Data-Driven Learning-Based Optimization for Distribution System State Estimation

Ahmed S. Zamzam, Xiao Fu, and Nicholas D. Sidiropoulos

Abstract

Distribution system state estimation (DSSE) is a core task for monitoring and control of distribution networks. Widely used algorithms such as Gauss-Netwon perform poorly with the limited number of measurements typically available for DSSE, often require many iterations to obtain reasonable results, and sometimes fail to converge. DSSE is a non-convex problem, and working with a limited number of measurements further aggravate the situation, as indeterminacy induces multiple global (in addition to local) minima. Gauss-Newton is also known to be sensitive to initialization. Hence, the situation is far from ideal. It is therefore natural to ask if there is a smart way of initializing Gauss-Newton that will avoid these DSSE-specific pitfalls. This paper proposes using historical or simulation-derived data to train a shallow neural network to 'learn to initialize' – that is, map the available measurements to a point in the neighborhood of the true latent states (network voltages), which is used to initialize Gauss-Newton. It is shown that this hybrid machine learning / optimization approach yields superior performance in terms of stability, accuracy, and runtime efficiency, compared to conventional optimization-only approaches. It is also shown that judicious design of the neural network training cost function helps to improve the overall DSSE performance.

Index Terms

Distribution network state estimation, phasor measurement units, machine learning, neural networks, Gauss-Newton, least squares approximation.

I. Introduction

State estimation (SE) techniques are used to monitor power grid operations in real-time. Accurately monitoring the network operating point is critical for many control and automation tasks, such as Volt/VAr optimization, feeder reconfiguration and restoration. SE uses measured quantities like nodal voltages, injections, and line flows, together with physical laws in order to obtain an estimate of the system state variables, i.e., bus voltage magnitudes and angles [1] throughout the network. SE techniques have also proven to be useful in network 'forensics', such as spotting bad measurements and identifying gross modelling errors [2].

Unlike transmission networks where measurement units are placed at almost all network nodes, the SE task in distribution systems is particularly challenging due to the scarcity of real-time measurements. This is usually compensated by the use of so-called *pseudo-measurements*. Obtained through short-term load and renewable energy forecasting techniques, these pseudo-measurements play a vital rule in enabling distribution system state estimation (DSSE) [3]–[5]. Several DSSE solvers based on weighted least squares (WLS) transmission system state estimation methods have been proposed [6]–[10]. A three-phase nodal voltage formulation was used to develop a WLS-based DSSE solver in [6], [7]. Recently, the authors of [11] used Wirtinger calculus to devise a new approach for WLS state estimation in the complex domain. In order to reduce the computational complexity and the storage requirements, the branch-based WLS model was proposed in [12], [13]. However, such gains can be only obtained when the target power system features only wye-connected loads that are solidly grounded. It is also recognized that incorporating phasor measurements in DSSE improves the observability and the estimation accuracy [14], [15]. Therefore, the DSSE approach developed in this paper considers the case where classical (quadratic)

A. S. Zamzam is with the ECE Dept., Univ. of Minnesota, Minneapolis, MN 55455, USA. X. Fu is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, 97331. N. D. Sidiropoulos is with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904. E-mails: ahmedz@umn.edu, xiao.fu@oregonstate.edu, nikos@virginia.edu.

and phasor (linear) measurements are available, as well as pseudo-measurements provided through short term forecasting algorithms.

WLS DSSE is a non-convex problem that may have multiple local minima, and working with a limited number of measurements further aggravates the situation, as it may introduce multiple global minima as well. Furthermore, Gauss-Newton type algorithms behave very differently when using different initializations—the algorithms may need many iterations, or even fail to converge. It is therefore natural to ask if there is a smart way of initializing Gauss-Newton that will avoid these pitfalls?

Contributions. In this paper, we propose a novel learning architecture for the DSSE task. Our idea is as follows. A wealth of historical data is often available for a given distribution system. This data is usually stored and utilized in various other network management tasks, such as load and injection forecasting. Even without detailed recording of the network state, we can reuse this data to simulate network operations off-line. We can then think of network states and measurements as (output,input) training pairs, which can be used to train a neural network (NN) to 'learn' a function that maps measurements to states. After the mapping function is learned, estimating the states associated with a fresh set of measurements only requires very simple operations—passing the measuremnts through the learned NN. This would greatly improve the efficiency of DSSE, bringing real-time state estimation within reach. Accurate and cheap DSSE using a NN may sound too good to be true, and in some sense (in its raw form) it is; but there is also silver lining, as we will see.

Known as universal function approximators, neural networks have made a remarkable comeback in recent years, outperforming far more complicated (and disciplined) methods in several research fields; see [16]–[18] for examples. One nice feature of neural networks and other machine learning approaches is that they alleviate the computational burden at the operation stage—by shifting computationally intensive 'hard work' to the off-line training stage.

However, accurately learning the end-to-end mapping from the available measurements to the exact network state is very challenging in our context—the accuracy achieved by convergent Gauss-Newton iterates (under good initializations) is hard to obtain using learning approaches. The mapping itself is very complex, necessitating a wide and/or deep NN that is hard to train with reasonable amounts of data. A Deep NN (DNN) also slows down real-time estimation, as passing the input through its layers is a sequential process that cannot be parallelized. To circumvent this obstacle, we instead propose to train a *shallow* neural network to 'learn to initialize'—that is, map the available measurements to a point in the neighborhood of the true latent states, which is then used to initialize Gauss-Newton; see Fig. 1 for illustration. We show that such a hybrid machine learning / optimization approach yields superior performance compared to conventional optimization-only approaches, in terms of stability, accuracy, and runtime efficiency. We demonstrate these benefits using convincing experiments with the benchmark IEEE-37 distribution feeder with several renewable energy sources installed and several types of phasor and conventional measurements, as well as pseudo-measurements. The key to success is *appropriate design of the NN training cost function* for the 'neighborhood-finding' NN. As we will see, the proposed cost function serves our purpose much better than using a generic cost function for conventional NN training.

Context. Machine learning approaches are not entirely new in the power systems / smart grid area. For instance, an online learning algorithm was used in [19] to shape residential energy demand and reduce operational costs. In [20], a multi-armed bandit online learning technique was employed to forecast the power injection of renewable energy sources. An early example of using NNs in estimation problems appeared in [21] as part of damage-adaptive intelligent flight control. Closer to our present context, [4] proposed the use of an artificial neural network that takes the measured power flows as input and aims to estimate the bus injections which is later used as pseudo-measurements in the state estimation. In contrast to our approach where the NN is used to approximate the network state given the conventional measurements as well as the pseudo-measurements, the authors of [4] designed an artificial NN to generate pseudo-measurements from the available power flow measurements. To the best of our knowledge, however, machine learning approaches have not yet been applied to the core DSSE optimization task, which is the focus of our work.

Notation: matrices (vectors) are denoted by boldface capital (small) letters; $(\cdot)^T$, $\overline{(\cdot)}$ and $(\cdot)^H$ stand for transpose, complex-conjugate and complex-conjugate transpose, respectively; and $|(\cdot)|$ denotes the magnitude of a number or the cardinality of a set.

II. Distribution System State Estimation

A. Network Representation

Consider a multi-phase distribution network consisting of $N+1$ nodes and L edges represented by a graph $\mathcal{G} := (\mathcal{N}, \mathcal{L})$, whose set of multi-phase nodes (buses) is indexed by $\mathcal{N} := \{0, 1, \ldots, N\}$, and $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ represents the lines in the network. Let the node 0 be the substation that connects the system to the transmission grid. The set of phases at bus n and line (l, m) are denoted by $\boldsymbol{\varphi}_n$ and $\boldsymbol{\varphi}_{lm}$, respectively. Let the voltage at the n-th bus for phase ϕ be denoted by $v_{n,\phi}$. Then, define $\mathbf{v}_n := [v_{n,\phi}]_{\phi \in \boldsymbol{\varphi}_n}$ to collect the voltage phasors at the phases of bus n. In addition, let the vector $\mathbf{v}$ concatenate the vectors $\mathbf{v}_n$ for all $n \in \mathcal{N}$. Lines $(l, m) \in \mathcal{L}$ are modeled as π-equivalent circuit, where phase impedance and shunt admittance are denoted by $\mathbf{Z}_{lm} \in \mathbb{C}^{|\boldsymbol{\varphi}_{lm}| \times |\boldsymbol{\varphi}_{lm}|}$ and $\check{\mathbf{Y}}_{lm} \in \mathbb{C}^{|\boldsymbol{\varphi}_{lm}| \times |\boldsymbol{\varphi}_{lm}|}$, respectively.

B. Problem Formulation

The DSSE task amounts to recovering the voltage phasors of buses given measurements related to real-time physical quantities, and the available pseudo-measurements. Actual measurements are acquired by smart meters, PMUs, and μPMUs that are placed at some locations in the distribution network. The measured quantities are usually noisy and adhere to

$$\tilde{z}_\ell = \tilde{h}_\ell(\mathbf{v}) + \xi_\ell, \qquad 1 \leq \ell \leq L_m \tag{1}$$

where ξ_ℓ amounts for the zero-mean measurement noise with known variance $\tilde{\sigma}_\ell^2$. The functions $\tilde{h}_\ell(\mathbf{v})$ are dependent on the type of the measurement, and can be either linear or quadratic relationships. In the next section, the specific form of $\tilde{h}_\ell(\mathbf{v})$ will be discussed. In addition, load and generation forecasting methods are employed to obtain pseudo-measurements that can help in identifying the network state. The forecasted quantities are modeled as

$$\check{z}_\ell = \check{h}_\ell(\mathbf{v}) + \zeta_\ell, \qquad 1 \leq \ell \leq L_s \tag{2}$$

where ζ_ℓ represents the zero-mean forecast error which has a variance of $\check{\sigma}_\ell^2$. Since $\check{z}_\ell$'s represent power-related quantities, they are usually modeled as quadratic functions of the state variable $\mathbf{v}$. While the value of the measurement noise variance $\tilde{\sigma}_\ell^2$ depends on the accuracy of the measuring equipment, the variance of the forecast error can be determined using historical forecast data.

Let $\mathbf{z}$ be a vector of length $L = L_m + L_s$ containing the measurements and pseudo-measurements, and $\mathbf{h}(\mathbf{v})$ the equation relating the measurements to the state vector $\mathbf{v}$, which will be specified in the next section. Adopting a weighted least-squares formulation, the problem can be cast as follows

$$\begin{aligned} \min_{\mathbf{v}} \; J(\mathbf{v}) &= \sum_{\ell=1}^{L_m} \tilde{w}_\ell \big(\tilde{z}_\ell - \tilde{h}_\ell(\mathbf{v})\big)^2 + \sum_{\ell=1}^{L_s} \check{w}_\ell \big(\check{z}_\ell - \check{h}_\ell(\mathbf{v})\big)^2 \\ &= \; (\mathbf{z} - \mathbf{h}(\mathbf{v}))^T \mathbf{W} (\mathbf{z} - \mathbf{h}(\mathbf{v})) \end{aligned} \tag{3}$$

where the values of $\tilde{w}_\ell$ and $\check{w}_\ell$ are inversely proportional to σ_ℓ^2 and $\check{\sigma}_\ell^2$, respectively. The optimization problem (3) is non-convex due to the nonlinearity of the measurement mappings $\mathbf{h}(\mathbf{v})$ inside the squares.

C. Available Measurements for DSSE

As indicated in the previous subsection, only few real-time measurements are usually available in distribution networks, relative to the obtainable measurements in transmission systems. Therefore, pseudo-measurements are used to alleviate the issue of solving an underdetermined problem. First, the measurements function $\tilde{h}(\mathbf{v})$ will be introduced for all types of available measurements. Then, the construction of the pseudo-measurements mappings $\check{h}(\mathbf{v})$ will be explained.

The measurement functions consist of:

• *phasor measurements* which represent the complex nodal voltages $\mathbf{v}_n$, or current flows $\mathbf{i}_{lm}$. The corresponding measurement function is linear in the state variable $\mathbf{v}$. These measurements are usually obtained by the PMUs and μPMUs. Each measurement of this type is handled as two measurements, i.e., the real and imaginary parts of the complex quantities are handled as two measurements. For the nodal voltages, the real and imaginary parts are given as follows

$$\Re\{v_{n,\phi}\} = \frac{1}{2}\,\mathbf{e}_{n,\phi}^T\,(\mathbf{v}_n + \overline{\mathbf{v}}_n), \tag{4}$$

$$\Im\{v_{n,\phi}\} = \frac{1}{2j}\,\mathbf{e}_{n,\phi}^T\,(\mathbf{v}_n - \overline{\mathbf{v}}_n) \tag{5}$$

where $\mathbf{e}_\phi$ is the ϕ-th canonical basis in $\mathbb{R}^{|\varphi_n|}$. In addition, the current flow measurements can be modeled as

$$\Re\{i_{lm,\phi}\} = \frac{1}{2}\,\mathbf{e}_{lm,\phi}^T\,\big(\mathbf{Y}_{lm}(\mathbf{v}_l - \mathbf{v}_m) + \overline{\mathbf{Y}}_{lm}(\overline{\mathbf{v}}_l - \overline{\mathbf{v}}_m)\big) \tag{6}$$

$$\Im\{i_{lm,\phi}\} = \frac{1}{2j}\,\mathbf{e}_{lm,\phi}^T\,\big(\mathbf{Y}_{lm}(\mathbf{v}_l - \mathbf{v}_m) - \overline{\mathbf{Y}}_{lm}(\overline{\mathbf{v}}_l - \overline{\mathbf{v}}_m)\big) \tag{7}$$

where $\mathbf{Y}_{lm}$ is the inverse of $\mathbf{Z}_{lm}$, and $\mathbf{e}_{lm,\phi}$ is the ϕ-th canonical basis in $\mathbb{R}^{|\varphi_{lm}|}$.

• *real-valued measurements* which encompass voltage magnitudes $|v_{n,\phi}|$, current magnitudes $|i_{lm,\phi}|$, and real and reactive power flow measurements $p_{lm,\phi}, q_{lm,\phi}$. These measurements are obtained by SCADA systems, Distribution Automation, Intelligent Electronic Devices, and PMUs. The real-valued measurements are nonlinearly related to the state variable $\mathbf{v}$. The measured voltage magnitude square, and active and reactive power flows can be represented as quadratic functions of the state variable $\mathbf{v}$, see [22]. The current magnitude squared can be written as follows

$$|i_{lm,\phi}|^2 = (\mathbf{v}_l - \mathbf{v}_m)^H \mathbf{y}_{lm,\phi}^H \mathbf{y}_{lm,\phi} (\mathbf{v}_l - \mathbf{v}_m) \tag{8}$$

where $\mathbf{y}_{lm,\phi}$ is the ϕ-th row of the admittance matrix $\mathbf{Y}_{lm}$. Therefore, all the real-valued measurements can be written as quadratic measurements of the state variable $\mathbf{v}$.

The available real-time measurements are usually insufficient to 'pin down' the network state, as we have discussed. In this case, the system is said to be unobservable. Hence, pseudo-measurements that augment the real-time measurements are crucial in DSSE as they help achieve network observability. Pseudo-measurements are obtained through load and generation forecast procedures that aim at estimating the energy consumption or generation utilizing historical data and location-based information. They are considered less accurate than real-time measurements, and hence, assigned low weights in the WLS formulation. The functions governing the mapping from the state variable to the forecasted load and renewable energy source injections can be formulated as quadratic functions [22], [23].

Therefore, any measurement synthesizing function $h_\ell(\mathbf{v})$ can be written in the following form

$$h_\ell(\mathbf{v}) = \overline{\mathbf{v}}^T \mathbf{D}_\ell \mathbf{v} + \mathbf{c}_\ell^T \mathbf{v} + \overline{\mathbf{c}}_\ell{}^T \overline{\mathbf{v}} \tag{9}$$

where $\mathbf{D}_\ell$ is a Hermitian matrix. This renders $J(\mathbf{v})$ a fourth order function of the state variable, which is very challenging to optimize

The Gauss-Newton algorithm linearizes the first order optimality conditions to iteratively update the state variables until convergence. The algorithm is known to perform well in practice given that the algorithm is initialized from a point in the vicinity of the true network state, albeit lacking provable convergence result in theory. Several variants of the algorithm have been proposed in the literature using polar [6], rectangular [24] and complex [11] representations of the state variables. All these algorithms work to a certain extent, but failure cases are also often observed. Again, stable convergence performance is only observed when the initialization is close enough to the optimal solution of (3). This is not entirely surprising—given the non-convex nature of the DSSE problem.

III. Proposed Approach: Learning-aided DSSE Optimization

Assume that there exists a mapping $\mathbf{F}(\cdot)$ such that

$$\mathbf{F}(\mathbf{z}) = \mathbf{v};$$

i.e., $\mathbf{F}(\cdot)$ maps the (noiseless) measurements to the ground-truth states. An example of such mapping is an optimization algorithm that can optimally solve the DSSE problem in the noiseless case, assuming that the solution is unique. The algorithm takes $\mathbf{z}$ as input and outputs $\mathbf{v}$. In reality the actual (and the virtual) measurements will be noisy, so we can only aim for

$$\mathbf{F}(\mathbf{z}) \approx \mathbf{v};$$

which is also what optimization-based DSSE aims for in the noisy case.

Inspired by the recent successes of machine learning, it is intriguing to ask whether it is possible to learn mapping $\mathbf{F}(\cdot)$ from historical data. If the answer is affirmative and the learned $\hat{\mathbf{F}}(\cdot)$ is easy to evaluate, then the DSSE problem could be solvable in a very efficient way *online*, after the mapping $\hat{\mathbf{F}}(\cdot)$ is learned *offline*.

In machine learning, neural networks are known as universal function approximators. In principle, a three-layer (input, hidden, output) NN can approximate any continuous multivariate function down to prescribed accuracy, if there are no constraints on the number of neurons [25]. This motivates us to consider employing a NN for approximating $\mathbf{F}(\mathbf{z})$ in the DSSE problem. A NN with vector input $\mathbf{z}$, vector output g, and one hidden layer comprising T neurons synthesizes a function of the folowing form

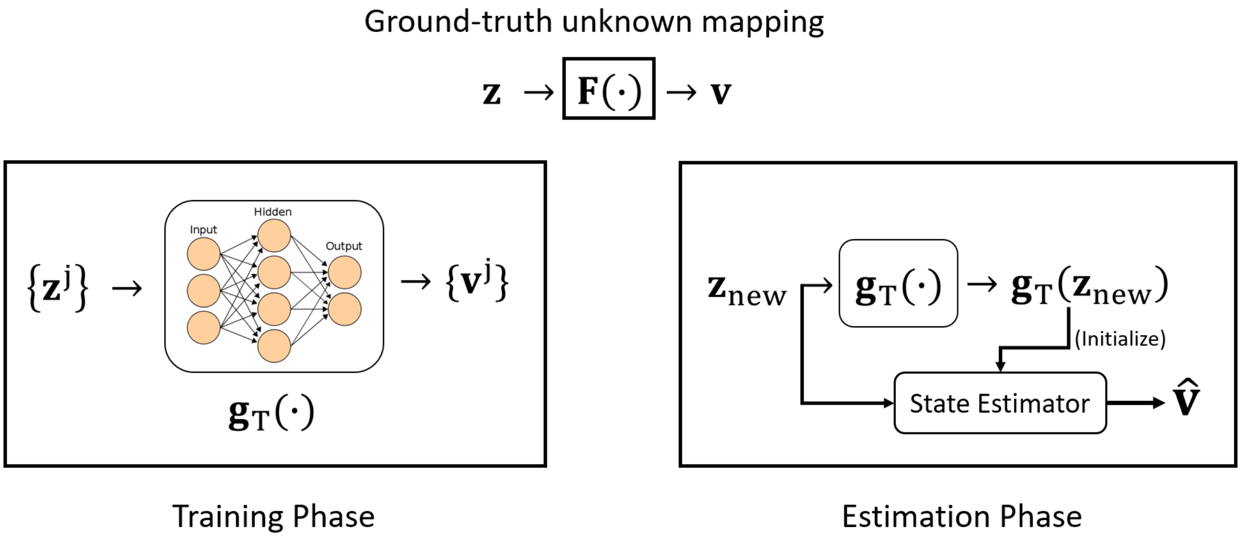
$$\mathbf{g}_T(\mathbf{z}) = \sum_{t=1}^{T} \boldsymbol{\alpha}_t \sigma(\mathbf{w}_t^T \mathbf{z} + \beta_t), \tag{10}$$

where $\mathbf{w}_t$ represents the linear combination of the inputs in $\mathbf{z}$ that is fed to the tth neuron (i.e., the unit represented by $\sigma(\mathbf{w}_t^T\mathbf{z} + \beta_t)$), β_t the corresponding scalar bias, and the vectors $\boldsymbol{\alpha}_t$'s combine the outputs of the neurons in the hidden layer to produce the vector output of the NN. The parameters $(\boldsymbol{\alpha}_t, \mathbf{w}_t, \beta_t)_{t=1}^T$ can be learned by minimizing the training cost function

$$\min_{\{\boldsymbol{\alpha}_t, \mathbf{w}_t, \beta_t\}_{t=1}^T} \sum_j \|\mathbf{v}^j - \mathbf{g}_T(\mathbf{z}^j)\|_2^2, \tag{11}$$

where the pair $(\mathbf{z}^j, \mathbf{v}^j)$ is a training sample of measurements and the corresponding underlying voltages to be estimated, in our context.

The above training cost function ideally seeks a NN that works perfectly—at least over the training set. This approach is similar in spirit to the one in [26], which considered a problem in wireless resource allocation with the objective of 'learning to optimize'—meaning, training a NN to learn the exact end-to-end input-output mapping of an optimization algorithm. Our experience has been that, for DSSE, such an approach works to some extent, but its performance is not ideal. Trying to learn the end-to-end DSSE mapping appears to be too ambitious, requiring very large T or a deep NN, and very high training sample complexity. To circumvent this obstacle, we instead propose to train a *shallow* neural network, as above, to 'learn to initialize'—that is, map the available measurements to a point in the neighborhood of the



Training Phase

Estimation Phase

Fig. 1: The proposed learning-based DSSE

true latent state, which is then used to initialize Gauss-Newton. More specifically, we propose using the following cost function for training the NN:

$$\min_{\{\mathbf{w}_t,\beta_t,\boldsymbol{\alpha}_t\}_{t=1}^T} \sum_j \max\{\|\mathbf{v}^j - \mathbf{g}_T(\mathbf{z}^j)\|_2^2 - \epsilon^2,\ 0\} \tag{12}$$

where the cost function indicates that the NN parameters are tuned with the relaxed goal that $\mathbf{g}_T(\mathbf{z}^j)$ lies in the ball of radius ϵ around $\mathbf{v}^j$. Fig. 3 illustrates the effect of changing the value of ϵ on the empirical loss function. The high-level idea is as follows: instead of enforcing minimization of $\sum_j \|\mathbf{v}^j - \mathbf{g}_T(\mathbf{z}^j)\|_2^2$, we seek a 'lazy' solution such that $\|\mathbf{v}^j - \mathbf{g}_T(\mathbf{z}^j)\|_2^2 \leq \epsilon$ for as many j as possible—in other words, it is enough to get to the right neighborhood. As we will show, this 'lowering of the bar' can significantly reduce the complexity of the NN (measured by the number of neurons) that is needed to learn such an approximate mapping, and with it also the number of training samples required for learning. These complexity benefits are obtained while still reproducing a point that is close enough to serve as a good initialization for Gauss-Newton, ensuring stable and rapid convergence. To back up this intuition, we have the following result.

Proposition 1. *Let $\sigma(\cdot)$ be any continuous sigmoidal function, and let $\mathbf{g}_T(\mathbf{z}) : \mathbb{R}^L \rightarrow \mathbb{R}^K$ be in the form*

$$\mathbf{g}_T(\mathbf{z}) = \sum_{t=1}^T \boldsymbol{\alpha}_t \sigma(\mathbf{w}_t^T \mathbf{z} + \beta_t).$$

Then, for approximating a continuous mapping $\mathbf{F} : \mathbb{R}^L \rightarrow \mathbb{R}^K$, the complexity for a shallow network to solve Problem (12) *exactly (i.e., with zero cost) for a finite number of bounded training samples $\big(\mathbf{z}^j, \mathbf{v}^j = \mathbf{F}(\mathbf{z}^j)\big)$ is at least in the order of*

$$T = \mathcal{O}\bigg(\Big(\frac{\epsilon}{\sqrt{K}}\Big)^{-\frac{L}{r}}\bigg).$$

where r is the number of continuous derivatives of $\mathbf{F}(\cdot)$.

The proof of the above proposition is relegated to Appendix A. Note that the boundedness assumption on the inputs is a proper assumption since these quantities represent voltages and powers. The implication here is very interesting, as controlling ϵ can drastically reduce the required T (and, along with it, sample complexity) while still ensuring an accurate enough prediction to enable rapid convergence of the ensuing Gauss-Newton stage. Furthermore, keeping the network shallow and T moderate makes the actual online computation (passing the input measurements through the NN to obtain the sought initialization) simple enough for real-time operation. This way, the relative strengths of learning-based and optimization-based methods can be effectively combined, and the difficulties of both methods can be circumvented.

One important remark is that Proposition 1 is derived under the assumption that $\mathbf{F}(\cdot)$ is a continuous mapping that can be parametrized with L parameters, which is hard to verify in our case. Nevertheless, we find that the theoretical result here is interesting enough and intuitively pleasing. In addition, Appendix B shows that the state estimation mapping is indeed continuous and finitely parametrizable in the case of a simple single-phase feeder. More importantly, as will be seen, this corroborating theory is consistent with our empirical results.

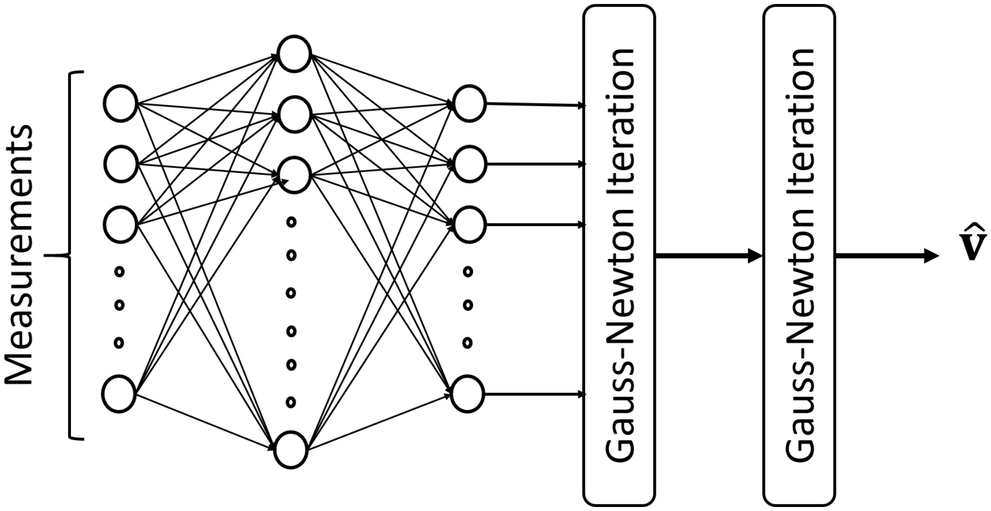


Fig. 2: Learning-based state estimator structure

In order to tune the neural network parameters, N_t training samples have to be used in order to minimize the cost function in (12). Two different ways can be utilized in order to obtain such training data. First, using historical data for load and generation, the network power flow equations can be solved to obtain the system state. Then, the measurements can be synthesized using (1) and (2). Hence, for each historical load and generation instance, a noiseless training pair $(\mathbf{z}^j, \mathbf{v}^j)$ can be generated. The second way to obtain the training data is to resort to an operating state estimation procedure. In this case, the goal of the neural network approach is to emulate the mapping of the estimator from the measurements space to the state space. The second approach suffers all the limitations of the current state estimation algorithms such as inaccuracy or computational inefficiency. In addition to providing noisy training pairs, these limitations result in a much more time consuming way of generating training data. Therefore, the first way is adopted for the rest of this paper, and the detailed procedure is presented in the experiments section.

IV. Experimental Results

The proposed state estimation procedure is tested on the benchmark IEEE-37 distribution feeder. The feeder is known to be a highly unbalanced system that has several delta-connected loads, which are blue-colored in Fig. 4. The feeder has nodes that feature different types of connections, i.e., single-, two-, and three-phase connections. Additionally, distributed energy resources are assumed to be installed at six different buses, which are colored in red in Fig. 4. In Table I, the types of the connections of all the loads and DERs are presented where (L) and (G) mean load and DER, respectively.

Historical load and generation data available in [27] modulated by the values of the loads are used to generate the training samples. Each time instance has an injection profile which is used as an input to

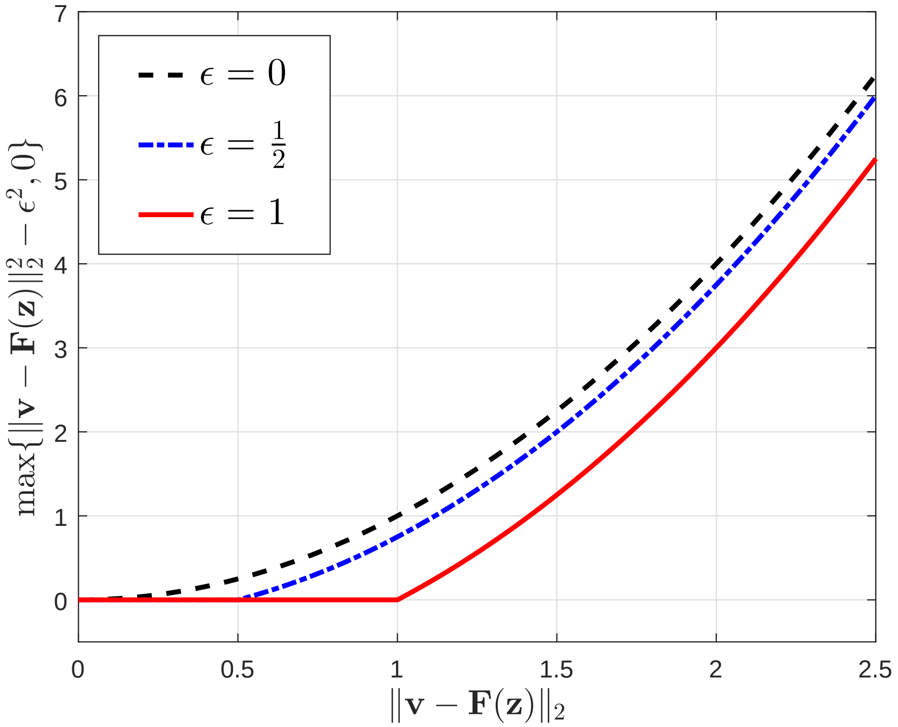


Fig. 3: The empirical loss function used for training

TABLE I: Loads and DER Connections.

Bus	Type	Connections	Bus	Type	Connections
705	(G)	a-b, b-c	728	(L)	a-b, b-c, c-a
706	(G)	b-c	729	(L)	a-b
707	(G)	b-c, c-a	730	(L)	c-a
708	(G)	b-c	731	(L)	b-c
710	(G)	a-b	732	(L)	c-a
711	(G)	c-a	733	(L)	a-b
712	(L)	c-a	734	(L)	c-a
713	(L)	c-a	735	(L)	c-a
714	(L)	a-b, b-c	736	(L)	b-c
718	(L)	a-b	737	(L)	a-b
720	(L)	c-a	738	(L)	a-b
722	(L)	b-c, c-a	740	(L)	c-a
724	(L)	b-c	741	(L)	c-a
725	(L)	b-c	742	(L)	a-b, b-c
727	(L)	c-a	744	(L)	a-b

the linearized power flow solver in [28]. The algorithm returns a voltage profile (network state variable) which is utilized to generate the value of the measurements at this point of time. A total of $100,000$ loading and generation scenarios were used to train a shallow neural network. The network has an input size of 103, 2048 nodes in the hidden layer, and output of size 210.

The available measurements are detailed as follows.

- *PMU measurements*: four PMUs are installed at buses 701, 704, 709, and 734 which are circled in Fig. 4. It assumed that the voltage phasors of all the phases are measured at these buses. This sums up to 12 complex measurement, i.e., 24 real measurements.
- *Current magnitude measurements*: The magnitude of the current flow is measured on all phases of the lines that are marked with a rhombus in Fig. 4. The number of current magnitude measurements

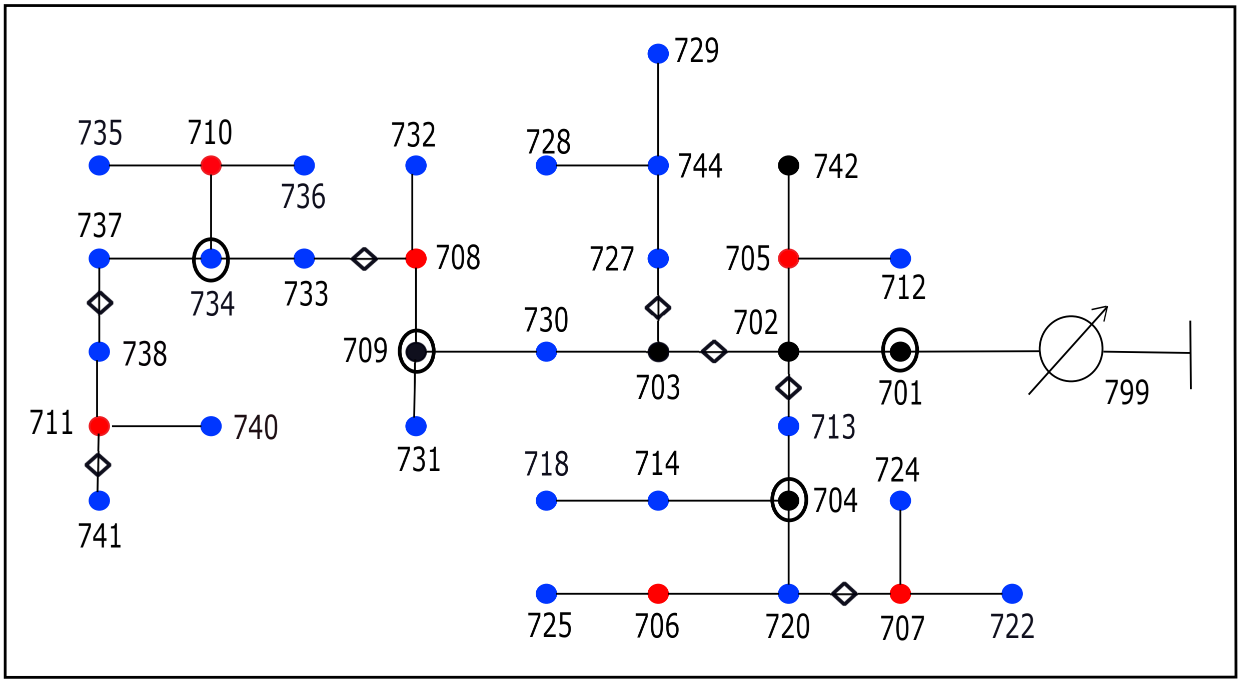


Fig. 4: IEEE-37 distribution feeder. Nodes in blue are with loads, and red nodes represent buses with DER installed. Buses with PMUs are circled, and the links where the current magnitudes are measured have a small rhombus on them.

is 21 real measurements.

- *Pseudo-measurments*: The aggregate load demand of the buses with load installed, which are blue-colored in Fig. 4, are estimated using a load forecasting algorithm using historical and situational data. Therefore, only two real quantities are obtained by the state estimator that relate to the active and reactive estimated load demand at the load buses. In addition, an energy forecast method is used to obtain an estimated injection from the renewable energy sources located at the DER buses which are colored in red in Fig. 4. The total number of load buses in the feeder is 23, and the number of distributed energy sources is 6. Therefore, the state estimator obtains 58 real pseudo-measurements relating to the active and reactive forecasted demand/injection at these buses.

The state estimator obtains noisy measurements and inexact load demands and energy generation quantities. It is assumed that the noise in the PMU voltage measurements is drawn from a Gaussian distribution with zero mean and a standard deviation of 10^{-3}. Additionally, the noise added to current magnitudes is Gaussian distributed with a standard deviation of 10^{-2}. Finally, the differences between the pseudo-measurement and the real load demand and generations are assumed to be drawn from a Gaussian distribution with a standard deviation of 10^{-1}.

The shallow neural network is trained using the TensorFlow [29] software library with 90% of the data used for training while the rest is used for verification. After tuning the network parameters, noisy measurements are generated and then passed to the state estimator architecture in Fig. 2. In order to show the effect of the modified cost function, we test the networks trained with different values of ϵ on $1,000$ loading and generation scenarios. Fig. 5 shows the histogram of the distance between the output of the shallow NN and the true network state. With the conventional training cost function ($\epsilon = 0$) the resulting distribution is more spread than the histogram that we obtain through the network trained with a relaxed cost function ($\epsilon = 1$).

Two performance indices (13)-(14) are introduced to quantify the quality of the estimate as well as the performance of the proposed approach. The first index, which is denoted by ν, represents the Frobenius norm square of the estimation error. Also, the value of the cost function at the estimate is denoted by μ.

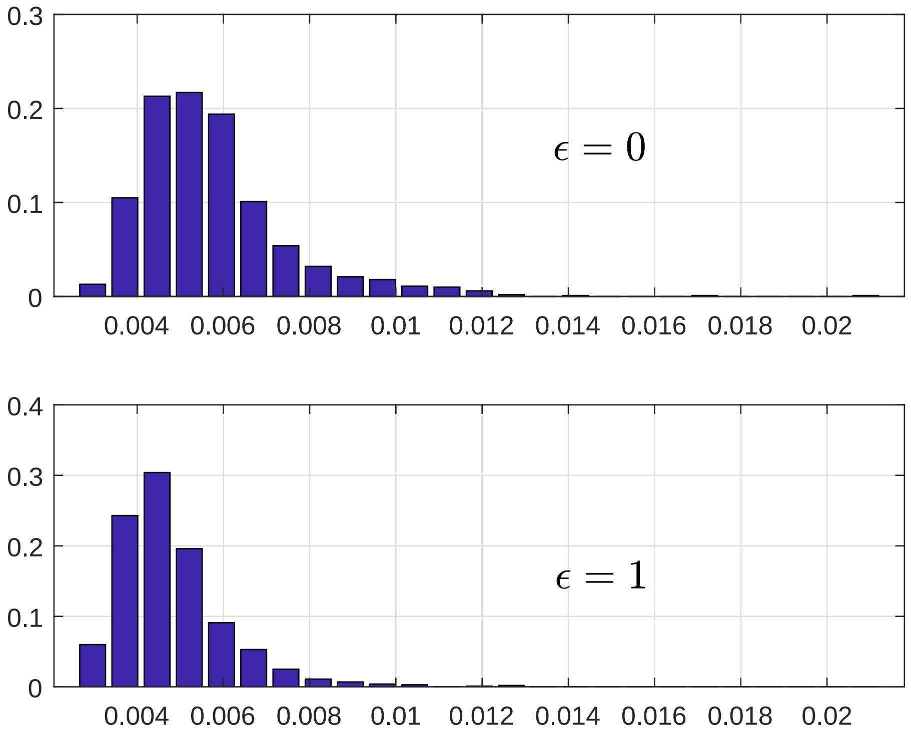


Fig. 5: Histogram of the distance between the shallow NN output and the true voltage profile with ($\epsilon = 0$) and ($\epsilon = 1$).

$$\nu = \|\hat{\mathbf{v}} - \mathbf{v}^{\text{true}}\|_2^2 \tag{13}$$

$$\mu = \sum_{\ell=1}^{L} (z_\ell - h_\ell(\hat{\mathbf{v}}))^2 \tag{14}$$

Furthermore, in order to show the effect of changing the cost function used for training, the average cost achieved using the proposed approach is shown in Table II when different values of ϵ are used for training cost function. In addition, the average number of iterations required by the Gauss-Newton iterates to converge to the optimal estimate is also presented. Using a positive value of ϵ can lead to savings up 25% in computations, which is valuable when solving the DSSE for large systems. Also, it can be seen that choosing non-zero values for ϵ enhances the performance of the proposed architecture. The estimation accuracy can be almost 5 times better using a positive ϵ. As the approximation requirement is relaxed while training the shallow NN, the network gains in generalization ability, accommodating more scenarios of loading and generation profiles.

To assess the efficacy of the proposed approach we compare it against the complex variable Gauss-Newton state estimator using [30] as a state-of-art Gauss-Newton solver for a real-valued optimization problem in complex variables. The shallow NN was trained with $\epsilon = \frac{1}{2}$ in the next comparisons.

In Table III, the average accuracy achieved in estimating the true voltage profile using both the Gauss-Newton method and the proposed architecture is presented for 1000 scenarios. In the Gauss-Newton implementation, the complex voltages provided by the PMUs are used to initialize the voltage phasors corresponding to these buses. This provides a better initialization point to the Gauss-Newton algorithm which also enhances its stability. Still, the proposed approach is able to achieve almost 10 times better

TABLE II: The estimator performance with different values of (ϵ).

$\boldsymbol{\epsilon}$	# Iterations	μ
$\mathbf{0}$	7.035	8.968×10^{-3}
$\mathbf{\frac{1}{8}}$	6.825	5.531×10^{-3}
$\mathbf{\frac{1}{4}}$	6.095	3.417×10^{-3}
$\mathbf{\frac{1}{2}}$	5.675	1.822×10^{-3}
$\mathbf{\frac{1}{\sqrt{2}}}$	5.220	5.056×10^{-3}
$\mathbf{1}$	6.150	5.859×10^{-3}
$\mathbf{2}$	6.415	1.365×10^{-2}

TABLE III: Performance comparison of different state estimators

Method	ν	μ
Proposed	9.558×10^{-3}	1.822×10^{-3}
GN	9.845×10^{-2}	4.861×10^{-2}

accuracy on average. In addition, the fitting error which represents the WLS cost function is greatly enhanced using the proposed approach.

TABLE IV: Timing and convergence of different state estimators

Method	Time (ms)	# Divergence
Proposed	347	0
G-N	2468	28

In order to assess the computational time of the proposed algorithm, we tried 1000 simulated cases for the NN-assisted state estimator and the Gauss-Newton (optimization-only) state estimator. In Table IV, the number of divergent cases out of the 1000 trials is presented for both approaches. While the Gauss-Newton approach failed to converge in 28 scenarios, the proposed architecture has converged for all considered cases. In addition, the time taken by the proposed learning approach is almost four times less than the Gauss-Newton algorithm. This is due to the fact that only few Gauss-Newton iterations need to be done when the proposed approach is utilized.

V. Conclusion

This paper presented a data-driven learning-based state estimation architecture for distribution networks. The proposed approach designs a neural network that can accommodate several types of measurements as well as pseudo-measurements. Historical load and energy generation data is used to train a neural network in order to produce an approximation of the network state. Then, this estimate is fed to a Gauss-Newton algorithm for refinement. Our realistic experiments suggest that the combination offers fast and reliable convergence to the optimal solution. The IEEE-37 test feeder was used to test the proposed approach in scenarios that include distributed energy sources. The proposed learning approach shows superior performance results in terms of the accuracy of the estimates as well as computation time.

REFERENCES

[1] V. Kekatos, G. Wang, H. Zhu, and G. B. Giannakis, "PSSE redux: Convex relaxation, decentralized, robust, and dynamic approaches," *CoRR*, vol. abs/1708.03981, 2017. [Online]. Available: http://arxiv.org/abs/1708.03981

[2] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *IEEE Trans. Power App. Syst.*, vol. 94, no. 2, pp. 329–337, Mar 1975.

[3] A. K. Ghosh, D. L. Lubkeman, and R. H. Jones, "Load modeling for distribution circuit state estimation," *IEEE Trans. on Power Del.*, vol. 12, no. 2, pp. 999–1005, Apr 1997.

[4] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, "Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling," *IEEE Trans. on Power Systems*, vol. 27, no. 4, pp. 1888–1896, Nov 2012.

[5] I. Džafić, R. A. Jabr, I. Huseinagić, and B. C. Pal, "Multi-phase state estimation featuring industrial-grade distribution network models," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 609–618, Mar 2017.

[6] M. E. Baran and A. W. Kelley, "State estimation for real-time monitoring of distribution systems," *IEEE Trans. on Power Systems*, vol. 9, no. 3, pp. 1601–1609, Aug 1994.

[7] K. Li, "State estimation for power distribution system and measurement impacts," *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 911–916, May 1996.

[8] R. Singh, B. Pal, and R. Jabr, "Choice of estimator for distribution system state estimation," *IET Generation, Transmission & Distribution*, vol. 3, no. 7, pp. 666–678, July 2009.

[9] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. on Power Systems*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[10] G. Wang, A. S. Zamzam, G. B. Giannakis, and N. D. Sidiropoulos, "Power system state estimation via feasible point pursuit: Algorithms and cramer-rao bound," *IEEE Transactions on Signal Processing*, vol. 66, no. 6, pp. 1649–1658, Mar 2018.

[11] I. Dzafic, R. A. Jabr, and T. Hrnjic, "Hybrid state estimation in complex variables," *IEEE Transactions on Power Systems*, 2018, DOI:10.1109/TPWRS.2018.2794401.

[12] M. E. Baran and A. W. Kelley, "A branch-current-based state estimation method for distribution systems," *IEEE Trans. on Power Systems*, vol. 10, no. 1, pp. 483–491, Feb 1995.

[13] H. Wang and N. N. Schulz, "A revised branch current-based distribution system state estimation algorithm and meter placement impact," *IEEE Trans. on Power Systems*, vol. 19, no. 1, pp. 207–213, Feb 2004.

[14] A. G. Phadke, J. S. Thorp, and K. Karimi, "State estimation with phasor measurements," *IEEE Trans. on Power Systems*, vol. 1, no. 1, pp. 233–238, Feb 1986.

[15] R. Zivanovic and C. Cairns, "Implementation of PMU technology in state estimation: an overview," in *IEEE AFRICON*, Stellenbosch, South Africa, 1996, pp. 1006–1011.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, Stateline,NV, 2012, pp. 1097–1105.

[17] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks," *CoRR*, vol. abs/1302.4389, 2013. [Online]. Available: http://arxiv.org/abs/1302.4389

[18] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[19] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*. Gaithersburg, MD: IEEE, 2010, pp. 409–414.

[20] X. Fang, D. Yang, and G. Xue, "Online strategizing distributed renewable energy resource access in islanded microgrids," in *IEEE Global Telecommunications Conference (GLOBECOM)*. IEEE, 2011, pp. 1–6.

[21] S. Amin, V. Gerhart, E. Rodin, S. Amin, V. Gerhart, and E. Rodin, "System identification via artificial neural networks-applications to on-line aircraft parameter estimation," in *World Aviation Congress*, Anaheim, CA, 1997, p. 5612.

[22] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," *IEEE Trans. on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sep 2013.

[23] A. S. Zamzam, N. D. Sidiropoulos, and E. Dall'Anese, "Beyond relaxation and newton-raphson: Solving AC OPF for multi-phase systems with renewables," *IEEE Trans. on Smart Grid*, 2016, DOI:10.1109/TSG.2016.2645220.

[24] R. Nuqui and A. G. Phadke, "Hybrid linear state estimation utilizing synchronized phasor measurements," in *IEEE Power Tech*. IEEE, 2007, pp. 1665–1669.

[25] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[26] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," *CoRR*, vol. abs/1705.09412, 2017. [Online]. Available: http://arxiv.org/abs/1705.09412

[27] J. Bank and J. Hambrick, "Development of a high resolution, real time, distribution-level metering system and associated visualization, modeling, and data analysis functions," National Renewable Energy Laboratory (NREL), Golden, CO., Tech. Rep., 2013.

[28] A. Garces, "A linear three-phase load flow for power distribution systems," *IEEE Trans. on Power Systems*, vol. 31, no. 1, pp. 827–828, Jan 2016.

[29] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[30] L. Sorber, M. V. Barel, and L. D. Lathauwer, "Unconstrained optimization of real functions in complex variables," *SIAM Journal on Optimization*, vol. 22, no. 3, pp. 879–898, 2012.

[31] H. N. Mhaskar, "Neural networks for optimal approximation of smooth and analytic functions," *Neural computation*, vol. 8, no. 1, pp. 164–177, 1996.

[32] W. Rudin, *Principles of mathematical analysis*. McGraw-hill New York, 1964, vol. 3.

VI. Appendix A
Proof of Proposition 1

To prove the proposition, we first invoke the following lemma:

Lemma 1 ([25, Theorem 2]). Let $\sigma(\cdot)$ be any continuous sigmoidal function. Then, given any function $f(\cdot)$ that is continuous on the d-dimensional unit cube $\boldsymbol{I}_d = [0,1]^d$, and $\epsilon > 0$, there is a sum, $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$, of the form

$$g(\mathbf{z}) = \sum_{t=1}^{T} \alpha_t \sigma(\mathbf{w}_t^T \mathbf{z} + \beta_t) \tag{15}$$

for which,

$$|g(\mathbf{z}) - f(\mathbf{z})| < \epsilon \qquad \forall \mathbf{z} \in \boldsymbol{I}_d.$$

Proof of Proposition 1. Note that the vector-valued function $\mathbf{F}(\cdot)$ can be represented as K separate scalar-valued functions. In order to prove the proposition, we start by considering approximating a scalar-valued function $f_k(\mathbf{z})$ that represents the mapping between $\mathbf{z}$ and the k-th element of $\mathbf{F}(\mathbf{z})$.

Since $\mathbf{z}^j$'s are finite with length L, finite maximum and minimum along each dimension can be obtained. Let the vectors that collect the maximum and minimum values be denoted by $\overline{\mathbf{z}}$ and $\underline{\mathbf{z}}$, respectively. Then, each training sample $\mathbf{z}^j$ is replaced by $\tilde{\mathbf{z}}^j = \mathbf{D}_{\overline{\mathbf{z}}-\underline{\mathbf{z}}}(\mathbf{z}^j - \underline{\mathbf{z}})$, where $\mathbf{D}_{\overline{\mathbf{z}}-\underline{\mathbf{z}}}$ is a diagonal matrix that has the values of $\overline{\mathbf{z}} - \underline{\mathbf{z}}$ on the diagonal. Therefore, the vectors $\tilde{\mathbf{z}}^j$ are inside the L-dimensional cube $\boldsymbol{I}_L$. According to Lemma 1, there exists a sum $\tilde{g}_k(\tilde{\mathbf{z}})$ in the form of

$$\tilde{g}_k(\tilde{\mathbf{z}}) = \sum_{t=1}^{T_k} \tilde{\alpha}_{t,k} \sigma(\tilde{\mathbf{w}}_{t,k}^T \mathbf{z} + \tilde{\beta}_{t,k}) \tag{16}$$

that satisfies

$$|f_k(\tilde{\mathbf{z}}^j) - \tilde{g}_k(\tilde{\mathbf{z}}^j)| < \epsilon_1 \qquad \forall\ \tilde{\mathbf{z}}^j \tag{17}$$

for $\epsilon_1 > 0$. Then, let $g_k(\mathbf{z})$ be a mapping in the form of (16) where the parameters are given by

$$\alpha_{t,k} = \tilde{\alpha}_{t,k}, \quad \beta_{t,k} = \tilde{\beta}_{t,k} - \tilde{\mathbf{w}}_{t,k}^T \mathbf{D}_{\overline{\mathbf{z}}-\underline{\mathbf{z}}}\ \underline{\mathbf{z}}, \quad \mathbf{w}_{t,k} = \mathbf{D}_{\overline{\mathbf{z}}-\underline{\mathbf{z}}} \tilde{\mathbf{w}}_{t,k}.$$

Then, for all $\mathbf{z}^j$ we have

$$|f_k(\mathbf{z}^j) - g_k(\mathbf{z}^j)| < \epsilon_1. \tag{18}$$

This result holds for each of the K scalar elements of $\mathbf{F}(\mathbf{z})$. Therefore, by parallel concatenation of the K neural networks used to approximate the K scalar-valued functions, we obtain a shallow neural network that has K outputs and $(\sum_i T_i)$ neurons at the hidden layer. Setting $\epsilon = \sqrt{K}\ \epsilon_1$, we deduce that there exists a sum $\mathbf{g}_T(\mathbf{z})$ in the form of (10) that satisfies

$$\|\mathbf{F}(\mathbf{z}^j) - \mathbf{g}_T(\mathbf{z}^j)\|_2 < \epsilon \qquad \forall\ \mathbf{z}^j. \tag{19}$$

It is clear now that the parameters of this function $\mathbf{g}_T(\mathbf{z})$, i.e., $\boldsymbol{\alpha}_t$, $\mathbf{w}_t$, and β_t, achieve a zero cost function solving Problem (12), and hence is optimal in solving (12).

In addition, since an approximation can be realized using any sigmoid functions, the main result in [31] specifies that the minimum number of neurons required to achieve accuracy at least ϵ_1 for a scalar-valued function is given by

$$T = \mathcal{O}(\epsilon_1^{-\frac{L}{r}}) \tag{20}$$

where r denotes the number of continuous derivatives of the approximated function $f(\mathbf{z})$, and L represents the number of parameters of the function. In order to achieve ϵ accuracy for approximating $\mathbf{F}(\mathbf{z})$, at least one of the real-valued functions that construct $\mathbf{F}(\mathbf{z})$ has to achieve $\frac{\epsilon}{\sqrt{K}}$. Hence, the complexity of shallow neural networks that optimally solve (12) for $\epsilon > 0$ is at least

$$T = \mathcal{O}\Big(\big(\frac{\epsilon}{\sqrt{K}}\big)^{-\frac{L}{r}}\Big).$$

□

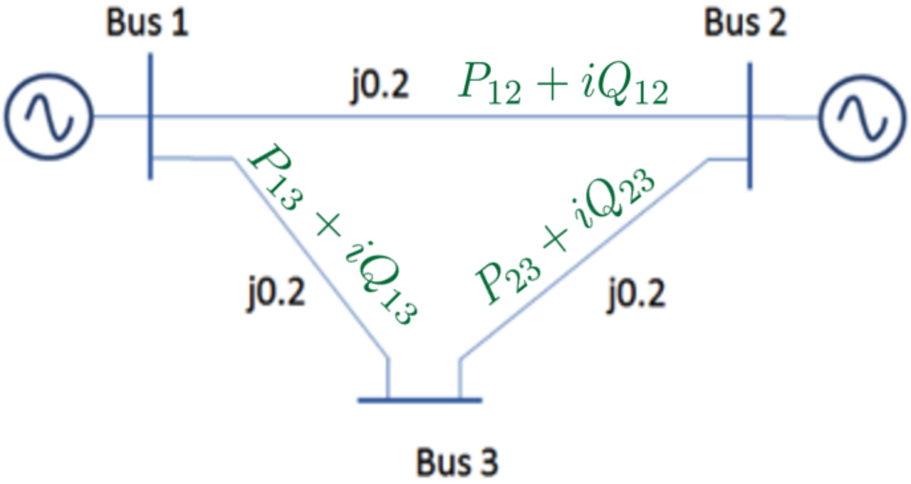


Fig. 6: Example 3-bus network

Appendix B
An Example Network

The neural networks are known as universal functions approximators. Nevertheless, the theoretical results on the ability to approximate function are usually limited to continuous functions. Hence, for continuous functions, the neural networks are expected to be able to achieve high approximation accuracy. Unfortunately, checking the continuity of the state estimation solution, which is an inverse mapping of a highly nonlinear function, is not simple to be checked.

In this appendix, a 3-bus balanced lossless network is presented, in Fig. 6, in order to inspect the continuity of the state estimation mapping. We assume that the simple network has 3 buses and that the magnitude of the voltages are measured at all buses. In addition, the active and reactive power flows are measured at all lines. Since, the phase at Bus 1 can be taken as a reference for the other buses, the state estimation problem amounts to estimating the lines phase differences, or equivalently, the phases at Bus 2 and Bus 3.

The power flow equations can be expressed as follows

$$P_{12} = B_{12}|v_1||v_2|\sin(\theta_{12}), \tag{21}$$

$$Q_{12} = |v_1|^2 - B_{12}|v_1||v_2|\cos(\theta_{12}), \tag{22}$$

$$P_{13} = B_{12}|v_1||v_3|\sin(\theta_{13}), \tag{23}$$

$$Q_{13} = |v_1|^2 - B_{12}|v_1||v_3|\cos(\theta_{13}) \tag{24}$$

where B_{ij} is the susceptance of the line between Bus i and Bus j, $|v_i|$ is the voltage magnitude at the i-th Bus, and θ_{ij} is the angle difference on the line (i,j). Assuming that the collected measurements are noiseless, the solution of the state estimation problem can be written in closed-form as

$$\theta_{12} = \sin^{-1}\left(\frac{P_{12}}{B_{12}|v_1||v_2|}\right), \tag{25}$$

$$\theta_{13} = \sin^{-1}\left(\frac{P_{13}}{B_{13}|v_1||v_3|}\right). \tag{26}$$

Claim 1. The mapping between the measurements $P_{12}, P_{13}, |v_1|,$ and $|v_2|$ and the state of the network, i.e., θ_{12} and θ_{13}, is continuous if

$$B_{12}|v_1||v_2| \geq \epsilon, \text{ and } \qquad B_{13}|v_1||v_3| \geq \epsilon \tag{27}$$

for any $\epsilon > 0$.

Proof. The proof is straightforward and build upon basic results from real functions analysis. First, the function

$$f_1(P_{12}, |v_1|, |v_2|) = \frac{P_{12}}{B_{12}|v_1||v_2|} \tag{28}$$

is continuous on $B_{12}|v_1||v_2| \in [\epsilon, \infty]$ for any $\epsilon > 0$. Then, the mapping functions in (25) is composite function of f_1 and $\sin^{-1}(\cdot)$ which is a continuous function. Therefore, the mapping in (25) is continuous on $B_{12}|v_1||v_2| \in [\epsilon, \infty]$ for any $\epsilon > 0$ [32]. The same follows for θ_{13}. □