Microsoft

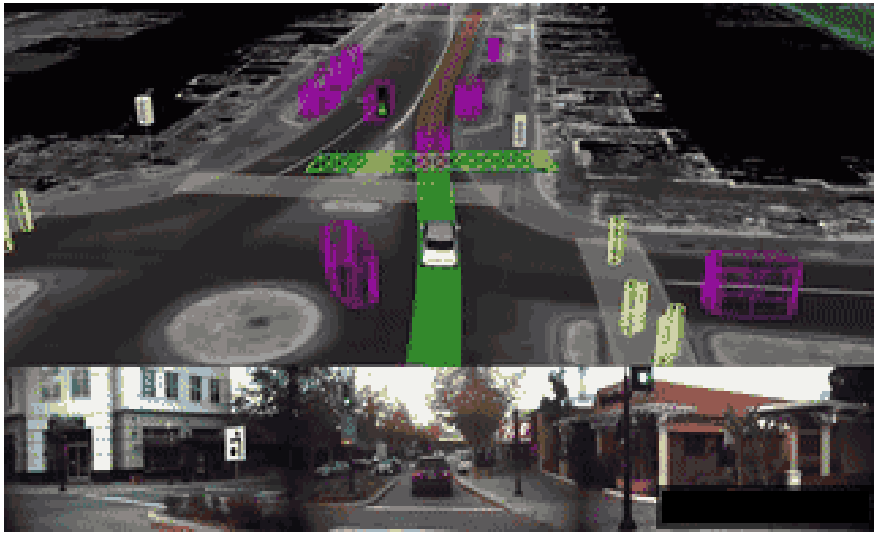
APAC Machine Learning & Data Science Community Summit

2017년 5월 20일(토)
상암동 누리꿈스퀘어 비즈니스타워 3층

Deep Learning with R

Poo Kuan Hoong
Malaysia R User Group

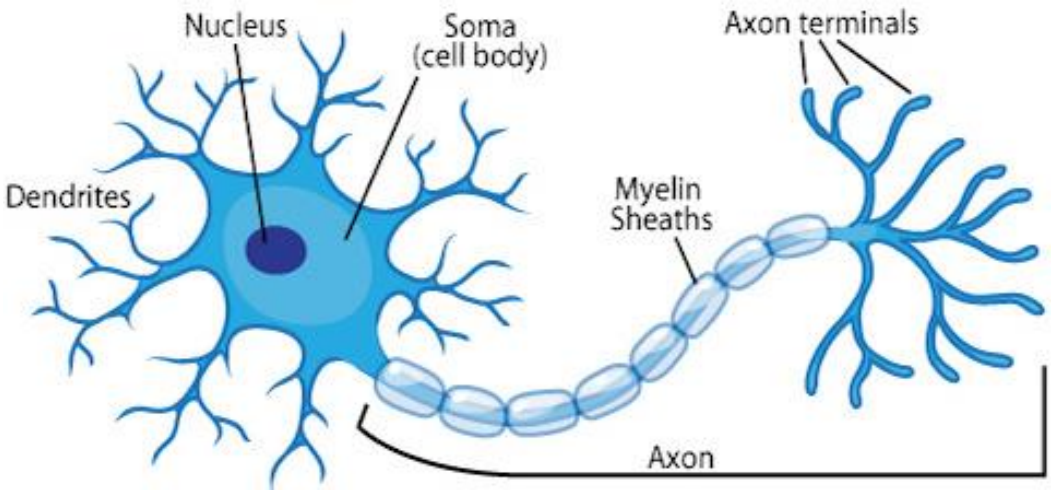
Deep Learning so far...

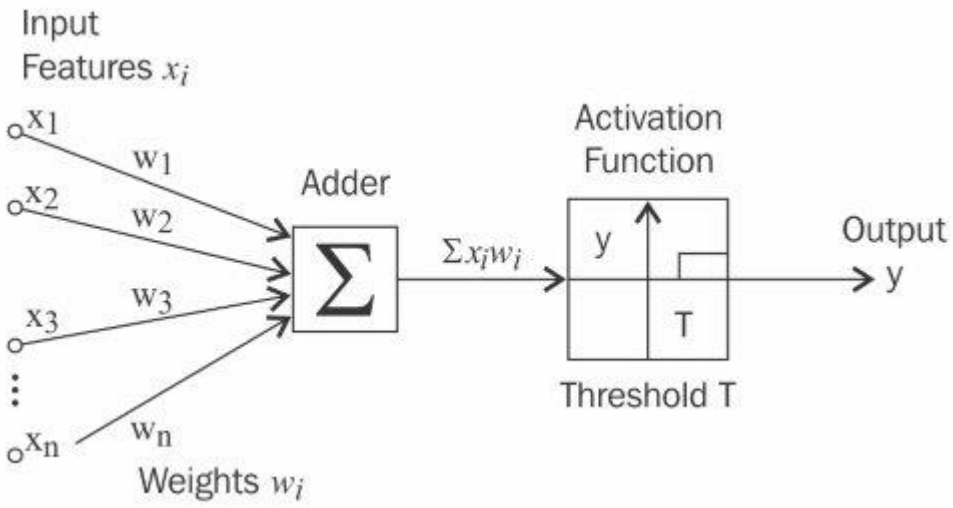


Introduction

- In the past 10 years, machine learning and Artificial Intelligence (AI) have shown tremendous progress
- The recent success can be attributed to:
  - Explosion of data
  - Cheap computing cost - CPUs and GPUs
  - Improvement of machine learning models
- Much of the current excitement concerns a subfield of it called “deep learning”.

Human Brain



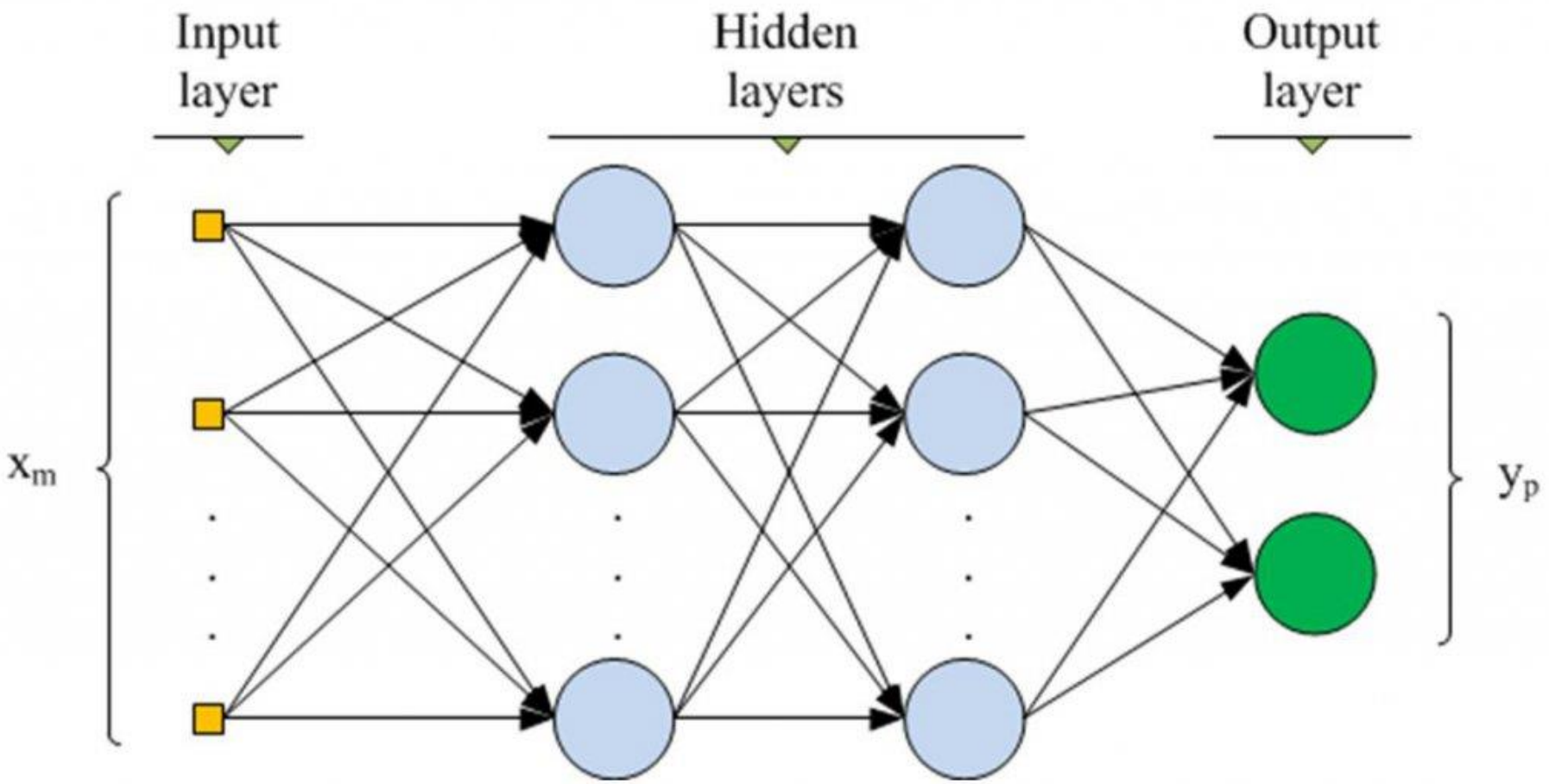


$$y = g\left(w_0 + \sum_{i=1}^{p} w_i x_i\right)$$

Neural Networks

- Deep Learning is primarily about neural networks, where a network is an interconnected web of nodes and edges.
- Neural nets were designed to perform complex tasks, such as the task of placing objects into categories based on a few attributes.
- Neural nets are highly structured networks, and have three kinds of layers - an input, an output, and so called hidden layers, which refer to any layers between the input and the output layers.
- Each node (also called a neuron) in the hidden and output layers has a classifier.

Neural Network Layers



Deep Learning

- Deep learning refers to artificial neural networks that are composed of many layers.
- It’s a growing trend in Machine Learning due to some favorable results in applications where the target function is very complex and the datasets are large.

Deep Learning: Strengths

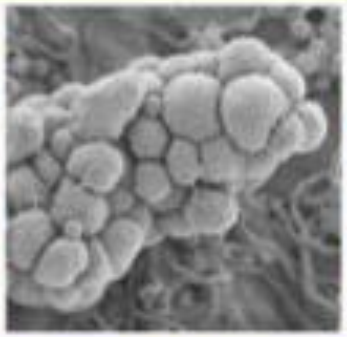
- Robust
  - No need to design the features ahead of time - features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- Generalizable
  - The same neural net approach can be used for many different applications and data types
- Scalable
  - Performance improves with more data, method is massively parallelizable

Deep Learning: Weaknesses

- Deep Learning **requires a large dataset**, hence long training period.
- In term of cost, Machine Learning methods like SVMs and other tree ensembles are very easily deployed even by relative machine learning novices and can usually get you reasonably good results.
- Deep learning methods **tend to learn everything**. It's better to encode prior knowledge about structure of images (or audio or text).
- The learned features are often **difficult to understand**. Many vision features are also not really human-understandable (e.g, concatenations/combinations of different features).
- Requires **a good understanding of how to model** multiple modalities with traditional tools.

Deep Learning: Applications

DEEP LEARNING EVERYWHERE


INTERNET & CLOUD	MEDICINE & BIOLOGY	MEDIA & ENTERTAINMENT	SECURITY & DEFENSE	AUTONOMOUS MACHINES
Image Classification Speech Recognition Language Translation Language Processing Sentiment Analysis Recommendation	Cancer Cell Detection Diabetic Grading Drug Discovery	Video Captioning Video Search Real Time Translation	Face Detection Video Surveillance Satellite Imagery	Pedestrian Detection Lane Tracking Recognize Traffic Sign

Deep Learning R Libraries

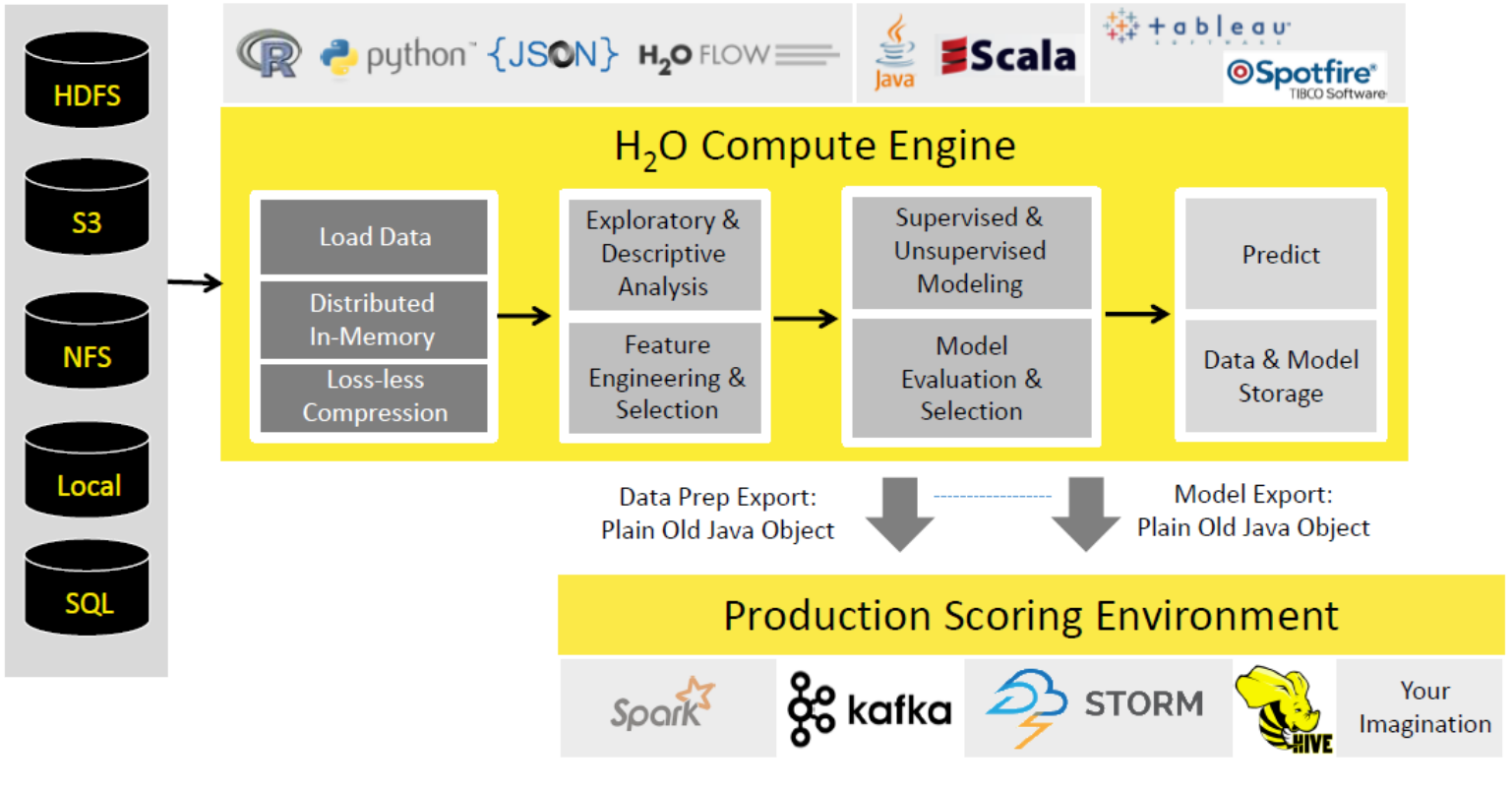
- MXNet: The R interface to the MXNet deep learning library.
- darch: An R package for deep architectures and restricted Boltzmann machines.
- deepnet: An R package implementing feed-forward neural networks, restricted Boltzmann machines, deep belief networ ks, and stacked autoencoders.
- Tensorflow for R: The tensorflow package provides access t o the complete TensorFlow API from within R.
- h2o: The R interface to the H2O deep-learning framework.
- Deepwater: GPU Enabled Deep Learning using h2o platform

H2O

- Founded in 2012
- Stanford & Purdue Math and Systems Engineer
- Headquarters: Mountain View, California, USA
- h2o is an open source, distributed, Java machine learning library
- Ease of Use via Web Interface
- R, Python, Scala, REST/JSON, Spark & Hadoop Interfaces
- Distributed Algorithms Scale to Big Data
- Package can be downloaded from http://www.h2o.ai/download/h2o/r

H2O.ai

High Level Architecture



Algorithms Overview

Supervised Learning

Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
Deep Neural Networks	• **Deep learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

Unsupervised Learning

Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data
Anomaly Detection	• **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

H2O + R

Installation

- Java 7 or later; R 3.1 and above; Linux, Mac, Windows
- The easiest way to install the h2o R package from CRAN
- Latest version: http://www.h2o.ai/download/h2o/r

```
> install.packages("h2o")
> library(h2o)
```

Design

All computations are performed in highly optimized Java code in the H2O cluster, initiated by REST calls from R

H2O info

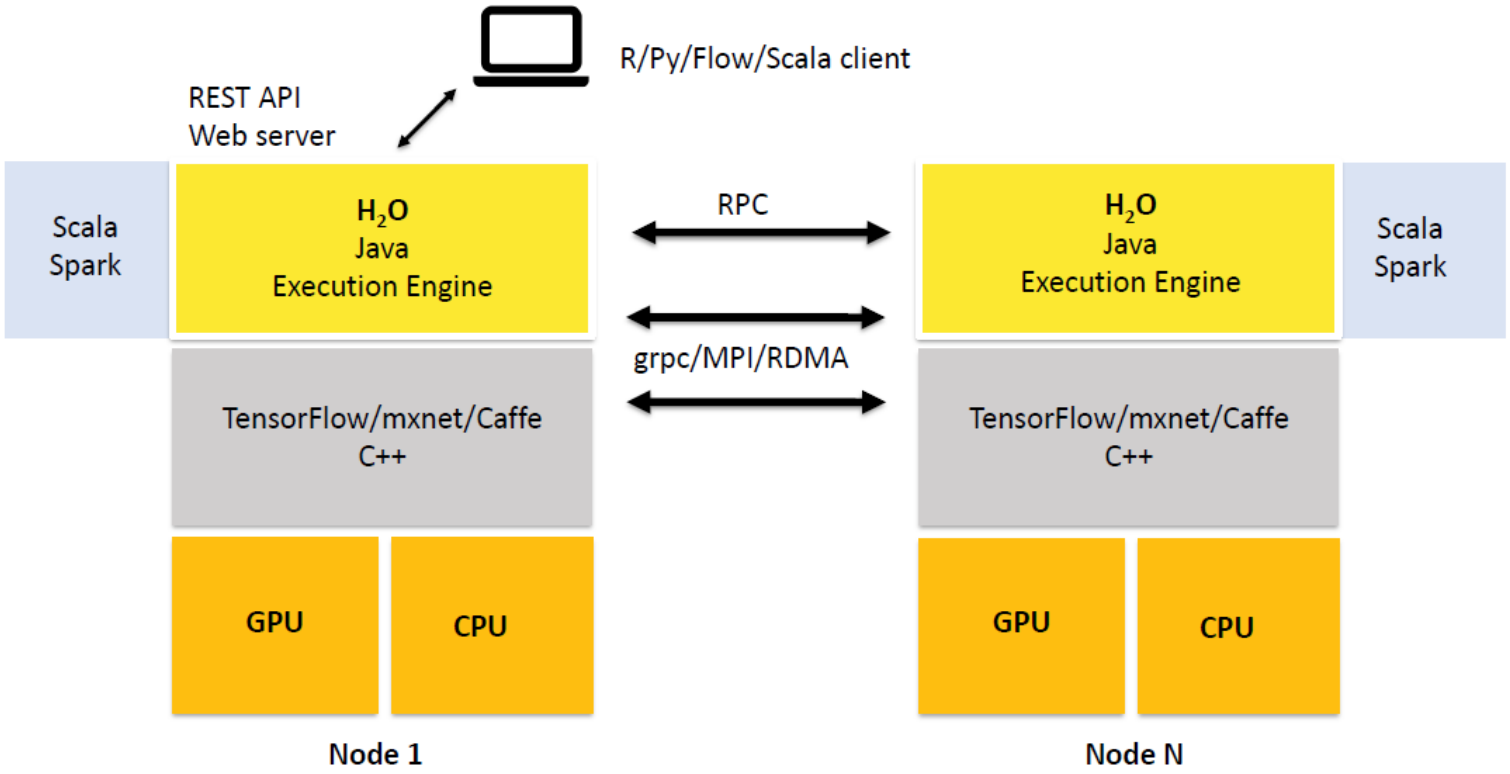
H2O Booklets

http://docs.h2o.ai/

H2O Deep Water

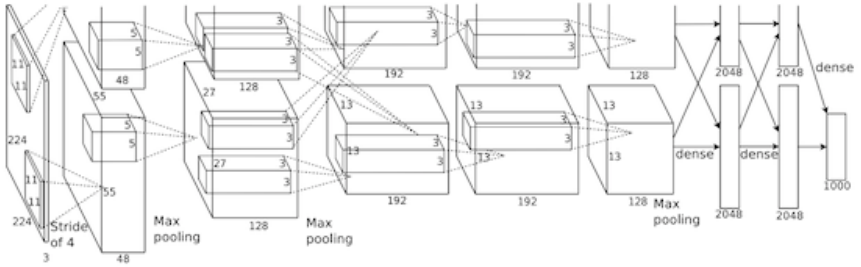
- Deep Water is the next generation distributed deep learning with H2O
- Deep Water is H2O for GPU that enabled deep learning on all data types integrating with TensorFlow, MXNEt and Caffe
- Inherits all H2O properties in scalability, ease of use and deployment

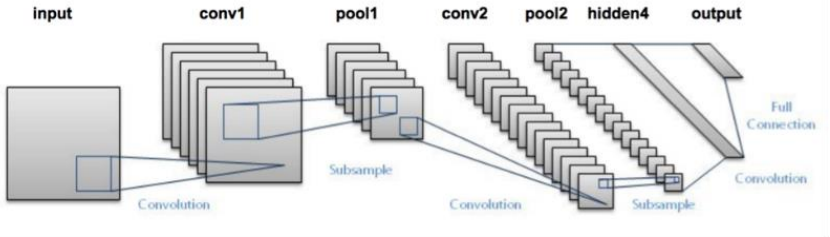
Deep Water Architecture

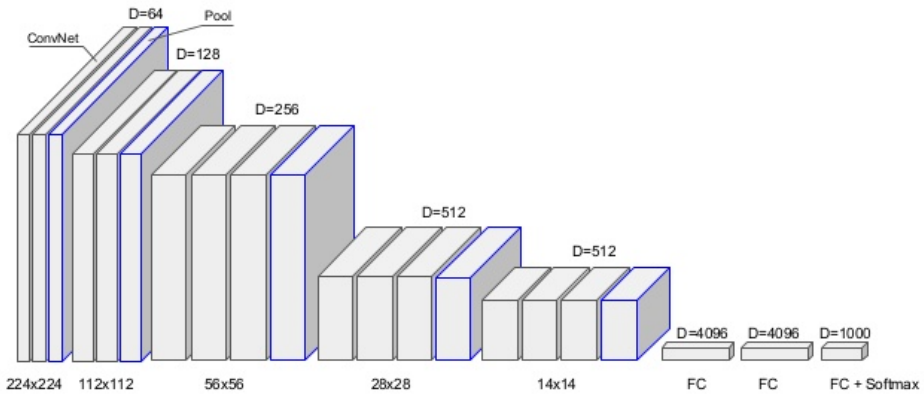


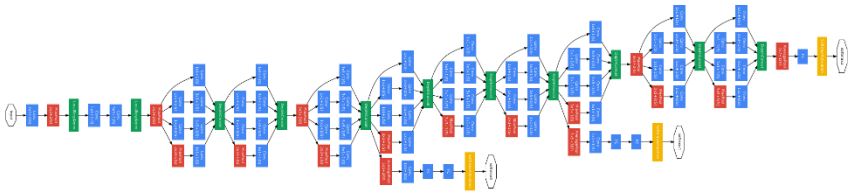
H2O: Deep Water

- Available networks in Deep Water
  - LeNet
  - AlexNet
  - VGGNet
  - Inception (GooLeNet)
  - ResNet (Deep Residual Learning)
  - Or build own customized network









H2O: Deep Water + R


```
model <- h2o.deepwater(x=path, y=response,
                       training_frame=df, epochs=50,
                       learning_rate=1e-3, network = "lenet")
model
```

Customize different network structures

MXNET

- **Founded by:** Uni. Washington & Carnegie Mellon Uni (~1.5 years old)
- **Supports most OS**: Runs on Amazon Linux, Ubuntu/Debian, OS X, and Windows OS
- **State of the art model support:** Flexible and efficient GPU computing and state-of-art deep learning i.e. CNN, LSTM to R.
- **Ultra Scalable:** Seamless tensor/matrix computation with multiple GPUs in R.
- **Ease of Use:** Construct and customize the state-of-art deep learning models in R, and apply them to tasks, such as image classification and data science challenges
- **Multi-language:** Supports the Python, R, Julia and Scala languages
- **Ecosystem**: Vibrant community from Academia and Industry

mxnet

Amazon + MXNET

APAC Machine Learning & Data Science Community Summit

siliconANGLE NEWS PEOPLE RESEARCH Wikibon TheCUBE SIGN IN

Amazon dives deeper into deep learning with MXNet framework

BY ERIC DAVID

UPDATED 14:45 EST . 22 NOVEMBER 2016

Nearly every major tech company, from Google Inc. to Facebook Inc., has been making deep learning an increasingly important aspect of their businesses, and today Amazon.com Inc. got into the game.

The company, whose Amazon Web Services is the leader in cloud computing, announced that it has chosen the MXNet software framework on which it will build its own foundation for deep learning, a branch of artificial intelligence that attempts to emulate in software the way the brain learns.

MXNET Architecture

- You can specify the context of the function to be executed within. This usually includes whether the function should be run on a CPU or a GPU, and if you specify a GPU, which GPU to use.

MXNET R Package

- The R Package can be downloaded using the following commands:

```
install.packages("drat", repos="https://cran.rstudio.com")
drat:::addRepo("dmlc")
install.packages("mxnet")
```

MXNET + R


```
#configure another network
data <- mx.symbol.Variable("data")
fc1 <- mx.symbol.FullyConnected(data, name = "fc1", num_hidden=10) #1st hidden layer
act1 <- mx.symbol.Activation(fc1, name = "sig", act_type="relu")
fc2 <- mx.symbol.FullyConnected(act1, name = "fc2", num_hidden=2) #2nd hidden layer
out <- mx.symbol.SoftmaxOutput(fc2, name = "soft")
```

Demo

- Github : http://bit.ly/APAC_MLDS

Lastly…

Thank you

Questions?

@kuanhoong

https://www.linkedin.com/in/kuanhoong

kuanhoong@gmail.com