Etching – 12-13 May

Etching is a technique used in printmaking and microfabrication to create patterns on a surface. In etching, a metal plate or other surface is covered with a protective coating, called a resist. The design is then drawn onto the resist, and the unprotected areas are exposed to an acid or other etching agent. The etching agent eats away at the exposed surface, creating a recessed design. The resist is then removed, and the resulting etched design can be used to create a print or other object

PR (photoresist material) and etchant(acid)

Paper-1:

Modeling of plasma dry etching (with **HAR** (high aspect ratio)>30) -> TCAD (Technology Computer aided design) – DL (seq2seq) – DL with physics informed model

Paper-2 (plasma etching)

The most devices of semiconductor enforce electrical isolation to prevent leakage currents. One of the ways to impose electrical isolation is called the Deep Trench Isolation (**DTI**) – by TCAD – baseline model (transformer) – inductive biases

Paper-3

Bayesian neural network with TCAD prior

Paper-4

HAR deep trench etching

Monte carlo – statistical method for predicting reaction

Paper-5 - Modeling and Control of a Chemical Process Network Using Physics-Informed Transfer Learning

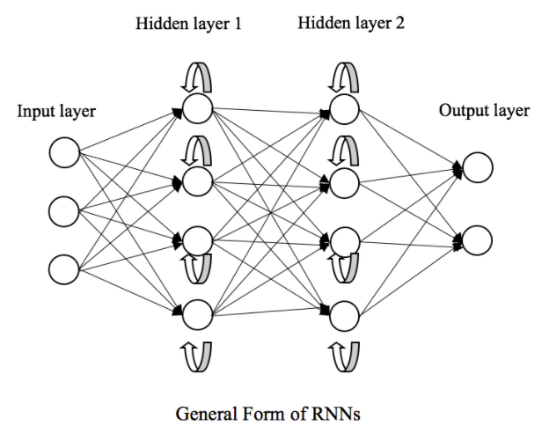
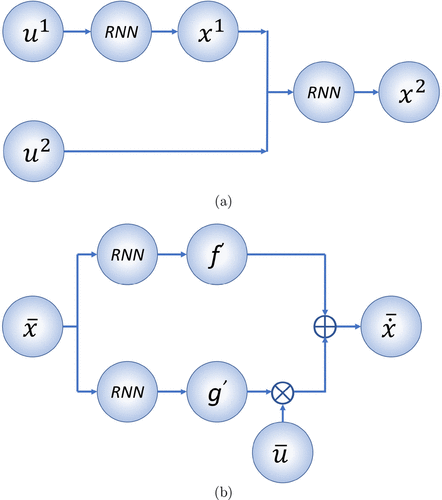
Machine learning modeling of a large-scale chemical process network generally requires a large amount of training data, to improve it transfer learning is used.

To address the problem of data scarcity, physics-informed machine learning is shown to have the potential to improve the prediction performance by embedding domain knowledge into the training process

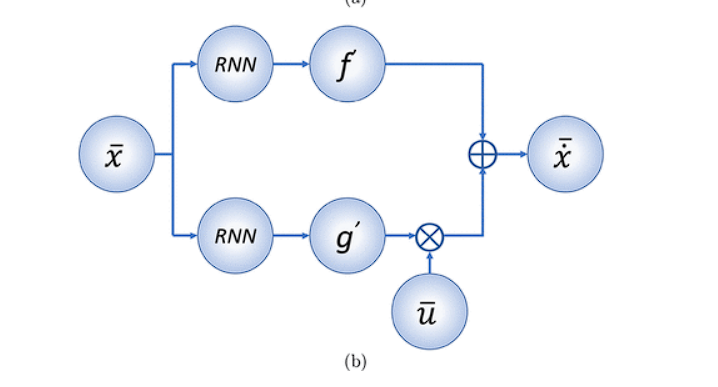
Physics-informed machine learning –

14 May

Fully connected RNN v/s Partially connected one

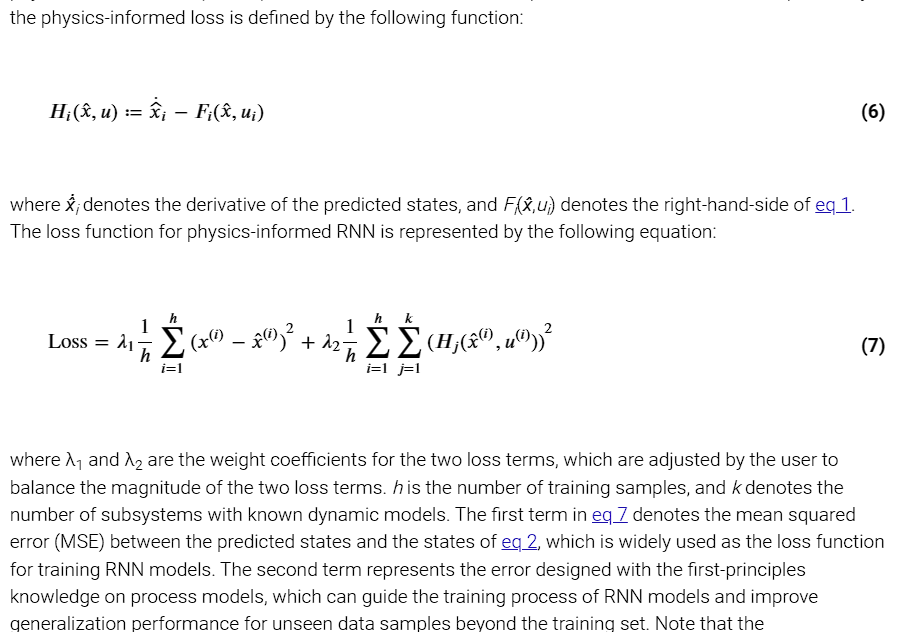
 While partially rnn won’t have each neuron connected to other neuron

* Modeling of a Process Network through Transfer Learning:
  + **Data-Based Transfer Learning:** The weights in the pretrained model can provide a good initial ‘guess’ for the parameters in the target RNN model. When the data from the target process is insufficient, the transfer learning model based on the pretrained model, which is trained with sufficient data from the source domain, can improve the prediction accuracy compared with the conventional RNN model that is developed using data only and does not have any knowledge from the source process. For example, fine-tuning the RNN weights with the initial values chosen based on the pretrained model using target process data can improve the modeling prediction accuracy. In ref (32), we developed transfer learning models for a single unit operation. - given the target process network of eq 2 of n subsystems, transfer learning is implementable provided that a source process network with the same number of subsystems and similar configurations in terms of unit operations involved in the network is available to generate massive training data sets for developing a pretrained model. In this case, we can follow the purely data-driven transfer learning method
  + **Physics-Informed Transfer Learning:** Therefore, based on the assumption that the pretrained models for a portion of the process network (i.e., m subsystems) are available, our goal is to develop an RNN model for the entire nonlinear process network of eq 2 of n subsystems, where m < n. - Domain knowledge such as process structure knowledge and first-principles knowledge can be used with transfer learning to develop RNN models for the entire process network. -However, since the subsystems may interact with each other, a new architecture of RNN model needs to be designed to incorporate the pretrained models for subsystems into the overall process model. To that end, a partially connected RNN model (PCRNN) that integrates the structure knowledge of the process network into the design of RNN architecture is introduced. Additionally, to improve the prediction accuracy in the presence of limited training data, we develop the physics-informed RNN (PIRNN) model by further integrating first-principles knowledge into the development of PCRNNs
  + **Transfer Learning Based Partially Connected RNN (TL-PCRNN):**

****

The first part is used to learn the dynamic behavior of the m subsystems, which is imported from the pretrained RNN models. For the remaining subsystems, the initial weights are chosen randomly. – M subsystems of source and target are same, so freeze the weights of pre trained model and continue improving remaining subsystems accuracy and when sufficient accuracy is achieve, then recompile all of the subsystems as a whole. - One of the benefits of freezing the weights for the m subsystems at the initial stage is that the training process can focus on minimizing the loss term corresponding to the remaining subsystems such that the overall prediction accuracy is improved quickly.

* + **Transfer-Learning-Based Physics-Informed RNN (TL-PIRNN):** In addition to a priori knowledge on process structure, domain knowledge of a chemical process network, such as physical laws, can also be utilized to improve the accuracy of the RNN model by introducing the physics-informed loss (PI-loss) term in the loss function in the presence of insufficient data. - To develop transfer-learning-based PIRNN (TL-PIRNN) model, the pretrained models for the m subsystems are integrated into the partially connected architecture of RNN models following the method in the previous section, and eq (below) is used as the loss function for the training process



Additionally, the loss function of eq 7 is used to optimize the weights in TL-PIRNN. After N(PI i.e physics-informed) epochs, all the weights are set to be ‘trainable’, and the PI-loss term is removed from the loss function. Finally, the RNN model is recompiled, and all the weights are updated using the standard MSE loss function only.

* Closed Loop Control Using TL methods:

To implement the RNN-based controller, a stability region should be characterized to ensure stable operation of the nonlinear process network for any initial conditions in this region. One way to approximate the stability region is to evaluate the time derivative of the Lyapunov function V over the entire state-space, and choose a set of states where V̇ can be rendered negative under a stabilizing controller. However, since the search space grows exponentially with the number of subsystems, it is computationally expensive to characterize the stability region for the entire process network. Therefore, in this section, we use the transfer learning method to accelerate the search process by taking advantage of the knowledge on the stability regions predetermined for the m subsystems with sufficient training data. Therefore, to develop a stability region for the entire process network of eq 2, the search space is limited to the remaining subsystems, i.e., n – m subsystems, and the computation time decreases exponentially using the knowledge of the predetermined stability region Dm transferred from the source process model for the m subsystems.

* EXAMPLE Case Studies:

The first case study is used to show that the TL-based RNN models achieve the desired prediction accuracy for the process network with limited data. Then, the computational benefits of the knowledge-based method for developing stability regions is illustrated by the process network in the second case study.

We have two CSTR(Conti. Stirred Tank Reactor) having reacn. A+B->C & so CSTR-1 is independent and CSTR-2 is dependent on CSTR-1 as they are serial.

* + C-1: TL-PCRNN – CSTR-1 have been trained on 1,36,000 data and tested on 24,000 data, now CSTR-2 has dataset of 10,000 div. as 5,000 for each which is low so model trained from CSTR-1 is used here which acts base at first hidden layer and are made frozen until some number of epochs.

TL-PIRNN - Based on the design of partially connected network structure, the first-principles knowledge of the second CSTR unit is further embedded in the development of RNN models to develop the TL-PIRNN model for the entire process network of eqns. Specifically, the first-principles model of the second CSTR is assumed to be known and utilized in the design of the loss function, in order to improve the generalization performance of the TL-PIRNN model beyond the training data set.

TL-PCRNN 2 and TL-PIRNN 2 are also ran where no freezing is done.

* + C-2: Case Study 2: Characterization of Stability Region via Transfer Learning –

To figure out how stable the system is, we use something called Lyapunov functions and simulate how they change over time for different starting conditions of the reactors. If these functions show that the system remains stable, we include those starting conditions in what we call the stability region.

We first find the stability region for one of the reactors and then use that knowledge to speed up finding the stability region for the whole system. This method saves a lot of time compared to searching for stability in the whole system from scratch.

Installing ViennaPS:

* Cloned and pulled git repo.
* Ran cmake -B build && cmake --build build, fixed all of the bugs temporarily and on Visual Studio specifically
* Ran cmake --install build --prefix inst which showed an error that some of lib and dll files are missing, so ran cmake --install build --prefix inst –config cfg which provides flexibility when installing artifacts for different build configurations, allowing you to control which version of your project is installed based on your needs and still while running cmake --install build --prefix inst –config Release I am getting the same error.
* After little tweakings and getting all libraries from cmake --install build --prefix inst –config cfg, I started to build app/application.cpp using gcc command which also showed many errors like unkwown header files which were solve by using -I <all header file locations> option in gcc command, and also did some tweakings in lines of code to temporarily run gcc command properly.