

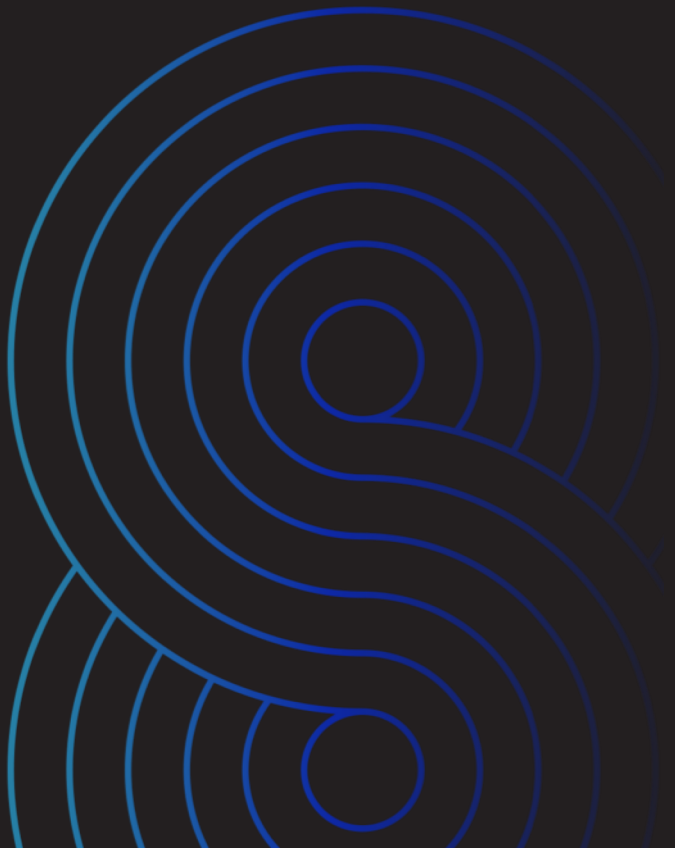
딥러닝계의 블루오션, Apache MXNet 공헌하기

윤석찬, Amazon Web Services

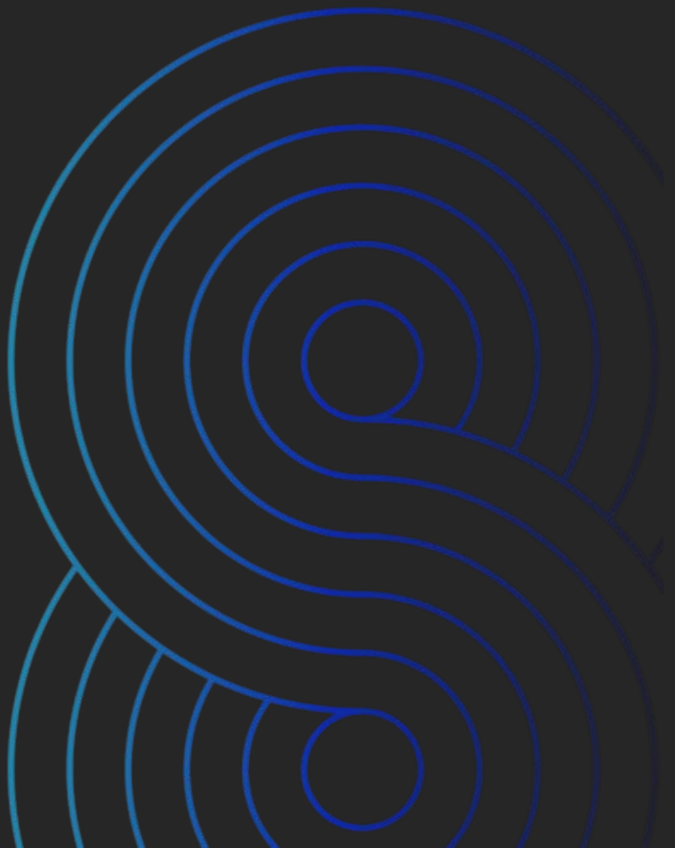
오규삼, 삼성 SDS

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



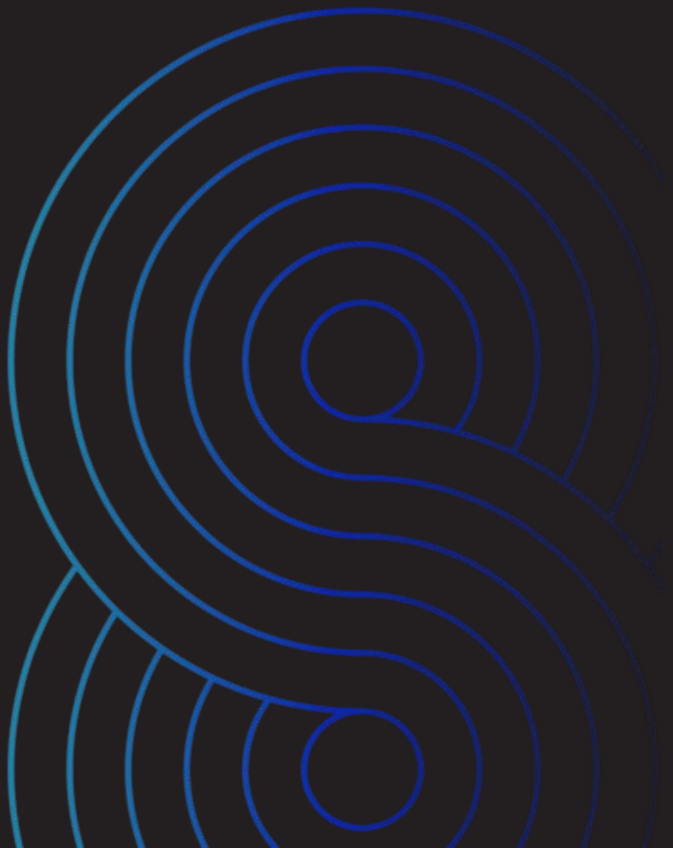
Deep Learning 101	01
Apache MXNet 소개	02
Demo: Amazon SageMaker를 통한 MXNet 학습 예제	03
MXNet 소스 공헌 경험 및 활용 사례	04
MXNet 공헌 및 커뮤니티 참여	05



Deep Learning 101

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



기계 학습이란? Machine Learning



데이터



반복 학습



정답

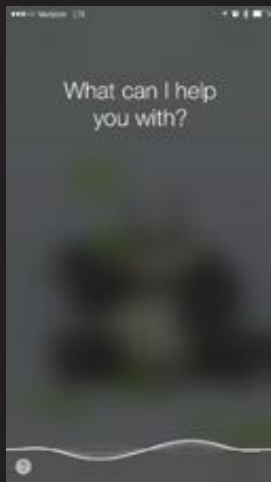
기계가 못하는 것 = 사람이 잘하는 것

SOSCON 2018





이미지 패턴 분석



음성 인식 및
자연어 처리

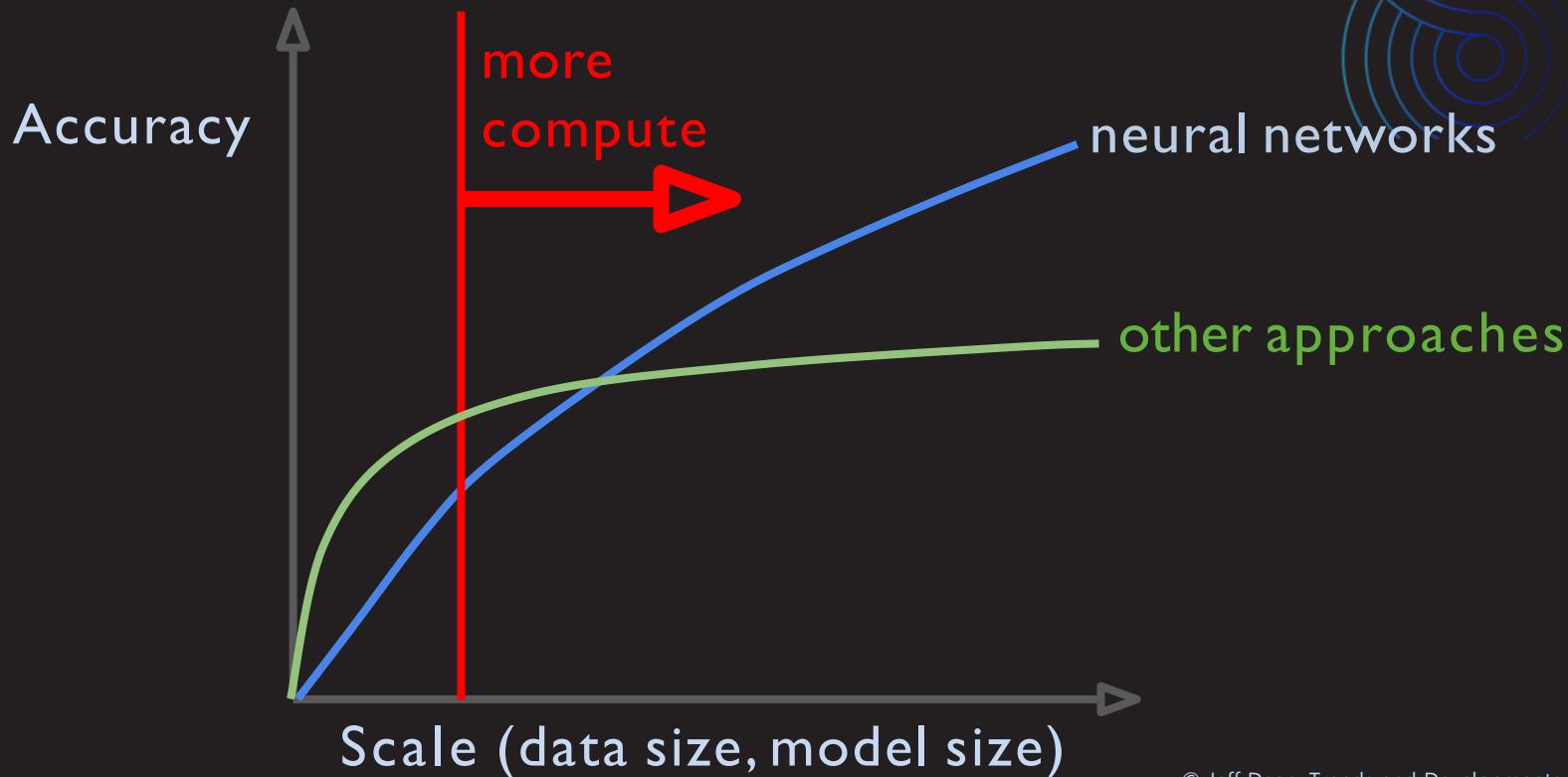


자율 주행 자동차

딥 러닝 - Deep Neural Network

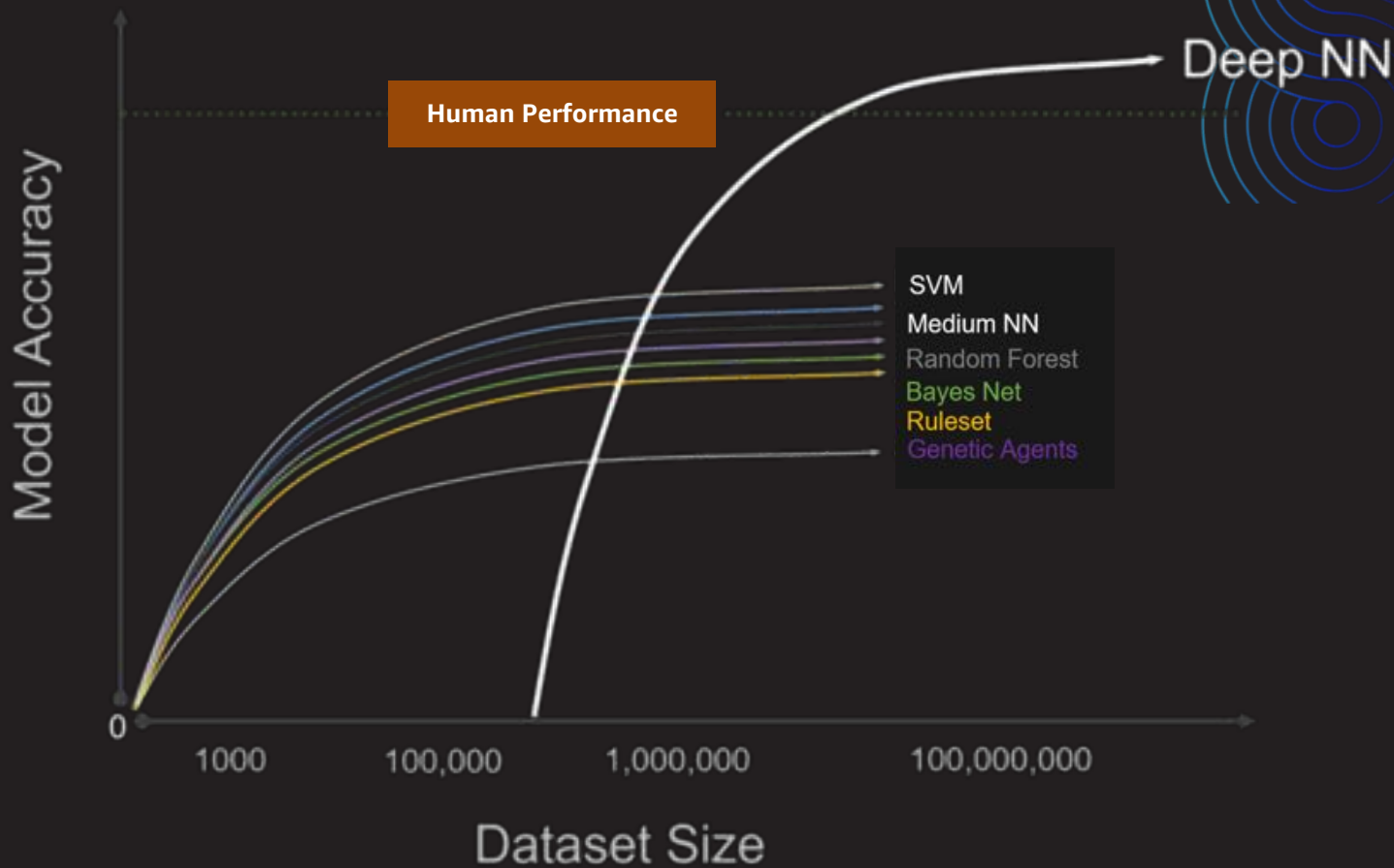
SOSCON 2018

1980s and 1990s



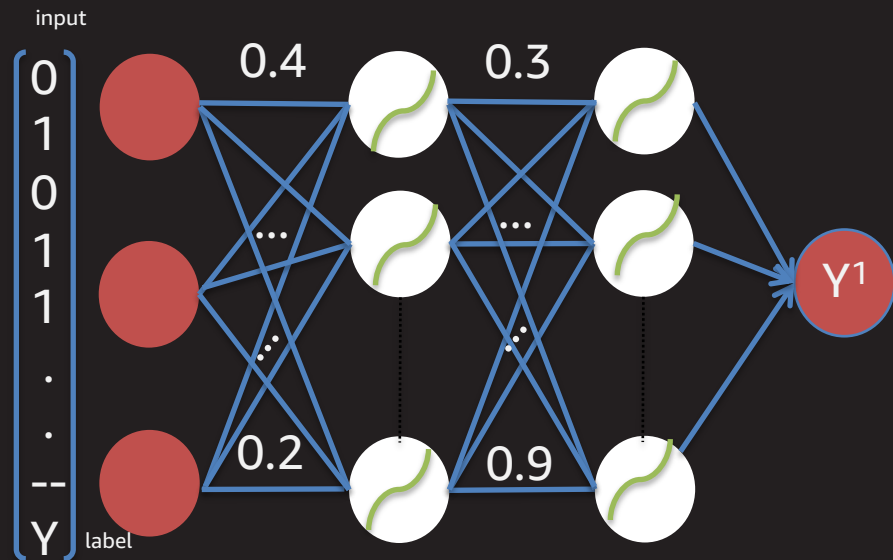
딥 러닝 - Deep Neural Network

SOSCON 2018

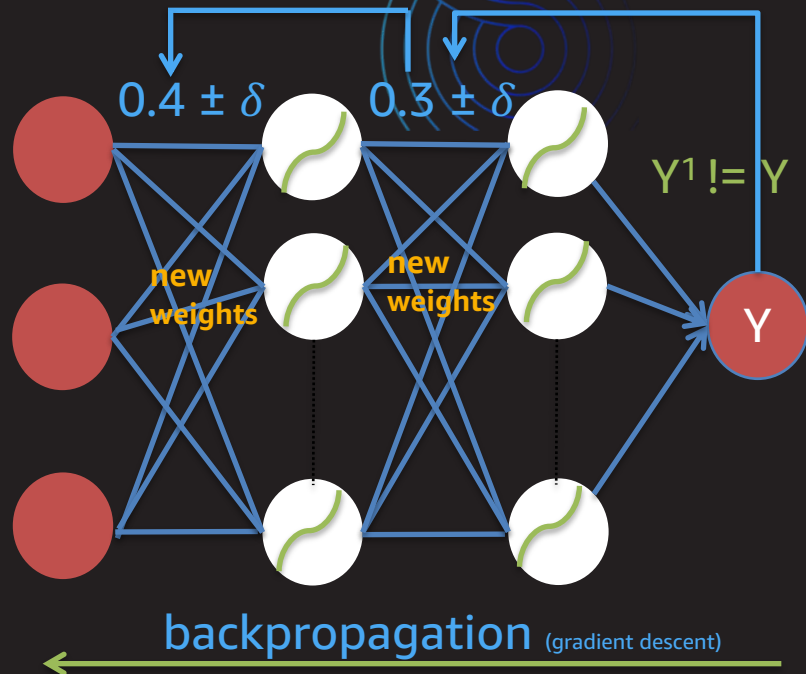


딥 러닝 학습 방법 - 역전파

SOSCON 2018

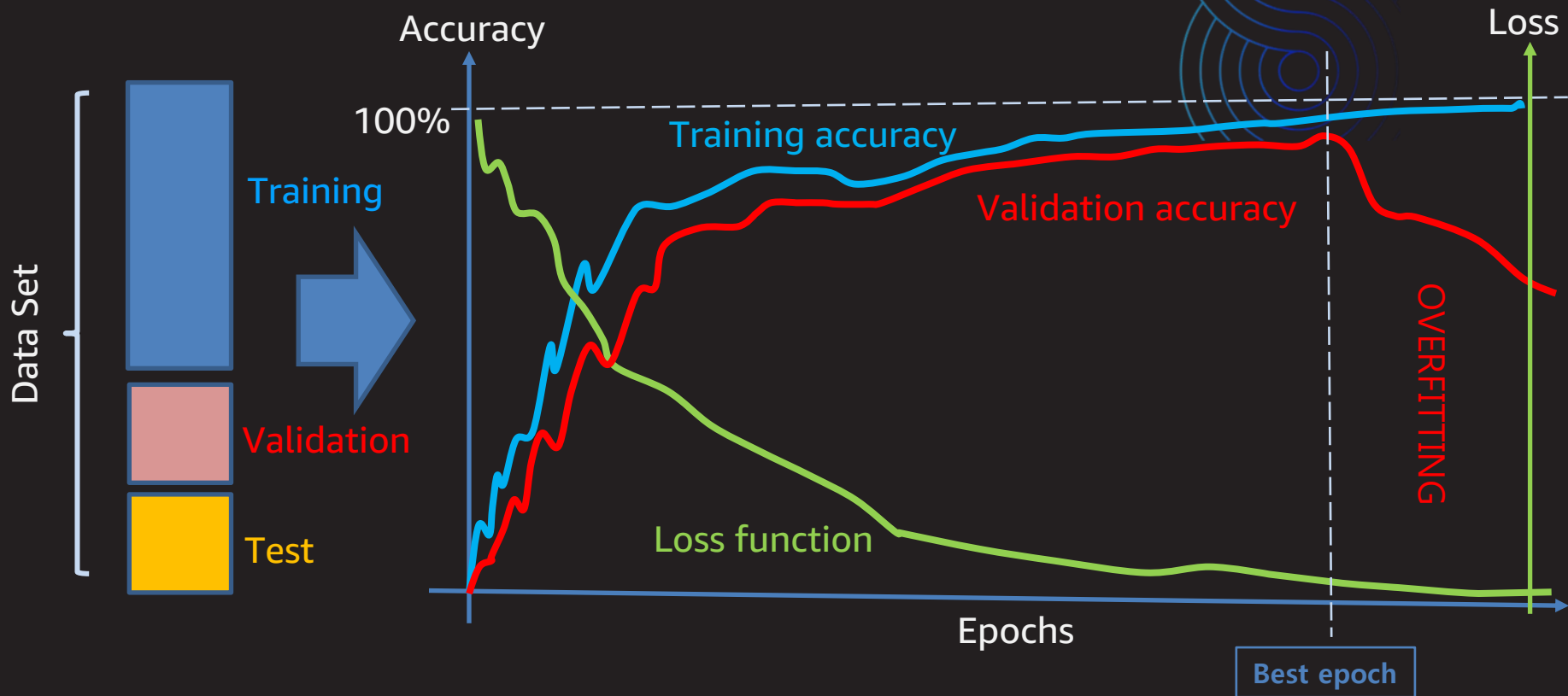


Hyper parameters {
Batch size
Learning rate
Number of epochs



딥 러닝 학습 방법 - 조기 종료

SOSCON 2018



딥 러닝 학습 방법 - 오픈 소스 활용

SOSCON 2018

High-Level
API



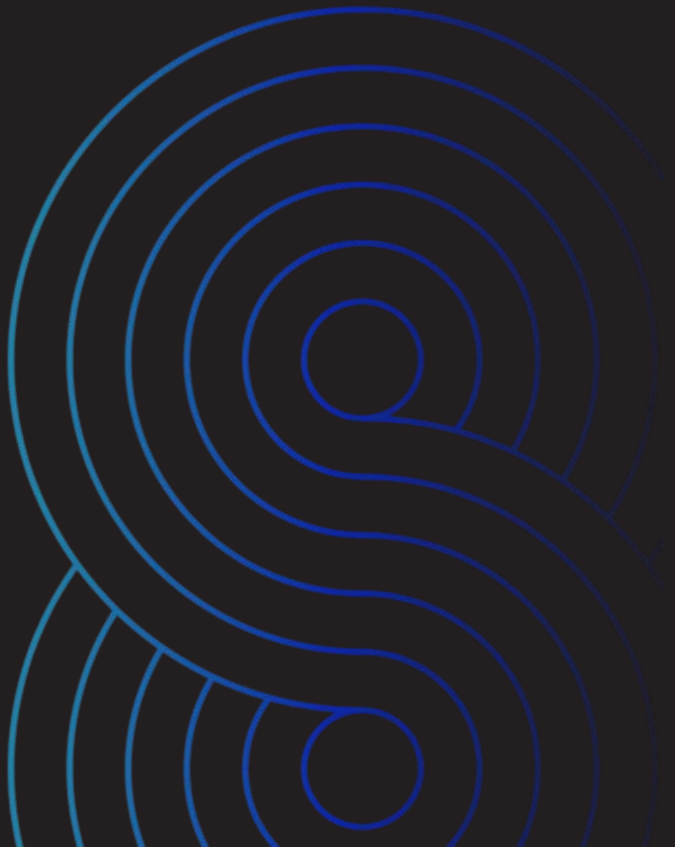
Frame-
work



Apache MXNet 소개

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

MXNet - Deep Learning Framework of Choice at AWS

By Werner Vogels on 22 November 2016 09:00 AM | [Permalink](#) | [Comments \(1\)](#)



Machine learning is playing an increasingly important role in many areas of our businesses and our lives and is being employed in a range of computing tasks where programming explicit algorithms is infeasible.

At Amazon, machine learning has been key to many of our business processes, from recommendations to fraud detection, from inventory levels to book classification to abusive review detection. And there are many more application areas where we use machine learning extensively: search, autonomous drones



Contact Info

Werner Vogels
CTO - Amazon.com

werner@allthingsdistributed.com

Other places

Follow werner on twitter if you want to

아마존이 선택한 딥러닝 프레임워크

SOSCON 2018



Apache 오픈 소스 프로젝트

SOSCON 2018

mxnet Install Glueon API Docs Community Versions(master) Search

Apache MXNet (Incubating)

A flexible and efficient library for deep learning.

[Install](#) [Learn More](#)

APACHE INCUBATOR

MXNet 1.3.0 Released

We're excited to announce the release of MXNet 1.3.0! This release includes Clojure bindings, Glueon package enhancements, ONNX export, TensorRT integration, and more!

[Learn More](#)

A 60-minute Glueon Crash Course

Check out our quick overview of how to use Glueon, the imperative interface of MXNet.

[Learn More](#)

Get the latest news from MXNet blogs on Medium

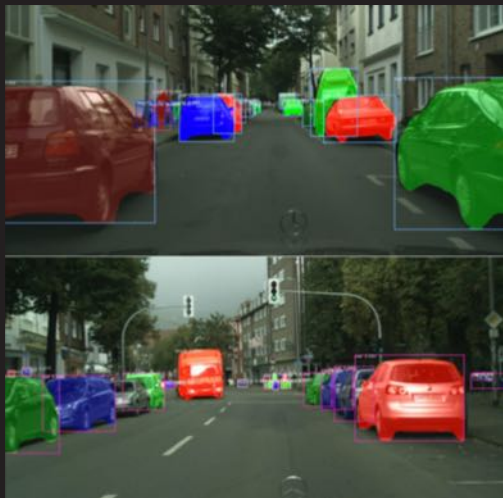
Contributors and customers of MXNet post about how-tos, guides, examples, and stories about their MXNet experiences.

[Learn More](#)

<https://mxnet.apache.org/>

MXNet 활용 사례 - 스타트업도 할 수 있다! SOSCON 2018

자율 주행을 위한 객체 탐지



<https://github.com/TuSimple/mx-maskrcnn>

SW 개발 및 Level 4 트럭 상용화 진행 중

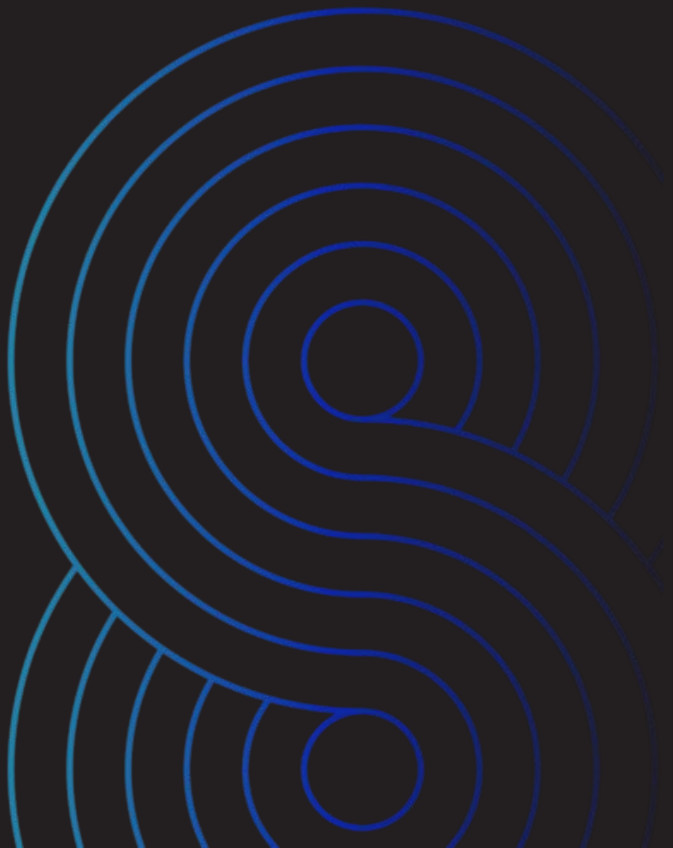


<http://www.tusimple.com>

MXNet의 주요 특징

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



1. 유연한 프로그래밍 모델

SOSCON 2018

명령형 NDArray API

```
>>> import mxnet as mx
>>> a = mx.nd.zeros((100, 50))
>>> b = mx.nd.ones((100, 50))
>>> c = a + b
>>> c += 1
>>> print(c)
```

선언형 Symbolic Executor

```
>>> import mxnet as mx
>>> net = mx.symbol.Variable('data')
>>> net = mx.symbol.FullyConnected(data=net, num_hidden=128)
>>> net = mx.symbol.SoftmaxOutput(data=net)
>>> texec = mx.module.Module(net)
>>> texec.forward(data=c)
>>> texec.backward()
```

명령형 NDArray 객체가
그래프의 입력으로 사용됨

2. 다양한 언어 및 플랫폼 지원

SOSCON 2018

- 지원 언어 - Python, Scala, R, Julia, Perl, C++
- 운영체제 - Linux, Mac, Windows, Cloud (AWS), Devices (Android)
- 컴퓨팅 - CPU, GPU
- 패키지 - PiP, Docker

Installing MXNet

Indicate your preferred configuration. Then, follow the

v1.3.0 ▾

Linux

MacOS

Windows

Cloud

Devices

Python

Scala

R

Julia

Perl

Cpp

CPU

GPU

Pip

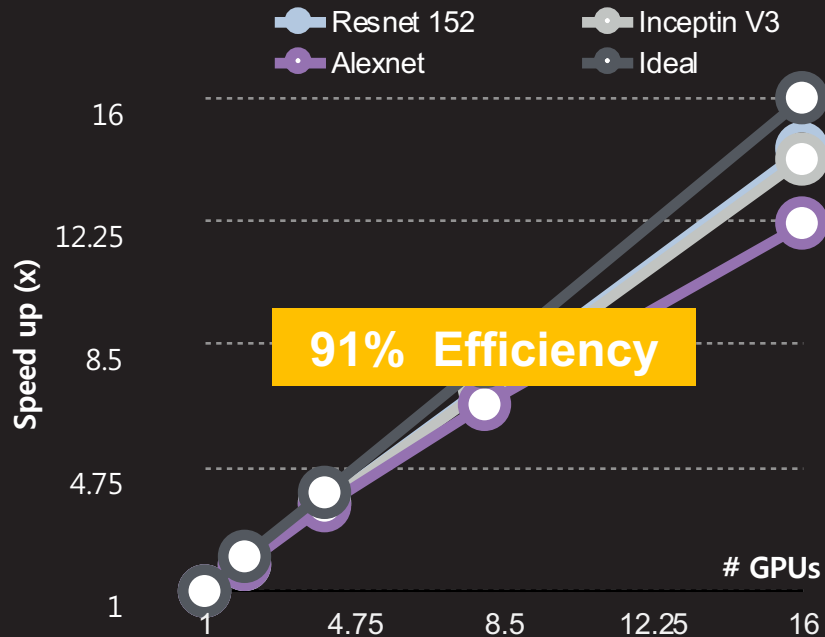
Docker

Build from Source

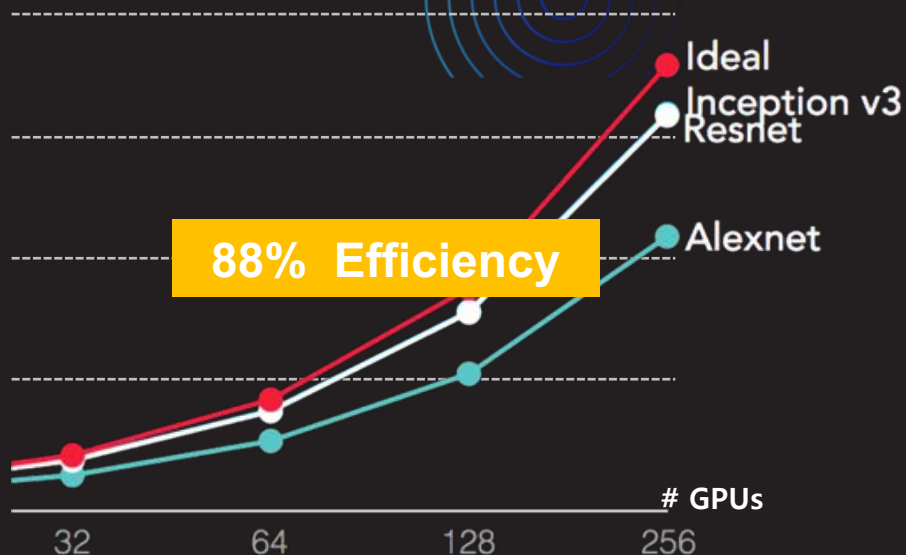
```
$ pip install mxnet
```

3. 멀티 GPU환경 최적화

SOSCON 2018



- P2.16xlarge (8 Nvidia Tesla K80 - 16 GPUs)
- Synchronous SGD (Stochastic Gradient Descent)



- 16x P2.16xlarge by AWS CloudFormation
- Mounted on Amazon EFS

3. 멀티 GPU환경 최적화

SOSCON 2018

- 학습에 사용할 GPU 목록을 명시하는 것으로 멀티 GPU 학습 가능
- MXNet은 자동으로 학습 데이터를 1/n로 나눠서 GPU 학습을 함

```
## train data
num_gpus = 4
gpus = [mx.gpu(i) for i in range(num_gpus)]
model = mx.model.FeedForward(
    ctx = gpus,
    symbol = softmax,
    num_round = 20,
    learning_rate = 0.01,
    momentum = 0.9,
    wd = 0.00001)
model.fit(X = train, eval_data = val,
        batch_end_callback = mx.callback.Speedometer(batch_size=
batch_size))
```

4. 모바일 환경으로 모델 배포

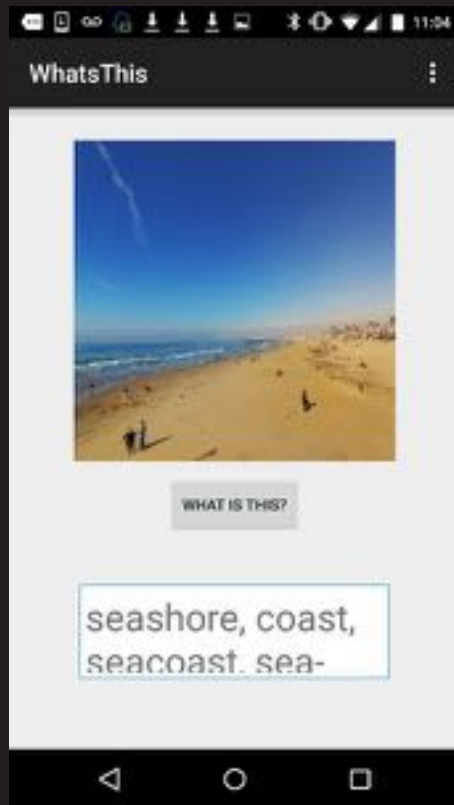
SOSCON 2018

Amalgamation

추론에 필요한 딥러닝 모델과 관련 파일을 하나의 .cc 파일로 변환
컴파일 후, 모바일 디바이스에서 활용

Apple Core ML 지원

Core ML은 Apple의 앱의 기계 학습 모델을 지원하는 Framework
MXNet 모델을 Core ML 형식으로 전환하는 도구





<http://gluon.mxnet.io>

주요 기능 소개

- 딥러닝 모델 개발을 단순하게 할 수 있음
- 간단하고 이해하기 쉬운 코드 제공
- 유연한 Neural Network 구현이 가능
- 빠른 성능 제공 (학습 속도 저하 없음)
- Microsoft 협업을 통한 개발 및 프레임 워크 추가 지원 예정

공헌 방법

<https://github.com/zackchase/mxnet-the-straight-dope>

Gluon Model Zoo란?

SOSCON 2018

미리 정의된 Neural Network 모델을 모듈로 제공
다양한 학습 모델 지원

- AlexNet, DenseNet, Inception V3, ResNet V1, ResNet V2, SqueezeNet, VGG, MobileNet

신경망 아키텍처만 이용하기

```
from mxnet.gluon.model_zoo import vision
```

```
resnet18 = vision.resnet18_v1()  
alexnet = vision.alexnet()
```

기존 학습된 모델 이용하기

```
from mxnet.gluon.model_zoo import vision
```

```
resnet18 = vision.resnet18_v1(pretrained=True)  
alexnet = vision.alexnet(pretrained=True)
```


Keras2 백-엔드 지원

SOSCON 2018

Keras 코드 그대로 활용 가능

```
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(x_train, y_train,
          batch_size=batch_size, epochs=15, validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test, batch_size=batch_size)
```

간단한 멀티 GPU 학습

```
gpu_list = ["gpu(0)", "gpu(1)", "gpu(2)", "gpu(3)"]
model.compile(loss='categorical_crossentropy',
              optimizer=opt, metrics=['accuracy'], context=gpu_list)
```

Demo: Amazon SageMaker를 통한 MXNet 텍스트 감성 분석 모델 학습

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

Amazon SageMaker X

대시보드

노트북 인스턴스

작업

리소스

모델

엔드포인트 구성

엔드포인트

Amazon SageMaker > 대시보드

개요



노트북 인스턴스

노트북에서 AWS 데이터를 탐색하고 알고리즘을 사용하여 교육 작업을 통해 모델을 만듭니다.

[노트북 인스턴스 만들기](#)

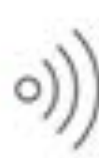

작업

데스크에서 또는 원격으로 교육 작업을 추적합니다. 고성능 AWS 알고리즘을 활용합니다.

[작업 보기](#)


모델

작업 출력에서 호스팅할 모델을 만들거나 외부에서 훈련된 모델을 Amazon SageMaker로 가져옵니다.

[모델 보기](#)


엔드포인트

개발자가 프로덕션에 사용할 엔드포인트 배포 엔드포인트를 통한 API 엔드포인트 변환.

[엔드포인트 보기](#)

최근 활동

다음 기간 동안의 최근 활동

[지난 30일](#)

노트북 인스턴스

작업

모델

엔드포인트

MXNet 소스 공헌 경험 및 활용 사례

오규삼, 삼성 SDS

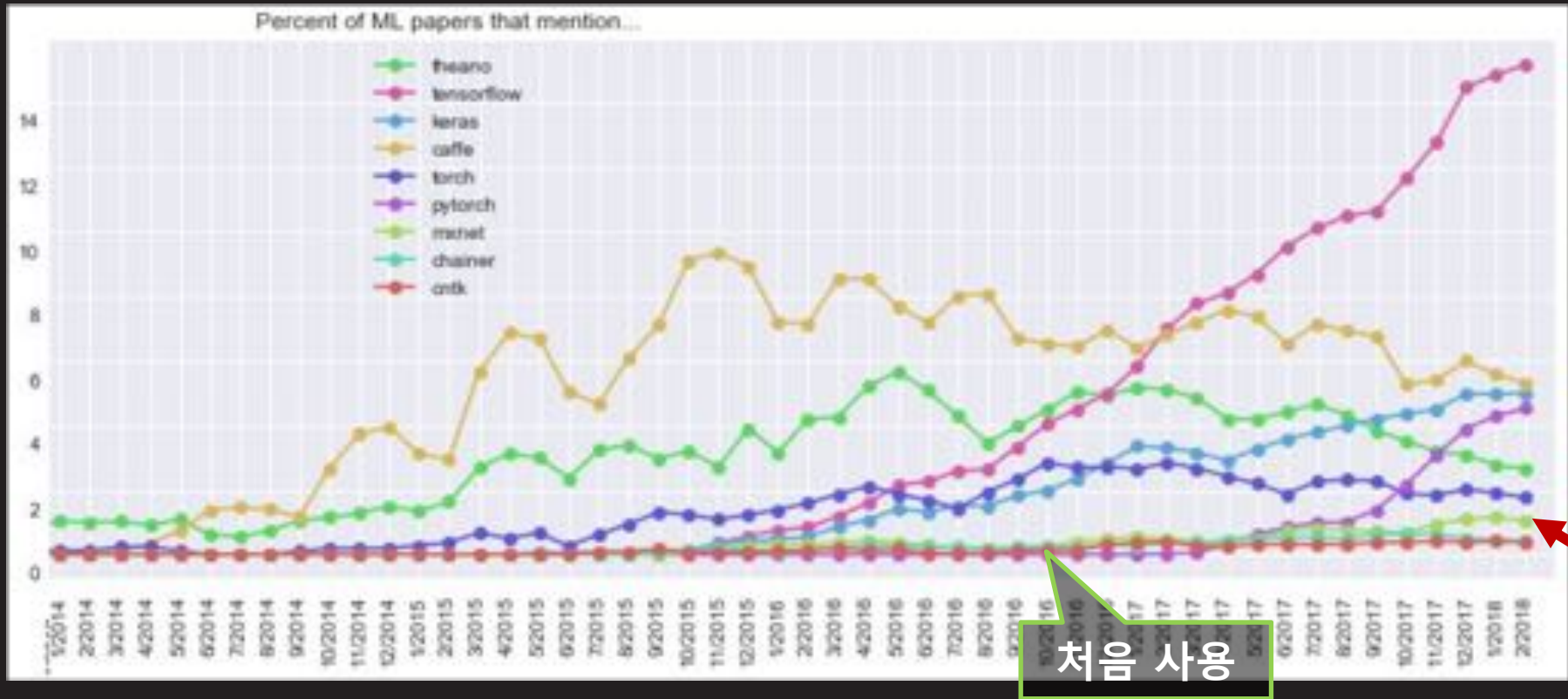
SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

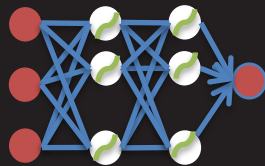


Apache MXNet - arXiv 인용 지수

SOSCON 2018



<https://twitter.com/karpathy/status/972295865187512320>



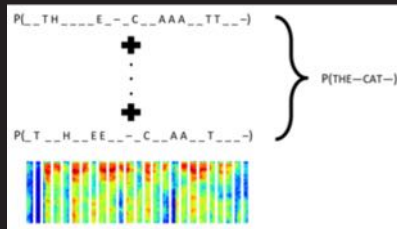
안녕하세요

Why MXNet

SOSCON 2018

CTC Loss

(Connectionist Temporal Classification)



Efficient Memory

vs. Tensorflow : Deep Bi-directional RNN + α layers

Multi + Distributed GPUs

vs. caffe : Distributed CTC





많은 부족한 점 발견 → 소스 공헌 목록

- 초기 진입 시, 매뉴얼 작성
- Deep Network의 시각화 지원
Distributed 환경에서 machine 제어 어려움 (deploy, stop 등..)
- Data Load 시 GPU가 동일해야 가능
- 돌아가는 예제 코드가 거의 없음
(특히 RNN 계열)

.....

코드 공헌 사례 - Deep Speech 2

SOSCON 2018

https://github.com/apache/incubator-mxnet/tree/master/example/speech_recognition

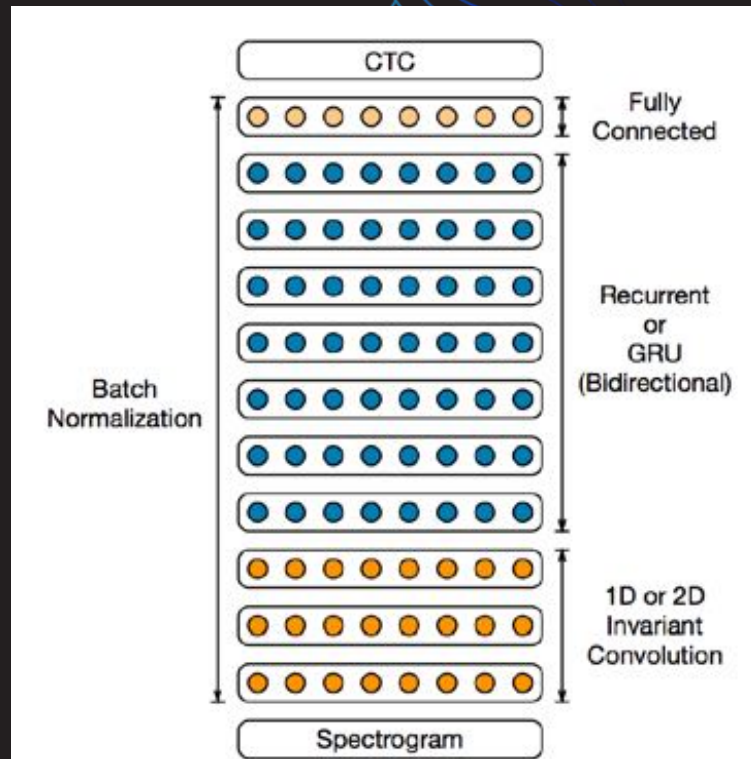
Deep Speech 2: End-to-End Speech Recognition in English and Mandarin

Baidu Research – Silicon Valley AI Lab*

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, Zhenyao Zhu

Abstract

We show that an end-to-end deep learning approach can be used to recognize either English or Mandarin Chinese speech—two vastly different languages. Because it replaces entire pipelines of hand-engineered components with neural networks, end-to-end learning allows us to handle a diverse variety of speech including noisy environments, accents and different languages. Key to our approach is our application of HPC techniques, resulting in a 7x speedup over our previous system [26]. Because of this efficiency, experiments that previously took weeks now run in days. This enables us to iterate more quickly to identify superior architectures and algorithms. As a result, in several cases, our system is competitive with the transcription of human workers when benchmarked on standard datasets. Finally, using a technique called Batch Dispatch with GPUs in the data center, we show that our system can be inexpensively deployed in an online setting, delivering low latency when serving users at scale.



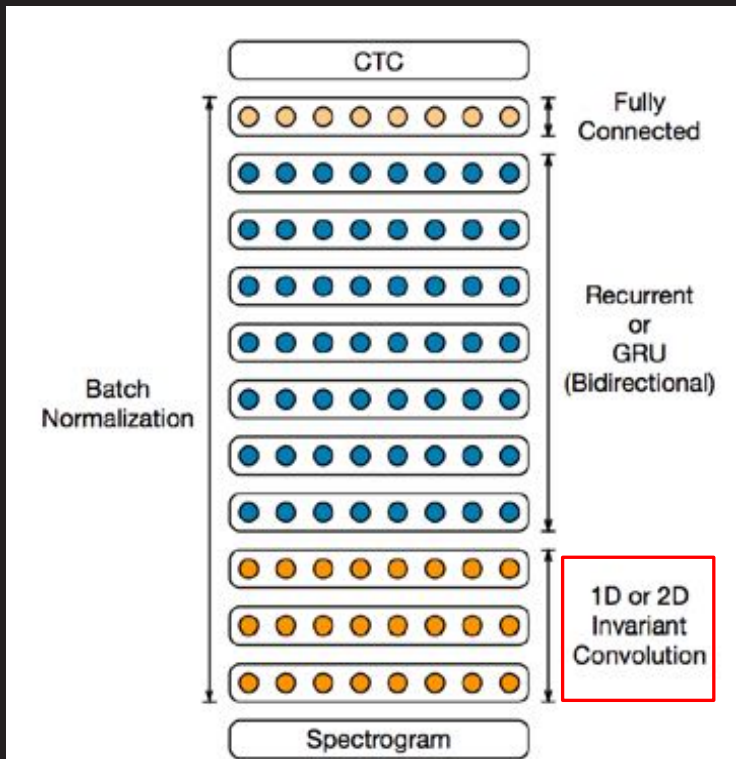
<https://arxiv.org/abs/1512.02595>

https://github.com/apache/incubator-mxnet/tree/master/example/speech_recognition

코드 공헌 사례 - Deep Speech 2

SOSCON 2018

arch_deepspeech.py : 80



```
net = conv(net=net,
           channels=channel_num,
           filter_dimension=conv_layer1_filter_dim,
           stride=conv_layer1_stride,
           no_bias=is_batchnorm
           )
if is_batchnorm:
    # batch norm normalizes axis 1
    net = batchnorm(net)

net = conv(net=net,
           channels=channel_num,
           filter_dimension=conv_layer2_filter_dim,
           stride=conv_layer2_stride,
           no_bias=is_batchnorm
           )
if is_batchnorm:
    # batch norm normalizes axis 1
    net = batchnorm(net)
```

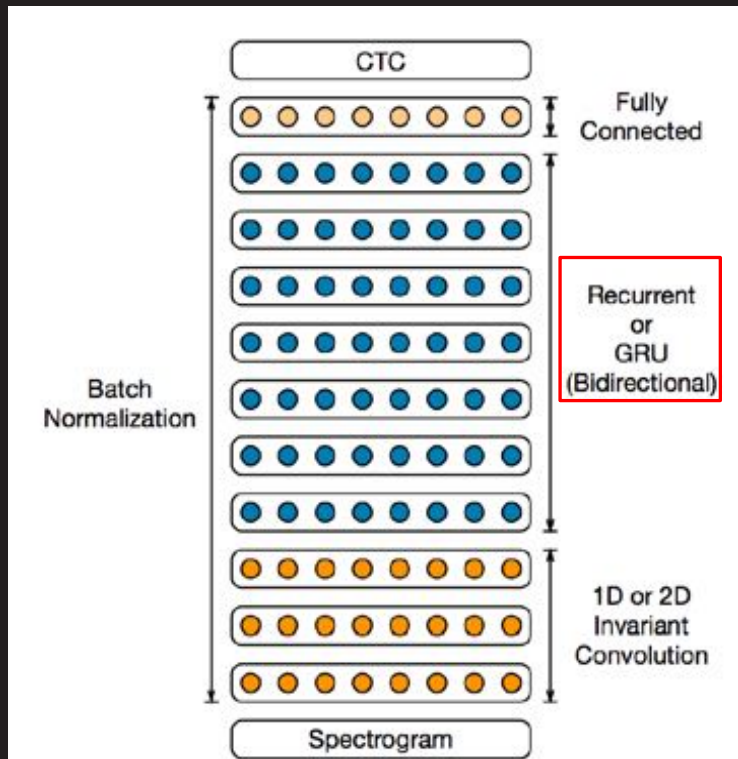
<https://arxiv.org/abs/1512.02595>

https://github.com/apache/incubator-mxnet/tree/master/example/speech_recognition

코드 공헌 사례 - Deep Speech 2

SOSCON 2018

arch_deepspeech.py : 102



```
if rnn_type == "bilstm":
    net = bi_lstm_unroll(net=net,
                          seq_len=seq_len_after_conv_layer2,
                          num_hidden_lstm_list=num_hidden_rnn_list,
                          num_lstm_layer=num_rnn_layer,
                          dropout=0.,
                          is_batchnorm=is_batchnorm)

elif rnn_type == "gru":
    net = gru_unroll(net=net,
                     seq_len=seq_len_after_conv_layer2,
                     num_hidden_gru_list=num_hidden_rnn_list,
                     num_gru_layer=num_rnn_layer,
                     dropout=0.,
                     is_batchnorm=is_batchnorm)

elif rnn_type == "bigru":
    net = bi_gru_unroll(net=net,
                        seq_len=seq_len_after_conv_layer2,
                        num_hidden_gru_list=num_hidden_rnn_list,
                        num_gru_layer=num_rnn_layer,
                        dropout=0.,
                        is_batchnorm=is_batchnorm)

else:
    raise Exception('rnn_type should be one of the followings, bilstm,gru,bigr')
```

코드 공헌 사례 - Capsule Network

SOSCON 2018

<https://github.com/apache/incubator-mxnet/tree/master/example/capsnet>

Dynamic Routing Between Capsules

Sara Sabour

Nicholas Frosst

Geoffrey E. Hinton
Google Brain
Toronto

{sasabour, frosst, geoffhinton}@google.com

Abstract

A capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of object such as an object or an object part. We use the length of the activity vector to represent the probability that the entity exists and its orientation to represent the instantiation parameters. Active capsules at one level make predictions, via transformation matrices, for the instantiation parameters of higher-level capsules. When multiple predictions agree, a higher level capsule becomes active. We show that a discriminatively trained, multi-layer capsule system achieves state-of-the-art performance on MNIST and is considerably better than a convolutional net at recognizing highly overlapping digits. To achieve these results we use an iterative routing-by-agreement mechanism: A lower-level capsule prefers to send its output to higher level capsules whose activity vectors have a big scalar product with the prediction coming from the lower-level capsule.



12/17(구현 당시) – 타 플랫폼 대비 가장 빠른 train 속도, 가장 낮은 에러율(MNIST, 0.29%)

<https://arxiv.org/abs/1710.09829>

<https://www.kdnuggets.com/2017/11/capsule-networks-shaking-up-ai.html>

MXNet 공헌 및 커뮤니티 참여 방법

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



소스 코드 공헌 - 정말 다양한 Issue들

SOSCON 2018

The screenshot shows the GitHub interface for the `apache/incubator-mxnet` repository. At the top, it displays repository statistics: 1,172 Watchers, 15,399 Stars, and 5,560 Forks. Below this, navigation tabs for Code, Issues (787), Pull requests (100), Projects (8), and Insights are visible. A search bar with the text "is:issue is:open" and buttons for "Filters", "Labels", "Milestones", and a green "New Issue" button are present. The main section lists 787 open issues, with a summary of 5,900 closed issues. The issues are sorted by default. The first four issues listed are:

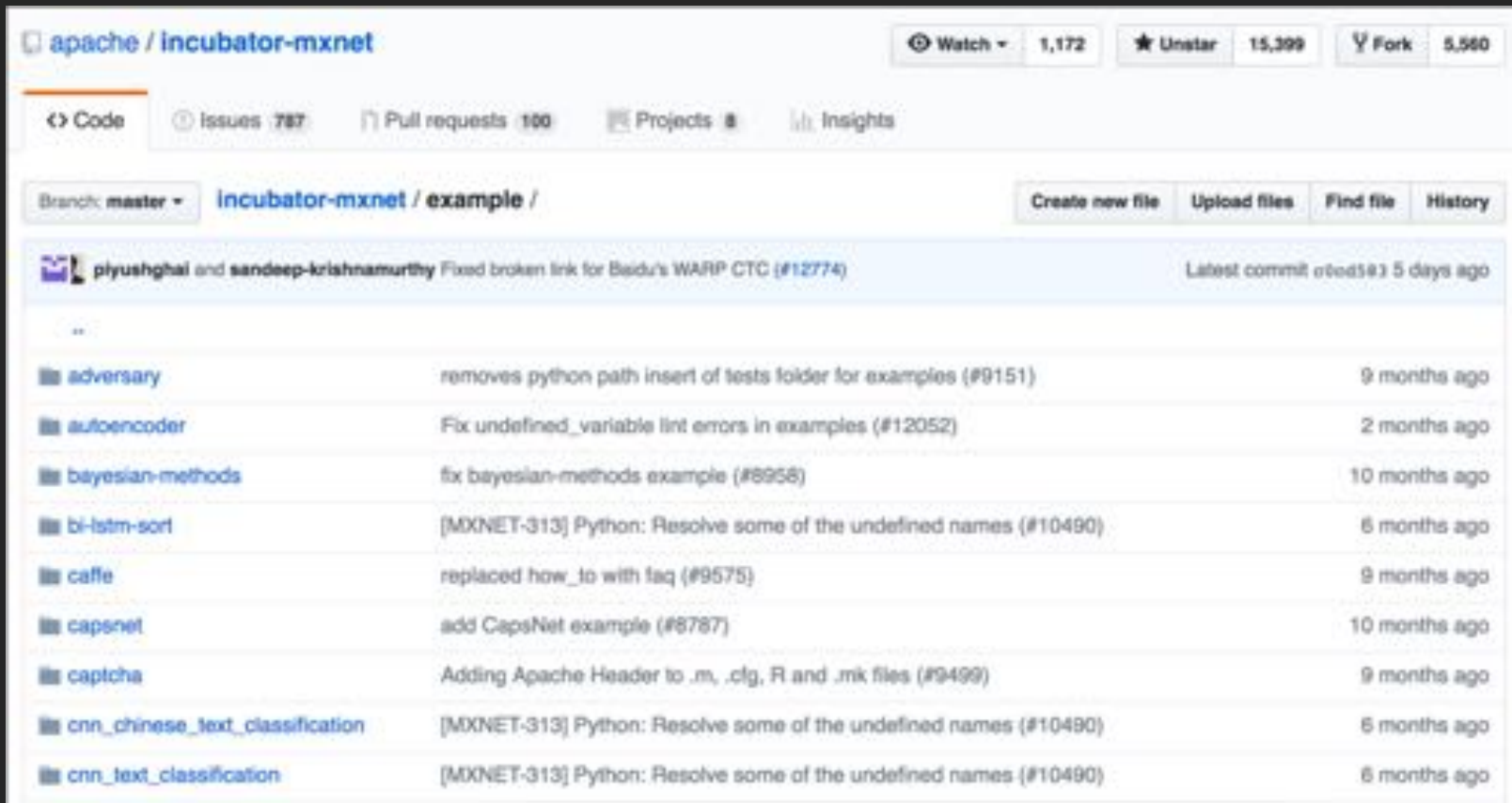
- Control flow operators do not work in imperative (NDArray) API** (NDArray, Question) - #12625, opened an hour ago by samskalicky, 2 comments.
- Is it necessary to wait CUDA stream when calling 'WaitToRead' or 'WaitToWrite'?** (Bug, CUDA) - #12623, opened a day ago by wkcn, 2 comments.
- [clojure-package] On modern Ubuntu, scala package cannot be loaded without legacy libcurl** (Clojure, Scala) - #12622, opened a day ago by kohr-h, 4 comments.
- To build a Multi-gpu tensor** (Python, Question) - #12620, opened 2 days ago by 729009982, 2 comments.

The fifth issue is partially visible: **mxnet-cu90mkl does not work on g2.8xlarge instance** (Breaking, Build, Installation) - #12619, 4 comments.

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

다양한 모델 샘플 코드 공헌 가능

SOSCON 2018



The screenshot shows the GitHub repository for Apache MXNet, specifically the 'example' directory. The repository is owned by 'apache' and has 1,172 watchers, 15,399 stars, and 5,560 forks. The 'example' directory is selected, showing a list of subdirectories and their commit history. The subdirectories include 'adversary', 'autoencoder', 'bayesian-methods', 'bi-lstm-sort', 'caffe', 'capsnet', 'captcha', 'cnn_chinese_text_classification', and 'cnn_text_classification'. Each subdirectory has a brief description of its contents and the date of the latest commit.

apache / incubator-mxnet

Watch 1,172 | Unstar 15,399 | Fork 5,560

Code | Issues 787 | Pull requests 100 | Projects 8 | Insights

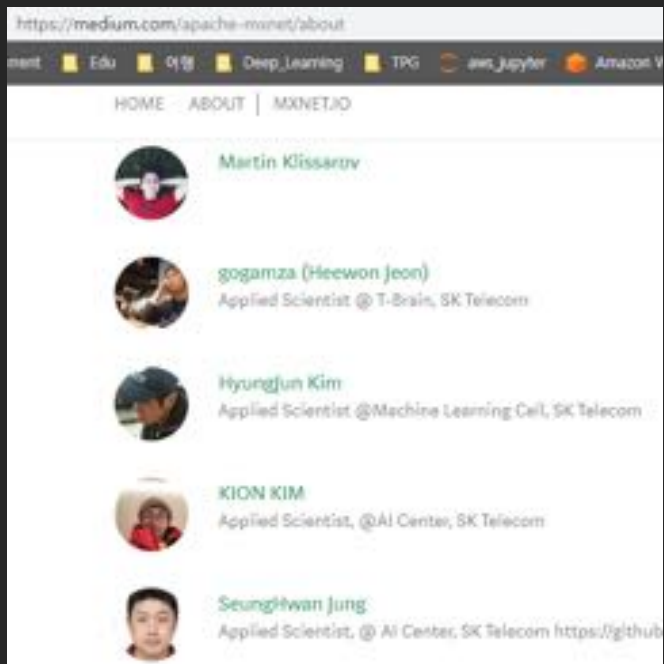
Branch: master ▾ | incubator-mxnet / example /

Create new file | Upload files | Find file | History

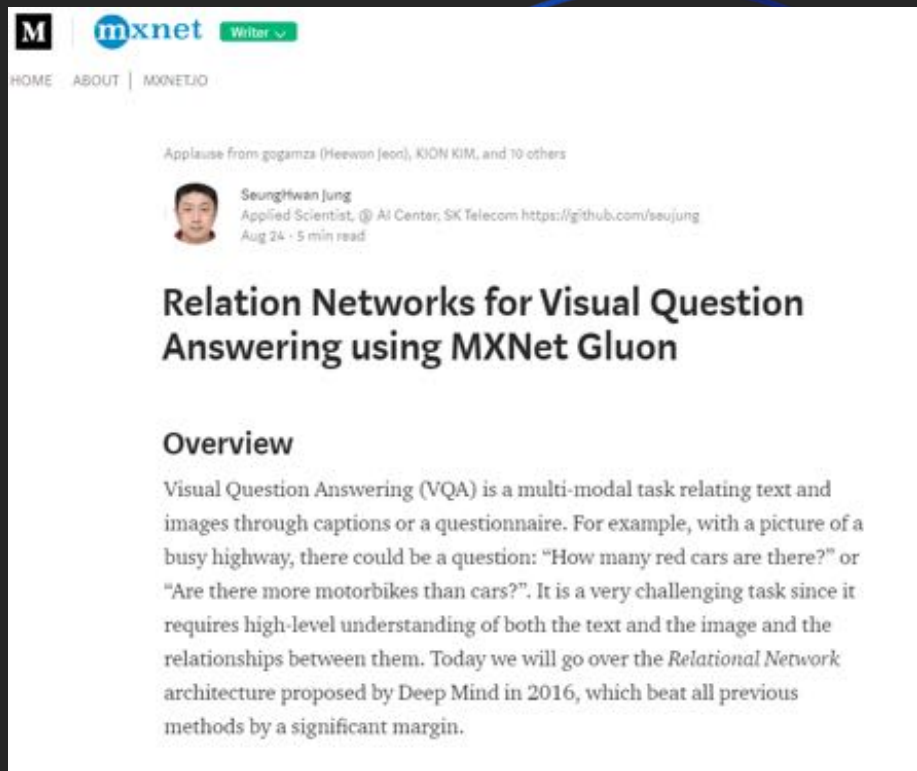
plyushghal and sandeep-krishnamurthy Fixed broken link for Baidu's WARP CTC (#12774) Latest commit 6ed593 5 days ago

adversary	removes python path insert of tests folder for examples (#9151)	9 months ago
autoencoder	Fix undefined_variable lint errors in examples (#12052)	2 months ago
bayesian-methods	fix bayesian-methods example (#8958)	10 months ago
bi-lstm-sort	[MXNET-313] Python: Resolve some of the undefined names (#10490)	6 months ago
caffe	replaced how_to with faq (#9575)	9 months ago
capsnet	add CapsNet example (#8787)	10 months ago
captcha	Adding Apache Header to .m, .clg, R and .mk files (#9499)	9 months ago
cnn_chinese_text_classification	[MXNET-313] Python: Resolve some of the undefined names (#10490)	6 months ago
cnn_text_classification	[MXNET-313] Python: Resolve some of the undefined names (#10490)	6 months ago

<https://github.com/apache/incubator-mxnet/tree/master/example>



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



MXNet 커뮤니티 참여

SOSCON 2018



MXNet 한국 사용자 모임

<https://www.facebook.com/groups/mxnetkr/>



<https://www.meetup.com/MXNet-Korea-User-Group/>



THANK YOU

Q&A

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

