DTU Electrical Engineering
Department of Electrical Engineering

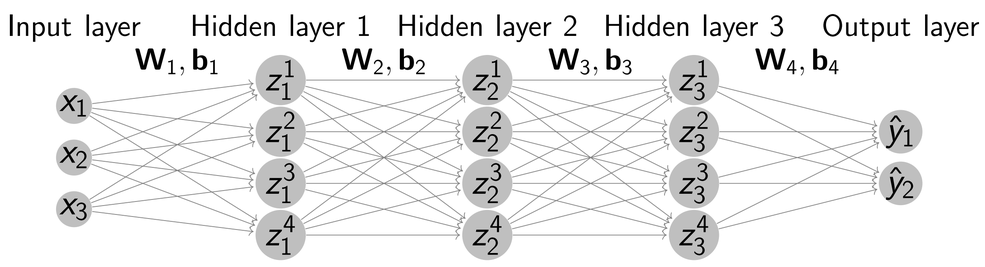
Verification of Physics-Informed Neural Networks: Formal Guarantees for Power System Applications

Andreas Venzke and Spyros Chatzivasileiadis
Department of Electrical Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark e-mail: {andven, spchatz}@elektro.dtu.dk.

Overview

- **Increasing uncertainty** in both generation and load increase **complexity** significantly and challenge secure power system operation.
- Machine learning approaches including neural networks have shown **promise** e.g. for power system security assessment.
- Neural networks are, however, treated so far as **black-box tools** and trained **physics-agnostic**. ⇒ **major obstacle** towards adoption in practice!
- For power system applications, the goals of this work [Ref. 1] [Ref. 2] are to:
  1. **provide formal guarantees** of neural network behaviour
  2. **train physics-informed** neural networks

Neural Network Architecture and Training



- Feed-forward **classification** and **regression** neural networks with non-linear ReLU activation functions $\hat{\mathbf{y}} = NN(\mathbf{x})$ predicting output $\hat{\mathbf{y}}$ for input $\mathbf{x} \in \mathcal{D}$:

For the input layer:	$\hat{\mathbf{z}}_1 = \mathbf{W_1 x} + \mathbf{b_1}$	
For each hidden ReLU layer k:	$\mathbf{z_k} = \max(\hat{\mathbf{z}}_k, 0)$	$\forall\, k = 1, ..., K$
	$\hat{\mathbf{z}}_{\mathbf{k}+1} = \mathbf{W_{k+1} z_k} + \mathbf{b_{k+1}}$	$\forall\, k = 1, ..., K-1$
For the output layer:	$\hat{\mathbf{y}} = \mathbf{W_{K+1} z_K} + \mathbf{b_{K+1}}$	

- Neural network training using samples $\mathbf{S} : \mathbf{x} \rightarrow \mathbf{y}$ with ground-truth $\mathbf{y}$ minimizing loss function $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ to determine weights $\mathbf{W}$ and biases $\mathbf{b}$
- Reformulation of max-operator $\mathbf{z_k} = \max(\hat{\mathbf{z}}_\mathbf{k}, \mathbf{0})$ using **integer variables** $\mathbf{r_k}$ for each neuron and layer proposed by [Ref. 3].

Formal Guarantees for Security Classifiers

- For **classification** neural networks, we verify **continuous** input regions with the same classification. ⇒ No **adversarial example** exists.
- For defined input $\mathbf{x}_{\text{ref}}$ with classification y_1 ($y_1 > y_2$) we compute guarantees:
  1. For distance ϵ evaluate if input $\mathbf{x}$ exists with different classification $\hat{y}_2$:

$$\max_{\mathbf{x} \in \mathcal{D}, \hat{\mathbf{y}}} \hat{y}_2 - \hat{y}_1 \quad \text{s.t. } \hat{\mathbf{y}} = NN(\mathbf{x}), |\mathbf{x} - \mathbf{x}_{\text{ref}}|_\infty \leq \epsilon$$

  2. Minimize distance ϵ from $\mathbf{x}_{\text{ref}}$ to input $\mathbf{x}$ with different classification $\hat{y}_2$:

$$\max_{\mathbf{x} \in \mathcal{D}, \hat{\mathbf{y}}, \epsilon} \epsilon \quad \text{s.t. } \hat{\mathbf{y}} = NN(\mathbf{x}), |\mathbf{x} - \mathbf{x}_{\text{ref}}|_\infty \leq \epsilon, \hat{y}_2 \geq \hat{y}_1$$

Incorporating Physical Constraints

- For **regression** neural networks, we consider **linear** constraints $\mathcal{F}(\mathbf{x}, \mathbf{y}) \leq 0$ imposed by physical principles.
- To compute **worst-case guarantees** for **trained** neural network, we solve:

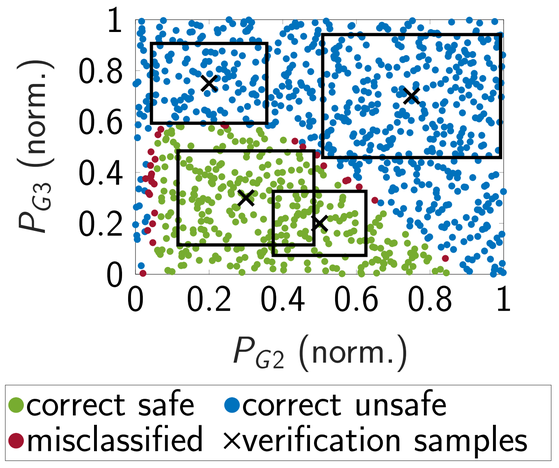
$$\max_{\mathbf{x} \in \mathcal{D}} \mathcal{F}(\mathbf{x}, \hat{\mathbf{y}}) \quad \text{s.t. } \hat{\mathbf{y}} = NN(\mathbf{x})$$

- To **train physics-informed** neural networks, we propose **adversarially robust** training using Fast Gradient Sign Method (FGSM) by [Ref. 4]:

$$\min_{\mathbf{W}, \mathbf{b}, \mathbf{x} \in \mathcal{D}, \mathbf{y}, \hat{\mathbf{y}}} \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) + \gamma \max_{\mathbf{x}_{\text{adv}}, \hat{\mathbf{y}}_{\text{adv}}} \mathcal{F}(\mathbf{x}_{\text{adv}}, \hat{\mathbf{y}}_{\text{adv}})$$
$$\text{s.t. } \hat{\mathbf{y}} = NN(\mathbf{x}) \qquad \text{s.t. } \hat{\mathbf{y}}_{\text{adv}} = NN(\mathbf{x}_{\text{adv}}), |\mathbf{x} - \mathbf{x}_{\text{adv}}|_\infty \leq \epsilon$$

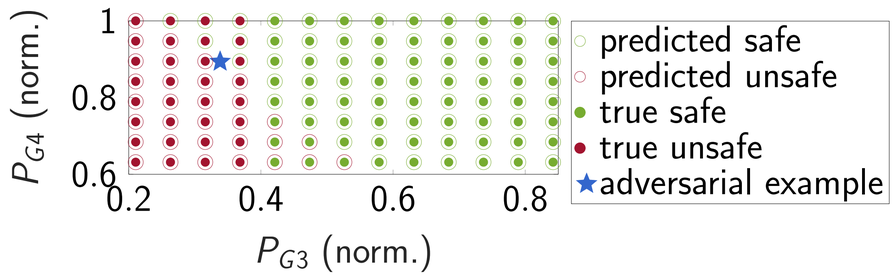
Verify Properties for Security Classifier

- **Application:** Neural network to classify operating points as safe or unsafe with respect to power system security criteria
- We solve mixed-integer linear programs (MILPs) for each verification sample to determine closest sample which changes classification
- Obtain classification guarantees for **continuous** input regions



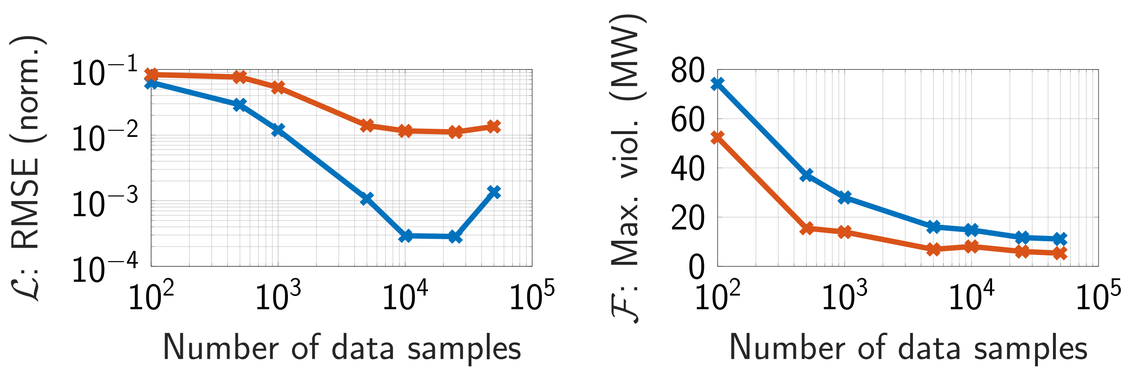
Identifying Adversarial Examples

- Small perturbations to input $\mathbf{x}$ which falsely lead to output mis-classification.



Worst-Case Constraint Violations

- DC Optimal Power Flow is linear program for market and power system operation minimizing generator cost subject to physical constraints.
- **Application:** Neural network to predict optimal generator dispatch for a given system loading.
- **Test case:** IEEE 30 bus system with 20 loads and 6 generators, total loading of 189 MW for base case. We consider $\mathcal{D} \triangleq \pm 20\%$ load variation.



Future Work

1. Explore **trade-off** between accuracy and satisfying physical constraints
2. Obtain guarantees related to **non-linear** physical constraints

References

[Ref. 1] A. Venzke, and S. Chatzivasileiadis. "Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications." IEEE Transactions on Smart Grid (2020).
[Ref. 2] A. Venzke, G. Qu, S. Low, and Chatzivasileiadis, S. "Learning Optimal Power Flow: Worst-Case Guarantees for Neural Networks". arXiv preprint arXiv:2006.11029. (2020)
[Ref. 3] V. Tjeng, K. Y. Xiao, and R. Tedrake. "Evaluating Robustness of Neural Networks with Mixed Integer Programming." ICLR 2019 (2019).
[Ref. 4] I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples." preprint arXiv:1412.6572 (2014).