IEEE RIT Chapter Seminar

AFRL HERITAGE | 1917–2017

100 YEARS OF U.S. AIR FORCE SCIENCE & TECHNOLOGY

Integrity ★ Service ★ Excellence

Machine Learning for Object Recognition from High Volume Radio Frequency Data

May 5, 2017

Uttam K. Majumder, MBA, PhD
Air Force Research Laboratory
Information Directorate

AFRL PA Approval: 88ABW-2017-1438 ; 88ABW-2017-1439

Outline

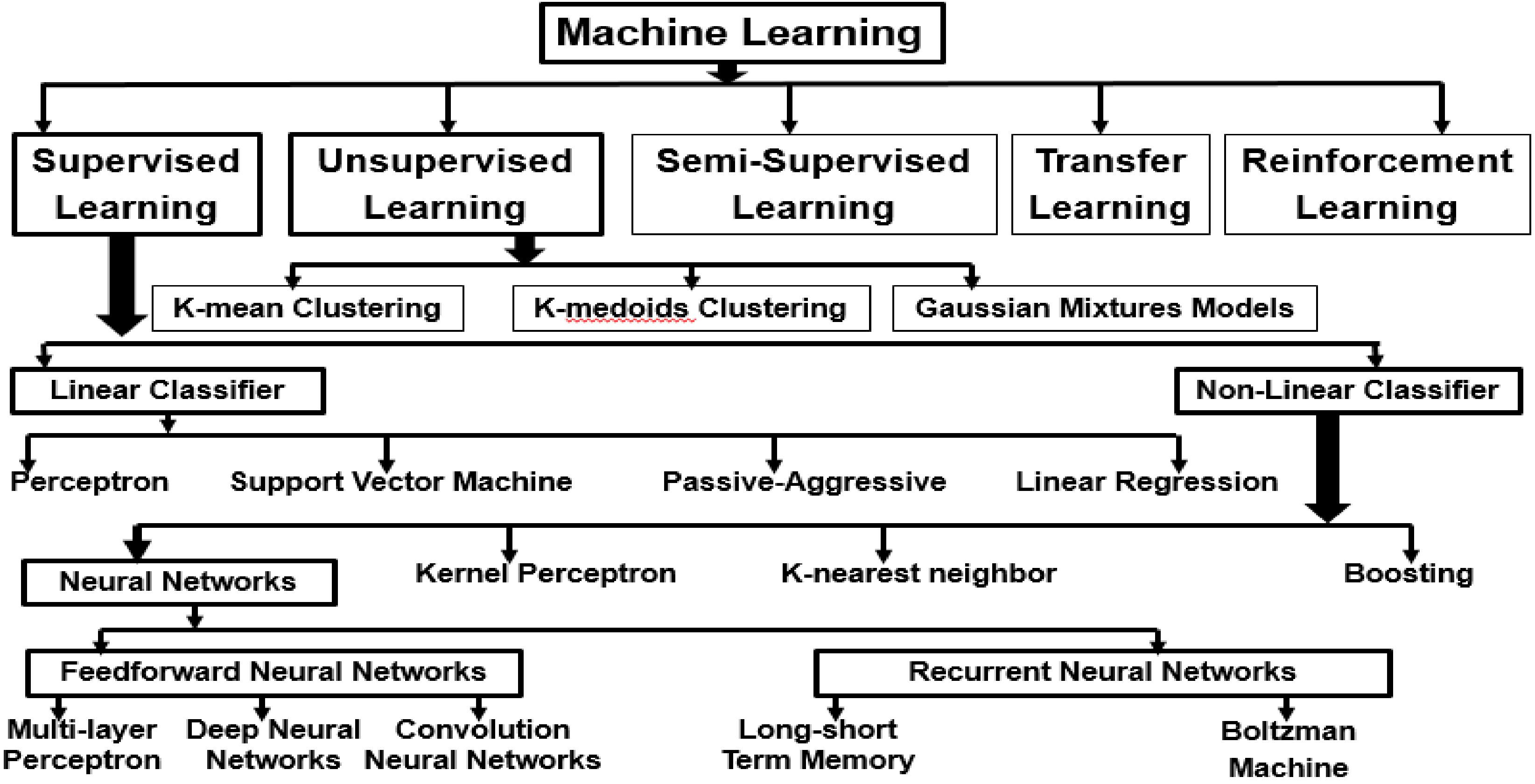
- ML Techniques
- Radio Frequency Data
- Big Data
- Research on Big Data
- High Performance Computing (HPC)
- GPU Enabled Target Classification from SAR Imagery
- Neurosynaptic Processor for Target Classification
- Summary

Introduction

- **Over the years, Machine Learning (ML) algorithms have been evolving**
  - Improved accuracy
  - Real-time execution
  - Solving more complex problems
- **Top ML Tools/Software**
  - TensorFlow (Google)
  - Caffe (UC Berkley)
  - Theano
  - Torch (Facebook)

Next, I will present an overview of ML algorithms

ML Algorithms in Broad Categories



Supervised Learning

- **Many Practical ML uses Supervised Learning**
- **Input/Training Data are given with a "Level" or "Tag"**
- **Supervised Learning seeks to develop a very robust mapping function** $Y = f(X)$
  where X is the input data and Y is the output data
- **The goal is that when we have new input data X (e.g. images),** f **can predict Y (e.g. class labels) for that data**
- SL Generally requires large amounts of training data
- Two main types of supervised learning:
  - Classification – the output value is a class label
  - Regression – the output value is a real value

Supervised Learning

- In general, Supervised Learning is divided into two areas:

1. **Linear Classifier**
   - Perceptron
   - Support Vector Machine
   - Passive Aggressive
   - Linear Regression
2. **Non-Linear Classifier**
   - Neural Networks
   - Kernel Perceptron
   - Boosting
   - K-nearest Neighbor

Supervised Learning

Linear Classifier: Perceptron

- Binary Classifier, Invented 1957, at Cornell Aeronautical lab by Frank Rosenblatt

- Also, known as single-layer perceptron (there is multi-layer perceptron). Perceptron is the simplest form of feedforward neural network
- The perceptron algorithm attempts to divide the input space into two halves with a single line
- Perceptron does not consider the entire dataset at the same time, rather, it looks at one example at a time, processes it, then moves onto the next example ("online")

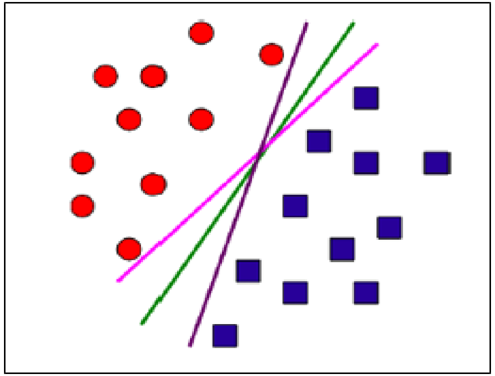


Fig. 1: Perceptron Algorithm [1]

Supervised Learning

- **Linear Classifier: Support Vector Machine (SVM)**
  - Invented 1963, by Vladimir Vapnik and Alexey Chervonenkis
- Can be used for Image Classification and Character Recognition
- Global Optimization for the loss function
- On-line version: Pegasos algorithm
- Support Vector Machine works similarly to Perceptron in that it attempts to divide the input space with a linear separator.

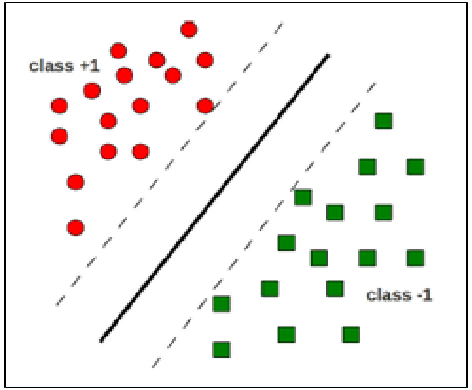


Fig. 2: SVM Algorithm [1]

- The main difference is that the SVM works to maximize the margin between the two classes in the input space and therefore may still update even when it guesses correctly.

Supervised Learning

➢Linear Classifier: Passive-Aggressive

- Invented 1963, by Vladimir Vapnik and Alexey Chervonenkis

▪ This is an extension of the SVM algorithm

▪ Uses point-by-point optimization for the loss function

Supervised Learning

➢ Linear Classifier: Linear Regression

- Ordinary Least Square is most popular algorithm
- Works to minimize the sum of the squared residuals. Given a regression line through the data, calculate the distance from each data point to the regression line, square that distance, and sum the distances for each point (i.e. the sum of the squared residuals).

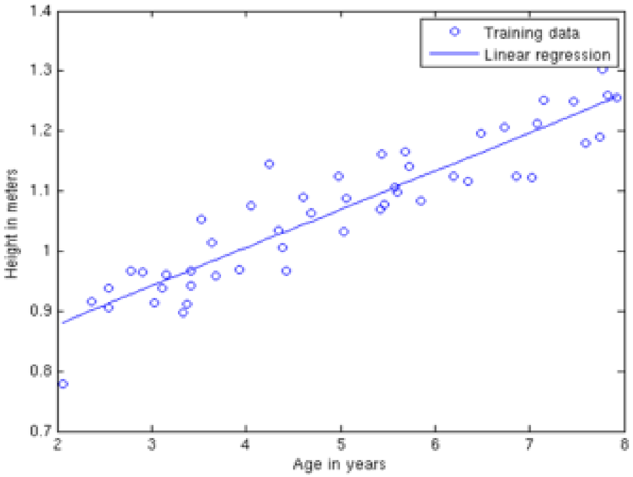


Fig. 3: Linear Regression [17]

- Both inputs and outputs are numeric

Supervised Learning

➢Non-Linear Classifier: Neural Networks

- Invented 1943, by Warren McCulloch and Walter Pitts

- **Feedforward Neural Networks (FNN)**
  - Multi-Layer Perceptron
  - Deep Neural Networks
  - Convolution Neural Networks
- **Recurrent Neural Network (RNN)**
  - LSTM
  - Boltzman Machine
  - Reservoir Computing
  - Liquid State Machine

Supervised Learning

➢Non-Linear Classifier: Neural Networks

- Neural Networks are designed to recognize numerical patterns in the input data, and ultimately learn the mapping function between the input and output data.
- A NN is a corrective feedback loop, rewarding weights that support correct guesses and punishing weights that lead to error
- Each hidden layer attempts to learn a distinctive set of features based on the previous layer's output. In general, the deeper the network, the m… earn

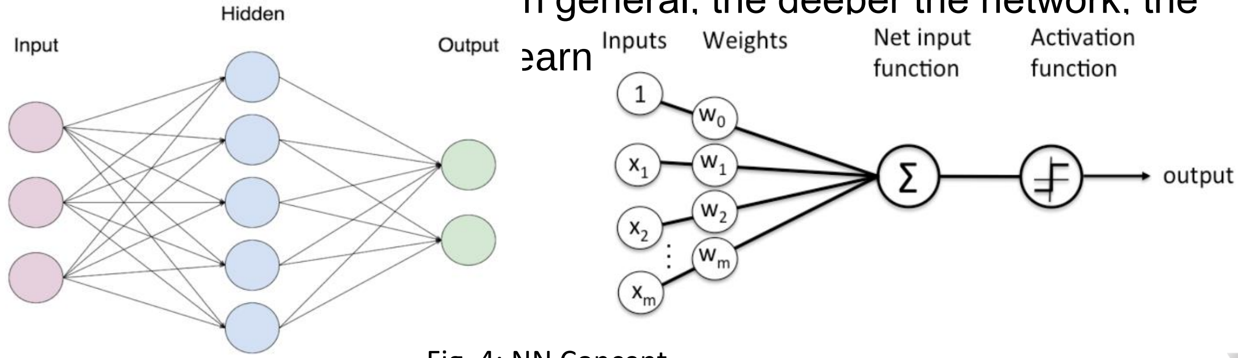


Fig. 4: NN Concept

Supervised Learning

- **Non-Linear Classifier:** Multi-layer Perceptron (MLP)

  - MLP is the simplest form of feedforward NN based upon Linear Perceptron.

  - Generally, MLP consists of three or more layers of non-linearly activating nodes

  - The network learns from backpropagation process

Supervised Learning

➢Non-Linear Classifier: Feed Forward Neural Networks

- **Deep Neural Networks (DNN)**
  - A network is considered “deep” if it has several hidden layers
- **Important DNNs**
  - ResNet
  - Wide ResNet
  - VGG (Visual Geometry Group)
  - AlexNet
  - GoogleNet
  - Generative Adversarial Networks

Many of these above DNNs use convolution filters for feature extraction; hence these could be referred to CNN as well

Supervised Learning

➢ Non-Linear Classifier: Convolutional NN (CNN)

- Use a combination of filters that each search the input space for very specific features
- Each filter creates a feature map that gets fed into the next layer to be filtered again
- CNN's create very highly dimensional representations of inputs depending on how many filters are used and as a result must be resized with pooling/down-sampling layers.

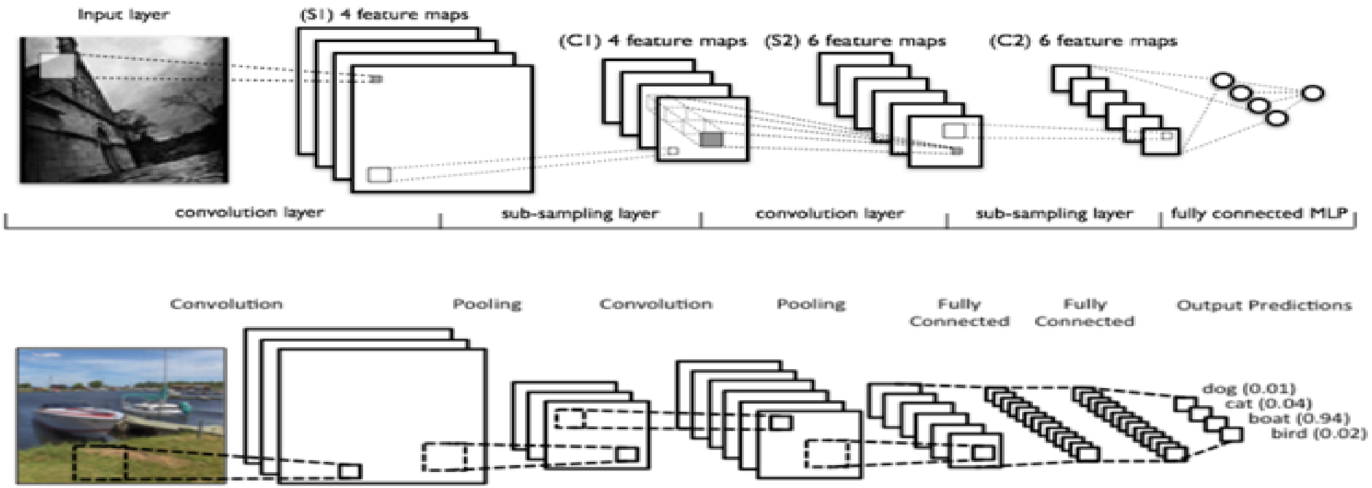


Fig. 5: CNN concept

Supervised Learning

- **Recurrent Neural Networks (RNN):** RNN is used for learning sequences. Applications such as speech recognition, handwriting recognition has been implemented using RNN.
- RNNs have feedback loops in learning process
- **Some Important RNNs:**
  - Long Short-term Memory (LSTM)
  - Boltzman Machine
  - Reservoir Computing
  - Liquid State Machine

Supervised Learning

➢ Non-Linear Classifier: Kernel Perceptron

- Works similarly to perceptron in that it attempts to divide the input space
- The key idea is that the algorithm uses a Kernel function to transform the input space. The transformed space does not have to have the same dimensionality as the input space, and in many cases it has a higher dimensionality. The Kernel Perceptron then attempts to construct a n-1 dimensional hyperplane in the n-dimensional transformed space which correlates to a non-linear separation in the original input space.
- The effective algorithm and update rules are the same as the classic perceptron algorithm except for the transform itself.

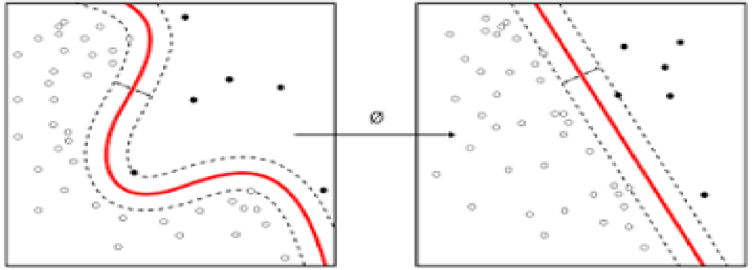


Fig. 6: Kernel Perceptron Concept [24]

Supervised Learning

➢ Non-Linear Classifier: K-nearest Neighbor (KNN)

- Works on entire data set at once; KNN does NOT learn any model, rather the model is the training set itself
- For each new instance, the algorithm searches through the entire training set, calculating the difference between the new instance and each training model.
- For classification, the output is the class with the K-most similar neighbors.
- For regression, the output value is based on the mean or median of the K-most similar instances.

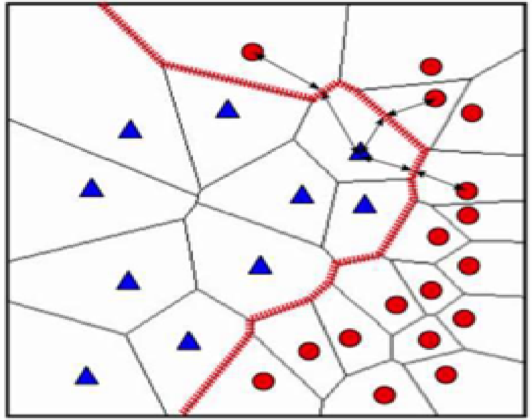


Fig. 7: K-nearest neighbor

Supervised Learning

➢Non-Linear Classifier: Boosting

- Use several different classifiers that are each good at identifying based on certain features
- Have each classifier vote on what the final result is
- H(x) = (w1*h1(x) + w2*h2(x) + … + wn*hn(x))
  - Where H(x) is the overall classifier, the h functions are the individual classifiers, and the w’s are the weights of each individual classifier
- “Wisdom of a weighted crowd of experts” – Prof. Patrick Winston, MIT
- The weights for each of the classifiers are updated based on the error they contribute
- The update rule for the weights is remarkably simple as it is a scaling

Unsupervised Learning

- In Unsupervised Learning, the algorithms discover “hidden pattern” from the input data.
  - During training, we only have the input data and no corresponding output variables (labels), as opposed to supervised learning where we have both
  - The goal of this type of learning is to model the underlying structure or distribution of the data
  - “Algorithms are left to their own devices to discover interesting structure in the data”
  - Clustering – discover inherent groupings in the data
  - Association – discover rules that describe large portions of the data
- **Unsupervised Learning Algorithms:**
  - Clustering
  - K-medoids
  - Gaussian Mixtures Models
  - Autoencoder

Unsupervised Learning

➢ Clustering

- Idea: A data set with N objects can be grouped into any number of clusters between 1 and N. The goal of the algorithm is to identify regions in which the data points are concentrated, and group the points accordingly.
- No guarantee that a globally optimal solution will be reached, as it depends on the initial seeding of the cluster centers (centroids)
  - K-means
    - Aims to partition N objects into K clusters, in which each observation belongs to the cluster with the nearest mean.
    - Goal is to minimize the average squared Euclidean distance of objects from their centroids
    - The measure of how well the centroids represent the members of their clusters is the residual sum of squares (RSS), which is the sum of the squared distances from each observation to its centroid.
  - K-medoids
    - This algorithm is very similar to K-means.
    - Rather than calculating centroids as the cluster centers, it uses medoids which are the most centrally located objects in the cluster, not just points in space.
    - Less sensitive to outliers

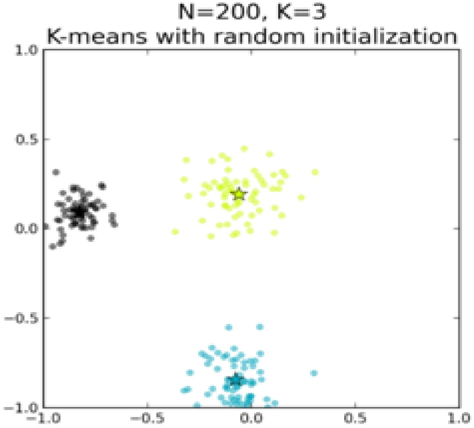


Fig. 8: Clustering

Unsupervised Learning

➢Gaussian Mixture Models (GMM)

- GMM is used for data clustering
- GMM Parameters are estimated from training data by using expectation-maximizatiom (EM) or Maximum-A-Posteriori (MAP) algorithms

Semi-Supervised Learning

- In Semi-supervised Learning, some input data are leveled. This information combined with unsupervised learning such as “clustering” can be used for classifying data
  - Some of the input data are labeled and some are not
  - Use a mixture of supervised and unsupervised learning techniques
  - Note: Most of the data in the world is unlabeled, there is only a small fraction that is. Unsupervised and semi-supervised learning techniques can be applied to much larger, unlabeled datasets, making them very appealing to some researchers

AFRL PA Approval: 88ABW-2017-1438 ; 88ABW-2017-1439

Transfer Learning

- DNN Requires a large training dataset to produce accurate learning representation from test data
  - When a new dataset are comparatively small, we may use already learned data that are similar to a new dataset
  - Transfer Learning offers a way to leverage existing dataset to perform well on new dataset
  - The idea is to use information that you already know and that you have already worked hard to learn and apply that knowledge to a new, similar dataset

Reinforcement Learning

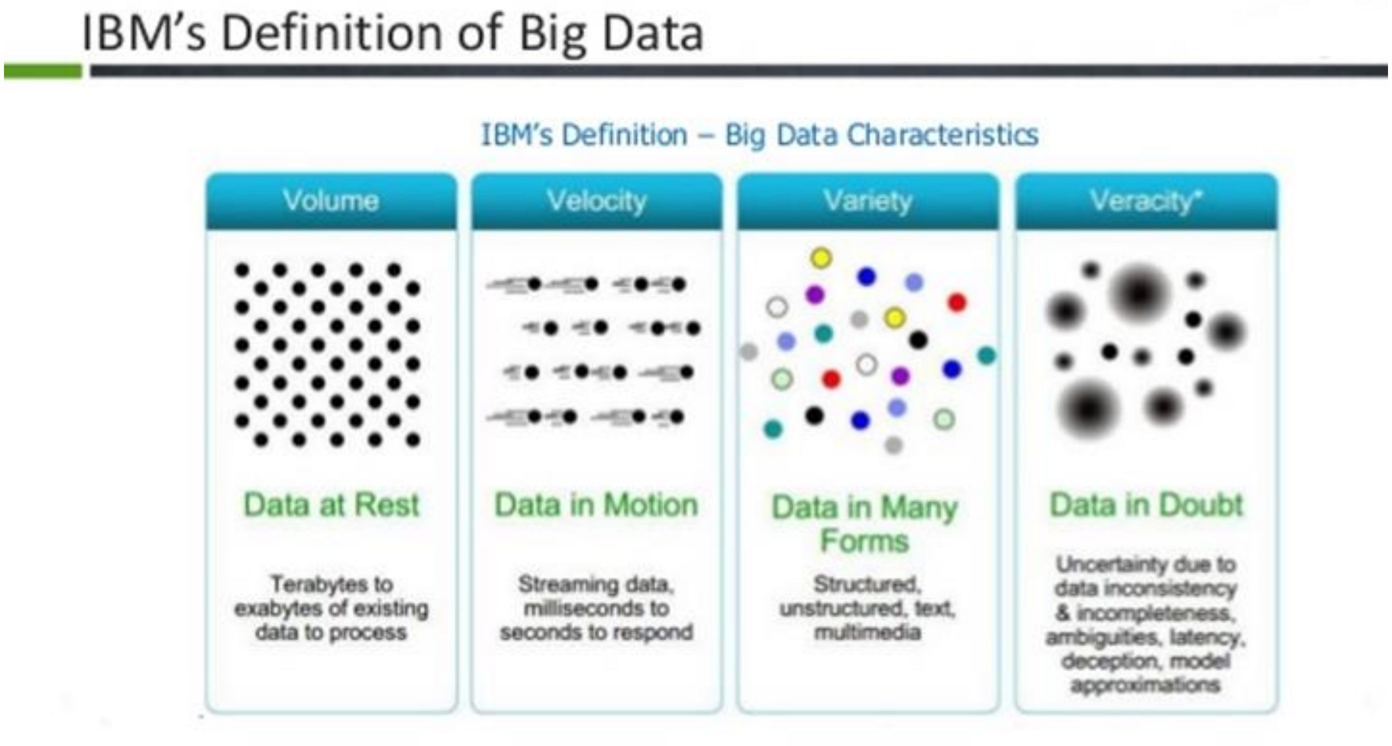
- Learning by interacting with the environment
- The learning agent learns from the consequences of its actions, rather than from being explicitly taught, and it selects it actions based on its past experiences (exploitation) and also by new choices (exploration).
- The agent receives a numerical reward for each of its actions which encodes a level of success. The agent then learns to perform actions that maximize the reward.

Applications of ML for Big Data Analytics

- **Image Classification**
  - Security/Surveillance
- Cyber Security
  - Intruder detection/classification
- **Natural Language Processing**
  - User/Customer Review Analysis
  - Image Content Analysis
  - Character/text recognition
- **Autonomous Systems**
  - Self-driving Cars
- **Sentiment Analysis**
  - Twitter Feed /News Analysis; Recommender systems
- **Biomedical Applications**
  - Diagnosis of malignant tumor from MRI/X-Ray by correlating various patients data (age, race, family history etc.)

Radio Frequency Data

Big Data



https://www.slideshare.net/EdurekaIN/introduction-to-big-data-hadoop-i

Research On Big Data

- Operational deployment considerations, computation efficiency (SWaP-C)
  - ***The need for HPC for real-time computing***
- *Model fidelity* complimented with data collections for synthetic-measured data analysis
- *Transfer Learning* over operating spaces (range, resolution, target settings)
- *Big data* (volume, velocity, veracity, variety) collaboration policies – what data are accessible for analytics
- Robust evaluation: Validation, Verification, for reproducible results

The Need for Real-time Computing

→ In 90's, Machine Learning such as Neural Networks was less popular due to various ***Tech Barriers and Needs***

- ► Computational Resources were Scarce and Expensive
- ► Limited Sensors or Digitized Business Data to be Analyzed

✓ Today, computational resources are not as expensive as in the past; however, abundant of Sensors and Business

data needs to be analyzed in ***Real-time***

✓ HPC Enables ***ML algorithm based*** decision making in ***real-time or near real-time***

The Advent of HPC

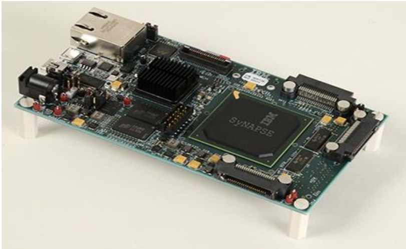
- Since Late 90’s, Computing Technology Has Advanced in an Astounding Pace (The Moore’s Law)
- We are Living in the Age of HPC
    - Faster memory, CPU, I/O communication, and storage as well as compact/smaller size
    - Multi-core Computers
    - Graphics Processing Units
    - Energy-efficient/low-power computing devices
- More to come
    - Memristor Devices
    - Specialized Chip/cores for Sparse Graph Processing

Recent HPC Hardware Used for ML Algorithms

CPU
Few, fast cores (1 - 16)
Good at sequential processing

GPU
Many, slower cores (thousands)
Originally for graphics
Good at parallel computation

IBM's TrueNorth



FPGA

GPU Enabled Target Classification Measured SAR Data

- Training, validation, and testing data come from the **MSTAR*** program sponsored by DARPA and the AFRL in the 1990s
- **10 target classes** with images taken at various angles
  - **15 Degree Elevation Angle** dataset for **training, 17 Degree** dataset for **testing**
  - Roughly **250 images** per target class, per angle
  - Generally considered an incredibly small dataset for a deep learning application
- **Using a single GPU at AFRL/RI HPC**

* MSTAR: Moving and Stationary Target Acquisition and Recognition

AFRL PA Approval: 88ABW-2017-1438 ; 88ABW-2017-1439

Target types

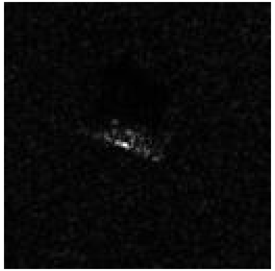
Table 1. The number of images of each object at different depression angles.

Targets	BMP2	BTR70	T72	BTR60	2S1	BRDM2	D7	T62	ZIL131	ZSU234
17°	233	233	232	256	299	298	299	299	299	299
15°	587	196	582	195	274	274	274	273	274	274

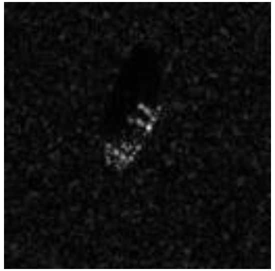
BMP2-C21 BTR70 T72-132 BRDM2 BTR60

T62 ZSU234 2S1 D7 ZIL131

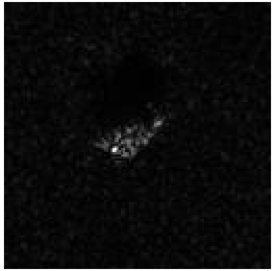
SAR Imagery



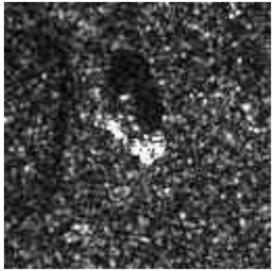
BMP2



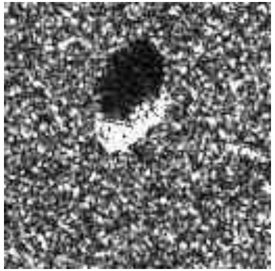
BTR70



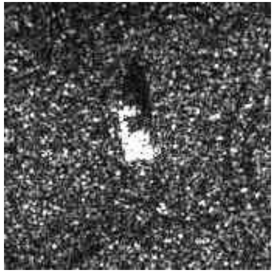
T72



BRDM2

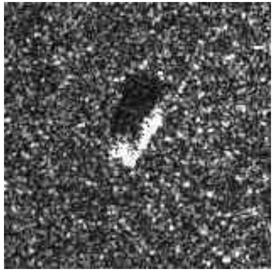


BTR60

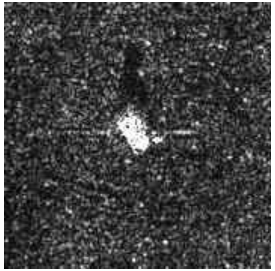


T62

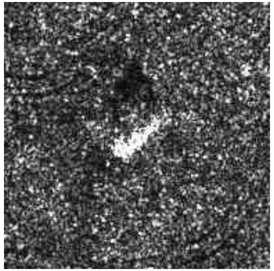
ZSU234



2S1



D7



ZIL131

Software Tools

- **Python –** Data augmentation methods
- **Caffe –** Deep learning framework employed via DIGITS and command line

Caffe

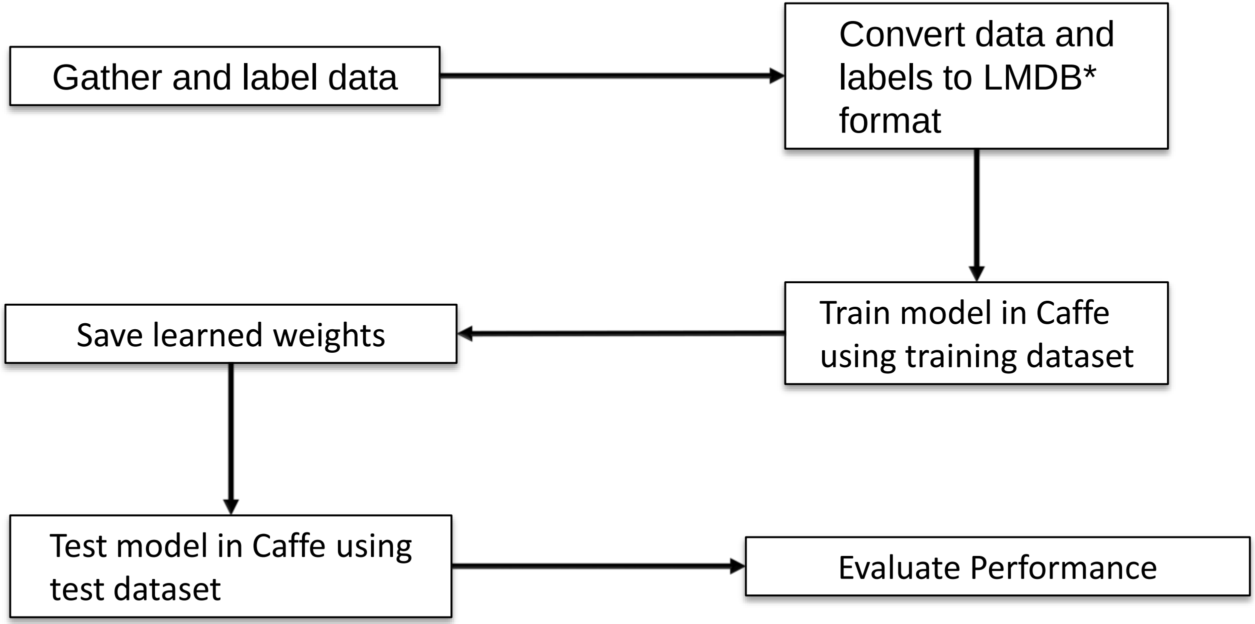
- Deep Learning framework developed by the Berkeley Vision and Learning Center (BVLC)

- Written in highly optimized C++/CUDA code

- Easily define network architectures

- Modify DL models as needed for an application

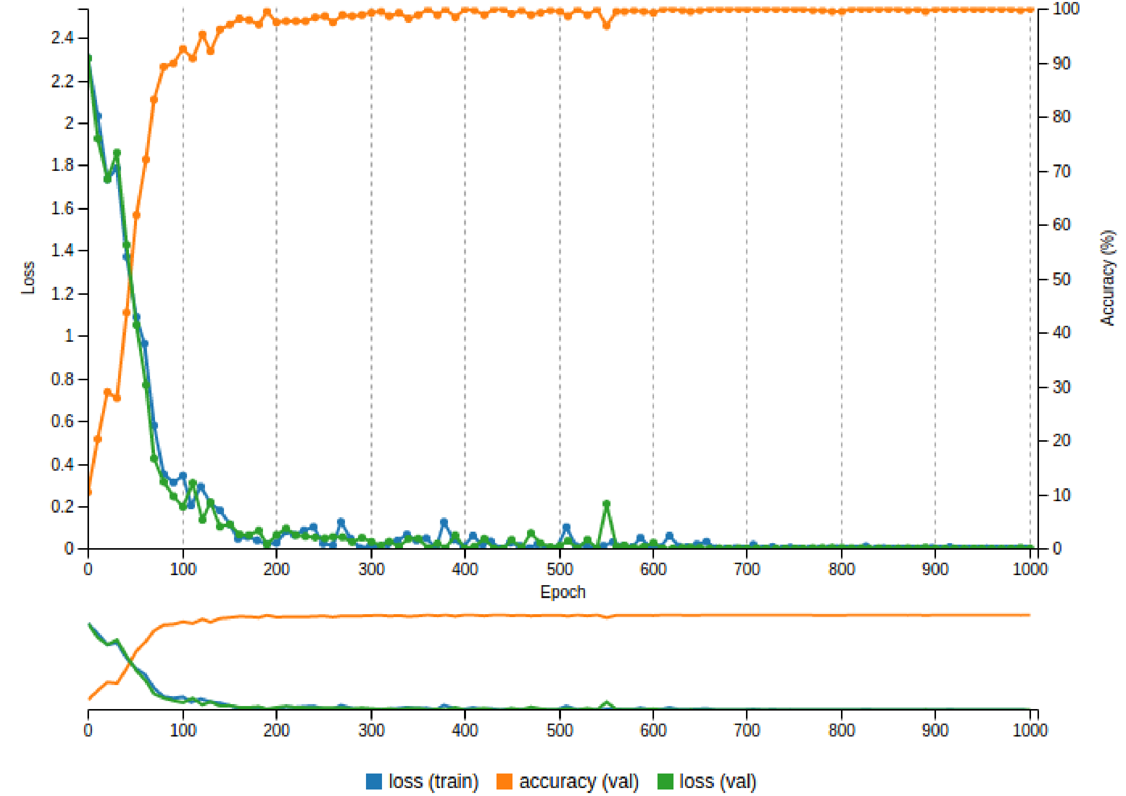
Caffe ML Algorithm Flow



* LMDB: Lightning Memory-Mapped Database Manager

Clean training run

Neural Net reaches over **99% accuracy** on validation set



AFRL PA Approval: 88ABW-2017-1438 ; 88ABW-2017-1439

AFRL HERITAGE | 1917-2017

Classification results on Measured Data

~99% accuracy on 10-target classification using Caffe

State-of-the-art results

```
uttam@uttam-Lenovo-Edge-2-1580:~/caffe$ ./build/tools/caffe test -model train_val.prototxt
-weights snapshot_iter_39000.caffemodel -gpu=0 -iterations=500
```

```
I0815 12:03:19.780280  3181 caffe.cpp:300] Batch 996, accuracy = 1
I0815 12:03:19.780316  3181 caffe.cpp:300] Batch 996, loss = 0.0115048
I0815 12:03:19.810564  3181 caffe.cpp:300] Batch 997, accuracy = 1
I0815 12:03:19.810609  3181 caffe.cpp:300] Batch 997, loss = 0.0182748
I0815 12:03:19.840967  3181 caffe.cpp:300] Batch 998, accuracy = 0.953125
I0815 12:03:19.841004  3181 caffe.cpp:300] Batch 998, loss = 0.154708
I0815 12:03:19.871201  3181 caffe.cpp:300] Batch 999, accuracy = 1
I0815 12:03:19.871237  3181 caffe.cpp:300] Batch 999, loss = 0.000220068
I0815 12:03:19.871246  3181 caffe.cpp:305] Loss: 0.115863
I0815 12:03:19.871261  3181 caffe.cpp:317] accuracy = 0.989016
I0815 12:03:19.871305  3181 caffe.cpp:317] loss = 0.115863 (* 1 = 0.115863 loss)
```

Key network parameters

Learning rate 0.001
Batch size 64
1000 training epochs

5 Convolution layers
3 InnerProduct (FC) layers
2x2 stride 1 max pool filters

Dropout regularization

Target Classification Using DNN on Synthetic SAR Data

- Training, validation, and testing data used from Synthetic Radar Data
- **30 target classes** with images taken at various elevation angles and a single azimuth angle
- Instead of Backprojection Image formation, we used Range-Doppler Map of the Targets
- **We found about 99% accuracy on Target classification**

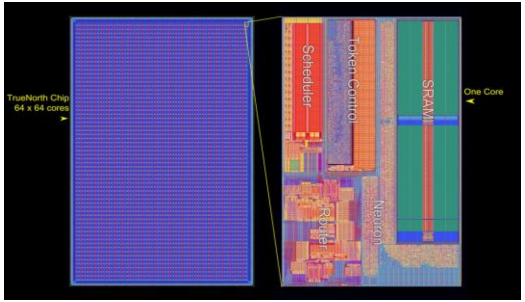
Target Classification Using DNN on Synthetic and Measured SAR Data

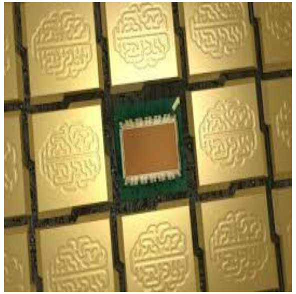
- The objective of this research is to evaluate performance of target classification using Synthetic vs. Measured SAR data (or vice versa) and identifying the "Gap/Tech Challenges" to generate High Fidelity Synthetic SAR data
- We implemented Training on measured SAR data for three targets and Tested on Synthetic SAR data (of the same targets)
- **We found very low accuracy on Target classification**
- This is due to the fact that quality (i.e. NIIRS) of synthetic data must be very close to measured data
  - This will require huge HPC resources and expertise in Computational Electro-magnetic

➢ **TRANSFER LEARNING**

IBM's TrueNorth (TN)

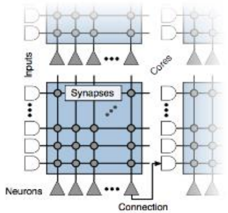
- TN is a neuromorphic CMOS chip inspired by human brain.
- TN is developed based on a parallel, event-driven, non-von Neumann kernel for neural networks that is efficient with respect to computation, memory and communication.
- TrueNorth chip consists of 4096 cores, tiles as a 64x64 array.
- Each chip consist of over 1 million neurons and over 256 million synapses.

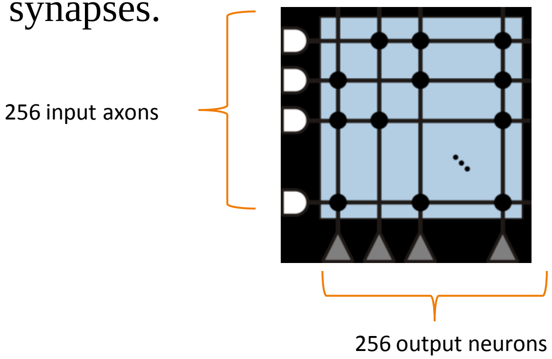




TrueNorth Contd…

- Each TN neuro-synaptic core is a fully connected neural network with 256 input axons and 256 output neurons, connected by 256x256 synapses.



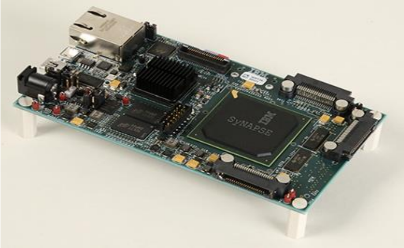


- Each chip consumes approximately 70mW power while running a typical vision application.
- Input spike activates an axon, which drives all connected neurons. Neurons integrate incoming spikes, weighted by synaptic strength.

TN Hardware

TrueNorth is available in three different hardware configuration

- **NS1e platform**: Main processing element of the NS1e is a single TN chip and it's coupled with a Xilinx Zynq (FPGA) and two ARM cores connected to 1GB DDR3 SDRAM. The average power consumption is between 2W to 3W with TN consuming only ~3% of the total power.



- **NS1e-16 platform**: It is constructed using sixteen NS1e boards, with aggregate capacity of 16 million neurons and 4 billion synapses, interconnected via a 1Gig-Ethernet packer switched network.
- **NS16e platform**: This architecture integrates 16 TN chip into a scale-up solution. It is capable of executing neural networks 16 times larger than the NS1e.

MSTAR Classification Results

- Core count : 3736 Image size: 44x 44
- Accuracy : 96.66%

Lyr	Layer Size Row x Col x Ftr	(Grp)	Patch Str	Patch Row Col Ffr	TN Cores Base+Splt	Patch-in-Image Str	Patch-in-Image Row x Col
I	44 x 44 x 1						
P1	44 x 44 x 12	(1)	1	[3 x 3 x 1]	0 +0	1	[3 x 3]
C2	22 x 22 x 128	(4)	2	[4 x 4 x 3]	484 +726	2	[6 x 6]
C3	22 x 22 x 128	(4)	1	[1 x 1 x 32]	484 +0	2	[6 x 6]
C4	11 x 11 x 128	(16)	2	[2 x 2 x 8]	576 +0	4	[8 x 8]
C5	11 x 11 x 256	(32)	1	[3 x 3 x 4]	384 +0	4	[16 x 16]
C6	11 x 11 x 256	(8)	1	[1 x 1 x 32]	288 +0	4	[16 x 16]
C7	11 x 11 x 256	(8)	1	[1 x 1 x 32]	288 +0	4	[16 x 16]
C8	5 x 5 x 256	(32)	2	[2 x 2 x 8]	288 +0	8	[20 x 20]
C9	5 x 5 x 512	(64)	1	[3 x 3 x 4]	256 +0	8	[36 x 36]
C10	5 x 5 x 512	(16)	1	[1 x 1 x 32]	144 +0	8	[36 x 36]
C11	2 x 2 x 512	(32)	2	[2 x 2 x 16]	64 +0	16	[44 x 44]
C12	2 x 2 x 1024	(16)	1	[1 x 1 x 32]	32 +0	16	[44 x 44]
C13	2 x 2 x 1024	(16)	1	[1 x 1 x 64]	32 +0	16	[44 x 44]
C14	2 x 2 x 2040	(8)	1	[1 x 1 x 128]	32 +0	16	[44 x 44]

Summary

- Big Data trends will require novel machine learning algorithms and computing systems development to address
  - Operational deployment considerations, computation efficiency (SWaP-C)
    - Real-time or Near Real-time Training
  - *Filling the Gap/mismatch between measured and synthetic data*
  - *Transfer Learning* over operating spaces (range, resolution, target settings)
  - collaboration policies – what data are accessible for analytics
  - Robust evaluation of the algorithms
  - Higher Detection and Classification but Reduced FAR

References

- Chen, S., Wang, H., Xu, F., & Ya-Qiu, J. (2016). Target classification using the deep convolutional networks for SAR images. *IEEE Transactions on Geoscience and Remote Sensing, 54*(8), 4806-4817.
- Karpathy, Andrej. "CS231n Convolutional Neural Networks for Visual Recognition." *CS231n Convolutional Neural Networks for Visual Recognition*. Stanford University, n.d. Web. 04 Jun. 2016.
- "Caffe.," Berkeley Vision and Learning Center. Web. 04 Jun. 2016. <http://caffe.berkeleyvision.org/>.
- Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." (2016): n. pag. Cornell University Library. 9 May 2016. Web. 22 July 2016.
- Song, H.; Ji, K.; Zhang, Y.; Xing, X.; Zou, H. Sparse Representation-Based SAR Image Target Classification on the 10-Class MSTAR Data Set. Appl. Sci. 2016, 6(1), 26
- Anderson, Peter. "A Practical Introduction to Deep Learning with Caffe."*Australian Centre for Robotic Vision*. The Australian National University, Dec. 2015. Web.

Questions