Artificial intelligence based surrogate modeling 1: case thermodynamic data banks. Case studies with precipitation processes

Peyman Davvalo Khongar, Ulla Ojaniemi, Timo Niemi

VTT Technical Research Centre of Finland

18.12.2018

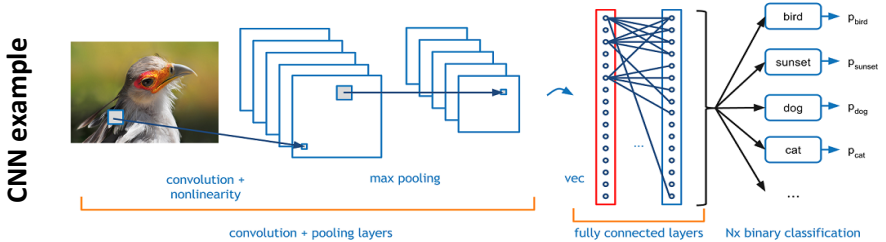
Contents

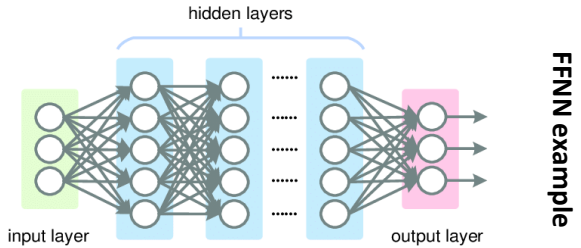
- Introduction to artificial intelligence (A.I.)
- Ettringite patch process
- Continuous 1-phase ettringite process
- Electrochemical water treatment (EWT) process
- Conclusion and future works
- Ideas on modelling factory platform

Introduction to artificial intelligence (A.I.)

What is A.I.?

- There is **no** solid definition for A.I.
- It is the science and engineering of making **intelligent machines**, particularly intelligent computer programs to achieve **complex engineering goals** through **simplified** approaches.
- Some of the wildly used areas are signal processing, **image analysis**, natural language generation, speech recognition.
- Generally AI has been divided into two major subfields:
  - particular goals (**Machine learning**)
  - **Neural Networks**





- Some examples of neural networks types:
  - **Feed forward neural network** (**FFNN**)
    - ✓ General regression
  - Convolutional neural network (CNN)
    - ✓ Image analysis
  - Recurrent neural network (RNN)
    - ✓ Speech recognition

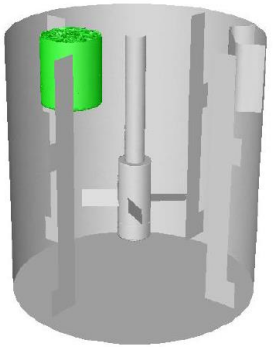
Why A.I. in this project?

- As we mentioned, A.I. is to achieve **complex engineering goals** through **simplified** approaches.
- Main target of this part of the project was to achieve an alternative approach to **Gibbs energy minimization** method for calculation of chemical species **equilibria**.
  - We have combined HSC-9.1 (a chemical equilibrium solver) with ANSYS-Fluent for batch and EWT processes.
    - ✓ **Problems**:
      - ➢ HSC is a windows-based program, thus cases should be run in windows machines **only**.
      - ➢ Using full cores over batch process case takes almost **10 min** over 1 iteration for HSC solver to calculate the chemical equilibrium and **20 sec** for flow solver in ANSYS-Fluent. For continuous 1-phase process, the ration is **40 min** to 20 sec.
- Summary of **main** target of this part of project

Design an A.I. to reduced computational time and cost of chemical equilibrium calculation while maintaining the accuracy.

Ettringite patch process

Creation of CFD model for the Ettringite patch process development



- Geometry:
  - Volume 5.25 L
  - Patch region 55.9 mL
- Grid size: 259 000 cells
- Temperature 295.75 K
- 650 rpm

Green region: $AlCl_3$ batch feed

Solution with $CaSO_4$, $Ca(OH)_2$

Thermodynamic equilibrium calculation,

- SO_4^{2-}, Ca^{2+}, OH^-, H^+, Al^{3+}, Cl^-, $Al(OH)_4^-$
- $Ca(OH)_2$ (portlandite), gibbsite, ettringite

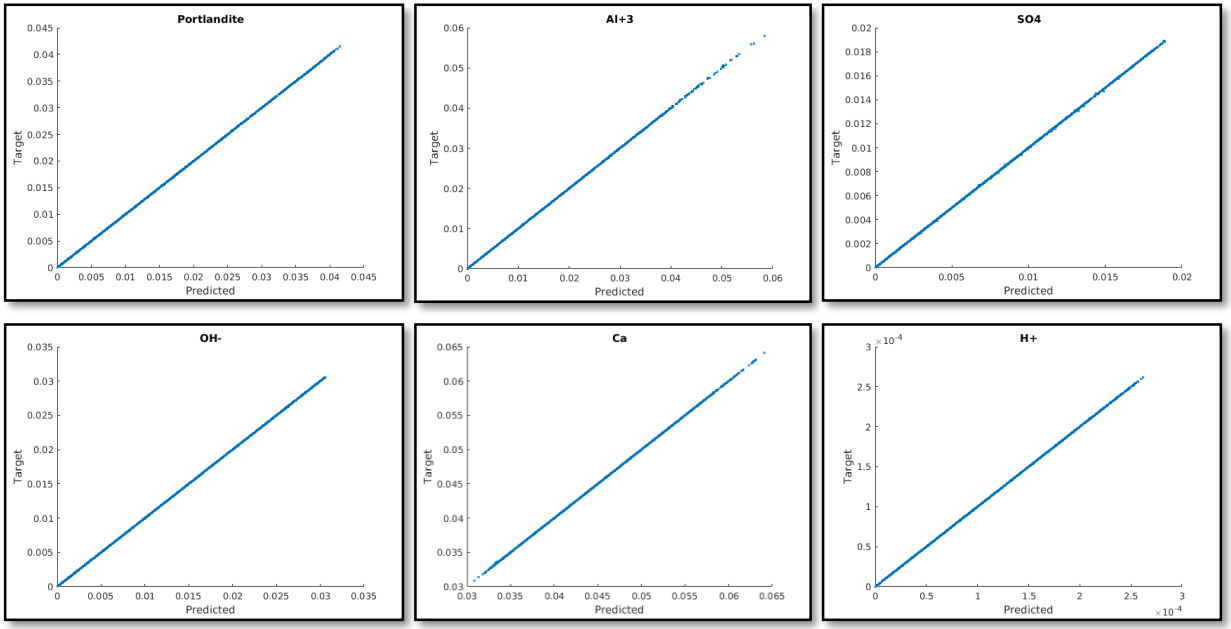
Patch process:

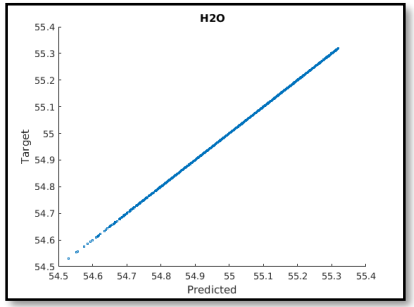
Computational Fluid Dynamics (CFD) modeling coupled with A.I. equilibrium calculation for simulating patch process.

- Precipitating solid particles in the process liquid in the reactor
- The modelling of precipitation was based on an assumption of local ion equilibrium in the liquid phase, due to the fast dissociation reactions.
- The mixing process was modelled with multiphase unsteady CFD including species transport and applying MRF (Moving Reference Frame) technique for modelling rotating mesh.
- Solving of thermodynamic equilibrium in system with several chemical components was carried out using two A.I. designs (in C++ and Matlab) coupled to the precipitation kinetics in CFD (Ansys-Fluent 19.1)
  - Equilibrium was calculated with A.I. networks in every computational cell in every time step during the simulation
  - For Matlab code, Matlab-Runtime used to couple CFD and AI in this project
  - The final design of our networks are Multi-Input, Single-Output type (**MISO**), **FFNN**. Single output refers to only one species.
  - Output of **training set** was achieved by **HSC 9.1**. The input of training set is made depending of the network type.

  For detail explanation of networks refer to Appendix.

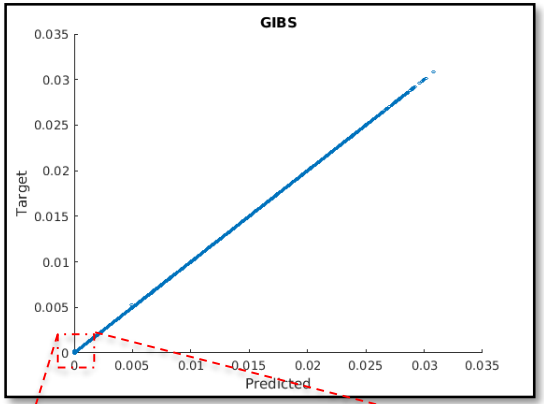
Predicted VS target (Full species prediction network)

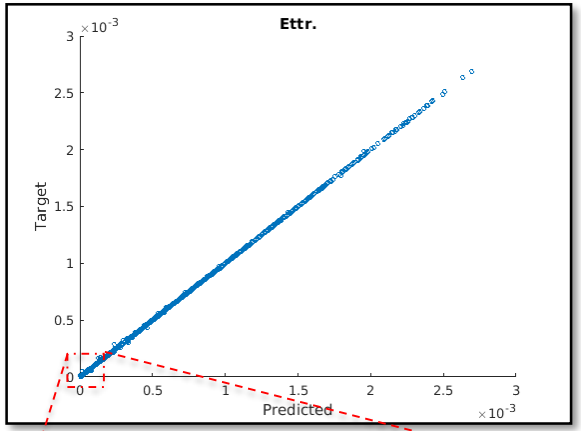


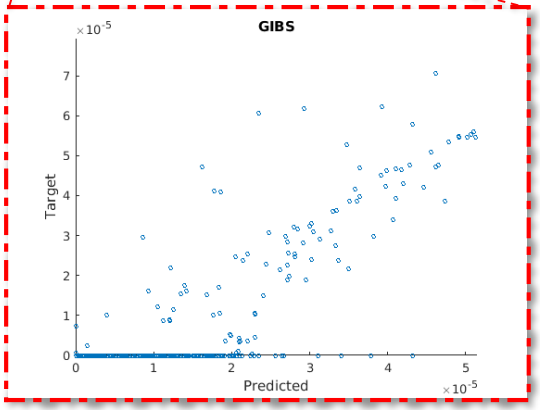


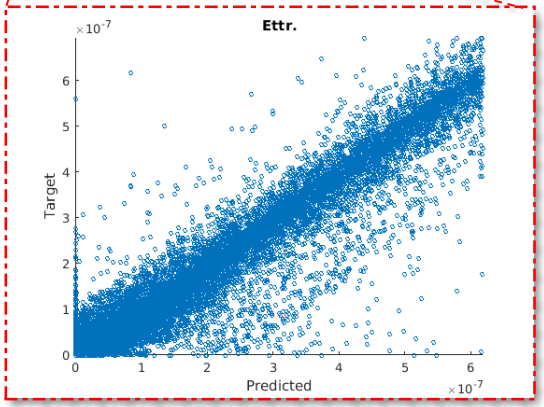
Predicted by FSN vs target values of each species

Predicted VS target (Full species prediction network)



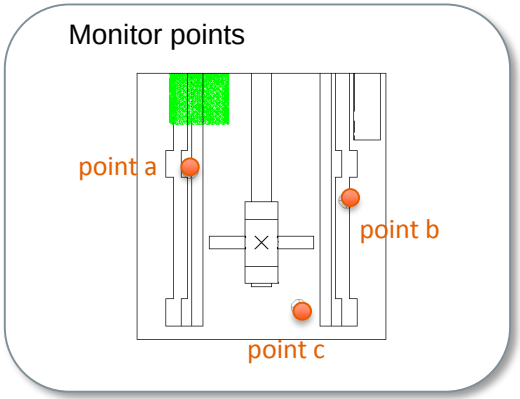


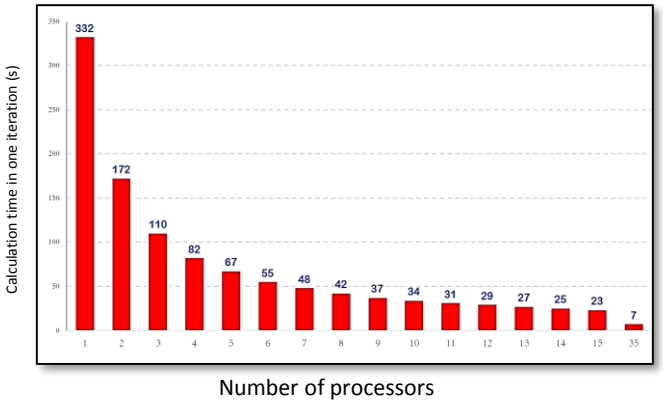




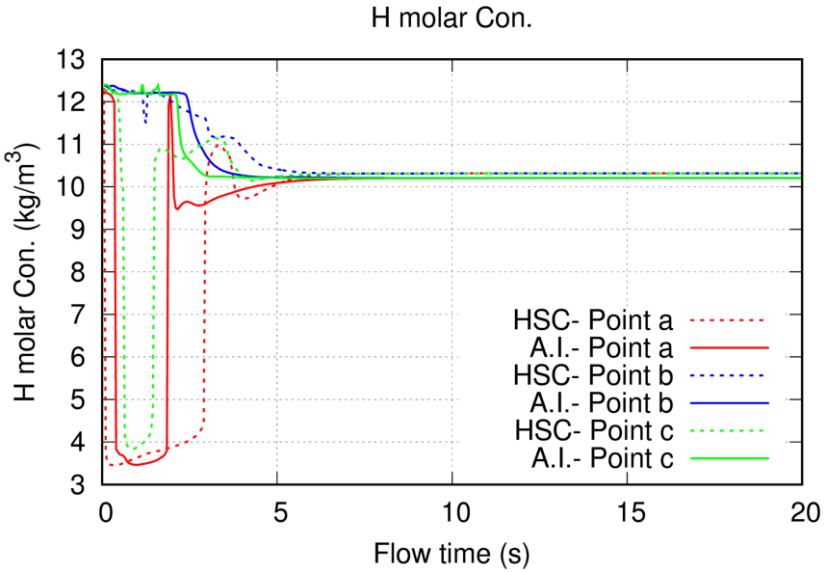
Gibbsite and ettringite contain zero values, log10 nonlinear transform over output can not be done, both nets would fail over low decimals of those species.

Some results





Computational time vs number of processor for full species prediction network, ran in UDF



Molar concentration of H^+ calculated in monitor points

Continuous 1-phase Ettringite process

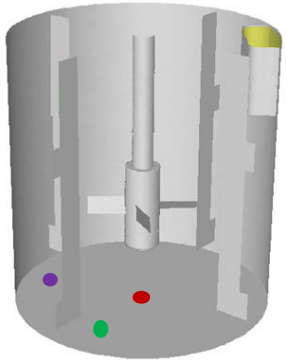
Continuous 1-phase Ettringite process

- Geometry:
  - Volume 5.25 L
- Grid size: 259 000 cells
- Temperature 295.75 K
- **400** rpm
- Flow rates:
  - CaOH2: 0.41 L/hour
  - Water: 9.46 L/hour
  - AlCl3: 0.12 L/hour

Thermodynamic equilibrium calculation,

- SO_4^{2-}, Ca^{2+}, OH^-, H^+, Al^{3+}, Cl^-, $Al(OH)_4^-$
- $Ca(OH)_2$ (portlandite), gibbsite, ettringite

Solution with $CaSO_4$, $Ca(OH)_2$



Water feed

$CaOH_2$ feed

$AlCl_3$ feed

Outlet

Continuous process:

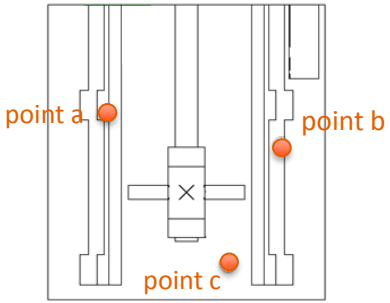
Computational Fluid Dynamics (CFD) modeling coupled with A.I and HSC chemical equilibrium calculation for simulating patch process.

- Simulations were ran with **single** phase (liquid) due to the reduction of computational time.
- The modelling of precipitation was based on an assumption of local ion equilibrium in the liquid one phase.
- The mixing process was modelled with single phase unsteady CFD including species transport and applying MRF (Moving Reference Frame) technique for modelling rotating mesh.
- Solving of thermodynamic equilibrium in system with several chemical components was carried out using two A.I. (in C++ and Matlab) and HSC 9.1, coupled to the precipitation kinetics in CFD (Ansys-Fluent 19.1).
  - Equilibrium was calculated with A.I. network and HSC 9.1 in every computational cell in every time step during the simulation
  - For Matlab code, Matlab-Runtime used to couple CFD and AI in this project
  - For HSC an interface file was coded to couple with Ansys-Fluent

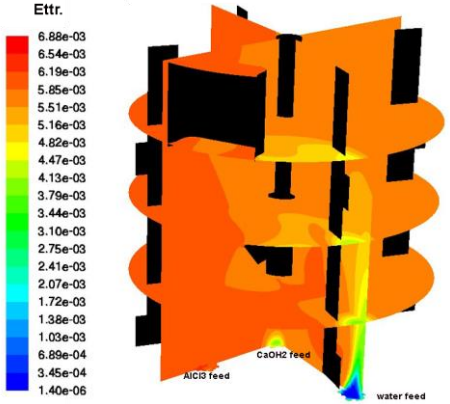
Some results

- The average value of PH in all monitor point were approximately 10.
- Average face values of SO_4 molar Con. over three planes were, 0.00074983 and 0.00074988 at 84.6 sec and 91.1 sec, respectively. This shows that the steady-state is reached.

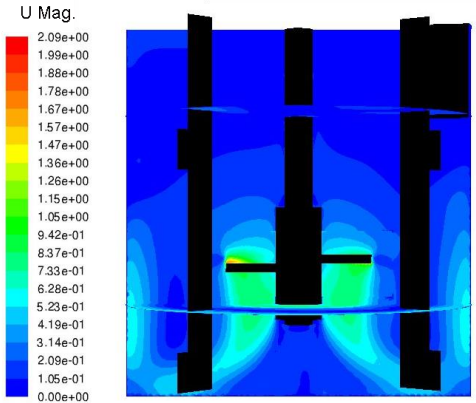
Monitor points

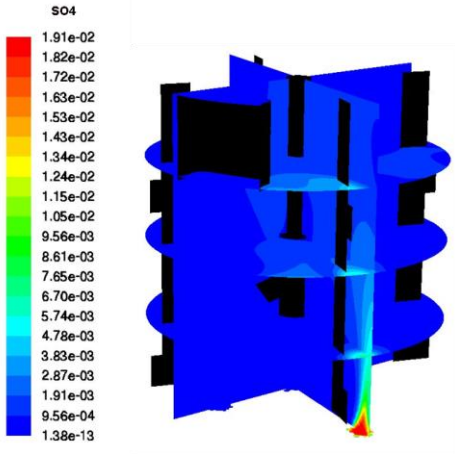


*Ettrignite molar Con. (*kmol/m3*)*



Velocity magnitude. (m/s)



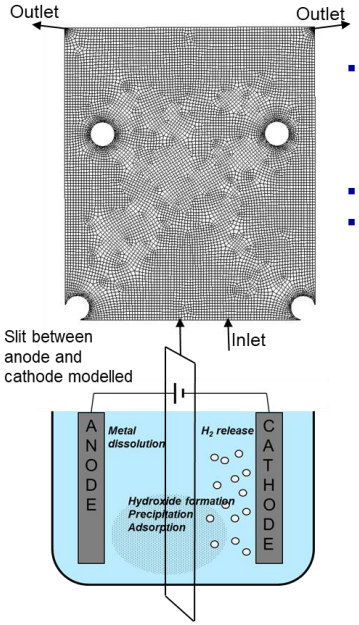


*SO4 molar Con. (*kmol/m3*)*

Time step: 18218, Flow time= 91.1 sec

Electrochemical Water Treatment (EWT) process

Creation of CFD model for the EWT process development



- Geometry:
  - Width 0.924 m
  - Depth 0.006 m
  - Height 1.085 m
- Grid size: 165 000 cells
- Temperature 288.15 K

Fe^{2+}, Fe^{3+}, Ni^{2+}, MnO_2^{2-}, H^+, OH^-, O_2, H_2, $HAsO_4^{2-}$, $H_2AsO_4^-$, H_3AsO_4, $Sb(OH)_4^-$, $Sb(OH)_3$

Anode: dissolution of metal ions by oxidation

Cathode: reduction of water into hydrogen (gas)

Bulk liquid: hydroxide formation, reactions with pollutants, precipitation, settling (or flotation) of particles

Electrochemical Water Treatment (EWT):

Computational Fluid Dynamics (CFD) modeling coupled with HSC Chemical equilibrium calculation for simulating precipitation process.

- A simplified CFD model for EWT process developed (Ansys Fluent 18.1):
  - Time dependent simulation with species transport
  - Single phase approach
  - Precipitating compounds: $FeSbO_3$, $FeAsO_4$, $Mn(OH)_2$, $Ni(OH)_2$, $Fe(OH)_2$
- Thermodynamic equilibrium calculated with HSC 9.1 at the end of the time step within every computational cell
  - Interface for coupling CFD and HSC developed (HSC 9.1)
- The effect of electric forces on the ionic species transport modelled with user defined functions (UDF).
- The model developed for the quite complex process is simplified, but nevertheless contains the most important elements needed to describe the process
- However, the EWT process is not the best application for the chemical equilibrium model, because it is known that the system does not reach full equilibrium.

Constrained reactions with CFD

- If kinetic rates were applied for the precipitates formation, every precipitate (a total amount of five precipitates) would have to be defined as solid phase, or as a new specie compound
  - Computational time needed would increase due to the complexity
  - The coupled simulation was already quite time consuming

- Tests were made in order to restrict the reactions with UDFs (UserDefinedFunctions)

1. Equilibrium with reduced components was calculated with HSC, and the rest of the components were calculated by UDFs
   - Faster simulation due to the reduced amount of component with HSC:

     $[Fe^{2+}], [Fe^{3+}], [H^{+}], [OH^{-}], [Fe(OH)_2]$

   - The resulting species concentrations for H^{+} and OH^{-} were not necessarily realistic, and need further studies

2. Only a predefined part of the specie concentrations were provided for equilibrium calculation with HSC

   $\boldsymbol{A}[MnO_2^{2-}], \boldsymbol{B}[H_3AsO_4], \boldsymbol{C}[Sb(OH)_3], \boldsymbol{D}[Ni^{2+}]$

   - Parameter fitting needed depending on the residence time of component in a reactor.

Conclusion and future works

Conclusion and future works

- Since finding chemical equilibrium of species using Gibbs minimization methods is in general time consuming, we have developed two A.I. networks (FSN & PSN) to calculate these equilibriums.

- Running in parallel, the resulting A.I networks are very fast and computationally inexpensive, particularly PSN network, in comparison to HSC chemical equilibrium solver.

- The main limitation of our networks is that there is no guarantee in preserving chemical mass balance. For this purpose further studies required to define a loss function that preserve this. Another limitation is that the network are case base, meaning that for new set of chemical reactions, new training is needed.

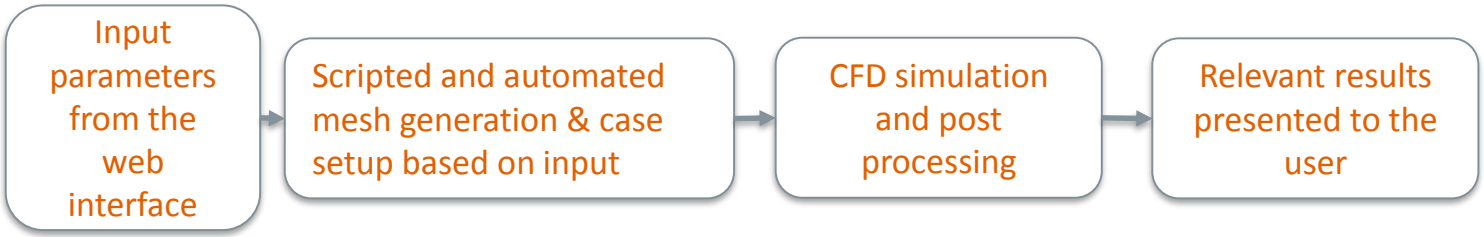
- We conclude that A.I. can be useful for chemical equilibrium. However further investigations are needed.

Ideas on modelling factory platform

Coupling to Modelling Factory platform

- The A.I. networks were designed and trained in MATLAB and HSC, for evaluation only the MATLAB-Runtime is needed
  - Straightforward to deploy pre-trained models to the platform
  - Evaluation also possible with pure python or C++, no external requirements
- For CFD simulations open source CFD code OpenFOAM offers an alternative to ANSYS-Fluent
  - No license fees, has comparable functionality
  - Easily scriptable, case configuration through text files
  - Contains tools for automatic, parametrized mesh generation, pre- and post processing

Example of automated workflow



Questions?

Appendix

Feed-forward neural network

Step 1: Initialization

- Weight and biases

Step2: Feed-Forward

- Calculation of H_1, H_2

$$H_1 = I_1W_1 + I_2W_3 + B_1W_5$$
$$H_2 = I_1W_2 + I_2W_4 + B_1W_6$$

- Use an nonlinear activation function such as sigmoid over hidden layers

$$HA_1 = \frac{1}{1+e^{-H_1}}$$
$$HA_2 = \frac{1}{1+e^{-H_2}}$$

- Output node calculation and Activation over output node (generally linear)

$$O_1 = HA_1W_7 + HA_2W_9 + B_2W_{11} \qquad OA_1 = O_1$$
$$O_2 = HA_1W_8 + HA_2W_{10} + B_2W_{12} \qquad OA_2 = O_2$$

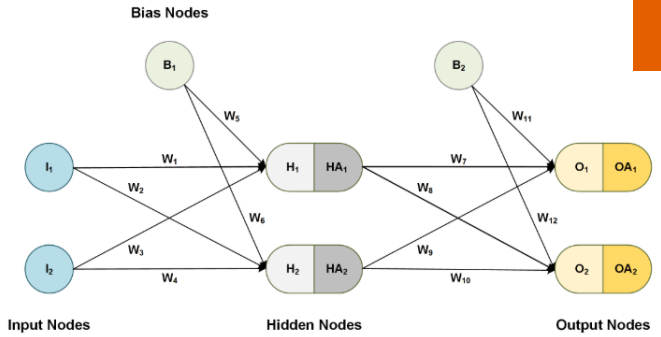
- Error calculation

$$e = \frac{1}{n}\sum_{i=1}^{2}\left(y_i - OA_i\right)^2$$

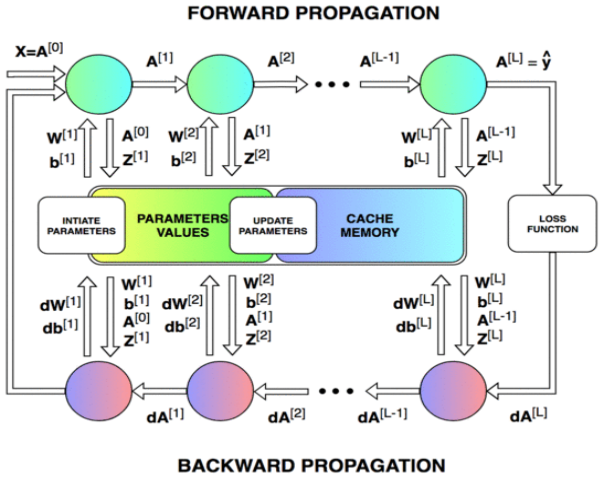
Step 3: Backpropagation

$$W_i^{k+1} = W_i^k - \eta\frac{\partial e}{\partial W_i^k}$$

The derivatives are calculated using chain rules.



Example of a shallow neural network



Summary of FFNN

Problem with A.I.?

- Mathematically **hard** or **impossible** to **explain** the architecture of a designed network. (The creators of the best deep networks performing very well now are unable to fully explain the network, even CNN)
- **No** knowledge of **amount** of **training data** needed that a network get trained well.
- **No** certain method regarding **number** of neurons, hidden layers, activation functions and initialization of weights and biases.
- **High** possibility of **wrong** solution over **not seen** data.
- **High precision** achievement of data might lead to **over fitting** of the network. (A.I. memorizes the data rather than learning it)
- **Training** of the data might take very long time and computationally very **expensive**.

Description of our networks

- Using multi-input, multi-output (**MIMO**) A.I could not trained since species were in different numerical order. Training was very time consuming and the results were **not** in high accuracy. Increasing number neurons, layer and etc. did not help in achievement of a unified network. That's the reason we moved to MISO.

Full species prediction network (FSN)

- Input: SO_4^{2-}, Ca^{2+}, OH^-, H^+, Al^{3+}, Cl^-, $Al(OH)_4^-$, $Ca(OH)_2$ (portlandite), gibbsite, ettringite
- Input generation: from CFD-HSC case, random data caused training impossible with any and further configuration.
- Prediction over all except Cl^- (not changing during the simulation), $Al(OH)_4^-$ (very small values, negligible)
- Nonlinear reversible preprocessing output transformation: log except gibbsite and ettringite. Without considering this nonlinear function over input data, the network is unable to predict the low decimals (e.g. 10^{-30})
  - Gibbsite and ettringite contain zero values, this approach would fail over those species, undefined log). $\log(\alpha + x)$ did not solved the problem.
- Designed in **MATLAB** due to **Levenberg–Marquardt** optimizer. (high accurate)

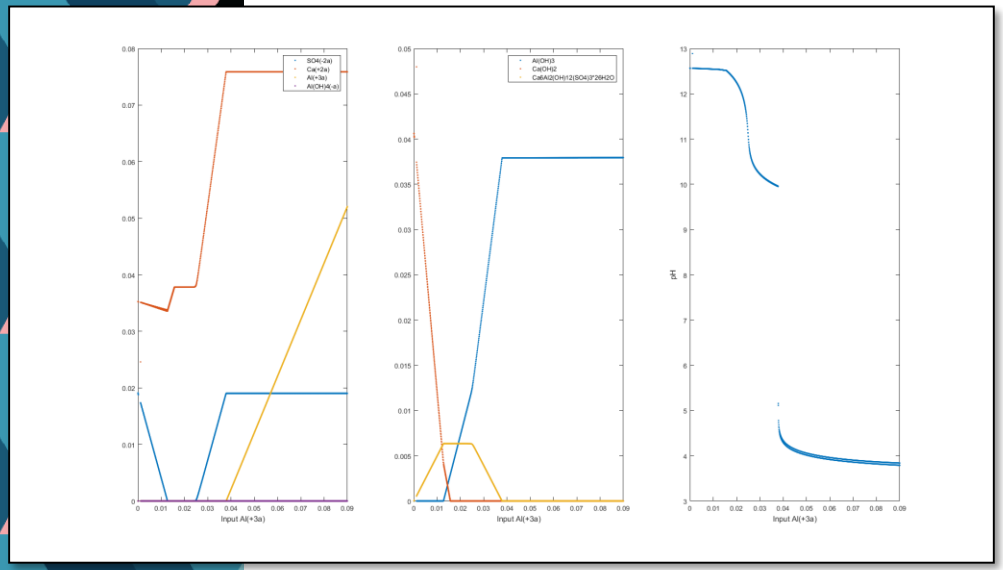
partial species prediction network (PSN)

- Input: Total values of SO_4, Ca, OH, H, Al
- Input generation: random species data
- Output: OH^-, H
- Nonlinear reversible preprocessing output transformation: log for H
- Network is designed in Matlab but evaluation in C++.

Both networks are designed to **not** have **overfitting**.

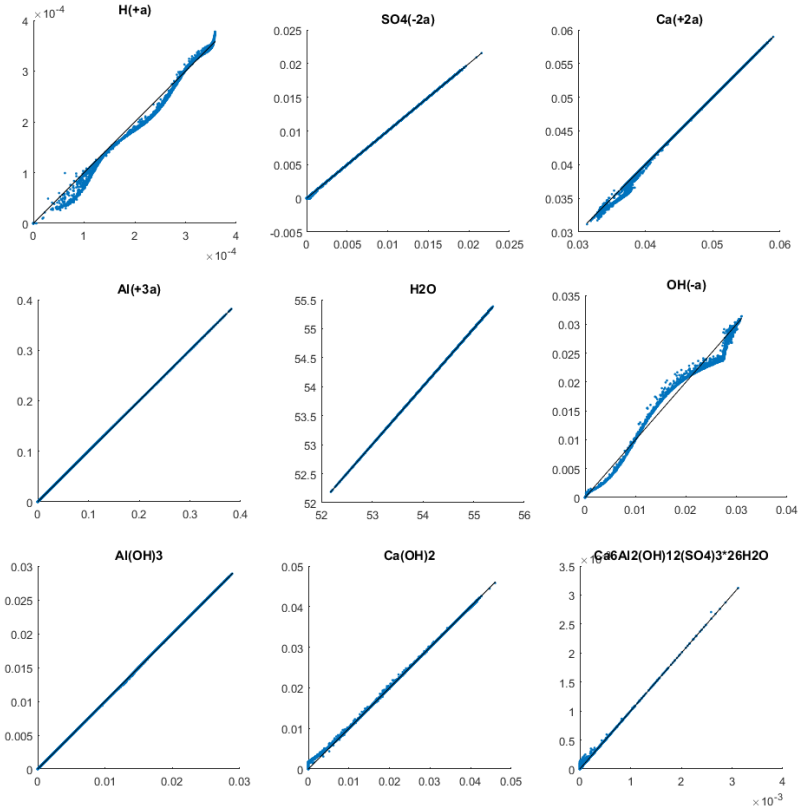
PSN is very **fast** in comparison to FSN and favor to **random** data, however **less accurate** with respect to **FSN**.

Predicted VS target (partial species prediction network)



In PSN, all species except OH and H are found based on the above relations. Using total values of species and two separate network, OH and H equilibriums are calculated.

Although PSN is less accurate, it is more general due to random training set and faster. (less than 1sec computational time)



Predicted by PSN vs target values of each species