Hands-on Exercise: Machine Learning for Power Systems

Prepared by Andreas Venzke, Jochen Stiasny, and Spyros Chatzivasileiadis

February 10, 2020

1 Introduction

This assignment has two goals. First, to implement a Decision Tree *and* a Neural Network that can assess the security of a power system, i.e. determine if any given operating point is safe or unsafe. Second, to get familiar with physics-informed neural networks.

The objectives to be achieved at the end of the assignment are the following:

- Decision Trees and Neural Networks for Power System Security Assessment
  - Get familiar with the database
  - Implement a Decision Tree
  - Implement a Neural Network
  - Investigate the impact of different properties of Neural Networks on the classification accuracy
  - Compare the accuracy of the Decision Tree vs. the accuracy of the Neural Network

- Physics-Informed Neural Networks
  - Get familiar with the code
  - Identify strengths and opportunities for improvements (aka shortcomings :)) of the physics-informed neural networks for continuous time

2 Preparation

Make sure you have all the required modules installed

1. scikit-learn
2. tensorflow
3. matplotlib
4. pyDOE
5. Examples of commands:
   - if you run Python on Mac: `pip install PackageName` in a terminal
   - if you are on Windows with Anaconda, then `conda install PackageName` in Anaconda Prompt with "Run as an Administrator" (*not* the Windows command prompt)

3 Tasks

Note: All files and code can be downloaded in a single file from `https://www.chatziva.com/downloads.html`

3.1 Decision Trees (DT) and Neural Networks (NN) for Power System Security Assessment

1. Download the training database

Database prepared by Florian Thams, Andreas Venzke, and Lejla Halilbasic. See the end of the document for more information about the database.

2. Download the python .py file `DT_and_NN_Security_Assessment_14bus.py`

3. Go through the code, try to understand it, and add a comment to every command to explain its function (you can also run it, this should help your understanding).

4. Train a decision tree and a neural network and evaluate their accuracy.

5. Is the database balanced? Is accuracy a sufficient metric? If not, evaluate the different performance metrics, e.g. Matthews correlation coefficient.

6. Experiment with different neural network topologies and activation functions (e.g. ReLU, sigmoid, etc); Report your findings on how this impacts the accuracy metrics.

7. Compare the performance metrics for the decision tree and the neural network you trained. Which has the best performance? What are the advantages and disadvantages of each method?

Optional tasks:

8. Experiment with the different datasets, after you have trained your first decision tree and neural network for only N-1 security and for both N-1 security and small signal stability).

9. The decision tree and the neural network are bound to have some classification errors. Implement a N-1 AC security assessment to validate parts of the database and check if there are any misclassifications.

3.2 Physics-Informed Neural Networks (PINN)

1. Download the python .py file `PINN_inference_swing_equation.py`

2. Go through the code, try to understand it, and add a comment to every command to explain its function (you can also run it, this should help your understanding). The following references might (or might not) be helpful:

- G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. 2019. `https://arxiv.org/pdf/1911.03737.pdf`
- M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-Informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", Journal of Computational Physics, vol.378, pp. 686-707, 2019 `https://arxiv.org/abs/1711.10561`

3. Set the number of collocation points to Nf=100. How is the PINN performance and the computation time?

4. Set the number of collocation points to Nf=1000. How it the PINN performance compared to the previous task?

5. Optional: If you have a powerful laptop, set the number of collocation points to Nf=10'000 (this will take some time, maybe 1 hour). How it the PINN performance now?

3.3 General Questions

1. The DT and NN in Section 3.1 perform a different function from the NN in Section 3.2. What is their main difference? (hint: think about the outputs)

2. What was the size of the *external* training data you needed for the training of the Decision Trees and the Neural Networks in Section 3.1 and what was the size of these external data in in Section 3.2? In general, would you expect that a classification or a regression neural network would require more training data?

3. What are your conclusions about the impact of a balanced training database on the DT and NN training?

4. What are the benefits and the shortcomings of Physics-Informed Neural Networks? Where should future research focus, in order to remove the barriers for real-life application of these methods?

4 Database

You will receive two datasets that were prepared for the IEEE 14-bus system. You are welcome to use these datasets for your future studies, your work or your research. If you use them please cite the following paper:

> F. Thams, A. Venzke, R. Eriksson and S. Chatzivasileiadis, "Efficient Database Generation for Data-Driven Security Assessment of Power Systems," in *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 30-41, Jan. 2020. doi: 10.1109/TPWRS.2018.2890769
>
> and/or
>
> L. Halilbasic, F. Thams, A. Venzke, S. Chatzivasileiadis, P. Pinson. Data-driven Security-Constrained AC-OPF for Operations and Markets. In 20th Power Systems Computation Conference, Dublin, Ireland, pages 1-7, June 2018.

4.1 Difference between the two datasets

Database OPF with VG: Q-limits are not enforced. For this dataset, we run the standard power flow algorithm (provided by `Matpower`), which does not check if the PV buses violate their reactive power limits.

Database OPF without VG: Q-limits are enforced. For this dataset, we run a power flow algorithm (again provided by `Matpower`), which checks if the PV buses violate their reactive power limits. If a PV bus (which is usually a generator bus) injects reactive power that exceeds the Q-limits of the generator, the PV-bus is transformed to a PQ-bus, with $Q = Q_{\text{limit}}$, and the voltage is allowed to vary. This is a more realistic implementation of the power flow, as in reality, if the determined reactive power cannot be provided, then the voltage will necessarily change.

4.2 Contents of each database

Database OPF with VG: 49'615 points, Q-limits not enforced

Database OPF without VG: 675'367 points, Q-limits enforced

For every database, we assessed each operating point for both N-1 security and small signal stability. For more information on the method, please see [1].

N-1 security: We run a power flow considering the base case and the single outage of each component. If any of the power flow cases violate component limits (line flow limits or voltage limits) the setpoint is classified as N-1 insecure. Considered contingencies include all line outages (except for lines 7-8 and 6-13 that make the problem infeasible, i.e. 14-bus system is not N-1 secure for these outages).

Small-signal stability: We consider a full dynamic model for each generator (6th-order), including governor, Automatic Voltage Regulator (AVR type I, 3-states), and Power System Stabilizer (PSS). We set the stability limit at 3% damping ratio. All operating points with a damping ratio below 3% are considered insecure. For more info and the data assumed, please see: [1].

References

[1] F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 30–41, Jan 2020.