Deep Learning with Apache MXNet

Julien Simon, AI Evangelist, EMEA
@julsimon

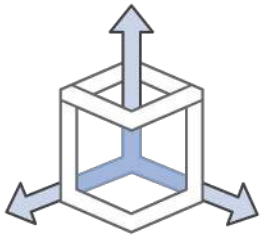
What to expect

- Apache MXNet
- Demos using Jupyter notebooks
- Resources
- Q&A

Apache MXNet: Open Source library for Deep Learning

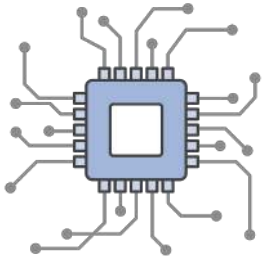
Programmable

Simple syntax, multiple languages



Portable

Highly efficient models for mobile and IoT

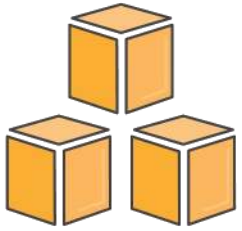


High Performance

Near linear scaling across hundreds of GPUs

Most Open

Accepted into the Apache Incubator



Best On AWS

Optimized for Deep Learning on AWS

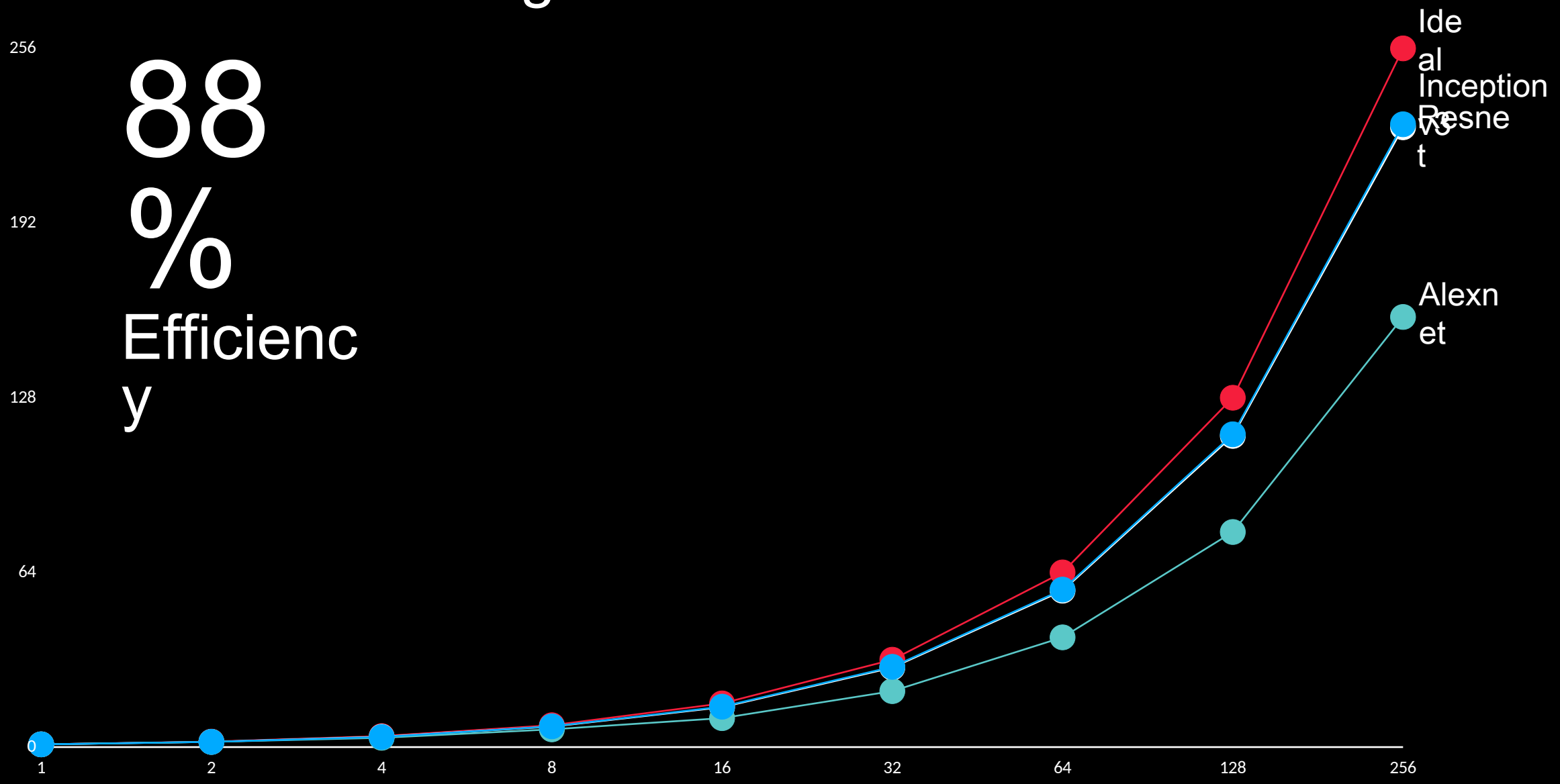
CPU or GPU: your choice

```
mod = mx.mod.Module(lenet)

mod = mx.mod.Module(lenet, context=mx.gpu(0))

mod = mx.mod.Module(lenet,
context=(mx.gpu(7), mx.gpu(8), mx.gpu(9)))
```

Multi-GPU Scaling With MXNet



The Apache MXNet API

- Storing and accessing data in multi-dimensional arrays
  → *NDArray* API
- Building models (layers, weights, activation functions)
  → *Symbol* API
- Serving data during training and validation
  → Iterators
- Training and using models
  → *Module* API

mx.model.FeedForward

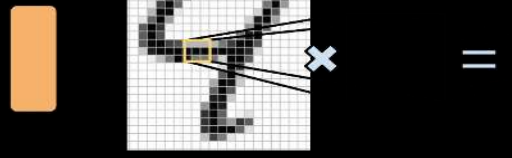
model.fit

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),

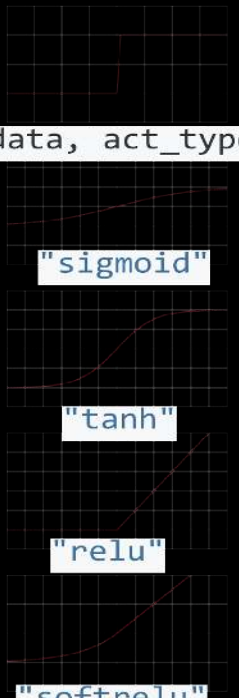
stride=(2,2)

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

$$cos(w, \textbf{queen}) = cos(w, \textbf{king}) - cos(w, \textbf{man}) + cos(w, \textbf{woman})$$

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

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



Gluon: Deep Learning gets even easier

https://github.com/gluon-api/

- Available now in MXNet, soon in Microsoft Cognitive Toolkit
- Developer-friendly high-level API
- Dynamic networks can be modified during training
- No compromise on performance
- Extensive model zoo

https://aws.amazon.com/blogs/aws/introducing-gluon-a-new-library-for-machine-learning-from-aws-and-microsoft/

Gluon Model Zoo

vgg11	VGG-11 model from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg13	VGG-13 model from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg16	VGG-16 model from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg19	VGG-19 model from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg11_bn	VGG-11 model with batch normalization from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg13_bn	VGG-13 model with batch normalization from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg16_bn	VGG-16 model with batch normalization from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
vgg19_bn	VGG-19 model with batch normalization from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.

VGG	VGG model from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.
get_vgg	VGG model from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper.

resnet18_v1	ResNet-18 V1 model from "Deep Residual Learning for Image Recognition" paper.
resnet34_v1	ResNet-34 V1 model from "Deep Residual Learning for Image Recognition" paper.
resnet50_v1	ResNet-50 V1 model from "Deep Residual Learning for Image Recognition" paper.
resnet101_v1	ResNet-101 V1 model from "Deep Residual Learning for Image Recognition" paper.
resnet152_v1	ResNet-152 V1 model from "Deep Residual Learning for Image Recognition" paper.
resnet18_v2	ResNet-18 V2 model from "Identity Mappings in Deep Residual Networks" paper.
resnet34_v2	ResNet-34 V2 model from "Identity Mappings in Deep Residual Networks" paper.
resnet50_v2	ResNet-50 V2 model from "Identity Mappings in Deep Residual Networks" paper.
resnet101_v2	ResNet-101 V2 model from "Identity Mappings in Deep Residual Networks" paper.
resnet152_v2	ResNet-152 V2 model from "Identity Mappings in Deep Residual Networks" paper.

ResNetV1	ResNet V1 model from "Deep Residual Learning for Image Recognition" paper.
ResNetV2	ResNet V2 model from "Identity Mappings in Deep Residual Networks" paper.
BasicBlockV1	BasicBlock V1 from "Deep Residual Learning for Image Recognition" paper.
BasicBlockV2	BasicBlock V2 from "Identity Mappings in Deep Residual Networks" paper.
BottleneckV1	Bottleneck V1 from "Deep Residual Learning for Image Recognition" paper.
BottleneckV2	Bottleneck V2 from "Identity Mappings in Deep Residual Networks" paper.
get_resnet	ResNet V1 model from "Deep Residual Learning for Image Recognition" paper.

mobilenet1_0	MobileNet model from the "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" paper, with width multiplier 1.0.
mobilenet0_75	MobileNet model from the "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" paper, with width multiplier 0.75.
mobilenet0_5	MobileNet model from the "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" paper, with width multiplier 0.5.
mobilenet0_25	MobileNet model from the "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" paper, with width multiplier 0.25.

MobileNet	MobileNet model from the "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" paper.

densenet121	Densenet-BC 121-layer model from the "Densely Connected Convolutional Networks" paper.
densenet161	Densenet-BC 161-layer model from the "Densely Connected Convolutional Networks" paper.
densenet169	Densenet-BC 169-layer model from the "Densely Connected Convolutional Networks" paper.
densenet201	Densenet-BC 201-layer model from the "Densely Connected Convolutional Networks" paper.

DenseNet	Densenet-BC model from the "Densely Connected Convolutional Networks" paper.

inception_v3	Inception v3 model from "Rethinking the Inception Architecture for Computer Vision" paper.
Inception3	Inception v3 model from "Rethinking the Inception Architecture for Computer Vision" paper.

alexnet	AlexNet model from the "One weird trick..." paper.
AlexNet	AlexNet model from the "One weird trick..." paper.

squeezenet1_0	SqueezeNet 1.0 model from the "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size" paper.
squeezenet1_1	SqueezeNet 1.1 model from the official SqueezeNet repo.

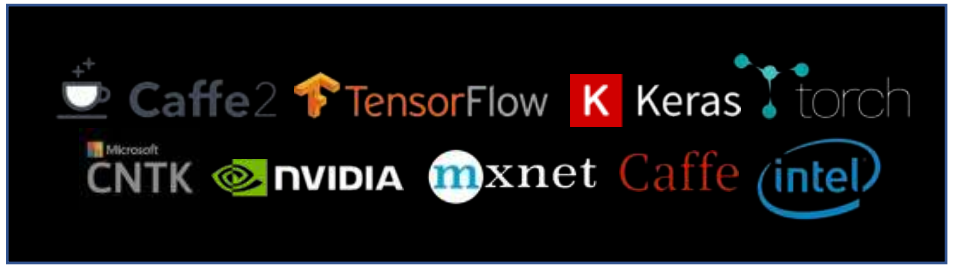
SqueezeNet	SqueezeNet model from the "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size" paper.

VGG
ResNet
AlexNet
DenseNet
SqueezeNe
t
Inception
MobileNet

https://mxnet.incubator.apache.org/versions/master/api/python/gluon/model_zoo.html

Deep Learning in practice

- One-click launch
- Single node or distributed
- CPU, GPU, FPGA
- NVIDIA & Intel libraries
- Anaconda Data Science Platform
- Python w/ AI/ML/DL libraries



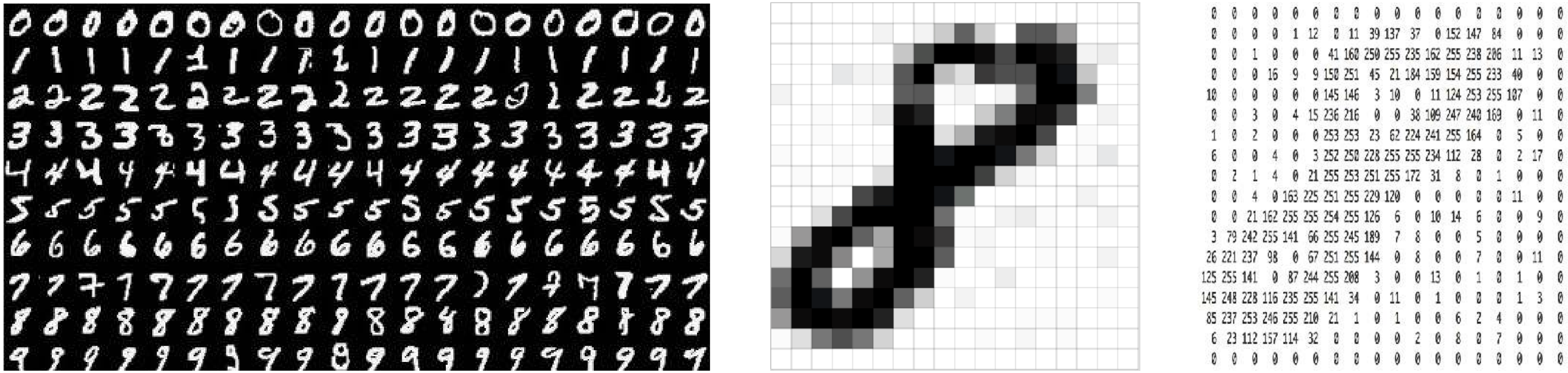
Demos

https://github.com/juliensimon/dlnotebooks

1) Synthetic data set
2) Learn MNIST with a Multi-Layer Perceptron
3) Learn MNIST with the LeNet CNN
4) Predict handmade MNIST samples

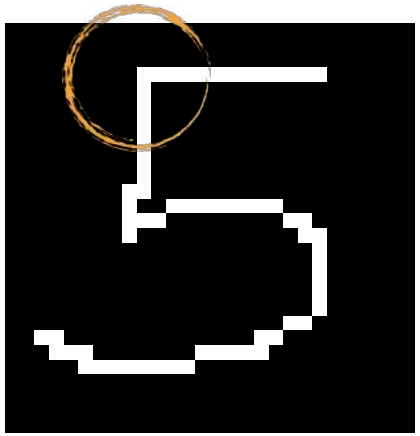
The MNIST data set

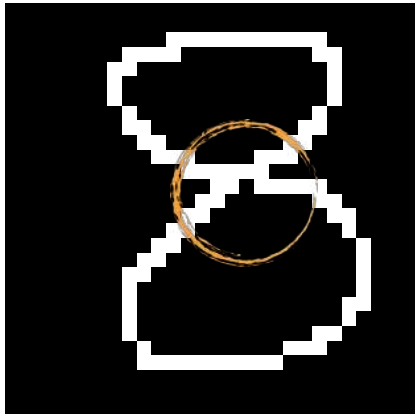
- 70,000 hand-written digits
- 28x28 grayscale images



Training models on MNIST

- MLP does well on MNIST, but not on real-life samples.
- MLP flattens the 28x28 image into a 784-byte vector
  → the 2-D relationship between pixels is lost.
- Look at the ‘5’: the top-left angle is a unique feature. No other digit exhibits this. Same thing for the intersection on the ‘8’.
- CNNs such as LeNet work on 2-D images and learn to detect these unique geometric features.
- The result is a more accurate and more robust network.





Resources

https://aws.amazon.com/machine-learning

https://aws.amazon.com/blogs/ai

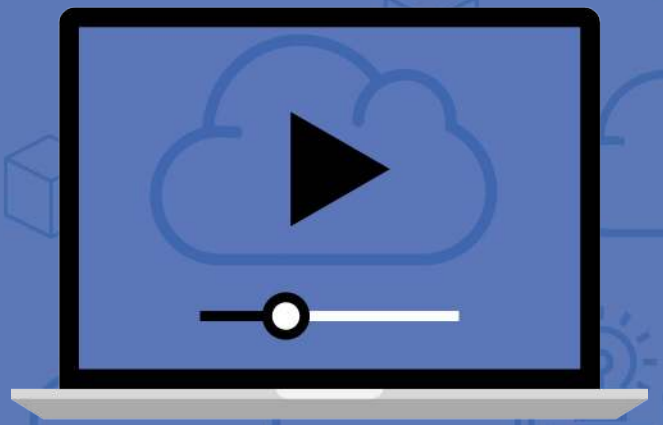
https://mxnet.incubator.apache.org

https://github.com/apache/incubator-mxnet

https://github.com/gluon-api

https://medium.com/@julsimon

Thank you!



Julien Simon, AI Evangelist, EMEA
@julsimon