AWS

SUMMIT

Scalable Deep Learning on AWS using Apache MXNet

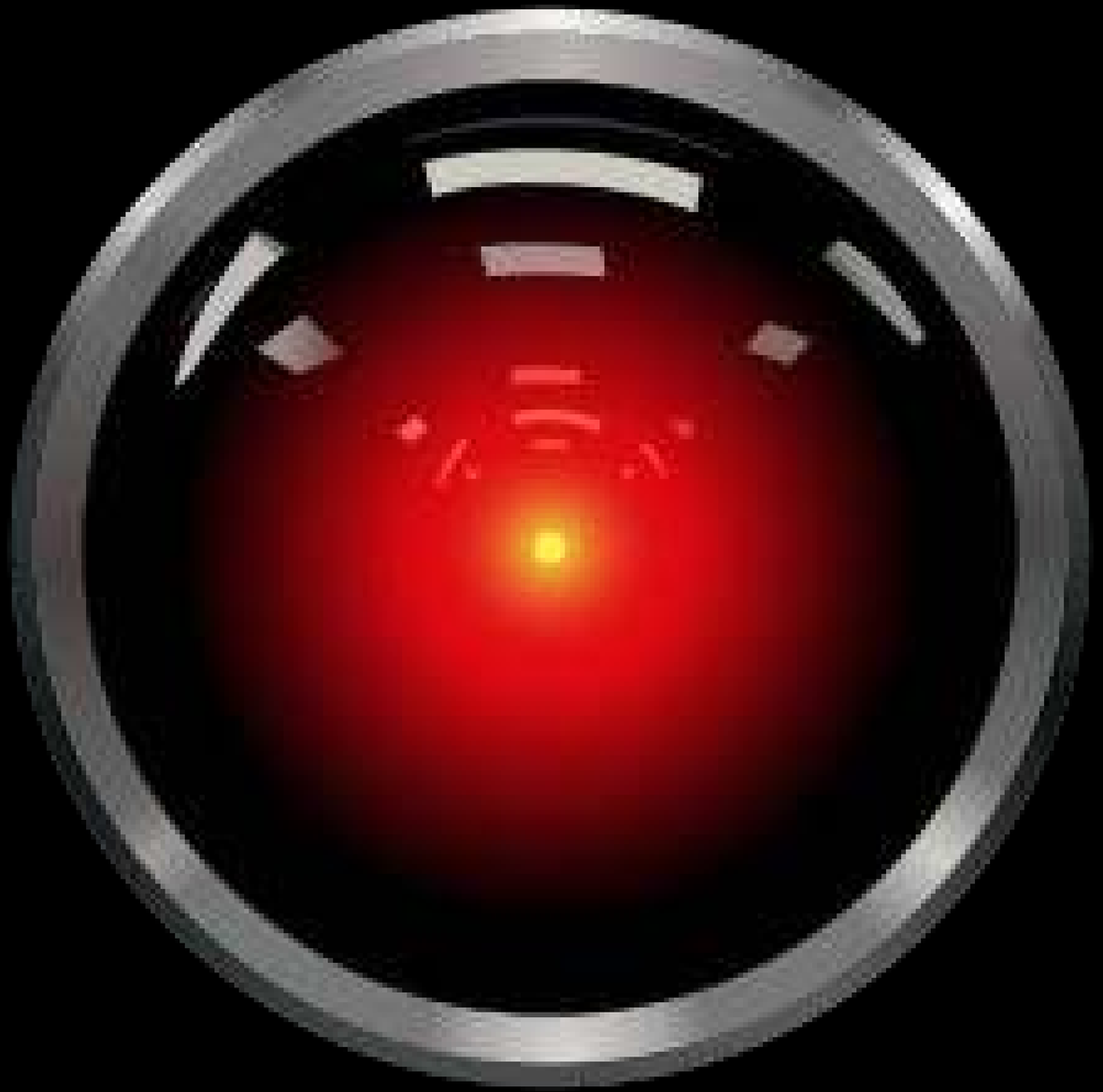
Julien Simon, Principal Technical Evangelist
@julsimon

amazon web services

Agenda

- AI: The Story So Far
- Applications of Deep Learning
- Apache MXNet Overview
- Apache MXNet API
- Code and Demos
- Tools and Resources

AI: The Story So Far

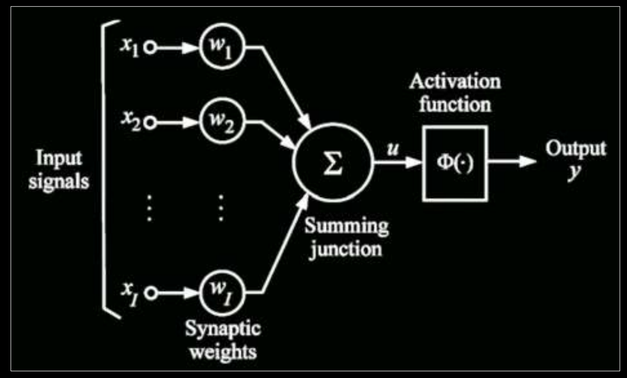


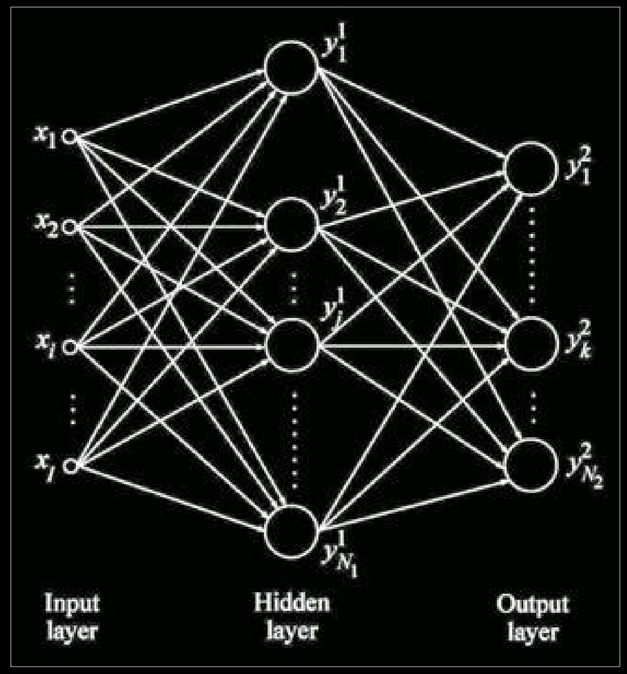
Where is HAL?

- Machine Learning is now a commodity, but still no HAL in sight
- Traditional Machine Learning doesn't work well with problems where features can't be explicitly defined
- So what about solving tasks that are easy for people to perform, but hard to describe formally?
- Is there a way to get informal knowledge into a computer?

Neural Networks, Revisited

- Universal approximation machine
- Through training, a neural network discovers features automatically
- Not new technology!
  - Perceptron - Rosenblatt, 1958
    image recognition, 20x20 pixels
  - Backpropagation - Werbos, 1975
- They failed back then because:
  - Data sets were too small
  - Solving large problems with fully connected networks required too much memory and computing power, aka the Curse of Dimensionality





Why It’s Different This Time

Everything is digital: large data sets are available

- Imagenet: 14M+ labeled images - http://www.image-net.org/
- YouTube-8M: 7M+ labeled videos - https://research.google.com/youtube8m/
- AWS public data sets - https://aws.amazon.com/public-datasets/

The parallel computing power of GPUs make training possible

- Simard et al (2005), Ciresan et al (2011)
- State of the art networks have hundreds of layers
- Baidu’s Chinese speech recognition: 4TB of training data, +/- 10 Exaflops

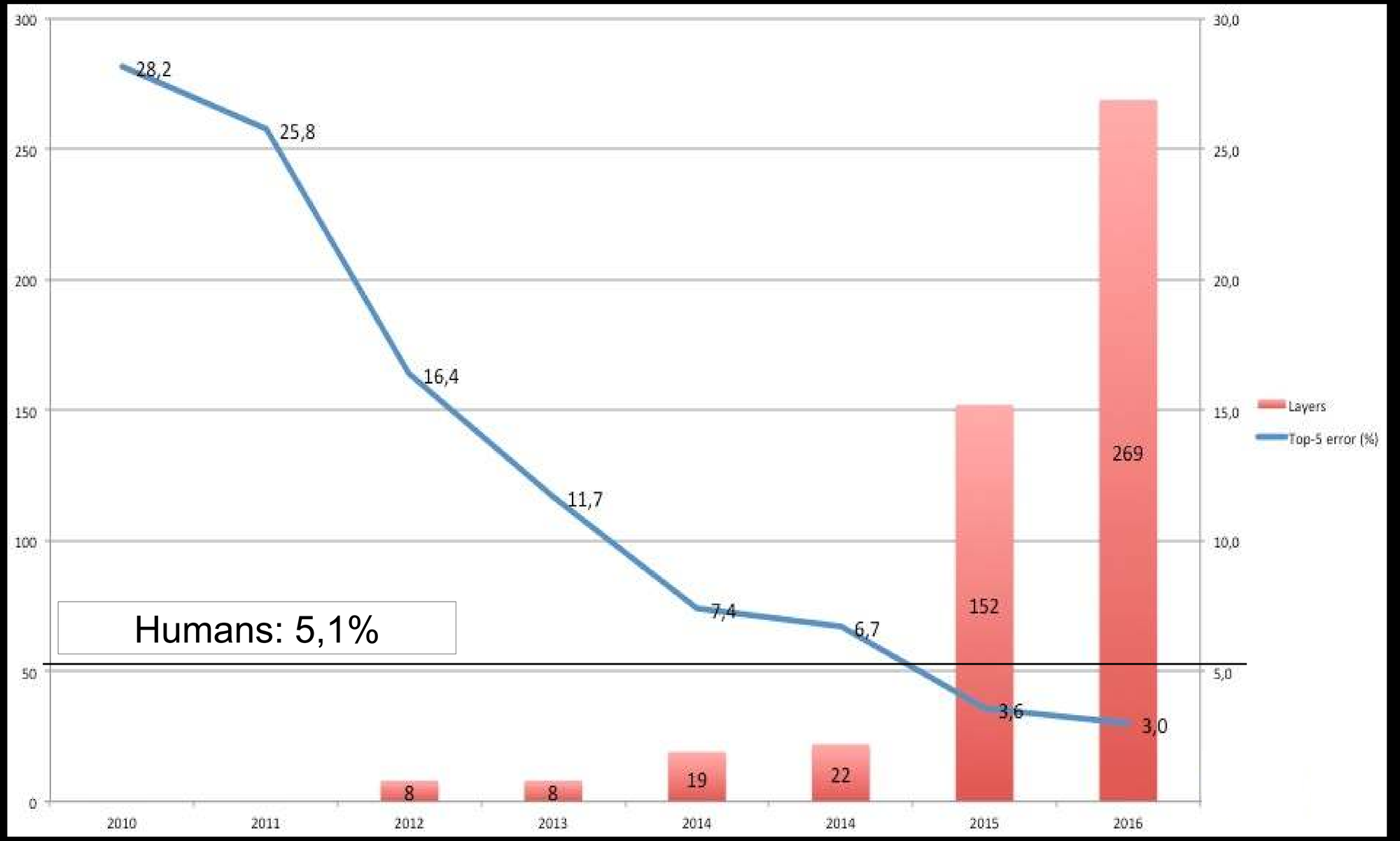
Cloud scalability and elasticity make training affordable

- Grab a lot of resources for fast training, then release them
- Using a DL model is lightweight: you can do it on a Raspberry Pi

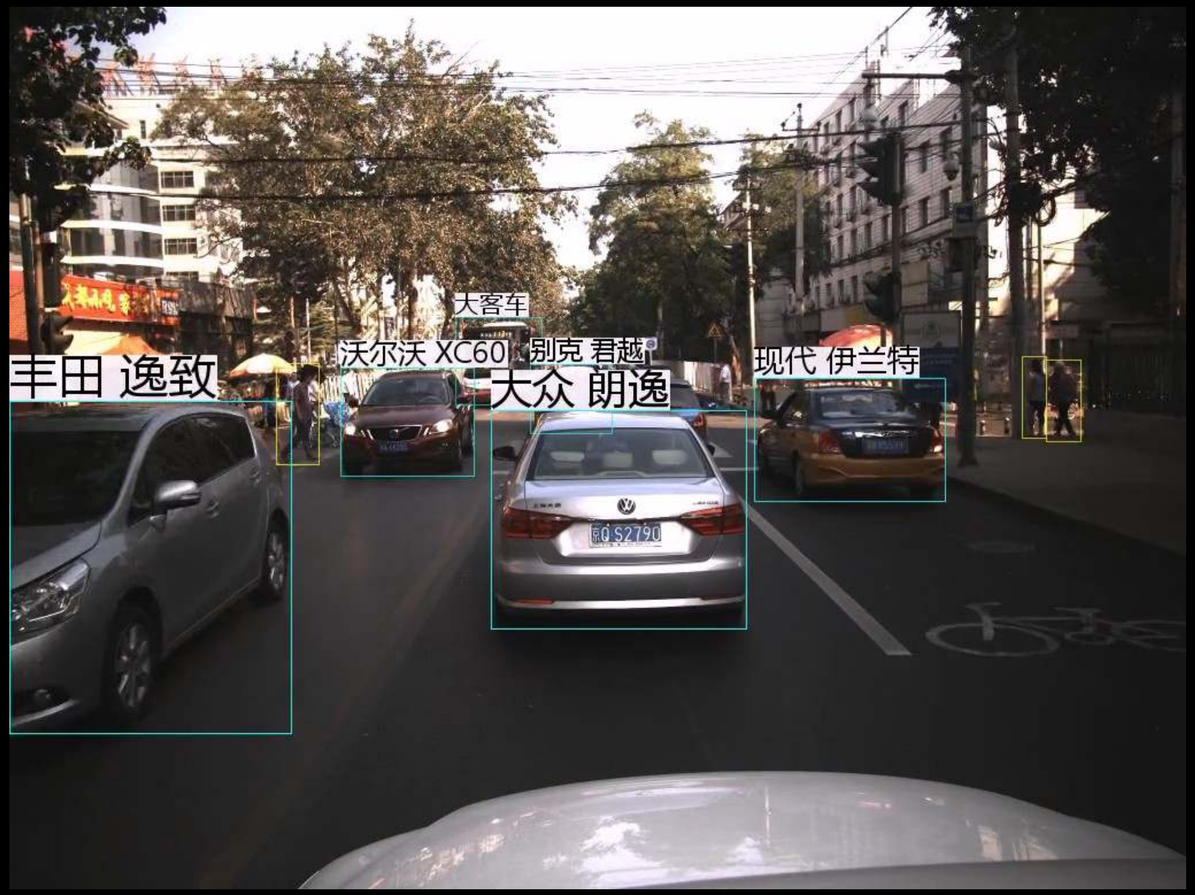
Applications of Deep Learning

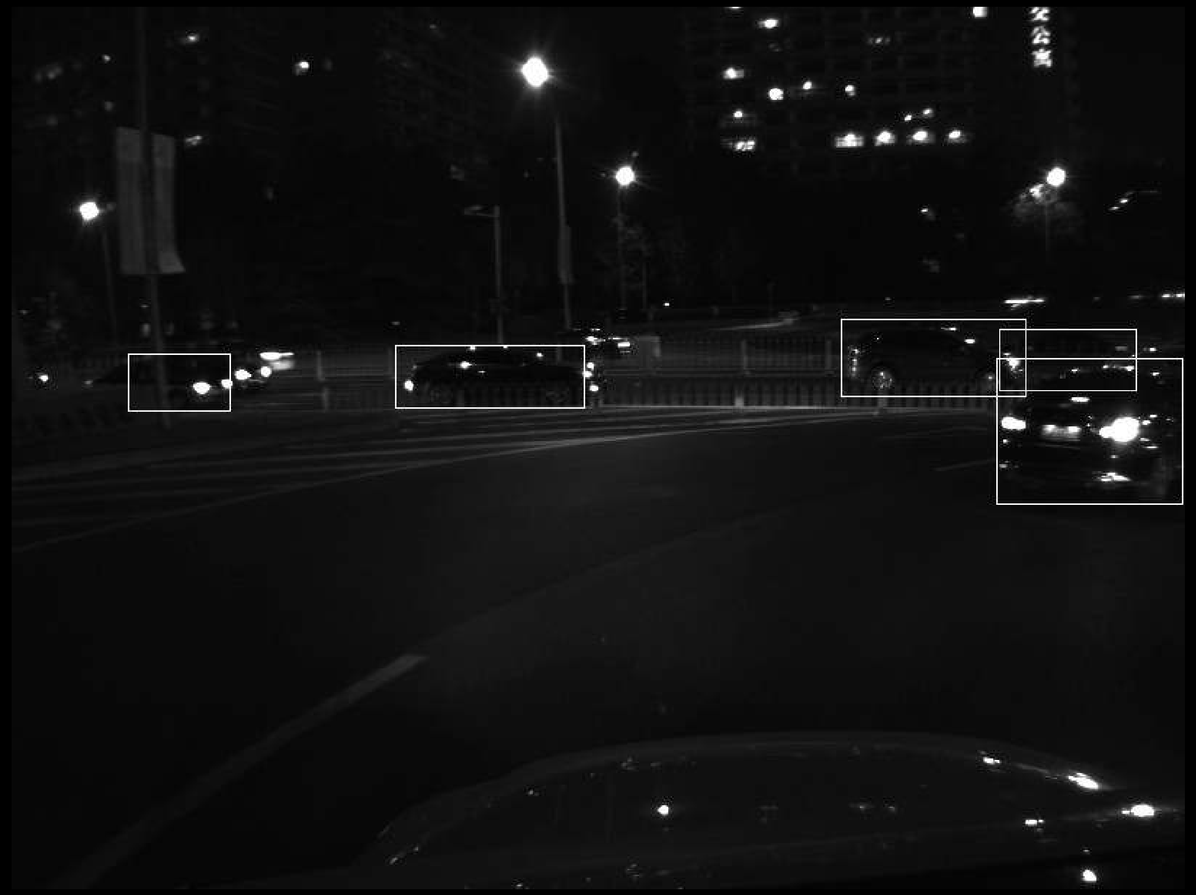
ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Same breed?



Autonomous Driving Systems






hue
PHILIPS
P
BBC
NEW
WIKIPE
The Economist
wink
IF
we
mo
amazon

Amazon Echo

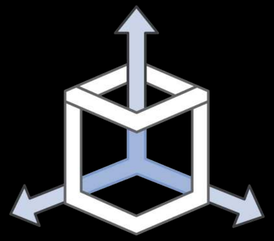
https://medium.com/@julsimon/create-your-own-basquiat-with-deep-learning-for-much-less-than-110-million-314aa07c9ba8

Apache MXNet Overview

Apache MXNet

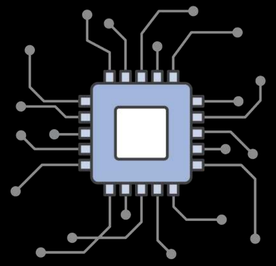
Programmable

Simple syntax,
multiple languages



Portable

Highly efficient
models for mobile
and IoT

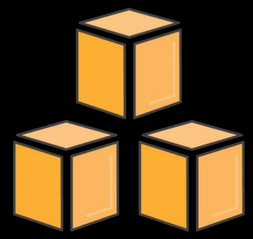


High Performance

Near linear scaling
across hundreds of GPUs

Most Open

Accepted into the
Apache Incubator



Best On AWS

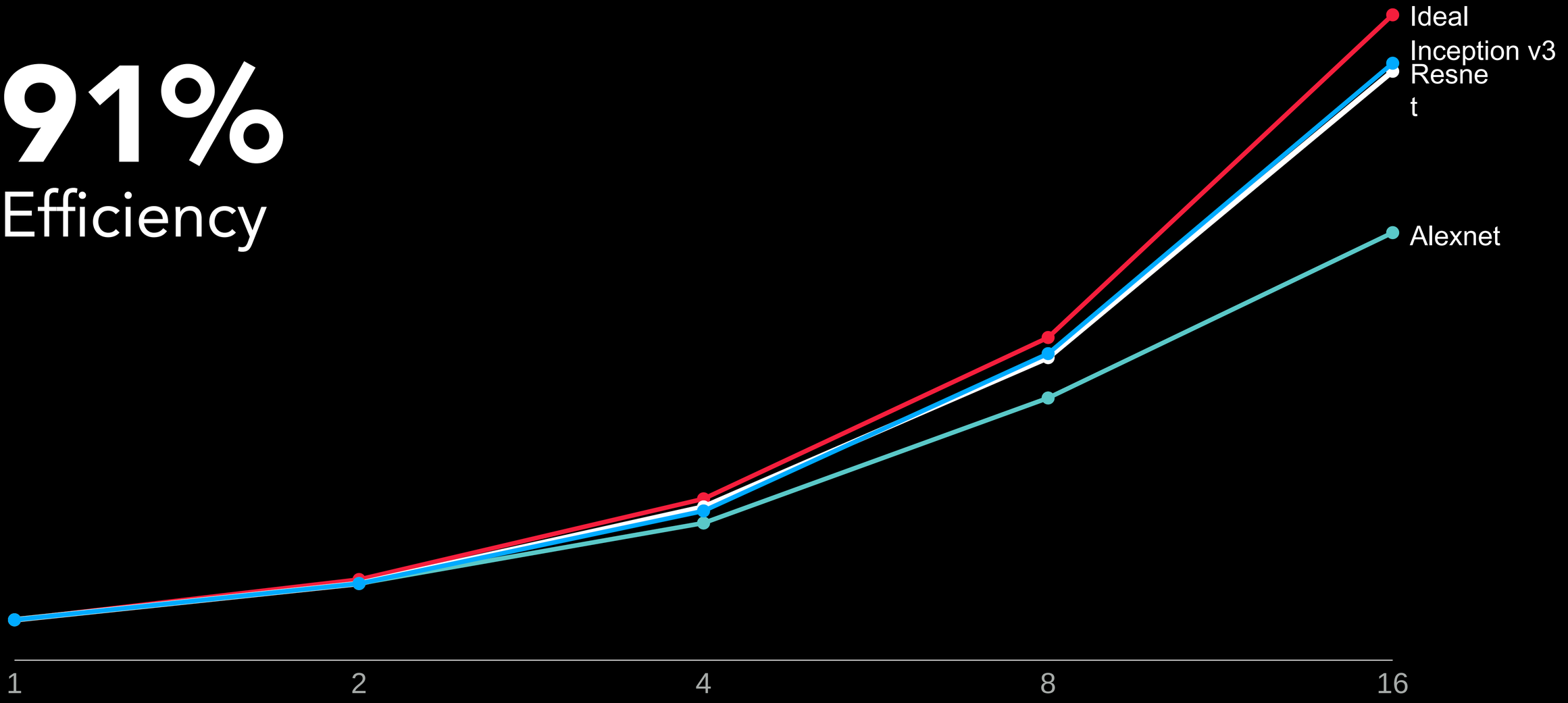
Optimized for
deep learning on
AWS

Deep Learning Framework Comparison

	Apache MXNet	**TensorFlow**	**Cognitive Toolkit**
Industry Owner	N/A – Apache Community	Google	Microsoft
Programmability	Imperative and Declarative	Declarative only	Declarative only
Language Support	R, Python, Scala, Julia, Cpp. Javascript, Go, Matlab and more..	Python, Cpp. Experimental Go and Java	Python, Cpp, Brainscript.
Code Length| AlexNet (Python)	44 sloc	107 sloc using TF.Slim	214 sloc
Memory Footprint (LSTM)	2.6GB	7.2GB	N/A

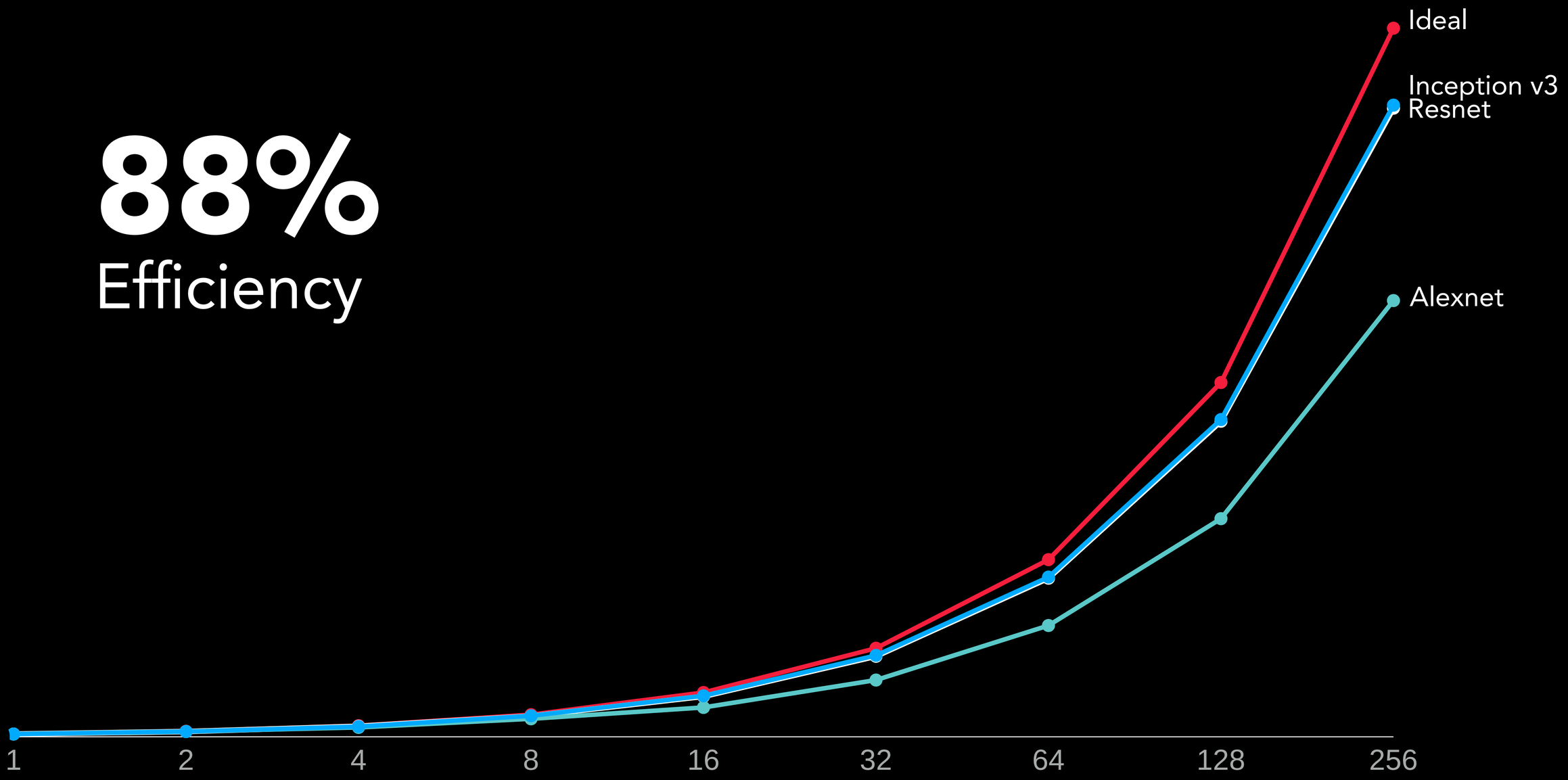
Multi-GPU Scaling With MXNet

91%
Efficiency



Multi-Machine Scaling With MXNet

88%
Efficiency



Apache MXNet API

Apache MXNet | The Basics

- ***NDArray*:** Manipulate multi-dimensional arrays in a command line paradigm (imperative).
- ***Symbol*:** Symbolic expression for neural networks (declarative).
- ***Module***: Intermediate-level and high-level interface for neural network training and inference.
- **Loading Data**: Feeding data into training/inference programs.
- **Mixed Programming**: Training algorithms developed using *NDArrays* in concert with *Symbols*.

https://medium.com/@julsimon/an-introduction-to-the-mxnet-api-part-1-848febdcf8ab

Imperative Programming

```
import numpy as np
a = np.ones(10)
b = np.ones(10) * 2
c = b * a
d = c + 1
```

Easy to tweak in Python

PROS

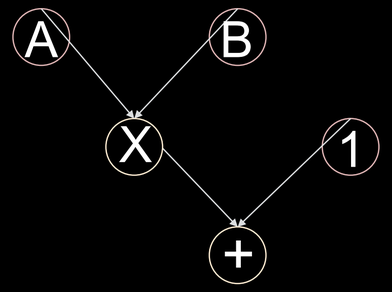
- Straightforward and flexible.
- Take advantage of language native features (loop, condition, debugger).
- E.g. Numpy, Matlab, Torch, …

CONS

- Hard to optimize

Declarative Programming

```
A = Variable('A')
B = Variable('B')
C = B * A
D = C + 1
f = compile(D)
d = f(A=np.ones(10),
      B=np.ones(10)*2)
```



C can share memory with D because C is deleted later

PROS

- More chances for optimization
- Cross different languages
- E.g. TensorFlow, Theano, Caffe

CONS

- Less flexible

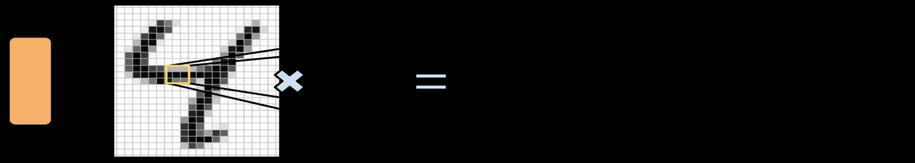
MXNet Symbol API

mx.model.FeedForward

model.fit

Input

mx.sym.**FullyConnected**(data, num_hidden=128)



mx.sym.**Convolution**(data, kernel=(5,5), num_filter=20)

mx.sym.**Pooling**(data, pool_type="max", kernel=(2,2),

stride=(2,2)

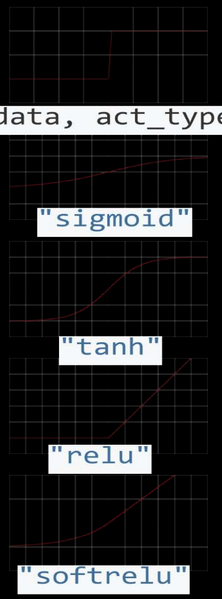
lstm.lstm_unroll(num_lstm_layer, seq_len, len, num_hidden, num_embed)

*cos(w, **queen**) = cos(w, **king**) - cos(w, **man**) + cos(w, **woman**)*

mx.symbol.**Embedding**(data, input_dim, output_dim = k)

mx.sym.SoftmaxOutput

Output



mx.sym.**Activation**(data, act_type="xxxx")

"sigmoid"

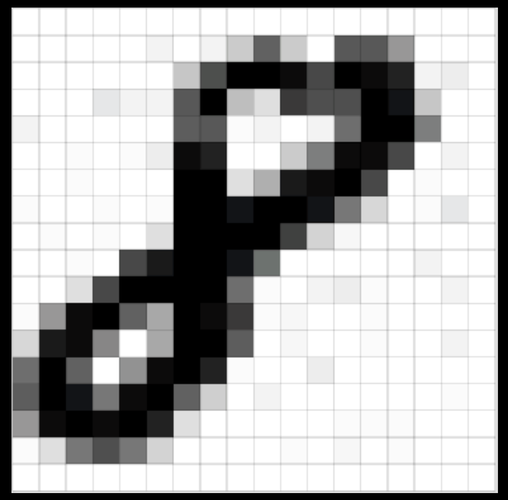
"tanh"

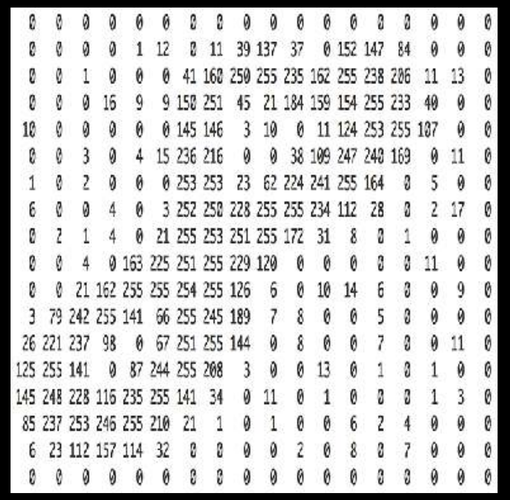
"relu"

"softrelu"

Demo #1 – Training MXNet on MNIST

https://medium.com/@julsimon/training-mxnet-part-1-mnist-6f0dc4210c62
https://github.com/juliensimon/aws/tree/master/mxnet/mnist



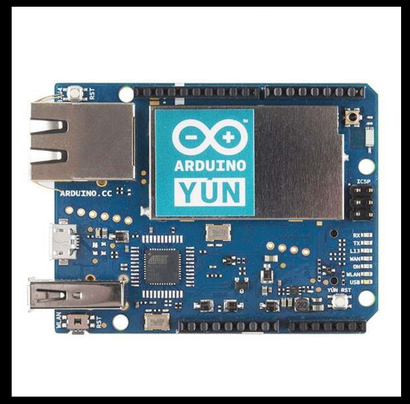


Demo #2 – Object Detection on a Raspberry Pi

https://medium.com/@julsimon/an-introduction-to-the-mxnet-api-part-6-fcdd7521ae87

GoPiGo

@CallMeJohnnyPi



Arduino Yùn

AWS
IoT
MQTT

mxnet

Amazon AI

Intelligent Services Powered By Deep Learning

Tools and Resources

AWS Deep Learning AMI

Up to~40k CUDA cores

Apache MXNet

TensorFlow

Theano

Caffe

Torch

Keras

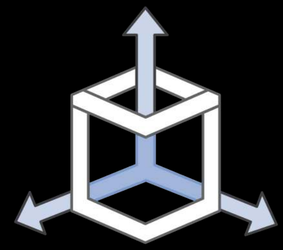
Pre-configured CUDA drivers, MKL

Anaconda, Python3

Ubuntu and Amazon Linux

+ CloudFormation template

+ Container Image



One-Click GPU or CPU Deep Learning

Additional Resources

MXNet Resources

- MXNet Blog Post | AWS Endorsement
- Read up on MXNet and Learn More: mxnet.io
- MXNet Github Repo
- MXNet Recommender Systems Talk | Leo Dirac

AWS Resources

- Deep Learning AMI |Amazon Linux
- Deep Learning AMI | Ubuntu
- CloudFormation Template Instructions
- Deep Learning Benchmark
- MXNet on Lambda
- MXNet on ECS/Docker


AWS
SUMMIT
Thank You!
@julsimon
amazon
web services