Deploying AI to Embedded and Enterprise Systems

Bernhard Suhm
Machine Learning & Statistics

Greg Coppenrath
Code Generation & Certification
Example: Developing Health Monitoring for Electrical Grids
Deployment to Embedded and Enterprise Systems

**Enterprise**
- Health Monitoring of Distribution Transformers
  - SIEMENS

**Embedded**
- Card to Classify Blood Type
  - IDNEO
Agenda

Deploying AI to production is difficult

Three specific challenges:
1. Limitations of Embedded hardware
2. Ongoing changes in environment or system behavior
3. Scale to production load in Enterprise systems
Two Approaches for integrating AI with Larger System

Embedded Systems
- CPU
- GPU
- FPGA

Enterprise Systems

MATLAB

Code Generation

Compiler
Embedded Deployment of Acoustic Scene Recognition
Embedded Deployment of Acoustic Scene Recognition

Squeezenet ~5MB
ResNet-50 ~100MB

Limited resources
Embedded Deployment of Acoustic Scene Recognition

- Squeezennet ~5MB
- ResNet-50 ~100MB

Limited resources
Quiz: Which Sounds do you hear?
Embedded Deployment of Acoustic Scene Recognition

Squeezenet ~5MB
ResNet-50 ~100MB

Limited resources
Embedded Deployment of Acoustic Scene Recognition

Reformat the data

SqueezeNet ~5MB
ResNet-50 ~100MB

Limited resources
Embedded Deployment of Acoustic Scene Recognition

Reformat the data

Convolutional Neural Networks (CNN)

SqueezeNet ~5MB
ResNet-50 ~100MB

Limited resources
How can Embedded Deployment Be Enabled?
How can Embedded Deployment Be Enabled?
How can Embedded Deployment Be Enabled?
How can Embedded Deployment Be Enabled?
How can Embedded Deployment Be Enabled?
How can Embedded Deployment Be Enabled?
How can Embedded Deployment Be Enabled?

Diagram showing:
- Original network structure
- Pruning network structure
- Layer Fusion

Icons and symbols:
- Disk
- Running figure
- Target
- Equal sign
- Tools (hammer and screwdriver)

19
How can Embedded Deployment Be Enabled?

- Original
- Pruning
- Layer Fusion
- Quantizing
Deep Learning Quantization: Acoustic Scene Classification

Calibration dataset → Calibrate → Quantize → Validate

Trained ‘Single’ DNN

Quantized ‘INT8’ DNN
Deep Learning Quantization: Acoustic Scene Classification

Use Deep Network Quantizer to Optimize the Inference Network

```matlab
load('trainedNet');
analyzeNetwork(trainedNet);
numData = size(xTrain);
umData = numData(end);
augImds = augmentedImageDatastore(trainedNet.Layers(1).InputSize, xTrain, yTrain);
calDS = augImds.subset(1:floor(numData * 0.8));
valDS = augImds.subset(floor(numData * 0.8)+1:numData);
dq = dlquantizer(trainedNet, 'ExecutionEnvironment', 'GPU');
dq.calibrate(calDS)
```

- Load trained network
- Split data: calibration – 80%, validation – 20%
- Launch Deep Network Quantizer App
Deep Learning Quantization: Acoustic Scene Classification
Deep Learning Quantization: Acoustic Scene Classification
Deep Learning Quantization: Acoustic Scene Classification

Validation Results

<table>
<thead>
<tr>
<th>Memory (MB)</th>
<th>Top-2 Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP32 4.7 MB</td>
<td>90%</td>
</tr>
<tr>
<td>INT-8 1.2 MB</td>
<td>89%</td>
</tr>
</tbody>
</table>

Learnable Parameters

- FP32
- INT-8

Top-2 Accuracy

- FP32 0.3%
- INT-8 75%
Deep Learning Quantization: Acoustic Scene Classification

Validation Results

Learnable Parameters

<table>
<thead>
<tr>
<th></th>
<th>FP32</th>
<th>INT-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>4.7</td>
<td>1.2</td>
</tr>
<tr>
<td>75%</td>
<td>1.2</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Top-2 Accuracy

<table>
<thead>
<tr>
<th></th>
<th>FP32</th>
<th>INT-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>90%</td>
<td>89%</td>
</tr>
<tr>
<td>0.3%</td>
<td>0.3%</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

- **CPU**
- **GPU**
- **FPGA**

Code Generation
AI models reflect System behaviors and Environment

(illustration only; not based on actual data)
AI models reflect System behaviors and Environment

![Graph showing system behaviors and environment](image-url)
Deployed Models Need to Adapt.
Model Updates in Embedded Deployment
Model Updates in Embedded Deployment

Production Data

MATLAB

Embedded Systems

CPU
GPU
FPGA

Update Parameters only

C code

```c
10 static emlrtSInfo emlrtS1 = { 4,
11  "classifyIonsphere",
12  "C:\\General\\johrree\\sandbox\\temp\\fes\",
13  0};
14
15 /\ Function Definitions
16 \*/
17 void classifyIonsphere(classifyIonsphere
18  const real_T X[11994], cell_wrap_0 label
19 { 
20  real_T toAlpha[90];
21  real_T exp_temp[94];
```
Agenda

Deploying AI to Embedded and Enterprise systems is difficult

Three specific challenges:
1. Limitations of Embedded hardware
2. Ongoing changes environment or system behavior
3. Scale to production load in Enterprise systems
Enterprise Deployment of AI
Integrate with Enterprise Systems and Scale to Production Load
Integrate with Enterprise Systems and Scale to Production Load
Example: Incremental Health Monitoring

Sensor data

Anomaly Detection loop

while seqn % ... there's more data to process

% Retrieve buffer of data
datafilter = (sensordata.key == thisAsset) & (sensordata.SequenceNumber <= seqn+batchsize);

streamdata = sensordata(datafilter,:);

% Detect Anomalies with incremental One-class SVM
[nextState, results] = detectAnomalyLocal(streamdata, state);

% Remember results and update state of incremental learner
anomalies(datafilter) = results.anomaly;
score(datafilter) = results.score;
timestamps(datafilter) = results.timestamp;
state = nextState;

seqn = seqn + batchsize; % step through batch test data
end
Incremental Learning within Streaming Architecture

```matlab
incMdl = incrementalLearner(mdl);

while dataStreaming
    featureChunk = extractFeatures(streamdata);
    inclMdl = updateMetricsAndFit(incMdl, featureChunk, labels);
End
```
Incremental Learning within Streaming Architecture

Sensor Data → Connector → MATLAB Production Server → Request Broker → Connector → Dashboard

K/V Store
Operationalize AI without recoding

Development

MATLAB

DESIGN

PLAN

RELEASE

BUILD

TEST

OPERATE

Operations

DEPLOY

FEEDBACK

MONITOR

IT/OT Systems

MATLAB, Excel, Java, C/C++, .NET, dll, Python, {RESTful API}

Deep Network Quantizer
Operationalize AI without recoding - Model DevOps

Development

DESIGN
PLAN
BUILD
TEST

Operations

DEPLOY
RELEASE
FEEDBACK
MONITOR

MATLAB Production
Server

Request
Broker
Worker

Streaming
Data

kafka
RabbitMQ

Dashboards
Tableau
Power BI
Qlik

TIBCO Software
Conclusions

Deploy to Embedded and Enterprise systems from one codebase

Tools for handling deployment-specific challenges:
- Fit models to embedded hardware with Quantization / Fixed-Point conversion
- Scale to data and users with MATLAB Production Server
- Incrementally adapt deployed models to maintain performance

Design, Deploy and Maintain AI-powered systems in one framework
Learn More

Check out our handout with links to customer stories, documentation – and examples which you can try out in MATLAB Online

DevOps for Software and Systems: Operationalization of Algorithms and Models

Deploying AI on PLCs
Thank you