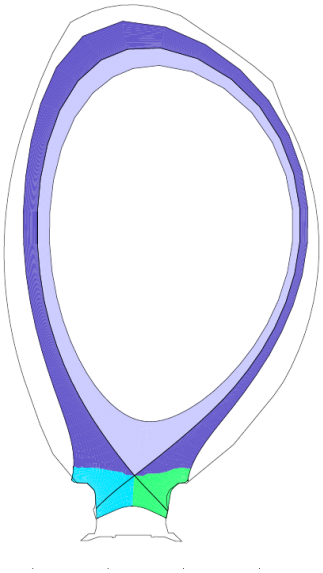
SOLPS rearranges the poloidal meshgrid into a rectangular grid with 38 rows and 86 columns. The transformation ensures that the local relations and dependencies within the poloidal meshgrid are not lost by maintaining the adjacency of the gridcells. These rectangular grids would be fashioned into inputs and outputs for the FCN model. Each of the 7 parameters are represented by 38 x 86 grids as shown in figure 5. Out of the 7 grids, 5 characterises the plasma state and teh remaining 2 characterises the neutral state.

3.3. Case Setup

The SOLPS simulations were performed for a case designed on an actual JET (Joint European Torus) experiment under the shot number 89241 [17]. Our case assumes toroidal



(a) Poloidal meshgrid discretising the domain of interest.



(b) Poloidal domain space split into 4 different operational zones.

Figure 4: Discretisation of the Poloidal Space. In (b), Blue indicates the inner divertor, green the outer divertor, lighter shade of purple shows the core and darker shade of purple represents the scrape-off layer. [7]

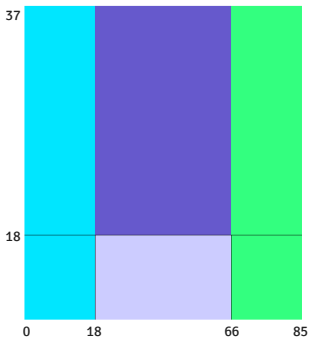


Figure 5: Rectangular grid which forms the computational space for the SOLPS modelling.

symmetry within the Tokamak reducing our focus only onto the poloidal cross-section. Our primary focus was to model the edge plasma and neutral behaviour when a steady state is slightly perturbed. Thus, the case was pre-run until it reaches the steady state regime. The edge in this case was characterised by three entities: Deuterium ion, it's neutral component and the electrons. Toroidal magnetic field was set at 2.2 Tesla. A combination of Neutral Beam Injection (NBI), Ion Cyclotron Resonance Heating (ICRH) and Ohmic heating contributed to the total power provided to the plasma. The decision to restrict ourselves to the steady state regime was taken to ensure that the proof of

concept can be built with minimal data, thus reducing the simulations adn hence the time needed to build the synthetic dataset.

3.4. Data Generation

In order to create a diverse dataset, the predefined case (already in steady state) was run under different physical conditions. We tune three parameters that have significant influence on the plasma and neutral profiles :

- Heating Power - Ohmic heating (MW)
- Puffing Rate - Rate of change of plasma density (s^{-1})
- Pump Intensity - Percentage change of neutral density (%)

The dataset was built from SOLPS in two phases. In Phase 1, the input grids that characterise the plasma and neutral state at the initial point in time are built. Phase 2, the input grids are simulated further to develop the output grids.

3.4.1. ***Phase 1*** The predetermined case setup is simulated from *t=0s* to *t=0.234s* up until the simulation arrives at steady state. The simulation arrives at steady state with the parameters as indicated in table below :

Heating power	4.080 MW
Puffing rate	10^{22} s^{-1}
Pump intensity	94%

The steady state configuration was treated as the input setting for further SOLPS simulations under different combinations of our chosen parameters. The parameters were discretised within a domain range and all possible combinations of those discretrised values were simulated. The discretisation of the parameter range is seen in the table below:

Parameter	Min.	Max.	Iteration	Number of values
Heating power	3.0 MW	8.5 MW	+0.5 MW	12
Puffing rate	10^{18} s^{-1}	10^{22} s^{-1}	x5.0	9
Pump intensity	48%	98%	+5%	11

Total number of combinations = 12 x 9 x 11 = 1188

Each simulation was run for 1000 iterations with a time step of $\Delta t = 10^{-6}$ seconds, progressing the simulation by 1 ms.

3.4.2. ***Phase 2*** The grids generated in Phase 1 are used as inputs to build the output grids. Each of the 1188 simulated cases were run further, but this time under the same set of parameters. The heating power, puffing rate, and pump intensity would be fixed to the values described in the subsequent table and ran for a total of 2000 time iterations. With a time step of $\Delta t = 10^{-6}$ seconds, the output grids is 2 ms ahead of the initial grids, with SOLPS accommodating the changes introduced by the new values.

Heating power	5.000 MW
Puffing rate	10^{19} s^{-1}
Pump intensity	94%

The output of Phase 1 simulations form the input and those of the Phase 2 simulations the output upon which we build our FCN model. This provides us with a labelled dataset upon which we can perform input-output mapping.

4. Network Design and Training

The network was designed to engage in a routine of dimensionality reduction, that can effectively capture the interdependencies shown by the distributions of physical parameters in space [18]. Due to the spatial nature and relationships of the grids we utilised convolutions to perform dimensionality reduction. Convolutions can effectively capture the general trends found across the grid while reducing the complexity of the data [19]. We observed that while performing dimensionality reductions on the grids it was paramount to maintain the aspect ratio of the grid (refer supplementary section 10.2). We had randomly sampled the weights and biases to from a uniform distribution between 0 and 1. To effectively model the complex nonlinearities within the plasma and neutral behaviour it was necessary the to employ a network of sufficient depth [20]. It was observed the performance of the network was optimum at 20 hidden layers with nonlinear capabilities. However, we were discouraged from building phenomenally deep

networks as they displayed tendencies of strongly over-fitting to the training data, and gave us poor generalisation (refer supplementary section 10.1) .

Since the task was to perform image mapping, we had to ensure that the dimensions of the outputs matched with those of the inputs. While we employed dimensionality reduction approaches in the initial levels of the network towards the output side, we had to employ transposed convolutions to regain the required dimensionality. The figure 6 shows the internal layout of our network. We employ 13 convolutional layers, 5 transposed convolutional layers, and 3 pooling layers to perform the modelling. Physical profiles are heavily neighbourhood dependent, and employing convolutional layers over dense layers ensured that this local connectivity could be captured. In total the network consisted of 2,289,639 trainable parameters.

The plasma input branch is fed with 5 rectangular grids of dimension 38 x 86, and the neutrals input consists of 2 rectangular grids of the same dimensions. Forking the inputs and outputs has been found to pick up the more salient features of the physics governing the different behaviour sof plasma and neutrals. The method also expresses higher accuracy when compared to network designs with a single input with the plasma and neutral grids entangled together throughout the network.

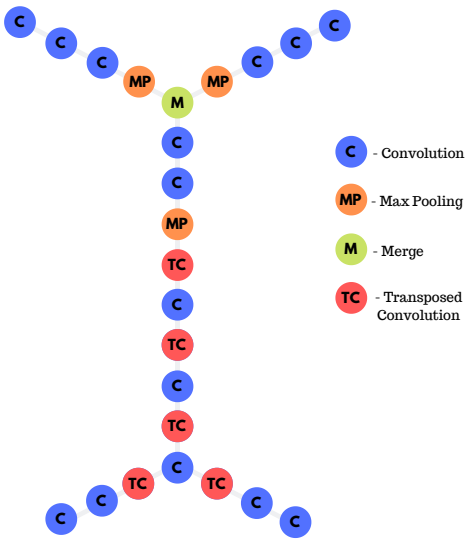


Figure 6: Network structure exposing the internal configuration of the various layers.

We designed, built and tested the network using the Keras API while employing tensorflow at the backend [21] . Out of the 1188 data samples as described in section 3.4, we found 22 of the SOLPS simulations to be corrupt, leaving us with 1166 data samples to train and test the model with. The grids characterising each physical parameter were normalised between 0 and 1 by taking the maximum and minimum values

for each parameter across the entire dataset. The hyperparameter search showed us that the network performed comfortably optimal with the Adadelta optimizer and with mean squared error as the loss function. 20 percent of the entire data samples were sampled randomly and taken to be the test set leaving the rest to be split up for the training and validation. Trained over 100 iterations with a batch size of 10 samples, we observed a mean squared error of 9.2E-05 across the test dataset.

5. Results

5.1. PCA

To understand the diversity within the dataset and its impact on the trained network, we employed a principal component analysis on the entire dataset. By utilising a SVD approach we reduced the dimensionality of the dataset to two components. The principal components of the difference between the final and initial states were taken to characterise the variation within the dataset. The analysis showed that the majority of the dataset is clustered onto the bottom left quadrant as shown in figure 7. Both the test and train datasets were randomly sampled across the dataset making it inclusive of the points taken from all the quadrants. To demonstrate the ability of our FCN, the results we have shown are for a point sampled from the cluster in the top right quadrant.

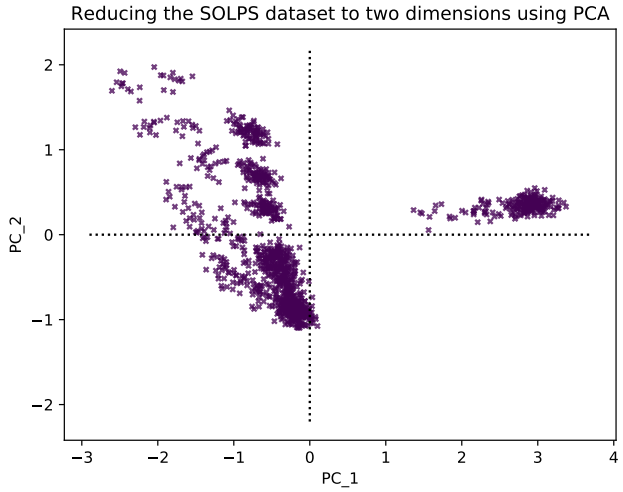
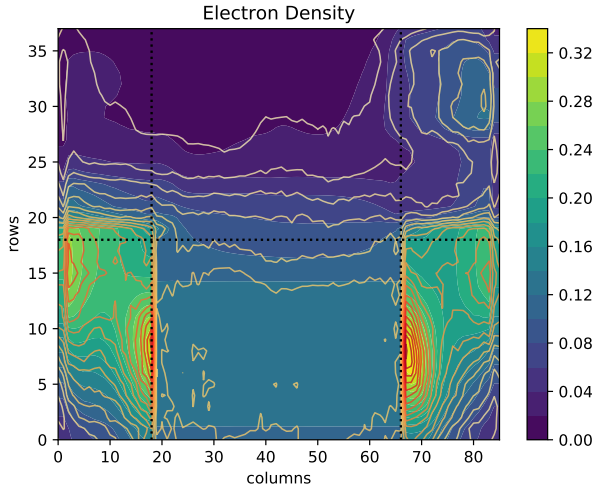


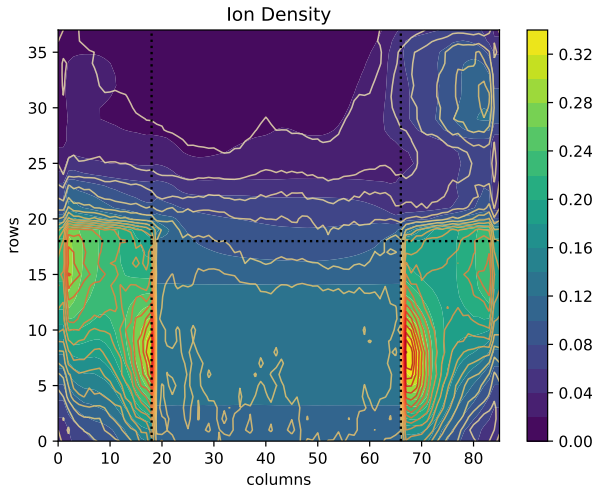
Figure 7: Principal Component Analysis on the entire dataset, reducing and representing the dataset into two dimensions.

5.2. Output Comparison

The effectiveness of our FCN approach is best understood visually. To this end, we have over-imposed contour maps of the neural network outputs on top of the actual outputs (obtained through SOLPS). The SOLPS outputs are represented in blue and green filled plots while the FCN solutions are outlined with the contour lines ranging from yellow to red.

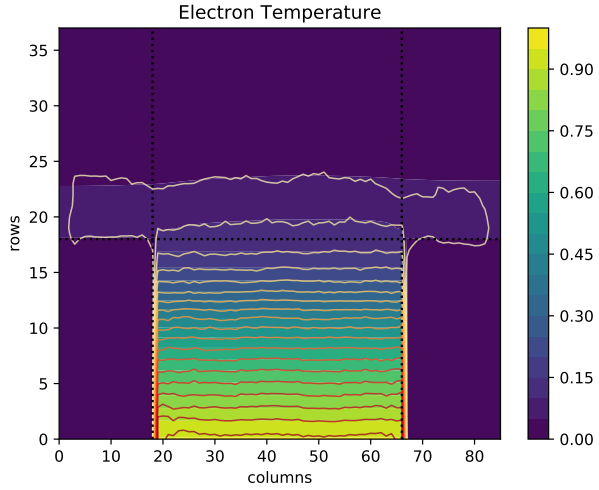


(a) Electron Density

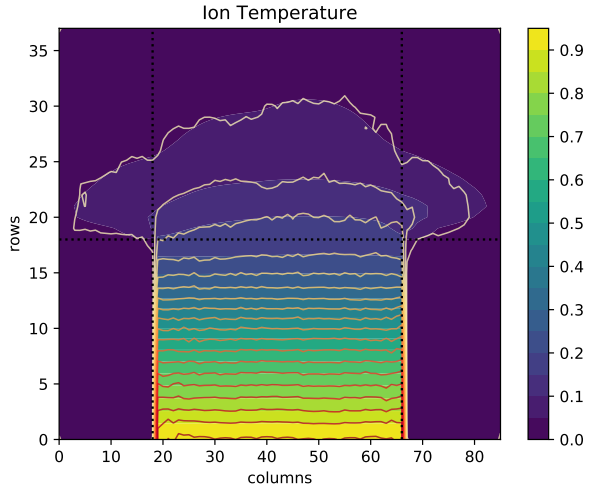


(b) Ion Density

Figure 8: Density Profiles across the computational grids. Figure 8a represents the electron density while figure 8b represents that of the ions. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the Tokamak as described in section 3.2.

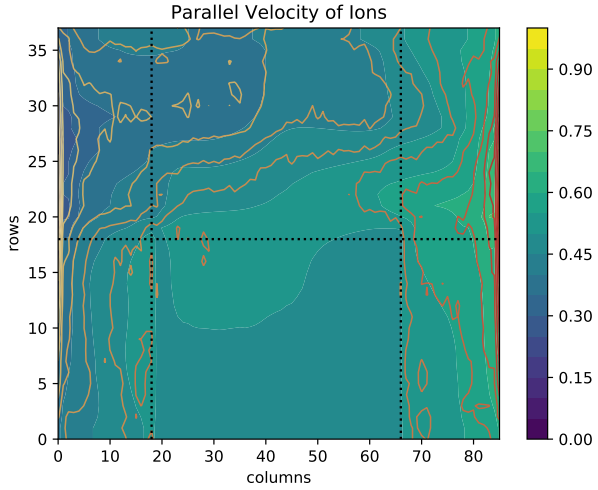


(a) Electron Temperature

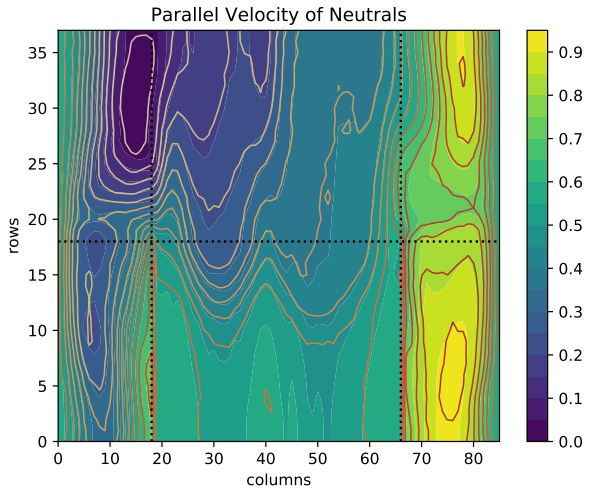


(b) Ion Temperature

Figure 9: Temperature Profiles across the computational grids. Figure 9a represents the electron temperature while figure 9b represents that of the ions. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the Tokamak as described in section 3.2.



(a) Prallel Velocity of Ions



(b) Parallel Velocity of neutrals

Figure 10: Parallel Velocity distribution across the computational grids. Figure 10a represents distribution for the ions while figure 10b represents that of the neutrals. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the Tokamak as described in section 3.2.

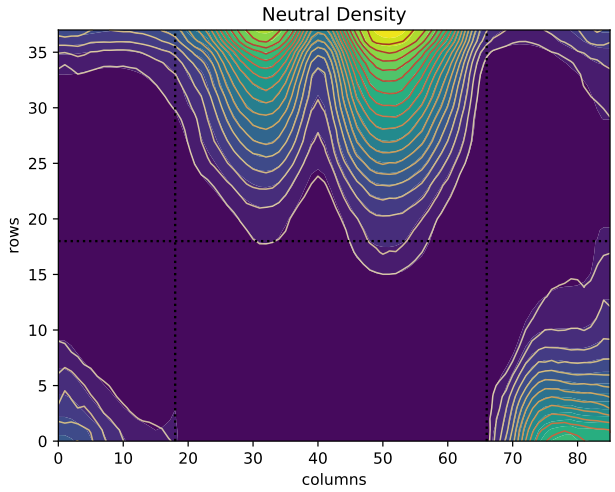


Figure 11: Density Profiles of the neutrals across the computational grid. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the Tokamak as described in section 3.2.

Glancing across from figures 8 to 11, it can be seen that there are good correlations and a good fit between the SOLPS solutions and the FCN output. There are some discrepancies especially seen in the figures 8a, 8b and 10a. This is attributed to the larger variation of these parameters within the training dataset leading to the intentional build up of under-fits to ensure generalisation.

5.3. Time Gain

SOLPS simulations were implemented on the Marconi cluster. Each simulation of SOLPS was run on 2 x 18 cores of Intel Xeon E5-2697 with each core clocking 2.3 GHz [22]. On the Marconi framework each cluster that advanced the simulation by 2 milliseconds took somewhere between 2 and 3 hours (depending on the case). In contrast, the FCN took 117 minutes to train on a CPU arrangement of 16 cores of Intel Xeon E5-2640 v3 clocking 2.6 GHz, on a considerably less powerful machine. However, once trained the FCN could arrive at the solution in under 0.033 seconds on the same framework. Thus, our novel FCN approach reduces the time required to model the edge plasma and neutral evolution by more than 5 orders of magnitude while not compromising vastly on the accuracy of the output.

5.4. Limitations

Neural networks when employed for regression purposes, often comes with an issue of determinacy, mainly due to the nature of its approximateness. The higher the di-

mensionality of the output, the larger this issue grows. This sense of indeterminacy is inducted into the model to avoid over-fitting and produce model robustability [23]. Thus, the issue of determinacy is multiplied by the dimensionality of the output, and it is multiplied by 3268 in this case. However, what we lose in precision for our model is made up by generalisation. Our models shows significant capability to approximate the solution and get us in the neighbourhood of the accurate output. The other limitation our approach faces is that in order to build and employ our model, we are still dependent on data generated using the SOLPS framework. Our trained model will only work within the confines of the case defined in section 3.3. However guaranteed that we have a more diverse and comprehensive database, this approach could be used to build an approximate solver for the edge physics. While developing this approach, we could not ignore the irony that to avoid the time lag that SOLPS brings forth by using our FCN, we have to first build the data and enter into this time sink, that we are all the while trying to avoid.

6. Conclusion

This work demonstrates a successful application of neural networks in a case of complex nonlinear phenomena. We show that an optimized configuration of neural networks can efficiently map the evolution of the entire plasma at the edge of a tokamak at all points of the simulation grid. Mapping at this scale for a strongly nonlinear system is performed for the first time to the best of our knowledge.

7. Future Work

The approximation capabilities of our FCN based approach can be utilised for more complex multi-species plasmas. it may be used in novel numerical methods such as the Parareal Algorithm. The Parareal algorithm employs a time parallelisation approach to better the SOLPS convergence time [15]. We hope that our FCN approach can solve the majority of the evolution within a fraction of the time with a better prediction of the final result as compared to traditional methods.

8. Acknowledgement

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. The research was also funded by the FuseNet Association which is gratefully acknowledged.

9. References

[1] S. I. Krasheninnikov, A. Yu. Pigarov, D. A. Knoll, B. LaBombard, B. Lipschultz, D. J. Sigmar, T. K. Soboleva, J. L. Terry, and F. Wising. Plasma recombination and molecular effects in tokamak divertors and divertor simulators. *Physics of Plasmas*, 4(5):1638–1646, 1997.

[2] Gianpiero Colonna. Boltzmann and vlasov equations in plasma physics. In *Plasma Modeling*, 2053-2563, pages 1–1 to 1–23. IOP Publishing, 2016.

[3] G Giruzzi, J Garcia, J F Artaud, V Basiuk, J Decker, F Imbeaux, Y Peysson, and M Schneider. Advances on modelling of ITER scenarios: physics and computational challenges. *Plasma Physics and Controlled Fusion*, 53(12):124010, nov 2011.

[4] S. Wiesen, D. Reiter, V. Kotov, M. Baelmans, W. Dekeyser, A.S. Kukushkin, S.W. Lisgo, R.A. Pitts, V. Rozhansky, G. Saibene, I. Veselova, and S. Voskoboynikov. The new solps-iter code package. *Journal of Nuclear Materials*, 463:480 – 484, 2015. PLASMA-SURFACE INTERACTIONS 21.

[5] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[6] Jean-Gabriel Attali and Gilles Pagès. Approximations of functions by a multilayer perceptron: a new approach. *Neural Networks*, 10(6):1069 – 1081, 1997.

[7] M. Stanojevic X. Bonnin D.P Coster, A. Kukushkin et al. Solps manual. http://solps-mdsplus.aug.ipp.mpg.de:8080/solps/Documentation/solps.pdf, 2015.

[8] A.S. Kukushkin, H.D. Pacher, V. Kotov, G.W. Pacher, and D. Reiter. Finalizing the iter divertor design: The key role of solps modeling. *Fusion Engineering and Design*, 86(12):2865 – 2873, 2011.

[9] D.P. Coster. Detachment physics in solps simulations. *Journal of Nuclear Materials*, 415(1, Supplement):S545 – S548, 2011. Proceedings of the 19th International Conference on Plasma-Surface Interactions in Controlled Fusion.

[10] S Wiesen, M Groth, S Brezinsek, and M Wischmeier and. Modelling of plasma-edge and plasma–wall interaction physics at JET with the metallic first-wall. *Physica Scripta*, T167:014078, feb 2016.

[11] R. Schneider, X. Bonnin, K. Borrass, D. P. Coster, H. Kastelewicz, D. Reiter, V. A. Rozhansky, and B. J. Braams. Plasma edge physics with b2-eirene. *Contributions to Plasma Physics*, 46(1-2):3–191.

[12] S. I. Braginskii. Transport Processes in a Plasma. *Reviews of Plasma Physics*, 1:205, 1965.

[13] S. Voskoboynikov V. Rozhansky D. Reiter S. Wiesen V. Kotov P. Börner W. Dekeyser, M. Baelmans. B2-b2.5 code benchmarking.

[14] Detlev Reiter, Martine Baelmans, and P Börner. The eirene and b2-eirene codes. *Fusion Science and Technology*, 47, 02 2005.

[15] D. Samaddar, D.P. Coster, X. Bonnin, C. Bergmeister, E. Havlíčková, L.A. Berry, W.R. Elwasif, and D.B. Batchelor. Temporal parallelization of edge plasma simulations using the parareal algorithm and the solps code. *Computer Physics Communications*, 221:19 – 27, 2017.

[16] D Coster, A Chankin, and C. Konz. Solps modelling: inputs and outputs, and the connection between. Max Planck Institute for Plasma Physics, EURATOM Association, Garching, Germany., 2005.

[17] Paul-Henri Rebut. The joint european torus (jet). *The European Physical Journal H*, 43(4):459–497, Dec 2018.

[18] Jarkko Venna and Samuel Kaski. Nonlinear dimensionality reduction as information retrieval. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 572–579, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[19] Francesco Visin Vincent Dumoulin. A guide to convolution arithmetic for deep learning. *stat.ML*, 2018.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] François Chollet et al. Keras. https://keras.io, 2015.

[22] F. Iannone, G. Bracco, C. Cavazzoni, R. Coelho, D. Coster, O. Hoenen, A. Maslennikov, S. Migliori, M. Owsiak, A. Quintiliani, B. Palak, V. Pais, F. Robin, E. Rossi, and I. Voitsekhovitch. Marconi-fusion: The new high performance computing facility for european nuclear fusion modelling. *Fusion Engineering and Design*, 129:354 – 358, 4 2018.

[23] S. Lawrence and C. L. Giles. Overfitting and neural networks: conjugate gradient and backpropagation. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 1, pages 114–119 vol.1, July 2000.

[24] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[25] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015.

[26] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[27] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[28] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117, 2015.

Supplementary Section

To optimally build and train a network according to our approach defined in section 2, we conducted an extensive hyper-parameter search. The model was trained under different neural network architectures, using various optimisers. We also experimented with different kinds of nonlinear activation functions to see which function could capture the essence of the nonlinearity expressed by our physics parameters. We observed that the best fitting of contour plots were observed while using the the optimiser 'Adadelta' [24]. To build a robust model, we experimented with different kinds of regularisation strategies, and observed that the introduction of a few layers of max pooling to have the most consistent and accurate response. While attempting to model using different arbitrary nonlinearities found within the Keras framework, the function 'ReLU' gave the most favourable fit and was chosen to impose nonlinearity at every layer except the last [25]. Since we chose to normalise the values between zero and unity, the output layers were equipped with a Sigmoid Function [26]. This allowed the output values to be within the domain range, allowing us to perform mapping via regression.

10. Model Architecture Variations

Image mapping of the physics parameters requires the dimensions of the output to match with that of the input. Since we employ a dimensionality reduction approach to fragment the input grids and extract information from them, the reduced grids will have to be built back up to the original dimensions. This dimensionality constraint served as the north star while configuring the layers of the network. As the information was passed across each layer the information was compressed onto smaller grids. The output dimensions of a convolutional layer can be given as :

$$O_c = \frac{I_c - K}{S} + 1 \tag{1}$$

where,
O_c - Output Dimensions, I - Input Dimensions , K - Filter Size, S - Stride .

The initial convolutional layers serves to capture the general characteristics and behaviours across the grid [27]. As the convolutional operations were conducted on profiles with spatial dependencies, the initial layers were deployed with larger filter sizes. The reduced grids produced after sequential convolutions were brought back to the original dimensions through a series of transposed convolution and convolution sets. Transposed

Convolutions would perform the exact opposite processing brought about by a convolution, albeit adding a layer of nonlinearity to the data [19]. The output dimensions of a transposed convolutional layer is formed by rearranging equation 2 :

$$O_{TC} = (I_{TC} - 1)S + K \tag{2}$$

where,
O_{TC} - Output Dimensions, I_{TC} - Input Dimensions , K - Filter Size, S - Stride .

By following up every transposed convolutional layer with a convolutional layer, we observed better control over the blow up of the errors introduced by the dimensionality reduction and subsequent expansion. In an attempt to minimise the information loss brought about by the dimensionality reduction, we chose to avoid padding the values altogether.

10.1. Depth Variations

Keeping in line with the architecture as shown in the figure 2, we built networks of varying depths to optimise the FCN with the nonlinearity that would best model the nonlinear plasma and neutral evolution. We observed that to capture the profiles we required a network of sufficient depth [28]. Having too few layers led to a poorer, less accurate fit. However having an exceeding number of layers led to over-fitting of the model, moving further away from generalisation. We measured the performance (Average Mean Squared Error) against models with different depths having varying number of trainable parameters and found the most accurate fit with 20 layers (including the branched layers). While counting the layers, we have only taken into account those layers that adds a component of nonlinearity to the network. Each of the models we trained were run for 100 epochs over the entire training dataset with a batch size of 10.

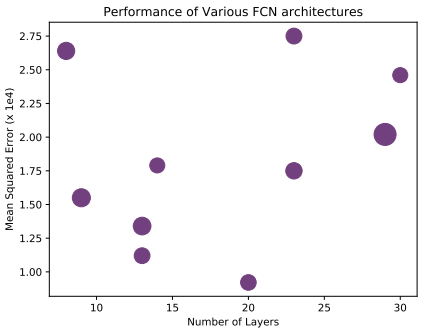


Figure 12: Scatter Plot measuring the impact of the depth of the FCN on the performance. The number of trainable parameters is indicated by marker size.

10.2. Convolutional Approach

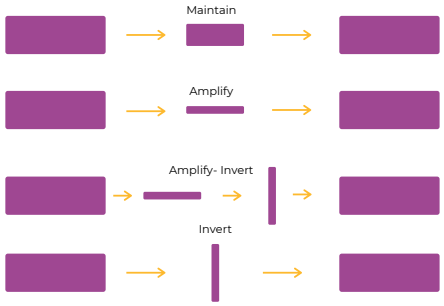


Figure 13: Various Convolutional Strategies. 'Maintain' ensures that the aspect ratio is held up across the network, 'Amplify' will exaggerate the aspect ratio, 'Invert' will flip the dimensional dominance and 'Amplify-Invert' will initially amplify the aspect ratio and then invert it before reverting to the original dimensions at the output.

While performing dimensionality reduction of the input grids via convolutions, we have the inherent freedom to choose the extent to which we reduce the horizontal and vertical dimensions of the grids. Since these grids are not just a collection of pixels but represent physics parameters scattered across a physical space, it was paramount to ensure that the convolutions will take into account the extent of the spatial dependencies of the physical parameter point in the poloidal space restructured into the rectangular grids. We experimented with convolutional approaches that would attempt tp amplify, invert and a combination of the both approaches onto the 38:86 aspect ratio of the input grids (as shown in figure 13). We noticed that the model gained higher performance and fit to the test dataset when we maintained the aspect ratio. By doing so we believe we are ensuring the spatial relations of the profiles across the grids. In figure 14, we can notice how much dependence the convolutional dimensional reduction approaches had on the performance of the model.

11. Chosen Optimum Model

Across the hyper-parameter search that we had conducted we observed the lowest accuracy with a FCN having a total of 20 hidden nonlinear layers. In the internal layout of the chosen model can be seeen in figure 15. The plasma and the neutral grids fed in at their respective input points would undergo through three convolutional layers each, before being concatenated. The concatenated full physics model undergoes through 8 layers of convolutions and transposed convolutions with data interchanged between the reduced plasma and neutral grids. Closer to the FCN output, the network bifurcates to separate the plasma and neutral grids. Both the plasma and the neutral components would pass through three layers of nonlinearities before forming the output. To bring

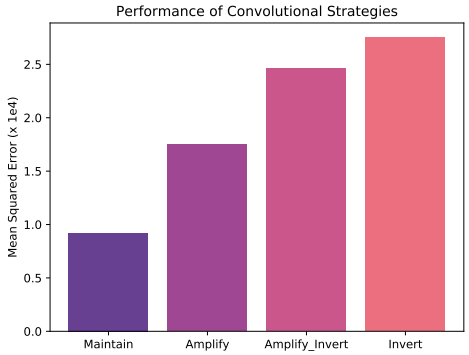


Figure 14: Performance of the various convolutional approaches measured as the mean squared error across the test dataset.

about some generalisation into the model, we have introduced Max Pooling layers at three points in the network. Figure 15 gives clear indication as to how we have employed convolutions to reduce the dimensions of the profiles, while ensuring that the aspect ratio of the initial grids are not skewed with much.

12. Discussion

It should be noted that our chosen model is not necessarily the optimum model to serve as an approximate solver for the edge physics in a Tokamak, but simply the best we observed in our hyper-parameter search. Our paper serves to highlight the impact of our approach of using an FCN with bifurcated inputs and outputs, in modelling the edge plasma and neutral behaviour.

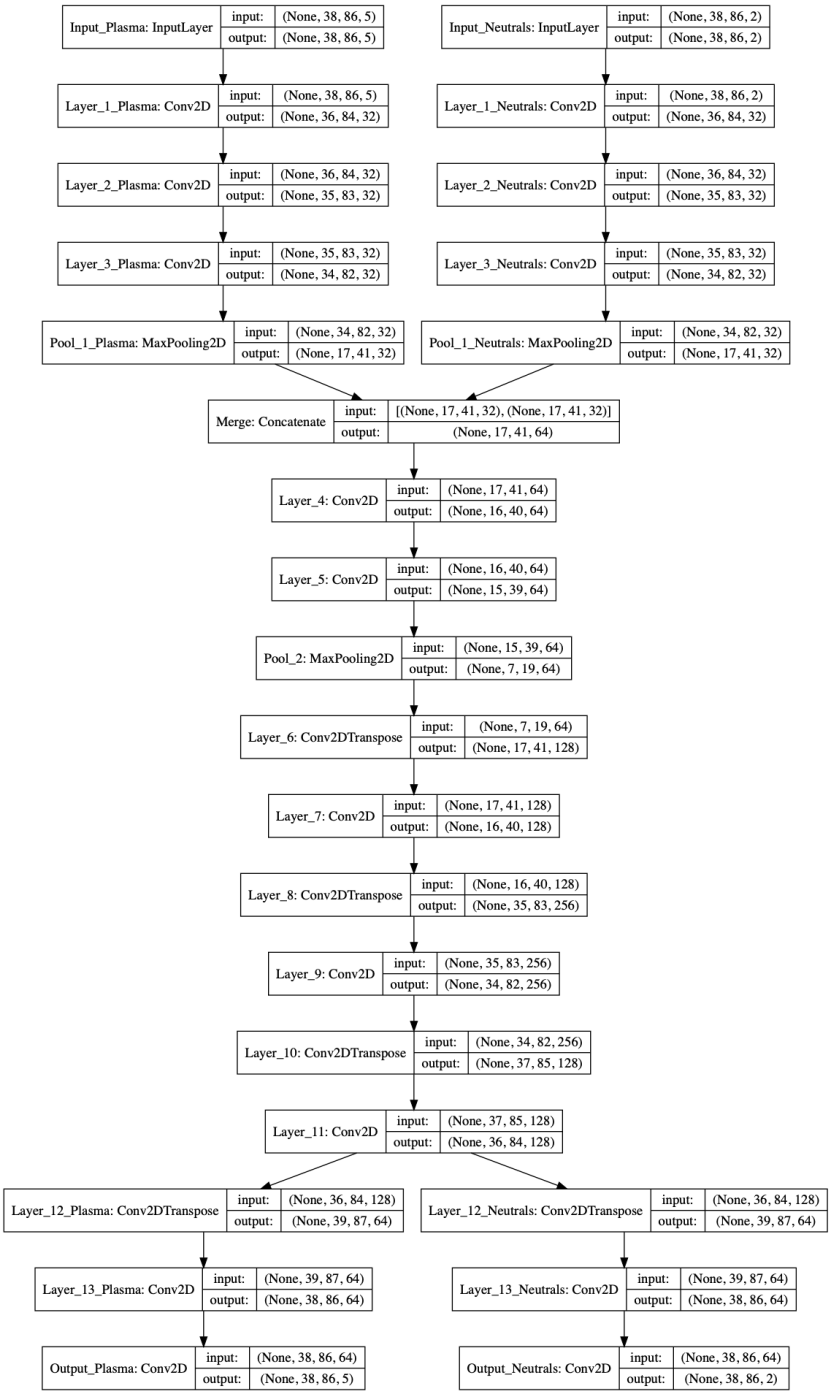


Figure 15: Internal layout of the optimum FCN model that we have chosen. 'Conv2D' represents the two-dimensional convolution operation, while the 'Conv2DTranspose' represents the two-dimensional transposed convolution operation.