EFFICIENT NEURAL NETWORK ARCHITECTURE FOR TOPOLOGY IDENTIFICATION IN SMART GRID

Yue Zhao[*], *Jianshu Chen*[†] *and H. Vincent Poor*[‡]

[*]Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794
[†]Microsoft Research, Redmond, WA 98052
[‡]Department of Electrical Engineering, Princeton University, Princeton, NJ 08544
Email: yue.zhao.2@stonybrook.edu, jianshuc@microsoft.com, poor@princeton.edu

ABSTRACT

Identifying arbitrary power grid topologies in real time based on measurements in the grid is studied. A learning based approach is developed: binary classifiers are trained to approximate the maximum a-posteriori probability (MAP) detectors that each identifies the status of a distinct line. An efficient neural network architecture in which features are shared for inferences of all line statuses is developed. This architecture enjoys a significant computational complexity advantage in the training and testing processes. The developed classifiers based on neural networks are evaluated in the IEEE 30-bus system. It is demonstrated that, using the proposed feature sharing neural network architecture, a) the training and testing times are drastically reduced compared with training a separate neural network for each line status inference, and b) a small amount of training data is sufficient for achieving a very good real-time topology identification performance.

Index Terms— Online power grid topology identification, line outage detection, machine learning, neural networks, cascading failures

1. INTRODUCTION

Lack of situational awareness in abnormal system conditions is a major cause of blackouts in power networks [1]. Cascading failures can quickly develop if earlier failures are not identified and contained in real time, and can lead to increasingly complex network topology changes. Protective control methods without knowledge of network topology changes may further aggravate the failure scenarios [2]. It is thus essential to identify power network topologies and failure scenarios before effective failure response mechanisms can be applied.

Identifying power grid topologies in real time under complex failure scenarios is however very challenging: The number of possible topologies grows exponentially with the number of unknown line statuses, which can be relatively large as in what has happened in major blackouts [1]. Other limitations in practice such as behaviors of human operators under time pressure, missing and contradicting information, and privacy concerns over data sharing can make this problem even harder. Assuming a small number of line failure scenarios, exhaustive search methods have been developed in [3], [4], [5] and [6] based on hypothesis testing, and in [7] and [8] based on logistic regression. To overcome the computational limit of these exhaustive search methods, assuming sparsity of simultaneous line outages, [9] has developed sparsity exploiting outage identification methods with overcomplete observations. Without assuming sparsity of line outages, a graphical model based approach has been developed for identifying arbitrary grid topologies using message passing algorithms [10]. Furthermore, a learning-to-infer methodology has recently been developed that effectively overcomes the issue of the exponential complexity of identifying arbitrary grid topologies in real time [11]. Related works on non-real-time topology identification include [12] and [13] among others.

Extending the learning-to-infer approach, in this paper, we develop learning methods based on *neural networks* to identify arbitrary power grid topologies in real time. For each line in the grid, a binary classifier is trained whose decision boundary approximates that of the maximum a-posteriori probability (MAP) detector of this line's status. The training and testing data can be generated in an arbitrarily large amount using Monte Carlo simulations. Thus neural network models can be trained without worrying about overfitting.

Two neural network architectures are developed: a) a separate neural network is trained for each line status inference, and b) a single joint neural network is trained whose features computed by its hidden layers are shared for classifying all line statuses. The two neural network architectures are evaluated in the IEEE 30-bus network [14] for identifying topologies with an arbitrary number of simultaneous line outages. It is demonstrated that, while both architectures offer very good line outage identification performance, compared with the separate training architecture, the feature sharing architecture greatly reduces the training and testing times.

This research was supported in part by the National Science Foundation under Grant ECCS-1549881.

2. PROBLEM FORMULATION

We consider a power network with N buses, and a baseline topology with L lines. We denote the incidence matrix of the baseline topology by $M \in \{-1, 0, 1\}^{N \times L}$. We use a binary variable s_l to denote the status of a line l, with $s_l = 1$ for a connected line l, and 0 otherwise. The actual topology of the network can then be represented by $\boldsymbol{s} = [s_1, \ldots, s_L]^T$. In this paper, we employ the DC power flow model for the sake of simplicity [15]. However, we note that the developed methodology can be directly extended to the AC power flow model. We denote the power injections and voltage phase angles at all the buses by $\boldsymbol{P} \in \mathbb{R}^N$ and $\boldsymbol{\theta} \in \mathbb{R}^N$, respectively. Based on the DC power flow model, we have

$$\boldsymbol{P} = MS\Gamma M^T \boldsymbol{\theta}, \tag{1}$$

where $S = \mathrm{diag}(s_1, \ldots, s_L)$, $\Gamma = \mathrm{diag}(\frac{1}{x_1}, \ldots, \frac{1}{x_L})$, and x_l is the reactance of line l.

We focus on identifying the network topology $\boldsymbol{s}$ based on real time measurements of $\boldsymbol{\theta}$ provided by phasor measurement units (PMUs) located at a subset of the buses $\mathcal{M}$, as well as knowledge of $\boldsymbol{P}$. We model the PMU measurements as

$$\boldsymbol{y} = \boldsymbol{\theta}_{\mathcal{M}} + \boldsymbol{v}, \tag{2}$$

where $\boldsymbol{\theta}_{\mathcal{M}}$ is formed by entries of $\boldsymbol{\theta}$ from buses in $\mathcal{M}$, and $\boldsymbol{v} \sim N(0, \sigma^2 I)$ contains the measurement noise.

As the observations $\boldsymbol{y}$ are made under the true underlying topology $\boldsymbol{s}$ among many possibilities, identifying the network topology $\boldsymbol{s}$ can be formulated as a hypothesis testing problem: Given $\boldsymbol{y}$ and $\boldsymbol{P}$, we would like to identify the topology $\boldsymbol{s}$ that best "fits" the relations (1) and (2). Under a Bayesian inference framework, a MAP detector would pick $\mathrm{argmax}_{\boldsymbol{s}}\, p(\boldsymbol{s}|\boldsymbol{y}, \boldsymbol{P})$ as the identification decision, which minimizes the identification error probability. However, as the number of hypotheses grows exponentially with the number of unknown line statuses, performing the hypothesis testing based on an exhaustive search becomes computationally intractable. In general, there are up to 2^L topology hypotheses.

To solve the issue of exponential complexity, we approximately decouple the hypothesis testing problem into L separate *binary* hypothesis testing problems: for each line l, the MAP detector identifies $\mathrm{argmax}_{s_l \in \{0,1\}}\, p(s_l|\boldsymbol{y}, \boldsymbol{P})$. As a result, instead of minimizing the identification error probability of the vector $\boldsymbol{s}$ (i.e., "symbol" error probability), the binary MAP detectors minimize the identification error probability of each line status s_l (i.e., "bit" error probability). The posterior marginal $p(s_l|\boldsymbol{y}, \boldsymbol{P})$, however, is very difficult to compute. (Note that summing out all $s_k, k \neq l$ requires exponential computational complexity.) As a result, even for the binary MAP detector of s_l, it is not tractable to analytically compute its decision boundary, which can be very complicated in the domain of $\boldsymbol{y}$ and $\boldsymbol{P}$.

To find the unknown decision boundary of each binary MAP detector, we employ a learning based approach exploiting the idea of the recently developed "learning-to-infer" methodology [11]. The MAP detector for line l can be viewed as a binary classifier: For each tuple of $\boldsymbol{y}$ and $\boldsymbol{P}$, this classifier outputs either $s_l = 0$ or $s_l = 1$. To learn the decision boundary of the classifier, we generate a sufficiently large number of Monte Carlo samples of $\boldsymbol{s}, \boldsymbol{y}$ and $\boldsymbol{P}$ as labeled data, and then use supervised learning to obtain a classifier close to the true MAP classifier. As will be shown in the remainder of the paper, the MAP detector's decision boundary can be learned very well with a reasonably complex classifier model.

3. NEURAL NETWORK ARCHITECTURES FOR LINE STATUS INFERENCE

We use classifiers based on neural networks to capture the complex nonlinear decision boundary for the binary MAP inference of each line status. As we have L lines, a straightforward design architecture is to train a separate classifier for each single line l: the input layer of the neural network consists of $\boldsymbol{y}$ and $\boldsymbol{P}$, and the output layer consists of just one node predicting either $s_l = 0$ or 1. Thus, a total of L classifiers need to be trained. For training and testing, we generate labeled data $\boldsymbol{s}, \boldsymbol{y}$ and $\boldsymbol{P}$ randomly that satisfy (1) and (2), where $\boldsymbol{s} = [s_1, \ldots, s_L]^T$ consists of the L labels used by the L classifiers respectively. A diagram illustrating this neural network architecture is depicted in Figure 1. The function of the neural network for classifying s_l can be understood as follows: The hidden layers of neurons compute a number of *nonlinear features* of the input $\boldsymbol{y}$ and $\boldsymbol{P}$, and the output layer applies a binary linear classifier to these features and make a decision on s_l.

Next, we introduce a second architecture that allows classifiers for different lines to *share features*, which can lead to more efficient learning of the classifiers. Specifically, instead of training L separate neural networks each with one node in its output layer, we train *one* neural network whose output layer consists of L nodes each predicting a different line's status. An illustration of this architecture is depicted in Figure 2. As a result, the features computed by the hidden layers can all be used in classifying any line's status. The idea of using shared features is that certain *common* features may provide good predictive power in inferring *many different* lines' statuses in a power network.

Furthermore, using a single neural network with feature sharing can drastically reduce the computational complexity of both the training and the testing processes. Indeed, while using separate neural networks requires training of L classifiers, using a neural network that allows feature sharing involves training of only a single classifier. Note that, with similar sizes of neural networks, adding nodes in the output layer incurs only a very small increase in the training time. As a result, there is an $O(L)$ reduction in computation time for

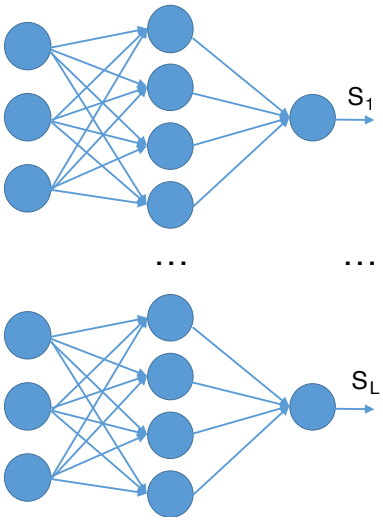

Fig. 1. L separately trained neural networks, (which could have multiple hidden layers).

this architecture with shared features, which can be significant savings for large power networks.

Evidently, compared with L separate neural networks, a shared neural network of the same size would have a performance degradation in classification due to a reduced expressive power of the model. However, such a performance degradation can be erased by increasing the size of the shared neural network. In fact, increasing the size of the shared neural network to be the sum of that of the separate neural networks leads to a classifier model that is strictly more general, and hence offers a performance enhancement as opposed to degradation. As will be shown later, it is sufficient to increase the size of the shared neural network architecture by a *much smaller factor* to achieve the same performance as the separate neural network architecture does.

This learning based approach has a major advantage in that labeled data can be generated in an arbitrarily large amount using Monte Carlo simulations. As a result, whenever overfitting is observed, it can in principle always be overcome by generating more labeled data for training. Thus, as long as the computation time allows, we can use neural network models of whatever complexity for approximating the binary MAP detectors, without worrying about overfitting.

Moreover, while the offline training process may take a reasonably long time, after the classifiers are trained, using them for online line status inference can be performed very rapidly, and is hence suitable for real time applications.

4. NUMERICAL EXPERIMENT

We evaluate the proposed learning based method for identifying power grid topologies with the IEEE 30 bus system as the baseline topology. There are 41 lines in total. As opposed to considering only a small number of line outages as in existing works, we allow *any number* of line outages, and investigate whether the learned classifiers can successfully recover the topologies.

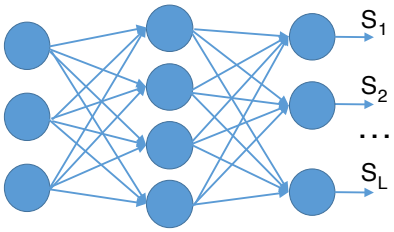

Fig. 2. A single jointly trained neural network (which could have multiple hidden layers) whose features are shared for inferring all L line statuses.

4.1. Data Set Generation

We generate the line statuses $\{s_l\}$ using independent and identically distributed (IID) Bernoulli random variables with $p(s_l = 1) = 0.6$. We do not consider disconnected power networks in this study, and exclude the line status samples if they lead to disconnected networks. As a result, there are three lines (9-11, 12-13, 25-26) that are always connected, and the dimension of the vector $\boldsymbol{s}$ reduces to 38.

We would like our classifier to be able to identify the topology for *generic values of power injections* as opposed to fixed ones. Accordingly, we generate $\boldsymbol{P}$ using the following procedure: We first generate bus voltage phase angles $\boldsymbol{\theta}$ as IID uniform random variables in $[0, 0.2\pi]$, and then compute $\boldsymbol{P}$ according to (1) under the baseline topology. Lastly, with each pair of generated $\boldsymbol{s}$ and $\boldsymbol{P}$, we generate IID phase angle measurement noise with a standard deviation of 0.01 degree, the state-of-the-art PMU accuracy [16].

In this study, we generate a total of $300K$ data samples, and use $200K$, $50K$ and $50K$ samples for training, validation, and testing, respectively. In comparison, the total number of connected topologies of the IEEE 30 bus system is on the order of $2^{38} = 2.75 \times 10^{11}$. We note that over 99% of the generated $300K$ connected topologies are distinct from each other. The *average number of disconnected lines* relative to the baseline topology is 7.8, which is significantly higher than those typically assumed in sparse line outage studies. In this experiment, we assume that all buses have PMUs that provide voltage phase angle measurements.

4.2. Hyperparameters of Neural Networks

We employ two-layer (i.e., one hidden layer) fully connected neural networks for both the separate training architecture and the feature sharing architecture. Rectified Linear Units (ReLUs) are employed as the activation functions in the hidden layer. In the output layer we employ support vector classifiers.

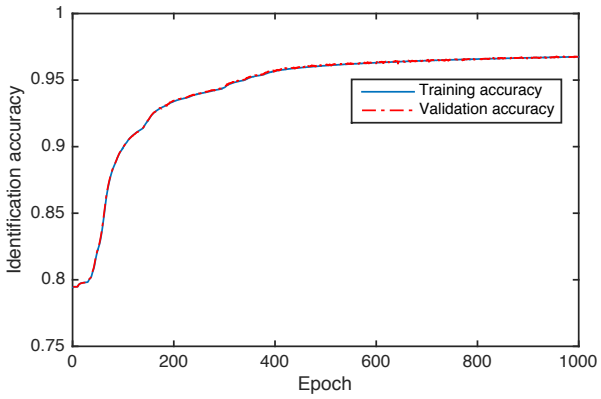

Fig. 3. Progressions of training and validation accuracies.

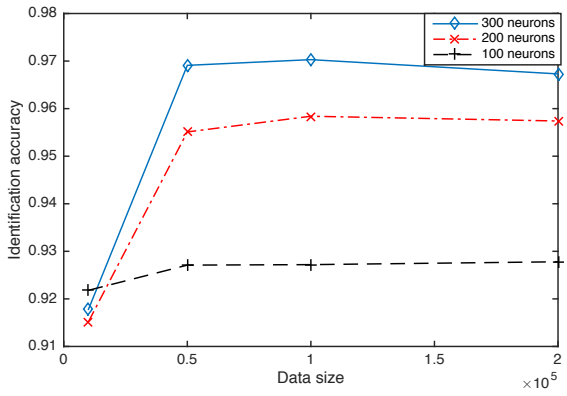

Fig. 4. Effect of model size and sample complexity.

In training the classifier, we use stochastic gradient descent (SGD) with momentum update.

4.3. Evaluation Results

4.3.1. The Separate Training Architecture vs. the Feature Sharing Architecture

We train the neural network classifiers and obtain the accuracy of identifying each line status. For separately training a neural network for each line status inference, we employ 75 neurons in the hidden layer, whereas for training a single neural network with feature sharing we employ 300 neurons. The sizes of the models are chosen such that both the separate training architecture and the feature sharing architecture achieve the *same average accuracy of 0.97*. For all neural networks, we run SGD for 1000 epochs in training. On a laptop with an Intel Core i7 3.1-GHz CPU and 8 GB of RAM, with the $200K$ data points, it takes about 7.3 hours to separately train 38 neural networks of size 75, but only 0.6 hour to train the one neural network of size 300 with feature sharing. We observe that the feature sharing architecture is about *12 times faster* to train than the separate training architecture while achieving the same performance. Such a speed advantage of the feature sharing architecture will become even more pronounced in larger power networks.

In Figure 3, we plot the achieved training and validation accuracies for every epoch with the feature sharing architecture. It is clear that the two curves stay very close to each other, and thus no overfitting is observed. We would like to further emphasize that the topologies and the power injections used to train the classifier are different from the ones in the validation and test sets. This is of particular interest because it means that our learned classifier is able to generalize well on the unseen test topologies and power injections based on its knowledge learned from the training data.

4.3.2. Model Size and Sample Complexity

In the proposed learning based method, obtaining labeled data is not an issue since data can be generated in an arbitrarily large amount using Monte Carlo simulations. This leads to two questions that are of particular interest: to learn a good classifier, a) what size of a neural network is needed? and b) how much data needs to be generated? To answer these questions, we vary the size of the neural network with shared features from 100 neurons to 300 neurons, as well as the training data size from $10K$ to $200K$, and evaluate the learned classifiers. We plot the testing results in Figure 4. We observe that a) with only $10K$ training data, neural network models of size 200 and 300 are severely overfit, as they perform even worse than a small size model of 100 neurons, and b) for these three model sizes, it is sufficient to train the neural network model with just $50K$ training data, and the achieved testing accuracy is no worse than that achieved with $200K$ training data.

5. CONCLUSION

We have developed a learning based method using neural networks for identifying arbitrary topologies of power grids in real time. The MAP detector of the entire topology is decoupled as a number of binary MAP detectors of line statuses. To learn all these binary MAP detectors' decision boundaries, classifiers based on neural networks are trained. A neural network architecture with feature sharing among all line status inferences is introduced, which offers a significant speed advantage in training and testing. Labeled data for training the classifiers can be generated in an unlimited amount using Monte Carlo simulations. We have evaluated the proposed methods with the IEEE 30-bus system. It has been demonstrated that the neural network architecture with feature sharing offers excellent performance in identifying arbitrary network topologies, and a small amount of data is sufficient for training effective classifiers. An interesting future direction is to investigate how different prior distributions for data set generation affect the online topology inference performance.

6. REFERENCES

[1] *US-Canada Power System Outage Task Force, Final Report on the August 14, 2003 Blackout in the United States and Canada*, 2004.

[2] *Arizona-Southern California Outages on September 8, 2011: Causes and Recommendations*, FERC, NERC, 2012.

[3] J. E. Tate and T. J. Overbye, "Line outage detection using phasor angle measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644 – 1652, Nov. 2008.

[4] J. E. Tate and T. J. Overbye, "Double line outage detection using phasor angle measurements," in *Proc. IEEE Power and Energy Society General Meeting*, July 2009.

[5] Y. Zhao, J. Chen, A. Goldsmith, and H. V. Poor, "Identification of outages in power systems with uncertain states and optimal sensor locations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1140–1153, Dec. 2014.

[6] Y. Zhao, A. Goldsmith, and H. V. Poor, "On PMU location selection for line outage detection in wide-area transmission networks," in *Proc. IEEE Power and Energy Society General Meeting*, July 2012, pp. 1–8.

[7] T. Kim and S. J. Wright, "PMU placement for line outage identification via multiclass logistic regression," *arXiv preprint arXiv:1409.3832*, 2014.

[8] M. Garcia, T. Catanach, S. Vander Wiel, R. Bent, and E. Lawrence, "Line outage localization using phasor measurement data in transient state," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3019–3027, 2016.

[9] H. Zhu and G. B. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2215–2224, Nov. 2012.

[10] J. Chen, Y. Zhao, A. Goldsmith, and H. V. Poor, "Line outage detection in power transmission networks via message passing algorithms," in *Proc. 48th Asilomar Conference on Signals, Systems and Computers*, 2014, pp. 350–354.

[11] Y. Zhao, J. Chen, and H. V. Poor, "Learning to infer: A new variational inference approach for power grid topology identification," in *Proc. IEEE Workshop on Statistical Signal Processing*, 2016.

[12] M. He and J. Zhang, "A dependency graph approach for fault detection and localization towards secure smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 342–351, June 2011.

[13] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, and L. Schenato, "Identification of power distribution network topology via voltage correlation analysis," in *IEEE Conference on Decision and Control*, 2013, pp. 1659–1664.

[14] *Power Systems Test Case Archive, University of Washington Electrical Engineering, https://www.ee.washington.edu/research/pstca/*.

[15] J. D. Glover, M. Sarma, and T. Overbye, *Power System Analysis & Design*, Cengage Learning, 2011.

[16] A. von Meier, D. Culler, A. McEachern, and R. Arghandeh, "Micro-synchrophasors for distribution systems," in *Proc. IEEE Innovative Smart Grid Technologies (ISGT)*, 2013.