Integrating Scientific Theory with Machine Learning

Miodrag Bolic

Health Devices Research Group (HDRG)
School of Electrical Engineering and Computer Science
University of Ottawa

March 27, 2021

Outline

Current approaches in modeling

Machine learning

- Can find patterns in problems where complexity prohibits the explicit programming of a system's exact physical nature.
- Not enough data to train sufficiently generalized models.
- Purely data-driven model might not meet constraints such as dictated by natural laws
- Need for models to be interpretable and explainable
- Require homogeneous labeled training data

Mechanistic models

- "A picture is worth a thousand words" -¿ "A model is worth a thousand datasets" [7]
- Often, cannot capture complex dynamics in the system
- They are simplified to allow for handling complexity

Hybrid models

- Best of both worlds
- Can be complex and difficult to train

Presenting knowledge

- *General knowledge*: knowledge independent of the task and data domain.
- *Domain knowledge*: knowledge in any field such as biology, physics, chemistry, and engineering.

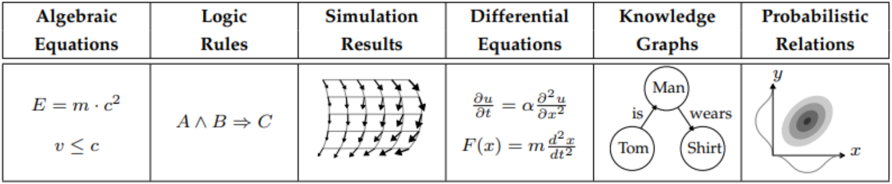
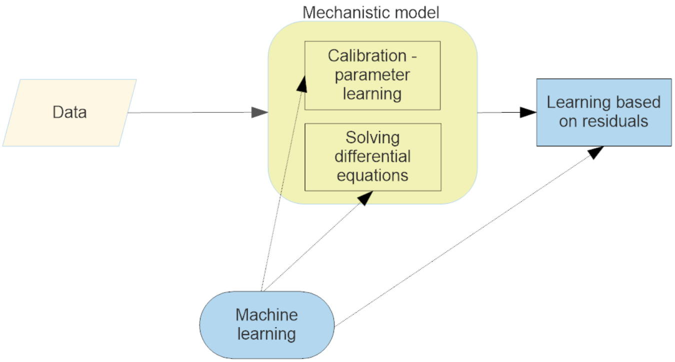


Figure: Domain knowledge representation [11] [1]

[1]Note that many figures are copied and the sources are referenced!

Mechanistic models augmented with ML



Information Flow of Informed ML

Informed machine learning describes learning from a hybrid information source that consists of data and prior knowledge. The prior knowledge is pre-existent and separated from the data and is explicitly integrated into the machine learning pipeline [11].

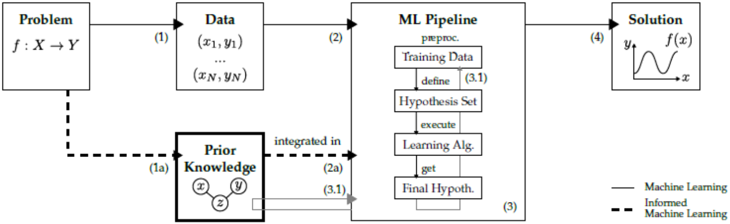
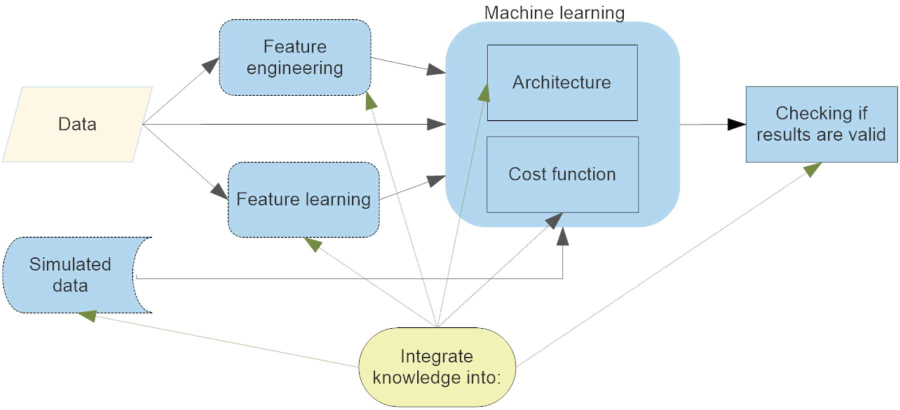


Figure: Machine learning flow [11]

Knowledge informed ML



Residual modeling

- The hypothesis: the obtained solution from the first principles model does not explain the system behavior -¿ created features are still well-correlated with the target variable (mismatch).
- ML aims to learn the residual pattern of the system behavior.

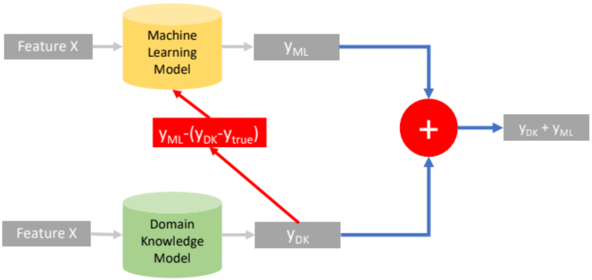


Figure: Residual modeling [4]

- Y_{ML}: machine learning predicted label
- Y_{DK}: domain knowledge predicted label
- Y_{true}: ground truth label

ML-based parameter optimization

- The ML model serves as an estimator of unmeasured process parameters that are difficult to model from first principles
- Used to model a fedbatch bioreactor [6]

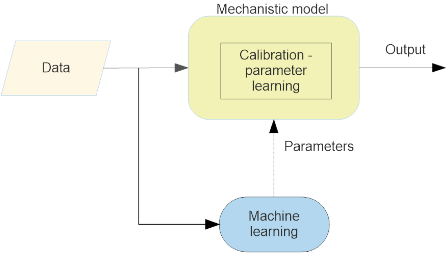


Figure: Parameter estimation

- Data: Biomass concentration at time k
- Output: Predicted biomass concentration at $k+1$
- Unobserved parameter: Growth rate

Differential equations

uOttawa

Infer solutions to partial differential equations, and obtain physics-informed surrogate models [9]:

- Neural networks can represent an arbitrary functions when given appropriate weights.
- Therefore it can approximate any arbitrary function that represents a solution of a differential equation: $u = NN(x)$
- We can also find du/dx, d^2u/dx^2 through back-propagation.
- Assume that we are given a differential equation with boundary conditions.
- The goal is to minimize the mean square error loss formed by differential equation and boundary conditions using automated differentiation.

Data and Training

uOttawa

Training data

- Features
  - feature engineering
- Data augmentation
  - image transforms
  - simulations: generate a large amount of data from mechanistic models for training.

Feature learning

- Unsupervised learning - knowledge can still be incorporated
- Variational autoencoders
  - VAEs jointly learn an inference model and a generative model, allowing them to infer latent variables from observed data.

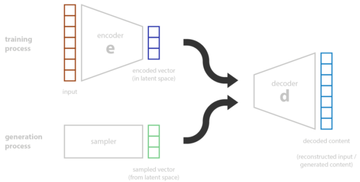


Figure: Understanding Variational Autoencoders

Learning algorithm

- $Mass = Density \cdot Volume$: ML does not know that this is not supposed to be violated
- Domain knowledge is into a loss function and performs regularization

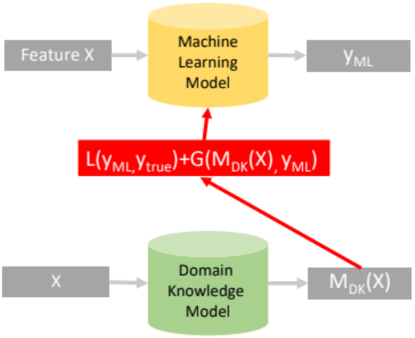


Figure: Knowledge in the loss function [4]

- function $L()$ is machine learning loss function
- function $G()$ is regularization term: a measure of consistency between domain knowledge and predicted label
- function $M_{DK}()$: a domain knowledge transformation of feature X

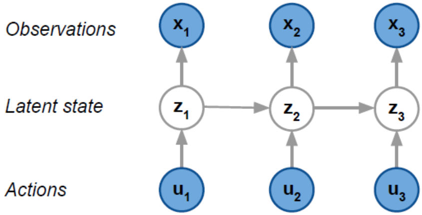
Architectures

- Model structure incorporates the mechanistic model
- We will introduce state-space models first and show how they can be integrated with RNNs
- Integration is done with variational autoencoders

State-space models

State-space models:

- probabilistic scenarios: We do not have perfect knowledge about the state of the system, and the system can develop non-deterministically over time
- are numerically efficient to solve
- can describe differential equations



- Transition equation:

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \epsilon_t$$

- Emission equation:

$$\mathbf{x}_t = \mathbf{C}_t \mathbf{z}_t + \boldsymbol{\delta}_t$$

Figure: Linear state-space model [3]

Filtering in state-space models

Kalman filter:

- Kalman filter is optimal for linear Gaussian problems.
- Generalizes many common time-series models
- Strong modelling assumptions:
  - Linear transitions and emissions
  - Gaussian transitions and measurement noise

Non-linear filters

- Extended Kalman filters (non-linear observation equation, Gaussian noise)
- Particle filters (non-linear, non Gaussian)
- Problems
  - Transition model still have difficulties handling complex non-linear dynamics
  - Does not capture long-term dependencies in data (Markov models)

Stochastic recurrent neural networks I

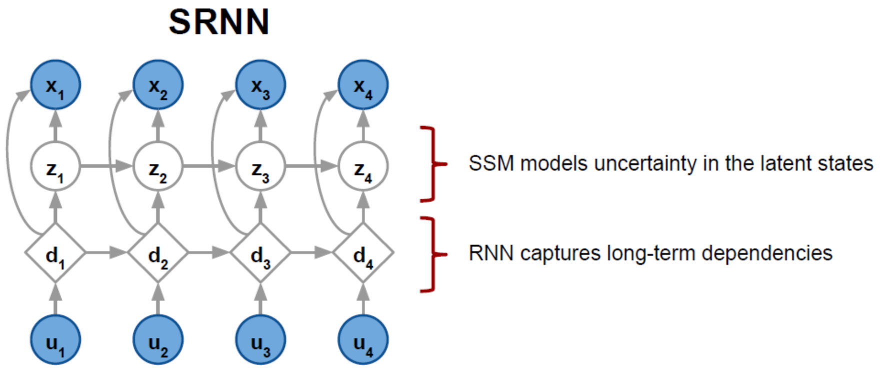


Figure: Merged RNN and state-space model [3]

Gaussian processes

- Each data points is a random variable generated from multivariate normal distribution
- The relationship between random variables determines the shape of the latent function.
- Advantages:
  - Regression and prediction with confidence intervals [10]
  - Learning the parameters of the state space models or differential equations [8]
  - Time series where data is not uniformly sampled.
  - Allow for Bayesian optimization

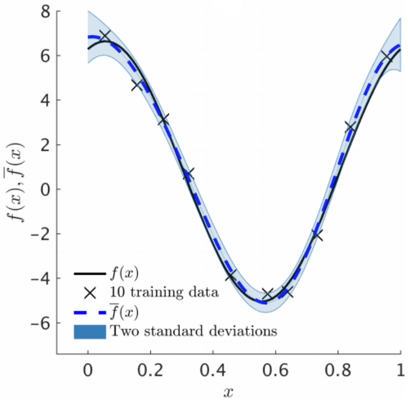


Figure: Gaussian process regression example [8]

Sequential decision making

uOttawa

Incorporating human knowledge into:

- Reward
- Policy and action selection
- Building the overall controller

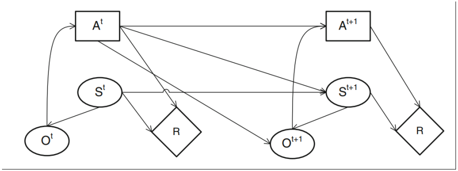


Figure: Probabilistic model of decision-making problem of an agent. The oval nodes represent the state (S) and the observation (O). The rectangle is the decision node (A) and the diamond is the reward function (R). [2]

Suggestion for research I

- Design a simulator that will allow us to simulate the bioprocess and bioreactor based on mechanistic or hybrid model [5]. This is important for:
  - digital twin
  - generating data for testing algorithms
- Merging mechanistic and ML models in this field has just started
  - there is great research opportunity to be first to apply some of these ML approaches on data from bioreactor.
- Gaussian processes for uncertainties and optimization

Suggestion for research II

- Pre-training or intelligent initialization of the parameters of the ML model
  - Transfer learning

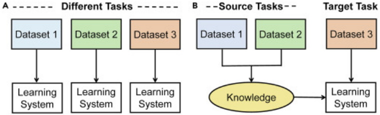


Figure: Transfer learning [1]

  - Meta learning
    - learning from other processes
    - from our data collected using different sensors and in different ways

Tools for scientific machine learning

Subject | AD Frameworks	ADIFOR or TAF	ADOL-C	Stan	Julia (Zygote.jl, Tracker.jl, ForwardDiff.jl, etc.)	TensorFlow	PyTorch
Language	Fortran	C++	Misc.	Julia	Python, Swift, Julia, etc.	Python
Neural Networks	neural-fortran	OpenNN	None	Flux.jl	Built-in	Built-in
Neural Differential Equations	Sundials (ODE+DAE)	Sundials (ODE+DAE)	Sundials (ODE+DAE)	DifferentialEquations.jl / DiffEqFlux.jl (ODE, SDE, DDE, DAE, hybrid, (S)PDE)	DifferentialEquations.jl (through Tensorflow.jl)	torchdiffeq (non-stiff ODEs)
	FATODE	PETSc TS	Built-in (non-stiff ODE)	Sundials.jl (ODE through DiffEqFlux.jl)		diffeqpy
Probabilistic Programming	None	CPProb	Built-In	Gen.jl	Edward	Pyro
				Turing.jl	PyMC4	pyprob
Sparsity Detection	Built-in (TAF)	Built-in	None	SparsityDetection.jl	None	None
Sparse Differentiation	Built-in (TAF)	Built-in	None	SparseDiffTools.jl	None	None
GPU Support	CUDA	CUDA	OpenCL	CUDANative.jl + CuArrays.jl	Built-in	Built-in
Distributed Dense Linear Algebra	ScaLAPACK	Elemental	None	Elemental.jl	Built-in	torch.distributed (no factorizations)

Scale	None	Poor	Fair	Excellent
Explanation	No automatic differentiation compatible library exists. Suggestion for a library to wrap.	Functionality exists, but is feature-incomplete or AD compatibility is incomplete. If no AD support, then AD support can easily be added since the library already defines adjoints.	The basic features exist, but has some major features missing or are not AD-compatible.	Has all of the main features and is fully compatible with the automatic differentiation tooling.

Figure: Comparison of tools readily usable with differentiable programming (automatic differentiation) frameworks copied from here

References I

[1] Changyu Deng, Xunbi Ji, Colton Rainey, Jianyu Zhang, and Wei Lu.
Integrating machine learning with human knowledge.
iScience, 23(11):101656, 2020.

[2] P. Doshi, Y. Zeng, and Q. Chen.
Graphical models for interactive pomdps: representations and solutions.
Auton Agent Multi-Agent Syst, 18, 2009.

[3] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther.
Sequential neural models with stochastic layers.
In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2207–2215, Red Hook, NY, USA, 2016. Curran Associates Inc.

[4] Xiaowei Jia.
Ml applications in scientific domains, 2020.

[5] Harini Narayanan, Lars Behle, Martin F. Luna, Michael Sokolov, Gonzalo Guillén-Gosálbez, Massimo Morbidelli, and Alessandro Butté.
Hybrid-ekf: Hybrid model coupled with extended kalman filter for real-time monitoring and control of mammalian cell culture.
Biotechnology and Bioengineering, 117(9):2703–2714, 2020.

References II

[6] D. C. Psichogios and Lyle H. Ungar.
A hybrid neural network-first principles approach to process modeling.
Aiche Journal, 38:1499–1511, 1992.

[7] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman.
Universal differential equations for scientific machine learning, 2020.

[8] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis.
Machine learning of linear differential equations using gaussian processes.
Journal of Computational Physics, 348:683–693, Nov 2017.

[9] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis.
Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.

[10] P. Swain, K. Stevenson, and A. Leary et al.
Inferring time derivatives including cell growth rates using gaussian processes.
Nat Commun, 7, 2016.

References III

[11] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker.
Informed machine learning – a taxonomy and survey of integrating knowledge into learning systems.
2020.

Recommended videos

- Probabilistic data fusion and physics-informed machine learning: A new paradigm for modeling and computation under uncertainty
- ML applications in scientific domain