MATLAB EXPO 2019

Deep Learning e Reinforcement Learning per l’intelligenza artificiale

Giuseppe Ridinò
Why MATLAB for Artificial Intelligence?
Artificial Intelligence

Development of computer systems to perform tasks that normally require human intelligence
A.I. Applications

Object Classification

Speech Recognition

Predictive Maintenance

Signal Classification

Automated Driving

Stock Market Prediction
Artificial Intelligence

- Development of computer systems to perform tasks that normally require human intelligence

Machine Learning

- Decision Trees
- K-means
- Nearest Neighbor
- Logistic Regression
- SVM
- Deep Learning
- Reinforcement Learning
- Gaussian Mixture
Machine Learning and Deep Learning

Unsupervised Learning
[No Labeled Data]
Clustering

Machine Learning
Machine Learning and Deep Learning

- **Unsupervised Learning** [No Labeled Data]
  - Clustering

- **Supervised Learning** [Labeled Data]
  - Classification
  - Regression
Machine Learning and Deep Learning

Supervised learning typically involves feature extraction

Deep learning typically does not involve feature extraction
Deep Learning

- Subset of machine learning with **automatic feature extraction**
  - Learns features and tasks directly from data
  - More Data = better model
Deep Learning Uses a Neural Network Architecture

Input Layer

Hidden Layers (n)

Output Layer
Deep Learning Datatypes

- Image
- Signal
- Numeric
- Text
Deep Learning Workflow

Prepare Data
- Data access and preprocessing
- Ground truth labeling

Train Model
- Model design, Hyperparameter tuning
- Model exchange across frameworks
- Hardware-accelerated training

Deploy
- Multiplatform code generation (CPU, GPU)
- Edge deployment
- Enterprise Deployment
Why MATLAB for A.I. Tasks?

- Increased productivity with interactive tools
- Generate simulation data for complex models and systems
- Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains
Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Labeling  Training  Model Exchange

Full A.I. workflows that cannot be easily replicated by other toolchains
Labeling for deep learning is repetitive, tedious, and time-consuming...

but necessary
Signal Labeler

Define Labels

Interactively Label Signals

View properties of labels
User Story – Veoneer (Autoliv)

- Automotive
  - Software and hardware for active safety, autonomous driving, occupant protection, and brake control
- Building radar sensor – check accuracy using LiDAR-based verification
- Human analyzes hours of recorded data
- Used MATLAB to semi-automate labeling and tracking of 3D LiDAR point clouds.
Manual Labeling for 25 events took over 20 minutes. After full automation with MATLAB’s tools, it took 5 minutes.
Use Deep Network Designer to Create Networks
Transfer Learning with Pre-trained Models

- Inception-v3
- ResNet-101
- VGG-16
- Inception-ResNet-v2
- ResNet-18
- GoogLeNet
- DenseNet-201
- VGG-19
- SqueezeNet
- AlexNet
- ResNet-50

Import & Export Models Between Frameworks

- Keras-Tensorflow Importer
- Caffe Model Importer
- ONNX Model Converter
Model Exchange with MATLAB

Open Neural Network Exchange

MATLAB

ONNX

PyTorch

Caffe2

MXNet

Core ML

CNTK

Keras-
Tensorflow

Caffe
Why MATLAB for A.I. Tasks?

- Increased productivity with interactive tools
- Generate simulation data for complex models and systems
- Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains
Why MATLAB for A.I. Tasks?

Generate simulation data for complex models and systems

Full A.I. workflows that cannot be easily replicated by other toolchains
Reinforcement Learning vs Machine Learning vs Deep Learning

Supervised learning typically involves feature extraction

Deep learning typically simplifies feature extraction

Machine Learning

Unsupervised Learning [No Labeled Data]

Supervised Learning [Labeled Data]

Clustering
Classification
Regression

Deep Learning
Reinforcement Learning vs Machine Learning vs Deep Learning

Reinforcement learning:

- Learning through trial & error [interaction]
- It’s about learning a behavior or accomplishing a task
What is Reinforcement Learning?

- **What is Reinforcement Learning?**
  - Type of machine learning that trains an ‘agent’ through repeated interactions with an environment

- **How does it work?**
  - Through a trial & error process that uses a reward system to maximize success
Reinforcement Learning enables the use of Deep Learning for Controls and Decision Making Applications

Controls

Robotics

A.I. Gameplay

Autonomous driving
How Does Reinforcement Learning Work?

STATE → AGENT → ACTION → REWARD → ENVIRONMENT
A Practical Example of Reinforcement Learning

Training a Self-Driving Car

Vehicle’s computer learns how to drive… (agent)
- using sensor readings from LIDAR, cameras,… (state)
- that represent road conditions, vehicle position,… (environment)
- by generating steering, braking, throttle commands,… (action)
- to avoid collisions and lane deviation… (reward).

The goal of Reinforcement learning is for the agent to find an optimal algorithm for performing a task.
Deep Networks are commonly found in the agent, because they can model complex problems.

- Turn left
- Turn right
- Brake
- Accelerate
Reinforcement Learning Workflow

**Prepare Data**
- Data access and preprocessing
- Ground truth labeling

**Train Model**
- Reinforcement learning
  - Training agent to perform task
  - Developing reward system to optimize performance

**Deployment**
- Multiplatform code generation (CPU, GPU)
- Edge deployment
- Enterprise Deployment

Simulink – generate data for dynamic systems (planes, cars, robots, etc.)
Why MATLAB and Simulink for Reinforcement Learning?

Virtual models allow you to simulate conditions hard to emulate in the real world.
Using MATLAB and Simulink for Reinforcement Learning

- Reinforcement learning is a dynamic process
- Decision making problems
  - Financial trading, calibration, etc.
- Controls-based problems
  - Lane-keep assist, adaptive cruise control, robotics, etc.
Why MATLAB for A.I. Tasks?

- Increased productivity with interactive tools
- Generate simulation data for complex models and systems
- Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains
Why MATLAB for A.I. Tasks?

- Increased productivity with interactive tools
- Generate simulation data for complex models and systems
- Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains
Deployment and Scaling for A.I.

MATLAB

Embedded Devices

Enterprise Systems
Embedded Devices – Automatic Code Generation

MATLAB Code → Auto-generated Code (C/C++/CUDA) → Deployment Target
Deploying Deep Learning Models for Inference

Deep Learning Networks → Coder Products

Intel MKL-DNN Library

NVIDIA TensorRT & cuDNN Libraries

ARM Compute Library
With GPU Coder, MATLAB is fast

Single Image Inference (Titan V, Linux)

GPU Coder is faster than TensorFlow, MXNet and Pytorch

- TensorFlow
- MXNet
- GPU Coder
- PyTorch

Intel® Xeon® CPU 3.6 GHz - NVIDIA libraries: CUDA10 - cuDNN 7 - Frameworks: TensorFlow 1.13.0, MXNet 1.4.0 PyTorch 1.0.0
Run thousands of simulations in parallel with MATLAB Parallel Server to save hours of training time.

```matlab
>> parpool(parcluster('HPC1'),100);
>> parfor i = 1:3000,
>>     c(:,i) = eig(rand, 1000);
>> end
```
Deployment to the cloud with MATLAB Compiler and MATLAB Production Server
Musashi Seimitsu Industry Co., Ltd.
Detect Abnormalities in Automotive Parts

MATLAB use in project:
- Preprocessing of captured images
- Image annotation for training
- Deep learning based analysis
  - Various transfer learning methods (Combinations of CNN models, Classifiers)
  - Estimation of defect area using Class Activation Map (CAM)
  - Abnormality/defect classification
- Deployment to NVIDIA Jetson using GPU Coder

Automated visual inspection of 1.3 million bevel gear per month
Why MATLAB for A.I. Tasks?

- Increased productivity with interactive tools
- Generate simulation data for complex models and systems
- Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains