MATLAB EXPO

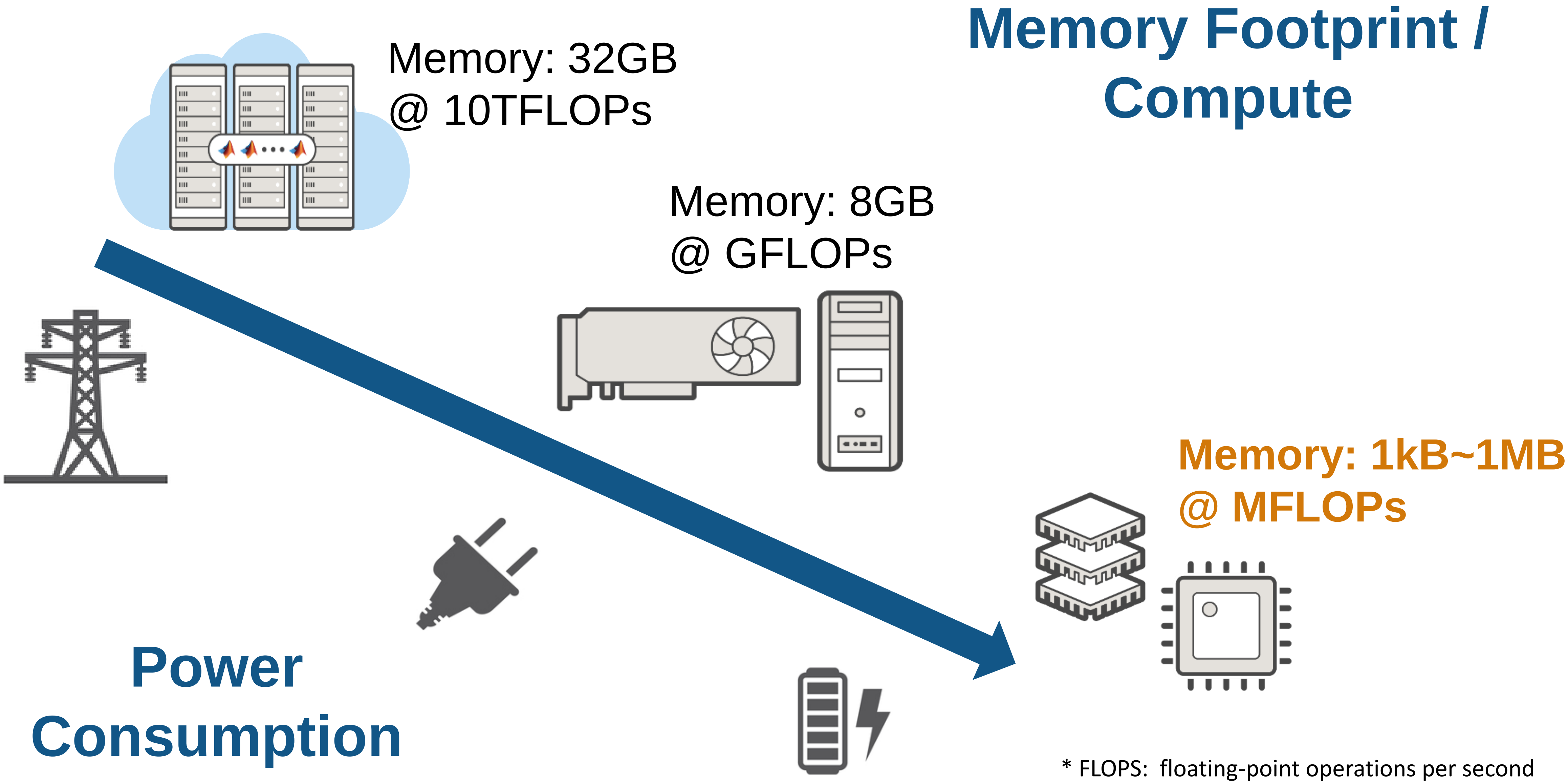
임베디드 시스템 적용을 위한 AI 개발

신행재 부장, 매스웍스코리아

MathWorks®

Edge AI innovates many industries!

Hardware Constraints



Memory Footprint / Compute

Memory: 32GB
@ 10TFLOPs

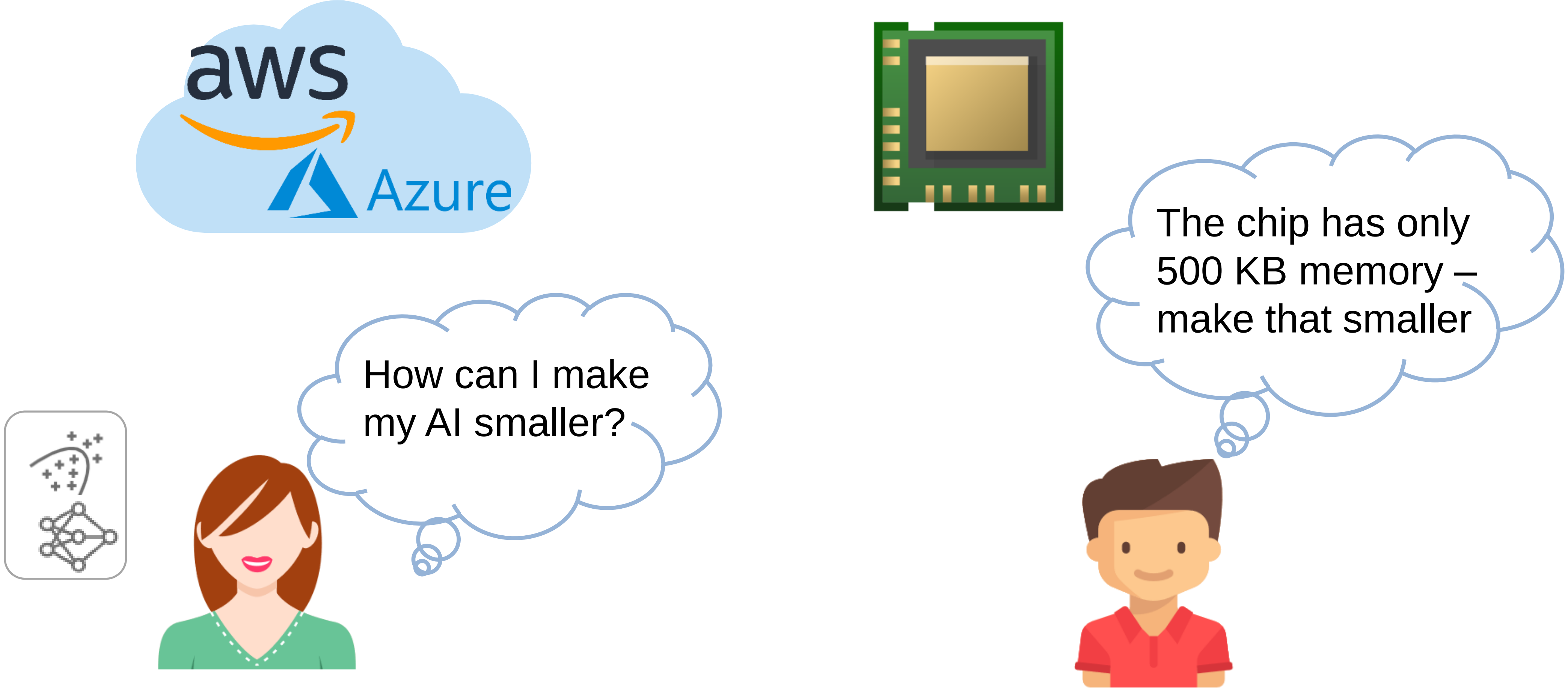
Memory: 8GB
@ GFLOPs

Memory: 1kB~1MB
@ MFLOPs

Power Consumption

* FLOPS: floating-point operations per second

What is “Edge” (Embedded) AI?



How can I make my AI smaller?

The chip has only 500 KB memory – make that smaller

Data Scientist

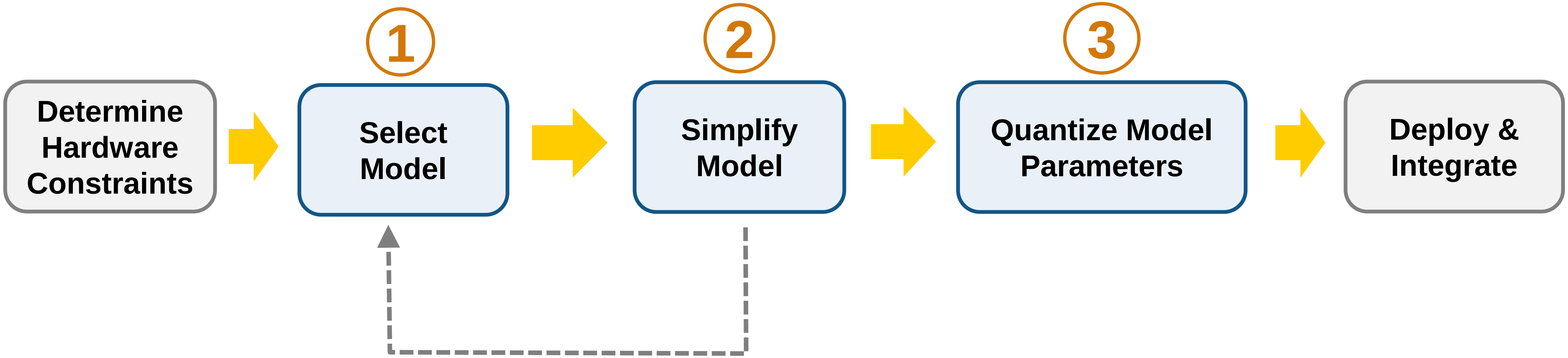
Embedded Software Engineer

Why is Edge AI (Model Compression) difficult?

AI is often big

Knowledge Gap

Model Compression Workflow



Determine Hardware Constraints

1 Select Model

2 Simplify Model

3 Quantize Model Parameters

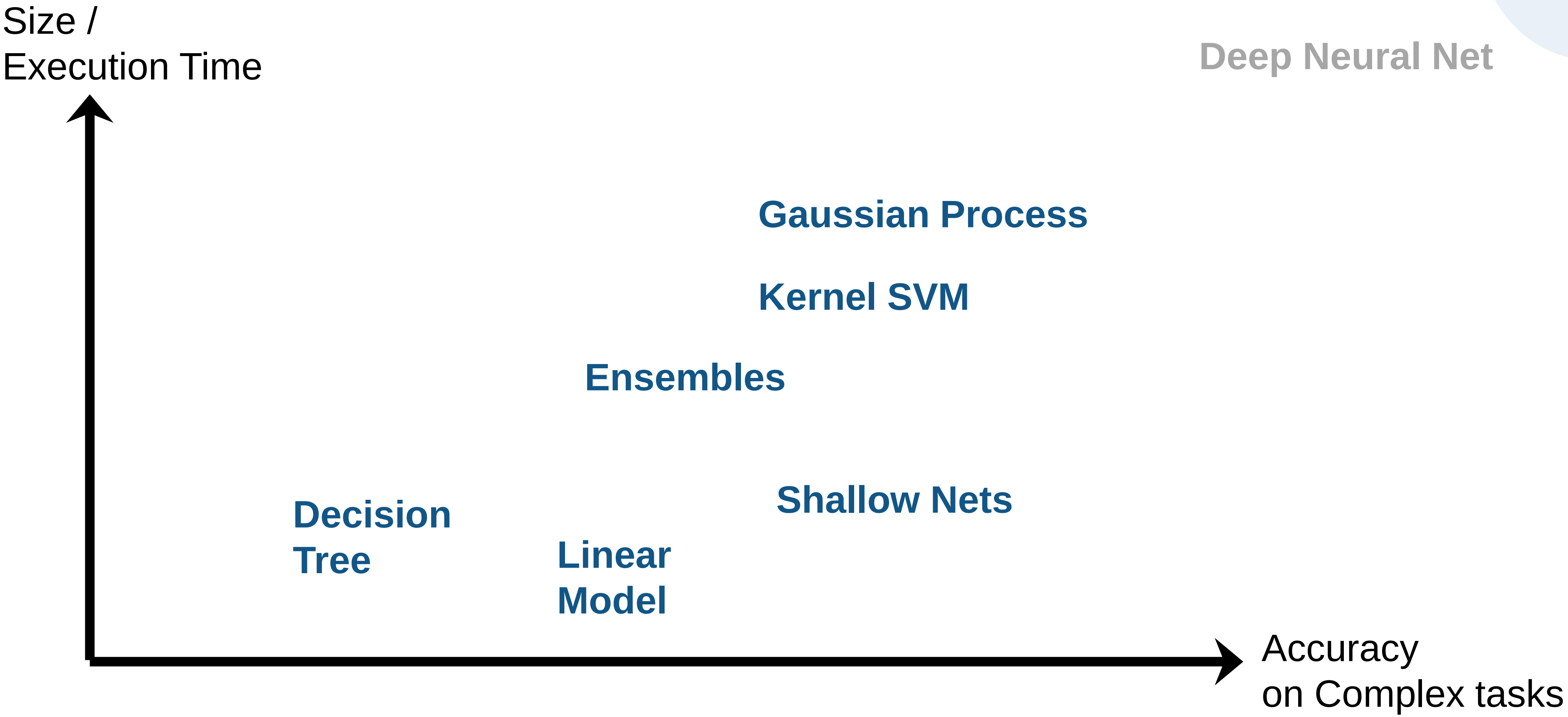
Deploy & Integrate

Compressing Machine Learning

Step 1 Size aware model selection

Small

Large



Size / Execution Time

Deep Neural Net

Gaussian Process

Kernel SVM

Ensembles

Shallow Nets

Decision Tree

Linear Model

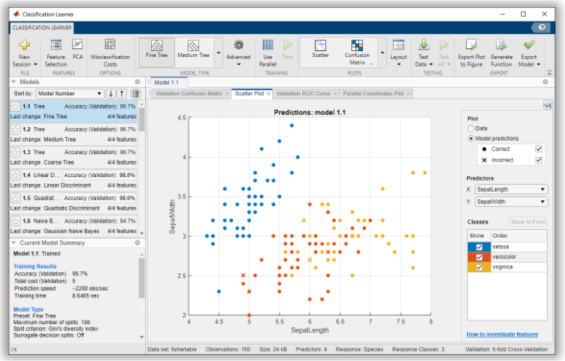
Accuracy on Complex tasks

Model Compression Workflow for Machine Learning

Small Large

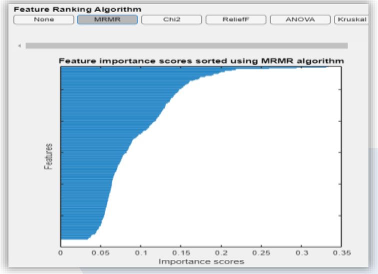
1

Classification / Regression Learner

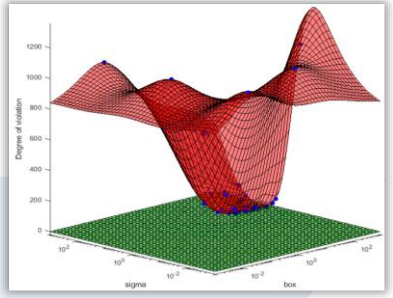


In-App Feature Selection

2

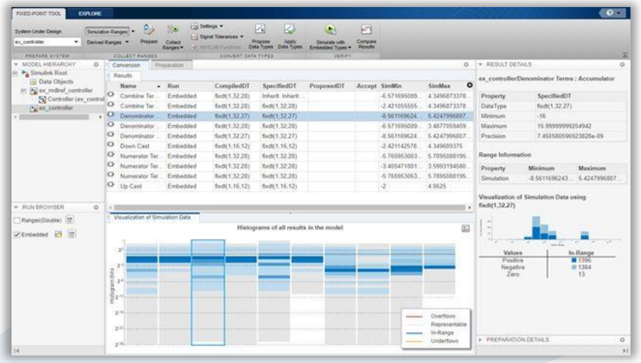


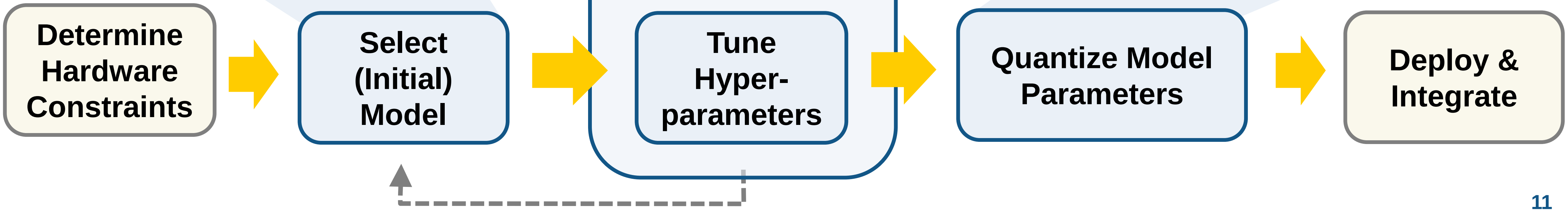
Bayesopt



3

Fixed Point Designer / Native Simulink Block




Determine Hardware Constraints

Select (Initial) Model

Simplify Model

Select Features

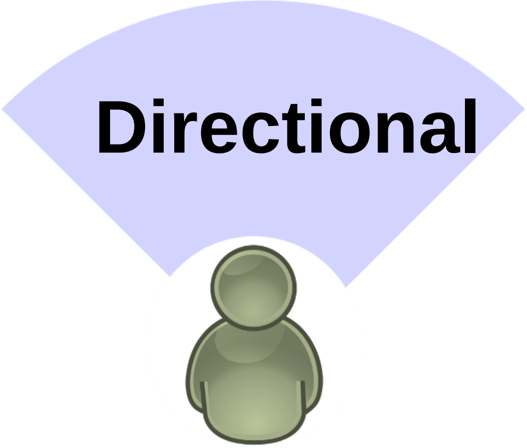
Tune Hyper-parameters

Quantize Model Parameters

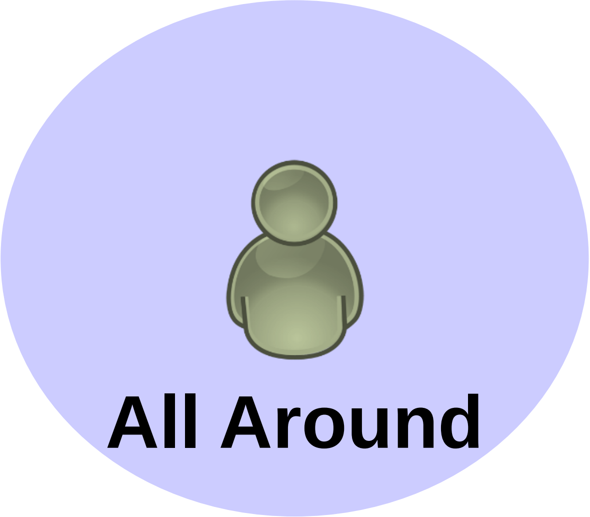
Deploy & Integrate

Demo: Embedding AI in an intelligent Hearing Aid

0.5 to 256 kB
on-chip memory

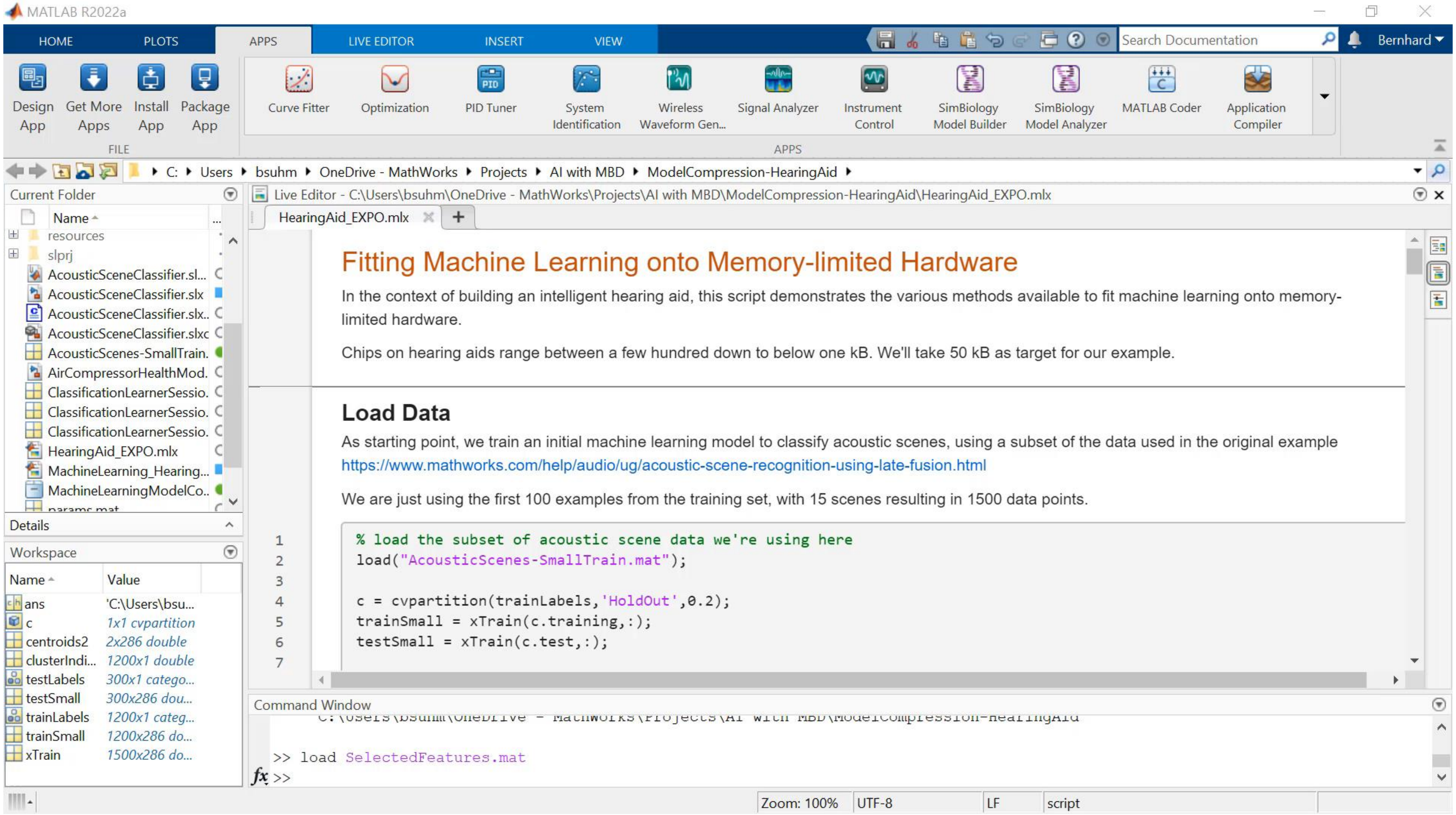


Directional



All Around

Demo: Fit Machine Learning for Intelligent Hearing Aid


MATLAB R2022a
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
Search Documentation
Bernhard
Design App
Get More Apps
Install App
Package App
FILE
Curve Fitter
Optimization
PID Tuner
System Identification
Wireless Waveform Gen...
Signal Analyzer
Instrument Control
SimBiology Model Builder
SimBiology Model Analyzer
MATLAB Coder
Application Compiler
APPS
C: ▸ Users ▸ bsuhm ▸ OneDrive - MathWorks ▸ Projects ▸ AI with MBD ▸ ModelCompression-HearingAid ▸
Current Folder
Name
resources
slprj
AcousticSceneClassifier.sl...
AcousticSceneClassifier.slx
AcousticSceneClassifier.slx..
AcousticSceneClassifier.slxc
AcousticScenes-SmallTrain.
AirCompressorHealthMod.
ClassificationLearnerSessio.
ClassificationLearnerSessio.
ClassificationLearnerSessio.
HearingAid_EXPO.mlx
MachineLearning_Hearing...
MachineLearningModelCo..
params.mat
Details
Workspace
Live Editor - C:\Users\bsuhm\OneDrive - MathWorks\Projects\AI with MBD\ModelCompression-HearingAid\HearingAid_EXPO.mlx
HearingAid_EXPO.mlx
Command Window
Zoom: 100%
UTF-8
LF
script

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

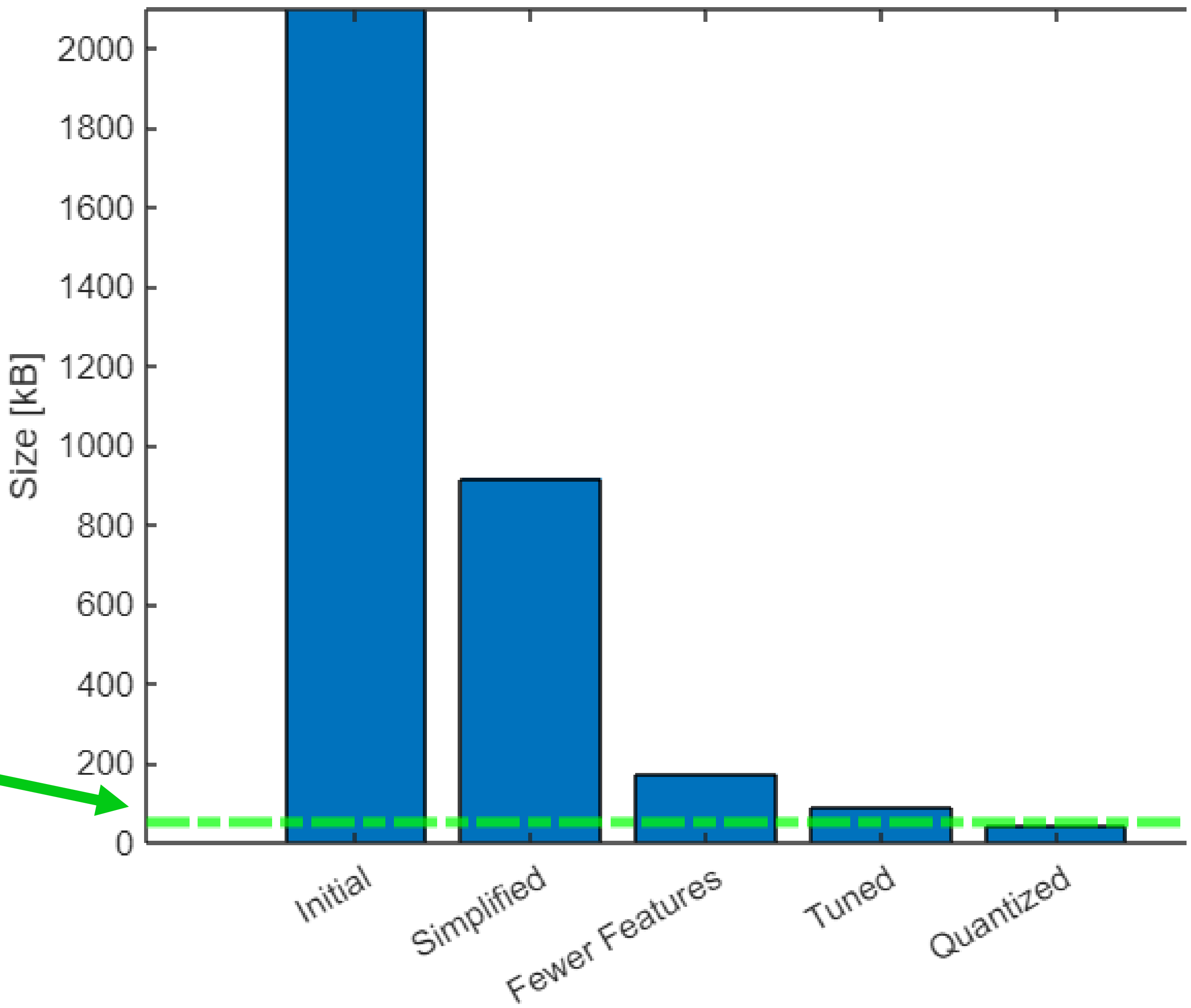
We are just using the first 100 examples from the training set, with 15 scenes resulting in 1500 data points.

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

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

Name	Value
ans	'C:\Users\bsu...
c	1x1 cvpartition
centroids2	2x286 double
clusterIndi...	1200x1 double
testLabels	300x1 catego...
testSmall	300x286 dou...
trainLabels	1200x1 categ...
trainSmall	1200x286 do...
xTrain	1500x286 do...

```
C:\Users\bsuhm\OneDrive - MathWorks\Projects\AI with MBD\ModelCompression-HearingAid

>> load SelectedFeatures.mat
fx >>
```

Machine Learning Demo Size Reduction by factor 20

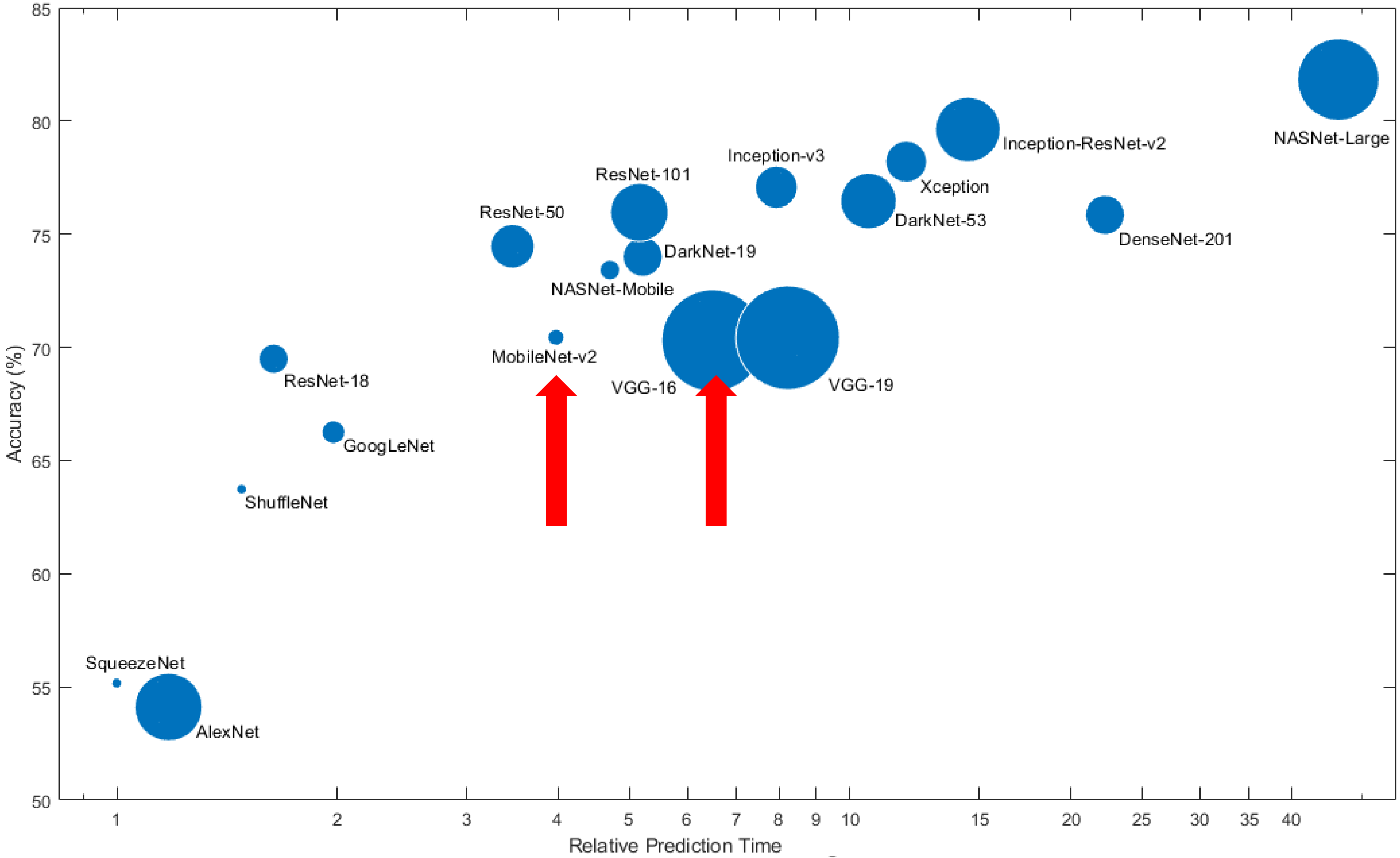


Target Hardware
Size: 50 kB

Compressing Deep Learning

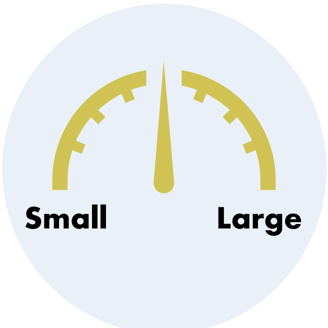
Step 1 Size aware model selection

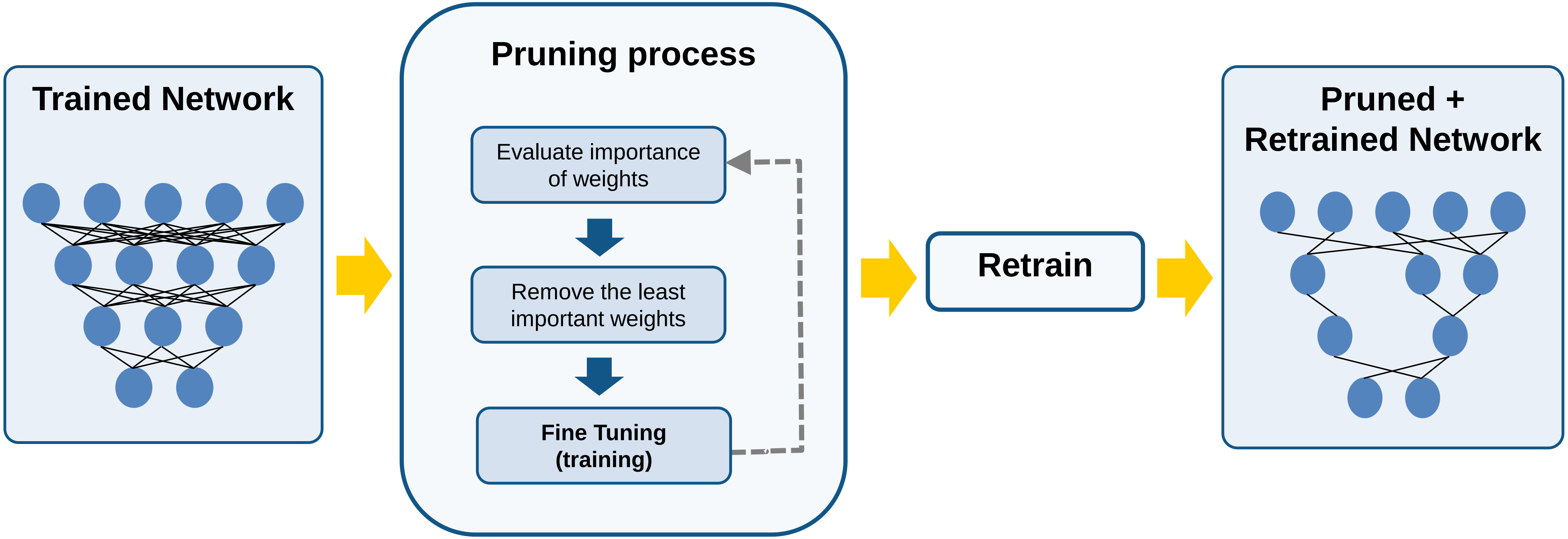
Small Large



Step 2 Smart pruning

Remove **unimportant** parts of the network





Trained Network

Pruning process

Evaluate importance of weights

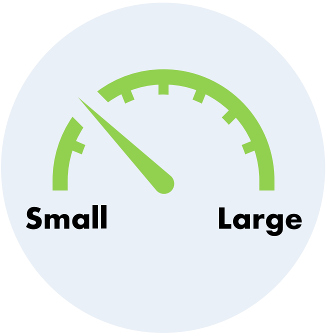
Remove the least important weights

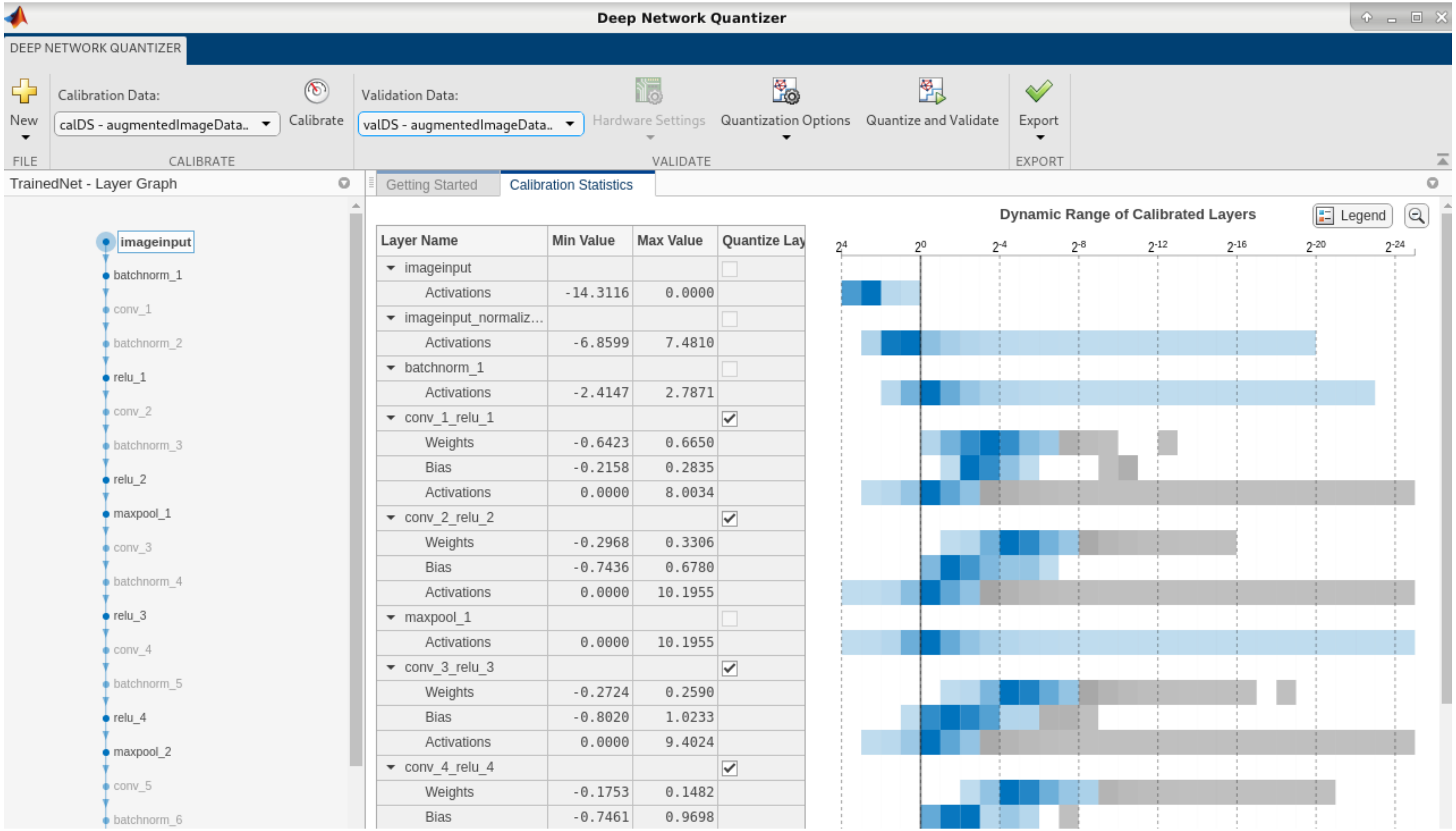
Fine Tuning (training)

Retrain

Pruned + Retrained Network

Step 3 Quantize your model





Layer Name	Min Value	Max Value	Quantize Lay
imageinput			□
Activations	-14.3116	0.0000	
imageinput_normaliz...			□
Activations	-6.8599	7.4810	
batchnorm_1			□
Activations	-2.4147	2.7871	
conv_1_relu_1			☑
Weights	-0.6423	0.6650	
Bias	-0.2158	0.2835	
Activations	0.0000	8.0034	
conv_2_relu_2			☑
Weights	-0.2968	0.3306	
Bias	-0.7436	0.6780	
Activations	0.0000	10.1955	
maxpool_1			□
Activations	0.0000	10.1955	
conv_3_relu_3			☑
Weights	-0.2724	0.2590	
Bias	-0.8020	1.0233	
Activations	0.0000	9.4024	
conv_4_relu_4			☑
Weights	-0.1753	0.1482	
Bias	-0.7461	0.9698	

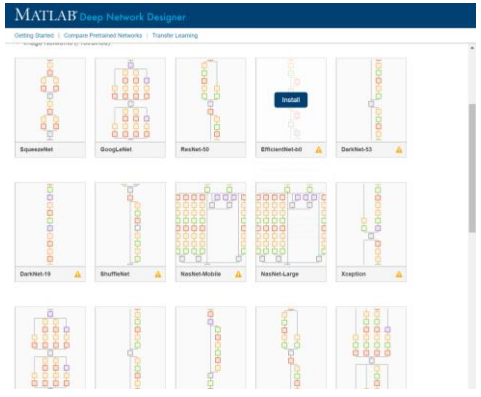
Deep Learning Demo: Scene classification

Classify 10 classes

More difficult problem → more complex model

Functionality for Compressing Deep Neural Nets

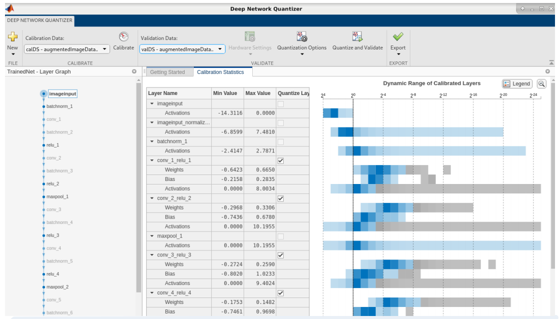
1 Deep Network Designer



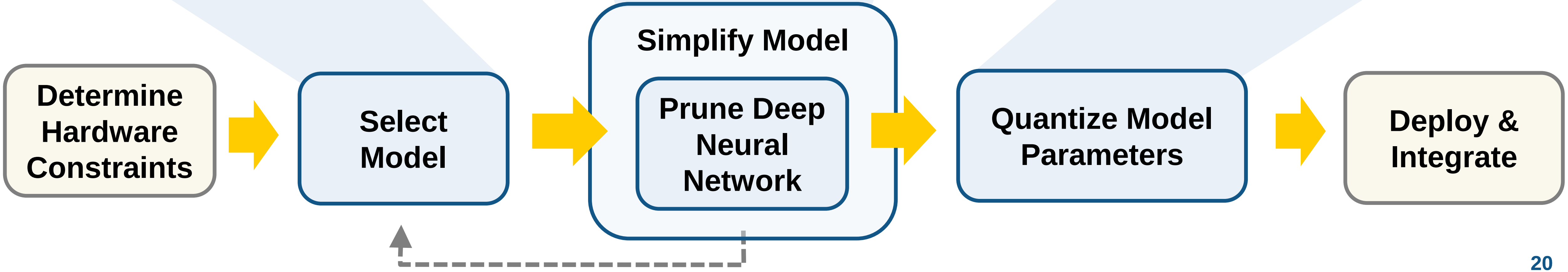
2 Taylor Pruning

```
taylorPrunableNetwork(net)
```

3 Deep Network Quantizer



Layer Name	Min Value	Max Value	Quantize Lay
▾ imageinput			□
Activations	-14.3116	0.0000	
▾ imageinput_normaliz...			□
Activations	-6.8599	7.4810	
▾ batchnorm_1			□
Activations	-2.4147	2.7871	
▾ conv_1_relu_1			☑
Weights	-0.6423	0.6650	
Bias	-0.2158	0.2835	
Activations	0.0000	8.0034	
▾ conv_2_relu_2			☑
Weights	-0.2968	0.3306	
Bias	-0.7436	0.6780	
Activations	0.0000	10.1955	
▾ maxpool_1			□
Activations	0.0000	10.1955	
▾ conv_3_relu_3			☑
Weights	-0.2724	0.2590	
Bias	-0.8020	1.0233	
Activations	0.0000	9.4024	
▾ conv_4_relu_4			☑
Weights	-0.1753	0.1482	
Bias	-0.7461	0.9698	



Determine Hardware Constraints

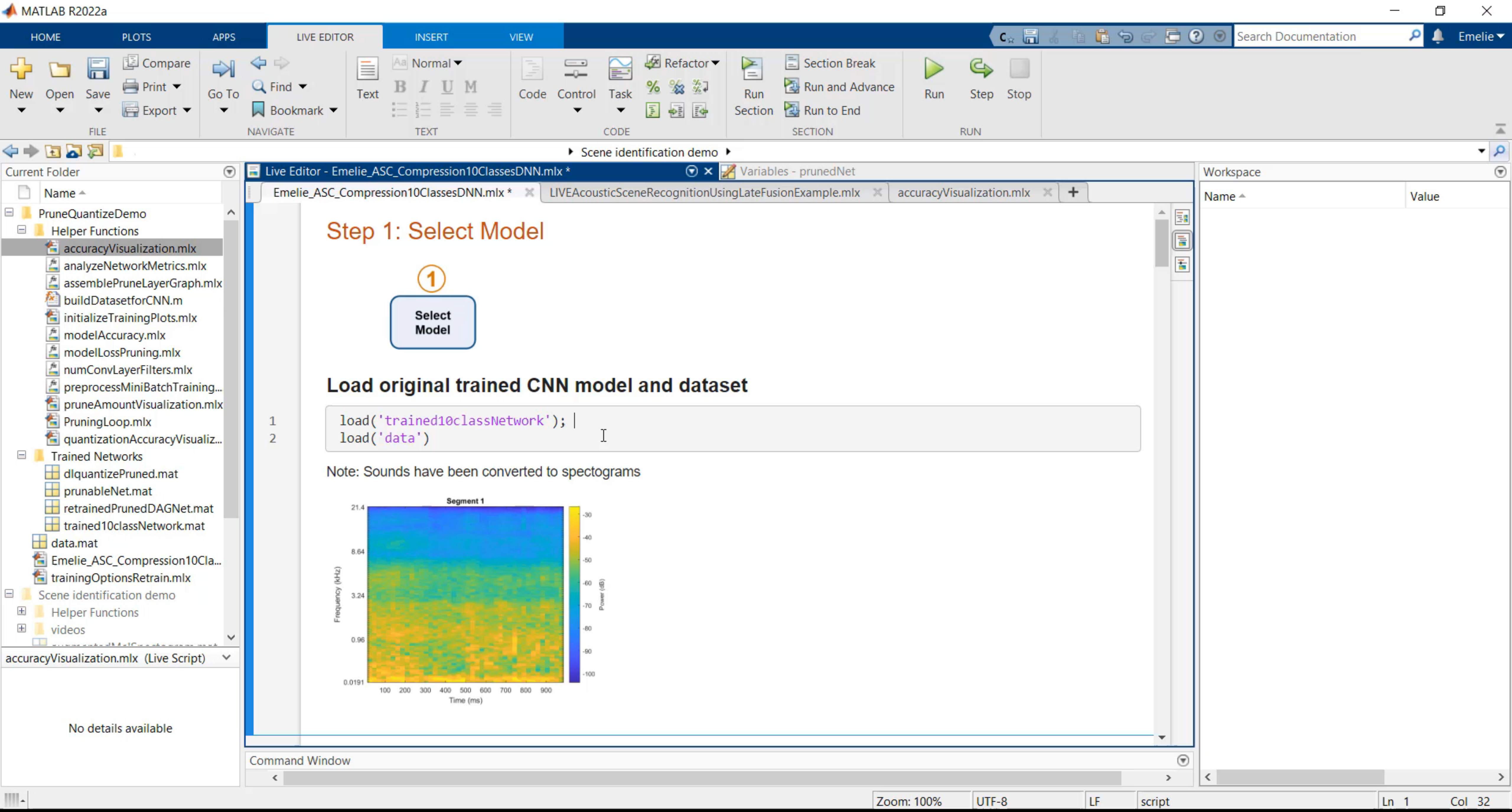
Select Model

Simplify Model

Prune Deep Neural Network

Quantize Model Parameters

Deploy & Integrate



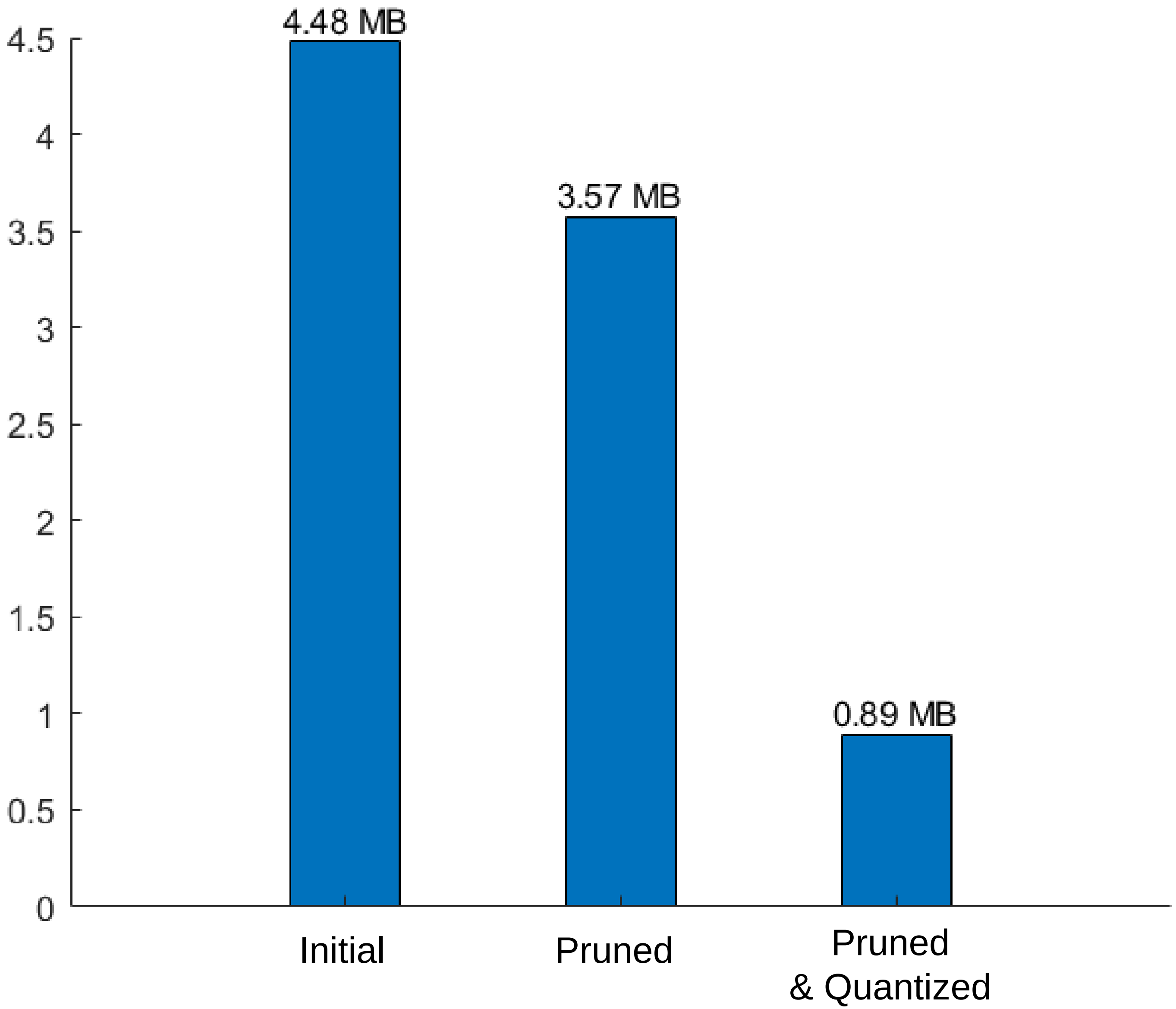
Step 1: Select Model

Load original trained CNN model and dataset

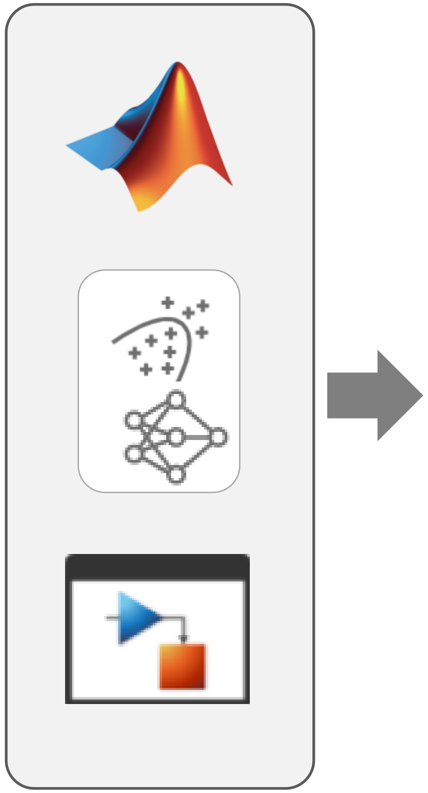
```
load('trained10classNetwork');
load('data')
```

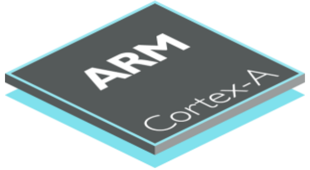
Note: Sounds have been converted to spectograms

Deep Learning Demo Size Reduction by factor 5



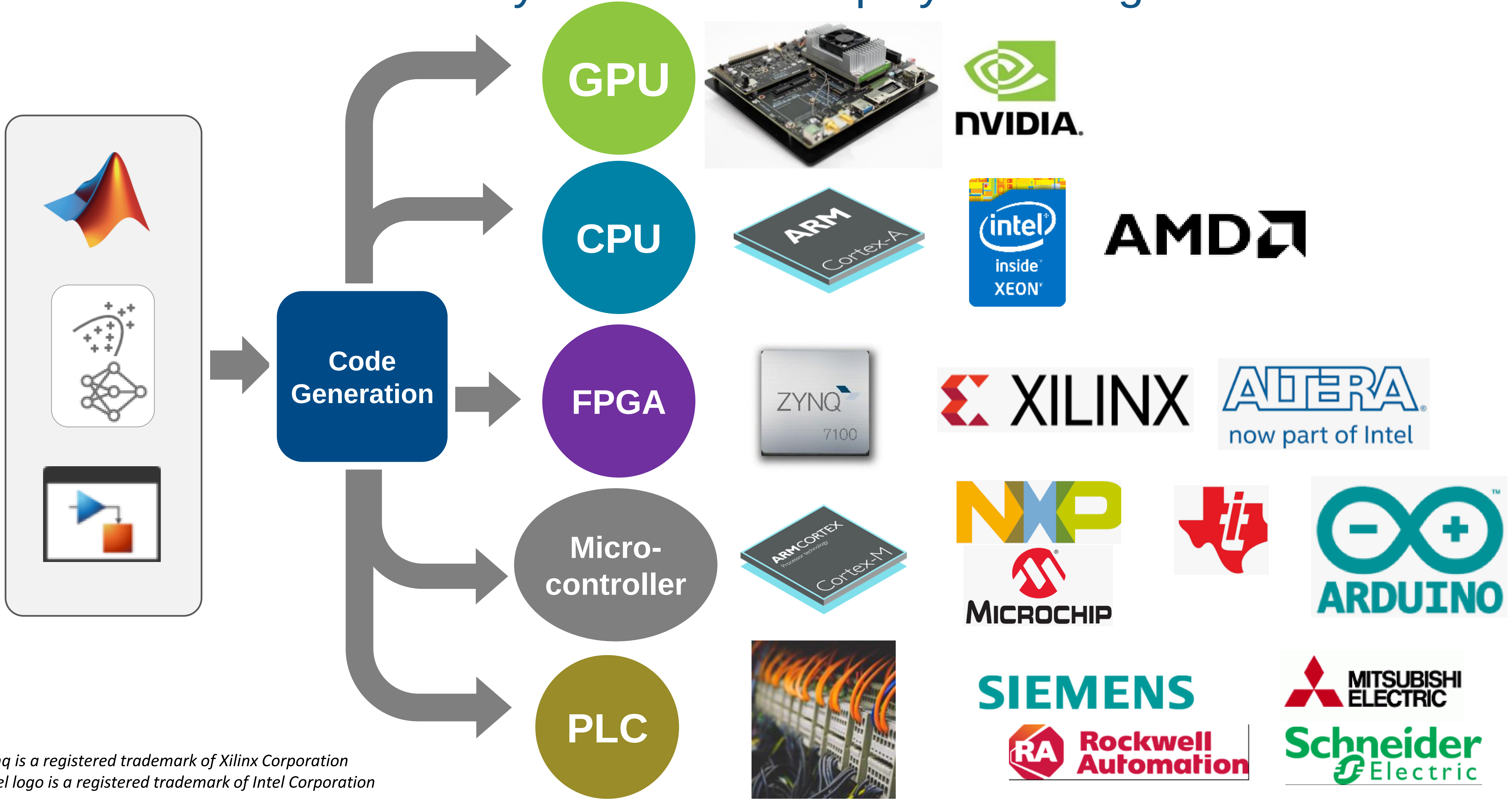
One Codebase – Many Embedded Deployment targets





Zynq is a registered trademark of Xilinx Corporation
Intel logo is a registered trademark of Intel Corporation

One Codebase – Many Embedded Deployment targets



Code Generation

GPU

CPU

FPGA

Micro-controller

PLC

Zynq is a registered trademark of Xilinx Corporation
Intel logo is a registered trademark of Intel Corporation

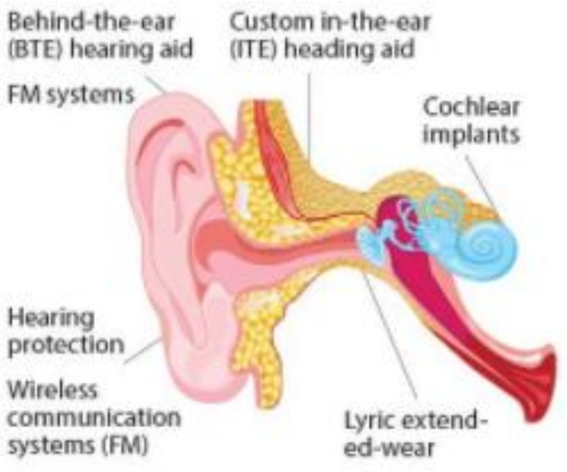
Conclusions

You can fit AI for many applications onto limited hardware

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

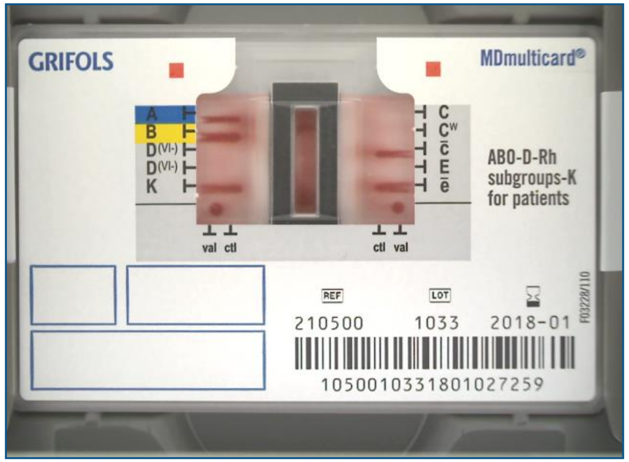
Same high-level Workflow for any type of AI

Learn More


Behind-the-ear (BTE) hearing aid
Custom in-the-ear (ITE) heading aid
FM systems
Cochlear implants
Hearing protection
Wireless communication systems (FM)
Lyric extend-ed-wear

Hearing Implant using MBD

Autonomous Tractor



Card to Classify Blood Type IDNEO

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)

MATLAB EXPO

Thank you

MathWorks®

MATLAB EXPO