Machine learning for power systems: present and future

Presentation by Spyros Chatzivasileiadis
Associate Professor, DTU
Machine learning is not a calculation. It is an estimation.

- Machine learning will **never be as accurate** as a model that fully describes a system or process.
  - Why: ML does not calculate a function. It estimates its result.

- Then, **why** shall we apply Machine Learning?
  1. extremely fast
  2. good alternative if we do not have full knowledge of the actual model
     - Handle **very complex systems**
     - Infer from incomplete data

- **Data is key**

  ![Image of a black box with “Garbage in” and “Garbage out”]

ML Opportunities and Barriers for Power systems

- ML can work well for forecasting/predicting
  - Weather/wind/PV or load forecasting, prediction of electricity prices, prediction of failures

**Barriers**

1. Why would we use a “black box” to decide about a **safety-critical application**?
   - Example #1: ML for security assessment
   - Example #2: ML for any type of power system optimization/optimal power flow

2. Accuracy is a purely statistical ML performance metric. Who guarantees that the Neural Network can handle well previously unseen operating points?

3. Why would we depend on **incomplete data**, when we have developed **detailed physical models** over the past 100 years?
Security Assessment with Neural Networks

1. Split the database in a training set and a test set

Database of secure/insecure operating points
Any combination of static/dynamic security criteria (e.g. N-1 & Small-signal stability)

2. Train a neural network

3. Test the neural network

4. Is accuracy high enough?

NN Output:
Binary classification: secure/insecure
Extremely fast: up to 100-1’000x faster
Why accuracy is not enough? (or “Data is key”)

- Example: Power System Security Assessment; Classify SAFE or UNSAFE

<table>
<thead>
<tr>
<th>Total operating points</th>
<th>Actually Safe (Total = 20)</th>
<th>Actually Unsafe (Total = 980)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted safe</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>Predicted Unsafe</td>
<td>19</td>
<td>950</td>
</tr>
</tbody>
</table>

Accuracy = \frac{1 + 950}{1000} = 95%

- 95% accurate but we have misclassified almost all truly safe points!

- We need high-quality training databases, and
- We need Neural Network Verification for safety-critical applications
  - Formal guarantees about the classification over continuous input regions
  - We no longer rely on statistical performance metrics
  - Systematic identification of adversarial examples
Neural Network Verification: HOW?

1. Convert the neural network to a set of linear equations with binaries
   • The Neural Network can be included in a mixed-integer linear program

2. Formulate an optimization problem (MILP) and solve it → certificate for NN behavior

3. Assess if the neural network output complies with the ground truth

Two types of optimization problems:

A. Certify that in a region around a given input $x_{\text{ref}}$ the neural network maintains the same classification → guarantee that all input points (continuous range) in the neighborhood will be classified the same

B. Find the minimum distance from $x_{\text{ref}}$ that the classification changes (possibility of adversarial examples)

From Neural Networks to Mixed-Integer Linear Programming

- Most usual activation function: ReLU
- ReLU: Rectifier Linear Unit

Non-linear activation functions  
Linear weights

input  
output
From Neural Networks to Mixed-Integer Linear Programming

- Linear weights
- On every node: a non-linear activation function
  - ReLU: $u_j = \max(0, w_{ij}u_i + b_i)$
- But ReLU can be transformed to a piecewise linear function with binaries

```
\text{ReLU}
```

```
\text{MILP}
```
From Neural Networks to Mixed-Integer Linear Programming

- Input: Active power gen. setpoints
  \[ x = [p_{g1}, p_{g2}, \ldots, p_{gN}]^T \]

- Output
  - Binary classification: safe/unsafe
  - Output vector \( y \) with two elements:
    \[ y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \]
    - \( y_1 \geq y_2 \): safe
    - \( y_1 < y_2 \): unsafe
Certify the output for a continuous range of inputs

- We assume a given input $x_{\text{ref}}$ with classification $y$: $y_1 > y_2$

1. For distance $\epsilon$ evaluate if input $x$ exists with different classification $y_2$

$$\max_{x,y} \quad y_2 - y_1$$

s.t. $$y = NN(x)$$

$$|x - x_{\text{ref}}|_\infty \leq \epsilon$$

Why use incomplete datasets when we have detailed physical models?

- **Physics-Informed Neural Networks**
  1. Include the physical models inside the NN training $\rightarrow$ need for less data, probably smaller NN sizes
  2. **Extremely fast**: can potentially replace solvers for systems of differential-algebraic equations
  3. Turn **NN training** from supervised to **unsupervised learning**

Single Machine Infinite Bus Example: Physics-Informed NN is 87x faster than ODE solver

Takeaways

1. **Extremely fast** (up to 1’000x faster): Revolutionize power system computation → replace non-linear and differential-algebraic solvers for e.g. OPF, sec. assessment, etc.

2. **Data is key**: Sampling beyond statistics
   - Use physical modeling (e.g. convex relaxations of OPF) to accelerate generation of high-quality data

3. We need **Verification of Neural Networks** for safety-critical applications

4. **Physics-Informed Machine Learning**: Use the already available models in the NN training

---

Did not have the time to talk about in this presentation:

5. **Interpretable AI**: we need to better understand how ML behaves to remove the barriers for its application in power systems

6. **From NN to MILP**: Decision Trees/Neural Networks can capture previously intractable constraints and convert them to a MILP

7. Before you apply ML/RL on power systems: Think! **ML/RL is an estimation, not a calculation!**
   - Given infinite time or data ML/RL will converge to the correct decision/optimal strategy. Do you have solid reasons why spending so many resources for ML/RL training will lead to a better performance than a Linear Program or Kalman filter? If not, then better stick to the conventional approach.
Thank you!

Spyros Chatzivasileiadis
Associate Professor, PhD
www.chatziva.com
spchatz@elektro.dtu.dk

Some datasets and code on Machine Learning for Power systems (e.g. Physics-Informed ML, NN verification, etc) available at:
www.chatziva.com/downloads.html