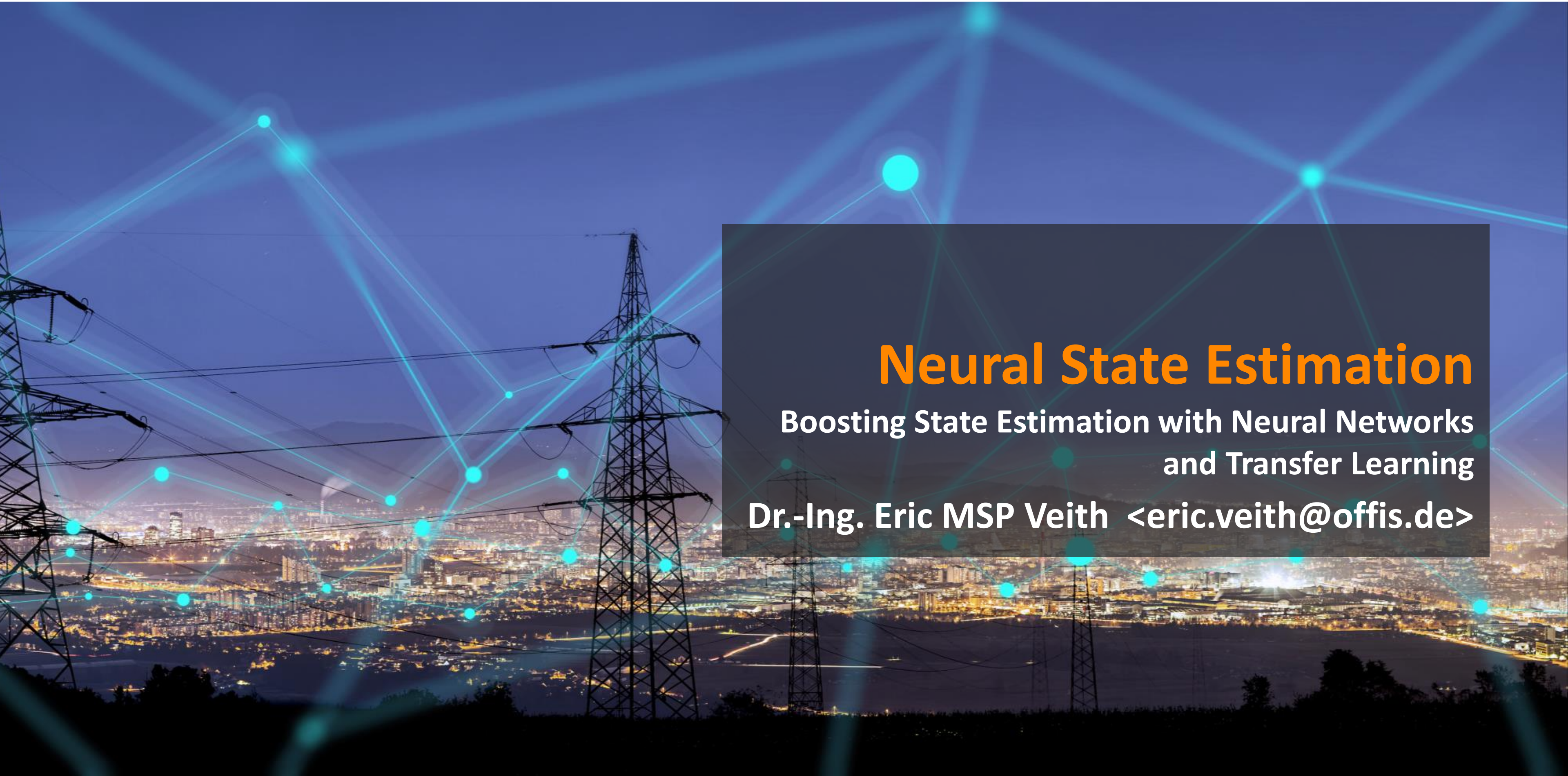


Neural State Estimation

Boosting State Estimation with Neural Networks and Transfer Learning

Dr.-Ing. Eric MSP Veith <eric.veith@offis.de>

Energy Informatics at OFFIS

Energy research in Oldenburg

- since the 80s “regenerative” (Luther, Schellnhuber, Appelrath,...)
- since OFFIS was founded in 1991 also “Energy Informatics” (EI)

EI milestones of OFFIS (and Uni-Informatik), e.g.

- 1995: First wind power information system in Germany
- 2003: Early “decentralized energy management systems”
- 2008: with eTelligence at E-Energy
- 2010: First Energy Informatics Professorship in Germany
- 2017: SINTEG project partner (Designetz and Enera)

Current state

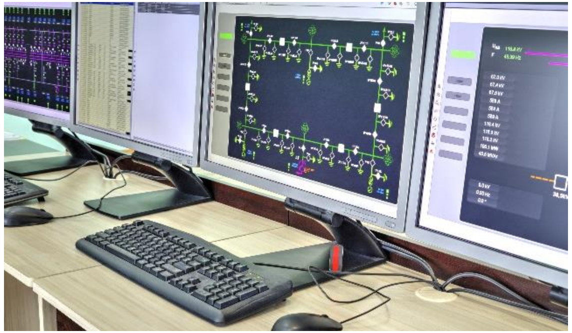
- Largest EI team in D/EU (> 100 employees)
- Federal, EU, contract projects e.g. with EWE, Innogy, TenneT, E.ON, Siemens, BTC, ABB, Alcatel Lucent, FhG-Instituten, KIT, AIT/ Wien, KTH/ Stockholm, DTU/ Risø

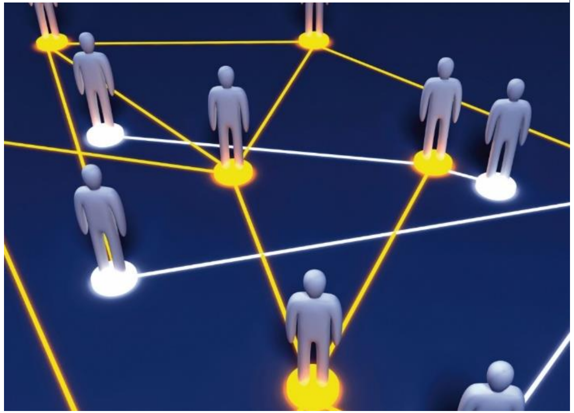
Kopernikus-Projekte für die Energiewende — Bundesministerium für Bildung und Forschung

Schaufenster intelligente Energie Digitale Agenda für die Energiewende" (**SINTEG**) — Bundesministerium für Wirtschaft und Energie

PYRATE

TRANSENSE





© AdobeStock_phadventure - oben und © Foltolia/Spectral Design - unten

Central research questions in the division

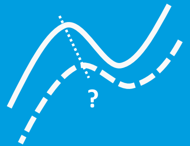
How can the digitization of the energy system transformation be made safe and secure?

How can the systemic characteristics of digitised energy systems be modelled, simulated and tested?

How can digitised energy systems be efficiently designed, operated and optimised in the field of conflicting objectives and dynamic constraints?

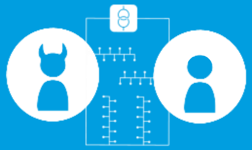
AI Research at OFFIS: Power Systems Intelligence

Resilient, De-Carbonized Power Grids through Artificial Intelligence



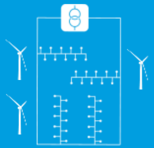
Concept Drift

eXplainable AI



ARL

Anticipatory Design



Sustainability through AI

Data Integration and Processing Layer

% whoami

Eric MSP Veith

Computer Scientist

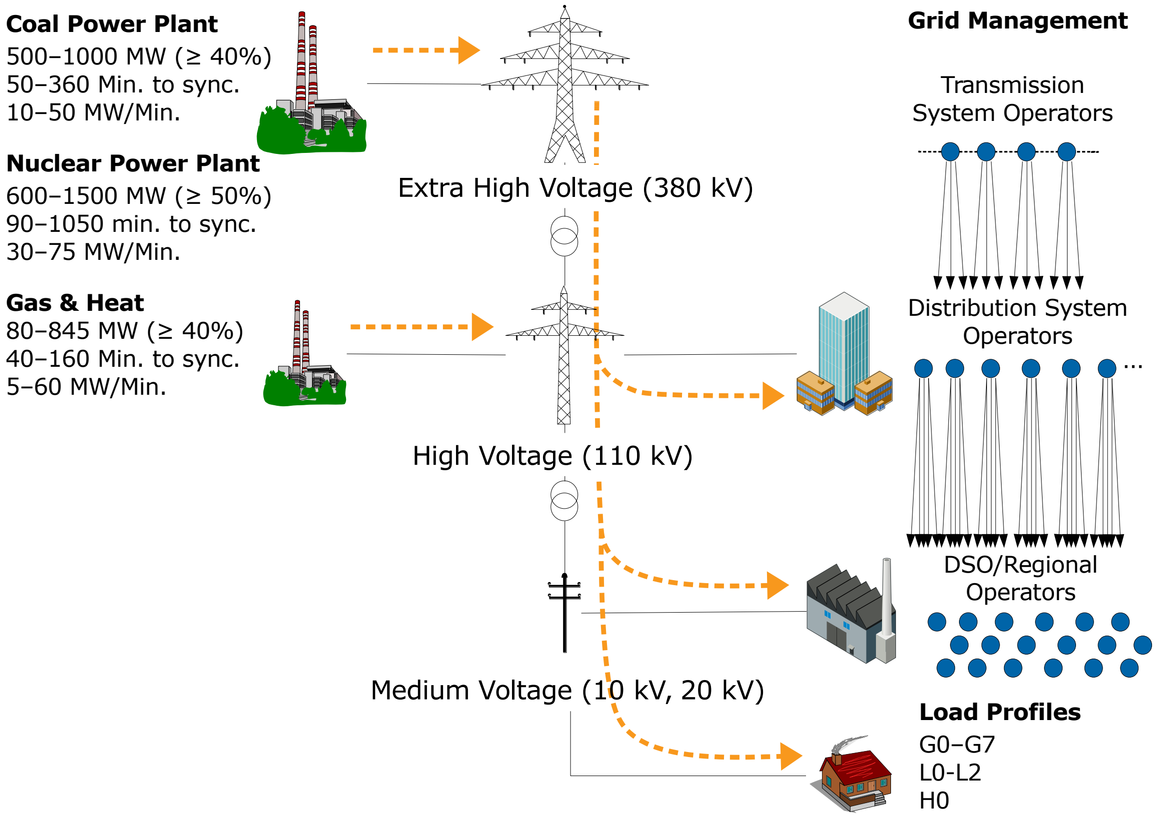
- > University of Appl Sciences Worms (Diploma 2010)
  (NFS Integration in High-Availablity Storage Cluster)
- > Freiberg University of Mining and Technology (PhD 2017)
  Universal Smart Grid Agent

R&D Group Manager at OFFIS, Oldenburg
Power Systems Intelligence

- > AI in Critical Infrastructures
- > Adversarial Resilience Learning
- > Deep Reinforcement Learning, Continual Learning,
  eXplainable Reinforcement Learning

A Not So Long Time Ago…

…A Power Grid Near, Near-By.



The power grid:
A Hierarchy.

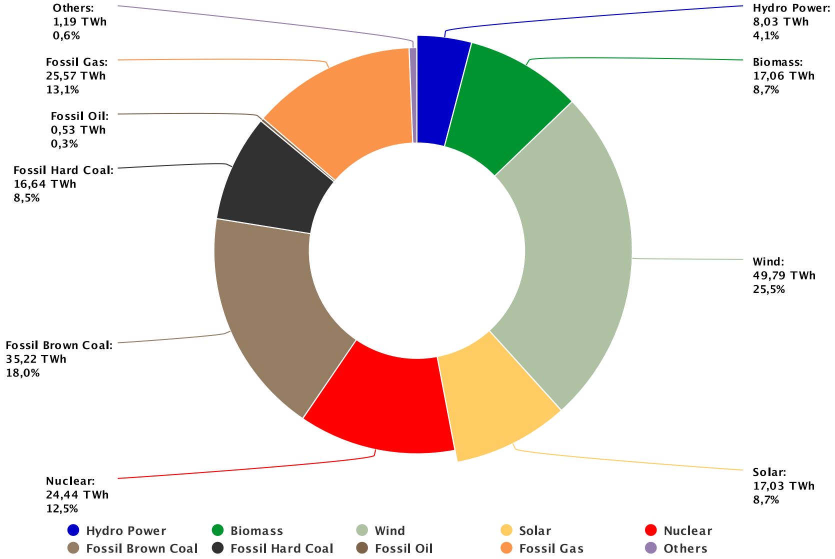
TSOs always needed current data.

DSOs never before.

These times **are long since over…**

The Rise of the Renewables

Toppling the Hierarchy

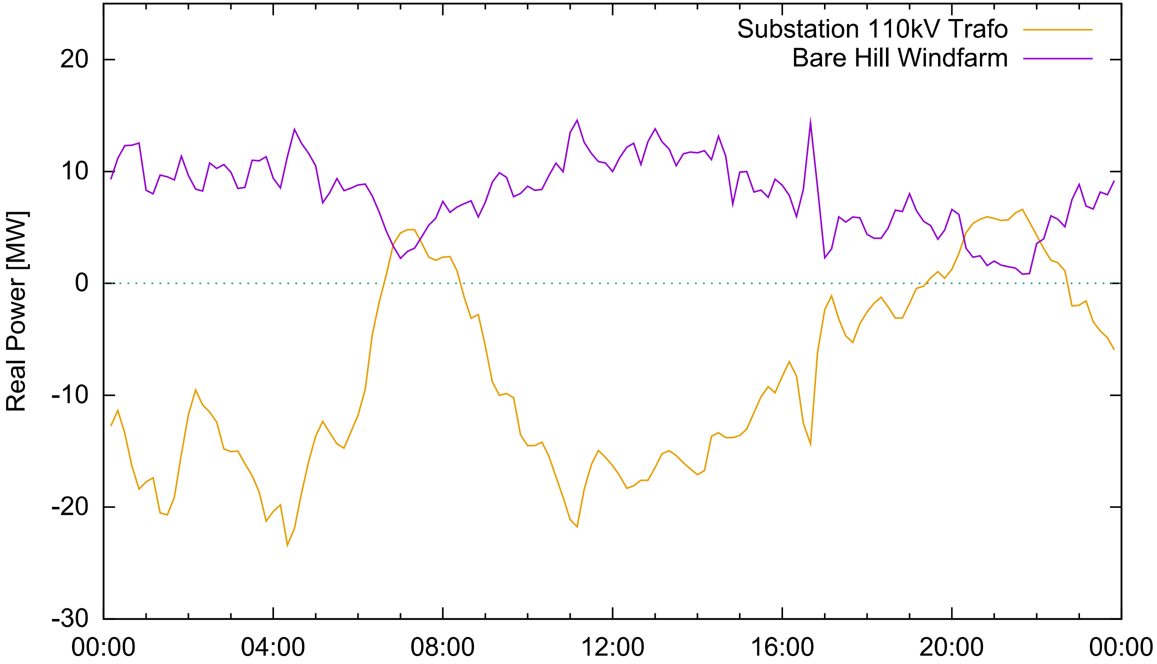


energy-charts.info/Bruno Burger

47% from Renewables in 2021 (Germany)

A Hierarchy No More

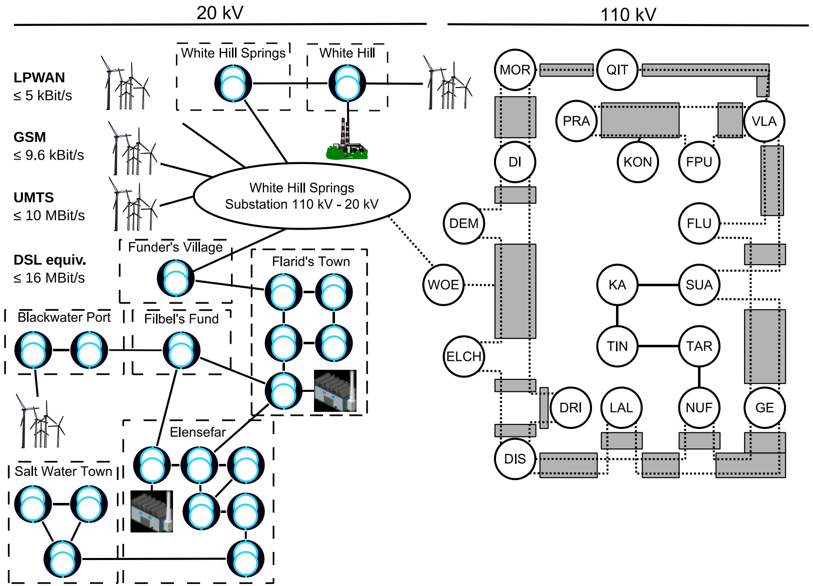
Typical Load Flow at a 110-20kV Transformer



Mostly feed-back!

Network Transparency

Or: Why It Is Important to Know the State of the Grid



Grid state must be known.

- Bi-directional feed-in
- Atypical usage patterns
- Wish to use infrastructure more efficiently
- Implementation of regional energy markets

Deploy more sensors?

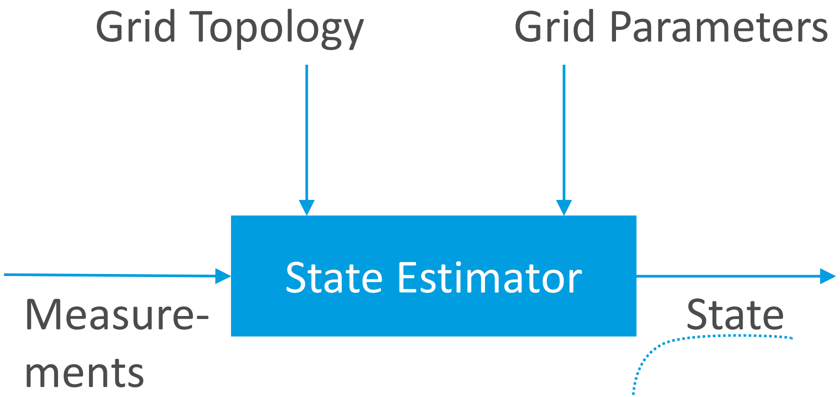
- ICT often a problem
- Costs of installation & data processing

Solution? Make edulcated guesses.

- Called *State Estimation*

State Estimation

Estimating the State of the Grid from Measurements



$$r = \frac{m}{2n-1} - 1$$

Redundancy Factor

$r = 1 : m = 2n$

1. Topology

- Initial design
- State of switches, breakers, etc.

2. Parameters

- Position of tap changers
- Shunt capacitors

3. Gathering Measurements

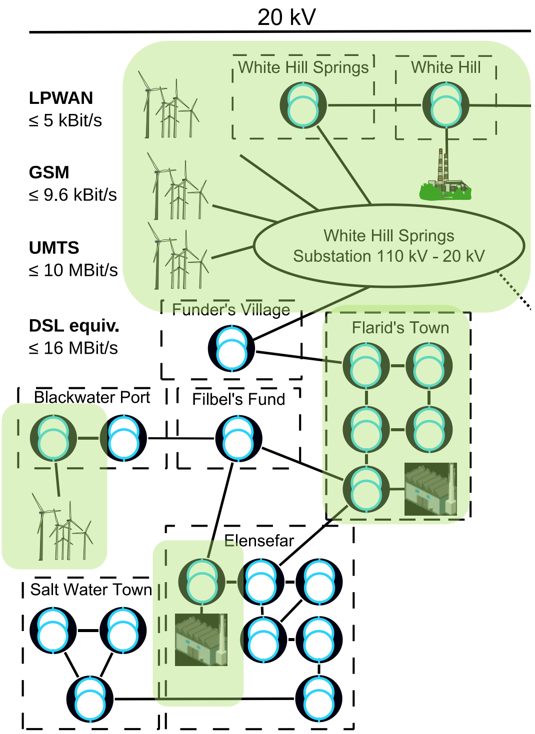
- Voltage, Angles

4. Estimate State

- Weighted least squares to elminiate bad data and estimate state

Pseudo-Measurements

Replacing Unknown Values



A grid has usually only “islands of observability.”

- Important grid connectors (transformers)
- Major consumers/feed-in

Missing values substituted with Pseudo Measurements

- E.g., switch closed, but impedance unknown: 0
- Inferring voltages from current flow at branch interconnection

$$P_{mk}V_m + (P_mk\ \cos\theta_{km} - Q_mk\sin\theta_{km}V_k) = 0$$
$$Q_{km}V_m + (P_mk\ \sin\theta_{km} - Q_mk\sin\theta_{km}V_k) = 0$$

Speed and Ill-Conditioning

How to Optimize State Estimation

State Estimation is iterative at heart:

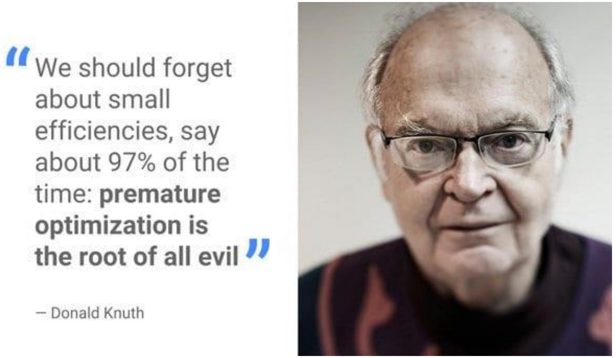
- $$\boldsymbol{G}(\boldsymbol{x}^{v})\Delta\boldsymbol{x}^{v} = \boldsymbol{H}'(\boldsymbol{x}^{v})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})$$
  $$\boldsymbol{x}^{v+1} = \boldsymbol{x}^{v} + \Delta\boldsymbol{x}^{v}$$
- G: Gain Matrix; H: Hessian
- Elements of $\boldsymbol{H}$ are functions of $(\boldsymbol{\theta}, \boldsymbol{V})$, vary by each iteration

State Estimation is compute-intensive, but can be optimized:

- Normalize by voltage (i.e., $P \to \frac{P}{V}, Q \to \frac{Q}{V}, \ldots$), then $\boldsymbol{H} = \text{const}$
- Obtain $\boldsymbol{H}_{QV}$ from asset data and set $b_{km} = -\frac{1}{x_{km}}$, similar for $\boldsymbol{H}_{P\theta}$

Optimization can be the Root of Evil

Not Quite Premature, but Still…



A necessary optimization to speed up convergence

> Assume a specific relation of impedance to resistance (X/R)

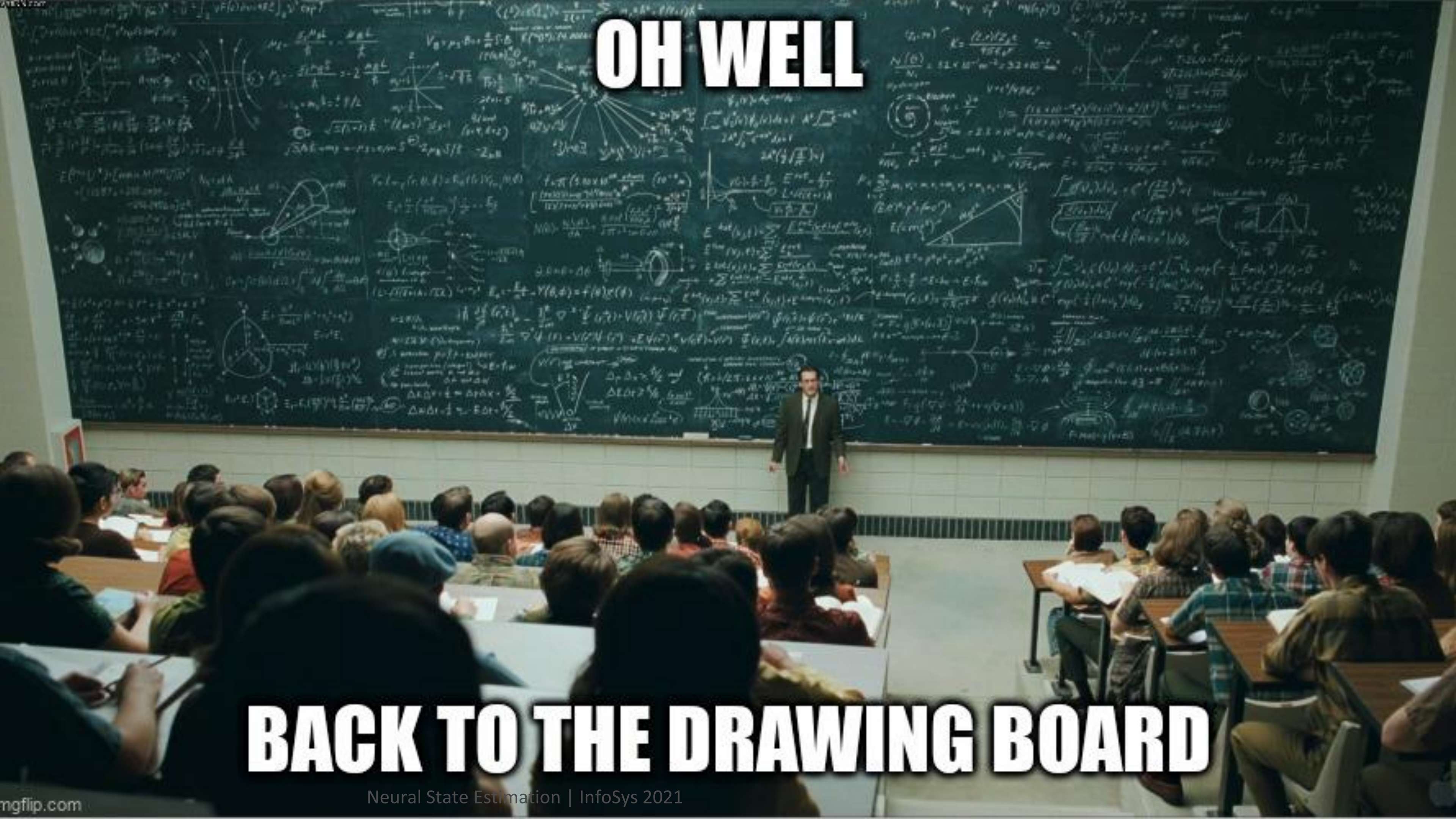
> Fix H, approximate as if DC

> … works very well for transmission grids!

The optimization doesn’t hold for DG anymore.

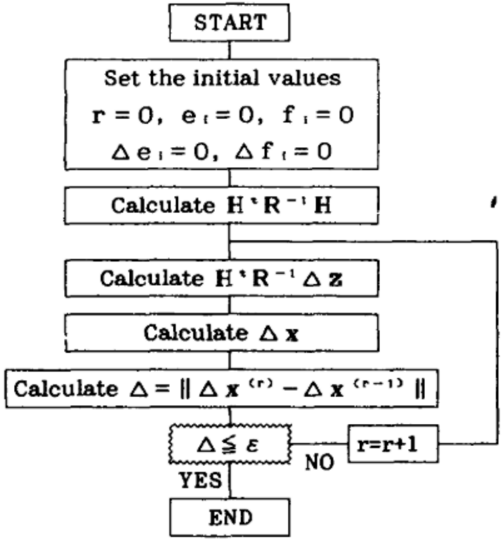
> DSOs employing SE unexpected even in the early 2000s

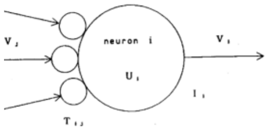
> Now a necessity! (Remember power flow patterns in DGs?)

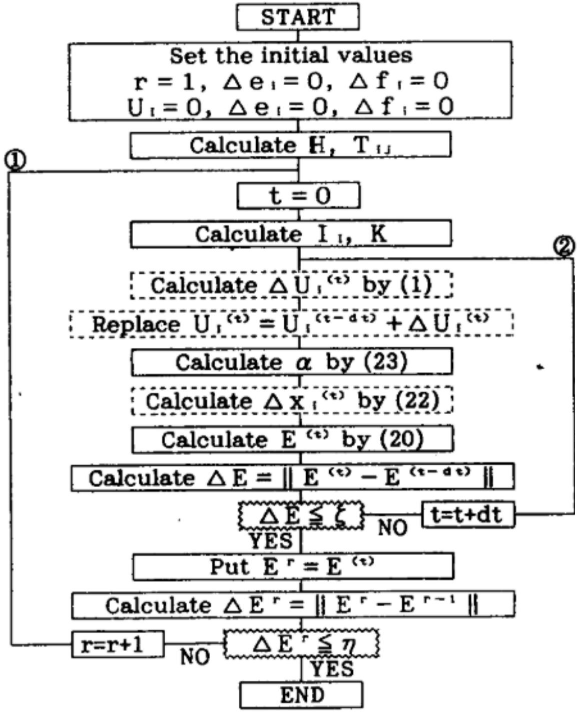

OH WELL
BACK TO THE DRAWING BOARD
imgflip.com

State Estimation Using Artificial Neural Networks

Publications as Early as 1991







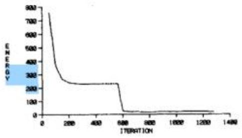


Figure 9 : Convergence characteristic of the energy function (α = 0.11)

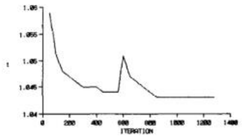
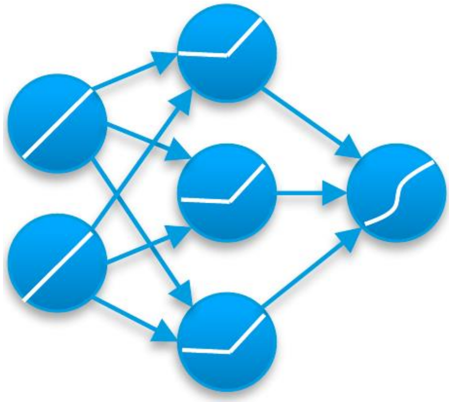
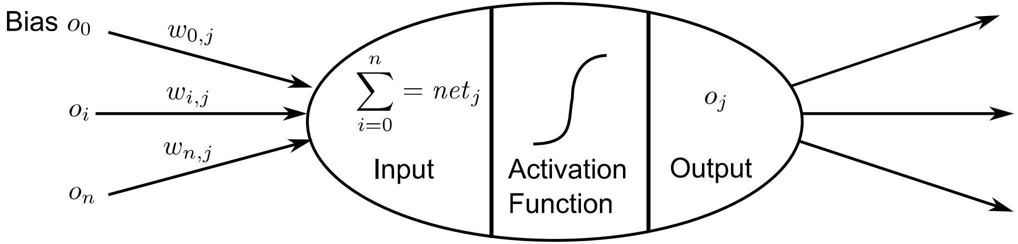


Figure 10 : Convergence characteristic of the result of el (α = 0.11)

Nakagawa, Hayashi & Iwamoto (!) replace SE with Hopfield Networks in 1991.



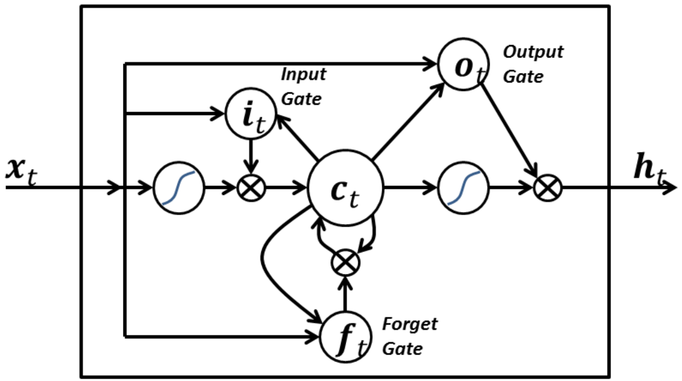


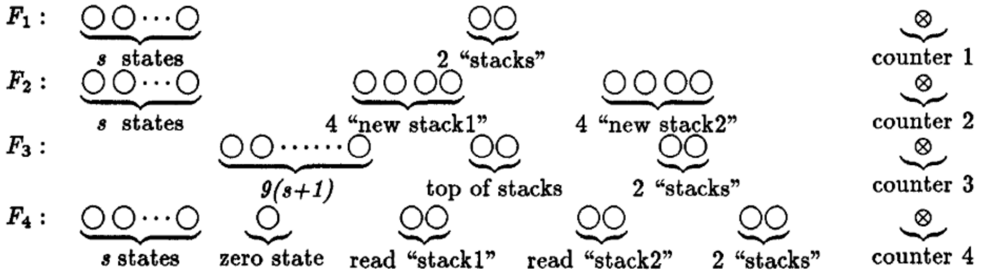
Multi-Layer Perceptrons: Universal approximators of any Borel-measurable function

- $f(5) = 2$: $f(x) = x - 3$; $f(4) = 2$: $f(x) = \left\lfloor \frac{x}{2} \right\rfloor$, $f(x) = 2$, … - infty much
- $F(\boldsymbol{x}, \boldsymbol{\theta}) = F^{(n)}(F^{(n-1)}(F^{(n-2)}(\cdots(F^{(1)}(\boldsymbol{x}, \boldsymbol{\theta})))))$

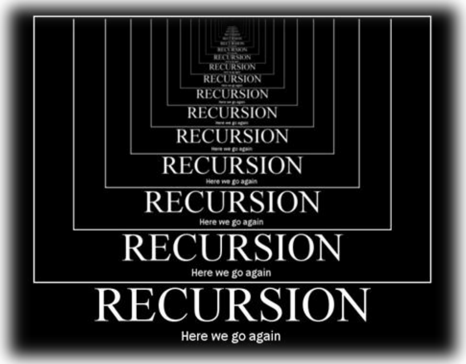
Why Does This Work? Pt. 2

A Shallow Introduction To… Recursive Neural Networks





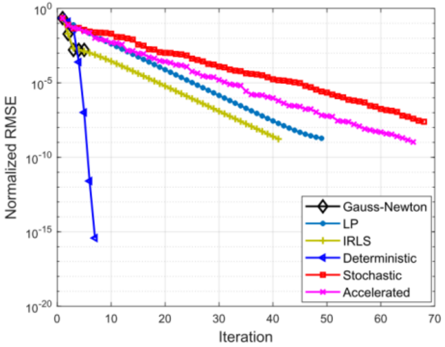
Siegelmann & Sontag show 1992, that Recurrent Neural Networks are universal approximators of dynamic systems.



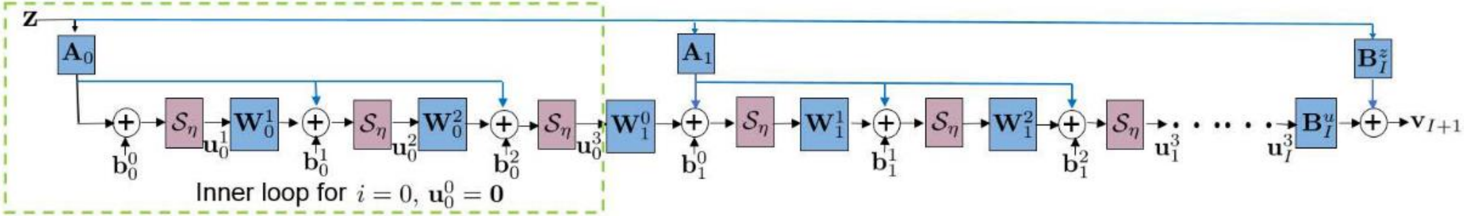

MONTH
DAY
YEAR
AM
PM
HOUR
MIN
JUL 05 2010 01:21
DESTINATION TIME
MONTH
DAY
YEAR
AM
PM
HOUR
MIN
OCT 26 1985 01:22
PRESENT TIME
MONTH
DAY
YEAR
AM
PM
HOUR
MIN
OCT 26 1985 01:20

ANN architecture built to optimize proximate-linear solver

- Based on Least-Absolute-Value estimator: robust, but slow (both in contrast to Weighted Least Squares)
- Prox-linear algorithms minimize a sequence of convex quadratic subproblems



- Prox-linear algorithms still struggle with changing conditions (RES feed-in!)
- Idea: Unroll solver iterations into DNN architecture based on phyiscal model



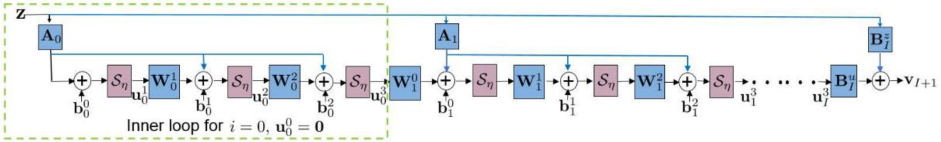
Wedding Prox-Linear Algorithms and DNNs

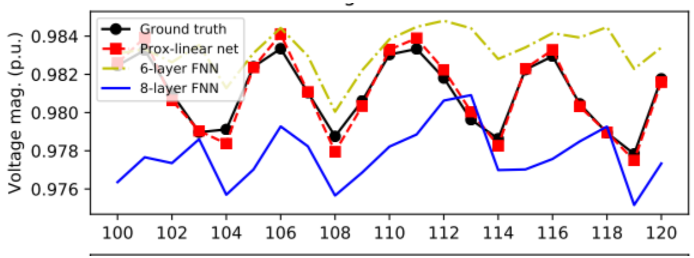
Hybrid of Data-Driven and Physical Model

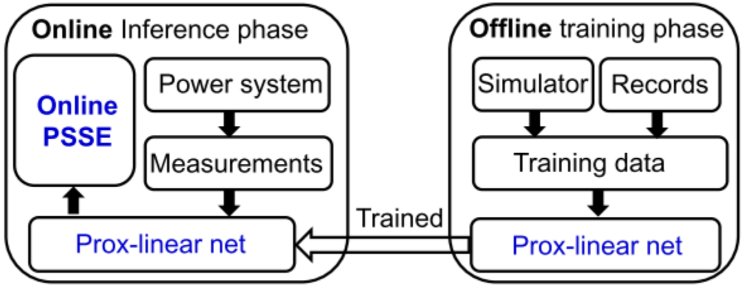
Algorithm 1 Reduced-complexity prox-linear solver.

Input: Data $\{(z_m, \mathbf{H}_m)\}_{m=1}^{M}$, step sizes $\{\mu_i\}$, η, and initialization $\mathbf{v}_0 = \mathbf{1}$, $\mathbf{u}_0^0 = \mathbf{0}$.

for $i = 0, 1, \ldots, I$ **do**
  Evaluate $\mathbf{W}_i^k$, $\mathbf{A}_i$, and $\mathbf{b}_i^k$ according to (7).
  Initialize $\mathbf{u}_i^0$.
  for $k = 0, 1, \ldots, K$ **do**
    Update $\mathbf{u}_i^{k+1}$ using (6).
  end for
  Update $\mathbf{v}_{i+1}$ using (5).
end for





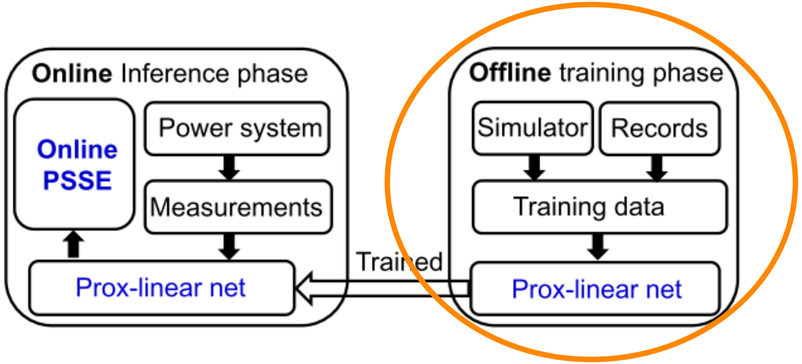


Wang, G., Giannakis, G. B., & Chen, J. (2019). Robust and scalable power system state estimation via composite optimization. *IEEE Transactions on Smart Grid*, *10*(6), 6137–6147. https://doi.org/10.1109/TSG.2019.2897100

Zhang, L., Wang, G., & Giannakis, G. B. (2019). Real-Time Power System State Estimation and Forecasting via Deep Unrolled Neural Networks. *IEEE Transactions on Signal Processing*, *67*(15), 4069–4077. https://doi.org/10.1109/TSP.2019.2926023

Isn’t There Still a Catch?

The Problem of Training



- An ANN still needs training (obviously).
- Prox-linear nets are still ANNs.
- Model-building (training) happens on a given data-set.

The ANN (prox-linear net) is suitable only for the power grid it was trained on.

Transfer Learning to the Rescue

Transferring Models Between Problem Domains

Source Domain $\mathcal{D}_S$

Target Domain $\mathcal{D}_T$

Feature Space $\mathcal{X}_S$

Marginal Distribution $P(\mathcal{X}_S)$

Mapping Function $F: \mathcal{X}_S \longmapsto \mathcal{Y}_S$

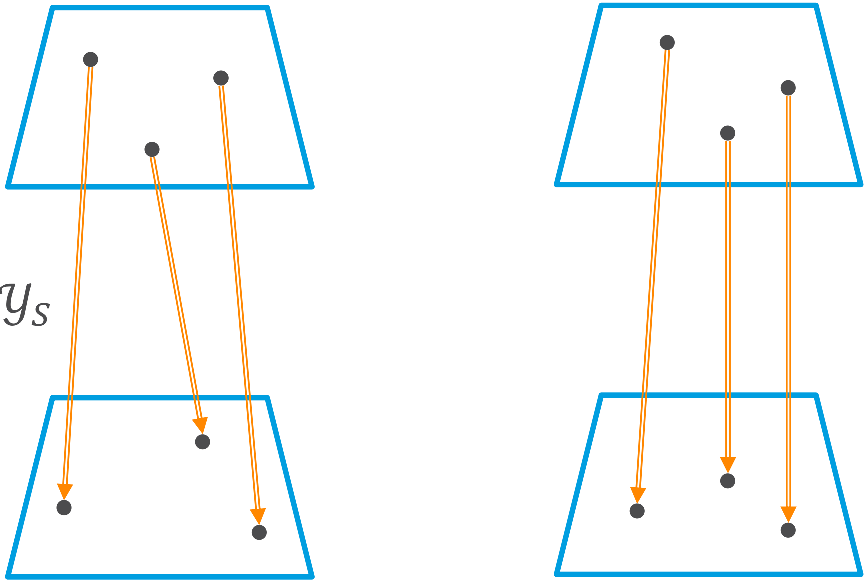
Label Space $\mathcal{Y}_S$

$\mathcal{X}_T$

$P(\mathcal{X}_T)$

$F: \mathcal{X}_T \longmapsto \mathcal{Y}_T$

$\mathcal{Y}_T$



Types of Transfer Learning

The Easy and the Hard Way

Homogeneous Transfer Learning

- $\mathcal{X}_S = \mathcal{X}_T$
- Marginal propability distributions match
- Simple approach: Throw away input/output layer and retrain

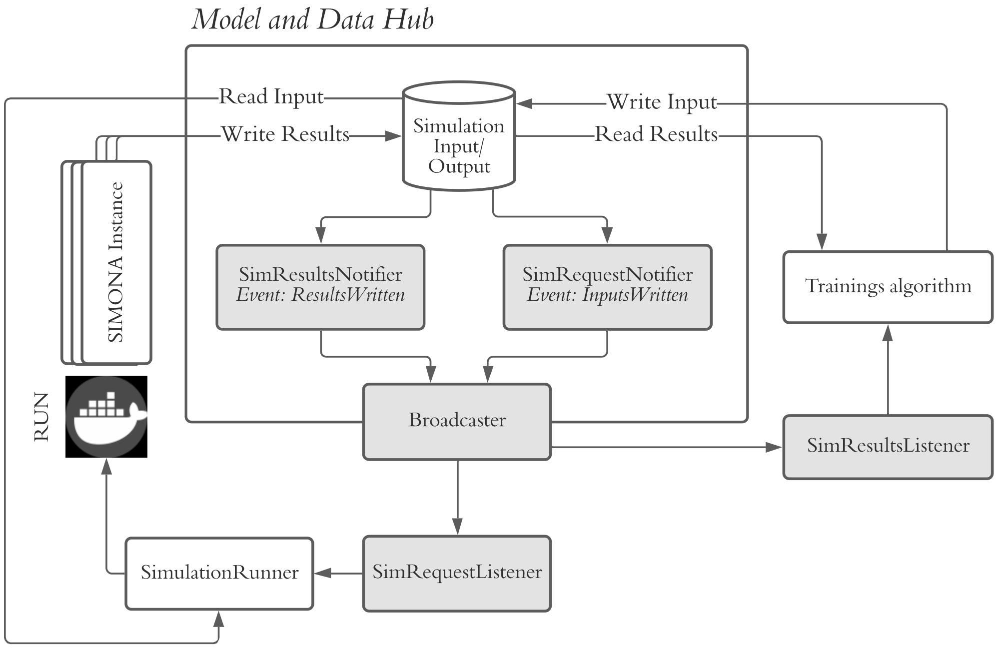
Heterogeneous Transfer Learning

- $\mathcal{X}_S \neq \mathcal{X}_T$
- Different marginal propability distributions
- Specific Transfer Learning algorithm

Transfering Neural State Estimation is almost always Heterogenous Transfer Learning.

Putting It All Together

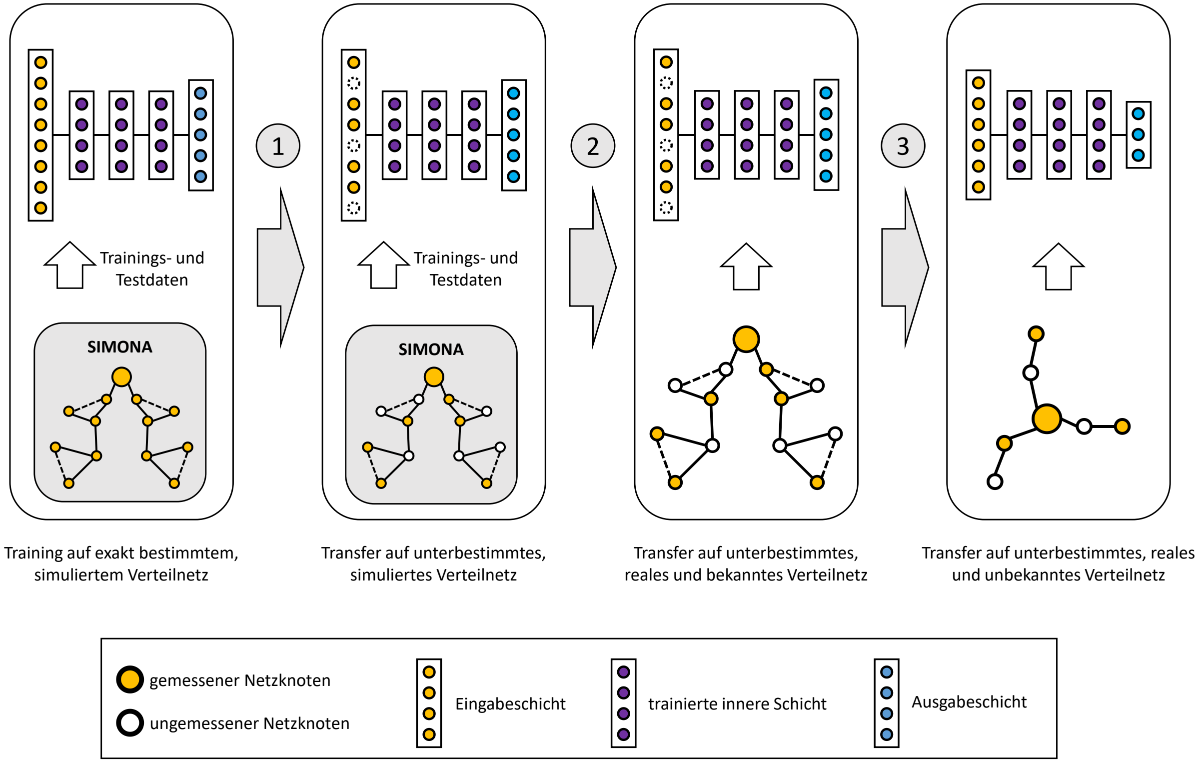
Project Transense



1. Create simulation model from real data
2. Train Prox-Linear Net PSSE
3. Interface between Training and Simulator (SIMaaS) ensures minimal amount of samples
4. Employ & evaluate Transfer Learning

https://www.offis.de/en/offis/project/transense.html

Validation Approach in the Project



TRANSENSE

Neural State Estimation

Boosting State Estimation with Neural Networks and Transfer Learning

Dr.-Ing. Eric MSP Veith <eric.veith@offis.de>