CURTISS-WRIGHT

Defense Solutions Division

Deep Learning for Cognitive EW with COTS

Chad Augustine, Product Manager, Integrated Systems

CURTISS-WRIGHT

Overview

- Important Notes on information and terminology
- What is Deep Learning?
- How to apply it
- Applicable Commercial-of-the-Shelf (COTS) Hardware and Software
- The Road Ahead

Note about the information in this presentation

- **All the information in this document is public**
- **It's all available via standard online sources (Wikipedia, Google, various "open courseware" websites, etc.)**
- **The goal of this presentation is to emphasize what anyone in the world can do to implement Cognitive systems using Deep Learning with COTS Hardware and Software**
- **The implication of that should be clear:**
  - The tools are widely available
  - Page 11 explains why this is not a problem

A note about a terminology overlap with the defense industry

- **Throughout this presentation, the following terms appear**
  - "Classify"
  - "Classifier"
  - "Classified"
- **These are fundamental terms in the theory and practice of Deep Learning**
- **In no place in this presentation are they used in the Defense industry context of marking and protecting information for security purposes**

Basic Decision Making in a Computer: "IF – THEN"

- **It's really simple logic, right? → "If there is good beer in my glass, then I will drink it."**
- **We can add all sorts of permutations to this:**
  - Variations of good beer, different types of containers, even different ways to drink it.
  - Even add options for drinking good things that aren't beer.
- **We can add "if-then" statements for everything, right?**
- **Isn't it just a matter of carefully taking into account *every single case, creating an exhaustive list of all combinations for every single imaginable type of beer, container, drinking style…… perhaps not.***
- **All the craft-brews (and home-brew) beers in the world make this functionally impossible. How would you ever keep it all up to date? What does "good" mean?**

Fundamentals of "Machine Learning"

- **Basically "Artificial Intelligence", or better "Weak Artificial Intelligence"**
  - The spam filter on your email, search engines, and your Netflix suggestions are all examples of applications of "Machine Learning"
- **Machine Learning requires some sort of training, e.g. what is good email vs. spam**
  - Reinforcement: Try, try again… until it does it right. Playing a game to win.
  - Supervised: teacher provides examples of input and output, guides the learning
  - Unsupervised : Free-form. Find and discover the patterns. Report back.
- **Using a "classifier" to separate what is vs. is not**
  - We can think of this as a "filter"

How is Deep Learning different than Machine Learning?

- **Deep Learning is a type of Machine Learning, just "deeper" filtering**
- **It "goes deeper" by classifying via many layers or dimensions of classification**
- **Information is passed through many classifiers, iteratively, e.g. Beer:**
  - Light or Dark?
  - Strength?
  - Wheat or hops?
  - Filter on other flavors – citrus, berries, spice, etc.
- **Each of these classifiers can then be trained for what is "good" and what is "bad"**
- **The sum of all these classifiers becomes a Deep Neural Net (DNN) which can be presented with an arbitrary input**
- **Based on the training, the DNN will be able to determine "good" or "bad" from that input**

Why is this important?

- **Deep Learning is a sophisticated way to train a computer to become a subject matter expert for a defined task, for example:**
  - Arbitrary handwriting and speech recognition
  - Autonomous driving https://blogs.nvidia.com/blog/2015/02/24/deep-learning-drive/
  - Game playing (e.g. "Go" – Google's deep learning system just beat the top human in the world – a major task far harder than winning at chess) https://www.technologyreview.com/s/546066/googles-ai-masters-the-game-of-go-a-decade-earlier-than-expected/
  - Sophisticated information analysis (e.g. finding and associating deep patterns and connections in high complexity signals, such as art, music, finance, travel, weather, etc.)
- **Deep Neural Nets can be trained in a high performance environment, then replicated on lower cost hardware**
  - For example, Qualcomm efforts to put real-time DNNs for facial recognition on phones
  - https://www.qualcomm.com/news/onq/2015/03/02/qualcomm-zeroth-advancing-deep-learning-devices-video
  - Some top level classifier training per user as needed.

Under the hood, what's the challenge?

- Classifiers are signal filters, and signal filters involve a lot of math (multiplication and addition) in many dimensions
- Deep learning applies many classifiers across signals, and mathematically combines all of those
- The classifiers don't just work on the whole signal at once, but instead process the signal in many pieces (e.g. analyzing a small chunk of a picture at a time)
- Add up all of this, and it is a huge amount of math to perform on a signal
- For real time applications, like autonomous driving, it's a lot of math, and done really fast. And with that, a lot of storage (memory) to hold the signal and all the filtered signals (and filters of those filters, etc.) is a huge challenge.

We could not do this just a few years ago

What's changed?

- Lots and lots of cheap storage and memory allows us to hold all this data
- Graphics Processing Units (for games!) are really good at doing all this highly complex math in many dimensions (GPGPU)
- High speed interconnects and memory allow us to shove all this data in and out of the system without bottleneck
- Lots of open source software from both academia and industry (e.g. Google) for Deep Learning
- Anyone can do this. Here's a guide to build a system at home for <$1000, all with standard parts from the consumer PC world.
  - http://timdettmers.com/2015/03/09/deep-learning-hardware-guide/

Applying to a field of interest for Cognitive capabilities

APPLICATION OF TECHNOLOGY

- What are the fundamental signals of interest? E.g. Frequencies, shapes, etc.
- Subject matter experts (humans) can design and implement classifiers appropriate to these signals
- The subject matter-specific Deep Neural Nets need to be trained by the right human SMEs in the right settings and environments (simulated and real)
- These trained DNNs can then be "cloned" and deployed, and can be updated upon additional training of the original DNN.

INTELLECTUAL PROPERTY

- The selection of signals of interest
- The design of the classifiers, and the human SME knowledge that goes into that design
- The training process, setting, the actual hardware and software system hosting the DNN to be trained
- The trained DNN itself – it should be considered valuable "Intellectual Property" to be protected
- Valuable like a trained dog and the training methodology, but this dog can be cloned with training intact.

Applicable Commercial-of-the-Shelf (COTS) Hardware

- **Graphics Processing Units (GPGPU)**
  - Very flexible for training and usage
- **Field-Programmable Gate Arrays**
  - Good for cloning a DNN in a less “general purpose” form that isn’t expected to be used as the training system
  - Good for creating sophisticated classifiers which may be highly complex or specialized
- **Processing, Storage, Networks**
  - The overall data and process infrastructure. Some light use on lower end processing cores, more use for processors in supercomputers in parallelization.
- **A system of these building blocks can be scaled up or down, from small embedded systems on chip (e.g. Nvidia Tegra) to supercomputers**

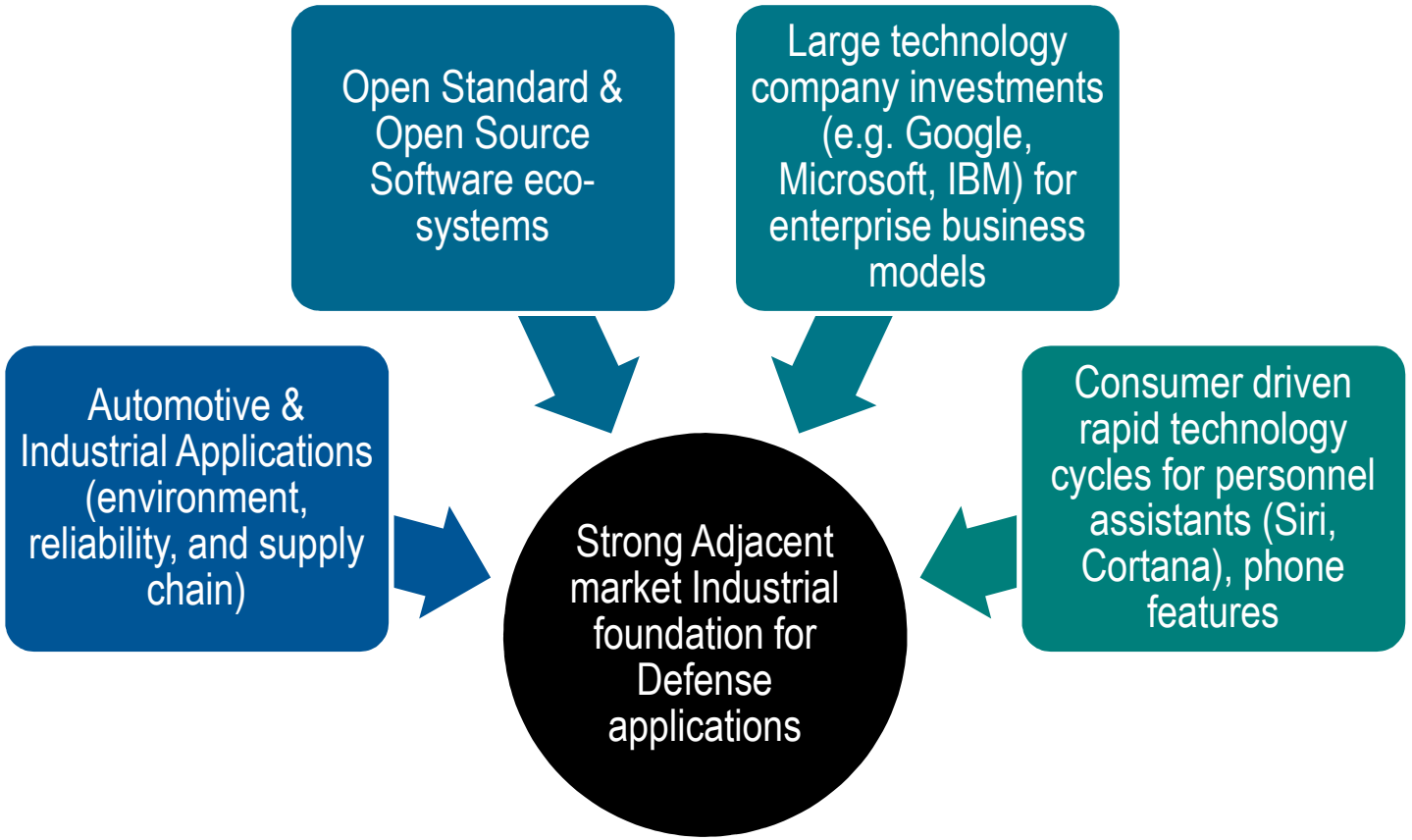
Applicable Commercial-of-the-Shelf (COTS) Software

Current March 2016 list on Wikipedia

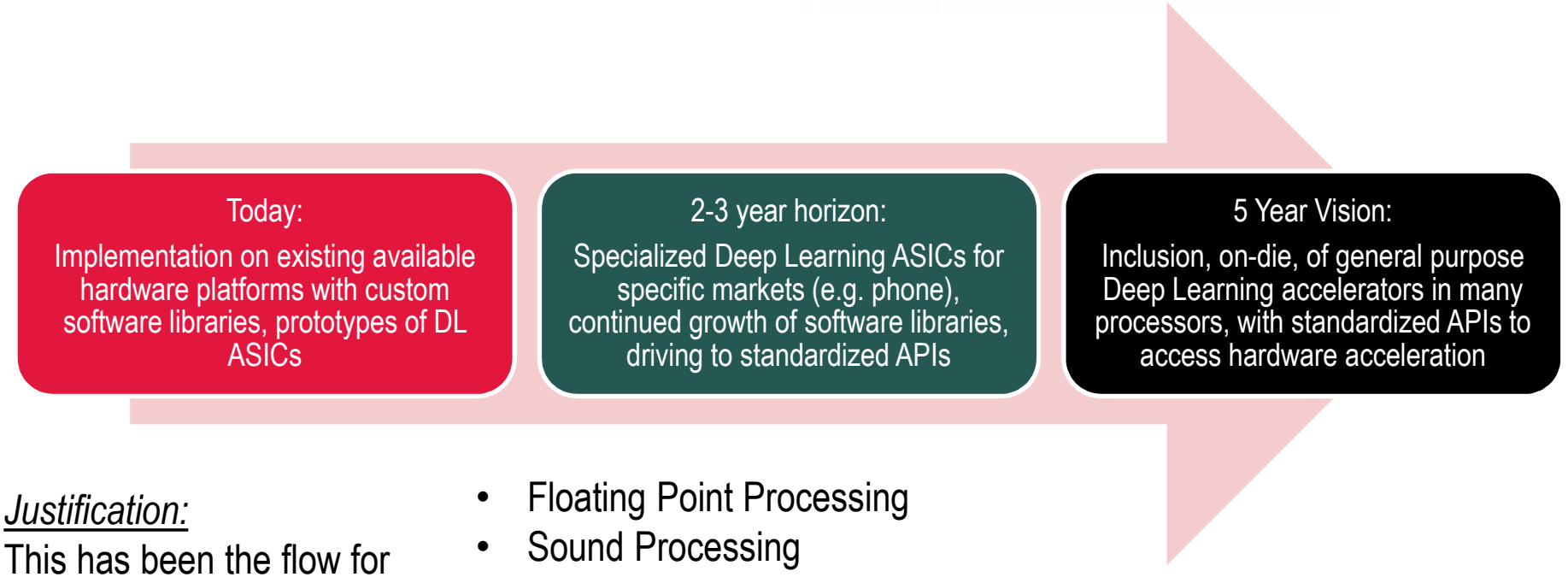
- Caffe — A deep learning framework specializing in image recognition.
- CNTK — open source deep-learning Computational Network Toolkit by Microsoft Research.
- ConvNetJS — A Javascript library for training deep learning models. It contains online demos.
- Deeplearning4j — An open-source deep-learning library written for Java with LSTMs and convolutional networks. It provides parallelization with CPUs and GPUs.
- Gensim — A toolkit for natural language processing implemented in the Python programming language.
- Keras — deep learning framework capable of running on top of either TensorFlow or Theano.
- NVIDIA cuDNN — A GPU-accelerated library of primitives for deep neural networks.
- OpenNN — An open source C++ library which implements deep neural networks and provides parallelization with CPUs.
- TensorFlow — Google's open source machine learning library in C++ and Python with APIs for both. It provides parallelization with CPUs and GPUs.
- Theano — An open source machine learning library for Python.
- Torch — An open source software library for machine learning based on the Lua programming language.
- Apache SINGA — A General Distributed Deep Learning Platform.

Google is providing free online course via Udacity using TensorFlow
https://www.udacity.com/course/deep-learning--ud730

Synergies with Defense Market for COTS HW and SW



The Road Ahead

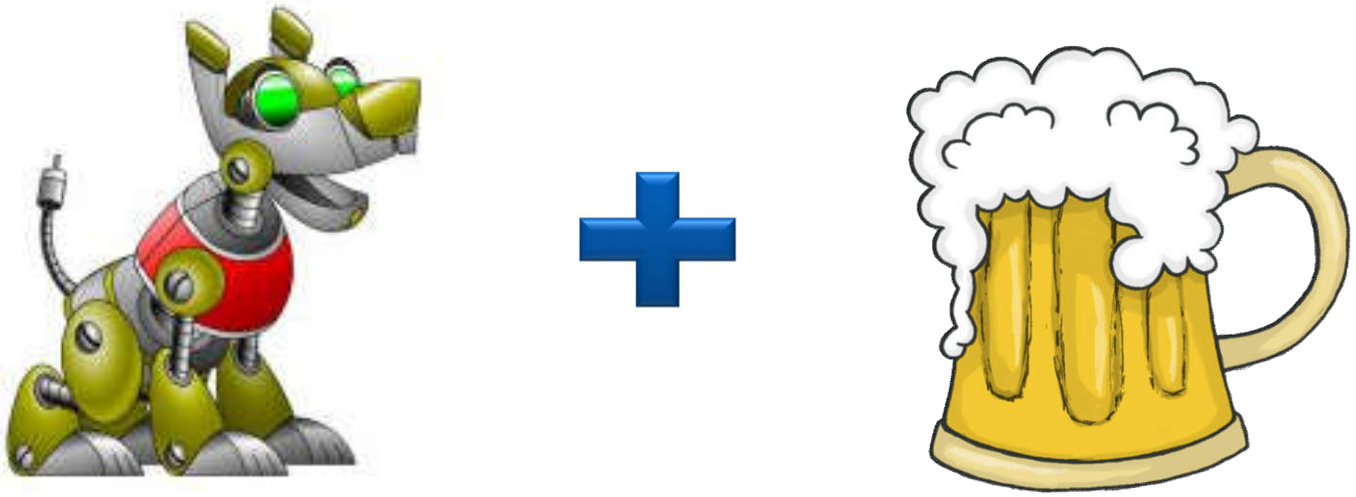


Justification:
This has been the flow for increased capabilities in computing systems for many years

- Floating Point Processing
- Sound Processing
- Graphics Acceleration
- Physics Engines
- Video Compression / Decompression
- Currently in this cycle for VR / AR

In conclusion

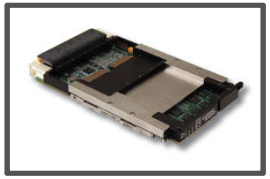
Deep Learning means….



…Beer-tasting Robot Dog fetches exactly the beer we want

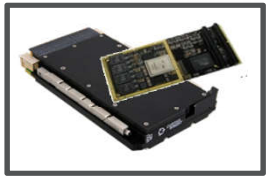
Application Ready COTS Systems

3U VPX™

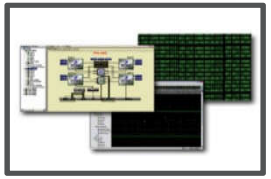

Single Board Computers

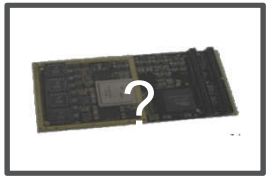

Storage


FPGA


Networking


GPGPUs


Software



3rd Party Hardware

Create several systems using the same basic building blocks!

Chad Augustine
Product Manager, Integrated Systems
chad.augustine@curtisswright.com
412.519.6969

Q&A

www.curtisswrightds.com