Initializing PV-PQ Switching in Power Flow Problems Using Neural Networks

Kevin Xiang and Daniel K. Molzahn
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, USA
{kxiang,molzahn}@gatech.edu

***Abstract*—To model reactive power limited generators within power flow problems, PV-PQ switching fixes generator voltages when reactive power outputs are within limits but allows the voltages to vary with a constant reactive power injection when limits are reached. Power flow algorithms often use heuristics that iteratively modify the generators' PV versus PQ representation as the algorithm executes. The convergence behavior and speed of power flow algorithms with these heuristics significantly depend on their initialization. To improve computational performance, we propose an approach for using neural networks to initialize PV-PQ switching heuristics. After offline training where the neural networks learn the power flow solution's PV vs. PQ generator statuses across varying load demands, the neural networks are deployed to initialize power flow algorithms in online applications. Numerical results demonstrate the effectiveness of this approach with speedup factors of 1.55× to 4.32× over the nominal generator PV-PQ status initialization.**

***Index Terms*—Neural networks, Power flow, PV-PQ switching, Initialization**

I. Introduction

Relating the power injections and voltage phasors, the power flow problem is at the heart of many power system analyses. Engineers regularly solve power flow problems in applications ranging from long-term planning to daily operations. Given this problem's ubiquity, computational speed is a key concern, especially in settings where the impacts of adjustments to a power grid's operations need to be known in near real time.

The development of power flow solvers can be traced back to analog computers in the early 1900s [1, 2] and then to early digital computers in the 1940s [3]. The question of appropriate generator models was already an important issue at this time, with contemporary discussors of [3] critiquing the use of a fixed reactive injection as opposed to a fixed voltage magnitude. The later development of sparsity-exploiting Newton methods in the 1960s [4, 5] and subsequent algorithms such as the Fast Decoupled Power Flow (FDPF) [6] enabled solution of large-scale power flow problems on digital computers [7].

Despite this progress, power flow problems can still pose substantial computational difficulties. For instance, [8] characterizes these problems as "maddeningly difficult" and [9] discusses convergence challenges in the context of synthetic grids. These challenges often result from the nonlinearity of the power flow equations in combination with the disjunctive nature of generator behavior due to reactive power limits. To model the behavior of automatic voltage regulators, power flow problems typically represent generators as PV buses where both the active power and voltage magnitude are fixed to specified setpoints P and V. The generator's reactive power output varies to maintain the voltage setpoint until reaching the maximum or minimum reactive power limit at which point the generator is instead modeled with a constant reactive power output at the associated limit, i.e., a PQ bus where both active power, P, and reactive power, Q, are specified.

Power flow algorithms account for reactive power limits via so-called "PV-PQ switching" heuristics that iteratively add or remove generator voltage variables and reactive power balance equations as the bus types change throughout the algorithm's execution. Compared to neglecting reactive power limits, PV-PQ switching can substantially increase the number of Newton iterations required to solve power flow problems, especially when many generators are close to their reactive power limits. Thus, PV-PQ switching is an active research topic.

For instance, reference [10] identifies when power flow problems with PV-PQ switching models are infeasible, [11] proposes logical constraints to reduce bus type oscillations, and [12] analyzes how PV-PQ switching impacts the number of power flow solutions and their stability characteristics. Other approaches avoid explicit PV-PQ switching by instead modeling the generators' voltage vs. reactive power behavior using sigmoids [13, 14], piecewise linear functions [15], or complementarity constraints [16, 17].

While the existing literature largely focuses on modeling, analyzing, and improving PV-PQ switching heuristics, the initialization of these heuristics can also substantially impact the computational performance of power flow solvers. Accordingly, this paper applies machine learning techniques to better initialize the PV-PQ switching heuristics used by power flow solution algorithms. Using machine learning to *initialize* conventional computational methods as opposed to replacing them altogether has two key advantages: 1) errors in the approximate solutions output by machine learning models can be corrected by conventional methods to obtain highly accurate results and 2) modified initializations can easily be input to mature and well-understood implementations of conventional methods. Thus, using machine learning models to initialize conventional solution methods leverages the advantages of

Support from NSF award #2145564 is gratefully acknowledged.

both computing paradigms.

Machine learning techniques have shown significant promise in better initializing other power system algorithms, notably including AC optimal power flow solvers which typically apply iterative methods to calculate minimum cost operating points that satisfy both the power flow equations and engineering limits. Various references apply random forest [18], decision tree [19], and neural network [20, 21] techniques to initialize these solvers. Machine learning has also been applied to initialize power flow solvers in references such as [22]–[24]. However, to the best of our knowledge, the existing work on machine learning based power flow initializations focuses on initializing voltage phasors, with limited or no consideration of PV-PQ switching models or associated initializations of the generator bus types.

Building on the existing literature, we propose a neural network approach to initialize PV-PQ switching heuristics. Following typical machine learning frameworks, we first compute a training dataset via offline solution of many power flow problems with varying power injections. We then use this dataset to train neural networks that take the power injections as inputs and predict whether each generator's reactive power output will be at the upper limit, the lower limit, or strictly between the limits at the solution to the power flow problem. To achieve acceptable accuracy, we developed several techniques to tailor the neural network design and training process for this application. After training, the neural network is deployed online to accelerate the computational speed of power flow solvers. We demonstrate the performance of our proposed approach via numerical experiments with MATPOWER's PV-PQ switching heuristics [25].

The remainder of this paper is organized as follows. Section II reviews the power flow problem with PV-PQ switching. Section III presents our proposed neural network approach for PV-PQ switching initialization. Section IV numerically evaluates the performance of this approach. Section V concludes the paper and discusses future work.

II. POWER FLOW FORMULATION WITH PV/PQ SWITCHING

Consider an n-bus system with buses $\mathcal{N} = \{1, \ldots, n\}$. The network admittance matrix is denoted as $\mathbf{Y} = \mathbf{G} + j\mathbf{B}$, where $j = \sqrt{-1}$. Each bus $i \in \mathcal{N}$ has an associated voltage phasor $V_i e^{j\theta_i}$ and complex power injection $P_i + jQ_i$. The power flow equations relate the power injections and voltage phasors:

$$P_i = V_i \sum_{k \in \mathcal{N}} V_k \left(\mathbf{G}_{ik} \cos\left(\theta_i - \theta_k\right) + \mathbf{B}_{ik} \sin\left(\theta_i - \theta_k\right)\right), \tag{1a}$$

$$Q_i = V_i \sum_{k \in \mathcal{N}} V_k \left(\mathbf{G}_{ik} \sin\left(\theta_i - \theta_k\right) - \mathbf{B}_{ik} \cos\left(\theta_i - \theta_k\right)\right). \tag{1b}$$

The set of buses $\mathcal{N}$ consists of load buses and generator buses. Load buses are modeled with constant active and reactive power injections, i.e., PQ buses. Conventionally, one of the generator buses is designated as the slack bus with a fixed voltage magnitude and angle set to zero. The active and reactive power outputs at the slack bus vary to ensure conservation of power throughout the system.

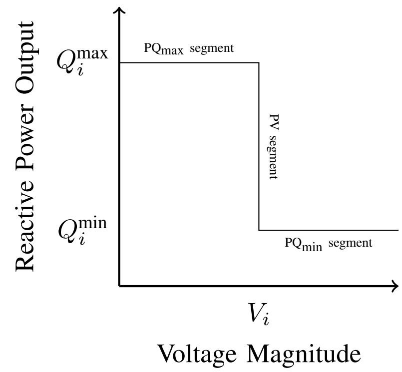


Fig. 1: Generator voltage–reactive power output characteristic.

With a conventional PV bus generator model, the remaining generator buses have fixed active power outputs and represent the behavior of automatic voltage regulators via fixed voltage magnitudes. Modeling generators as PV buses implicitly assumes that the generators can output any amount of reactive power. A more realistic generator model fixes the voltage magnitude in the same way as a PV bus when a generator's reactive power outputs are between the limits $Q_i^{\max}$ and $Q_i^{\min}$. Upon reaching the upper reactive power limit $Q_i^{\max}$, the generator's reactive power output remains at $Q_i^{\max}$ while the voltage magnitude can decrease. Conversely, upon reaching the lower reactive power limit $Q_i^{\min}$, the generator's reactive power output remains at $Q_i^{\min}$ while the voltage magnitude can increase. This behavior is represented by the curve in Fig. 1. The three line segments in this curve correspond to three bus type statuses, which we denote as $\mathrm{PQ_{max}}$ (at the upper reactive power limit), PV (between the upper and lower limits), and $\mathrm{PQ_{min}}$ (at the lower reactive power limit).

Thus, the power flow problem with reactive power limited generators seeks voltage phasor values which satisfy active power balance (1a) for all non-slack buses, reactive power balance (1b) for all load buses, and, for all non-slack generator buses, follow the curve in Fig. 1 with reactive power defined according to (1b). This yields a square system of $2n - 2$ equations in $2n - 2$ variables.

Iterative Newton-based methods are typically applied to solve power flow problems. As is the case when modeling generators as PV buses, Newton-based iterations address the smooth nonlinearities in the power flow equations. The non-smooth behavior associated with the generators' reactive power limits, as shown in Fig. 1, is usually handled via PV-PQ switching heuristics. If we knew whether each generator's reactive power output was at the upper limit, at the lower limit, or between the limits in the power flow solution, we would know the relevant segment of the curve in Fig. 1. In this case, we could instead solve a conventional power flow problem with the non-slack generators between the upper and lower reactive power limits modeled as PV buses and the remaining non-slack generators modeled as PQ buses with reactive power injections at either the upper ($\mathrm{PQ_{max}}$ buses) or lower ($\mathrm{PQ_{min}}$

buses) reactive power limits.

Of course, we do not know the power flow solution in advance as this is what we are trying to compute. Nevertheless, this intuition is the basis for typical PV-PQ switching heuristics. Starting from an initialization that predicts which segment in Fig. 1 corresponds to each generator's behavior at the eventual power flow solution, PV-PQ switching heuristics solve a conventional power flow problem with the associated bus types. If the resulting solution indeed lies on these segments of Fig. 1, the algorithm terminates. Otherwise, the PV-PQ switching heuristic changes the bus types for some subset of generators for which the solution does not lie on the curve in Fig. 1. For instance, if a generator were modeled as a PV bus but the resulting reactive power output were above $Q_i^{\max}$, selecting this generator would change the corresponding bus type to a PQ bus with reactive power injection dictated by $Q_i^{\max}$. Alternatively, if a generator were modeled as a PQ bus with reactive power output dictated by $Q_i^{\min}$ but the solution indicated a voltage magnitude that were less than the generator's setpoint voltage, selecting this generator would change the bus type to a PV bus with voltage magnitude equal to the setpoint. After updating the bus types, the power flow equations are solved again and the process repeats until all non-slack generators lie on the curve in Fig. 1.

Power flow algorithms with PV-PQ switching heuristics can thus be implemented via an inner loop that uses a Newton-based method to compute the voltage phasors corresponding to particular bus types and an outer loop that updates the bus types according to the PV-PQ switching heuristic. Different PV-PQ switching heuristics use varying strategies for selecting the subset of generators that switch bus types during each outer loop iteration. The convergence tolerance of the Newton-based inner loop could also vary as the overall algorithm proceeds.

While the initialization approach we will propose in Section III is applicable to many PV-PQ switching heuristics, we construct our training dataset and perform benchmarking using MATPOWER's PV-PQ switching heuristics [25]. In each outer loop iteration, MATPOWER's heuristics either update the bus types for all generators which do not satisfy the curve in Fig. 1 or only update the bus type for the generator that most violates this curve. Our numerical results first use the former heuristic, with the latter only attempted if the solver fails to converge. In both cases, MATPOWER converges the inner-loop's Newton method to high accuracy before updating the bus types.

III. Machine Learning Initialization Method

We would ideally initialize the bus types in a PV-PQ switching heuristic to match those of the eventual power flow solution as closely as possible. However, this is not always straightforward, particularly in systems with substantial power injection fluctuations due to, e.g., renewable generators. To address this challenge, this section presents our proposed machine learning model for bus type initializations. We first describe the machine learning model and then discuss sampling techniques for building training datasets.

A. Neural Network Model

Our approach constructs separate neural networks for each generator to predict the bus type for that generator at the power flow solution. Since voltage magnitude variations are most sensitive to nearby power injections, our neural networks only use information from nearby (three-hop-neighboring) buses to predict the bus type for each generator. Specifically, for buses within three hops of the generator, we input the values of the complex power injections and the generator voltage magnitudes. Excluding data from more distant buses enables the neural networks to focus on the most relevant information. The neural networks' bus types predictions are used to initialize a power flow algorithm with conventional PV-PQ switching heuristics (MATPOWER's heuristics, in our case). Note that the neural networks for each generator are independent and can thus be trained in parallel.

Based on the sequential model from the Keras library [26], our implementation uses a simple feedforward neural network with two hidden layers with 64 neurons and an output layer that returns a probabilistic prediction of which generator bus type ($\mathrm{PQ_{min}}$, PV, or $\mathrm{PQ_{max}}$) is most likely.

Our training process used the following hyperparameters: training data size of 10,000, early stopping patience of 10, initial training rate of 0.001, decay rate of 0.75, decay steps of 5000, and a categorical cross-entropy loss function.

B. Training Data Sampling

To generate training data, power injections and voltage magnitudes are randomly sampled from a uniform distribution around the test cases' nominal values. Our experiments consider power injection variations of $\pm 50\%$ of the nominal values and voltage magnitude variations of ± 0.10 per unit around the nominal voltage magnitudes. In practice, these ranges would be selected based on the amount of variation expected in the application of interest. To construct the training dataset, we solved the power flow equations for each random sample using MATPOWER's PV-PQ switching heuristics [25].

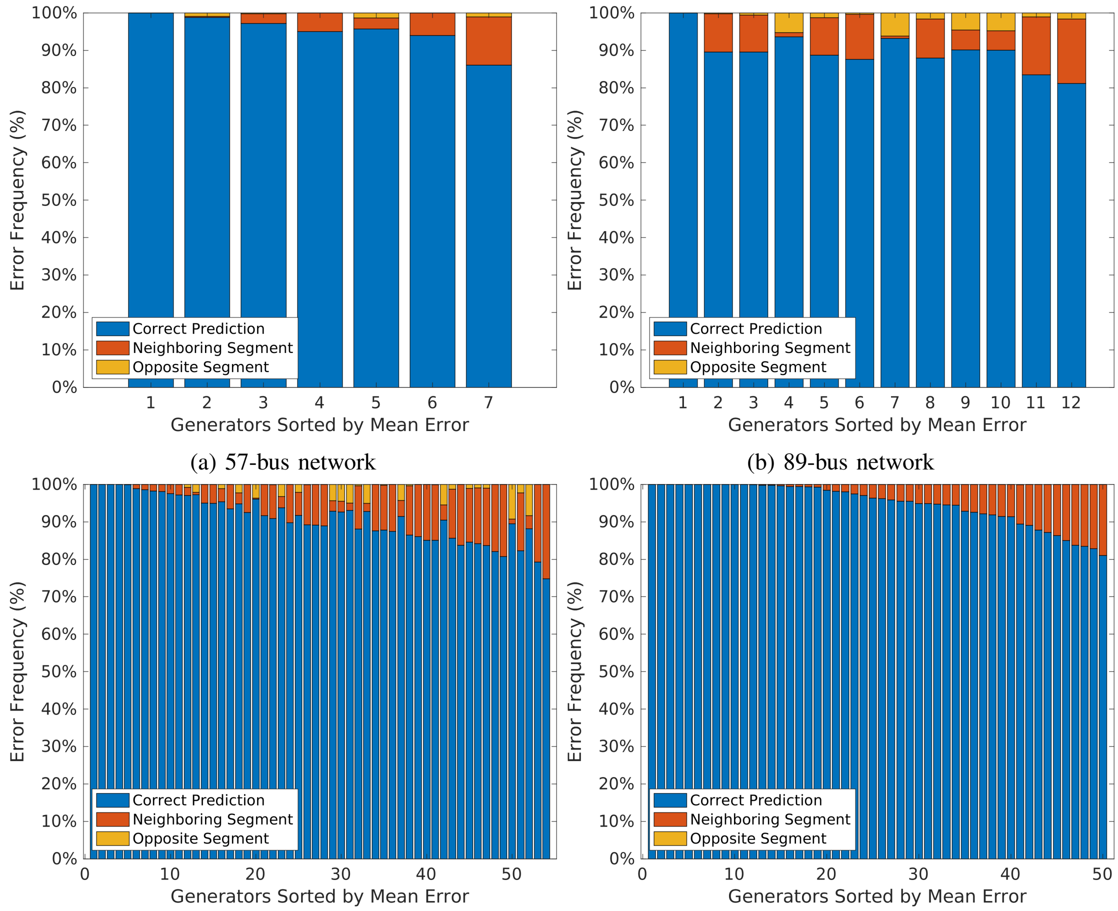
There were several data preprocessing steps needed for effective neural network training. Buses with zero power injection were removed from the training data as they provide no useful information. We also normalized the training data to be within $[0, 1]$ via linear rescaling. This latter step was especially important for effective training.

Cases where the power flow solver returned an infeasible solution were removed from training data but kept during testing to evaluate convergence rates. To avoid bias in the training data, we pruned samples so generator bus type configurations that occurred at least 10% of the maximum occurring configuration appeared at the same frequency. This also ensured configurations with few occurrences remained in the dataset without forcing undersampling.

IV. Experiments

A. Bus Type Prediction Accuracy

We benchmark our proposed approach using several standard test cases from MATPOWER [25]. Our test dataset was



(a) 57-bus network

(b) 89-bus network

(c) 118-bus network

(d) 145-bus network

Fig. 2: Prediction accuracy for each generator across various loading conditions for several test cases, sorted in order of increasing error as defined in (2).

TABLE I: Average Prediction Error for Several Test Cases

Test Case	Case57	Case89	Case118	Case145
Average Error	0.0281	0.0647	0.0501	0.0255

constructed in the same way as the training dataset described in the prior section but was not used during the training process.

The accuracies of machine learning models for classification problems is frequently evaluated using cross-entropy loss functions. However, our setting makes some categorical errors more significant than others. Specifically, if the model predicts a PQ_{min} configuration for a generator where the actual configuration is PQ_{max}, the power flow solver will likely take longer to converge than if the model had predicted a PV configuration. We therefore assess performance using the following metric:

$$\text{Error} = \frac{|f(p) - f(p')|}{2}, \tag{2}$$

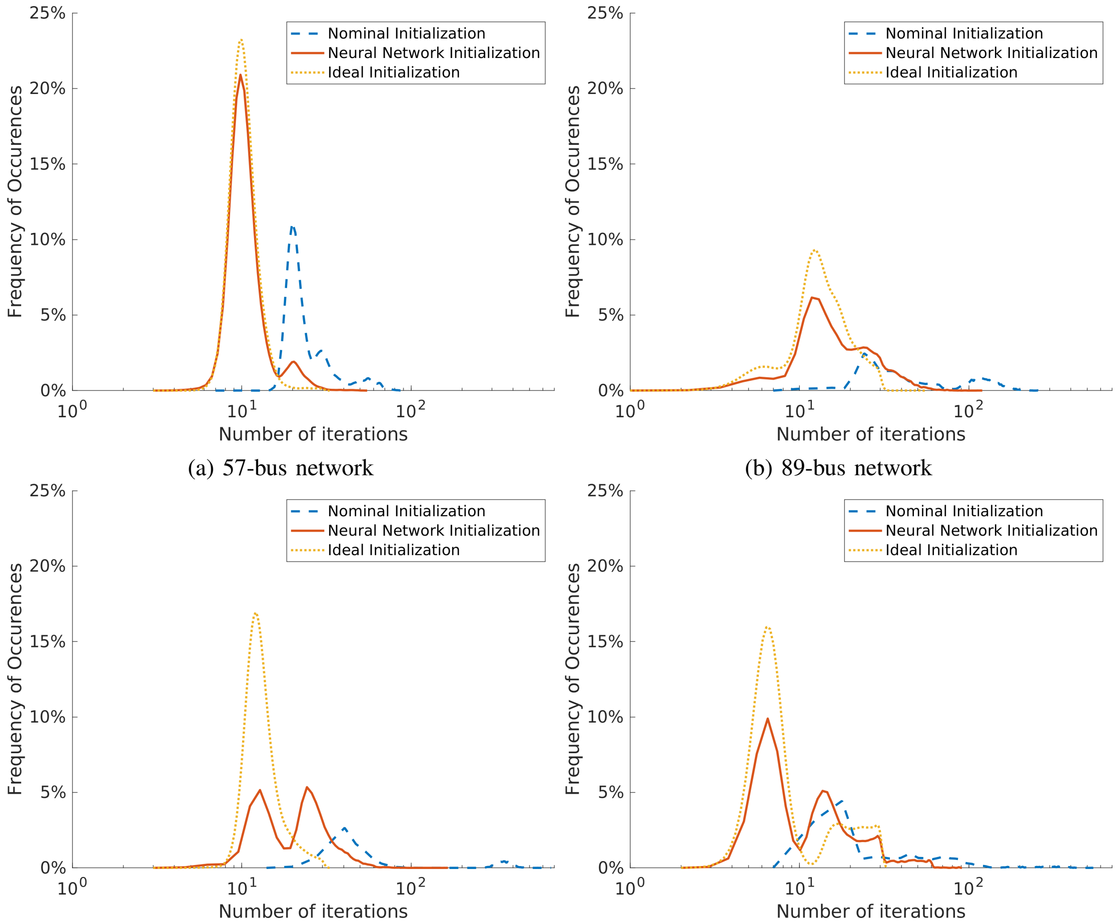
where p is the actual bus type configuration, p' is the predicted configuration, and $f : \{\text{PQ}_{\text{min}}, \text{PV}, \text{PQ}_{\text{max}}\} \rightarrow \{0, 1, 2\}$.

Table I summarizes the results of applying this error metric and Fig. 2 shows detailed distributional results. These results demonstrate that the neural network models accurately predict the bus types for a large majority of generators across a range of test cases. We note that these results are slightly influenced by a subset of generators whose configurations are the same across most loading conditions. However, bus types for other generators are usually predicted with less than 10% error.

B. *Computational Speed Improvements*

Our ultimate goal is that the predictions from the neural network models will improve computational speed. To evaluate this, we initialize MATPOWER's power flow solver with the models' predictions. To provide an upper bound on the best possible performance improvement, we benchmark against an ideal initialization that uses the actual power flow solution to initialize the bus types (but not the voltage phasors, which are set to a flat start of $1\angle 0^{\circ}$ in both initializations). Our comparisons consider the total computational time and the number of Newton iterations relative to a baseline initialization using the nominal bus types in the MATPOWER case files.

Summarizing this analysis, the results in Table II and the distributional results in Fig. 3 show that the neural network initializations yield considerable improvements over the nom-



(a) 57-bus network

(b) 89-bus network

(c) 118-bus network

(d) 145-bus network

Fig. 3: Distribution of Newton iterations with different PV-PQ initializations.

TABLE II: Effectiveness of the Neural Network Initializations Compared with Ideal Bus Type Initializations

Test Case	Case57	Case89	Case118	Case145
Time Improvement Over Nominal Initializations				
Neural Network Initialization	155%	746%	351%	432%
Ideal Initialization	191%	1591%	1029%	633%
Iteration Improvement Over Nominal Initializations				
Neural Network Initialization	143%	278%	565%	311%
Ideal Initialization	171%	436%	1089%	458%

inal bus type initializations. Many cases achieve well over half of the possible speed improvements that could be achieved by an ideal bus type initialization.

C. Convergence Rates

While our focus in this paper is on computational speed, convergence behavior is also an important performance assessment criterion. Table III shows the convergence rates for MATPOWER's power flow solver as the loading varies for several test cases. Since some of the loading conditions may actually yield infeasible power flow problems, it is most relevant to compare convergence rates between different initializations as opposed to considering the values in isolation.

TABLE III: Convergence Rates

Test Case	Case57	Case89	Case118	Case145
Nominal Initialization	99.99%	59.18%	99.95%	45.19%
Neural Network Initialization	99.91%	65.07%	91.58%	39.67%

We observe that the convergence rates associated with our neural network model are roughly comparable to those from the nominal bus type initializations, with neither dominating the other. We also note that having multiple initialization approaches provides engineers with reasonable alternatives to try in case of convergence failures.

D. Neural Network Training Times

The results shown so far provide relative comparisons among different initializations when solving power flow problems. While conducted offline where computing time is less crucial, it is also important to note the time required for training the neural networks. In particular, Table IV gives the times required for solving the 10,000 sampled power flows in data generation phase as well as the subsequent training of the neural networks. All computations were completed on Intel Xeon Gold 6226 @ 2.70GHz CPU cores, and neural network training took advantage of multiprocessing.

TABLE IV: Data Generation Time

Test Case	Case57	Case89	Case118	Case145
Data Generation Time	8.47 min	74.55 min	15.52 min	63.70 min
Network Training Time	10.80 min	8.72 min	25.81 min	17.42 min

V. Conclusion and Future Work

This paper has proposed the application of neural networks to select generator bus type initializations for the PV-PQ switching heuristics in power flow algorithms. By tailoring the neural network design and training process to this application, we obtain bus type initializations that substantially improve power flow solver times and the number of Newton iterations.

Building on the results in this paper, future work will assess more sophisticated techniques for designing the neural networks. For instance, varying the neural networks' sizes and hyperparameters based on the power system characteristics holds promise for further improvements. Moreover, we intend to select the neural networks' inputs using sensitivity information from nearby power flow solutions as opposed to our existing purely topological criterion. We also intend to combine the neural networks' bus type initializations with machine learning based voltage phasor predictions as studied in prior literature such as [22]–[24]. By doing so, we aim to further improve both convergence speed and convergence performance. Our results in this paper also suggest the merits of potential extensions to consider initializations of heuristics for other non-smooth behaviors in various power flow formulations (e.g., volt-var responses for inverter-based resources [27]). Finally, we plan to consider contingencies where the network topology changes between power flow problems.

Acknowledgment

This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA.

References

[1] "A device for calculating currents in complex networks of lines," *General Electric Review*, vol. 19, October 1916.

[2] A. Tympas, "Perpetually laborious: Computing electric power transmission before the electronic computer," *International Review of Social History*, vol. 48, no. S11, pp. 73–95, 2003.

[3] L. A. Dunstan, "Machine computation of power network performance," *Transactions of the American Institute of Electrical Engineers*, vol. 66, no. 1, pp. 610–624, 1947.

[4] N. Sato and W. F. Tinney, "Techniques for exploiting the sparsity or the network admittance matrix," *IEEE Transactions on Power Apparatus and Systems*, vol. 82, no. 69, pp. 944–950, 1963.

[5] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Transactions on Power Apparatus and systems*, no. 11, pp. 1449–1460, 1967.

[6] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Transactions on Power Apparatus and Systems*, no. 3, pp. 859–869, 1974.

[7] B. Stott, "Review of load-flow calculation methods," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 916–929, 1974.

[8] T. J. Overbye, X. Cheng, and Y. Sun, "A comparison of the AC and DC power flow models for LMP calculations," in *37th Annual Hawaii International Conference on System Sciences (HICSS)*, 2004.

[9] A. B. Birchfield, T. Xu, and T. J. Overbye, "Power flow convergence and reactive power planning in the creation of large synthetic grids," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6667–6674, 2018.

[10] D. K. Molzahn, V. Dawar, B. C. Lesieutre, and C. L. DeMarco, "Sufficient conditions for power flow insolvability considering reactive power limited generators with applications to voltage stability margins," in *9th IREP Symposium on Bulk Power System Dynamics and Control*, August 2013.

[11] J. Zhao, H.-D. Chiang, H. Li, and P. Ju, "On PV-PQ bus type switching logic in power flow computation," *16th Power Systems Computation Conference (PSCC)*, 2008.

[12] L. Zeng and H.-D. Chiang, "Multiple power flow solutions due to PV-PQ switching and their properties," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2022.

[13] Y. Kataoka, "A smooth power flow model of electric power system with generator reactive power limits taken into consideration," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 5286–5289, 2005.

[14] A. Agarwal, A. Pandey, M. Jereminov, and L. Pileggi, "Continuously differentiable analytical models for implicit control within power flow," *arXiv:1811.02000*, 2018.

[15] B. M. Allison, T. J. Overbye, and J. D. Weber, "Improved generator voltage control in power flow solutions," in *52nd North American Power Symposium (NAPS 2020)*, April 2021.

[16] L. Sundaresh and P. Nagendra Rao, "A modified Newton–Raphson load flow scheme for directly including generator reactive power limits using complementarity framework," *Electric Power Systems Research*, vol. 109, pp. 45–53, 2014.

[17] S. Gao, Y. Chen, Y. Song, S. Huang, S. Mei, D. Su, and C. Xu, "Fast decoupled power flow calculation with complementary constraints of reactive power generation and node voltage," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2020.

[18] K. Baker, "Learning warm-start points for AC optimal power flow," in *29th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019.

[19] Y. Cao, H. Zhao, G. Liang, J. Zhao, H. Liao, and C. Yang, "Fast and explainable warm-start point learning for AC optimal power flow using decision tree," *International Journal of Electrical Power & Energy Systems*, vol. 153, p. 109369, 2023.

[20] A. Deihim, D. Apostolopoulou, and E. Alonso, "Initial estimate of AC optimal power flow with graph neural networks," *Electric Power Systems Research*, vol. 234, p. 110782, 2024.

[21] X. Pan, M. Chen, T. Zhao, and S. H. Low, "DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems," *IEEE Systems Journal*, vol. 17, no. 1, pp. 673–683, 2023.

[22] L. Chen and J. E. Tate, "Hot-starting the AC power flow with convolutional neural networks," *arXiv:2004.09342*, 2020.

[23] S. N. Okhuegbe, A. A. Ademola, and Y. Liu, "A machine learning initializer for Newton-Raphson AC power flow convergence," in *IEEE Texas Power and Energy Conference (TPEC)*, 2024.

[24] S. N. Okhuegbe, A. A. Ademola, and Y. Liu, "Newton-Raphson AC power flow convergence based on deep learning initialization and homotopy continuation," to appear in *IEEE Transactions on Industry Applications*, 2025.

[25] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2010.

[26] F. Chollet, "The Sequential model." https://keras.io/guides/sequential_model, 2023. [Accessed 17-11-2024].

[27] J. W. Smith, W. Sunderman, R. Dugan, and B. Seal, "Smart inverter volt/var control functions for high penetration of PV on distribution systems," in *IEEE/PES Power Systems Conference and Exposition*, 2011.