

# MXNet Ecosystem

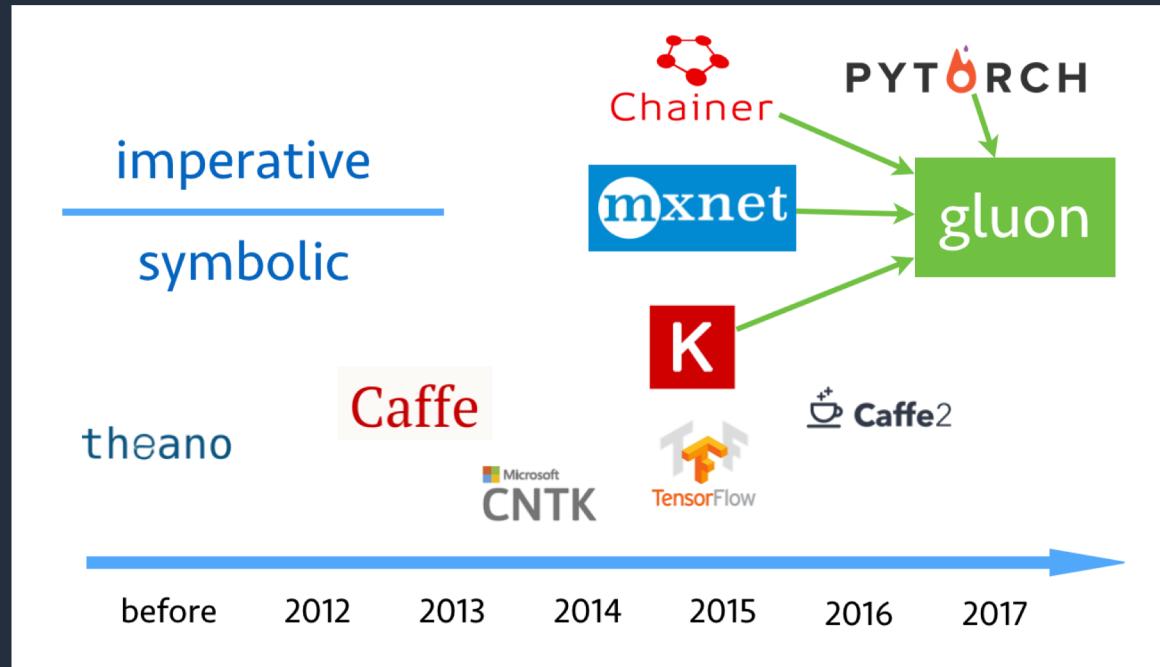


Cyrus Vahid – cyrusmv@amazon.com

Principal Solutions Engineer– AWS Deep Engine

\*

# Apache MXNet - History



# Multi-language Support

Java

Perl

Julia

Clojure

Python

Scala

C++

R

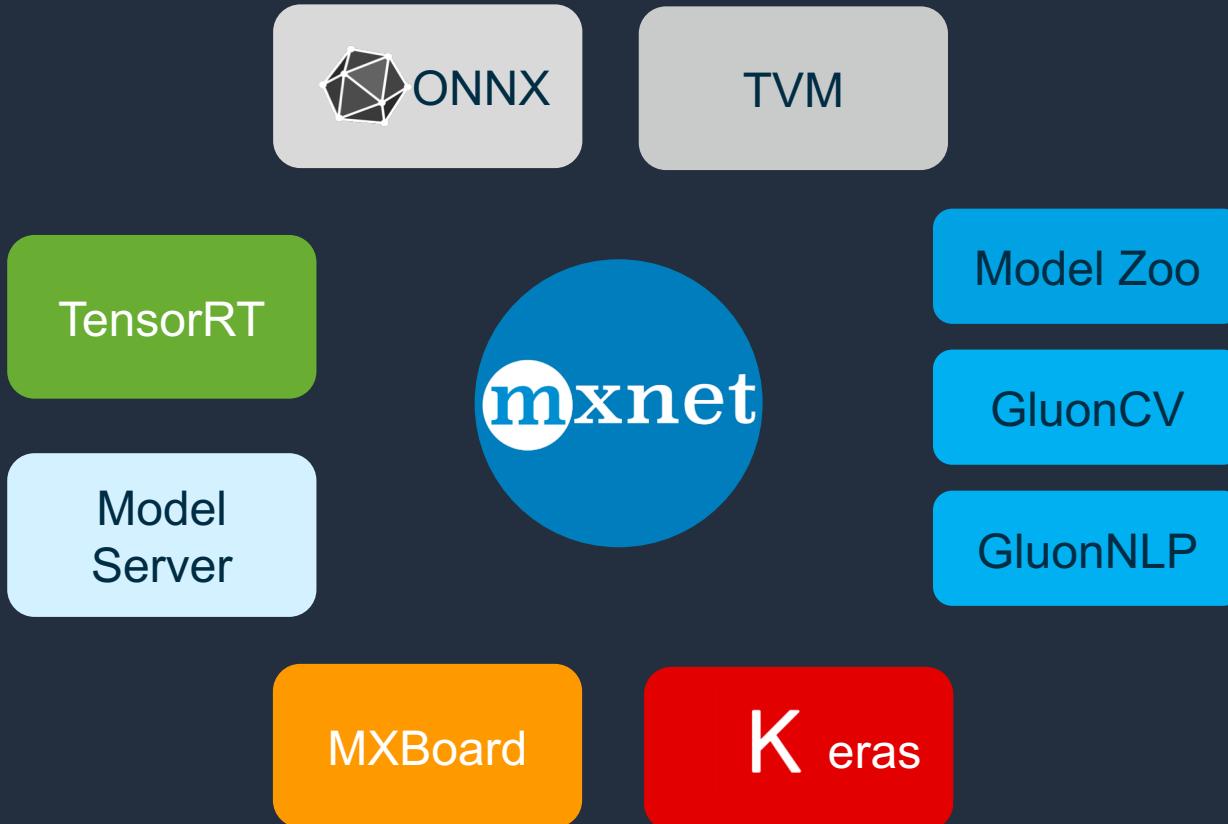
*Frontend*

While keeping high performance from efficient backend

*Backend*

C++

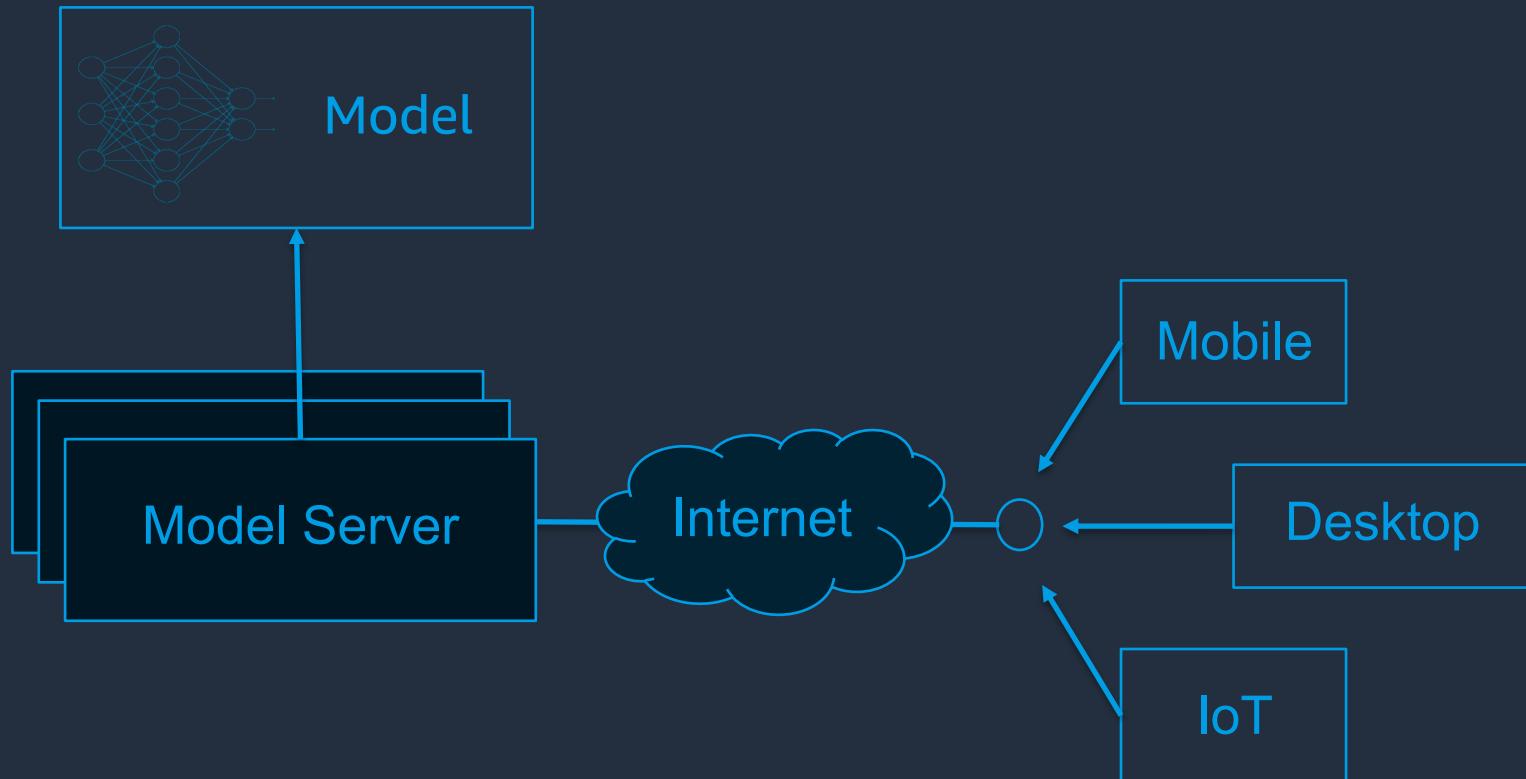
# Apache MXNet Ecosystem



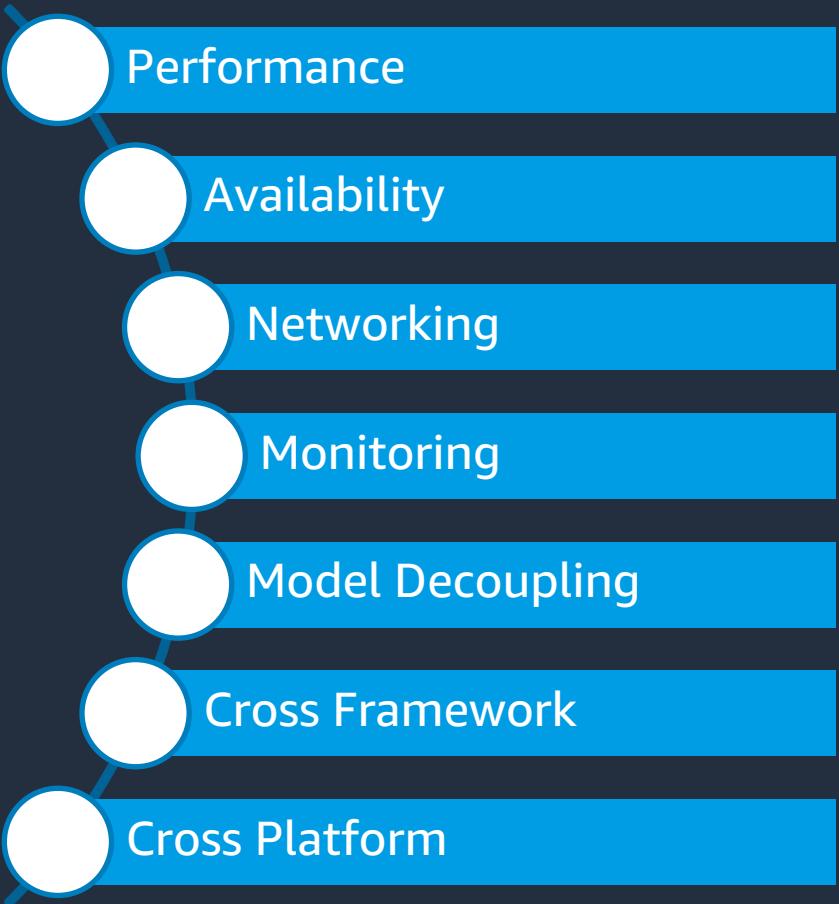


Model Server

# So what does a deployed model looks like?



# The Undifferentiated Heavy Lifting of Model Serving





# Model Archive

Trained  
Network

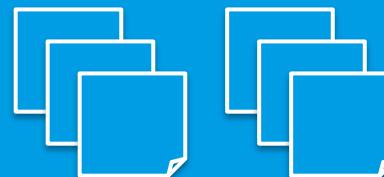
Model  
Signature

Custom  
Code

Auxiliary  
Assets

Model Export CLI

Model Archive



[Back](#)

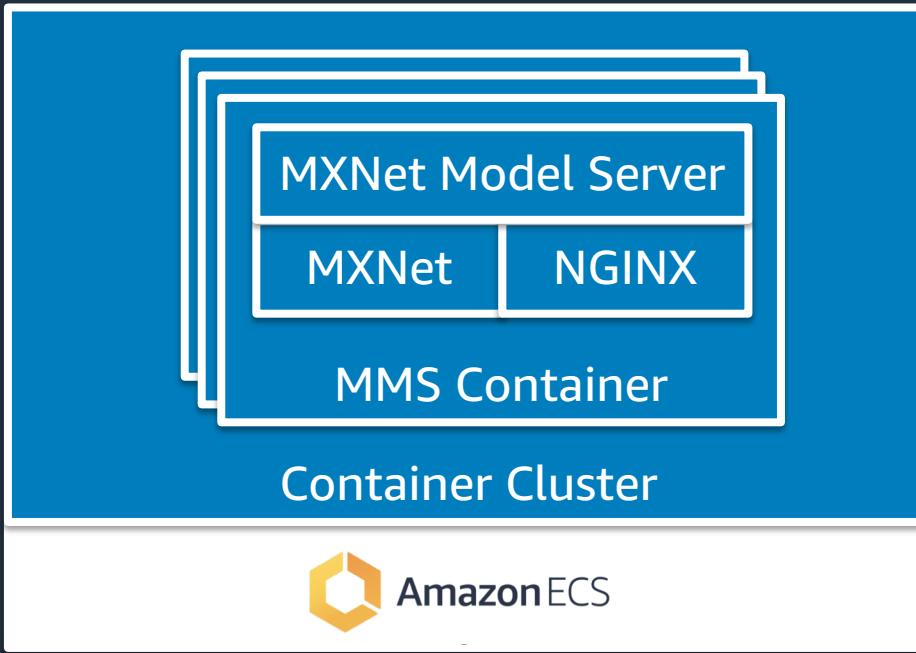


1. (bash)  
(mms-demo) 8c8590170440:code hag\$



# Containerization

Lightweight virtualization, isolation, runs anywhere



Pull or Build  
Push  
Launch



[Back](#)

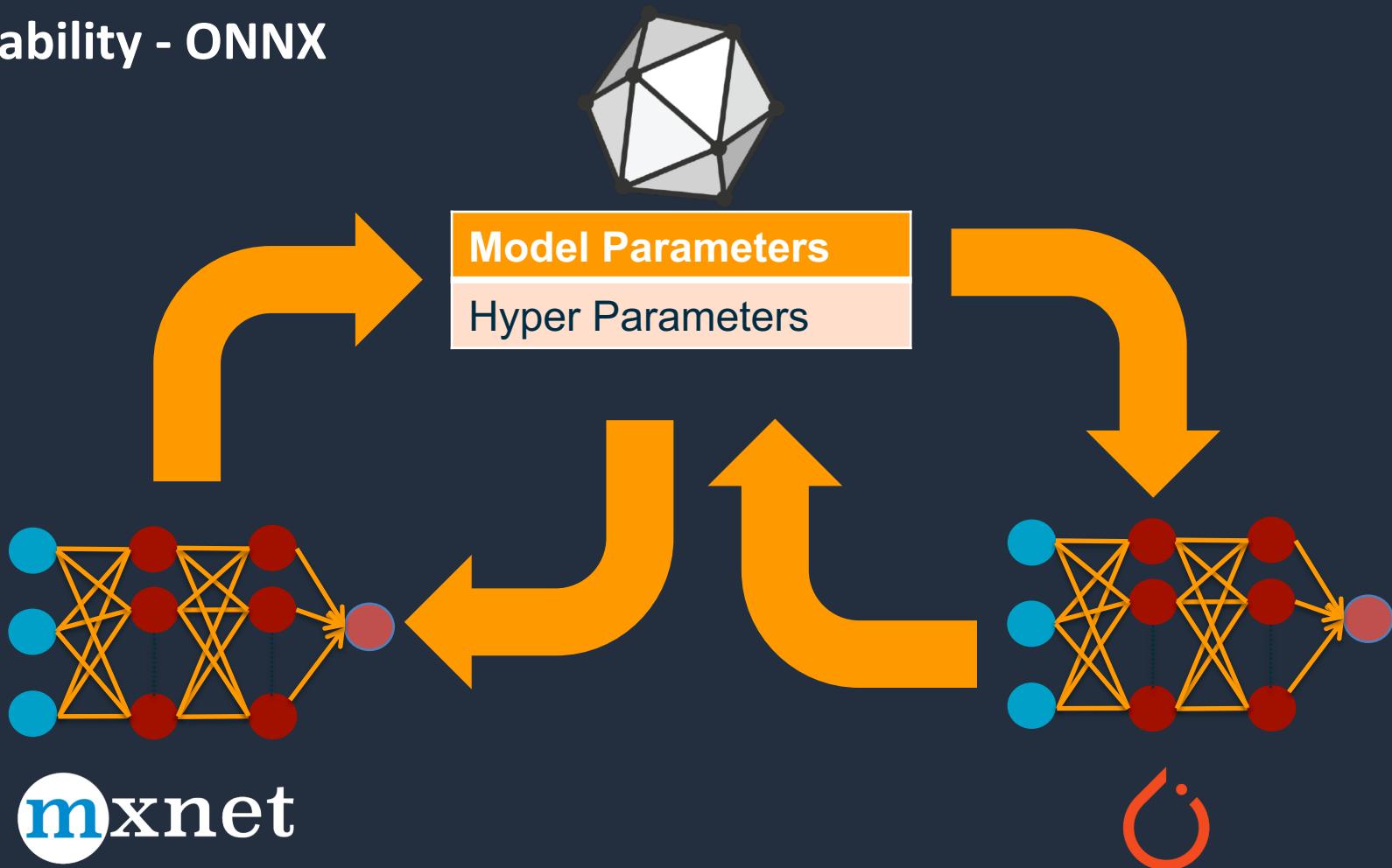


Demo: <https://bit.ly/2vqj4A7>



ONNX

# Portability - ONNX



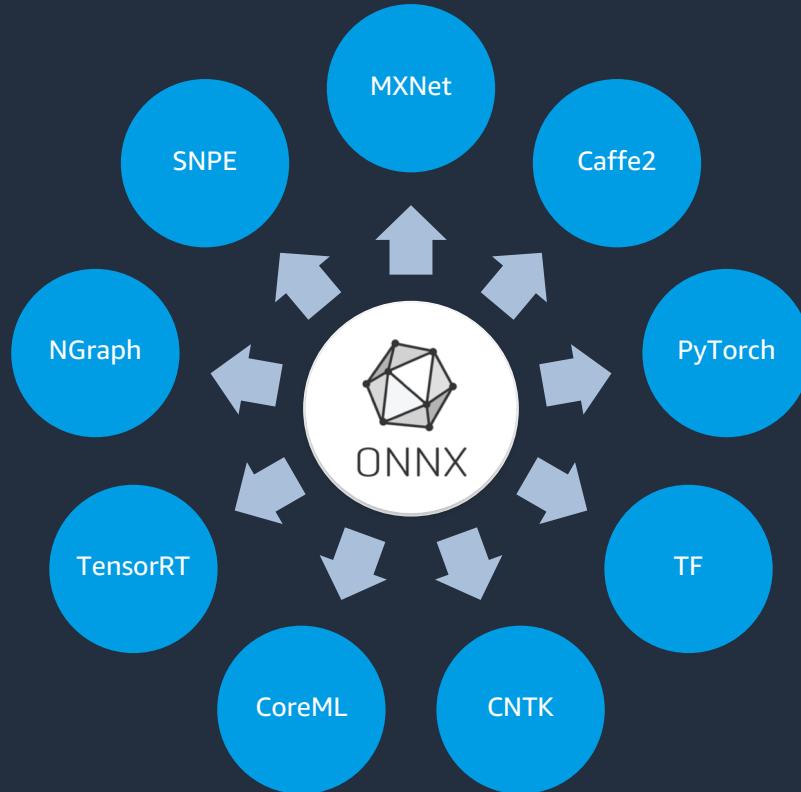
# Open Neural Network eXchange - Overview

Many Frameworks

Many Platforms

ONNX: Common IR

- Open source
- Community driven
- Simple



# Import ONNX model in MXNet– Usage Example

Build and train your model with PyTorch

Load your ONNX model with MXNet

Run inference, fine tune or save as MXNet model.

```
# Import into MXNet (from MXNet 1.2)
sym, arg_params, aux_params = onnx_mxnet.import_model('model.onnx')

# create module
mod = mx.mod.Module(symbol=sym, data_names=['input_0'], label_names=None)
mod.bind(for_training=False, data_shapes=[('input_0', input_img.shape)])
mod.set_params(arg_params=arg_params, aux_params=aux_params)
```

# Export MXNet model to ONNX – Usage

Build and train your model in MXNet

Export trained MXNet model to ONNX format

Import in other framework like cntk, caffe2 for inference

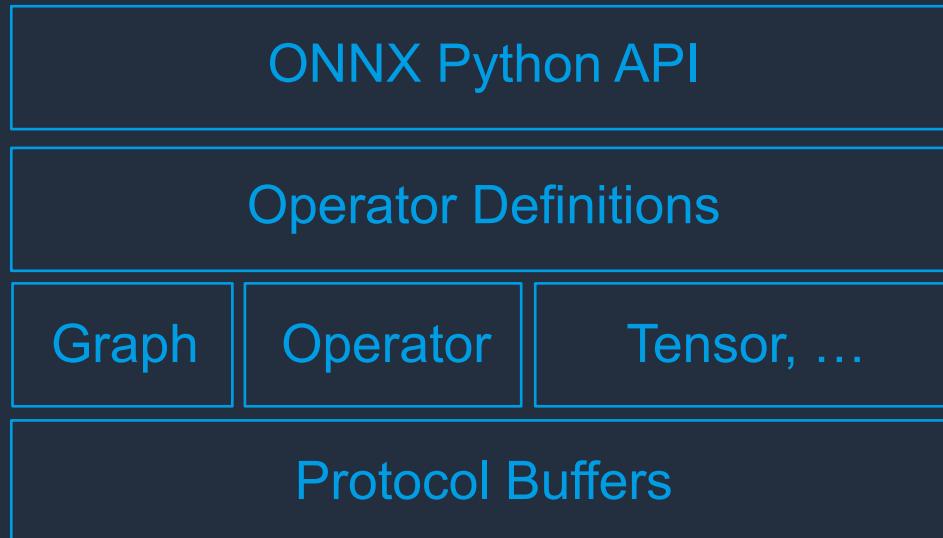
```
# Export MXNet model to ONNX format(from MXNet 1.3)
onnx_file_path = onnx_mxnet.export_model(sym, params, [input_shape],
input_data_type, onnx_file_path)
```

[Back](#)

# ONNX – Internals

## Protocol Buffers:

- Binary compact format
- Statically defined
- APIs for de/serialization
- Cross platform

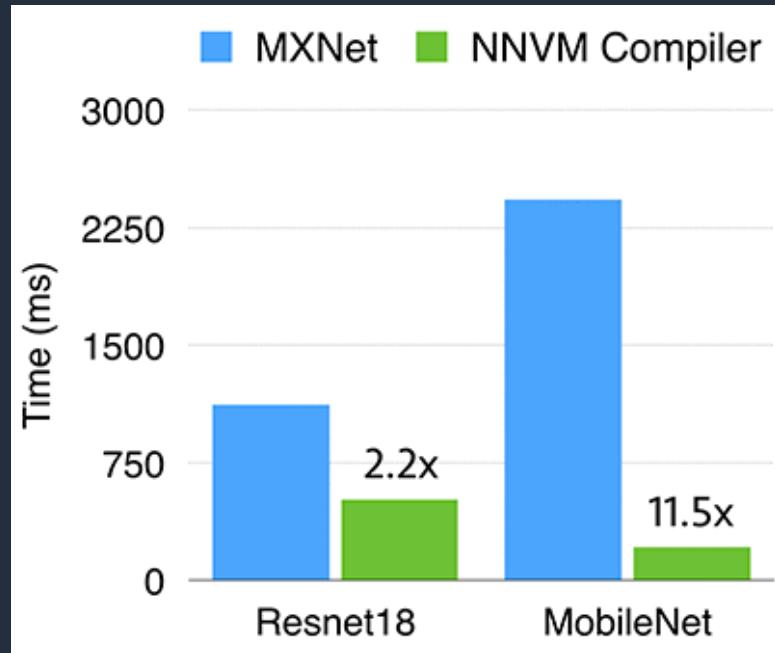
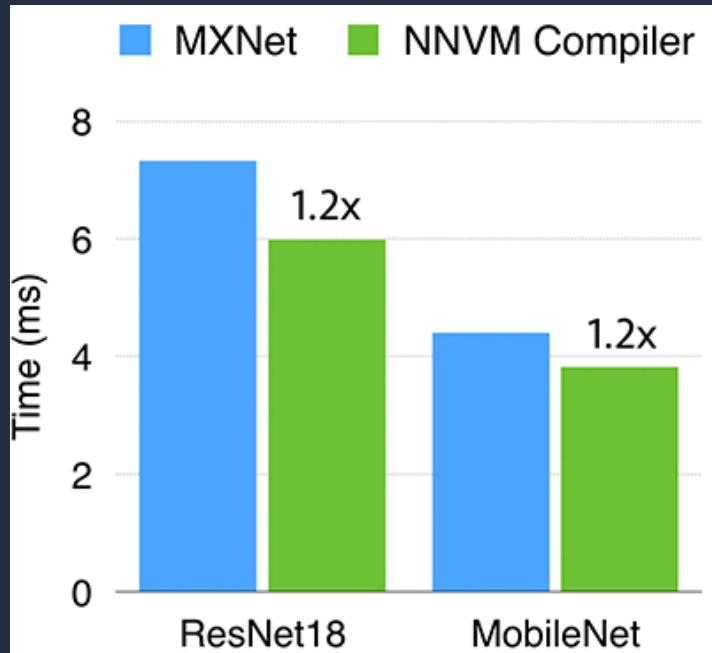


# ONNX – Coverage

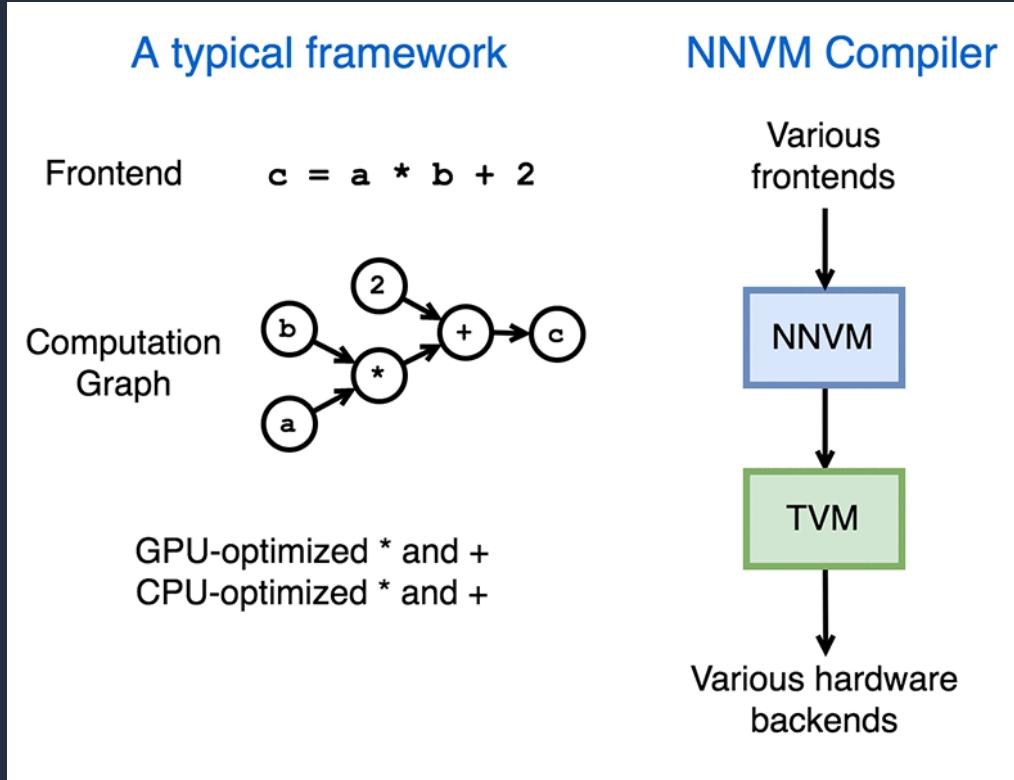
Framework	Export	Import
MXNet	Supported	Supported
Caffe2	Supported	Supported
PyTorch	Supported	Coming...
CNTK	Supported	Supported
Chainer	Supported (external)	N/A
TensorFlow	Supported (external)	Supported (external)
CoreML	Supported (external)	Supported (external)
SciKit-Learn	Supported (external)	N/A

TVM

# Inference Efficiency - TVM



# Portability - TVM



# TensorRT

# Inference Efficiency - TensorRT

Model Name	Relative TRT Speedup	HArdware
Resnet 101	1.99x	Titan V
Resnet 50	1.76x	Titan V
Resnet 18	1.54x	Jetson TX1
cifar_resnext29_16x64d	1.26x	Titan V
cifar_resnet20_v2	1.21x	Titan V
Resnet 18	1.8x	Titan V
Alexnet	1.4x	Titan V

<https://cwiki.apache.org/confluence/display/MXNET/How+to+use+MXNet-TensorRT+integration>



# Keras-MXNet

<https://github.com/awslabs/keras-apache-mxnet> )

# Keras – Apache MXNet

- Deep Learning for Humans
- 2<sup>nd</sup> most popular Deep Learning framework
- Keras users leverage MXNet's great performance

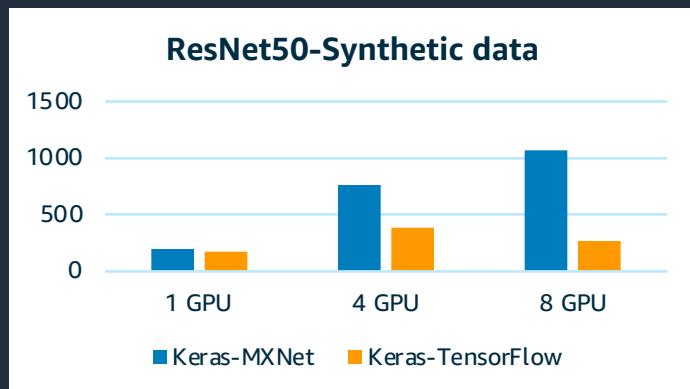
```
from keras.models import Sequential
model = Sequential()
from keras.layers import Dense
model.add(Dense(units=64, activation='relu', input_dim=100))
model.add(Dense(units=10, activation='softmax'))
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5, batch_size=32)
model.train_on_batch(x_batch, y_batch)
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
classes = model.predict(x_test, batch_size=128)
```

```
pip install mxnet-(mkl|cu92)
pip install keras-mxnet
---
~/.keras/keras.json
backend: mxnet
image_data_format: channels_first
---
```

# Keras Benchmarks

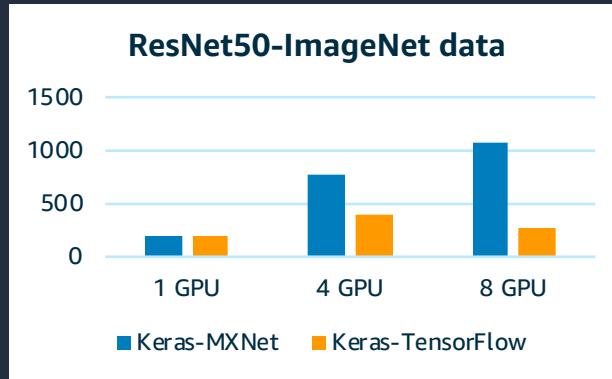
Setup: <https://github.com/awslabs/keras-apache-mxnet/tree/master/benchmark>

	Training	Inference
Instance	P3.8x Large, P3.16x Large	C5.xLarge, C4.8xLarge
Network	ResNet50v1	ResNet50v1
Batch size	32 * Num of GPUs	32
Image size	3*256*256	3*256*256

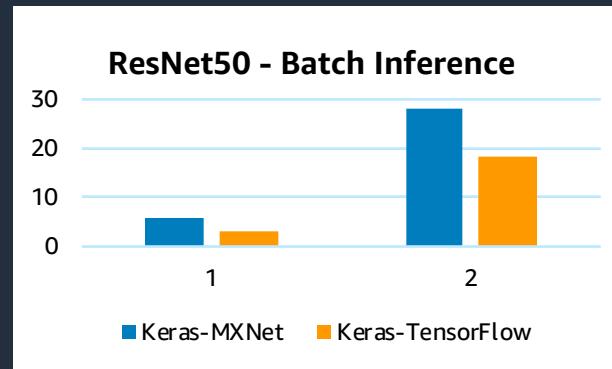


GPUs	Keras-MXNet [ Image/sec ]	Keras-TensorFlow [ Image/sec ]	Speed Up
1	194	184	1.05
4	764	393	1.94
8	1068	261	4.09

# Keras Benchmarks



GPUs	Keras-MXNet	Keras-TensorFlow	Speed Up
1	135	52	2.59
4	536	162	3.30
8	722	211	3.42



# Imperative API



Debuggable



Flexible



Scalable

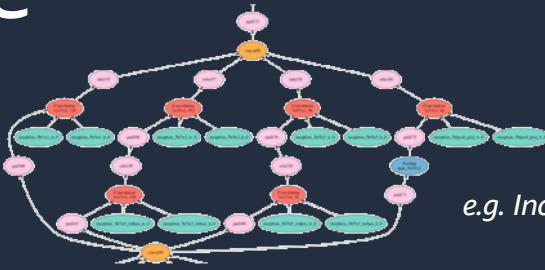
# Symbolic vs Imperative

Symbolic is “define, compile, run”

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])

model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(x_test, y_test))
```



e.g. Inception Stage

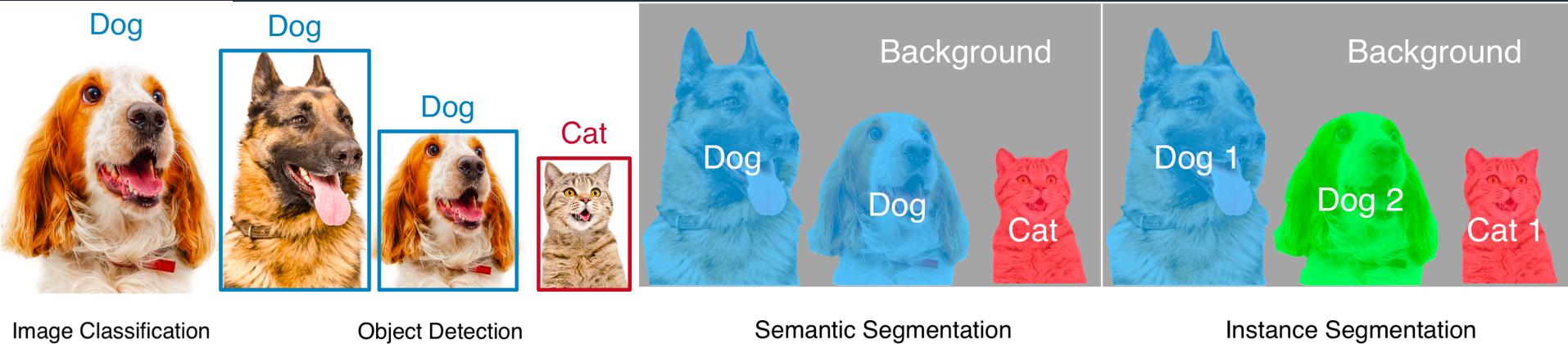
Imperative is “define-by-run”

```
net = nn.Sequential()
with net.name_scope():
    net.add(
        nn.Conv2D(channels=6, kernel_size=5, activation='relu'),
        nn.MaxPool2D(pool_size=2, strides=2),
        nn.Conv2D(channels=16, kernel_size=3, activation='relu'),
        nn.MaxPool2D(pool_size=2, strides=2),
        nn.Flatten(),
        nn.Dense(120, activation="relu"),
        nn.Dense(84, activation="relu"),
        nn.Dense(10)
    )
net.initialize(init=init.Xavier())

for epoch in range(10):
    for data, label in train_data:
        with autograd.record():
            output = net(data)
            loss = softmax_cross_entropy(output, label)
            loss.backward()
            trainer.step(batch_size)
```

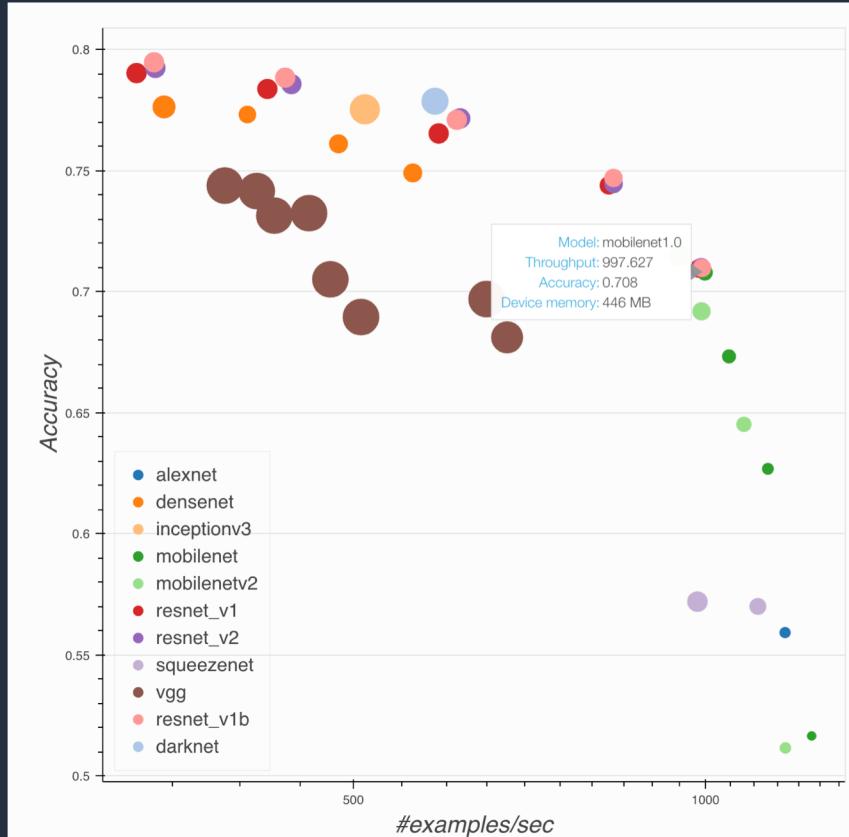
# GluonCV: a Deep Learning Toolkit for Computer Vision

<https://gluon-cv.mxnet.io>

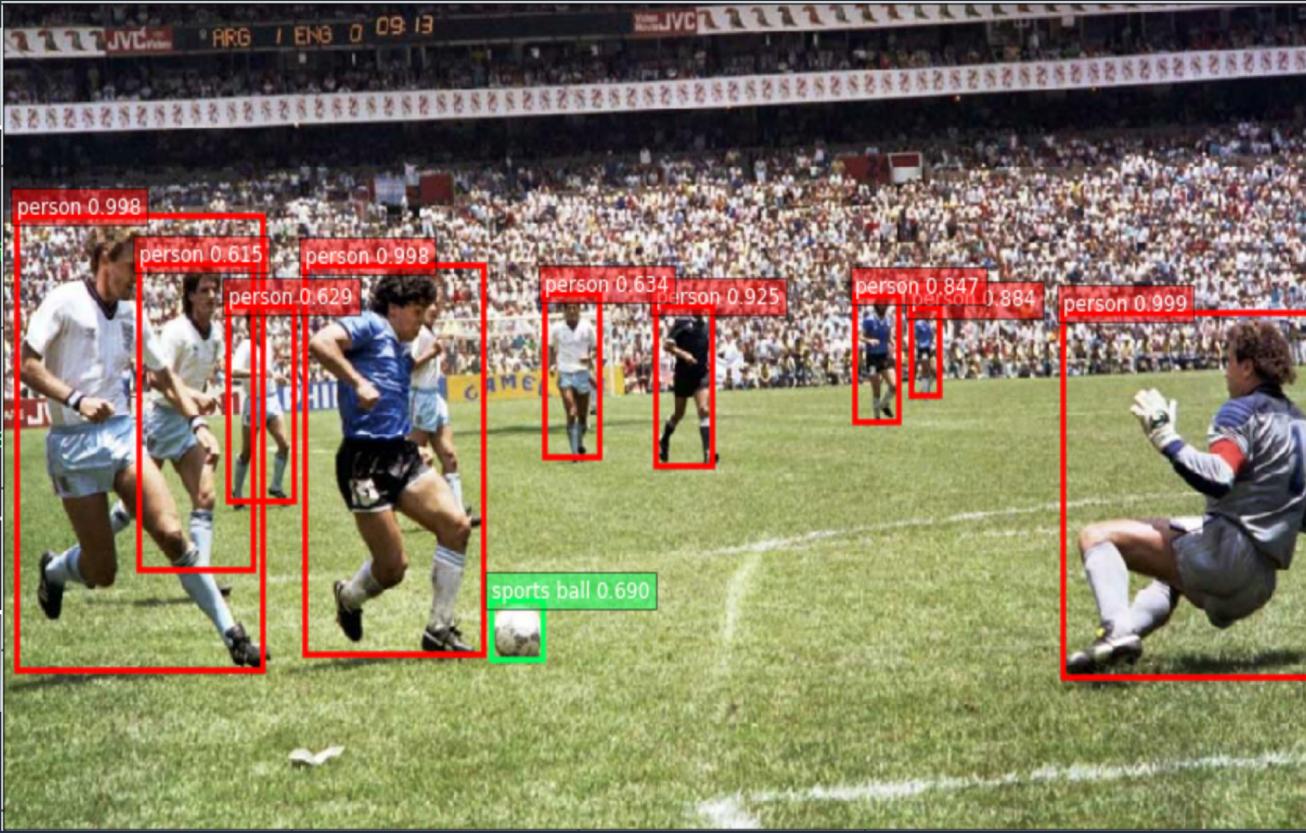


50+ Pre-trained models, with training scripts, datasets, tutorials

# GluonCV: pre-trained models, help to choose



# GluonCV: example code



○ ○

```
x, i  
ctx
```

```
net
```

```
clas
```

```
viz
```

2)

```
tx )
```

lasses )

# GluonNLP: a Deep Learning Toolkit for Natural Language Processing

<https://gluon-nlp.mxnet.io>

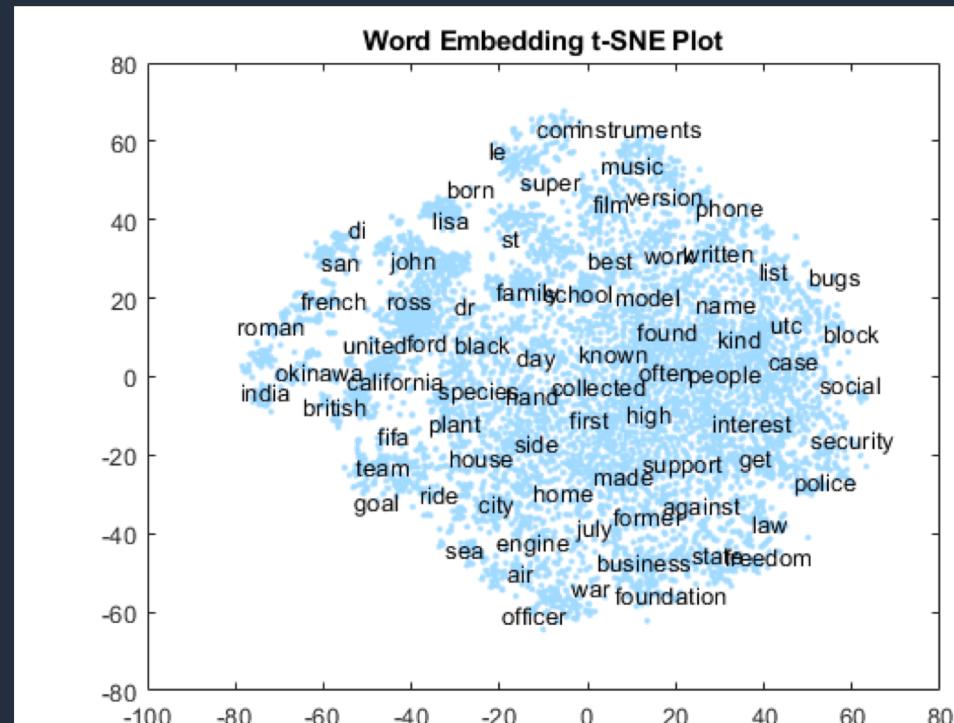
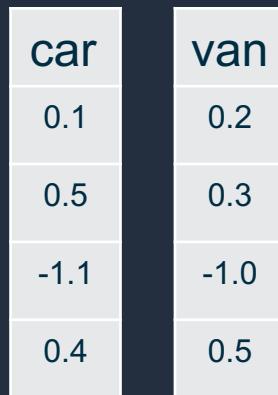


## Features (as of 0.3.2)

- Pre-trained models: over 300 word-embedding
- 5 language models
- Neural Machine Translation (Google NMT, Transformer)
- Flexible data pipeline tools and many public datasets.
- NLP examples such as sentiment analysis.

## Word embedding:

	car
dog	0
apple	0
eat	0
car	1
van	0
leaf	0
the	0
at	0



# Language modeling

Trained to predict the next word ( $P(w_n | w_i \dots w_{n-1})$ ):

- The winner of the 2018 FIFA world cup is  ?

○ ○ ○

```
lm_model, vocab = nlp.model.get_model(name='standard_lstm_lm_200', dataset_name='wikitext-2',
pretrained=True)
data = vocab[tokenizer("This movie is considered the")]

pred, _ = lm_model(mx.nd.array(data, dtype='float32').expand_dims(axis=1))

for p in pred[-1].squeeze().topk(k=20).asnumpy():
    print(vocab.idx_to_token[int(p)])
```

This movie is considered the <?>

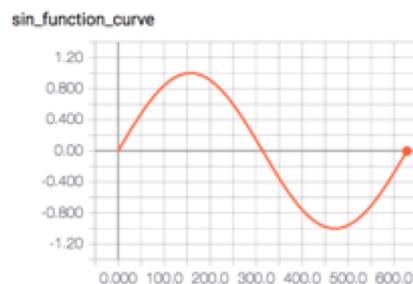
first	greatest	highest
most	main	final
<unk>	second	worst
only	last	sixth
same	name	third
best	largest	way
		primary

# MXBoard: MXNet plugin to TensorBoard

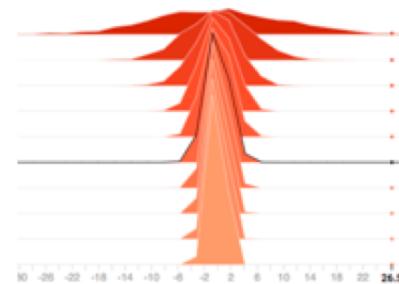
Graph



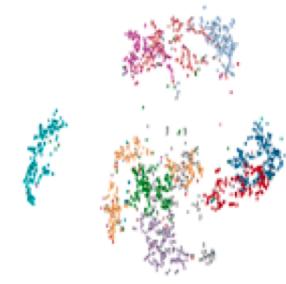
Scalar



Histogram



Embedding



Image



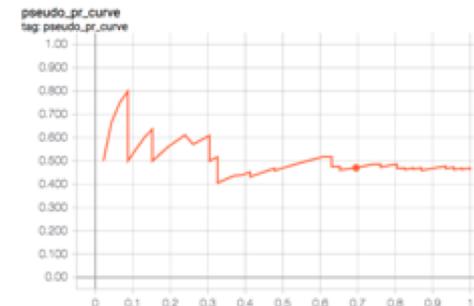
Text

markdown\_table  
tag: markdown\_table

step 0

Hello	MXNet,
This	is
so	awesome!

Precision-Recall Curve



Audio

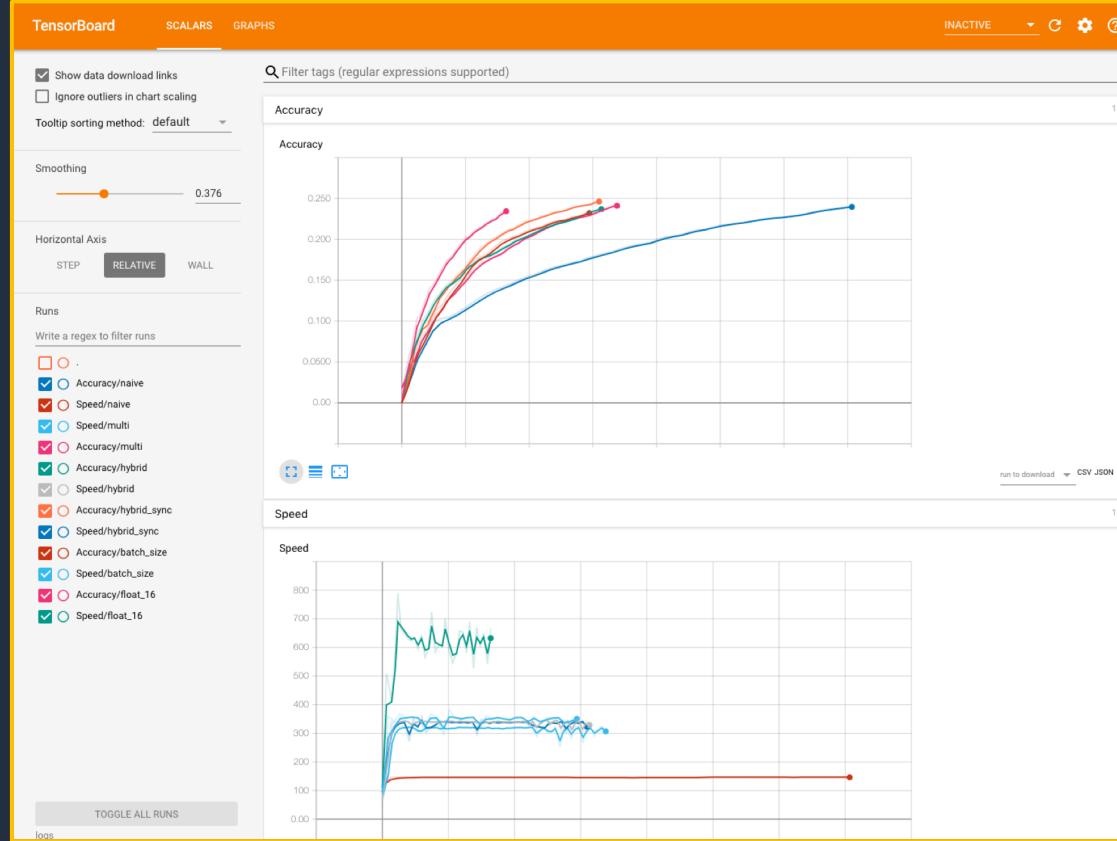
uniform\_audio

uniform\_audio  
step 0

▶ 0:00 / 0:01



# MXBoard



# Deep Learning acceleration



CUDA & CuDNN

```
pip install mxnet-cu92
```

TensorRT

```
pip install mxnet-tensorrt-cu92
```

MKL, MKLML & MKLDNN

```
e.g. pip install mxnet-mkl
```



# Apache MXNet community

# Keeping Up to Date

Medium: <https://medium.com/apache-mxnet>



## A Way to Benchmark Your Deep Learning Framework On-premise

MXNet Makes Us Faster And Stronger!



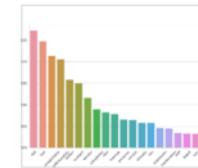
HyungJun Kim

Jul 30



## Let Sentiment Classification Model speak for itself using Grad CAM

Deep learning models are known for being black box models. However, according to our experience, recent developments in explainable methods...



gogamza (Heewon Jeon)

Jul 25

# Keeping Up to Date: Social

YouTube: /apachemxnet



Apache MXNet

295 subscribers

HOME VIDEOS PLAYLISTS CHANNELS

Uploads PLAY ALL

State-of-the-art Learning Rate Schedules 24:39  
Using Learning Rate Schedules in MXNet 19:18  
Choosing the Learning Rate with LR Finder 71 views • 3 weeks ago

49 views • 3 weeks ago

Apache MXNet @ApacheMXNet

Open source, scalable deep-learning engine.

mxnet.incubator.apache.org

Joined January 2017

13 Photos and videos

Choosing the Learning Rate with LR Finder

Twitter: @apachemxnet



Reddit: r/mxnet



reddit r/mxnet

Search Reddit

r/mxnet

Posts

VIEW SORT HOT

Benchmarking Your Deep Learning Framework On-premise medium.com/apache...  
Posted by u/thomasdit 13 minutes ago

Let Sentiment Classification Model speak for itself using Grad CAM medium.com/apache...  
Posted by u/ishitorix11x 4 days ago

GluonNLP – Deep Learning Toolkit for Natural Language Processing medium.com/apache...  
Posted by u/thomasdit 5 days ago

# Community

GitHub: <https://github.com/apache/incubator-mxnet>



Screenshot of the GitHub repository page for apache/incubator-mxnet.

The repository details are as follows:

- Owner: apache
- Name: incubator-mxnet
- Watchers: 1,160
- Stars: 14,694
- Forks: 5,395

The repository has 884 issues and 87 pull requests. The pull requests tab is selected, showing the following open pull requests:

- #11930: Disable flaky test: test\_spatial\_transformer\_with\_type (opened by larroy, 0 of 5 reviews)
- #11928: Generalized reshape\_like operator (opened by sbodenstein, 3 comments)
- #11921: [MXNET-711] Added updated logos to the powered by page (opened by kpmurali, 1 of 1 review)
- #11918: [WIP] Improve gather\_nd and scatter\_nd doc (opened by haojin2, 5 of 6 reviews)

Filters applied: is:pr is:open

Buttons: Labels, Milestones, New pull request

# Community

Discuss Forum: <https://discuss.mxnet.io/>



Topic	Category	Users	Replies	Views	Activity
CNN and invariance to feature translation on the image 0 votes	Discussion	E 🐻	1	4	2m
How to access the output values of a sub-sub custom Block 0 votes	Gluon	A 🐻	1	4	12m
MaxPool2D on odd dimensional layers 0 votes		S 🐻	0	4	5h
Mxnet GPU freezes python 1 vote	Performance	L 🐻 J 🐻	3	29	6h
How to use mxnet gpu verion in kaggle kernel? 0 votes		Y 🐻 Z 🐻	3	20	21h
MxNet 1.2.1–module get_outputs() 0 votes	Discussion	N 🐻	4	38	1d

# Community

Mailing list:

[dev@mxnet.apache.org](mailto:dev@mxnet.apache.org)

[user@mxnet.apache.org](mailto:user@mxnet.apache.org)

# MXNet Customer Momentum



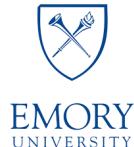
Software Platform Lab  
Seoul National University



BOREALIS AI  
RBC Institute for Research



Carnegie  
Mellon  
University



MediaNET LAB  
Media Network Laboratory  
KAIST  
Korea Advanced Institute of  
Science and Technology  
School of  
Electrical Engineering



# Thank you!