Share the EXPO experience
#MATLABEXPO
MATLAB with TensorFlow and PyTorch

David Willingham  Yann Debray  Sivylla Paraskevopoulou
The following presentation is inspired by situations and questions from current MATLAB and Python users
Questions about **deep learning** in situations such as…
Our data scientists use TensorFlow & PyTorch
Our engineers use MATLAB

How can our teams work together?
The latest models come out in TensorFlow & PyTorch first.

What support does MATLAB have for the latest models?
Deploying models into embedded systems is time-consuming

Can MATLAB help us deploy to embedded systems quicker?
Our presenters will roleplay a scenario to answer these questions

David
The Project Manager

Sivylla
The Engineering Team Lead

Yann
The Data Science Team Lead
The AI enabled car project has been approved!
Our best results come from following the AI system design workflow.
Here are the higher-level project requirements
Data Preparation requirements

- Add domain knowledge

- Make sure you have enough data
  - Generate synthetic data from system simulations
AI modeling requirements

- Find best model(s) for each application
- Don’t limit yourself to one tool
Simulation and Test requirements

- You must test the model in the overall system
- Requires integrating with system level simulations
Deployment requirements

- The aim is to target lowest cost, lowest power embedded devices available
One last requirement…

Find the best way both teams can work together
Which tools should we use?
Why don’t we use the tools together?
Sivylla and Yann are having a friendly conversation
Discussing their options…
Mitsui Chemicals Deploys AI and Automation Systems with TensorFlow and MATLAB

"MATLAB solved our problems on the field implementation and saved development time. That led to highly accurate development."
— Shintaro Masukawa, Mitsui Chemicals, Inc.

Mitsui Chemicals develops factory automation solutions by applying AI, mainly machine learning.

At the beginning of development, Mitsui Chemicals used Python+Keras (TensorFlow) for automated visual inspection of sheet-shaped products on the production line. However, ease of use and maintenance were issues in implementing the trained models in the field.

Mitsui Chemicals engineers chose MATLAB® to create applications with easy-to-understand user interfaces. The model was imported using the Deep Learning Toolbox™ Inspector for TensorFlow/Keras Model. The

Key Outcomes
• Reduced visual inspection time by 80%.
• Effectively used models trained in other frameworks.
• Deployed application with a user interface that anyone can use.

Products Used
• MATLAB
• Deep Learning Toolbox
• MATLAB Compiler
A short time later…
There are 3 ways MATLAB can work with TensorFlow & PyTorch

<table>
<thead>
<tr>
<th>#</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Co-execution with TensorFlow or PyTorch</td>
</tr>
<tr>
<td>2</td>
<td>Model converters for TensorFlow &amp; ONNX</td>
</tr>
<tr>
<td>3</td>
<td>MATLAB Deep Learning Model Hub</td>
</tr>
</tbody>
</table>
1. Co-execution with MATLAB, TensorFlow or PyTorch

- Requires MATLAB and TensorFlow / PyTorch
- Requires datatype conversion / reformatting
- Performance is impacted by data transfer between frameworks
- Allows testing of any model from TensorFlow / PyTorch in MATLAB
Co-execution allows:

- Calling Python from MATLAB to access any AI frameworks and networks
- Calling MATLAB from Python to reuse the domain specific processing
2. Model Import via TensorFlow & ONNX Converters

- Requires MATLAB only
- Import TensorFlow directly
- Import PyTorch via ONNX
- New features added each release
3. MATLAB Deep Learning Model Hub

- Requires MATLAB only
- Over 50 pretrained models
- Similar model collection with TensorFlow and PyTorch repositories
Deep Learning Workflows with MATLAB Networks

- Many examples with deep learning networks
- Examples with imported networks
  + new blog post on importing

TensorFlow model \(\text{import function}\) MATLAB network

- Predict
- Use in Simulink
- Transfer Learning
- Deploy

Embedded Device

file.exe

Standalone Application
## Our key challenges and their solutions

<table>
<thead>
<tr>
<th>#</th>
<th>Challenge</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Find and test the best model</td>
<td>• Co-execution</td>
</tr>
<tr>
<td>2</td>
<td>Deploying to embedded targets</td>
<td>• Check if the model exists in the MATLAB Deep Learning Model Hub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Import into MATLAB using model converters</td>
</tr>
</tbody>
</table>

![Flowchart](image_url)
Let’s get this done!
This now concludes our presentation

Links from today’s talk:

Co-execution between MATLAB and TensorFlow

Importing Models from TensorFlow, PyTorch, and ONNX

MATLAB Deep Learning Model Hub

MITSUI Chemicals User Story
Thank you