SAND2022-2656C

Deep Neural Networks as Surrogates for Intractable Constraints and Problem Dimension Reduction: SC AC-OPF

AIChE Annual Meeting

Zachary Kilwein, Jordan Jalving, Logan Blakely, Mike Eydenberg, Carl Laird, Fani Boukouvala

November 10, 2021

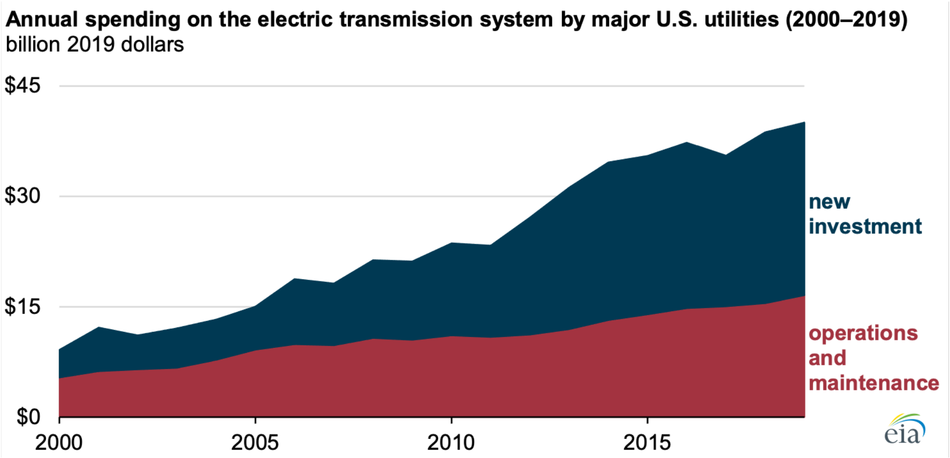
Georgia Tech School of Chemical & Biomolecular Engineering

Sandia National Laboratories

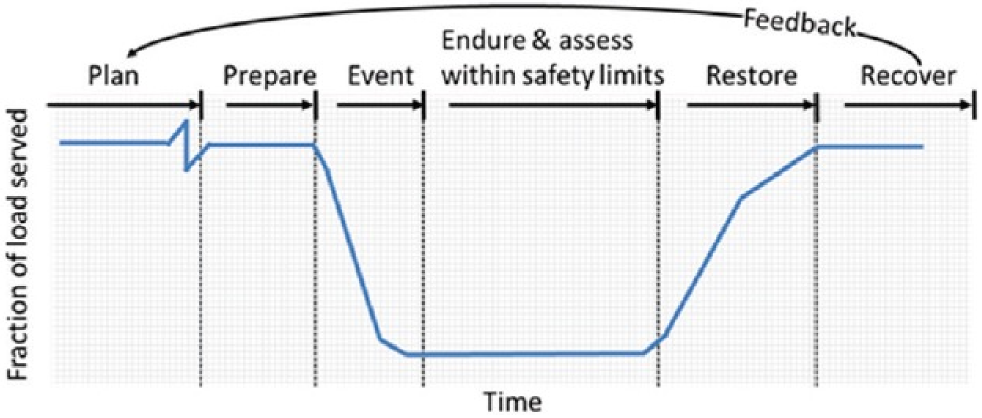
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Problem Motivation: Balancing Grid Security with Economic Dispatch

- Utilities spent $16.6 billion on systems operations and maintenance in 2019 [1]
- FERC predicts effective optimization could save tens of billions annually [2]
- Grids operated close to their optimal points pose risk of large-scale outage in case of contingency [3]
- Important to weigh **optimality** and **security**
- Contingency events may take a long time to rectify



Source: US Energy Information Administration

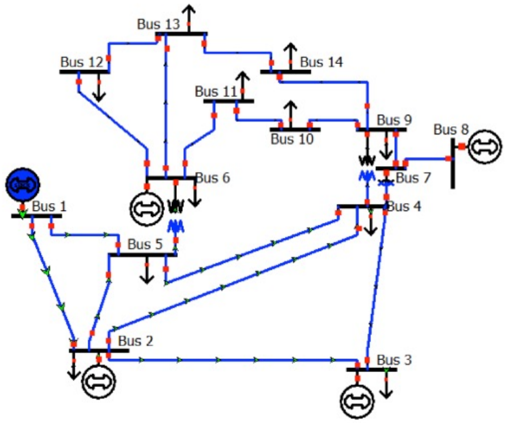


Source: Enhancing the Resilience of the Nations Electricity System (2017)

[1] US Energy Information Administration
[2] Cain et al 2012 FERC
[3] Capitanescu et al 2011 Electric Power Systems Research

Power Flow Model Description

- Optimal Power Flow (OPF) programs balance load demands with generator setpoints in the most economic way [4]:
  - AC-OPF: non-linear, non-convex
  - Convex relaxations of AC-OPF
  - DC-OPF: linearized version
  - Global [5] vs local methods
- Security constrained OPF used to optimize to more secure operating points [6]:
  - *N-1* security signifies an operating point that can handle any single outage in the system
  - Extremely computationally expensive, exponentially increases variable space



Source: IEEE PG-Lib OPF Case 14

$$Min\ C(p^g) \quad (1)$$

$$\Sigma_{k(n,m)} p^f_{k(n,m)} + \Sigma_{k(n,m)} p^t_{k(n,m)} - p^g_n + p^d_n = 0 \quad (2)$$

$$\Sigma_{k(n,m)} q^f_{k(n,m)} + \Sigma_{k(n,m)} q^t_{k(n,m)} - q^g_n + q^d_n = 0 \quad (3)$$

$$V^{min} \leq V \leq V^{max} \quad (4)$$

$$\theta^{min}_{nm} \leq \theta_n - \theta_m \leq \theta^{max}_{nm} \quad (5)$$

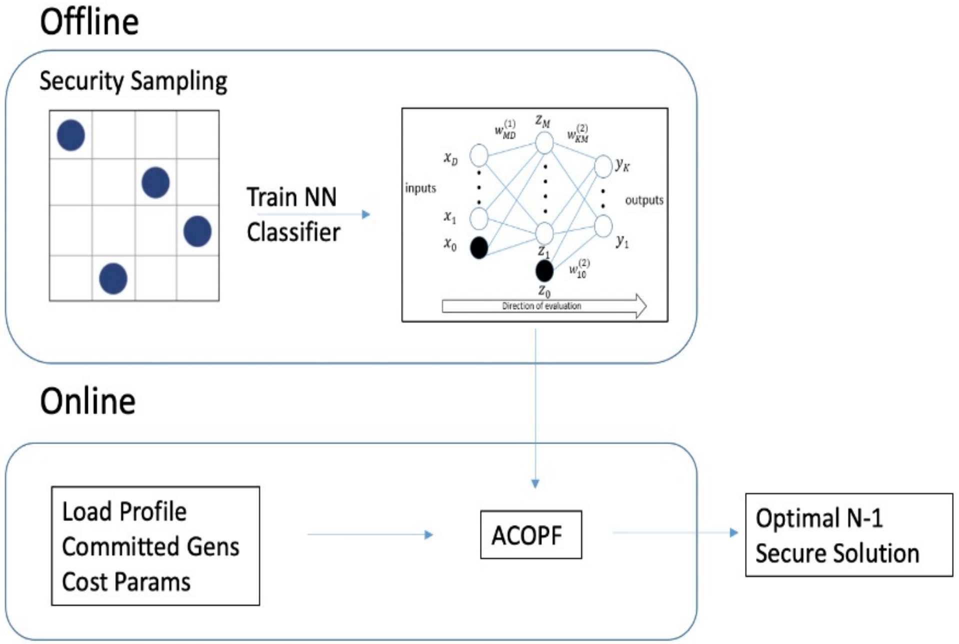
$$P^{min}_n < P_n < P^{max}_n \quad (6)$$

$$Q^{min}_n < Q_n < Q^{max}_n \quad (7)$$

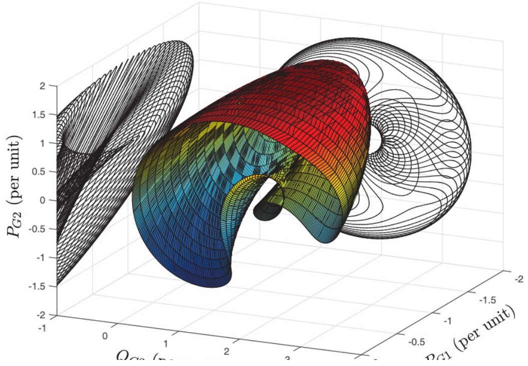
[4] Molzahn et al 2019 Foundations and Trends in Electric Energy Systems
[5] Liu et al 2018 IEEE Transactions on Power Systems
[6] Capitanescu et al 2011 Electric Power Systems Research

Big Picture: Using ML Surrogates to Embed Intractable Constraints into Nonlinear Programs (NLP)

- Embedding deep NN into large scale NLP problems
- NN embed complex security function
- Tractable solution time for online economic dispatch
- Can be used to boost security in large-scale grid problems
- Integration with familiar optimization (Pyomo) and power flow (Egret) software



Schematic of SC-OPF Model Solution



Visual Representation of Feasible Space for Small Grid (Source: Hiskens et al 2001 IEEE Transactions on Power Systems)

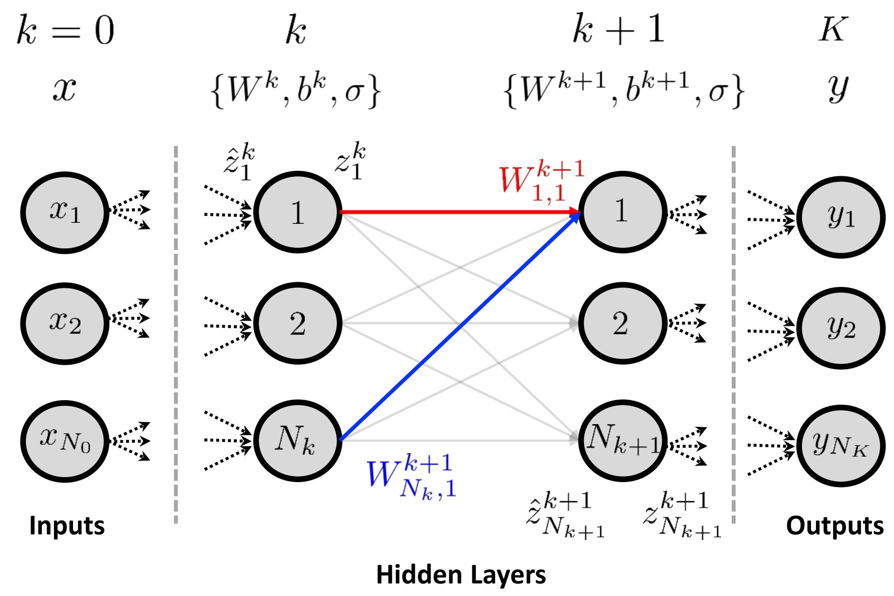
Literature Review

- Hybrid models have been used in PSE for many applications:
  - Parallel mechanistic and parametric models for improving predictions in activated sludge process [14]
  - Series NN and kinetic model for lignocellulosic fermentation [15]
- Black-box feasibility functions for Kriging, NN models [16-17]
  - Pharmaceutical, process models (low dimensionality)
- Linear regression models trained to map security boundary for given load profile[18]
  - Smaller case studies, not generalizable
- ReLU network as security constraint for MINLP and MILP approximation [19]
  - Integer variables, power flow equation approximation, IEEE Case 14
- Fit NN to full optimization solution instead of secure space [20]
  - Not flexible to changes in costing function, network parameters

[14] Lee et al 2002 Biotechnology and bioengineering
[15] Lopez et al 2020 Biofuels, Bioproducts and Biorefining
[16] Wang et al 2017 AIChE Journal
[17] Boukouvala et al 2012 Comp. & ChE
[18] Martinez et al 2010 IEEE Transactions on Power Systems
[19] Venzke et al 2020 Under Review
[20] Velloso et al 2020 Archive

Methodology: Formulating a NN into NLP (Full Space)

- Inputs *x*, fully connected sequential layers, outputs *y*
- Direct encoding of NN model into intermediate variables *z*
- Here activation functions are explicitly encoded into variables/constraints in optimization model



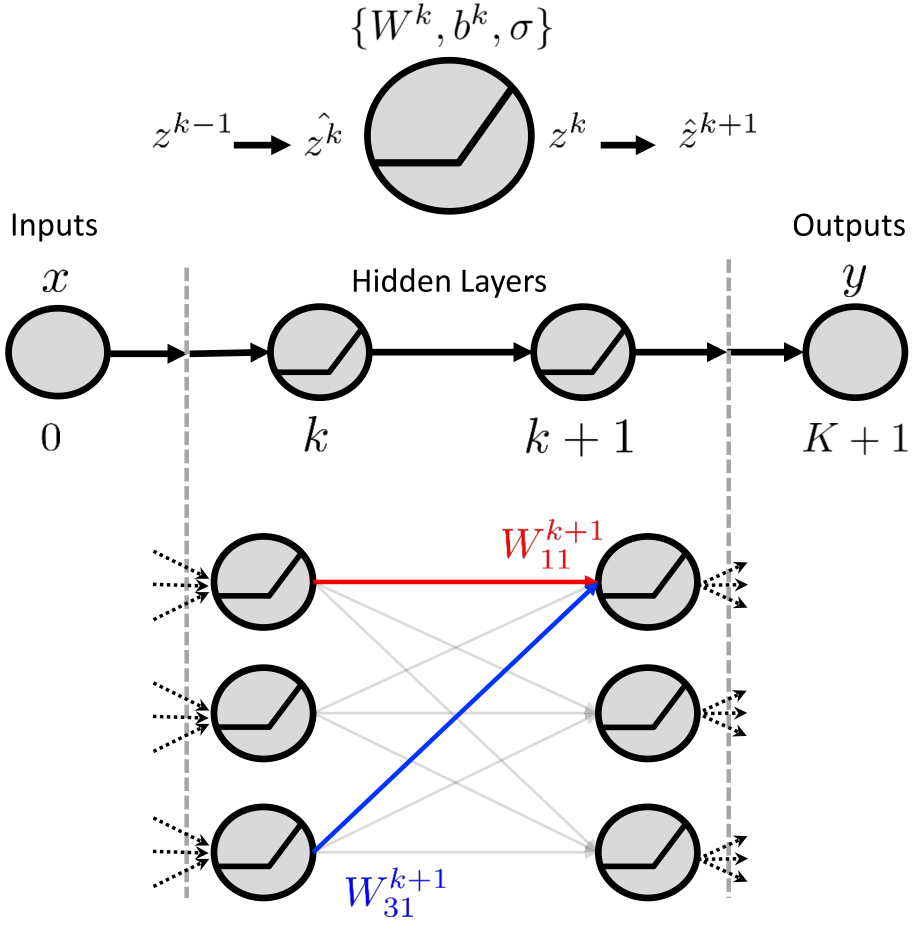
Full Space Representation of NN Model

$z_n^0 = x_n \longrightarrow \hat{z}_n^k = f(W, b, z_n^{k-1}) \longrightarrow z_n^k = \sigma(\hat{z}_n^k) \longrightarrow y_n = f(W, b, z_n^k)$

Methodology: Formulating a NN into NLP (Reduced Space)

- Inputs *x* map directly to outputs *y* with a single constraint
- Internals of NN not treated as variables
- Has shown to produce superior relaxations in global optimization [ref]
- Reduces overall model size
- Internal variables less important after training

Reduced Space Formulation
$\min_{x,y} f(x,y)$ s.t. $y = NN(x)$ $g(x,y) \leq 0$ $x \in X,\ y \in Y$



Reduced Space Representation of NN Model

Georgia Tech

Methodology: Formulating ReLU NNs (MILP, Complementarity)

- ReLU NNs are increasingly common in ML literature [ref]
- Can be represented as MILP
- Not advantageous for hybrid SC-OPF as underlying physics still non-linear (MINLP)
- Complementarity formulation relaxes model to eliminate integer variables

ReLU is piecewise linear:

$$z = \max(\hat{z}, 0)$$

MILP can give bounds:

$$0 \leq (z_n^k - \hat{z}_n^k) \perp z_n^k$$

Complimentarity:

$$\left(z_n^k - \hat{z}_n^k\right) z_n^k = \varepsilon \qquad \forall n \in N, \forall k \in K$$

$$\varepsilon \geq 0$$

Georgia Tech

OptML: Bridging Pyomo Optimization Tools and ML Libraries

```
import pyomo.environ as pyo
from tensorflow import keras

from optml import OptMLBlock
from optml.neuralnet import FullSpaceContinuousFormulation, ReducedSpaceContinuousFormulation
from optml.neuralnet import ReLUBigMFormulation
from optml.neuralnet import load_keras_sequential

neural_net = keras.models.load_model('Saved NN Model')
net=load_keras_sequential(neural_net)
formulation = ReducedSpaceContinuousFormulation(net)

m=pyo.ConcreteModel()

m.nn = OptMLBlock()
m.nn.y=pyo.Var(m.nn.outputindex)
m.nn.build_formulation(formulation, input_vars=[inputs], output_vars=m.nn.y)
```

- Simple way to load tensorflow based NNs into pyomo
- Tools for reduced space, full space, MILP, complimentarity
- Block structure to add combine with existing pyomo models

https://github.com/or-fusion/OptML/tree/main

Methodology: Model Guided Sampling

Sampling Algorithm for Boundary Points

Initial_Loads=[LF^{min},…,LF^{max}], Load_Directions=[$\overrightarrow{LD}^{0}, \ldots, \overrightarrow{LD}^{j}$]
for IL **in** Initial_Loads:
    for LD **in** Load_Dirs:

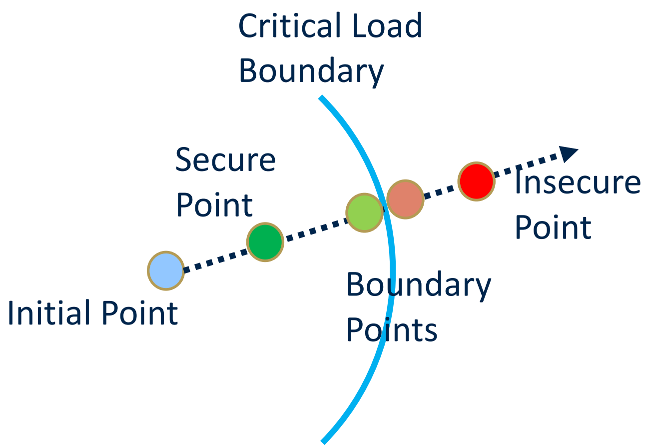
$$
\begin{aligned}
&\max(SF) \\
&s.t.\ g_0(x_0, u_0) = 0 \\
&\quad g_c(x_c, u_c) = 0 \\
&\quad x_L \leq x_0 \leq x_U, u_L \leq u_0 \leq u_U \\
&\quad x_L \leq x_c \leq x_U, u_L \leq u_c \leq u_U \\
&\quad p_l = IL * LD * SF \\
&\quad \frac{p_l}{\sqrt{p_l^2 + q_l^2}} = Power\ Factor
\end{aligned}
$$

Existing sampling approaches include use of hyperspheres [21]

LHS sampling of initial load conditions and ramping vectors

Optimization formulation to find max scaling factor before security constraint is violated

- Algorithm can be parallelized (single contingencies independent)
- Point where security constraint becomes active is much more informative to NN classifier

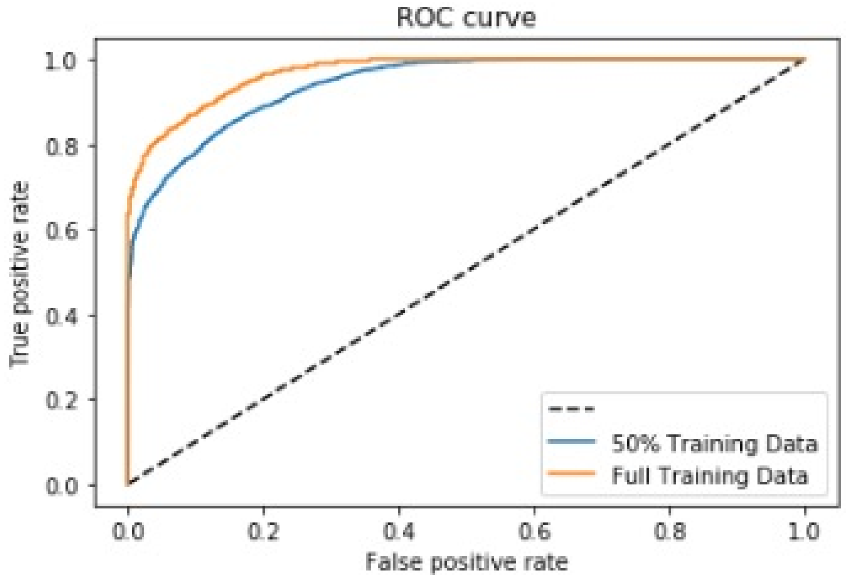


Sampling Strategy Visualization

[21] Venzke et al 2021 Electric Power Systems Research

Classifier Accuracy Results

- ROC explains trade-off between conservativeness and accuracy (can be tuned in optimization problem: softmax output)
- Sequential Model: 118 inputs, 2 hidden layers, 20 nodes
- **89% accuracy** for test points on IEEE Case 118
- ReLU and tanh networks are able to achieve identical accuracy



Receiver Operating Curve (ROC) for IEEE Case 118 Security Classifier

Georgia Tech

Methodology: NN SC-OPF Reformulation

- Incorporating security constraints makes problem very large
- IEEE Case 30: 450 -> 18,400 variables
- Classification of security in original variable space
- Secure space is encoded much more efficiently
- Expensive simulations can be done offline so that online solution is fast

Extensive Formulation

$$\min_{x_0,u_0,x_c,u_c} f(u_0) + \rho \sum_{c \epsilon C} f_c(u_c, u_0)$$
$$g_0(x_0, u_0) = 0$$
$$g_c(x_c, u_c) = 0 \quad \forall c \in C$$
$$x_L \leq x_0 \leq x_U, u_L \leq u_0 \leq u_U$$
$$x_L \leq x_c \leq x_U, u_L \leq u_c \leq u_U \quad \forall c \in C$$

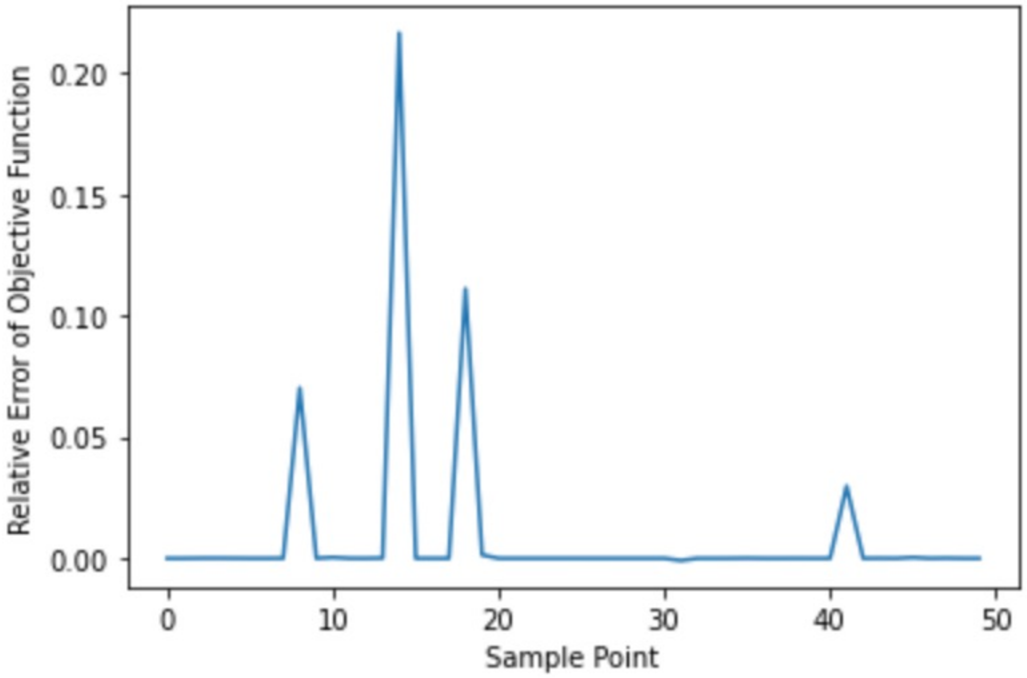
Hybrid Reformulation

$$\min_{x,u} f(u)$$
$$g(x, u) = 0$$
$$x_L \leq x \leq x_U, u_L \leq u \leq u_U$$
$$y_{sec} = NN(x, u)$$
$$y_{sec} \leq \alpha$$

Extensive vs Hybrid Comparison {Case 118}

- Obj-> Average: 0.85% optimality gap or relative error between objective function
  - Nearly identical obj fxn values for most cases
- CPU: 5.27 sec for extensive
- CPU: 0.32 sec for hybrid model (tanh)
- CPU: 0.35 sec for ReLU-complementarity

Model mismatch-> 0.615% average error in real power variables

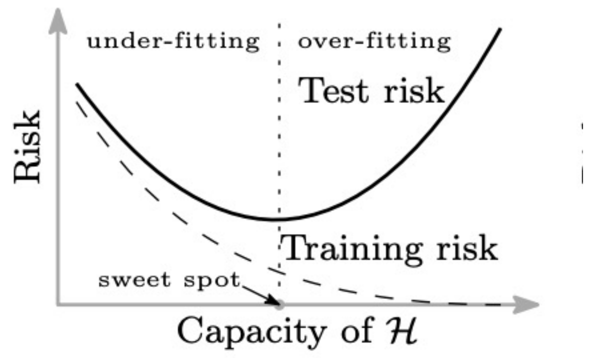


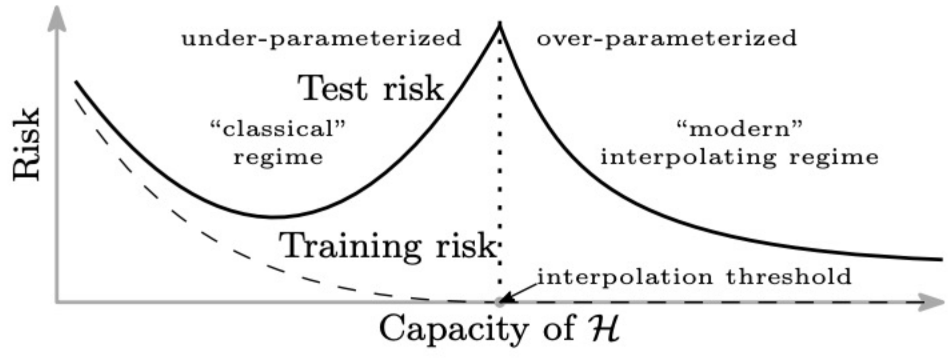
Conclusions

- Modern grid systems require intensive **optimization**
- Enforcing **grid security** can be mathematically challenging
- Neural Network classifiers can be formulated naturally within NLP OPF problems
  - ReLU, Tanh
  - Reduced, Full Space
- High accuracy conferred through feasibility sampling
- Balance of computation and accuracy

Future Work and Anticipated Challenges

- Larger case studies than Case 118
  - Parallelization of sampling algorithm
  - Input feature selection for NN classifier
  - Selection of most prescient contingency events
- NN sparsification, parameterization [23]:
  - Larger NN classifiers will be harder to optimize
  - Sparsification can greatly speed up NLP solution
  - Lottery ticket hypothesis (90% reduction possible with same accuracy)
- Further OptML tools:
  - New ML models
  - NN verification case studies





Non–intuitive generalization of over-parameterized deep ML models (Source: Reconciling Modern Machine Learning Practice and the bias-variance trade-off)

[23] Frankle et al 2018 Archived

Acknowledgements

DD-PSE Group at Georgia Tech:

Jackie Zhai, Sophie Kim, William Bradley, Jinhyeun Kim, Elisavet Anglou, Suryateja Ravutla

Redly Team at Sandia National Labs

Funding Source:

Sandia Grant GR00002344

Sandia National Laboratories

Georgia Tech

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Thank you!

Questions?

Supplementary Slides

AC Power Flow Equations

$$\sum_{k(n,m)\epsilon K_n^{out}} p_{k(n,m)}^{f} + \sum_{k(n,m)\epsilon K_n^{in}} p_{k(n,m)}^{t} - p_n^g + p_n^d = 0,\ \forall n \epsilon N \qquad (1)$$

$$\sum_{k(n,m)\epsilon K_n^{out}} q_{k(n,m)}^{f} + \sum_{k(n,m)\epsilon K_n^{in}} q_{k(n,m)}^{t} - q_n^g + q_n^d = 0,\ \forall n \epsilon N \qquad (2)$$

$$p_{k(n,m)}^{f} = g_k v_n^2 - v_n v_m \left(g_k cos\theta_{n,m} + b_k sin\theta_{n,m}\right),\ \forall k \epsilon K \qquad (3)$$

$$p_{k(n,m)}^{t} = g_k v_m^2 - v_n v_m \left(g_k cos\theta_{n,m} - b_k sin\theta_{n,m}\right),\ \forall k \epsilon K \qquad (4)$$

$$q_{k(n,m)}^{f} = -(b_k + b_k^{sh}) v_n^2 - v_n v_m \left(g_k sin\theta_{n,m} - b_k cos\theta_{n,m}\right),\ \forall k \epsilon K \qquad (5)$$

$$q_{k(n,m)}^{t} = -\left(b_k + b_k^{sh}\right) v_m^2 + v_n v_m \left(g_k sin\theta_{n,m} + b_k cos\theta_{n,m}\right),\ \forall k \epsilon K \qquad (6)$$

$$v_n^{min} \le v_n \le v_n^{max},\ \forall n \epsilon N \qquad (7)$$

$$\theta_{nm}^{min} \le \theta_n - \theta_m \le \theta_{nm}^{max},\ \forall \{n,m\} \epsilon K \qquad (8)$$

$$p_n^{min} \le p_n \le p_n^{max},\ \forall n \epsilon N \qquad (9)$$

$$q_n^{min} \le q_n \le q_n^{max},\ \forall n \epsilon N \qquad (10)$$

Georgia Tech

AC Optimal Power Flow Formulation

$$Min\ C(p^g) \quad (1)$$

$$s.t. \sum_{k(n,m)\epsilon K_n^{out}} p_{k(n,m)}^f + \sum_{k(n,m)\epsilon K_n^{in}} p_{k(n,m)}^t - p_n^g + p_n^d = 0,\ \forall n \epsilon N \quad (2)$$

$$\sum_{k(n,m)\epsilon K_n^{out}} q_{k(n,m)}^f + \sum_{k(n,m)\epsilon K_n^{in}} q_{k(n,m)}^t - q_n^g + q_n^d = 0,\ \forall n \epsilon N \quad (3)$$

$$p_{k(n,m)}^f = g_k v_n^2 - v_n v_m \left(g_k cos\theta_{n,m} + b_k sin\theta_{n,m}\right),\ \forall k \epsilon K \quad (4)$$

$$p_{k(n,m)}^t = g_k v_m^2 - v_n v_m \left(g_k cos\theta_{n,m} - b_k sin\theta_{n,m}\right),\ \forall k \epsilon K \quad (5)$$

$$q_{k(n,m)}^f = -(b_k + b_k^{sh}) v_n^2 - v_n v_m \left(g_k sin\theta_{n,m} - b_k cos\theta_{n,m}\right),\ \forall k \epsilon K \quad (6)$$

$$q_{k(n,m)}^t = -\left(b_k + b_k^{sh}\right) v_m^2 + v_n v_m \left(g_k sin\theta_{n,m} + b_k cos\theta_{n,m}\right),\ \forall k \epsilon K \quad (7)$$

$$v_n^{min} \leq v_n \leq v_n^{max},\ \forall n \epsilon N \quad (8)$$

$$\theta_{nm}^{min} \leq \theta_n - \theta_m \leq \theta_{nm}^{max},\ \forall \{n, m\} \epsilon K \quad (9)$$

$$p_n^{min} \leq p_n \leq p_n^{max},\ \forall n \epsilon N \quad (10)$$

$$q_n^{min} \leq q_n \leq q_n^{max},\ \forall n \epsilon N \quad (11)$$

$$(p_k^f + q_k^f)^2 \leq (Thermal\ Limit)^2,\ \forall k \epsilon K \quad (12)$$

$$(p_k^t + q_k^t)^2 \leq (Thermal\ Limit)^2,\ \forall k \epsilon K \quad (13)$$

Security Constrained AC Optimal Power Flow

$$Min\ C(p^g) \qquad (1)$$

$$s.t. \sum_{k(n,m)\epsilon K_n^{out}} p_{k(n,m)}^f + \sum_{k(n,m)\epsilon K_n^{in}} p_{k(n,m)}^t - p_n^g + p_n^d = 0,\ \forall n\ \epsilon\ N, \forall c\ \epsilon\ C \qquad (2)$$

$$\sum_{k(n,m)\epsilon K_n^{out}} q_{k(n,m)}^f + \sum_{k(n,m)\epsilon K_n^{in}} q_{k(n,m)}^t - q_n^g + q_n^d = 0,\ \forall n\ \epsilon\ N, \forall c\ \epsilon\ C \qquad (3)$$

$$p_{k(n,m)}^f = g_k v_n^2 - v_n v_m \left(g_k cos\theta_{n,m} + b_k sin\theta_{n,m}\right),\ \forall k\ \epsilon\ K, \forall c\ \epsilon\ C \qquad (4)$$

$$p_{k(n,m)}^t = g_k v_m^2 - v_n v_m \left(g_k cos\theta_{n,m} - b_k sin\theta_{n,m}\right),\ \forall k\ \epsilon\ K, \forall c\ \epsilon\ C \qquad (5)$$

$$q_{k(n,m)}^f = -\left(b_k + b_k^{sh}\right) v_n^2 - v_n v_m \left(g_k sin\theta_{n,m} - b_k cos\theta_{n,m}\right),\ \forall k\ \epsilon\ K, \forall c\ \epsilon\ C \qquad (6)$$

$$q_{k(n,m)}^t = -\left(b_k + b_k^{sh}\right) v_m^2 + v_n v_m \left(g_k sin\theta_{n,m} + b_k cos\theta_{n,m}\right),\ \forall k\ \epsilon\ K, \forall c\ \epsilon\ C \qquad (7)$$

$$v_n^{min} \le v_n \le v_n^{max},\ \forall n \in N, \forall c\ \epsilon\ C \qquad (8)$$

$$\theta_{nm}^{min} \le \theta_n - \theta_m \le \theta_{nm}^{max},\ \forall \{n, m\}\ \epsilon\ K, \forall c\ \epsilon\ C \qquad (9)$$

$$p_n^{min} \le p_n \le p_n^{max},\ \forall n\ \epsilon\ N,\ \forall c\ \epsilon\ C \qquad (10)$$

$$q_n^{min} \le q_n \le q_n^{max},\ \forall n\ \epsilon\ N,\ \forall c\ \epsilon\ C \qquad (11)$$

$$(p_k^f + q_k^f)^2 \le (Thermal\ Limit)^2,\ \forall k\ \epsilon\ K,\ \forall c\ \epsilon\ C \qquad (12)$$

$$(p_k^t + q_k^t)^2 \le (Thermal\ Limit)^2,\ \forall k\ \epsilon\ K,\ \forall c\ \epsilon\ C \qquad (13)$$

Two Successful Modeling Paradigms Used for Engineering Applications

First Principles

Machine Learning

- Conservation Balances
- Navier-Stokes
- Maxwell's Equations

- Kinetic models
- Thermo Eq. of State
- AC Power Flow

- Process Models
- Unit Ops

- Heuristic empirical models

- NN
- Gaussian Process

- First-principle (FP) models are difficult to develop, expensive to solve, but **generalize well** and are interpretable [1]
- Machine learning (ML) models are easy to develop, **fast to solve**, but generalize poorly and are not easily interpreted
- Creating models with both elements allows us to solve new problems, improve speed and accuracy of engineering model predictions [2]

Georgia Tech

[1] Grossman et al. 2000 AIChE Journal
[2] Venkatasubramanian 2019 AiChE Journal

NN SC-OPF Comparison Results

- Literature study for extensive form SC-ACOPF takes ~**400 sec** [22]
- NN-SC-ACOPF **averages 7 sec** and never exceeds 15 sec

Formulation	Avg Opt Gap	CPU (s)	% *N-1* Insecure
SCOPF Extensive			
NN SCOPF			

Georgia Tech

[22] Kang 2015 PhD Thesis Texas A&M University