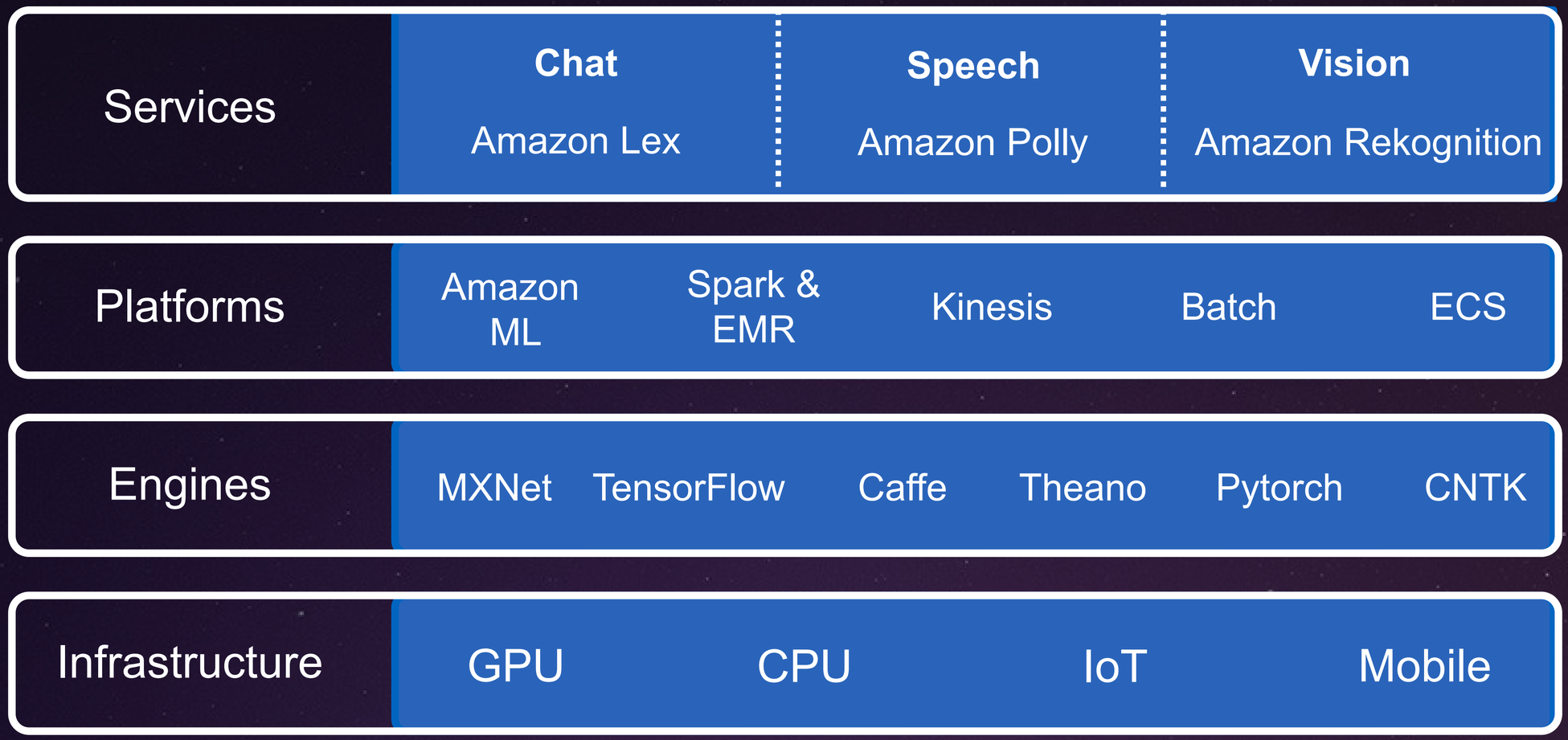
Scalable Deep Learning on AWS using Apache MXNet

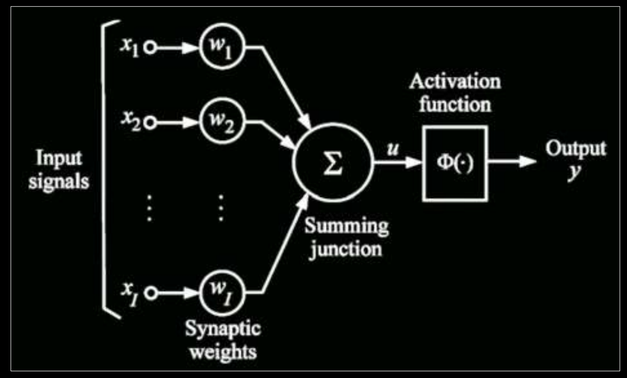
aws.amazon.com/evangelists/julien-simon
@julsimon

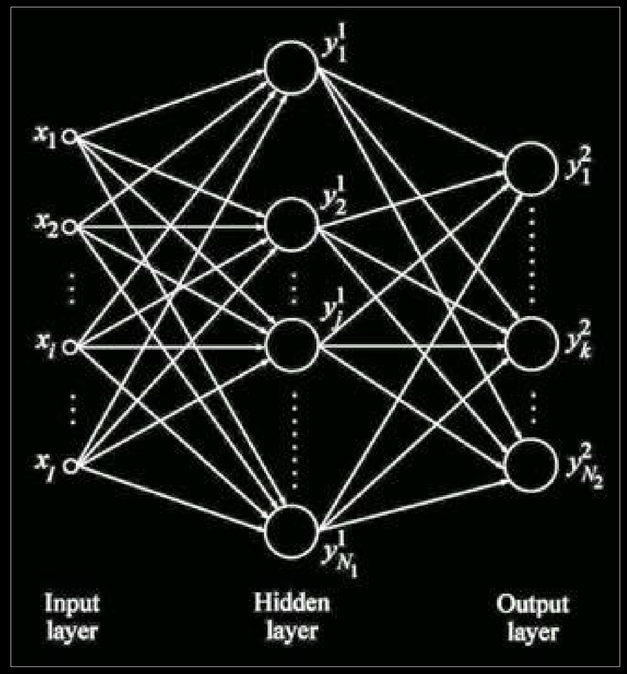
Amazon AI: Artificial Intelligence In The Hands Of Every Developer



Neural Networks in one slide

- 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

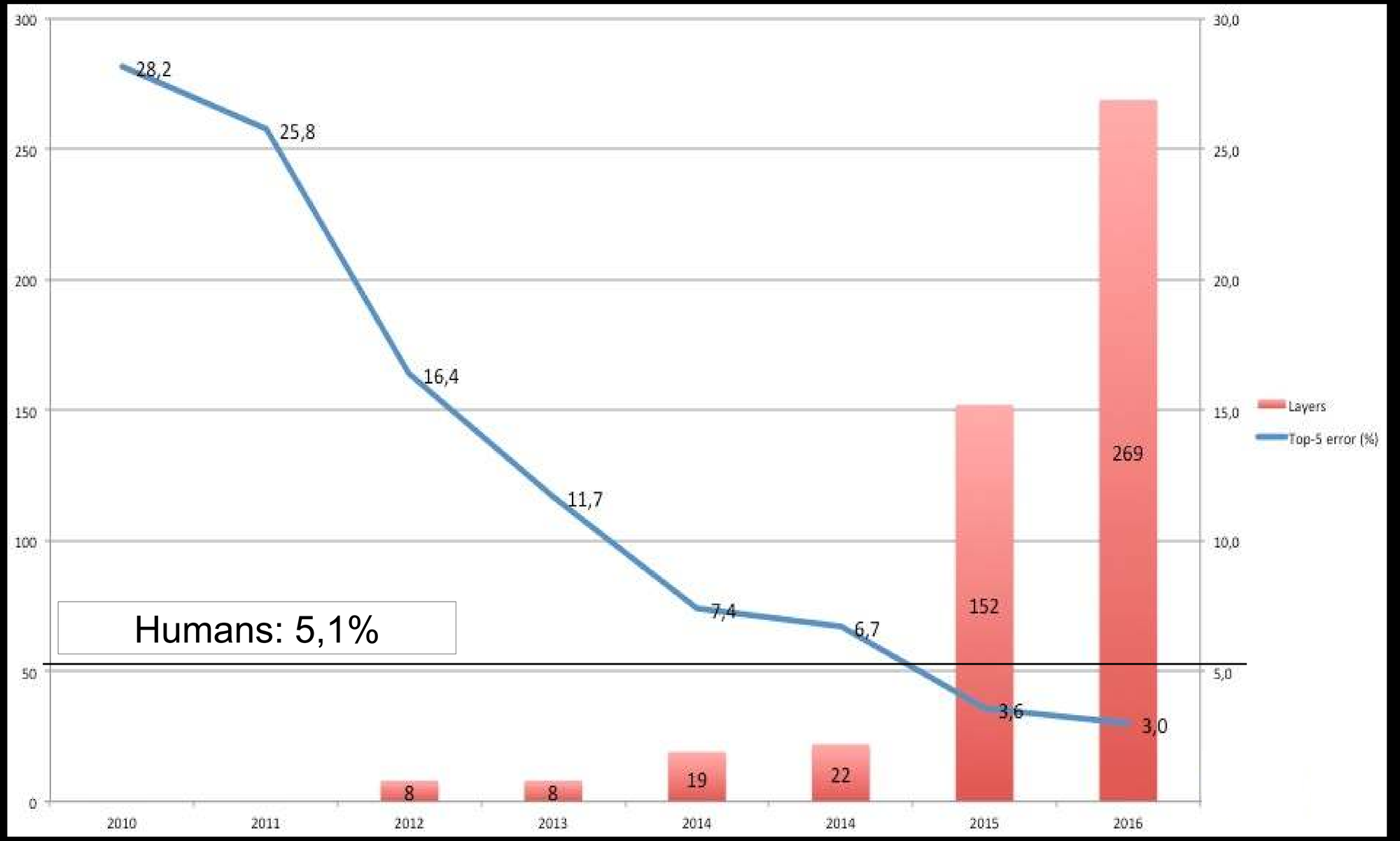
Cloud scalability and elasticity make training affordable

- 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?




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

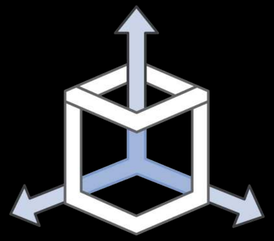
Amazon Echo

Apache MXNet Overview

Apache MXNet

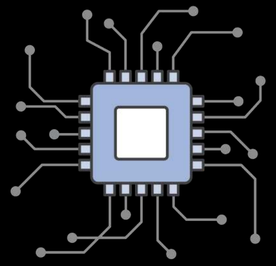
Programmable

Simple syntax,
multiple languages



Portable

Highly efficient
models for mobile
and IoT

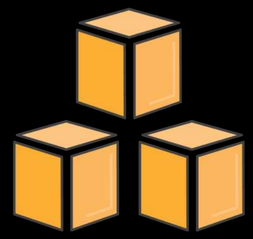


High Performance

Near linear scaling
across hundreds of GPUs

Most Open

Accepted into the
Apache Incubator

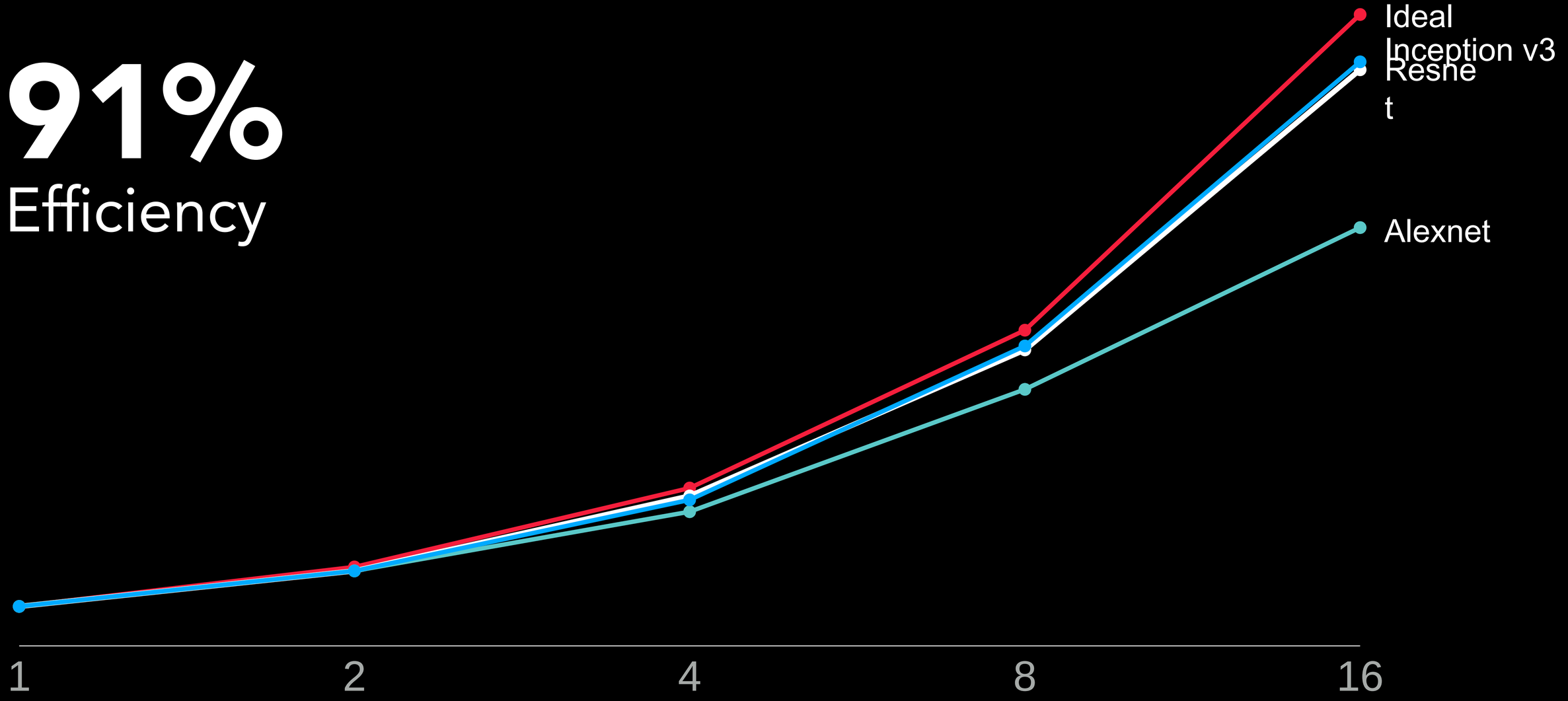


Best On AWS

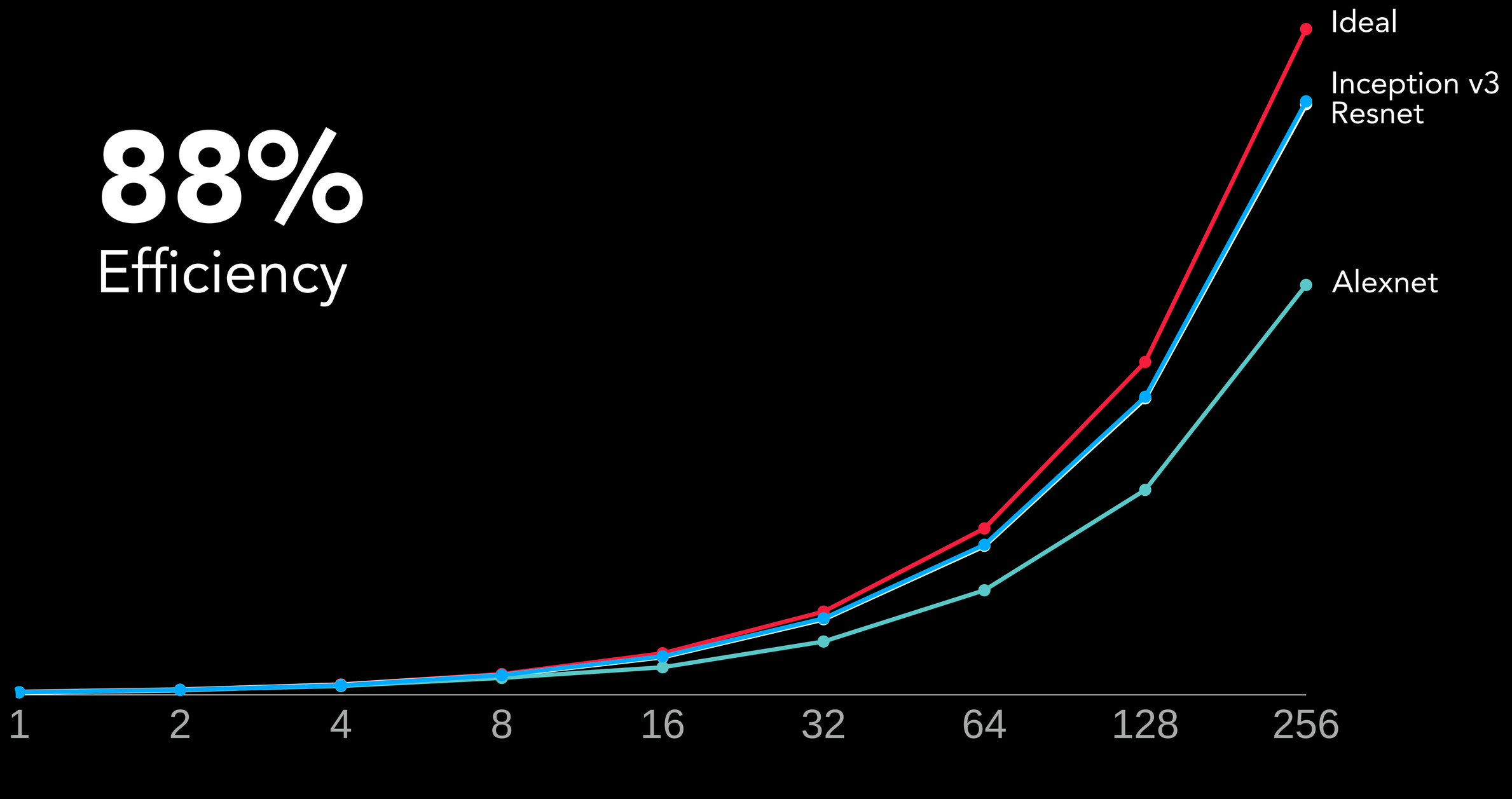
Optimized for
deep learning on
AWS

Multi-GPU Scaling With MXNet

91%
Efficiency



Multi-Machine Scaling With MXNet



Apache MXNet API

Apache MXNet | The Basics

- ***NDArray*:** Manipulate multi-dimensional arrays in a command line paradigm (imperative programming).
- ***Symbol*:** Symbolic expression for neural networks (declarative programming).
- ***Module:*** 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*.

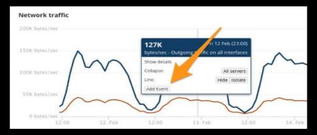
https://medium.com/@julsimon/an-introduction-to-the-mxnet-api-part-1-848febdcf8ab

mx.model.FeedForward

model.fit

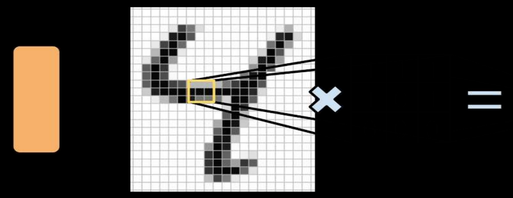
Input

Output



mx.sym.SoftmaxOutput

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),

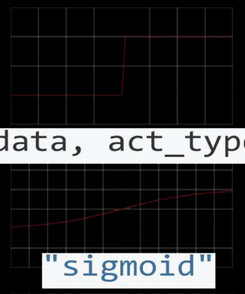
stride=(2,2)

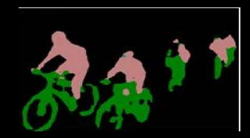
lstm.lstm_unroll(num_lstm_layer, seq_len, len, num_hidden, num_embed)

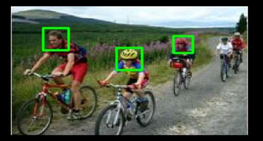
*cos(w, **queen**) = cos(w, **king**) - cos(w, **man**) + cos(w, **woman**)*

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

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

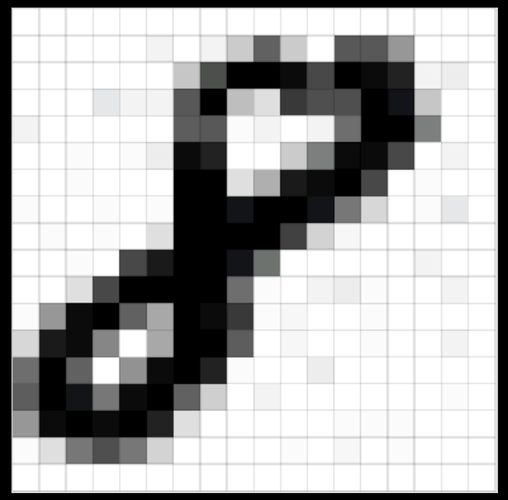






Demo – Training MXNet on MNIST

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



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	12	8	11	39	137	37	0	152	147	84	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	41	160	250	255	235	162	255	238	206	11	13	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	16	9	9	150	251	45	21	184	159	154	255	233	40	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	145	146	3	10	0	11	124	253	255	107	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	0	4	15	236	216	0	0	38	109	247	248	169	0	11	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	0	0	0	253	253	23	62	224	241	255	164	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	4	0	3	252	250	228	255	255	234	112	28	0	2	17	0	0	0	0	0	0	0	0	0	0	0	0
0	2	1	4	0	21	255	253	251	255	172	31	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	163	225	251	255	229	120	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	21	162	255	255	254	255	126	6	0	10	14	6	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0
3	79	242	255	141	66	255	245	189	7	8	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	221	237	98	0	67	251	255	144	0	8	0	0	7	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0
125	255	141	0	87	244	255	208	3	0	0	13	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
145	248	228	116	235	255	141	34	0	11	0	1	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0
85	237	253	246	255	210	21	1	0	1	0	0	6	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	23	112	157	114	32	0	0	0	0	2	0	8	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Additional Resources

MXNet Resources

- MXNet Blog Post | AWS Endorsement
- Read up on MXNet and Learn More: mxnet.io
- MXNet Github Repo

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!

aws.amazon.com/evangelists/julien-simon

@julsimon

amazon web services