Machine Learning for Embedded Software Engineers

Class 5: Near Real-time Machine Learning using Coral

April 26, 2019 Jacob Beningo



Presented by:



DesignNews

Course Overview

Topics:

- Introduction to Machine Learning
- Machine Learning Architectures for Embedded Systems
- Machine Learning Applications: Vision and Speech
- Machine Vision with OpenMV
- Near Real-time Machine Learning using Coral







Session Overview

- ML Processor Options
- Google Coral
- Embedded ML Examples
- Where to go from here





ML Processor Options







Google Coral TPU SOM

CPU	NXP i.MX 8M SOC (quad Cortex-A53, Cortex-M4F)	
GPU	Integrated GC7000 Lite Graphics	
ML accelerator	Google Edge TPU coprocessor	
RAM	1 GB LPDDR4	
Flash memory	8 GB eMMC	
Wireless	Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5GHz) Bluetooth 4.1	
Dimensions	48mm x 40mm x 5mm	

Source: Google



CONTINUING EDUCATION by:



Google Coral Dev Board



Source: Google





Google Coral Dev Board

Flash memory	MicroSD slot	
USB	Type-C OTG Type-C power Type-A 3.0 host Micro-B serial console	
LAN	Gigabit Ethernet port	
Audio	3.5mm audio jack (CTIA compliant) Digital PDM microphone (x2) 2.54mm 4-pin terminal for stereo speakers	
Video	HDMI 2.0a (full size) 39-pin FFC connector for MIPI-DSI display (4-lane) 24-pin FFC connector for MIPI-CSI2 camera (4-lane)	
GPIO	3.3V power rail 40 - 255 ohms programmable impedance ~82 mA max current	
Power	5V DC (USB Type-C)	
Dimensions	88 mm x 60 mm x 24mm	

Source: Google





Embedded ML Examples – STM32

1	Capture Data
2	Label Data
3	Train NN
4	STM32Cube.Ai
5	Run on STM32

DesignNews



Embedded ML Examples – STM32

/ STM32H743ZITx - NUCLEO-H743ZI /	fdmobilenetquant_h743_validontarget.ioc • Pinout & Configuration	File Edit Setup Control Window Help
Pinout & Configuration	Clock Configuration Additional Softwares	Results for "ascfloat", 16 inferences @400MHz/200MHz (complexit) y: 517361 MACC) duration : 9.737 ms (average)
ADZ	STMicroelectronics X-CUBE-AI 2.0.0 Mode and Configuration Configuration Reser Configuration Alst indexive Mark Protein Statings denoblenet.guant temphilemet.lost Mark Protein Statings denoblenet.guant temphilemet.lost Mode: binet/lost Kerais Visitation inputs: Rendom numbers V Validation inputs: Rendom numbers V Validation inputs: Rendom numbers V Validation onputs: Tome Compression: 56 07 KBytes Rend. 200 K	CPU cycles : 3895128 -2144/+1914 (average,-/+) CPU Workload : 0% cycles/MACC : 7.52 (average for all layers) used stack : 56 0 bytes used heap : DISABLED or NOT YET SUPPORTED Running PerfTest on "ascquant" with random inputs (16 iteration s) Results for "ascquant", 16 inferences @400MHz/200MHz (complexit y: 517361 MACC) duration : 4.188 ms (average) CPU cycles : 1675307 -2491/+3269 (average,-/+) CPU Workload : 0% cycles/MACC : 3.23 (average for all layers) used stack : 40 8 bytes used heap : DISABLED or NOT YET SUPPORTED
Destination Council 1 10004/12D Convolutional) 1, 112 2 30009/1000 Convolutional) 1, 112 3 10004/12D Convolutional) 1, 112 4 10004/12D Convolutional) 1, 102 4 10004/12D Convolutional) 1, 56 5 10009/12D Convolutional) 1, 56 6 10004/12D Convolutional) 1, 56 1 10004/12D Convolutional) 1, 57 8 10004/12D Convolutional) 1, 57 9 10004/12D Convolutional) 1, 57 10 10004/12D Convolutional) 1, 37 10 10004/12D Convolutional) 1, 32 10 10004/12D Convolutional) 1, 32 10 10004/12D Convolutional) 1, 32 11 11 1, 32 12 10004/12D Convolutional) 1, 32 13 10004/12D Convolutional) 1, 53 13 10004/12D Convolutional) 1, 53 13 10004/12D Convolutional) 1, 54 1	303 1, 132, 140, 45, 232 1, 132, 140, 145, 232 1, 132, 140, 10,055 56, 401, 13, 305 26, 401, 0,205 26, 401, 0,205 26, 401, 0,205 26, 401, 0,205 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,718 28, 401, 5,428 28, 401, 5,428 28, 401, 5,428 28, 401, 5,428 28, 401, 5,428 28, 401, 5,428 29, 401, 5,428 29, 401, 5,428 29, 41, 588, 5,428 21, 28, 401, 5,428 21, 28, 401, 5,428 21, 28, 401, 5,428 21, 28, 401, 5,428 21, 28, 401, 5,428 21, 28, 401, 5,428 21, 28, 401, 5,428	Running PerfTest on "fdmobilenetquant" with random inputs (16 i terations) Results for "fdmobilenetquant", 16 inferences @400MHz/200MHz (c omplexity: 23086462 MACC) duration : 151.852 ms (average) CPU cycles : 60741057 -19913/+20251 (average,-/+) CPU workload : 15% cycles/MACC : 2.63 (average for all layers) used stack . : 39 6 bytes used heap : DISABLED or NOT YET SUPPORTED
14 18 10504/(ID Convelutions) (). 3		PR + 40 22 - 70 # 105 PM







Embedded ML Examples - NXP













A few key take-a-ways

- Machine learning can be applied to nearly any embedded system
- Machine learning at the edge will be here faster than you think but slower than marketing is pushing
- Using a DSP capable processor can dramatically speed-up inference execution on MCU's





A few key take-a-ways

- There is a trade-off, balance between inference accuracy and memory/execution speed
- Testing and validation may be challenging in the foreseeable future
- There are many ML frameworks. Choose the one that best fits your needs
- Experiment with different frameworks and processors to get a good feel for ML capabilities





Where to go from here

Learning Neural Network Resources

- Videos
 - <u>https://www.youtube.com/watch?v=aircAruvnKk</u>
 - <u>https://www.youtube.com/watch?v=IHZwWFHWa-w</u>
 - <u>https://www.youtube.com/watch?v=Ilg3gGewQ5U</u>
 - <u>https://www.youtube.com/watch?v=tleHLnjs5U8</u>
- Books
 - <u>http://neuralnetworksanddeeplearning.com/</u>
 - Deep Learning
 - Python Machine Learning by Example



Where to go from here

Experimentation

- Develop a Python ML example that analyzes hand writing digits
- Walk through caffe training and inference tutorial:
 - <u>http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/</u>
- Apply a ML model to OpenMV module
- Setup and train an image classifier using Coral
 - Uses TensorFlow Lite for training





Where to go from here

Get Application Specific

- OpenMV
- STM32
- NXP Alexa Integration
- Keyword Spotting
- Object detection and classification
- Explore existing datasets





Additional Resources

- Download Course Material for
 - C/C++ Doxygen Templates
 - Example source code
 - Blog
 - YouTube Videos
- Embedded Bytes Newsletter
 - <u>http://bit.ly/1BAHYXm</u>



From <u>www.beningo.com</u> under

Blog > CEC – Machine Learning for Embedded Software
 Engineers

DesignNews



