MATLAB EXPO 2019

Deep Learning and Reinforcement Learning Workflows in AI

Valerie Leung
Why MATLAB for Artificial Intelligence?
Artificial Intelligence

Development of computer programs to perform tasks that normally require human intelligence
AI Applications

Object Classification

Speech Recognition

Predictive Maintenance

Signal Classification

Automated Driving

Stock Market Prediction
Machine Learning and Deep Learning

Machine Learning

Unsupervised Learning
[No Labeled Data]

Supervised Learning
[Labeled Data]

Clustering
Classification
Regression
Machine Learning and Deep Learning

- **Machine Learning**
  - Unsupervised Learning [No Labeled Data]
  - Supervised Learning [Labeled Data]
    - Clustering
    - Classification
    - Regression

- **Deep Learning**

  Machine learning typically involves **feature extraction**

  Deep learning typically **does not involve feature extraction**
Deep Learning Uses a Neural Network Architecture
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 AI Tasks?

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

Full AI workflows that cannot be easily replicated by other toolchains
Why MATLAB for AI Tasks?

Increased productivity with interactive tools

Labeling  Training  Model Exchange

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

but necessary
Partially automate ground truth labeling with Apps
Partially automate ground truth labeling with Apps

App for ground truth labeling dedicated to automotive applications
Applications developed using labeled data
User Story – Veoneer (Autoliv)

- **Automotive:**
  - Software and hardware for active safety, autonomous driving, occupant protection, and brake control

- **Application:**
  - Build radar sensor
  - Check accuracy using LiDAR-based verification

- Used MATLAB to semi-automate labeling and tracking of 3D LiDAR point clouds
Manual Labeling for 25 events took over 20 minutes. After automation with MATLAB tools, it took 5 minutes.
Design deep networks interactively

Design network with Deep Network Designer

Check for errors with Network Analyzer
Transfer Learning with Pre-trained Models

AlexNet
VGG-16
VGG-19
SqueezeNet
GoogLeNet
Inception-v3
DenseNet-201
MobileNetV2
ResNet-18
ResNet-50
ResNet-101
Xception
Inception-ResNet-v2

Import & Export Models Between Frameworks

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

ONNX = Open Neural Network Exchange
Why MATLAB for AI Tasks?

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

Full AI workflows that cannot be easily replicated by other toolchains
Why MATLAB for AI Tasks?

Generate simulation data for complex models and systems

Reinforcement Learning

Full AI workflows that cannot be easily replicated by other toolchains
Reinforcement Learning vs Machine Learning vs Deep Learning

Reinforcement learning learns through trial and error, i.e. through interaction.

It’s about learning a behavior or accomplishing a task.
What is Reinforcement Learning?

Reinforcement learning is a type of machine learning that trains an agent through repeated interactions with an environment through a trial & error process that uses a reward system to maximize success.
A Practical Example of Reinforcement Learning
Training a Self-Driving Car
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 [observation]
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

Generate Data
- Scenario Design
- Simulation-based data generation

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.)
Scaling up deep learning in parallel and in the cloud

Run thousands of simulations in parallel with MATLAB Parallel Server to save hours of training time.
MATLAB and Simulink for Reinforcement Learning

- Reinforcement learning is a dynamic process

- MATLAB and Simulink virtual models allow you to simulate conditions that are difficult or dangerous to emulate in the real world

- Suitable for:
  - Control-based problems, e.g. automated driving (lane keep assist, adaptive cruise control), robotics, etc.
  - Decision-making problems, e.g. financial trading, games, etc.
Why MATLAB for AI Tasks?

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

Full AI workflows that cannot be easily replicated by other toolchains
Why MATLAB for AI Tasks?

Ease of deployment and scaling to various platforms

Code Generation  Embedded Devices  Enterprise Systems

Full AI workflows that cannot be easily replicated by other toolchains
Deployment and Scaling for AI
Deploying Deep Learning Models for Inference

- Deep Learning Networks
- MATLAB Coder
- GPU Coder
- Intel MKL-DNN Library
- ARM Compute Library
- NVIDIA TensorRT & cuDNN Libraries
Benchmark of GPU Coder

Single Image Inference (Titan V, Linux)

Images/Sec

- **ResNet-50**
- **VGG-16**
- **Inception-V3**

- **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
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 AI Tasks?

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

Full AI workflows that cannot be easily replicated by other toolchains