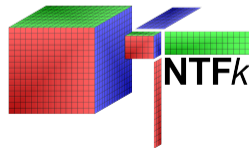
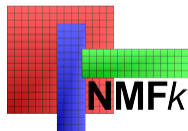


Unsupervised and Physics-Informed Machine Learning of Big and Noisy Data

Velimir V. Vesselinov (monty) (vvv@lanl.gov)

Earth and Environmental Sciences Division
Los Alamos National Laboratory, NM, USA

<http://tensors.lanl.gov>



- ▶ **Supervised** ML: learns everything from data
 - ⇒ requires big training datasets
 - ⇒ highly impacted by noise
- ▶ **Physics-informed** ML: learns from data but includes preconceived knowledge about the governing processes
 - ⇒ requires smaller training datasets
 - ⇒ produces better predictability with lower uncertainty
 - ⇒ robust to data noise
- ▶ **Unsupervised** ML: extracts features from data that can be applied for categorization and prediction
 - ⇒ unbiased analyses not impacted by data labeling, subject-matter-expert opinions, and physics assumptions ⇒ however, physics constraints can be added

- ▶ **Supervised** ML: learns everything from data
 - ⇒ requires big training datasets
 - ⇒ highly impacted by noise
- ▶ **Physics-informed** ML: learns from data but includes preconceived knowledge about the governing processes
 - ⇒ requires smaller training datasets
 - ⇒ produces better predictability with lower uncertainty
 - ⇒ robust to data noise
- ▶ **Unsupervised** ML: extracts features from data that can be applied for categorization and prediction
 - ⇒ unbiased analyses not impacted by data labeling, subject-matter-expert opinions, and physics assumptions ⇒ however, physics constraints can be added

- ▶ **Supervised** ML: learns everything from data
 - ⇒ requires big training datasets
 - ⇒ highly impacted by noise
- ▶ **Physics-informed** ML: learns from data but includes preconceived knowledge about the governing processes
 - ⇒ requires smaller training datasets
 - ⇒ produces better predictability with lower uncertainty
 - ⇒ robust to data noise
- ▶ **Unsupervised** ML: extracts features from data that can be applied for categorization and prediction
 - ⇒ unbiased analyses not impacted by data labeling, subject-matter-expert opinions, and physics assumptions ⇒ however, physics constraints can be added

- ▶ **Supervised** ML: requires “labeling” (prior categorization (knowledge) about the processed data)

Example: Recognizes images of cats and dogs after extensive training; but cannot recognize horses if not trained

Cannot discover something that we do not know already

- ▶ **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis**; **data-driven science**)

Example: Identifies features that distinguish images of animals (e.g., cats, dogs, horses, etc.)

- ▶ **Supervised** ML: requires “labeling” (prior categorization (knowledge) about the processed data)

Example: Recognizes images of cats and dogs after extensive training; but cannot recognize horses if not trained

Cannot discover something that we do not know already

- ▶ **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis**; **data-driven science**)

Example: Identifies features that distinguish images of animals (e.g., cats, dogs, horses, etc.)

- ▶ **Supervised** ML: requires “labeling” (prior categorization (knowledge) about the processed data)

Example: Recognizes images of cats and dogs after extensive training; but cannot recognize horses if not trained

Cannot discover something that we do not know already

- ▶ **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis**; **data-driven science**)

Example: Identifies features that distinguish images of animals (e.g., cats, dogs, horses, etc.)

- ▶ **Supervised** ML: requires “labeling” (prior categorization (knowledge) about the processed data)

Example: Recognizes images of cats and dogs after extensive training; but cannot recognize horses if not trained

Cannot discover something that we do not know already

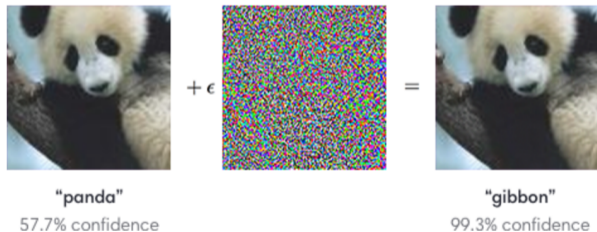
- ▶ **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis**; **data-driven science**)

Example: Identifies features that distinguish images of animals (e.g., cats, dogs, horses, etc.)

- ▶ **Supervised** ML: requires “labeling” (prior categorization (knowledge) about the processed data)
Example: Recognizes images of cats and dogs after extensive training; but cannot recognize horses if not trained
Cannot discover something that we do not know already
- ▶ **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis**; **data-driven science**)
Example: Identifies features that distinguish images of animals (e.g., cats, dogs, horses, etc.)

► Supervised ML

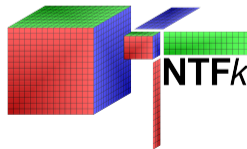
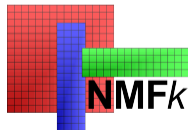
- introduces subjectivity (through the labeling process)
- does not provide insights why horses are different from dogs / cats
- cannot make predictions (that we do not know already)
- requires huge training (labeled) datasets
- we do not know why it works
- is impacted by “adversarial examples”



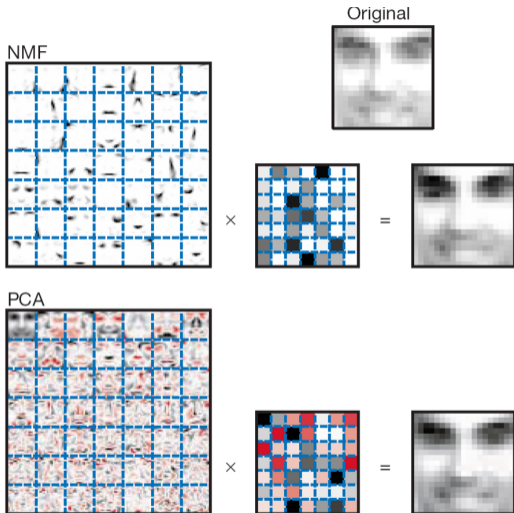
⇒ major limitations of the **supervised** ML methods for **science** applications

- ▶ Feature extraction (**FE**)
- ▶ Blind source separation (**BSS**)
- ▶ Detection of disruptions / anomalies
- ▶ Image recognition
- ▶ Separate physics processes
- ▶ Discover unknown dependencies and phenomena
- ▶ Develop reduced-order/surrogate models
- ▶ Identify dependencies between model inputs and outputs
- ▶ Guide development of physics models representing the data
- ▶ Make predictions
- ▶ Optimize data acquisition
- ▶ “Label” datasets for supervised ML analyses

- ▶ Novel LANL-patented, open-source, unsupervised Machine Learning (ML) methods and computational techniques
- ▶ Based in matrix/tensor factorization coupled with custom k -means clustering and nonnegativity/sparsity constraints:
 - NMF $_k$: Nonnegative **Matrix** Factorization
 - NTF $_k$: Nonnegative **Tensor** Factorization
 - <https://github.com/TensorDecompositions>
- ▶ Capable to efficiently process large datasets (TB's) utilizing GPU's, TPU's & FPGA's
⇒ **Julia**, Flux.jl, AutoOffLoad.jl, TensorFlow, PyTorch, MXNet

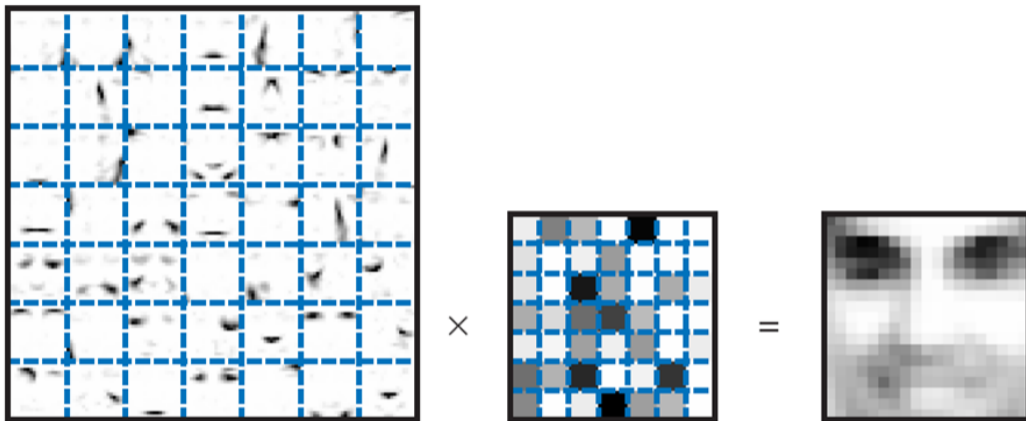


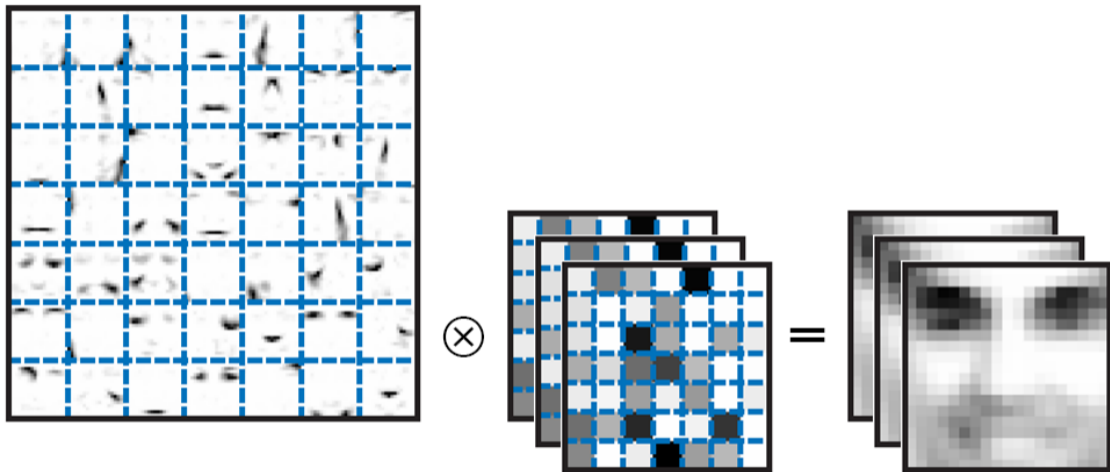
- ▶ NMF vs PCA (Lee & Seung, 1999)
- ▶ NMF: Nonnegative Matrix Factorization
- ▶ PCA: Principal Component Analysis

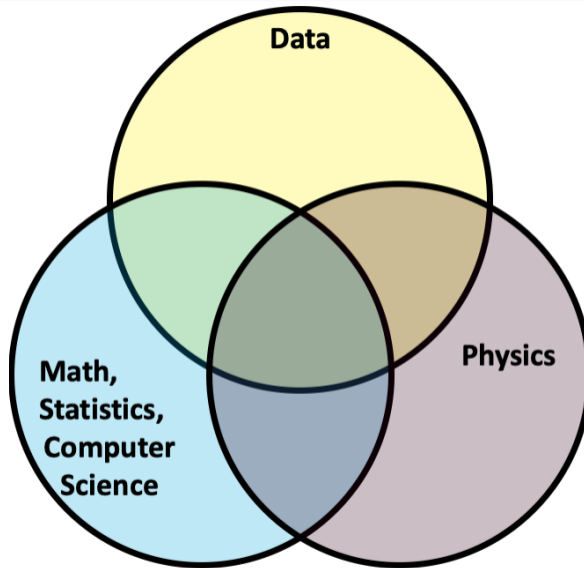


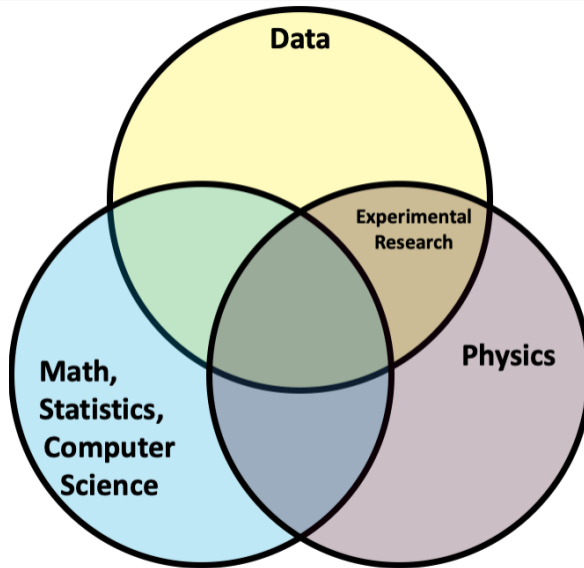
Nonnegativity constraints provide meaningful and interpretable results (+sparsity)

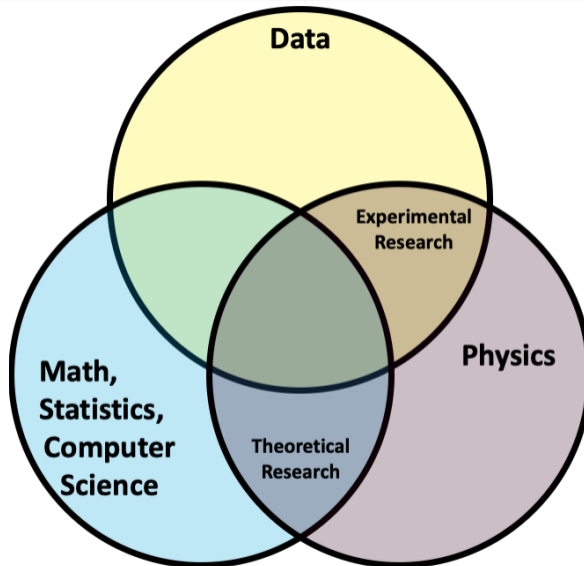
- ▶ **Tensors** (multi-dimensional/multi-modal/multi-way datasets) are everywhere:
 - ▶ observational data are typically a 5-D tensor ($x, y, z, t, \text{attributes}$)
 - ▶ model outputs are typically a 5-D tensor ($x, y, z, t, \text{attributes}$)
 - ▶ data dependency to N parameters will form a $(N + 5)$ -D tensor

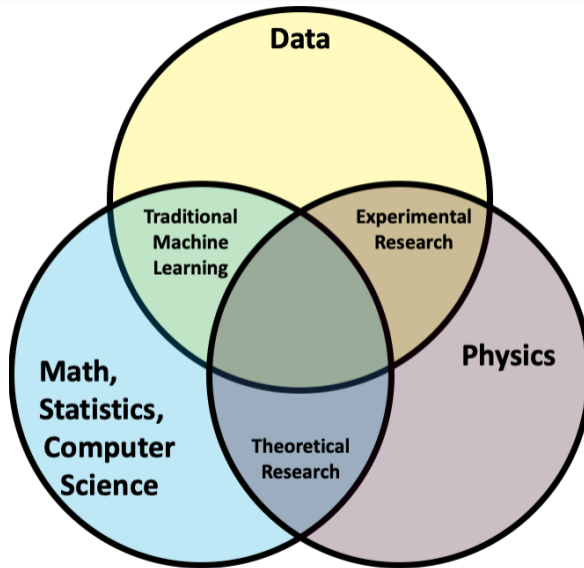


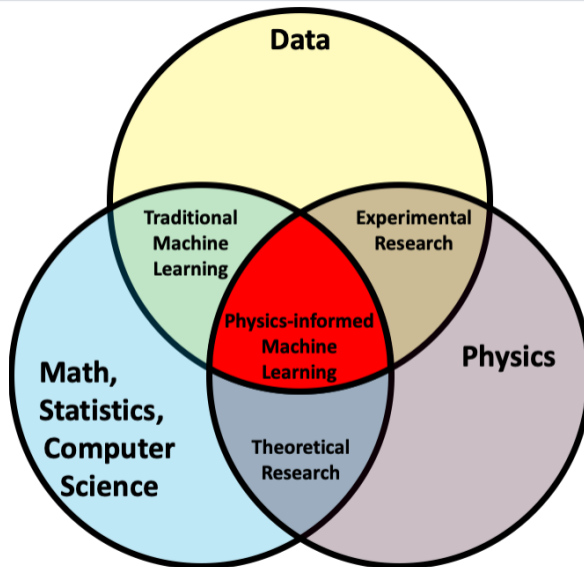




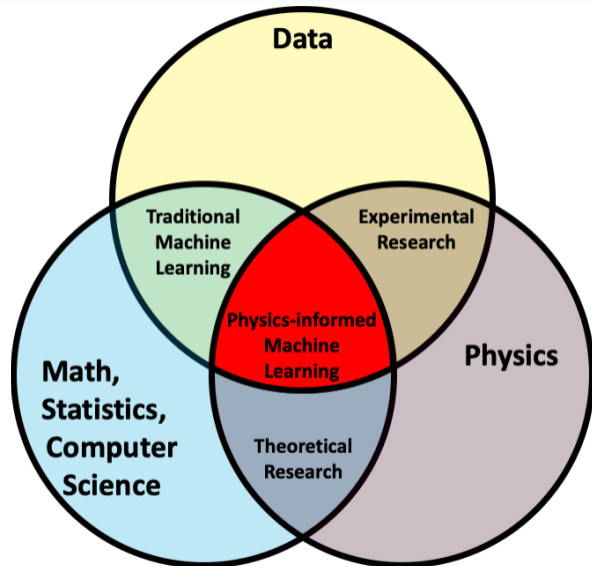


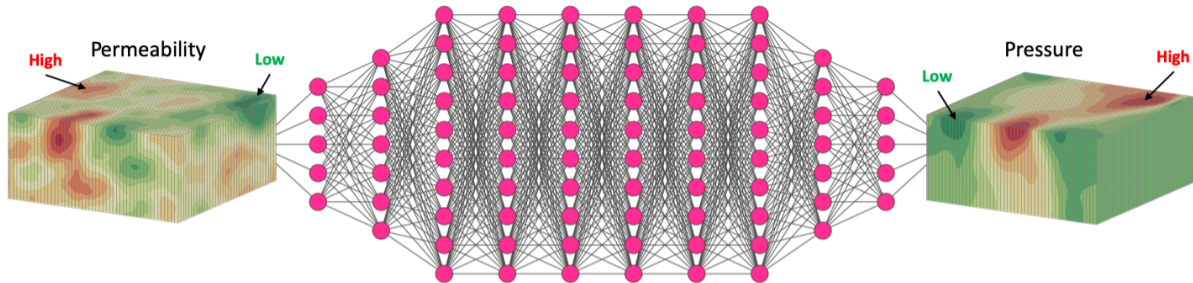




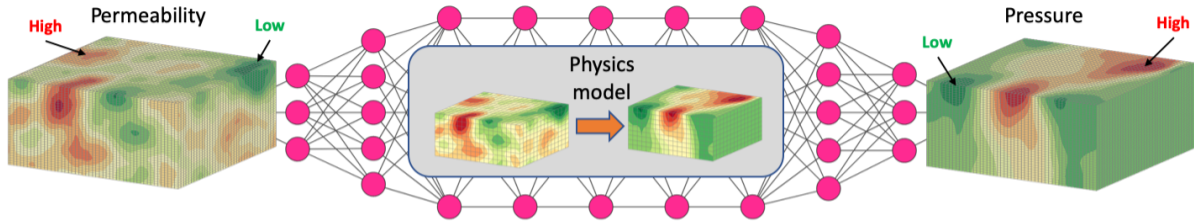


- ▶ **Empirical**: observations and experiments (since the cradle of our civilization)
- ▶ **Theoretical**: generalizations and models (since 1600's)
- ▶ **Computational**: analytical and numerical simulations (since 1950's)
- ▶ **Data-exploration**: unify data, simulations, and theory (since 2000's)

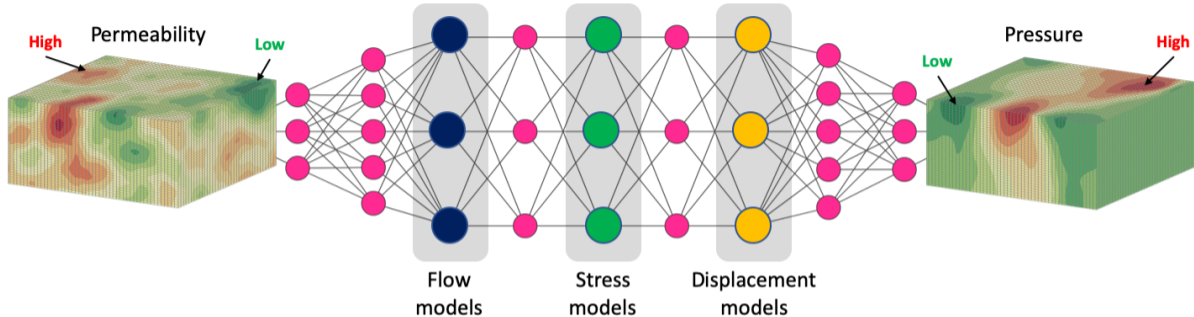





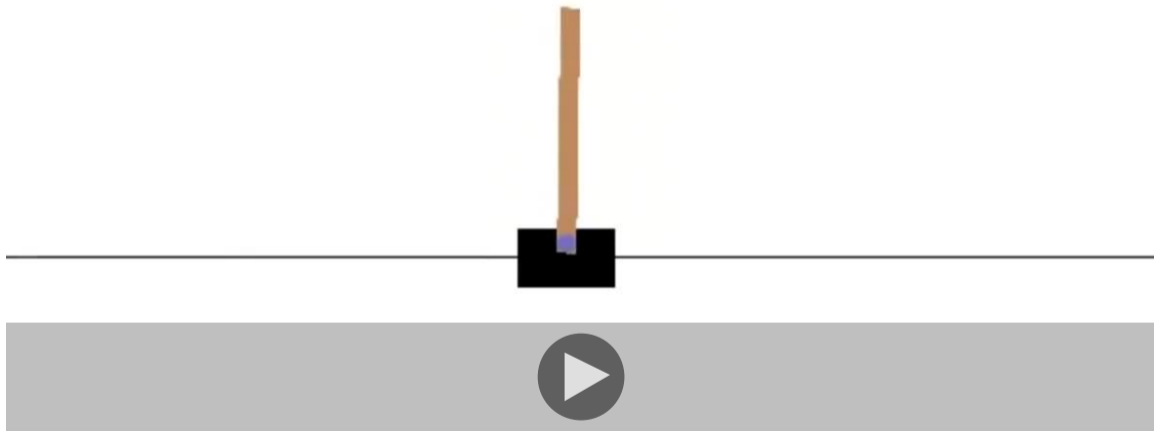
- ▶ it is a **black box**, **ad hoc** approach
- ▶ no preconceived knowledge about analyzed problem (general)
- ▶ all the neurons are $relu(Ax + b)$; A and b have no physical meaning; $relu()$ does not impose physics constraints
- ▶ neural networks needs to be very **deep** and **wide** to represent complex physics

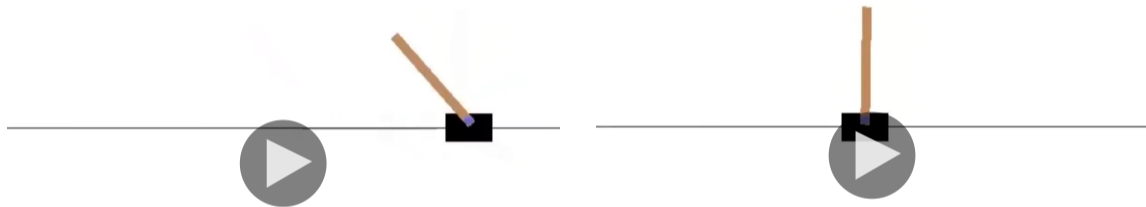


- ▶ include preconceived knowledge about analyzed problem (problem specific)
- ▶ neurons can represent $PhysicsModel(Ax + b)$; A and b have physical interpretation; $PhysicsModel()$ imposes physics constraints (e.g. conservation of mass/species)
- ▶ **PIML** models can be **trained (optimized) faster** and with **less training data**



- ▶ physics-informed layers (“fat” neurons) capture important governing processes (e.g., flow, stress, deformation, and displacement)
- ▶ can be done efficiently only through differentiable programming in 

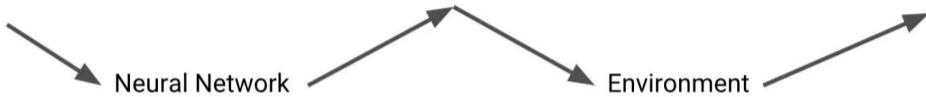




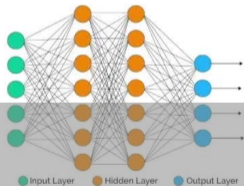
CartPole State

Control Parameters

Loss



$angle = -3^\circ$
 $velocity = 0.5^\circ/s$



{left, right}

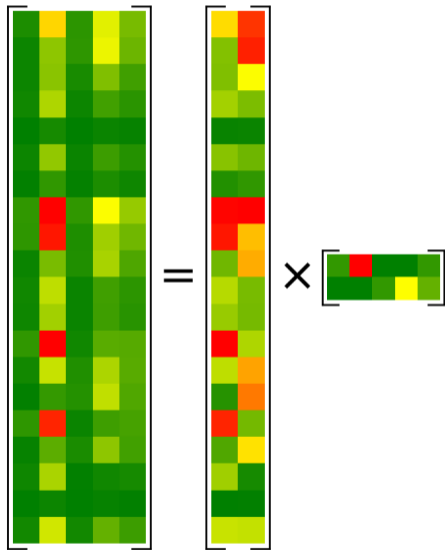


$angle^2$



X
[20 × 5]

X – data matrix
[attributes × observations]



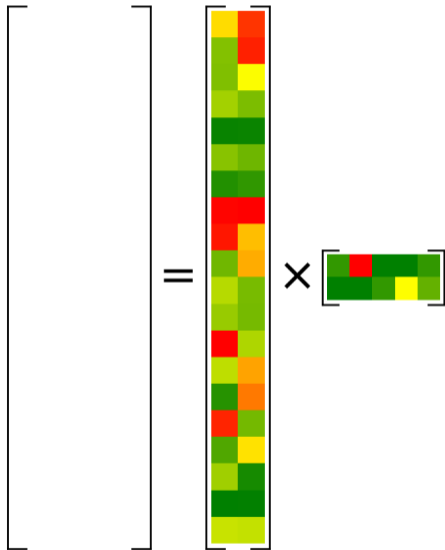
$$X = W \times H$$

$$[20 \times 5] = [20 \times 2] \times [2 \times 5]$$

X – **data** matrix
 [attributes × observations]

W – **feature (signal)** matrix
 [attributes × features]

H – **mixing** matrix
 [features × observations]



$$X = W \times H$$

$$[20 \times 5] = [20 \times 2] \times [2 \times 5]$$

X – **data** matrix

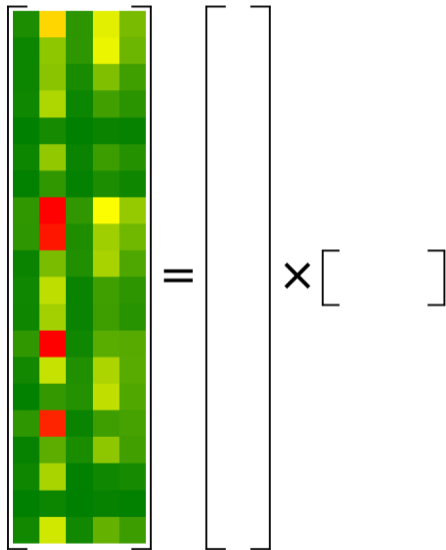
[**attributes** × **observations**]

W – **feature (signal)** matrix

[**attributes** × **features**]

H – **mixing** matrix

[**features** × **observations**]



$$X = W \times H$$

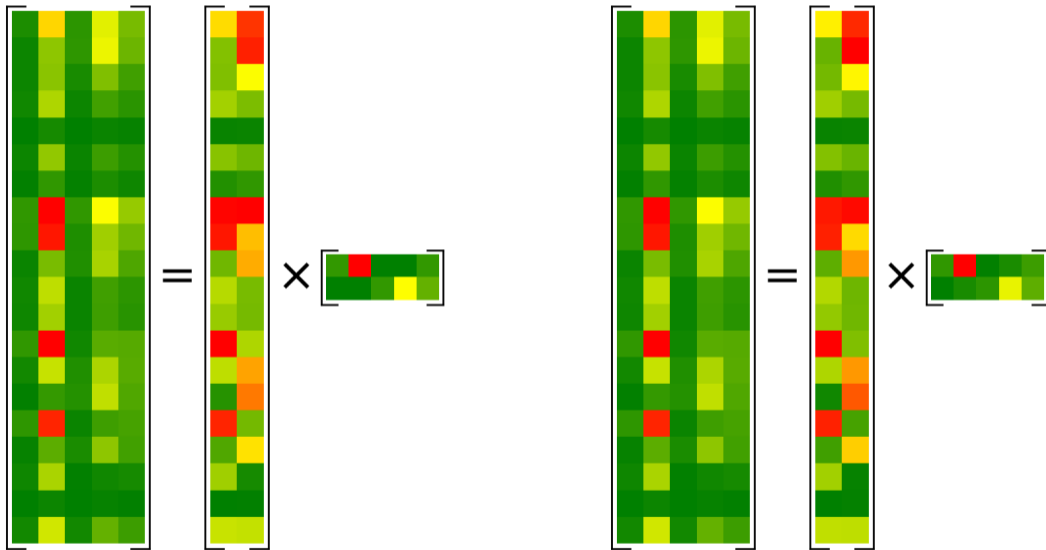
$$[20 \times 5] = [20 \times ?] \times [? \times 5]$$

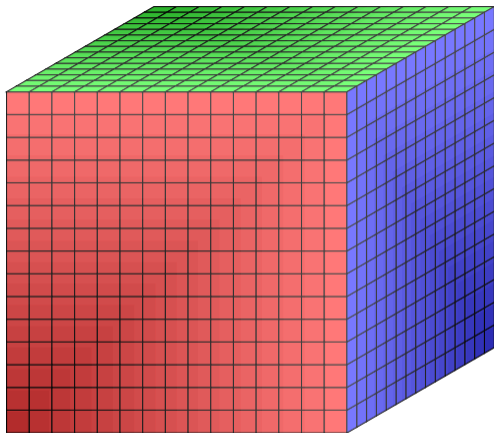
⇒ 100 **knowns**

⇒ **unknown** number of features
(2 or more)

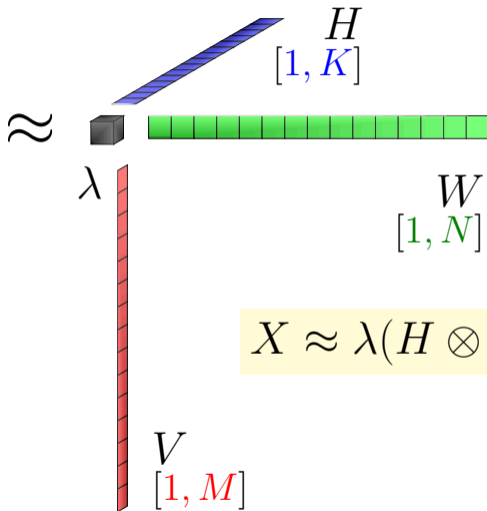
⇒ **unknown** matrix elements of W and H
(50 or more)

NMF_k: true vs. estimated matrix factorization

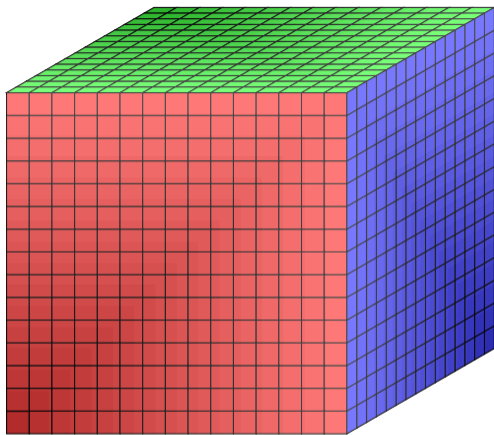




X
 $[K, M, N]$

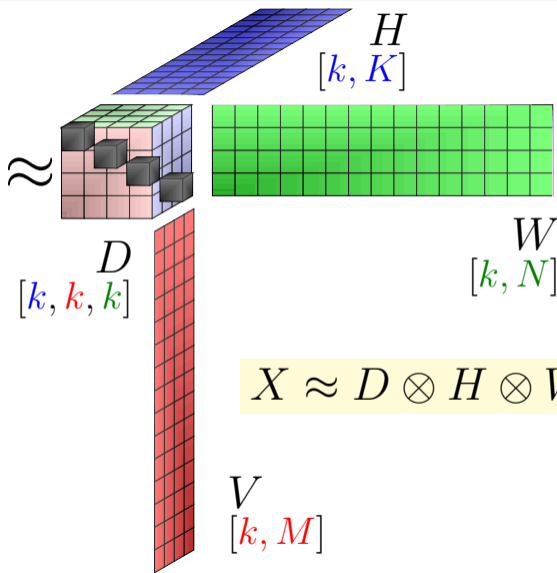


$$X \approx \lambda(H \otimes W \otimes V)$$



$$X$$

$$[K, M, N]$$



$$D$$

$$[k, k, k]$$

$$H$$

$$[k, K]$$

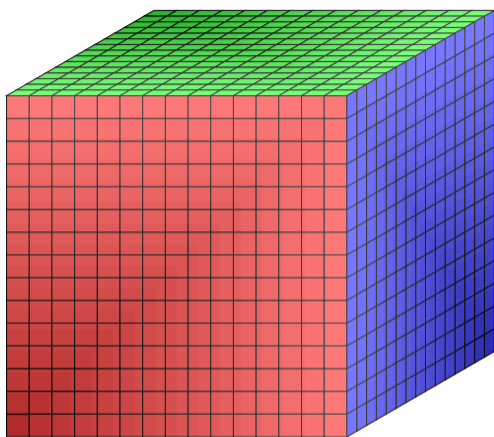
$$W$$

$$[k, N]$$

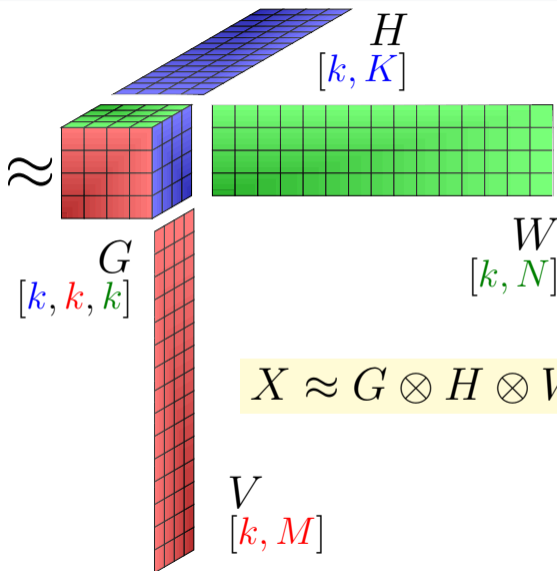
$$V$$

$$[k, M]$$

$$X \approx D \otimes H \otimes W \otimes V$$

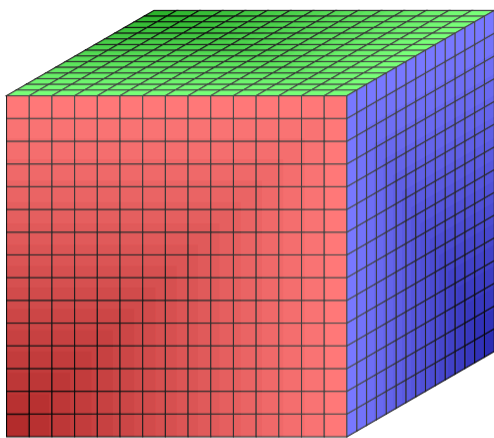


X
[K, M, N]



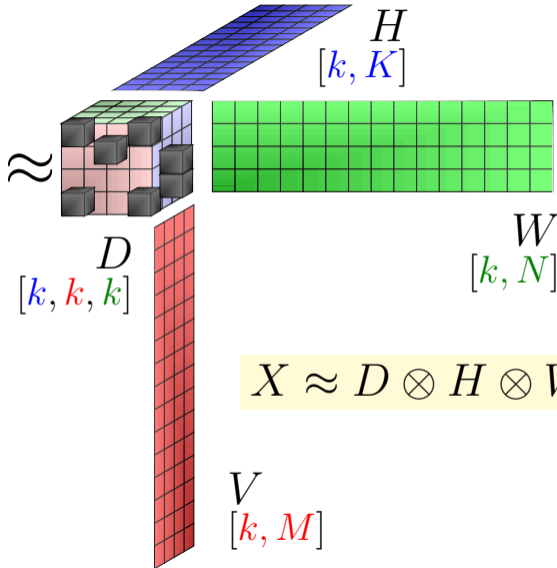
$$X \approx G \otimes H \otimes W \otimes V$$

Tucker Tensor Decomposition (3D case): Rank-7 Multirank-(3,3,4)



$$X$$

$$[K, M, N]$$



$$X \approx D \otimes H \otimes W \otimes V$$

$$V$$

$$[k, M]$$

▶ **Field Data:**

- ▶ Contamination
- ▶ Climate
- ▶ Geothermal
- ▶ Seismic
- ▶ Oil/gas production

▶ **Lab Data:**

- ▶ X-ray Spectroscopy
- ▶ UV Fluorescence Spectroscopy
- ▶ Microbial population analyses
- ▶ Isotope fractionation

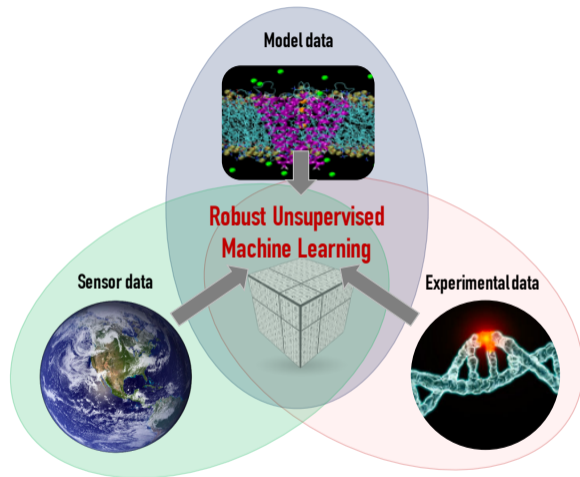
▶ **Operational Data:**

- ▶ LANSCE: Los Alamos Neutron Accelerator
- ▶ Oil/gas production

▶ **Model Outputs:**

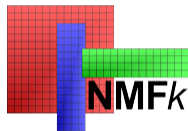
- ▶ Reactive mixing $A + B \rightarrow C$
- ▶ Phase separation of co-polymers
- ▶ Molecular Dynamics of proteins
- ▶ Climate modeling

- ▶ Developed **novel** unsupervised and physics-informed ML methods and computational tools
- ▶ Some of our tools have been recently patented
- ▶ Our ML methods have been used to solve various real-world problems (brought breakthrough discoveries related to human cancer research)



► Codes:

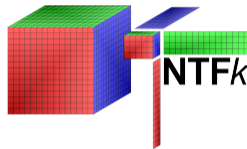
NMF_k



MADS



NTF_k



► Examples:

http://madsjulia.github.io/Mads.jl/Examples/blind_source_separation

<http://tensors.lanl.gov>

<http://tensordecompositions.github.io>

<https://github.com/TensorDecompositions>

<https://hub.docker.com/u/montyvesselinov>



- ▶ Vesselinov, Munuduru, Karra, O'Malley, Alexandrov, Unsupervised Machine Learning Based on Non-Negative Tensor Factorization for Analyzing Reactive-Mixing, **Journal of Computational Physics**, Special issue: Machine Learning, 2019.
- ▶ Stanev, Vesselinov, Kusne, Antoszewski, Takeuchi, Alexandrov, Unsupervised Phase Mapping of X-ray Diffraction Data by Nonnegative Matrix Factorization Integrated with Custom Clustering, **Nature Computational Materials**, 2018.
- ▶ Vesselinov, O'Malley, Alexandrov, Nonnegative Tensor Factorization for Contaminant Source Identification, **Journal of Contaminant Hydrology**, 2018.
- ▶ O'Malley, Vesselinov, Alexandrov, Alexandrov, Nonnegative/binary matrix factorization with a D-Wave quantum annealer, **PLOS ONE**, 2018.
- ▶ Vesselinov, O'Malley, Alexandrov, Contaminant source identification using semi-supervised machine learning, **Journal of Contaminant Hydrology**, 2017.
- ▶ Alexandrov, Vesselinov, Blind source separation for groundwater level analysis based on nonnegative matrix factorization, **WRR**, 2014.