임베디드 시스템 적용을 위한 AI 개발

신형재 부장, 매스웍스코리아
Edge AI innovates many industries!
Hardware Constraints

Memory Footprint / Compute

Memory: 32GB @ 10TFLOPs

Memory: 8GB @ GFLOPs

Memory: 1kB~1MB @ MFLOPs

* FLOPS: floating-point operations per second
What is “Edge” (Embedded) AI?

The chip has only 500 KB memory – make that smaller.

How can I make my AI smaller?

Data Scientist

Embedded Software Engineer
Why is Edge AI (Model Compression) difficult?

- AI is often big
- Knowledge Gap
Model Compression Workflow

1. Determine Hardware Constraints
2. Select Model
3. Simplify Model
4. Quantize Model Parameters
5. Deploy & Integrate
Compressing Machine Learning
Step 1: Size aware model selection

Accuracy on Complex tasks

Size /
Execution Time

Deep Neural Net

Gaussian Process
Kernel SVM
Ensembles
Shallow Nets
Linear Model
Decision Tree
Model Compression Workflow for Machine Learning

1. Classification / Regression Learner
   - Determine Hardware Constraints
   - Select (Initial) Model

2. In-App Feature Selection
   - Select Features
   - Tune Hyper-parameters

3. Bayesopt
   - Quantize Model Parameters
   - Deploy & Integrate

Fixed Point Designer / Native Simulink Block

Small
Large
Demo: Embedding AI in an intelligent Hearing Aid

- Directional
- All Around

0.5 to 256 kB on-chip memory
Demo: Fit Machine Learning for Intelligent Hearing Aid

Fitting Machine Learning onto Memory-limited Hardware

In the context of building an intelligent hearing aid, this script demonstrates the various methods available to fit machine learning onto memory-limited hardware.

Chips on hearing aids range between a few hundred down to below one kB. We'll take 50 kB as target for our example.

Load Data

As starting point, we train an initial machine learning model to classify acoustic scenes, using a subset of the data used in the original example [https://www.mathworks.com/help/audio/ug/acoustic-scene-recognition-using-late-fusion.html](https://www.mathworks.com/help/audio/ug/acoustic-scene-recognition-using-late-fusion.html)

We are just using the first 100 examples from the training set, with 15 scenes resulting in 1500 data points.

```matlab
% load the subset of acoustic scene data we're using here
load("AcousticScenes-SmallTrain.mat");

c = cpartition(trainLabels,'HoldOut',0.2);
trainSmall = xTrain(c.training,:);
testSmall = xTrain(c.test,:);

% load SelectedFeatures.mat
```
Machine Learning Demo  Size Reduction by factor 20

Target Hardware
Size: 50 kB
Compressing Deep Learning
Step 1  Size aware model selection
Step 2: Smart pruning

Remove *unimportant* parts of the network

**Trained Network**

**Pruning process**
- Evaluate importance of weights
- Remove the least important weights
- Fine Tuning (training)

**Retrain**

**Pruned + Retrained Network**
Step 3 Quantize your model
Deep Learning Demo: Scene classification

Classify 10 classes
More difficult problem → more complex model
Functionality for Compressing Deep Neural Nets

1. Deep Network Designer
2. Taylor Pruning
3. Deep Network Quantizer

Determine Hardware Constraints
Select Model
Prune Deep Neural Network
Quantize Model Parameters
Deploy & Integrate
Step 1: Select Model

Load original trained CNN model and dataset

```matlab
load('trained10classNetwork');
load('data');
```

Note: Sounds have been converted to spectrograms
Deep Learning Demo  Size Reduction by factor 5

- Initial: 4.48 MB
- Pruned: 3.57 MB
- Pruned & Quantized: 0.89 MB
One Codebase – Many Embedded Deployment targets

Zynq is a registered trademark of Xilinx Corporation
Intel logo is a registered trademark of Intel Corporation
One Codebase – Many Embedded Deployment targets
Conclusions

You can fit AI for many applications onto limited hardware

MathWorks tools make fitting AI models on constrained hardware a lot easier

Same high-level Workflow for any type of AI
To get your started:

- Learn about Embedded Deployment
- Quantization of classification SVM (Doc)
- Deploy Hand-Gesture Classifier onto Arduino (Doc)
- Generate C/C++ Code from Simulink (Video)
- Quantizing a Deep Neural Network (Video)
Thank you