Michael Deisher
michael.deisher@intel.com
Intel Corporation
Hillsboro, Oregon USA

IMPLEMENTATION OF EFFICIENT, LOW POWER DEEP NEURAL NETWORKS ON NEXT-GENERATION INTEL CLIENT PLATFORMS

Andrzej Polonski
andrzej.polonski@intel.com
Intel Technology Poland
Gdansk, Poland

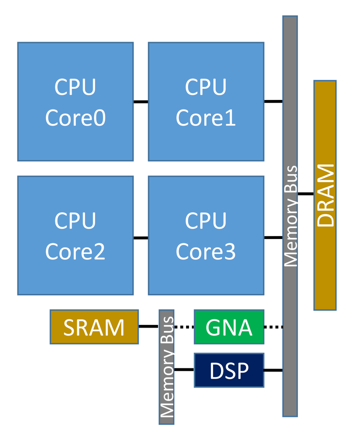
What is GNA?

Low power neural co-processor for continuous inference at the "edge"

Designed for Intel® Quark™, Intel Atom®, and Intel® Core™ based devices

Runs while application processor is in low power sleep state

Interfaces to system or private memory avoiding CPU cache pollution



How does GNA work?

Neural network topology stored in memory as list of layer descriptors

Layer types: affine, diagonal affine, Gaussian mixture model, recurrent, convolutional1D, transpose, copy

Activation function: piecewise linear (PWL) approximation

Layer Type	Input Orientation	Output Orientation	Batch Size	Weight Width	PWL Activation	Partial Output
Affine	column vector	column vector	1-8	1B, 2B	optional	yes
Diagonal Affine	column vector	column vector	1-8	1B, 2B	optional	no
Convolutional	row vector	row vector	1	2B	optional	no
Gaussian Mix	column vector	column vector	1-8	1B	no	yes
Recurrent	row vector	row vector	1-8	1B, 2B	required	no
Copy	row vector	row vector	1-8	N/A	N/A	no
Interleave	row vector	column vector	1-8	N/A	N/A	no
Deinterleave	column vector	row vector	1-8	N/A	N/A	no

All-integer math (inputs, outputs, weights, biases)

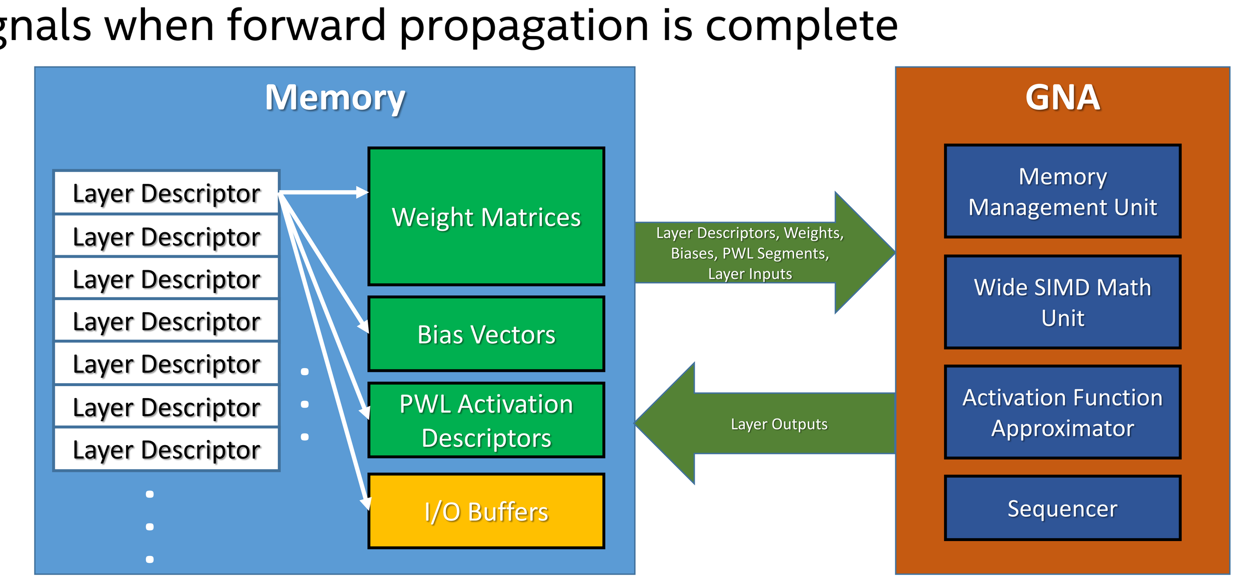
Batching of layer inputs for better throughput

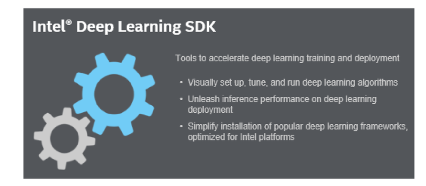
Optional on-the-fly pruning (affine layer)

Complex graphs (e.g., LSTM, GRU, TDNN) constructed from basic layer types

Stream processing model

- App processor configures memory, starts GNA, and sleeps or does useful work
- GNA signals when forward propagation is complete



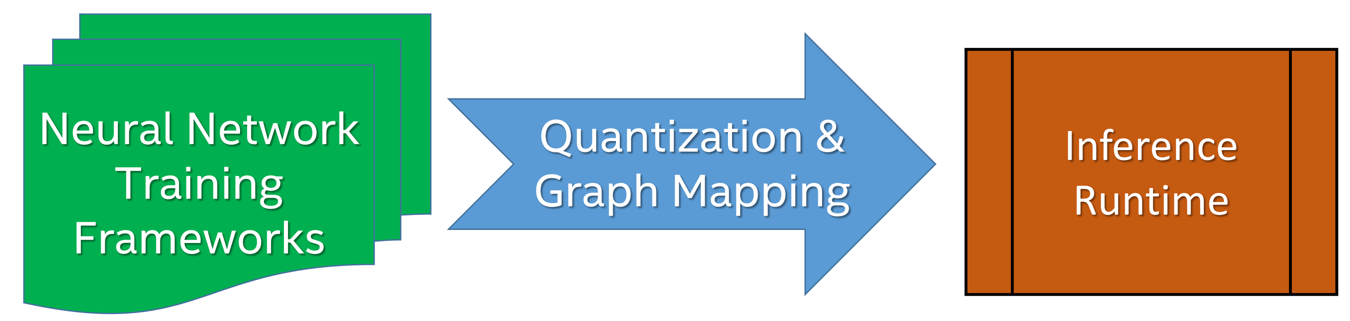


How is GNA used?

Start with floating point neural network trained in framework of choice

Import using Intel® Deep Learning SDK Deployment Tool or Kaldi example

Link with Intel® Deep Learning SDK Inference Engine or GNA native library



Kaldi, Tensorflow, CNTK, Theano, Caffe | Intel DL SDK, GNA Example Code | Intel DL SDK, GNA Native Library

GNA native library options

- Firmware API (for Intel® Quark™ running real-time operating system)
- Middleware API (for Intel Atom® and Intel® Core™ running Linux, Windows)

API Function	Description
GNADeviceOpen	acquire handle to GNA device
GNADeviceClose	release handle to GNA device
GNAAlloc	allocate memory (and pin so it cannot be swapped out)
GNAFree	free GNA memory (after unpinning)
GNAPropagateForward	propagate inputs through all layers of network
GNAWait	wait until propagation request completes or timeout

Complexity is hidden in layer descriptor list construction

- Handled transparently by Intel® Deep Learning SDK model optimizer
- Full control via GNA native library API

Intel® Deep Learning SDK Deployment Tool

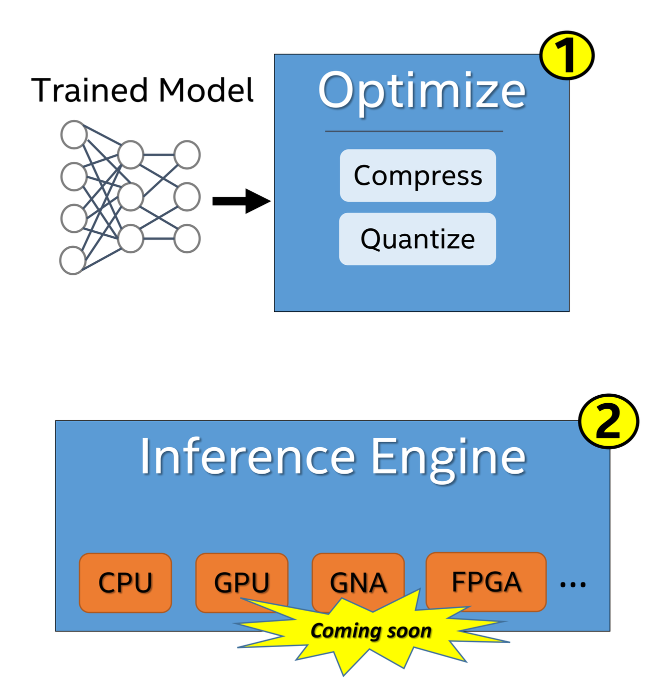
Enables full utilization of IA Inference while abstracting HW from developers

Optimize:

1 •Imports trained models from all popular DL framework regardless of training HW
•Model Canonicalization, Compression and Quantization

Deploy:

2 •One API across all Intel HW and systems
•Friendly Inference solution: (low footprint, easy API, control meeting Functional Safety)
•Optimizes Inference execution per target hardware under-the-hood



Ease of use + Embedded friendly + Extra performance boost

https://software.intel.com/en-us/deep-learning-sdk

What You Are Seeing: Live ASR Demonstration

Video playing TED talk by Aimee Mullens: The opportunity of adversity

Customer speech recognizer performing local real-time transcription

Acoustic model: 6-layer DNN w/ 2048 hidden nodes trained on 3000 hrs

Acoustic likelihood scoring is selectable:

- Native (customer optimized), GNA SW emulation (CPU), or GNA (HW)

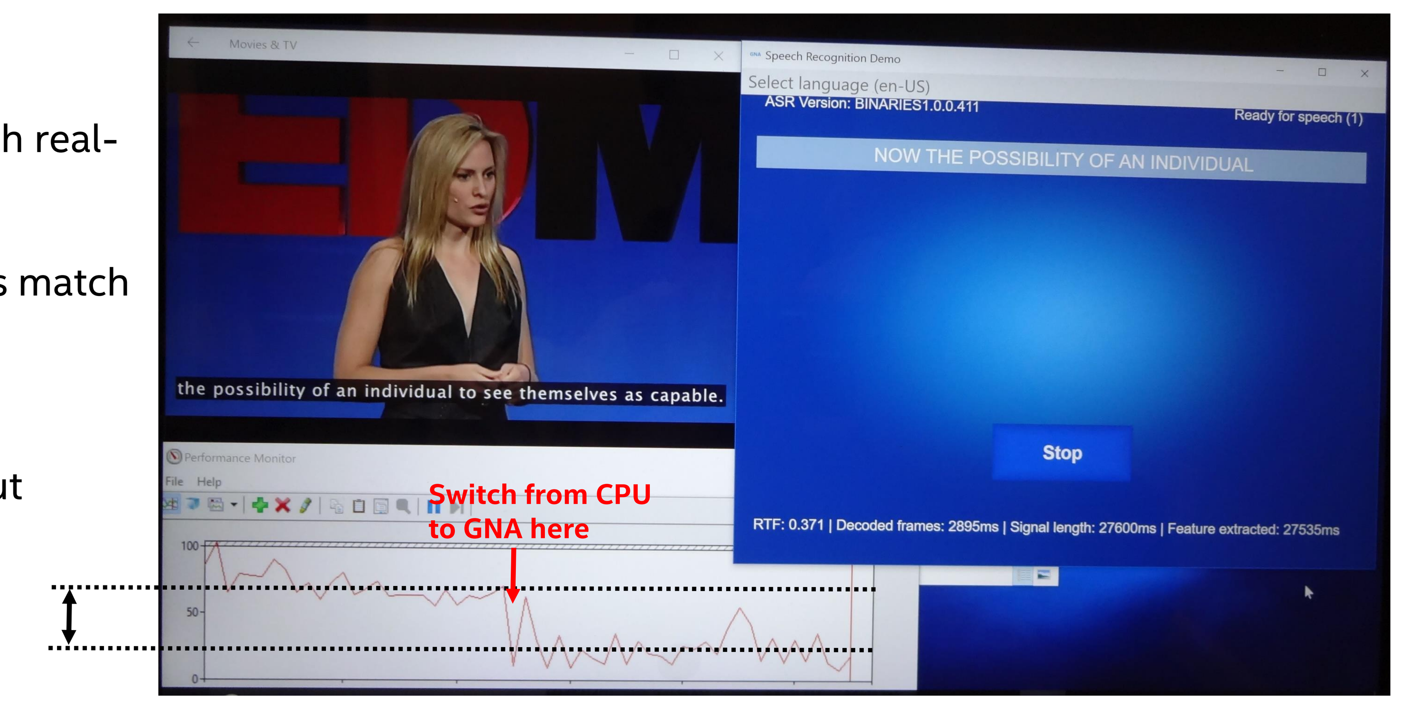
Performance monitor shows drop in CPU residency with GNA HW

Corresponding drop in power consumption not shown

Video playback with real-time transcription.

Recognition results match closed captioning.

Offload saves about 50% of a 1.1GHz Atom™ core in this configuration*



*Note that other configurations (e.g., higher CPU clock, use of more CPU cores, etc.) may improve performance at cost of higher power consumption

What You Are Seeing: Performance Demonstration

Real-time visualization of acoustic log-likelihoods

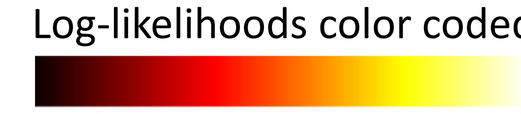
- DNN: additional layer added mapping outputs to phones
  - Ordered by class: non-speech, unvoiced fricatives, voiced stops, unvoiced stops, voiced fricatives, liquids & glides, nasals, front vowels, mid vowels, back vowels, diphthongs
- LSTM: CTC-trained phone outputs in natural order

Repeating cycle: score on CPU, score on GNA, …

Shows difference in scoring speed between CPU and GNA HW

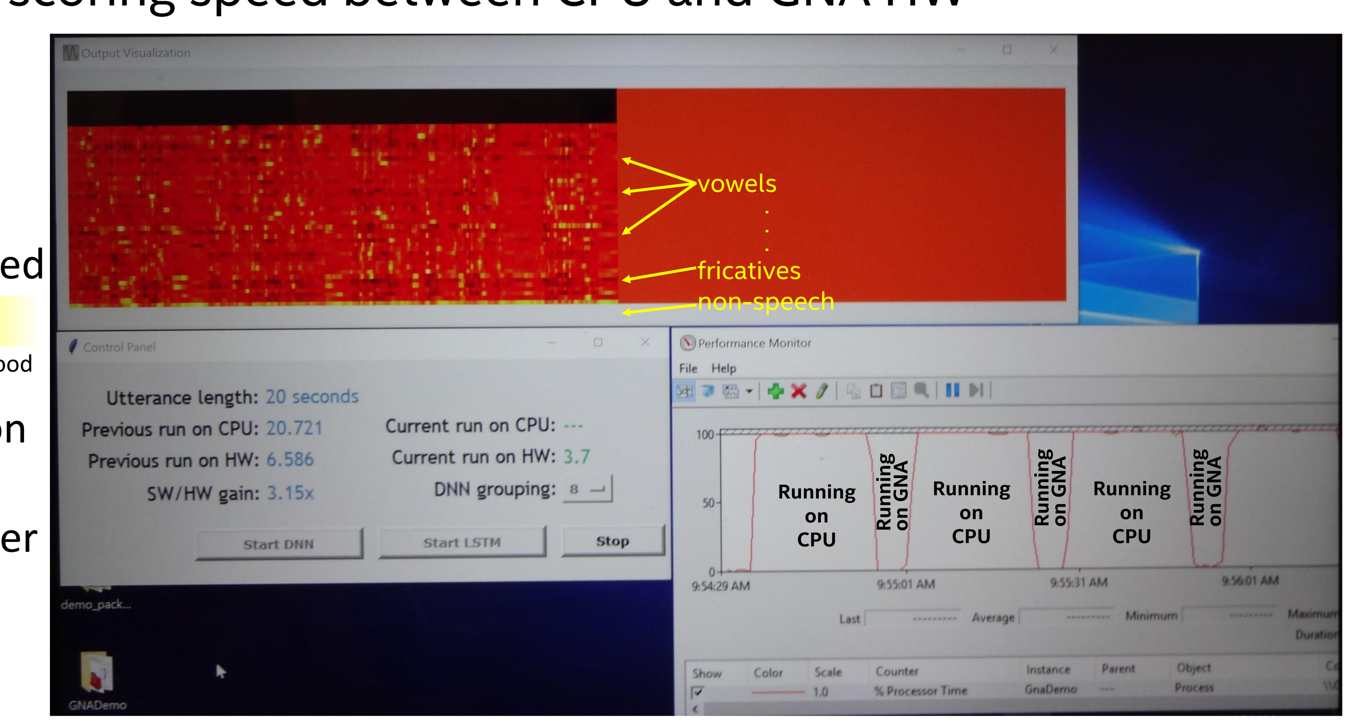
Batch scoring of 7-layer 2048 hidden node DNN

Log-likelihoods color coded



CPU utilization in Perfmon

Scoring on GNA is 3x faster than 1.1GHz Atom CPU in this configuration**



**Note that other configurations (e.g., higher CPU clock, etc.) may have lower utilization benefit while retaining significant power reduction