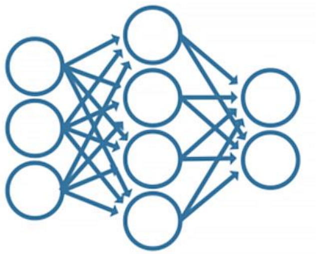
物理信息神经网络（PINN）在MATLAB中的实现

讲座日程

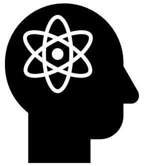
- 什么是物理信息机器学习 （PIML）？
- PIML 是做什么用的?
- 用户案例：使用 PINN 求解反问题

什么是物理信息机器学习（PIML）？



ML/DL

+



Physical understanding

=

Better methods to solve science & engineering problems

“A set of methods and tools that systematically ***integrate*** *recent advancements in* ***machine learning*** *algorithms* ***with mathematical models*** *developed in various scientific and engineering domains.”*

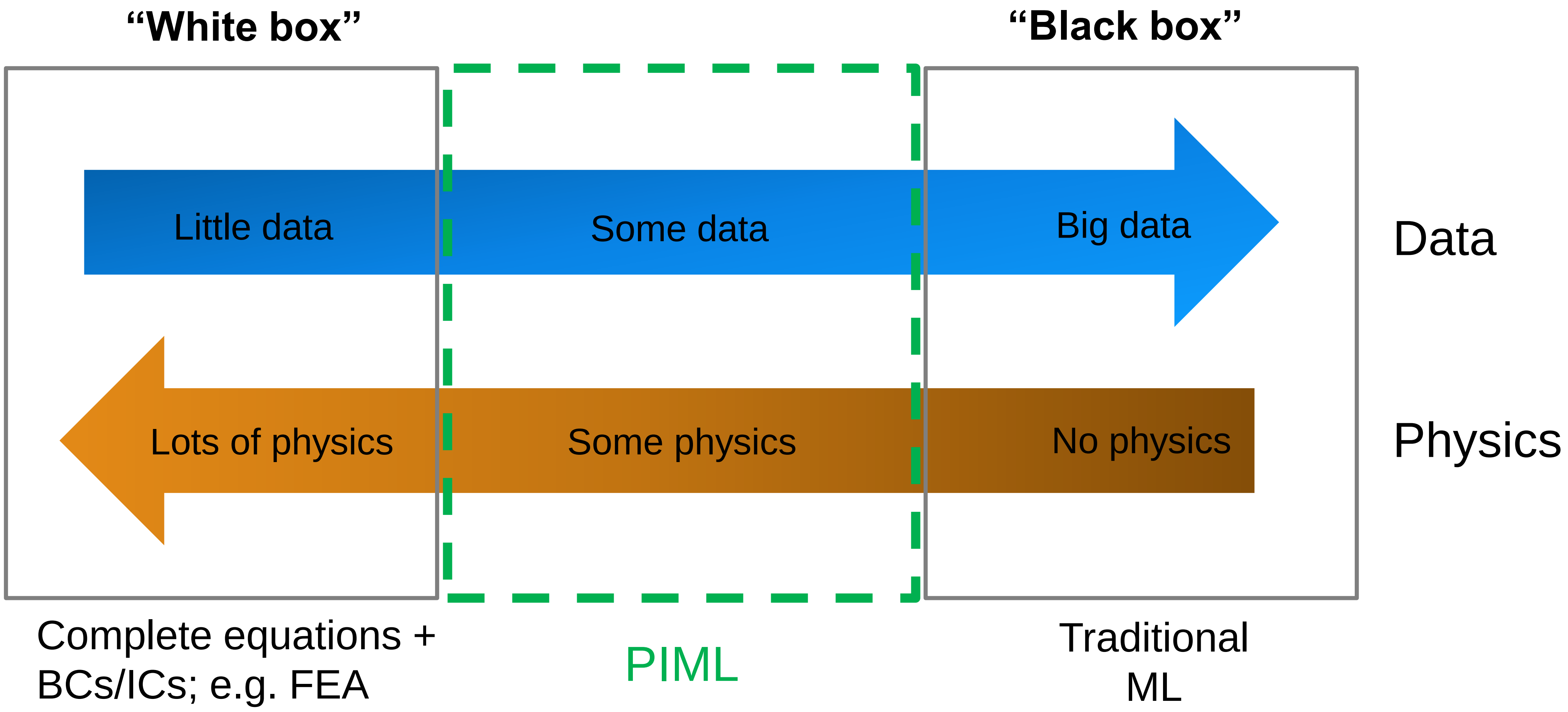
Ngheim, T.X., Drgona, J., Jones, C., Nagy, Z., Schwan, R., Dey, B., Chakrabarty, A., Di Cairano, S., Paulson, J.A., Carron, A., Zeilinger, M., Cortez, W.S., Vrabie, D.L., "Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems", American Control Conference (ACC), DOI: 10.23919/ACC55779.2023.10155901, May 2023.

“The process by which prior knowledge stemming from our ***observational, empirical, physical, or mathematical understanding*** *of the world can be* ***leveraged to improve*** *the performance of a* ***learning algorithm****.”*

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3, 422 - 440.

为什么选择 PIML？

PIML 擅长于具有一些数据和一些物理理解的情况

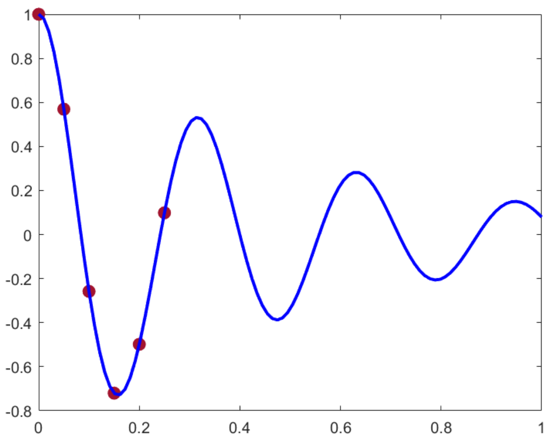


Challenges
Physical simulations can be expensive
Physics might only be partially known
(e.g. unknown coefficients, missing terms)

Challenges
Data acquisition
Interpretability
Physical inconsistencies

PIML 的工作原理

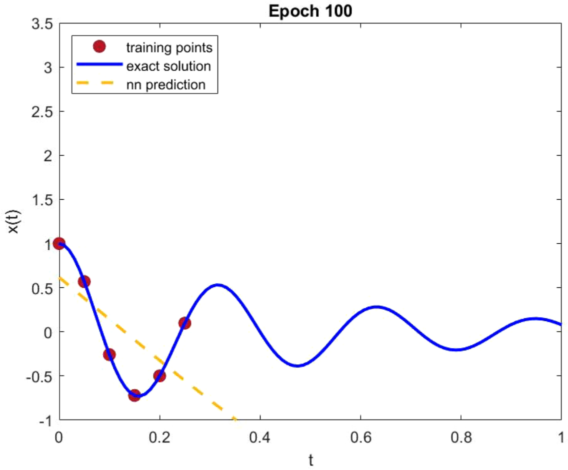
Purely data driven method



$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

How does a PIML work?

Purely data driven method

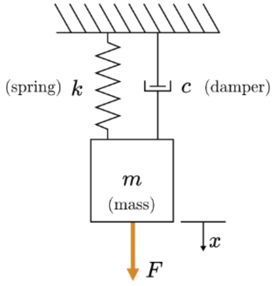


$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

Physics-Informed Neural Network

$$m\ddot{x} + c\dot{x} + kx = F$$

+

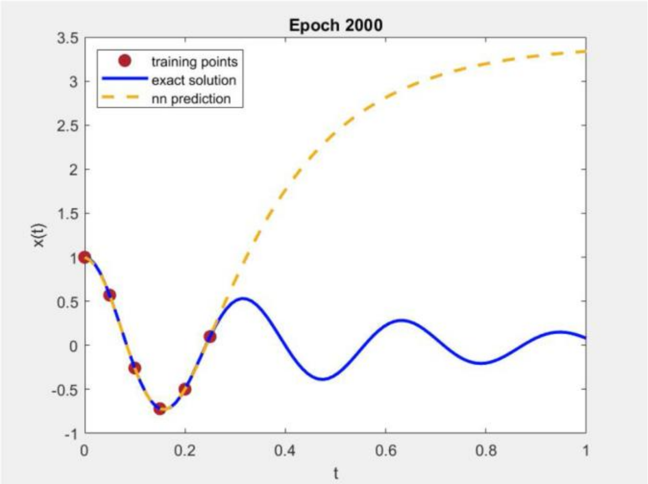


$$m\ddot{x} + c\dot{x} + kx = F$$

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

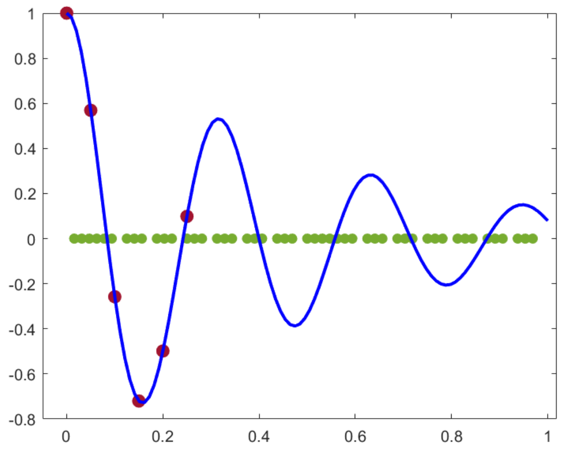
How does a PIML work?

Purely data driven method

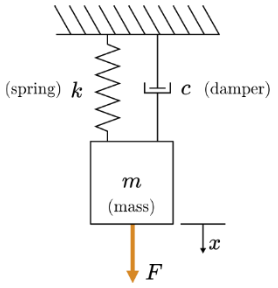


$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

Physics-Informed Neural Network



+



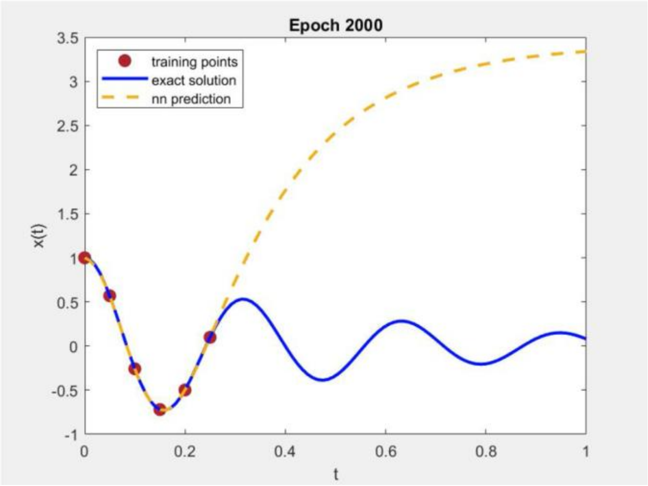
$$m\ddot{x} + c\dot{x} + kx = F$$

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

$$+\frac{1}{M}\sum_{j=1}^{M} \left| m\frac{d^2}{dt^2}x_{NN}(t_j,\theta) + c\frac{d}{dt}x_{NN}(t_j,\theta) + kx_{NN}(t_j,\theta) - F \right|^2$$

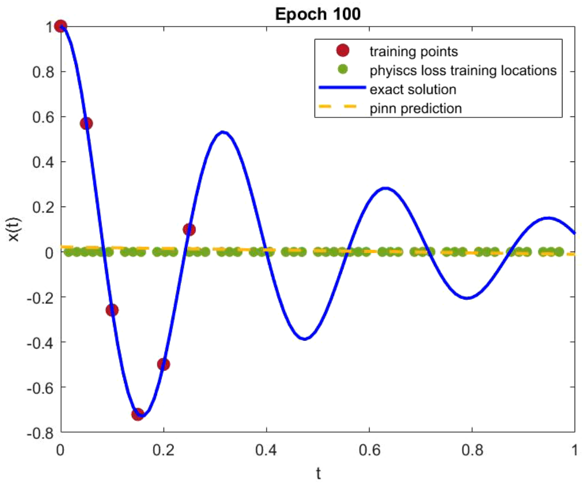
How does a PIML work?

Purely data driven method

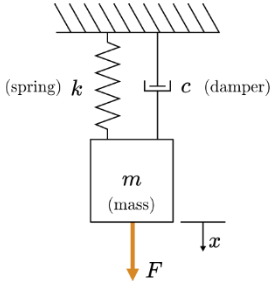


$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

Physics-Informed Neural Network



+



$$m\ddot{x} + c\dot{x} + kx = F$$

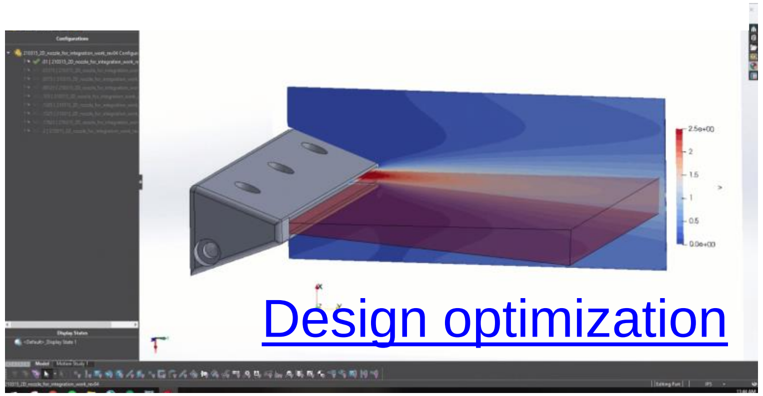
$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} |x_{NN}(t_i, \theta) - x_i^*|^2$$

$$+\frac{1}{M}\sum_{j=1}^{M} \left| m\frac{d^2}{dt^2}x_{NN}(t_j,\theta) + c\frac{d}{dt}x_{NN}(t_j,\theta) + kx_{NN}(t_j,\theta) - F \right|^2$$

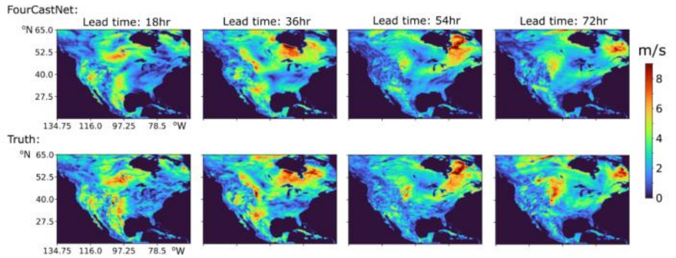
PIML 的应用

Forward, Inverse, Equation Discovery Problems

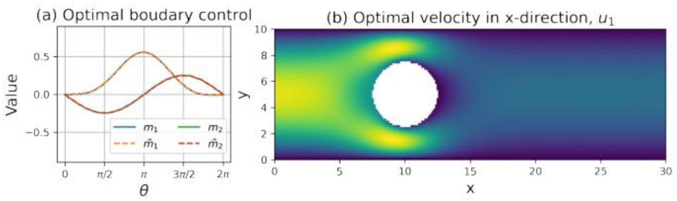
Forward Problems



[Design optimization](#)

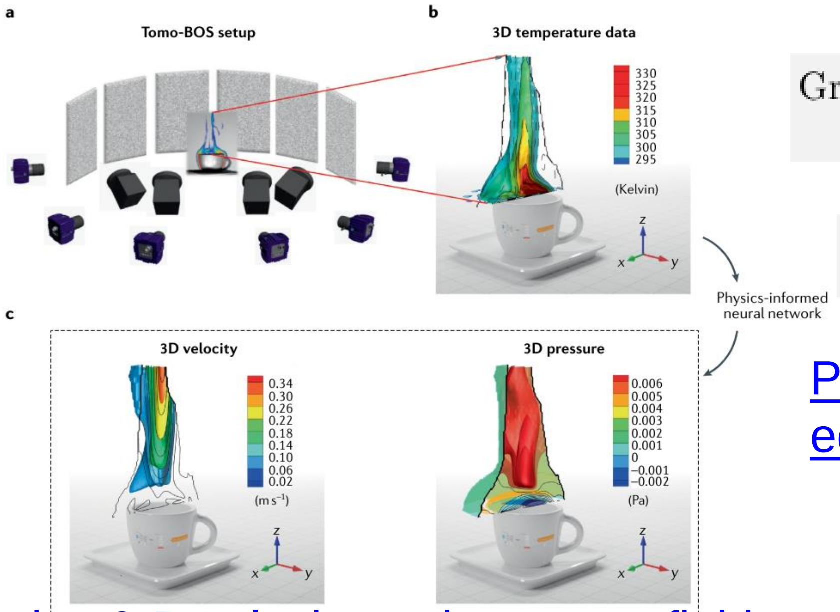


[Weather forecasting](#)

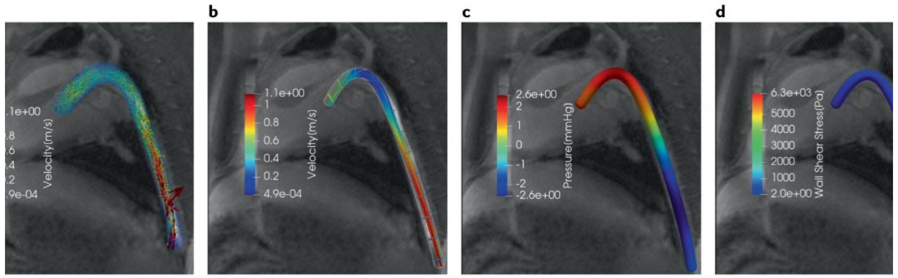


[Optimal control](#)

Inverse Problems



[Inferring 3-D velocity and pressure fields](#)



[4D-flow MRI](#)

Equation Discovery

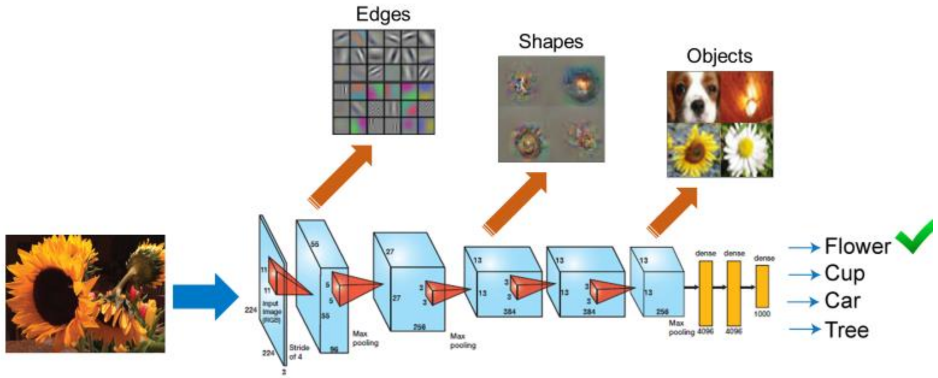
$$\text{Ground truth: } u_t = \Delta u + u - u^3 - v + 0.01$$
$$v_t = 100\Delta v + 0.25u - 0.25v$$

$$\text{Discovered: } u_t = 0.975\Delta u + 0.871u - 0.847u^3 - 0.924v + 0.010$$
$$v_t = 84.339\Delta v + 0.225u - 0.229v$$

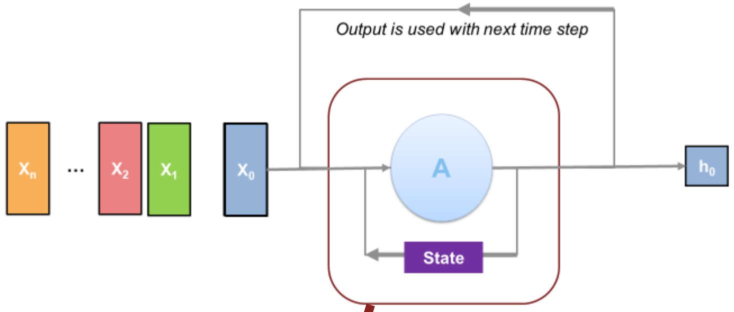
[Physics-informed learning of governing equations from scarce data](#)

R2019a 及更早版本 – 有限的神经网络类型

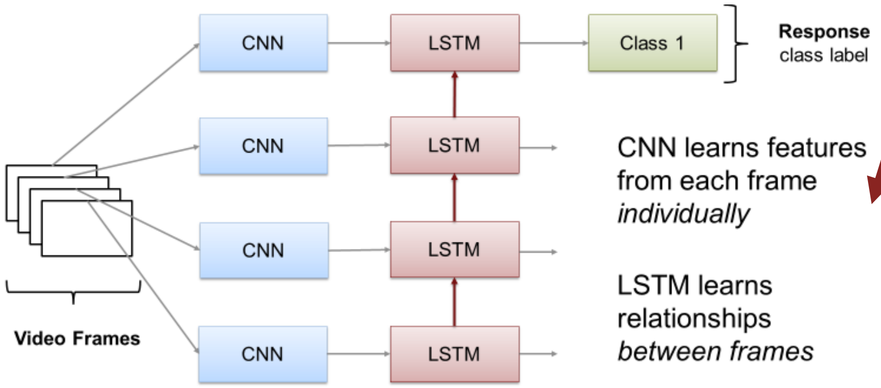
CNN (Convolutional)



LSTM (Recurrent)



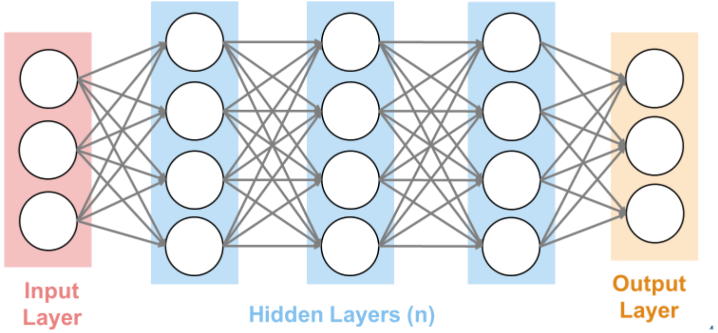
C-LSTM (Combined)



CNN learns features from each frame *individually*

LSTM learns relationships *between frames*

MLP (Multi-layer Perceptron)

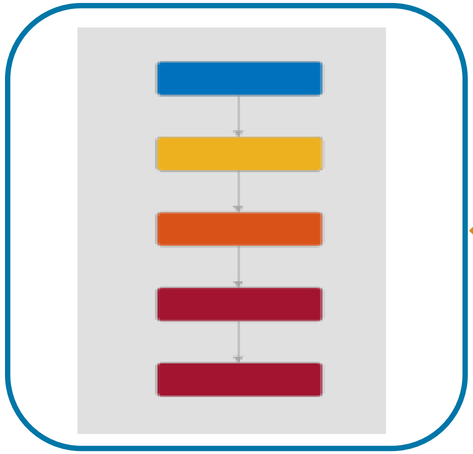


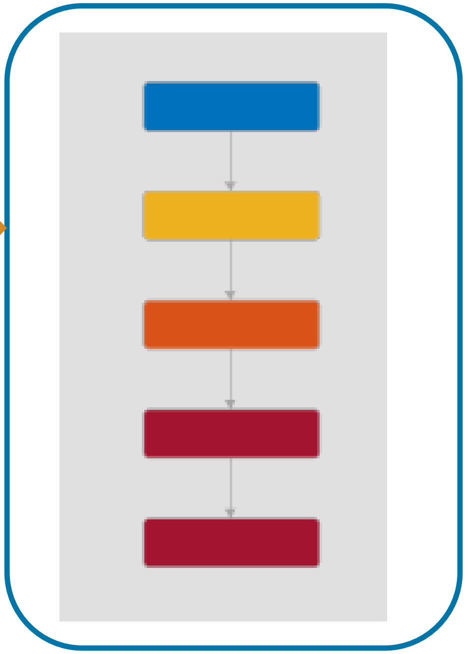
10

R2019b+ 中扩展的深度学习框架

R2019a | **R2019b R2020a**

Simpler Framework | Extended Framework





Series/DAG/Recurrent Networks

Advanced or Custom Neural Networks

MATLAB 和 Simulink 中 AI 的演进

Up to 2019

工具箱
- 统计和机器学习工具箱
- 深度学习工具箱
- 文本分析工具箱
- 强化学习工具箱
- 预测性维护工具箱

Apps
- 分类学习器
- 回归学习器
- Diagnostic Feature Designer
- Image Labeler
- 深度网络设计器
- Video Labeler
- 信号标注器

AutoML
- 特征选择

代码生成
- GPU Coder
- MATLAB Coder

互操作性
- TensorFlow-Keras 导入
- ONNX 支持

2020 - 2022

模型易得性
- Deep Learning Model Hub

Apps / 实时任务
- 试验管理器, Lidar Labeler
- Reinforcement Learning Designer
- 聚类和降维实时任务

AutoML
- 自动模型选择/调整
- 自动化特征工程

压缩和代码生成
- Quantization, Taylor Pruning
- Deep Learning HDL Coder
- TensorFlow Lite

可解释性
- LIME/Shapley

互操作性
- TensorFlow 模型导入、导出
- PyTorch 导入

机器学习
- 增量学习，增强集成
- 异常检测、漂移检测

基于模型的设计
- 图像分类和模型预测
- 循环神经网络
- 目标检测器和预测模块

2023 - 2024

压缩
- Taylor, Projection Pruning

深度学习
- Transformers
- L-BFGS 求解器
- 灵活的端到端神经网络工作流程

可解释性
- 机器学习的公平性
- 使用D-RISE进行可视化

可生成 AI
- AI chat playground
- OpenAI API 连接器

互操作性
- Simulink 中的联合执行模块

机器学习
- 回归直接预测
- 增量异常检测
- 系统辨识模型中的机器学习

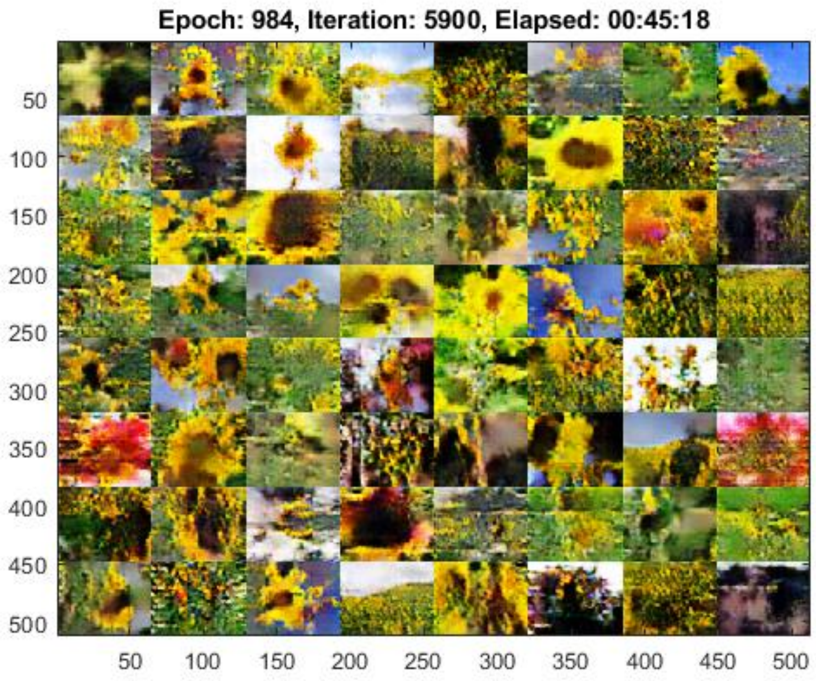
验证
- 分布外检测 （OOD） 和代码生成支持
- 鲁棒性

开发您自己的自定义深度神经网络

创建您自己的自定义网络或重新创建最近在研究中发布的网络

自定义训练循环，自定义损失函数	R2019b
自动微分，共享权重	
3D 层、MIMO	R2020a
代码生成支持	R2020b
高阶导数，一维卷积	R2021a
短时傅里叶变换层 *Signal Processing Toolbox*	R2021b

GANs, Siamese Networks	R2019b
CGANs	R2020a
图卷积网络	R2021a
贝叶斯神经网络	R2023a
Transformers, Neural ODE 层	R2023b



扩展深度学习框架增加了以下内容：

- **Unsupervised deep learning** (generative modeling)
- **Multiple input, multiple output** (MIMO) neural networks
- **Arbitrary functional programming** in neural networks
- **Complete customization when training** neural networks

Enabling Features

- Shared weights
- Automatic differentiation
- Flexible training structures

14

一个典型的深度学习训练循环包括？

1. 将整个数据集（纪元）拆分为更小的块（小块）。
2. 在每个小批量中，如有必要进行预处理。
3. 通过网络向前传递输入数据（推理）。
4. 将结果与目标进行比较（即计算损失函数）。
5. 对网络中的每一层或操作进行微分（计算梯度）与 loss 的损失。
6. 更新模型的权重和偏差（参数）。
7. 可视化流程或任何相关指标。

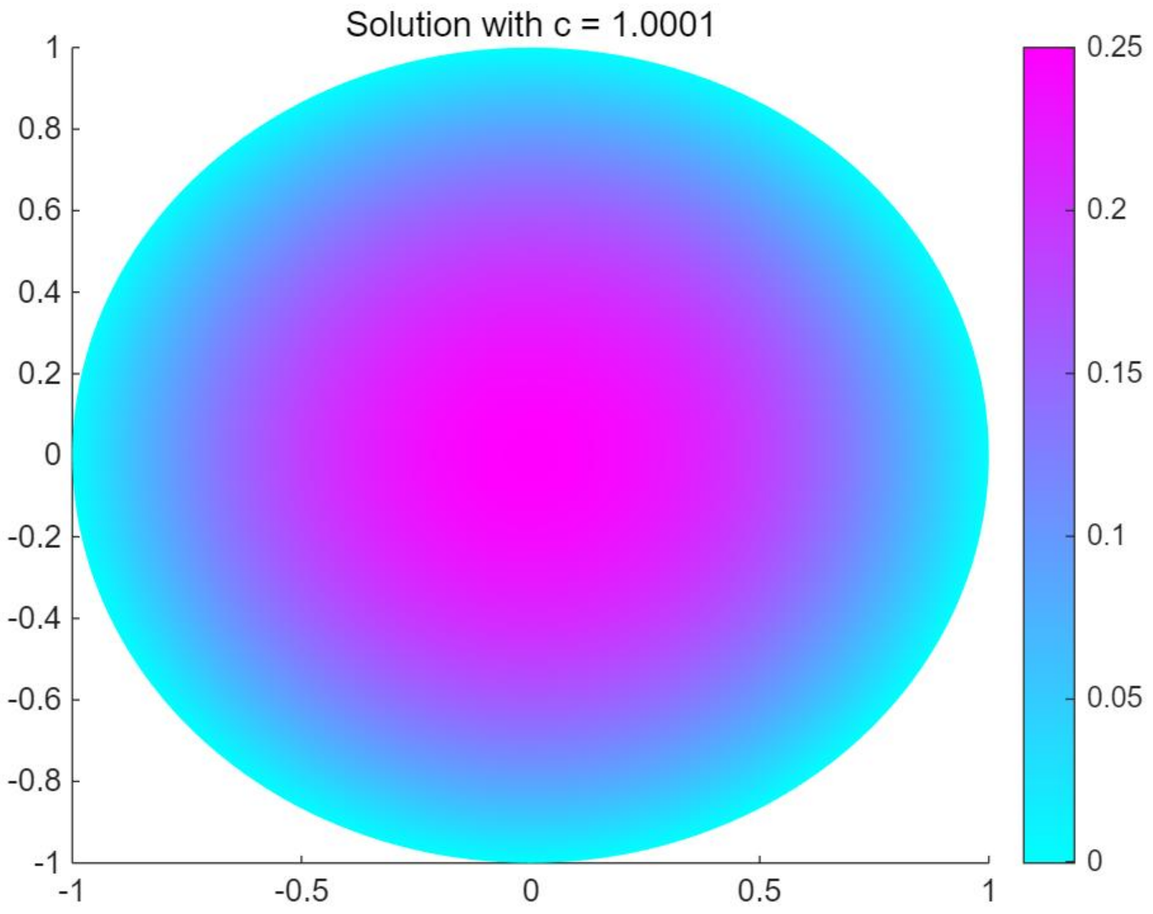
}（第 5–6 步）**Backpropagation**

扩展框架中的关键新名词

`dlarray`	Data container (inputs, intermediate calculations, outputs)
`dlnetwork`	Network container
`dlfeval`	Evaluate deep learning model or function
`dlgradient`	Compute gradients using automatic differentiation (autodiff)
`dlupdate*`	Perform one step of backpropagation using a solver

* *Equivalently,* `sgdmupdate`, `adamupdate`, *and* `rmspropupdate`

示例：使用 PINN 求解反问题



示例链接：https://github.com/matlab-deep-learning/Inverse-Problems-using-Physics-Informed-Neural-Networks-PINNs

了解更多

- [What Are Physics-Informed Neural Networks (PINNs)?]
- User story: Using Physics-Informed Machine Learning to Improve Predictive Model Accuracy
- 【深度学习实践】：使用神经网络求解偏微分方程 https://www.bilibili.com/video/BV1VT411E7sY/?share_source=copy_web&vd_source=65c2fd4545ab3146e610a2a44419b1b4

南京大学 PINN 应用

Questions?