

Fast Power system security analysis with Guided Dropout

Donnot et al.
Supplemental material

1 Appendix 1: neural networks architectures

For the generated data, neural networks have the following architecture.

First, we encode the input using fully connected neural network. These networks counts as 2 hidden layer of 200 units. Then each input decoder unit is connected to 372 units. We add a 372 units layers. The transition between these two 372 units is where the guided dropout will be apply, if any. Then, these 372 units are connected to two 200 units layer, to which we add another n^1 units layers which constitute the outputs of the network. The schematic architecture for each of the model is represented in the figure 1.

These configurations counts a wide number of meta parameters. We calibrated the encoding size of "200 units" such that the "one hot" model performs well on the "n-1 validation dataset". The intermediate layer of 372 units have been chosen such that, for the 118 nodes powergrid (counting 186 power lines) there were at least one conditional unit per lines (so at least 186 units) and as many standard unit as conditional unit (making it at least 186 regular units). To sum it up, this made 372 units.

To be exhaustive, we assign 186 units among these 372 to be conditional for the guided dropout model in the 118 nodes power grid, making one conditional unit per power lines. For the 30 nodes powergrid (having 41 power lines), we assign 5 conditional unit per power lines: this models count then $5 \times 41 = 205$ conditional units.

After settings, all these meta parameter, we further make sure that the "one var", "one hot" and "guided dropout" models did not overfit their training set, using the "n-1 validation dataset". There was not overfitting, as we see on the figures 2 in this supplementary materials and in figure 2 of the main paper. So we decided not to explore other neural network configurations.

¹Recall that "n" is the number of power lines in the power grid.

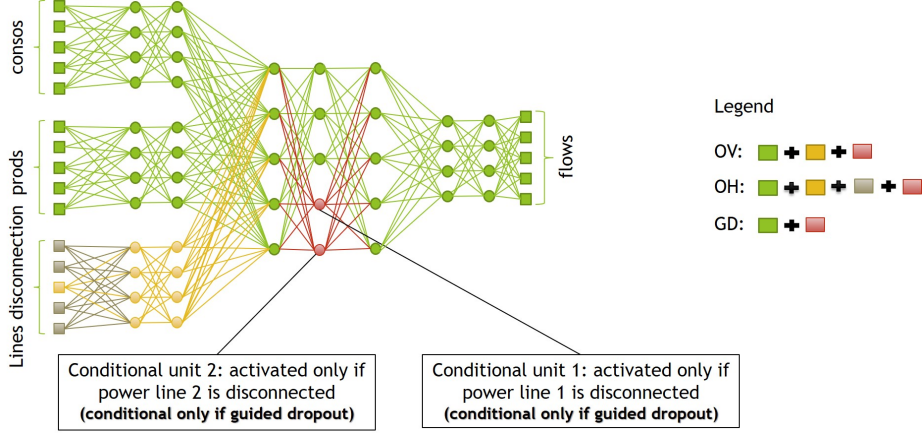


Fig. 1: Schematic architecture of the neural network used for these experiments. Lines denotes trainable parameters. Bias are not represented.

The same architecture (in term of number of units and hidden layers) is used for experiments in section presented in the published paper (see section 4 Experiments) and for the results on the 30 nodes power grid in this supplementary material (section 2 page ??). The number of parameters is shown in the table bellow.

Model type	30 nodes		118 nodes	
	Total parameters	Active parameters*	Total parameters	Active parameters*
DC	NA	NA	NA	NA
OM	23,672,129	23,672,129	117,536,004	117,536,004
OV	732,569	732,569	787,114	787,114
OH	740,569	740,569	824,114	824,114
GD	577,369	467,504	631,914	528,126

Table 1: Number of free parameters per models. (* active parameters: average number of parameters non set to 0 for a n-1 study).

We wanted to test our methodology without adding complementary units, to be as fair as possible during our comparison. Indeed, a model with a lot more parameters often performs better on the same task. That is why we decided to assign some unit of the 372 units intermediate layer to be "conditional unit" and didn't want to add conditional unit to the regular one.

2 Appendix 2: Results on the 30 nodes

In this subsection, we presents additional results on another matpower library [2] test case: the "case30". It was also introduced by [1]. This power grid count $n = 41$ power lines. Plain lines denotes the average error of 10 independent

training. The error bars represent the [25%, 75%] confidence interval computed with the same 10 runs.

The experiments setting for these results are the same than for the results of the 118 nodes power grid presented in the original paper. We first sample a n-1 dataset, for which we disconnect 0 or 1 power lines. We split this dataset in 3: 50% used for training, 25% for hyperparameters selection, and the last 25% use for reporting results shown in figure 2. For the "n-2" dataset, we sample 10 000 example for all the possible $41 * 40/2$ pairs disconnection. This represents the errors shown in figure 3.

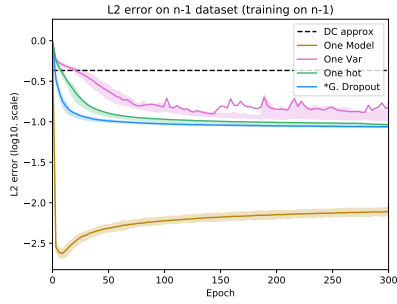


Fig. 2: **Regular generalization:** Mean-square error when training and testing with "n-1" data.

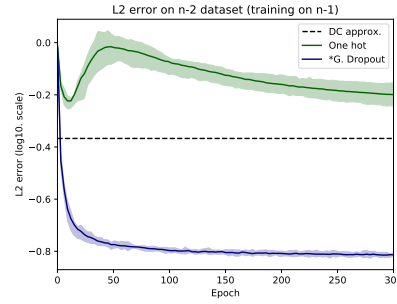


Fig. 3: **Super generalization:** Mean square error when training with "n-1" data and testing with "n-2" data.

As we can see on figures 3 and 2 the same conclusion drawn in the original paper holds. Indeed, both the "one-hot" and the "guided dropout" are able to perform pretty well on the hold-out n-1 test dataset. They achieve even better performance than the DC approximation (dashed lines on the plots). On the figure 3 (obtained on the n-2 test dataset, still while training with the n-1 train dataset), we can clearly see a huge improvement for the guided dropout models compared to the one-hot. The performance of the guided dropout only decreases by a small margin compared to the error on the n-1 test dataset, and it is still able to beat the DC approximation. On the contrary, the one-hot encoded model performs very poorly on this n-2 test dataset: it is not able to generalize its learning to more complex situations.

The main difference here resides in the weird shape of the super-generalization for the one-hot model (shown in figure 3). Indeed, the curves seem to have 3 different regimes. In the first one, the error decreases. But after approximately 10 epochs, it starts to increase again. The last regime sees this "super generalization" error continuously decreasing. We currently don't have any theoretical explanation for this behavior which will make an area of future studies.

3 Appendix 3: power flows equations

This section is a rapid overview of the problem that power flow software need to solve. For a more detailed version of these equations, one can look the matpower reference paper [2].

3.1 Model of the power grid

Let G be a grid with n nodes, m power lines.

The nodes of G are divided in two parts, namely the *generator nodes*, where at least one production unit (power plant, wind plant etc.) participating to voltage control is connected², and those called *load nodes*

To connect node i and node j there are element with complex impedance $Z_{i,j}$. If nothing connects the two, one can think of $Z_{i,j} = \infty$. Often, it is more convenient to think of the admittance Y , instead of the impedance Z . The admittance is nothing more than :

$$Y_{i,j} = \frac{1}{Z_{i,j}}$$

So if two nodes i and j are not connected, we have $Y_{i,j} = 0$.

The Ohm's law (also called Kirchoff's voltage law) between node k and node j , in complex form can be written as :

$$i_{k \rightarrow j} = Y_{i,j} \times (V_j - V_k)$$

There is another fundamental law in a power grid, the Kirchoff's power law. It states that, at a node i :

$$i_k = \sum_{j=1, j \neq i}^{n_{\text{node}}} i_{k \rightarrow j}$$

where i_k is all the complex current injected at node k and $\forall k, i_{k \rightarrow j}$ denote the (complex) current flowing from node k to node j .

If we denote by Y the matrix :

$$Y = \begin{bmatrix} \sum_{j \neq 1} Y_{1,j} & -Y_{1,2} & \dots & -Y_{1,n} \\ -Y_{2,1} & \sum_{j \neq 2} Y_{2,j} & Y_{2,3} & \dots \\ \vdots & & & \\ Y_{1,n} & Y_{2,n} & \dots & \sum_{j \neq n} Y_{n,j} \end{bmatrix}$$

²Actually, for the system to be properly specified, one node where there is a generator will be called a *slack bus*

and substituting Kirchoff's voltage law in Kirchoff's current law we have :

$$\begin{bmatrix} i_1 \\ \vdots \\ i_n \end{bmatrix} = Y \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

Y is commonly called the admittance matrix.

3.2 Equations to satisfy

A load-flow is a computation that takes as input:

- the real power for all load nodes P_D
- the reactive power for all loads nodes Q_D
- the real power for all generator nodes P_G
- the voltage magnitude $|V|$ for all generator nodes
- the voltage angle Θ for the slack bus
- the voltage magnitude $|V|$ for the slack bus

With these informations, a load-flow computes, for each load-bus the voltage angle Θ_l and magnitude $|V|_l$ and then derived other the interesting quantities, such as the active power flow, the reactive power flow, or the currents power flow on each power line of the system.

The power flow equations are, for each node (slack node, production node or load node) i of the power grid:

$$0 = -P_i + \sum_{k=1}^N |V|_i |V|_k (G_{i,k} \cos(\Theta_i - \Theta_k) + B_{i,k} \sin(\Theta_i - \Theta_k)) \quad \text{for the real power}$$

$$0 = Q_i + \sum_{k=1}^N |V|_i |V|_k (G_{i,k} \sin(\Theta_i - \Theta_k) - B_{i,k} \cos(\Theta_i - \Theta_k)) \quad \text{for the reactive power}$$

where:

- P_i is the real production injected at this node
- $G_{i,k}$ is the real part of the element in the bus admittance matrix, eg the real part of the admittance of the line connecting bug i to bus k (if any) or 0 (if not)
- $B_{i,k}$ is the imaginary part of the element in the bus admittance matrix, eg the imaginary part of the admittance of the line connecting bug i to bus k (if any) or 0 (if not)

For the system to be fully determined by these sets of equations, these equations are not written for the slack bus, and only the real part of this equation is written for the production nodes.

Once these quantities have been computed, one can compute the active power flows on each elements of the network. For example, for a given line connecting bus i to bus k with admittance Y at the origin node i having conductance S_i and susceptance B_i :

$$P_{i \rightarrow k} = |V_i| \cdot |V_k| * Y \cdot \sin(\Theta_i - \Theta_k) + |V_i|^2 \cdot S_i \quad (1)$$

$$Q_{i \rightarrow k} = -|V_i| \cdot |V_k| * Y \cdot \cos(\Theta_i - \Theta_k) + |V_i|^2 \cdot (Y - B_i) \quad (2)$$

$$I_{i \rightarrow k} = \frac{\sqrt{P_{i \rightarrow k}^2 + Q_{i \rightarrow k}^2}}{|V_i|} \quad (3)$$

3.3 DC approximation

For a more detailed information, the powerflows are shown in DCPowerFlowEquations.pdf. This section is greatly inspired from

DC power ow in unit commitment models chapter 3. In this section we will suppose that there is not transformers nor phase shifters. These two objects can of course be taken into account in the DC approximation, as shown in the two previous papers.

In this part, we will present one of the most used model to approximate the load-flow equations. In counterpart, some results of the AC model won't be accessible for example the losses or the voltage magnitudes. Despite these drawbacks, DC modelisation has two main advantages. First of all, it can always find a solution to its equations, and more importantly it is much faster to compute.

Let's recall the powerflow equations in the AC case:

$$0 = -P_i + \sum_{k=1}^N |V_i| |V_k| (G_{i,k} \cdot \cos(\Theta_i - \Theta_k) + B_{i,k} \sin(\Theta_i - \Theta_k)) \quad \text{for the real power}$$

$$0 = Q_i + \sum_{k=1}^N |V_i| |V_k| (G_{i,k} \cdot \sin(\Theta_i - \Theta_k) - B_{i,k} \cos(\Theta_i - \Theta_k)) \quad \text{for the reactive power}$$

The DC modeling will make three important assumptions:

1. the resistance (r) of a line is negligible its reactance (X)
2. For two connected buses (let's say i and k) the difference of phase $\Theta_i - \Theta_k$ is very small
3. The voltage magnitude at each bus is equal to its nominative value.

The impact of each of these assumptions on the power-flow equation will be discussed in the following subsections

3.3.1 $R \ll X$

The part has a big impact on the equations. First this induces that the losses are fully neglected.

For every lines, the admittance can be written:

$$\begin{aligned} Y &= \frac{1}{Z} \\ &= \frac{1}{R + jX} \\ &= \frac{R^2}{R^2 + X^2} - j \frac{X}{R^2 + X^2} \end{aligned}$$

And by definition, we have :

$$Y = G + jB$$

thus :

$$\begin{aligned} G &= \frac{R^2}{R^2 + X^2} \\ &\xrightarrow{R \rightarrow 0} 0 \\ &\text{and} \\ B &= \frac{-X}{R^2 + X^2} \\ &\xrightarrow{R \rightarrow 0} \frac{-1}{X} \end{aligned}$$

So the power flow equations become:

$$\begin{aligned} 0 &= -P_i + \sum_{k=1}^N |V|_i |V|_k (B_{i,k} \sin(\Theta_i - \Theta_k)) && \text{for the real power} \\ 0 &= Q_i + \sum_{k=1}^N |V|_i |V|_k (B_{i,k} \cos(\Theta_i - \Theta_k)) && \text{for the reactive power} \end{aligned}$$

3.3.2 $\Theta_i - \Theta_k \approx 0$

This will allow a linearization of the problem, as the trigonometric functions sin and cos will be approximate by the identity and the constant 1 (first order

approximation). The powerflow equations then becomes:

$$\begin{aligned} 0 &= -P_i + \sum_{k=1}^N |V|_i |V|_k B_{i,k} (\Theta_i - \Theta_k) && \text{for the real power} \\ 0 &= Q_i + \sum_{k=1}^N |V|_i |V|_k B_{i,k} && \text{for the reactive power} \end{aligned}$$

3.3.3 $|V|_j \approx |V|_{nom}$

The last non linearity in the previous equations arises due to the factor $|V|_i |V|_k$. Assuming that $|V|_j \approx |V|_{nom}$ will make them disappear. This is also a very strong assumption preventing us from getting voltage magnitude as a results of the DC approximation. This leads to:

$$|V|_i |V|_k \approx |V|_{nom} * |V|_{nom}$$

So at the end the equations are:

$$P_i = \sum_{k=1, k \neq i}^N B_{i,k} (\Theta_i - \Theta_k) \quad \text{for the real power} \quad (4)$$

$$Q_i = - \sum_{k=1}^N B_{i,k} = 0 \quad \text{for the reactive power} \quad (5)$$

3.4 Computation of current flows from DC equations

As we can see, the DC equations does not allow to capture flows in amps (A). Multiple methods allow to do that. We choose to do the following.

The AC equations 3 gives us:

$$I_{i \rightarrow k} = \frac{\sqrt{P_{i \rightarrow k}^2 + Q_{i \rightarrow k}^2}}{|V_i|}$$

We can obtained $P_{i \rightarrow k}$ from the DC equation 4, and $|V_i| = 1$ by assumptions. So the real problem is to compute $Q_{i \rightarrow k}$. Indeed, we can do better than simply assign $Q = 0$ in these formulas.

Fortunately for us, the DC equations allows to compute the voltage angles $\Theta_j \forall j$. So we can use the AC equations 2 to compute this reactive power flow values. The results of these computations is what we called the "DC approximation baseline" on this paper.

We are aware of other more advanced methods to transforms the active resulting from a DC computation to current flows that could reduce the error. But, these methods mostly rely on dispatching the losses, computed beforehand by other estimators, or by sensors data. That's why we decided not to take them into account in this paper.

References

- [1] O Alsac and B Stott. Optimal load flow with steady-state security. *IEEE transactions on power apparatus and systems*, (3):745–751, 1974.
- [2] R. D. Zimmerman and et al. Matpower. *IEEE Trans. on Power Systems*, pages 12–19, 2011.