Real-time capable turbulent transport modelling using the 10D QuaLiKiz Neural Network

K.L. van de Plassche[1], J. Citrin[1], C. Bourdelle[2], V. Dagnelie[1,3], F. Felici[4], A. Ho[1]

[1] *DIFFER - Dutch Institute for Fundamental Energy Research, Eindhoven, the Netherlands*

[2] *CEA, IRFM, F-13108, Saint-Paul-lez-Durance, France.*

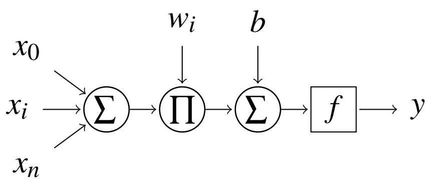
[3] *Utrecht University, Princetonplein 5, 3584 CC, Utrecht, The Netherlands.*

[4] *EPFL-SPC, CH-1015, Lausanne, Switzerland.*

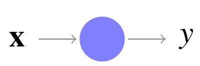
Particle and heat transport in the core of the tokamak is dominated by turbulent transport, and can be accurately described by gyrokinetics. Unfortunately, making predictions with high-fidelity nonlinear gyrokinetic codes is computationally extremely expensive, in the order of 10^6 CPUs per radial point per time step. Fortunately quasilinear gyrokinetic models, have been very successful in predicting particle and heat transport in tokamaks, successfully reproducing experimental profiles in many cases. As they are reduced models, they are typically many orders of magnitude faster to run. One such code is QuaLiKiz, which has been validated against JET, ASDEX-U and Tore-Supra profiles [1, 2, 3, 4]. While an impressive six orders of magnitude faster than high-fidelity gyrokinetic models, it is still too slow for efficient scenario optimization and realtime applications, demanding around $\sim$ 10 hours for 1s of JET evolution. Neural networks can be used to create a surrogate model, which can then be evaluated within a few microseconds while still accurately reproducing the results of the underlying model. As such, neural networks are used in this work to sketch a pathway to real-time capable turbulent transport modelling.

Neural Networks

Neural networks are nonlinear mappings, acting as universal approximators. The basic building block of a neural network is the neuron; an activation function f with a linear combination of weights w and biases b, whose values are optimized or 'trained' to match a desired input-output mapping.

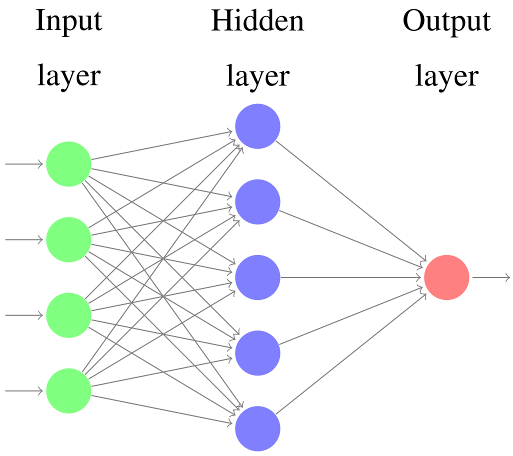


Block diagram of a single neuron



Schematic representation of a single neuron

In this work we use fully-connected *Feed Forward Neural Networks*. The neurons are organized in in layers, the output of the neuron of each layer connected to the input of the neurons in the next. As such, FFNNs are a simple matrix equation, albeit a non-linear one. They also has an analytical derivative, up to the order of which the activation function can be differentiated. These two features are essential for integration into rapid transport codes.



A fully-connected Feed Forward Neural Network. Shown is a network with one hidden layer, four input dimensions and a single output dimension. In principle a network can have many layers with a larger amount of neurons, as well as more in- and output dimensions.

QLKNN + RAPTOR transport code

A proof-of-concept feed-forward neural network regression of QuaLiKiz 'QLKNN-4Dadi' was trained[5] and an extended version 'QLKNN-4Dkin' was integrated in the fast transport modelling code RAPTOR [6]. The ensemble was shown to quite accurately reproduce H-mode JET shot #73342 [6]. To further improve predictions for a wider amount of shots, we seek a major extension of the proof-of-principle from 4D to 10D. A large database of 3.10^8 flux calculations over a 9D input space generated with the QuaLiKiz code is used to extend the original input space of ion temperature gradient R/L_{T_i}, ion-electron temperature ratio T_i/T_e, safety factor q and magnetic shear $\hat{s}$ with the electron temperature gradient R/L_{T_e}, density gradient R/L_n, normalized minor radius r/R, collisionality ν^* and effective ion charge Z_{eff}. The input ranges were chosen as a non-equidistant grid, focussing on experimentally relevant regimes. This database was then used to train neural networks, using a methodology aimed at ensuring consistency with known physical constraints, which was found essential to reproduce plasma profiles, especially when simulating multiple time-steps. Finally, a 10^{th} dimension, ExB shear, is added in post-processing using a new turbulence quenching rule [7].

Physically consistent neural network training

The used neural network training methodology used in this work seeks to match the following features of the underlying model:

1. Sharp instability thresholds
2. No spurious positive flux in stable region
3. Matching thresholds for all transport channels

The behaviour of the resulting neural network strongly depends on the cost function used in the gradient descend optimization during training. In general the cost function C has two terms: a term for the goodness-of-fit C_{good} and a term to regularize the regression C_{regu}. We expand this function with a term to strongly punish spurious flux in the stable zone C_{stab}.

$$C = C_{good} + C_{regu} + C_{stab}$$

1. and 2. are enforced by punishing goodness-of-fit only for the points in the unstable region. Naturally, C_{stab} is only non-zero for points in the stable region. Mean Square Error is used as the goodness-of-fit metric, and L2-norm and early stopping are used to regularize the network. This can be summarized as follows:

$$C_{good} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(QLK_i - NN_i)^2, & \text{if } QLK_i \neq 0 \\ 0, & \text{if } QLK_i = 0 \end{cases}$$

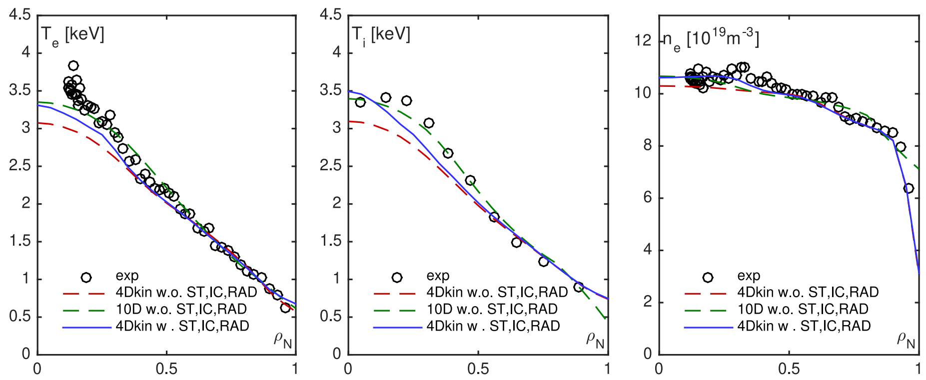
$$C_{regu} = \lambda_{L2}\sum_{i=1}^{k} w_i^2$$

$$C_{stab} = \begin{cases} 0, & \text{if } QLK_i \neq 0 \\ \frac{\lambda_{stab}}{n}\sum_{i=1}^{n} NN_i - c_{stab}, & \text{if } QLK_i = 0 \end{cases}$$

For all n QuaLiKiz predictions QLK and the corresponding neural network prediction NN for each point in the database. The training hyperparameters regularization scale λ_{L2}, stability punish scale λ_{stab}, stability punish threshold c_{stab} are then optimized with a simple hypergrid scan.

Application in RAPTOR

The trained networks have been embedded in the RAPTOR transport code [8]. A comparison between QLKNN-4Dkin and QLKNN-10D on JET shot #73342 can be found in the following figure:



QLKNN-10D and RAPTOR are able to reproduce the steady state flat-top phase of H-mode JET shot #73342. For this shot, QLKNN-10D clearly performs better than QLKNN-4Dkin.

In the 4-channel simulation the transport equations for ψ, T_e, T_i, and n_i were solved simultaneously. Boundary condition of kinetic profiles were prescribed at $\rho = 0.85$. The source profiles and equilibria were externally prescribed. It should be noted that for this version of QLKNN-10D $C_{stab} = 0$ and ν^* is clipped to a maximum of 0.05. Improving performance of QLKNN-10D for higher collisionality is currently under investigation.

In this work we have presented methodology of training and validating these neural networks, and shown the validity of using these networks in a transport code. The speed of the networks combined with RAPTOR allow for transport simulations at a speed that is unprecedented, and opens new avenues in the modelling of fusion experiments.

Acknowledgements:

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 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231

References

[1] J. Citrin *et al.* Plasma Physics and Controlled Fusion **59** 12400 (2017)
[2] C. Bourdelle *et al.*, Plasma Physics and Controlled Fusion **58**, 1 (2016)
[3] C. Bourdelle *et al.*, Nuclear Fusion (accepted, 2018)
[4] M. Marin *et al.*, this conference O2.102 (EPS Prague 2018)
[5] J. Citrin *et al.* Nuclear Fusion **55** 092001 (2015)
[6] F. Felici *et al.*, Nuclear Fusion (accepted, 2018)
[7] V.I. Dagnelie *et al.* University of Utrecht MSc thesis (2017)
[8] F. Felici *et al.*, this conference P4.1091 (EPS Prague 2018)