Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods

Ferdinando Fioretto,[1,2] **Terrence W.K. Mak,**[1] **Pascal Van Hentenryck**[1]
[1]Georgia Institute of Technology, [2]Syracuse University
ffiorett@syr.edu, wmak@gatech.edu, pvh@isye.gatech.edu

Abstract

The Optimal Power Flow (OPF) problem is a fundamental building block for the optimization of electrical power systems. It is nonlinear and nonconvex and computes the generator setpoints for power and voltage, given a set of load demands. It is often solved repeatedly under various conditions, either in real-time or in large-scale studies. This need is further exacerbated by the increasing stochasticity of power systems due to renewable energy sources in front and behind the meter. To address these challenges, this paper presents a deep learning approach to the OPF. The learning model exploits the information available in the similar states of the system (which is commonly available in practical applications), as well as a dual Lagrangian method to satisfy the physical and engineering constraints present in the OPF. The proposed model is evaluated on a large collection of realistic medium-sized power systems. The experimental results show that its predictions are highly accurate with average errors as low as 0.2%. Additionally, the proposed approach is shown to improve the accuracy of the widely adopted linear DC approximation by at least two orders of magnitude.

Introduction

The *Optimal Power Flow* (OPF) problem determines the generator dispatch of minimal cost that meets the demands while satisfying the physical and engineering constraints of the power system (Chowdhury and Rahman 1990). The OPF (aka AC-OPF) is a non-convex non-linear optimization problem and the building bock of many applications, including security-constrained OPFs (Monticelli et al. 1987), optimal transmission switching (Fisher, O'Neill, and Ferris 2008), capacitor placement (Baran and Wu 1989), expansion planning (Verma et al. 2016), and security-constrained unit commitment (Wang, Shahidehpour, and Li 2008).

Typically, generation schedules are updated in intervals of 5 minutes (Tong and Ni 2011), possibly using a solution to the OPF solved in the previous step as a starting point. In recent years, the integration of renewable energy in sub-transmission and distribution systems has introduced significant stochasticity in front and behind the meter, making load profiles much harder to predict and introducing significant variations in load and generation. This uncertainty forces system operators to adjust the generators setpoints with increasing frequency in order to serve the power demand while ensuring stable network operations. However, the resolution frequency to solve OPFs is limited by their computational complexity. To address this issue, system operators typically solve OPF approximations such as the linear DC model (DC-OPF). While these approximations are more efficient computationally, their solution may be sub-optimal and induce substantial economical losses, or they may fail to satisfy the physical and engineering constraints.

Similar issues also arise in expansion planning and other configuration problems, where plans are evaluated by solving a massive number of multi-year Monte-Carlo simulations at 15-minute intervals (Pache et al. 2015; Deutche-Energue-Agentur 2019). Additionally, the stochasticity introduced by renewable energy sources further increases the number of scenarios to consider. Therefore, modern approaches recur to the linear DC-OPF approximation and focus only on the scenarios considered most pertinent (Pache et al. 2015) at the expense of the fidelity of the simulations.

To address these challenges, this paper studies how to approximate OPFs using a Deep Neural Network (DNN) approach. The main goal of the OPF is to find generator setpoints, i.e., the amount of real power and the voltage magnitude for each generator. Approximating the OPF using DNNs can thus be seen as an empirical risk minimization problem. However, the resulting setpoints must also satisfy the physical and engineering constraints that regulate power flows, and these constraints introduce significant difficulties for machine learning-based approaches, as shown in (Ng et al. 2018; Deka and Misra 2019). To address these difficulties, this paper presents a DNN approach to the OPF (OPF-DNN) that borrows ideas from Lagrangian duality and models the learning task as the Lagrangian dual of the empirical risk minimization problem under the OPF constraints. Note also that the AC-OPF is an ideal application for machine learning, since it must be solved almost continuously. Hence significant data is available to train deep learning networks and improve them over time.

The contributions of this paper can be summarized as follows. (1) It proposes an approach (OPF-DNN) that uses a

$$\mathcal{O}(\dot{\boldsymbol{p}}^d, \dot{\boldsymbol{q}}^d) = \mathbf{argmin}_{\boldsymbol{p}^g, \boldsymbol{v}} \ \sum_{i\in\mathcal{N}} \text{cost}(p_i^g) \quad (1)$$

subject to:

$$\dot{v}_i^{\min} \leqslant v_i \leqslant \dot{v}_i^{\max} \quad \forall i \in \mathcal{N} \quad (2a)$$

$$-\dot{\theta}_{ij}^{\Delta} \leqslant \theta_i - \theta_j \leqslant \dot{\theta}_{ij}^{\Delta} \quad \forall (ij) \in \mathcal{E} \ (\bar{2b})$$

$$\dot{p}_i^{g\,\min} \leqslant p_i^g \leqslant \dot{p}_i^{g\,\max} \quad \forall i \in \mathcal{N} \quad (\bar{3a})$$

$$\dot{q}_i^{g\,\min} \leqslant q_i^g \leqslant \dot{q}_i^{g\,\max} \quad \forall i \in \mathcal{N} \quad (3b)$$

$$(p_{ij}^f)^2 + (q_{ij}^f)^2 \leqslant \dot{S}_{ij}^{f\,\max} \quad \forall (ij) \in \mathcal{E} \quad (\bar{4})$$

$$p_{ij}^f = \dot{g}_{ij} v_i^2 - v_i v_j (\dot{b}_{ij} \sin(\theta_i - \theta_j) + \dot{g}_{ij} \cos(\theta_i - \theta_j)) \quad \forall (ij) \in \mathcal{E} \quad (\bar{5a})$$

$$q_{ij}^f = -\dot{b}_{ij} v_i^2 - v_i v_j (\dot{g}_{ij} \sin(\theta_i - \theta_j) - \dot{b}_{ij} \cos(\theta_i - \theta_j)) \quad \forall (ij) \in \mathcal{E} \quad (5b)$$

$$p_i^g - \dot{p}_i^d = \sum_{(ij)\in\mathcal{E}} p_{ij}^f \quad \forall i \in \mathcal{N} \quad (\bar{6a})$$

$$q_i^g - \dot{q}_i^d = \sum_{(ij)\in\mathcal{E}} q_{ij}^f \quad \forall i \in \mathcal{N} \quad (6b)$$

output: $(\boldsymbol{p}^g, \boldsymbol{v})$ – The system operational parameters

Figure 1: AC Optimal Power Flow (AC-OPF)

DNN to predict the generator setpoints for the OPF; (2) It exploits the physical and engineering constraints in a Lagrangian framework using violation degrees; (3) It enhances the prediction accuracy by leveraging the availability of a solution to a related OPF (e.g., the solution to a closely related historical instances, which is almost always available); (4) It recasts the OPF prediction as the Lagrangian dual of the empirical risk minimization under constraints, using a subgradient method to obtain a high-quality solution.

OPF-DNN is evaluated on realistic medium-sized power system benchmarks: The computational results show significant improvements in accuracy and efficiency compared to the ubiquitous DC model. In particular, OPF-DNN provides accuracy improvements of up to two orders of magnitude and efficiency speedups of several orders of magnitude. *These results may open new avenues for power system analyses and operations under significant penetration of renewable energy*.

Preliminaries

The paper uses the following notations: *Variables* are denoted by calligraph lowercase symbols, *constants* by dotted symbols, and *vectors* by bold symbols. The hat notation $\hat{\boldsymbol{x}}$ describes the prediction of a value $\boldsymbol{x}$ and $\|\cdot\|$ denotes the $L2$-norm. The power flow equations are expressed in terms of complex *powers* of the form $S = (p + jq)$, where p and q denote active and reactive powers, *admittance* of the form $Y = (g + jb)$, where g and b denote the conductance and susceptance, and *voltages* of the form $V = (v\angle\theta)$, with magnitude v and phase angle θ.

Optimal Power Flow

The *Optimal Power Flow (OPF)* determines the least-cost generator dispatch that meets the load (demand) in a power network. A power network is viewed as a graph $(\mathcal{N}, \mathcal{E})$ where the nodes $\mathcal{N}$ represent the set of n *buses* and the edges

$$\textbf{minimize:} \ \|\boldsymbol{p}^g - \hat{\boldsymbol{p}}^g\|^2 + \|\boldsymbol{v} - \hat{\boldsymbol{v}}\|^2 \quad (2)$$

$$\textbf{subject to:} \ (2a) - (6b)$$

Figure 2: The Load Flow Model

$\mathcal{E}$ represent the set of e *transmission lines*. The OPF constraints include physical and engineering constraints, which are captured in the AC-OPF formulation of Figure 1. The model uses $\boldsymbol{p}^g$, and $\boldsymbol{p}^d$ to denote, respectively, the vectors of active power generation and load associated with each bus and $\boldsymbol{p}^f$ to describe the vector of active power flows associated with each transmission line. Similar notations are used to denote the vectors of reactive power $\boldsymbol{q}$. Finally, the model uses $\boldsymbol{v}$ and $\boldsymbol{\theta}$ to describe the vectors of voltage magnitude and angles associated with each bus. The OPF takes as inputs the loads $(\dot{\boldsymbol{p}}^d, \dot{\boldsymbol{q}}^d)$ and the admittance matrix $\dot{\boldsymbol{Y}}$, with entries $\dot{g}_{ij}$ and $\dot{b}_{ij}$ for each line $(ij) \in \mathcal{E}$; It returns the active power vector $\boldsymbol{p}$ of the generators, as well the voltage magnitude $\boldsymbol{v}$ at the generator buses. The objective function (1) captures the cost of the generator dispatch, and is typically expressed as a quadratic function. Constraints (2a) and ($\bar{2b}$) restrict the voltage magnitudes and the phase angle differences within their bounds. Constraints ($\bar{3a}$) and (3b) enforce the generator active and reactive output limits. Constraints ($\bar{4}$) enforce the line flow limits. Constraints ($\bar{5a}$) and (5b) capture *Ohm's Law*. Finally, Constraint ($\bar{6a}$) and (6b) capture *Kirchhoff's Current Law* enforcing flow conservation.

The DC Relaxation The DC model is a ubiquitous linear approximation to the OPF (Wood and Wollenberg 1996). It ignores reactive power and assumes that the voltage magnitudes are at their nominal values (1.0 in per unit notation). The model uses only the barred constraints in Figure 1. Constrains ($\bar{4}$) considers only the active flows and hence can be trivially linearized and Constraints ($\bar{5a}$) becomes $p_{ij}^f = -\dot{b}_{ij}(\theta_i - \theta_j)$. The quadratic objective is also replaced by a piecewise linear function. Being an approximation, a DC solution $\hat{\boldsymbol{p}}^g$ may not satisfy the AC model constraints. As result, prior to being deployed, one typically solves a *load flow optimization*, described in Figure 2. It is a least squares minimization problem that finds the closest AC-feasible solution to the approximated one.

Deep Learning Models

Supervised Deep Learning (SDL) can be viewed as the task of approximating a complex non-linear mapping from labeled data. Deep Neural Networks (DNNs) are deep learning architectures composed of a sequence of layers, each typically taking as inputs the results of the previous layer (LeCun, Bengio, and Hinton 2015). Feed-forward neural networks are basic DNNs where the layers are fully connected and the function connecting the layer is given by $\boldsymbol{o} = \pi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$, where $\boldsymbol{x} \in \mathbb{R}^n$ and is the input vector, $\boldsymbol{o} \in \mathbb{R}^m$ the output vector, $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ a matrix of weights, and $\boldsymbol{b} \in \mathbb{R}^m$ a bias vector. The function $\pi(\cdot)$ is often non-linear (e.g., a rectified linear unit (ReLU)).

OPF Learning Goals

The goal of this paper is to learn the OPF mapping $\mathcal{O} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$: Given the loads $(\boldsymbol{p}^d, \boldsymbol{q}^d)$, predict the setpoints $(\boldsymbol{p}^g, \boldsymbol{v})$ of the generators, i.e., their active power and the voltage magnitude at their buses. The input of the learning task is a dataset $\mathcal{D} = \{(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell)\}_{\ell=1}^N$, where $\boldsymbol{x}_\ell = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ and $\boldsymbol{y}_\ell = (\boldsymbol{p}^g, \boldsymbol{v})$ represent the ℓ^{th} observation of load demands and generator setpoints which satisfy $\boldsymbol{y}_\ell = \mathcal{O}(\boldsymbol{x}_\ell)$. The output is a function $\hat{\mathcal{O}}$ that ideally would be the result of the following optimization problem

$$\begin{aligned} \textbf{minimize:} \quad & \sum_{\ell=1}^{N} \mathcal{L}_o(\boldsymbol{y}_\ell, \hat{\mathcal{O}}(\boldsymbol{x}_\ell)) \\ \textbf{subject to:} \quad & \mathcal{C}(\boldsymbol{x}_\ell, \hat{\mathcal{O}}(\boldsymbol{x}_\ell)) \end{aligned}$$

where the loss function is specified by

$$\mathcal{L}_o(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \underbrace{\|\boldsymbol{p}^g - \hat{\boldsymbol{p}}^g\|^2}_{\mathcal{L}_p(\boldsymbol{y}, \hat{\boldsymbol{y}})} + \underbrace{\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|^2}_{\mathcal{L}_v(\boldsymbol{y}, \hat{\boldsymbol{y}})} \tag{3}$$

and $\mathcal{C}(\boldsymbol{x}, \boldsymbol{y})$ holds if there exist voltage angles $\boldsymbol{\theta}$ and reactive power generated $\boldsymbol{q}^g$ that produce a feasible solution to the OPF constraints with $\boldsymbol{x} = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ and $\boldsymbol{y} = (\boldsymbol{p}^g, \boldsymbol{v})$.

One of the key difficulties of this learning task is the presence of the complex nonlinear feasibility constraints in the OPF. The approximation $\hat{\mathcal{O}}$ will typically not satisfy the problem constraints. As a result, like in the case of the DC model discussed earlier, the validation of the learning task uses a load flow computation that, given a prediction $\hat{\boldsymbol{y}} = \hat{\mathcal{O}}(\boldsymbol{x}_\ell)$, computes the closest feasible generator setpoints.

Baseline Deep Learning Model

The baseline model for this paper assumes that function $\hat{\mathcal{O}}$ is given by a feed-forward neural network, whose architecture is part of the final network outlined in Figure 3 and discussed in detail later. While this baseline model is often accurate for many regression problems, the experimental results show that it has low fidelity for complex AC-OPF tasks. More precisely, a load flow computation on the predictions of this baseline model to restore feasibility produces generator setpoints with substantial errors. The rest of the paper shows how to improve the accuracy of the model by exploiting the problem structure.

Capturing the OPF Constraints

To capture the OPF constraints, this paper uses a Lagrangian relaxation approach based on constraint violations (Fontaine, Laurent, and Van Hentenryck 2014) used in generalized augmented Lagrangian relaxation (Hestenes 1969). The Lagrangian relaxation of an optimization problem

$$\begin{aligned} \textbf{minimize:} \quad & f(\boldsymbol{x}) \\ \textbf{subject to:} \quad & h(\boldsymbol{x}) = 0; \;\; g(\boldsymbol{x}) \leqslant 0 \end{aligned}$$

is given by

$$\textbf{minimize:} \quad f(\boldsymbol{x}) + \lambda_h h(\boldsymbol{x}) + \lambda_g g(\boldsymbol{x})$$

where λ_h and $\lambda_g \geqslant 0$ are the Lagrangian multipliers. In contrast, the violation-based Lagrangian relaxation is

$$\textbf{minimize:} \quad f(\boldsymbol{x}) + \lambda_h |h(\boldsymbol{x})| + \lambda_g \max(0, g(\boldsymbol{x}))$$

with $\lambda_h, \lambda_g \geqslant 0$. In other words, the traditional Lagrangian relaxation exploits the satisfiability degrees of constraints, while the violation-based Lagrangian relaxation is expressed in terms of violation degrees. The satisfiability degree of a constraint measures how well the constraint is satisfied, with negative values representing the slack and positive values representing violations, while the violation degree is always non-negative and represents how much the constraint is violated. More formally, the satisfiability degree of a constraint $c : \mathbb{R}^n \rightarrow \mathit{Bool}$ is a function $\sigma_c : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\sigma_c(\boldsymbol{x}) \leqslant 0 \equiv c(\boldsymbol{x})$. The violation degree of a constraint $c : \mathbb{R}^n \rightarrow \mathit{Bool}$ is a function $\nu_c : \mathbb{R}^n \rightarrow \mathbb{R}^+$ such that $\sigma_c(\boldsymbol{x}) \equiv 0 \equiv c(\boldsymbol{x})$. For instance, for a linear constraints $c(\boldsymbol{x})$ of type $A\boldsymbol{x} \geqslant b$, the *satisfiability degree* is defined as

$$\sigma_c(\boldsymbol{x}) \equiv \boldsymbol{b} - A\boldsymbol{x}$$

and the *violation degrees* for inequality and equality constraints are specified by

$$\nu_c^{\geqslant}(\boldsymbol{x}) = \max(0, \sigma_c(\boldsymbol{x})) \qquad \nu_c^{=}(\boldsymbol{x}) = |\sigma_c(\boldsymbol{x})| \,.$$

Although the resulting term is not differentiable (but admits subgradients), computational experiments indicated that violation degrees are more appropriate for predicting OPFs than satisfiability degrees. Observe also that an augmented Lagrangian method uses both the satisfiability and violation degrees in its objective.

To define the violation degrees of the AC-OPF constraints, the baseline model needs to extended to predict the reactive power dispatched $\boldsymbol{q}^g$ and the voltage angles $\boldsymbol{\theta}$ of the power network. Given the predicted values $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^g$, and $\hat{\boldsymbol{q}}^g$, the satisfiability degree of (a subset of) the OPF constraints can be expressed as:

$$\begin{aligned} &\sigma_{2a}^L(\hat{v}_i) = (\dot{v}_i^{\min} - \hat{v}_i) \qquad \sigma_{2a}^R(\hat{v}_i) = (\hat{v}_i - \dot{v}_i^{\max}) && \forall i \in \mathcal{N} \\ &\sigma_{3a}^L(\hat{p}_i^g) = \dot{p}_i^{g\,\min} - \hat{p}_i^g \qquad \sigma_{3a}^R(\hat{p}_i^g) = \hat{p}_i^g - \dot{p}_i^{\max} && \forall i \in \mathcal{N} \\ &\sigma_{6a}(\hat{p}_i^g, \dot{p}_i^d, \tilde{\boldsymbol{p}}^f) = \sum_{(ij) \in \mathcal{E}} \tilde{p}_{ij}^f - (\hat{p}_i^g - \dot{p}_i^d) && \forall i \in \mathcal{N} \end{aligned}$$

where σ_{2a}^L and σ_{2a}^R correspond to Constraints (2a) and capture the distance of the predictions $\hat{v}_i$ from the voltage bounds. Similarly, σ_{3a}^L and σ_{3a}^R relate to Constraints ($\bar{3}a$) and describe the distance of the predicted generator active dispatches from their bounds. Finally, the functions σ_{6a} relate to the Kirchhoff Current Laws (Constraints ($\bar{6}a$)) and express the violation of flow conservation at a bus. Here $\tilde{p}_{ij}^f$ is the active power flow for line $(ij) \in \mathcal{E}$ computed from Constraints ($\bar{5}a$), using the predicted quantities $\hat{v}_i, \hat{v}_j, \hat{\theta}_i$, and $\hat{\theta}_j$. The complete list of the satisfiability degrees of the OPF constraints is detailed in the extended version of this paper (Fioretto, Mak, and Van Hentenryck 2019).

The violation degrees associated with the satisfiability degree above are defined as follows:

$$\nu_{2a}(\hat{\boldsymbol{v}}) = \tfrac{1}{n} \textstyle\sum_{i \in \mathcal{N}} \left(\nu_c^{\geqslant}\left(\sigma_{2a}^L(\hat{v}_i)\right) + \nu_c^{\geqslant}\left(\sigma_{2a}^R(\hat{v}_i)\right)\right)$$

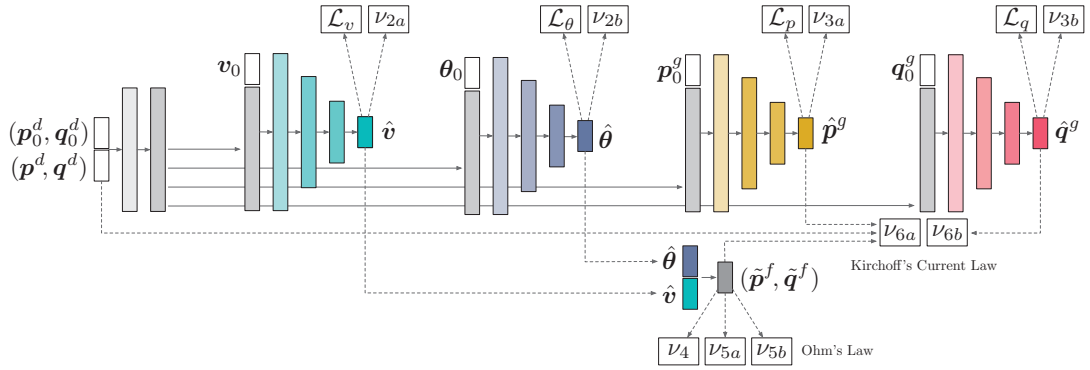

Figure 3: The OPF-DNN Model: Each layer is fully connected with ReLU activation. White boxes correspond to input tensors, dark, colored, boxes correspond to output layers. Loss components and violation degrees are shown as white rectangles.

$$\nu_{3a}(\hat{\boldsymbol{p}}) = \frac{1}{n}\sum_{i\in\mathcal{N}}\left(\nu_c^{\geqslant}\left(\sigma_{3a}^L(\hat{p}_i)\right) + \nu_c^{\geqslant}\left(\sigma_{3a}^R(\hat{p}_i)\right)\right)$$

$$\nu_{6a}(\hat{\boldsymbol{p}}^g, \dot{\boldsymbol{p}}^d, \boldsymbol{p}^f) = \frac{1}{e}\sum_{ij\in\mathcal{E}}\nu_c^{=}\left(\sigma_{6a}(\hat{p}_i^g, \dot{p}_i^d, \tilde{\boldsymbol{p}}^f)\right),$$

where n and e denote the number of buses and transmission lines, respectively. These functions capture the average deviation by which the prediction violates the associated constraint. The complete set of the OPF constraints violation degrees is given in the supplemental material. The violations degrees define penalties that will be used to enrich the DNN loss function to encourage their satisfaction. Prior describing the DNN objective, we introduce a further extension that exploits yet another aspect of the structure of the OPF.

Exploiting Existing Solutions

The solving of an OPF (or a load flow) rarely happens in a cold-start: OPFs are typically solved in the context of an existing operating point and/or with the availability of solutions to similar instances (*hot-start*). As a result, the learning task can exploit this existing configuration, which is called the *hot-start state* in this paper. The hot-start state is a tuple $\boldsymbol{s}_0 = \left(\boldsymbol{p}_0^d, \boldsymbol{q}_0^d, \boldsymbol{p}_0^g, \boldsymbol{q}_0^g, \boldsymbol{v}_0, \boldsymbol{\theta}_0\right)$, describing the load, the generation, and the voltages that are solutions to a related OPF. The learning can then use a new, enriched, training dataset, defined as follows:

$$\mathcal{D} = \Big\{\big((\overbrace{\boldsymbol{p}_0^d, \boldsymbol{q}_0^d, \boldsymbol{p}_0^g, \boldsymbol{q}_0^g, \boldsymbol{v}_0, \boldsymbol{\theta}_0, \boldsymbol{p}^d, \boldsymbol{q}^d}^{\boldsymbol{x}_1})_1, (\overbrace{\boldsymbol{p}^g, \boldsymbol{q}^g, \boldsymbol{v}, \boldsymbol{\theta}}^{\boldsymbol{y}_1})_1\big), \ldots,$$
$$\big((\underbrace{\boldsymbol{p}_0^d, \boldsymbol{q}_0^d, \boldsymbol{p}_0^g, \boldsymbol{q}_0^g, \boldsymbol{v}_0, \boldsymbol{\theta}_0, \boldsymbol{p}^d, \boldsymbol{q}^d}_{\boldsymbol{x}_N})_N, (\underbrace{\boldsymbol{p}^g, \boldsymbol{q}^g, \boldsymbol{v}, \boldsymbol{\theta}}_{\boldsymbol{y}_N})_N\big)\Big\}.$$

The elements $\boldsymbol{x}_\ell \in \mathbb{R}^{8n}$ are vectors describing the hot-start state $\boldsymbol{s}_0$ (e.g., the configuration in the previous timestep) and the current loads $(\boldsymbol{p}^d, \boldsymbol{q}^d)$. The elements $\boldsymbol{y}_\ell \in \mathbb{R}^{4n}$ are vectors describing the optimal generator and voltage settings for input data $\boldsymbol{x}_\ell$. The collection of the elements $\{\boldsymbol{x}_\ell\}_{\ell=1}^N$ is denoted by $\mathcal{X}$ and the elements $\{\boldsymbol{y}_\ell\}_{\ell=1}^N$ by $\mathcal{Y}$. The goal remain that of learning a mapping $\hat{\mathcal{O}}$. Note that, despite some proximity of loads in subsequent states, the OPF non linearities often cause severe variations in the operational parameters outputs. Therefore, as confirmed by our experimental results, the learning mechanism cannot rely exclusively on the information encoded in the hot-start state.

Objective

It is now possible to define the final loss function used to train the OPF-DNN. First, the loss is augmented to consider the predictions of voltage phase angles and the reactive power of generators, since these are required to compute the violation degrees associated with the OPF constraints. The resulting loss function $\mathcal{L}_o(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is:

$$\underbrace{\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|^2}_{\mathcal{L}_v(\boldsymbol{y},\hat{\boldsymbol{y}})} + \underbrace{\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2}_{\mathcal{L}_\theta(\boldsymbol{y},\hat{\boldsymbol{y}})} + \underbrace{\|\boldsymbol{p}^g - \hat{\boldsymbol{p}}^g\|^2}_{\mathcal{L}_p(\boldsymbol{y},\hat{\boldsymbol{y}})} + \underbrace{\|\boldsymbol{q}^g - \hat{\boldsymbol{q}}^g\|^2}_{\mathcal{L}_q(\boldsymbol{y},\hat{\boldsymbol{y}})}. \quad (4)$$

It minimizes the mean squared error between the optimal voltage and generator settings $\boldsymbol{y}$ and the predicted ones $\hat{\boldsymbol{y}}$.

Moreover, the objective function includes the Lagrangian relaxation based on the OPF physical and engineering constraints violation degrees. Given the set $\mathcal{C}$ of OPF constraints, the associated loss is captured by the expression

$$\mathcal{L}_c(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \sum_{c\in\mathcal{C}} \lambda_c \nu_c(\boldsymbol{x}, \hat{\boldsymbol{y}}).$$

The model loss function sums these two terms, i.e.,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{y}}) = \mathcal{L}_o(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \mathcal{L}_c(\boldsymbol{x}, \hat{\boldsymbol{y}}).$$

The Network Architecture

The network architecture is outlined in Figure 3. The input layers on the left process the tensor of loads $(\boldsymbol{p}_0^d, \boldsymbol{q}_0^d)$ of the hot-start state $\boldsymbol{s}_0$ and the input loads $(\boldsymbol{p}^d, \boldsymbol{q}^d)$. The network has 4 basic units, each following a decoder-encoder structure and composed by a number of fully connected layers with ReLU activations. Each subnetwork predicts a target variable: voltage magnitudes $\hat{\boldsymbol{v}}$, phase angles $\hat{\boldsymbol{\theta}}$, active power generations $\hat{\boldsymbol{p}}^g$, and reactive power generations $\hat{\boldsymbol{q}}^g$. Each sub-network takes as input the corresponding tensor in the hot-start state $\boldsymbol{s}_0$ (e.g., the sub-network responsible for predicting the voltage magnitude $\hat{\boldsymbol{v}}$ takes as input $\boldsymbol{v}_0$), as well as the last hidden layer of its input subnetwork, that processes the load tensors.

The predictions for the voltage magnitude $\hat{\boldsymbol{v}}$ and angle $\hat{\boldsymbol{\theta}}$ are used to compute the load flows $(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f)$, as illustrated on the bottom of the Figure. The components of the losses are

highlighted in the white boxes and a full description of the network architecture is provided in the extended version of this paper (Fioretto, Mak, and Van Hentenryck 2019).

Lagrangian Duality

Let $\hat{\mathcal{O}}[\boldsymbol{w}]$ be the resulting OPF-DNN with weights $\boldsymbol{w}$ and let $\mathcal{L}[\boldsymbol{\lambda}]$ be the loss function parametrized by the Lagrangian multipliers $\boldsymbol{\lambda} = \{\lambda_c\}_{c\in\mathcal{C}}$. The training aims at finding the weights $\boldsymbol{w}$ that minimize the loss function for a given set of Lagrangian multipliers, i.e., it computes

$$LR(\boldsymbol{\lambda}) = \min_{\boldsymbol{w}} \mathcal{L}[\boldsymbol{\lambda}](\boldsymbol{x}, \boldsymbol{y}, \hat{\mathcal{O}}[\boldsymbol{w}](\boldsymbol{x})).$$

It remains to determine appropriate Lagrangian multipliers. This paper proposes the use of Lagrangian duality to obtain the optimal Lagrangian multipliers when training the OPF-DNN, i.e., it solves

$$LD = \max_{\boldsymbol{\lambda}} LR(\boldsymbol{\lambda}).$$

The Lagrangian dual is solved through a subgradient method that computes a sequence of multipliers $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^k, \ldots$ by solving a sequence of trainings $LR(\boldsymbol{\lambda}^0), \ldots, LR(\boldsymbol{\lambda}^{k-1}), \ldots$ and adjusting the multipliers using the violations, i.e.,

$$\boldsymbol{w}^{k+1} = \operatorname*{argmin}_{\boldsymbol{w}} \mathcal{L}[\boldsymbol{\lambda}^k](\boldsymbol{x}, \boldsymbol{y}, \hat{\mathcal{O}}[\boldsymbol{w}^k](\boldsymbol{x})) \quad \text{(L1)}$$

$$\boldsymbol{\lambda}^{k+1} = \Big(\lambda_c^k + \rho\, \nu_c(\boldsymbol{x}, \hat{\mathcal{O}}[\boldsymbol{w}^{k+1}](\boldsymbol{x})) \mid c \in \mathcal{C}\Big). \quad \text{(L2)}$$

In the implementation, step (L1) is approximated using a Stochastic Gradient Descent (SGD) method. Importantly, this step does not recomputes the training from scratch but uses a hot start for the weights $\boldsymbol{w}$.

The overall training scheme is presented in Algorithm 1. It takes as input the training dataset $(\mathcal{X}, \mathcal{Y})$, the optimizer step size $\alpha > 0$ and the Lagrangian step size $\rho > 0$. The Lagrangian multipliers are initialized in line 1. The training is performed for a fixed number of epochs, and each epoch optimizes the weights using a minibatch of size b. After predicting the voltage and generation power quantities (line 4), the objective and constraint losses are computed (lines 5 and 6). The latter uses the Lagrangian multipliers $\boldsymbol{\lambda}^k$ associated with current epoch k. The model weights are updated in line 7. Finally, after each epoch, the Lagrangian multipliers are updated following step (L2) described above (lines 8 and 9).

Experiments

This section evaluates the predictive accuracy of OPF-DNN and compares it to the AC model and its linear DC approximation. It also analyzes various design decisions in detail.

Data sets The experiments examine the proposed models on a variety of power networks from the NESTA library (Coffrin, Gordon, and Scott 2014). For presentation simplicity, the analysis focuses primarily on the IEEE 30, 118, and 300-bus networks. However, the results are consistent across the entire benchmark set (see (Fioretto, Mak, and Van Hentenryck 2019)). The ground truth data are constructed as follows: For each network, different benchmarks are generated

Algorithm 1: *Learning Step*

input: $(\mathcal{X}, \mathcal{Y})$: Training data
α, ρ : Optimizer and Lagrangian step sizes, reps.

1 $\boldsymbol{\lambda}^0 \leftarrow 0 \;\; \forall c \in \mathcal{C}$
2 **for** *epoch* $k = 0, 1, \ldots$ **do**
3 **foreach** $(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ *minibatch*$(\mathcal{X}, \mathcal{Y})$ *of size* b **do**
4 $\hat{\boldsymbol{y}} \leftarrow \hat{\mathcal{O}}[\boldsymbol{w}](\boldsymbol{x})$
5 $\mathcal{L}_o(\hat{\boldsymbol{y}}, \boldsymbol{y}) \leftarrow \frac{1}{b}\sum_{\ell\in[b]} \mathcal{L}_v(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell) + \mathcal{L}_\theta(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell) + \mathcal{L}_p(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell) + \mathcal{L}_q(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell)$
6 $\mathcal{L}_c(\boldsymbol{x}, \hat{\boldsymbol{y}}) \leftarrow \frac{1}{b}\sum_{\ell\in[b]} \sum_{c\in\mathcal{C}} \lambda_c^k \nu_c(\boldsymbol{x}_\ell, \hat{\boldsymbol{y}}_\ell)$
7 $\omega \leftarrow \omega - \alpha\nabla_\omega(\mathcal{L}_o(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \mathcal{L}_c(\boldsymbol{x}, \hat{\boldsymbol{y}}))$
8 **foreach** $c \in \mathcal{C}$ **do**
9 $\lambda_c^{k+1} \leftarrow \lambda^k + \rho\nu_c(\boldsymbol{x}, \hat{\boldsymbol{y}})$

				$\Delta_{1\%}\, p^d$		$\Delta_{3\%}\, p^d$	
Test Case	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert(\mathcal{X},\mathcal{Y})\rvert$	(%)	MW	(%)	MW
14_ieee	14	40	395806	2.05	5.334	3.15	8.181
30_ieee	30	82	273506	2.47	7.010	3.36	9.533
39_epri	39	92	287390	2.49	156.3	3.42	213.9
57_ieee	57	160	269140	2.65	33.20	3.67	45.92
73_ieee_rts	73	240	373142	2.72	233.2	3.80	324.9
89_pegase	89	420	338132	2.50	204.0	3.53	288.0
118_ieee	118	372	395806	3.03	128.6	3.98	169.1
162_ieee_dtc	162	568	237812	3.10	296.5	4.04	385.9
189_edin	189	412	69342	2.85	39.09	3.72	50.94
300_ieee	300	822	235732	3.25	775.9	4.22	1007.0

Table 1: The Power Networks Adopted as Benchmarks.

by altering the amount of nominal load $\boldsymbol{x} = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ within a range of $\pm 20\%$. The loads are thus sampled from the distributions $\boldsymbol{x}' = (\boldsymbol{p}^{d\prime}, \boldsymbol{q}^{d\prime}) \sim \text{Uniform}(0.8\boldsymbol{x}, 1.2\boldsymbol{x})$. The resulting benchmarks thus have load demands that vary by a factor of up to 20% of their nominal values: Many of them become congested and significantly harder computationally than their original counterparts. A network value that constitutes a dataset entry $(\boldsymbol{x}', \boldsymbol{y}')$ is a feasible OPF solution obtained by solving the AC-OPF problem detailed in Figure 1.

When the learning step exploits an existing hot-start state $\boldsymbol{s}_0$, the training test cases have the property that the total active loads $\|\boldsymbol{p}_0^d\|_1$ in $\boldsymbol{s}_0$ are within 1% and 3% of the total active loads $\|\boldsymbol{p}^d\|_1$. Note that, while the aggregated loads follow this restriction, the individual loads may have far greater variations. Those are illustrated in Table 1 for the 1% ($\Delta_{1\%}p^d$) and 3% ($\Delta_{3\%}p^d$) cases, where the average variations are expressed both in percentage of the total load and in absolute values (MWs). As can be seen, the variations are significant. The table also describes the dataset sizes, including the number of buses and transmission lines of the networks. The data are normalized using the per unit (pu) system so that all quantities are close to 1. The experiments use a $80/20$ train-test split and report results on the test set.

Settings The experiments examine the OPF-DNN models whose features are summarized in Table 2. $\mathcal{M}_{\text{B}}$ refers to the baseline model: It minimizes the loss function $\mathcal{L}_o$ described in Equation (3). $\mathcal{M}_{\text{C}}$ exploits the problem constraints and

Model	$\mathcal{M}_{\text{B}}$	$\mathcal{M}_{\text{C}}$	$\mathcal{M}_{\text{CS}}$	$\mathcal{M}_{\text{CS}}^{\text{D}}$	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$
Exploit **C**onstraints	☐	☑	☑	☑	☑
Exploit hot-start **S**tate	☐	☐	☑	☑	☑
Lagrangian **D**ual Update	☐	☐	☐	☑	☑
Prior State Accuracy	–	–	1%	1%	3%

Table 2: The DNN Models Adopted.

Test Case	Type	$\hat{\boldsymbol{p}}^g$	$\hat{\boldsymbol{v}}$	$\hat{\boldsymbol{\theta}}$	$\tilde{\boldsymbol{p}}^f$
30_ieee	$\mathcal{M}_{\text{B}}$	3.3465	14.699	4.3130	27.213
	$\mathcal{M}_{\text{C}}$	3.1289	2.7346	1.5930	1.6820
	$\mathcal{M}_{\text{CS}}$	0.3052	0.3130	0.0580	0.2030
	$\mathcal{M}_{\text{CS}}^{\text{D}}$	**0.0055**	**0.0070**	**0.0041**	**0.0620**
	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$	0.0030	0.0160	0.0080	0.2120
118_ieee	$\mathcal{M}_{\text{B}}$	0.2150	7.1520	4.2600	38.863
	$\mathcal{M}_{\text{C}}$	0.1810	6.9150	4.6520	6.4730
	$\mathcal{M}_{\text{CS}}$	0.0380	0.1170	1.2750	0.6640
	$\mathcal{M}_{\text{CS}}^{\text{D}}$	**0.0340**	**0.0290**	**0.2070**	**0.4550**
	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$	0.0070	0.0210	0.0590	0.3030
300_ieee	$\mathcal{M}_{\text{B}}$	0.0838	28.025	12.137	125.47
	$\mathcal{M}_{\text{C}}$	0.0914	14.727	7.7450	34.133
	$\mathcal{M}_{\text{CS}}$	0.0174	3.1130	7.2330	26.905
	$\mathcal{M}_{\text{CS}}^{\text{D}}$	**0.0126**	**0.0610**	**2.5670**	**1.1360**
	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$	0.0260	0.2270	0.9980	1.9300

Table 3: Prediction Errors (%).

minimizes the loss: $\mathcal{L}_o + \sum_{c\in\mathcal{C}} \lambda_c \nu_c$, with $\mathcal{L}_o$ defined in Equation (4) and all λ_c set to 1. The suffix S is used for the models that exploit a hot-start state. In particular, $\mathcal{M}_{\text{CS}}$ uses the same loss function as $\mathcal{M}_{\text{C}}$, but it adopts the architecture outlined in Figure 3. Finally, $\mathcal{M}_{\text{CS}}^{\text{D}}$ extends $\mathcal{M}_{\text{CS}}$ by learning the Lagrangian multipliers λ_c using the Lagrangian dual scheme described in Algorithm 1. The latter model is also denoted with OPF-DNN in the paper. The experimental results also used a model (called $\mathcal{M}_{\text{C}}^{\text{D}}$) that extends $\mathcal{M}_{\text{C}}$ by using the Lagrangian dual scheme to adjust the values of the multipliers, and a model (called $\mathcal{M}_{\text{CS}}^{\text{L}}$) that extends $\mathcal{M}_{\text{CS}}$ by learning the values of the multipliers during training. The former performs better, in general, than $\mathcal{M}_{\text{C}}$, but significantly worse than $\mathcal{M}_{\text{CS}}$, and the latter performs similarly to $\mathcal{M}_{\text{CS}}$. All the models that exploit a hot-start state are trained over datasets using states differing by at most 1%, except for model $\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$, which denotes the $\mathcal{M}_{\text{CS}}^{\text{D}}$ model trained with states differs by at most 3%. The reader is referred to (Fioretto, Mak, and Van Hentenryck 2019) for an extensive analysis of the results.

The training uses the Adam optimizer with learning rate ($\alpha = 0.001$) and β values $(0.9, 0.999)$ and was performed for 80 epochs using batch sizes $b = 64$. Finally, the Lagrangian step size ρ is set to 0.01. Additional information are reported in (Fioretto, Mak, and Van Hentenryck 2019).

Prediction Errors

We first analyze the prediction error of the DNN models. Table 3 reports the average L1 distance between the predicted generator power $\hat{\boldsymbol{p}}^g$, voltage magnitude $\hat{\boldsymbol{v}}$ and angles $\hat{\boldsymbol{\theta}}$ and the original quantities. It also reports the errors of the predicted flows $\tilde{\boldsymbol{p}}^f$ (which use the generator power and voltage predictions) and are important to assess the fidelity of the predictions. The results focuses on comparing models $\mathcal{M}_{\text{B}}$ $\mathcal{M}_{\text{C}}$ $\mathcal{M}_{\text{CS}}$ $\mathcal{M}_{\text{CS}}^{\text{D}}$, and report $\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$ for completeness. The distances are reported in percentage: $\frac{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_1}{\|\boldsymbol{x}\|_1} \times 100$, for quantity $\boldsymbol{x}$, and best results are highlighted in bold. For completeness, the results report an extended version of model $\mathcal{M}_{\text{B}}$, which predicts quantities $\boldsymbol{\theta}$ and $\boldsymbol{q}^g$; A clear trend appears: The prediction errors decrease with the increasing of the model complexity. In particular, model $\mathcal{M}_{\text{C}}$, which exploits the problem constraints, predicts much better voltage quantities and power flows than $\mathcal{M}_{\text{B}}$. $\mathcal{M}_{\text{CS}}$, which also exploits a hot-start state, improves $\mathcal{M}_{\text{C}}$ predictions by one order of magnitude in most of the cases. *Finally, the use of the Lagrangian dual to find the best weights ($\mathcal{M}_{CS}^{D}$) further improves $\mathcal{M}_{CS}$ predictions by up to an additional order of magnitude.* The results are consistent for $\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$.

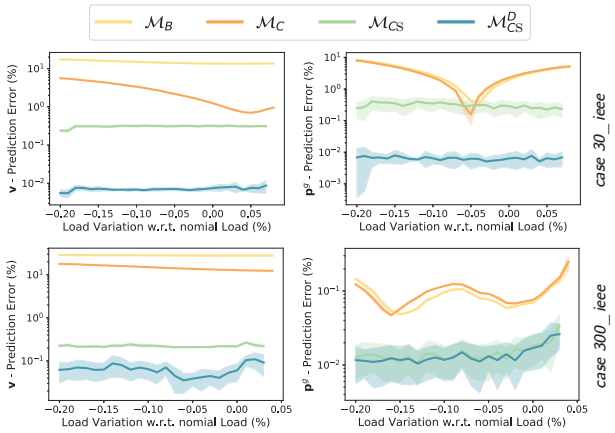

Figure 4: Prediction Errors at varying of the network loads.

Figure 4 further illustrates the importance of modeling the problem constraints and exploiting a hot-start state. It illustrates the prediction errors on the operational parameters $\boldsymbol{v}$ (left) and $\boldsymbol{p}^g$ (right) at the varying of the demands in the power networks. The plots are in log-10 scale and clearly indicate that the models exploiting the problem structure better generalize to the different network settings.

Load Flow Analysis

Having assessed the predictive capabilities of OPF-DNN, the next results focus on evaluating its practicality by simulating the prediction results in an operational environment. The idea is to measure how much the predictions need to be adjusted in order to satisfy the operational and physical constraints. The experiments perform a load flow (Figure 2) on the predicted $\hat{\boldsymbol{p}}^g$ and $\hat{\boldsymbol{v}}$ values. In addition to comparing the DNN model variants, the results also report the deviations of the linear DC model from an AC-feasible solution. The DC model is widely used in power system industry. The results also reports the performance of a baseline load flow model LF_S that finds a feasible solution using the hot-start state $\boldsymbol{s}_0$ as reference point in its objective function. These results highlight the value of learning in OPF-DNN: The reference

Test Case		DC	$\mathcal{M}_{\text{B}}$	$\mathcal{M}_{\text{C}}$	$\mathcal{M}_{\text{CS}}$	$\mathcal{M}_{\text{CS}}^{\text{D}}$	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$	DC	LF$_S$	$\mathcal{M}_{\text{B}}$	$\mathcal{M}_{\text{C}}$	$\mathcal{M}_{\text{CS}}$	$\mathcal{M}_{\text{CS}}^{\text{D}}$	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$
30_ieee	p^g	2.6972	2.0793	1.9688	0.1815	**0.0007**	0.0019	0.1907	13.504	2.1353	1.8268	0.2735	**0.0058**	0.0030
	v	1.2929	83.138	0.4309	0.0944	**0.0037**	0.0019	3.4931	0.4829	6.2996	2.7458	0.4299	**0.0086**	0.0126
118_ieee	p^g	0.2011	0.1071	0.0359	0.0043	**0.0038**	0.0010	0.5865	3.8034	0.1353	0.1557	0.0372	**0.0368**	0.0068
	v	1.9971	3.4391	0.8995	0.0956	**0.0866**	0.0050	2.2780	0.9772	4.5972	6.0326	0.1599	**0.1335**	0.0200
300_ieee	p^g	0.1336	0.0447	0.0339	0.0091	**0.0084**	0.0124	0.1717	14.813	0.0644	0.0766	0.0204	**0.0175**	0.0247
	v	3.8526	31.698	10.292	0.2383	**0.1994**	0.2507	0.6854	1.5737	2.9985	2.1296	1.1553	**0.2196**	0.1575
Total Avg. (%)	p^g	0.7751	0.9843	0.9719	0.0777	**0.0197**	0.0279	0.6090	7.7780	0.5694	0.5307	0.1096	**0.0356**	0.0268
	v	2.4284	36.288	10.334	0.8780	**0.1995**	0.1345	1.7870	0.8213	3.3879	2.4985	0.9429	**0.2136**	0.1230

Table 4: Average distances (%) for the active power (top rows) and voltage magnitude (bottom rows) of the Load Flow solutions w.r.t. the corresponding predictions (left table) and w.r.t. the AC-OPF solutions (right table).

Test Case	DC	LF$_S$	$\mathcal{M}_{\text{B}}$	$\mathcal{M}_{\text{C}}$	$\mathcal{M}_{\text{CS}}$	$\mathcal{M}_{\text{CS}}^{\text{D}}$	$\mathcal{M}_{\text{CS}}^{\text{D(3\%)}}$
14_ieee	5.1792	4.5246	0.7562	0.6290	0.2614	**0.0007**	0.0001
30_ieee	7.9894	8.2411	2.9447	2.1316	0.5433	**0.0180**	0.0078
39_epri	0.9094	2.2869	0.1901	0.0752	0.0537	**0.0003**	0.0027
57_ieee	1.7758	3.8445	1.1115	1.0609	0.2025	**0.0527**	0.0001
73_ieee_rts	2.6846	1.4581	9.4364	3.2399	0.5143	**0.4586**	0.0356
89_pegase	1.5089	2.6287	0.3284	0.3274	0.3347	**0.1494**	0.0237
118_ieee	4.7455	1.0389	1.0973	1.1897	**0.5300**	0.5408	0.1620
162_ieee_dtc	6.2090	4.2094	0.5021	0.8360	0.3162	**0.2845**	0.1535
189_edin	9.9803	7.5561	5.3851	2.7770	0.7135	**0.3177**	0.3500
300_ieee	4.7508	6.6394	1.9543	1.1576	0.3233	**0.3011**	0.6226
Total Avg. (%)	4.5733	4.2428	2.3706	1.3424	0.3793	**0.2124**	0.1358

Table 5: Load Flow vs. AC-OPF cost distances (%).

point alone is not sufficient to find high quality solutions.

The results are tabulated in Table 4. The left table reports the L1 distances, in percentage, of the predictions $\hat{\boldsymbol{p}}^g$ and $\hat{\boldsymbol{v}}$ to the solutions $\boldsymbol{p}^g$ and $\boldsymbol{v}$ of the load flows. Trends similar to the previous section are observed, with $\mathcal{M}_{\text{CS}}^{\text{D}}$ being substantially more accurate than all other DNN versions. The table also shows that $\mathcal{M}_{\text{CS}}^{\text{D}}$ is up to two orders of magnitude more precise than the DC model. The right table reports the L1 distances of the load flow solutions to the optimal AC-OPF solutions. The results follow a similar trend, with the OPF-DNN model ($\mathcal{M}_{\text{CS}}^{\text{D}}$) being at least one order of magnitude more precise than the DC model and the baseline LF$_S$ model. The bottom rows of the table show the average results over all the power network adopted in the experimental analysis. Note that the very high accuracy of OPF-DNN may render the use of a load flow optimization, to restore feasibility, unnecessary. *These results are significant: They suggest that OPF-DNN has the potential to replace the DC model as an AC-OPF approximation and deliver generator setpoints with greater fidelity.*

Solution Quality and Runtime

Finally, the last results compare the accuracy and runtime of the proposed DNN models, the DC approximation, and the load flow baseline LF$_S$, against the *optimal* AC-OPF solutions. The solution quality is measured by first finding the closest AC feasible solution to the predictions returned by the DC or by the DNN models. Then, the cost of the dispatches are compared to the original ones. Table 5 reports the average L1-distances of the dispatch costs. The last row reports the average distances across all the test cases. The analysis of the DNN variants exhibits the same trends as before, with the networks progressively improving the results as they exploit the problem constraints ($\mathcal{M}_{\text{C}}$), a hot-start state ($\mathcal{M}_{\text{CS}}$), and use the Lagrangian dual ($\mathcal{M}_{\text{CS}}^{\text{D}}$).

Finally, Table 6 illustrates the average time required to find an AC OPF solution, the AC load flow with a reference solution, a linear DC approximation, and a prediction using OPF-DNN ($\mathcal{M}_{\text{CS}}^{\text{D}}$) on the test dataset. Recall that the dataset adopted uses a load stress value of up to 20% of the nominal loads and hence the test cases are often much more challenging than their original counterparts. The last row of the table reports the average speedup of the models compared to the AC OPF. *Observe that OPF-DNN finds dispatches whose costs are at least one order of magnitude closer to the AC solution than those returned by the DC approximation, while being several order of magnitude faster.*

Related Work

Within the energy research landscape, DNN architectures have mainly been adopted to predict exogenous factors affecting renewable resources, such as solar or wind. For instance, Anwar et al. 2016 uses a DNN-based system to predict wind speed and adopt the predictions to schedule generation units ahead of the trading period, and Boukelia et al. 2017 studied a DDN framework to predict the electricity costs of solar power plants coupled with a fuel backup system and energy storage.

Another power system area in which DNNs have been adopted is that of *security assessment*: Arteaga et al. (2019) proposed a convolutional neural network to identify safe vs. unsafe operating points to reduce the risks of a blackout. Donnot et al. (2019) use a ResNet architecture to predict the effect of interventions that reconnect disconnected transmission lines in a power network.

In terms of OPF prediction, the literature is much sparser. The most relevant work uses a DNN architecture to learn the set of active constraints (e.g., those that, if removed, would improve the value of the objective function) at optimality in the linear DC model (Ng et al. 2018; Deka and Misra 2019). Once the set of relevant active constraints are identified, exploiting the fact that the DC OPF is a linear program, one can run an exhaustive search to find a solution that satisfies the active constraints. While this strategy is efficient when the number of active constraints is small, its computational efficiently decreases drastically when its number increases due to the combinatoric nature of the problem. Additionally, this strategy applies only to the linear DC approximation.

This work departs from these proposals and predicts the

Test Case	AC	LF_S	DC	OPF-DNN
14_ieee	0.0332	0.0430	0.0075	0.0000
30_ieee	0.1023	0.0755	0.0148	0.0000
39_epri	0.2169	0.0968	0.0232	0.0000
57_ieee	0.3288	0.1394	0.0359	0.0000
73_ieee_rts	0.3081	0.2979	0.0496	0.0000
89_pegase	1.4503	0.6014	0.0601	0.0000
118_ieee	0.4207	0.7819	0.0785	0.0001
162_ieee_dtc	1.8909	0.7393	0.2016	0.0000
189_edin	4.0081	0.4490	0.0865	0.0001
300_ieee	8.0645	1.4850	0.2662	0.0001
Avg speedup	1x	2.76x	15.2x	$>10^4$x

Table 6: Average runtime in seconds.

optimal setpoints for the network generators and bus voltages in the AC-OPF setting. Crucially, the presented model actively exploits the OPF constraints during training, producing reliable results that significantly outperform classical model approximations (e.g., DC-OPF). This work also provides a compelling alternative to real-time OPF tracking (Tang, Dvijotham, and Low 2017; Liu et al. 2018): OPF-DNN always converges instantly with very high accuracy and can be applied to a wider class of applications.

Conclusions

The paper studied a DNN approach for predicting the generators setpoint in optimal power flows. The AC-OPF problem is a non-convex non-linear optimization problem that is subject to a set of constraints dictated by the physics of power networks and engineering practices. The proposed OPF-DNN model exploits the problem constraints using a Lagrangian dual method as well as a related hot-start state. The resulting model was tested on several power network test cases of varying sizes in terms of prediction accuracy, operational feasibility, and solution quality. The computational results show that the proposed OPF-DNN model can find solutions that are up to several order of magnitude more precise and faster than existing approximation methods (e.g., the commonly adopted linear DC model). These results may open a new avenue in approximating the AC-OPF problem, a key building block in many power system applications, including expansion planning and security assessment studies which typically requires a huge number of multi-year simulations based on the linear DC model. Current work aims at improving the (currently naive) implementation to test the approach on very large networks whose entire data sets are significantly larger than the GPU memory.

Acknowledgments This research is partly supported by NSF Grant 1709094.

References

Anwar, M. B.; El Moursi, M. S.; and Xiao, W. 2016. Novel power smoothing and generation scheduling strategies for a hybrid wind and marine current turbine system. *IEEE Transactions on Power Systems* 32(2):1315–1326.

Arteaga, J. H.; Hancharou, F.; Thams, F.; and Chatzivasileiadis, S. 2019. Deep learning for power system security assessment. In *2019 IEEE Milan PowerTech*.

Baran, M. E., and Wu, F. F. 1989. Optimal capacitor placement on radial distribution systems. *IEEE Transactions on Power Delivery* 4(1):725–734.

Boukelia, T.; Arslan, O.; and Mecibah, M. 2017. Potential assessment of a parabolic trough solar thermal power plant considering hourly analysis: ANN-based approach. *Renewable Energy* 105:324 – 333.

Chowdhury, B. H., and Rahman, S. 1990. A review of recent advances in economic dispatch. *IEEE Transactions on Power Systems* 5(4):1248–1259.

Coffrin, C.; Gordon, D.; and Scott, P. 2014. NESTA, the NICTA energy system test case archive. *CoRR* abs/1411.0359.

Deka, D., and Misra, S. 2019. Learning for DC-OPF: Classifying active sets using neural nets. In *2019 IEEE Milan PowerTech*.

Deutche-Energue-Agentur. 2019. The e-highway2050 project. http://www.e-highway2050.eu. Accessed: 2019-11-19.

Donnot, B.; Donon, B.; Guyon, I.; Liu, Z.; Marot, A.; Panciatici, P.; and Schoenauer, M. 2019. LEAP nets for power grid perturbations. In *European Symposium on Artificial Neural Networks*.

Fioretto, F.; Mak, T. W. K.; and Van Hentenryck, P. 2019. Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods. *CoRR* abs/1909.10461.

Fisher, E. B.; O'Neill, R. P.; and Ferris, M. C. 2008. Optimal transmission switching. *IEEE Transactions on Power Systems* 23(3):1346–1355.

Fontaine, D.; Laurent, M.; and Van Hentenryck, P. 2014. Constraint-based lagrangian relaxation. In *Principles and Practice of Constraint Programming*, 324–339.

Hestenes, M. R. 1969. Multiplier and gradient methods. *Journal of optimization theory and applications* 4(5):303–320.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521:436–444.

Liu, J.; Marecek, J.; Simonetta, A.; and Takač, M. 2018. A coordinate-descent algorithm for tracking solutions in time-varying optimal power flows. In *Power Systems Computation Conference*.

Monticelli, A.; Pereira, M.; and Granville, S. 1987. Security-constrained optimal power flow with post-contingency corrective rescheduling. *IEEE Transactions on Power Systems* 2(1):175–180.

Ng, Y.; Misra, S.; Roald, L.; and Backhaus, S. 2018. Statistical learning for DC optimal power flow. In *Power Systems Computation Conference*.

Niharika; Verma, S.; and Mukherjee, V. 2016. Transmission expansion planning: A review. In *International Conference on Energy Efficient Technologies for Sustainability*, 350–355.

Pache, C.; Maeght, J.; Seguinot, B.; Zani, A.; Lumbreras, S.; Ramos, A.; Agapoff, S.; Warland, L.; Rouco, L.; and Panciatici, P. 2015. Enhanced pan-european transmission planning methodology. In *IEEE Power Energy Society General Meeting*.

Tang, Y.; Dvijotham, K.; and Low, S. 2017. Real-time optimal power flow. *IEEE Transactions on Smart Grid* 8(6):2963–2973.

Tong, J., and Ni, H. 2011. Look-ahead multi-time frame generator control and dispatch method in PJM real time operations. In *IEEE Power and Energy Society General Meeting*.

Wang, J.; Shahidehpour, M.; and Li, Z. 2008. Security-constrained unit commitment with volatile wind power generation. *IEEE Transactions on Power Systems* 23(3):1319–1327.

Wood, A. J., and Wollenberg, B. F. 1996. *Power Generation, Operation, and Control*. Wiley-Interscience.