aicas technology

Deep Learning Applications in the Embedded Space

Mike Elliott
Software Engineer
aicas GmbH

Learning

A computer program is said to learn from experience E with respect to some task ***T*** and some performance measure ***P***, if its performance on ***T***, as measured by ***P***, improves with experience ***E***.

– Tom Mitchell (1997)

- Tasks
  - Classification
  - Quantization
- Deep Learning
  - Use of multiple layers of nonlinear processing

Machine Learning

At the highest level, is a computer process that extracts specific features from data to solve predictive problems:

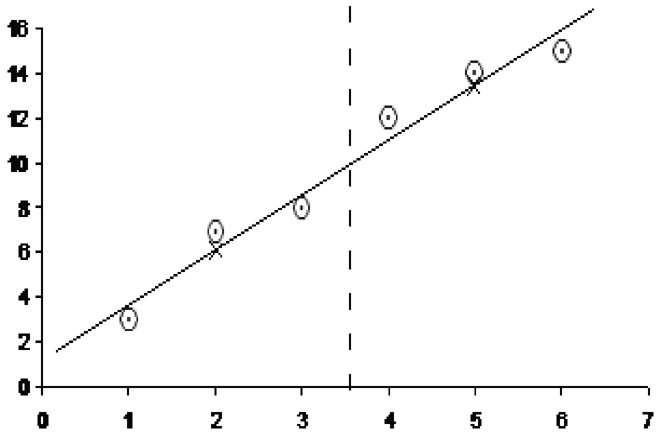
- Self-driving cars
- Classifying objects such as tumors
- Detecting pedestrians
- Detecting anomalies to prevent network intrusion or fraud
- Scanning social media sentiment and perception for marketing purposes
- Image interpretation

Algorithms

- Linear Regression
- Logistic Regression
- Neural Network (deep learning)
  - Modeled on biological neuron (dendrites, soma, axon)
- Convolved Neural Network (CNN)
  - Synthesized vision
    - Multi-stage Hubel-Weisel architecture
    - Work on a cat's primary visual cortex (1962)
- Training and inference

Linear Regression

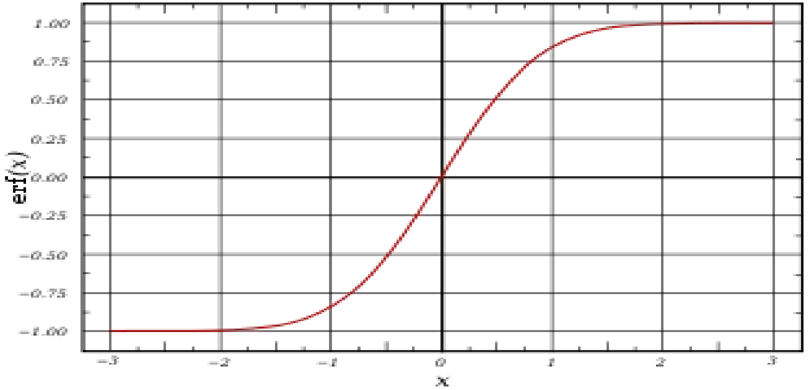
- Quantization
  - Predict housing price
  - Multiple features
  - Square footage, number of floors, etc.



- Plot some points
  - Lay a meter stick on the points to find the straight line
    - Okay, least squares does better
  - Multiple dimensions

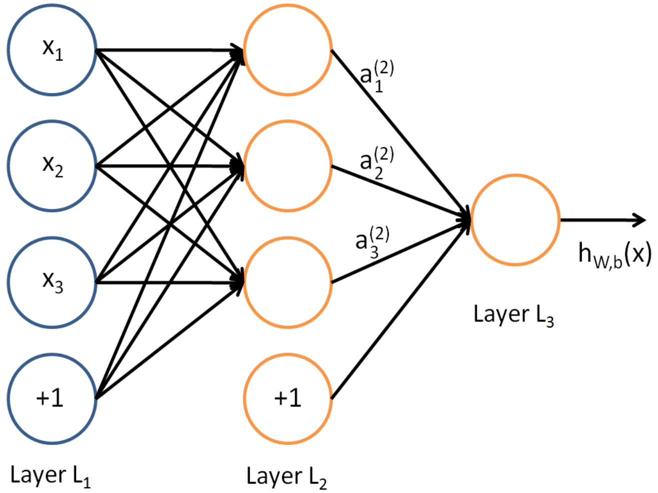
Logistic Regression

- Classification
  - Is the tumor malignant?
  - Is this image a 7?
- Yields a probability
- Sigmoid function
  - Easy to calculate derivative
- The term regression is retained for historical reasons



Neural Network

- Modeled on actual neural tissue
  - Dendrites, soma, axon
- Multi-layer perceptron
  - Input Layer
  - Hidden Layer (or layers)
  - Output Layer
- Convolutional Neural Network (image recognition)
- Training through back propagation and gradient descent



Languages

- Algorithm Development
  - Octave, Matlab, R, Python (NumPy)

- Production
  - C/C++
    - OpenCL, CUDA
  - Java (or any JVM language)
    - Java, Scala, Clojure, etc.
    - OpenCL, CUDA

Hardware Constraints

- Power, Cooling, Memory
  - Ideally on the order of 1 W
- Processor
  - CPU (Central Processing Unit)
  - GPGPU (General Purpose Graphical Processing Unit)
  - FPGA (Field Programmable Gate Array)
  - ASIC (Application Specific Integrated Circuit)
- Sensors
  - I/O bandwith

GPGPU

- Well understood
  - CUDA (NVidia)
  - OpenCL (Apple)
  - Parallelism inherent
  - Every processor executes every instruction
- Power hungry
- Adjunct to CPU
  - Limited data bandwidth

FPGA

- Can reduce power consumption by 1 or 2 orders of magnitude
- Parallelization with Digital Signal Processors (DSPs)
  - More power efficient than signal processing on CPUs
- High I/O rate and bandwidth
  - 1000+ pins
- Block RAM can be used for an internal feature model
- Dynamically configure hardware at startup/runtime
  - Switch image filters

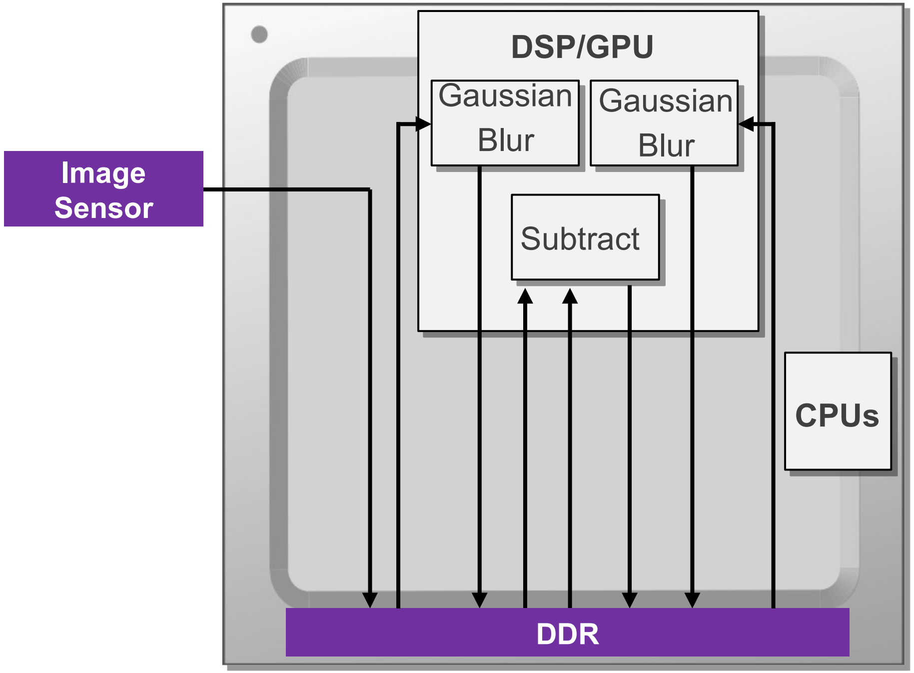
Software Development Considerations

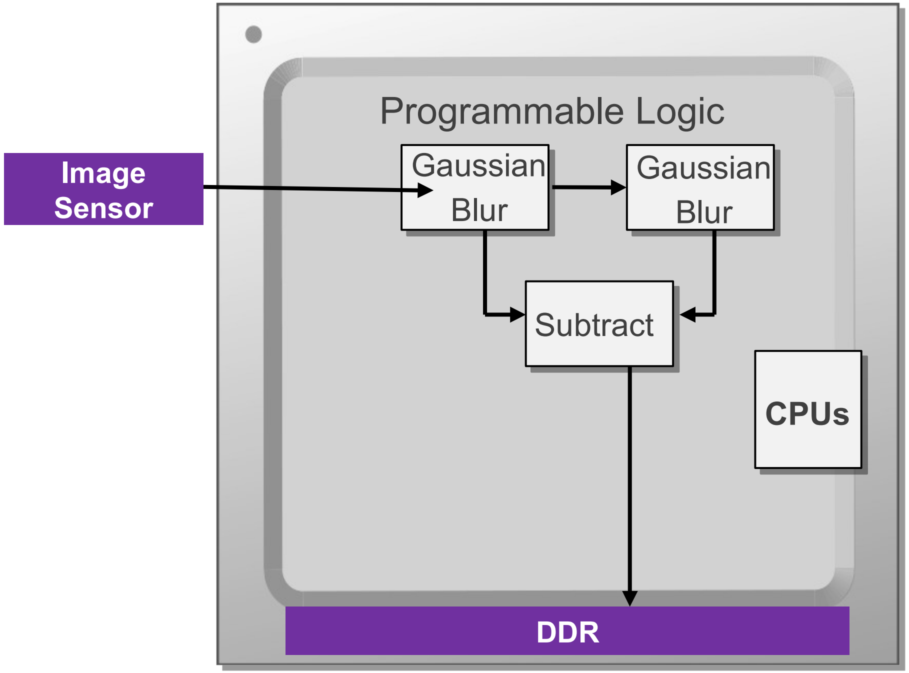
- Ease of programming
- Ease of deployment
- Ease of maintenance
- Systems training
- Simulation
- Portability
    - Code
    - Data
        - Trained model

Real-Time Constraints

- Determinism
  - Deadlines, jitter, latency
- Task partitioning
- Security
  - Civil airspace: DO-326, DO-355, DO-356
- Safety
  - Civil airspace: DO-178C, DO-322
  - Automotive: ISO 26262
- Inherent language safety is important

Latency and Pixel Streaming



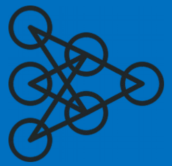


XILINX ALL PROGRAMMABLE.

Performance and Scaling

6x
Images/sec/watt

Machine Learning
Inference



42x
Frames/sec/watt

Computer Vision

		Xilinx ZU9	Xilinx ZU5	Nvidia TX1
GoogLeNet @ batch = 1	Images/s	**370.0**	**155.0**	70
	Power (W)	**7.0**	**4.5**	7.9
	Images/s/watt	**53.0**	**34.5**	8.9
		Xilinx ZU9	**Xilinx ZU5**	**nVidia TX1**
CV:: StereoLBM @1080p	Frames/s	**700**	**296**	28
	Power (W)	**4.8**	**3.3**	7.9
	Frames/s/watt	**145.8**	**89.7**	3.5
		Xilinx ZU9	**Xilinx ZU5**	**nVidia TX1**
CV:: LK Dense Optical Flow @720p	Frames/s	**170**	**73**	7
	Power (W)	**4.8**	**3.3**	7.9
	Frames/s/watt	**35.4**	**22.1**	0.9

Inference Models and Analysis

- Training can be (and usually is) separate from inference
- CNNs only capture spatial patterns in data.
  - If data is just as useful when exchanging columns, CNNs are wrong
- Tuning
  - Number of features, size of features
  - Window size, window stride
  - Number of neurons
  - How many layers and what type

Model Training

- Can be computationally expensive
  - Parallel processing – FPGA, GPGPU
- Distributed processing
  - Map reduce (Hadoop)
- Real-Time training
  - Unsupervised learning
  - Not only feedback, but an interpretation of correctness

CNN/DNN toolkits

- Deeplearning4j (Skymind)
    - Ready to run now using Real-time Java
- Caffe (Berkeley Vision and Learning Center)
- CNTK (Microsoft)
- Theano (University of Montreal, et. al.)
- TensorFlow (Google)
- Torch (Ronan Collobert)

References

- Scaling up Machine Learning (various)
- Deep Learning Demystified (Rohrer)
- Stanford CS 231 Course Notes (Karpathy)
- UMich EECS 598 Course Notes (Lee)
- The Black Magic of Deep Learning (Markou)
- Conv Nets, a Modular Perspective (Olah)
- Neural Information Processing Systems (NIPS) Proceedings

Product Lines

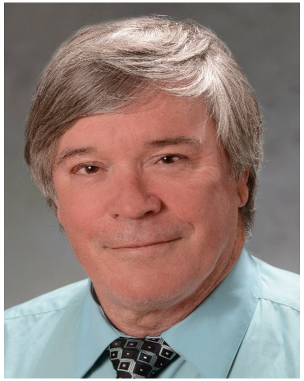
JamaicaVM

JamaicaCAR

Jamaica-IoT

JamaicaTrace

Presenter



Michael Elliott

aicas GmbH
Emmy-Noether-Str. 9
76131 Karlsruhe
Germany

elliott@aicas.de

+49 721 663 968-63
+1 562 645-3355

Michael Elliott is a software engineer with a deep passion for modern software practice, embedded systems architectures and safety- and security-critical software. His recent work has involved embedded systems running Java Virtual Machines (JVMs) and JVM languages with a focus on both safety critical aspects of software and image recognition with convolutional neural networks. Additionally, Mike was a member of the committee creating the standard for software used in aircraft operating in civil airspace (DO-178C) and modern software practices in airborne software (DO-332) along with airborne security standards including DO-326 and DO-355.

Mike has a Bachelor of Science in Information and Computer Science from the University of California, Irvine and a Master of Science in Software Engineering from Edinburgh University, Edinburgh, Scotland.