

Learning

1. Bias for Action

a. Could have shown but didnt

- i. Sam - AWS isengaard, launching EC2 instances need security groups. It creates one by default but then everyone uses the default ones for every single time new EC2 instance is created leading to upper limit reached.

1. N Virginia (Washington DC - IAD Intl Airport Dulles (us-east-1)

ii. Lin -

APPENDIX

highly competitive object recognition benchmark tasks (CIFAR-10, CIFAR-100, SVHN, and ImageNet)

CIFAR - **Canadian Institute For Advanced Research**

10 refers to # of classes

SVHN - Street View House Numbers

DenseNet (Dense CNN)

FCN - Fully Convolutional Network

FC Layer - Fully Connected layer

FPN - Feature Pyramid Network

VGG - Visual Geometry Group, deep convolutional network for object recognition (good performance on ImageNet dataset)

NCHW / NHCW aka Channel first or channel last -

N: number of images in the batch

H: height of the image

W: width of the image

C: number of channels of the image (ex: 3 for RGB, 1 for grayscale..)

TESTING

Tolerance, RTOL - relative ATOL - absolute

Flaky tests - tests that fail intermittently on CI builds and they may indicate stability problems or improper handling of edge cases

Travis and Jenkins - CI (continuous integration) used for testing

OOM for OD - Out of Memory error for Object Detection

Idempotent test

Testing libraries in Python

- unittest
- pytest
- nosetest

SEMANTIC SEGMENTATION IN DL

- Classification - prediction for whole input (e.g. img)
- Localization/Detection - classes + additional info abt spatial location of those classes
- Semantic Segmentation - fine-grained inference (dense predictions inferring labels for every pixel)

Integration tests and Main service can be in 2 separate languages

Canary tests - minimal tests to quickly and automatically verify that the everything you depend on is ready.

Adv - Same time investigating code when environment itself isn't set up.

Canary Deployment - deploy to a small fraction of your users to check everything's fine before rolling out to more users.

Computer time is cheap. Human time is expensive

"trying to optimize Java for Lambda code start times is something of a black art."

Backend / workflows - Java - language everybody is familiar with and the ecosystem at Amazon is very strong.

Debug flag is default set to 0 to reduce pain point for the new users

ROI ALIGN - REGION OF INTEREST

PRELU

Parametrized Rectified Linear Unit

SSD - Single shot Multi box

DHT - distributed hash table

- Motivated by peer-to-peer systems (Bit torrent, Napster, Coral CDN, Gnutella)
- Properties:
 - Autonomy, Decentralization
 - Fault tolerance
 - Scalability
- Structure
 - Keyspace Partitioning
 - Consistent Hashing
 - Rendezvous Hashing aka highest random weight
 - Locality-preserving Hashing
 - Overlay network

Elastic Training

- OD (On Demand) vs RI (Reserved instances)

CI

- Kernighan-Lin algorithm
 - a heuristic algorithm for finding partitions of graphs.
 - The goal of the algorithm is to partition V into two disjoint subsets A and B of equal (or nearly equal) size, in a way that minimizes the sum T of the weights of the subset of edges that cross from A to B .
 - unweighted graph (goal is to min # of crossing edges) i.e. $w=1$ for each edge
 - Greedy algo
 - $O(n^2 \log n)$.

MMS Bug bash

- uncovered issue with byte array bytes

- open cv reads image and converts into byte (mxnet1.3 had an issue - didn't support byte array, only supported bytes)

Python

- `__all__` → customizes the asterisk in `from <module> import *` and `from <package> import *`
- A module is a .py file meant to be imported
- Package is a module that can contain submodules (and subpackages). It's a directory with `__init__.py`
 - Any module that has `__path__` attribute is a package
 - If a directory has a bunch of python scripts (modules) but no `__init__.py`, it isn't a package
 - Packages can be nested to any depth, provided that the corresponding directories contain their own `__init__.py` file
- Relative imports
 - `from <> import;`
 - `import <>` is always absolute.
 - absolute imports can use `from <> import` by omitting the leading dots
 - `import .foo` is prohibited since


```
import XXX.YYY.ZZZ
XXX.YYY.ZZZ
```

 - But `.foo` isn't usable
 -
 -
- Calling a function using object of class
- It gave an error when I used `ndarray.abs(data)` or `ndarray.one_hot(data)`. But worked when I used `data.abs()` or `data.one_hot()`
 - Wrong - `ndarray.one_hot(ndarray.argmax(data, -1), data.shape[-1])`
 - Right - `data.argmax(-1).one_hot(data.shape[-1])`

Able to reproduce the results = rerun it and get the same output, deterministic results/execution

Software Release

- Experimental / POC (proof of concept)
- Pre-alpha
- Alpha
 - white-box testing (additionally gray-box / black-box)
 - ends with feature freeze
- Beta
 - perpetual beta = "release early, release often" (Open Source dictum)
 - (Gmail, Maps) (Users = co-developers)
 - developing in the open, new features slipstreamed in on daily/weekly/monthly basis
 - Open vs Closed beta (Public vs private)
- RC (Release candidate) / going silver
- RTM (Release to manufacturing) / going gold

- GA (General Availability)
- End of life

Amazon-specific

- Amazon Standard Identification Numbers (ASINs) are unique blocks of 10 letters and/or numbers that identify items.
- Product2Vec
 - generating embeddings for products at Amazon scale
 - office Precision@k metrics
 - Substitute vs Complement
 - an embedding is trained on *some* sort of ground truth, and we can't expect that type of ground truth is suitable for every possible use case across amazon

RL

- Edge computing and Edge devices
 - **distributed computing paradigm** in which computation is largely or completely performed on **distributed device nodes** known as smart devices or edge devices as opposed to primarily taking place in a *centralized cloud environment*. Edge computing vs data offboarding
 - Advantage
 - reducing transmission costs, shrinking latency
 - improving quality of service (QoS).
 - remove single PoF
 - edge device is a device which provides an entry point into enterprise or service provider core networks.
 - Eg -routers, routing switches, integrated access devices (IADs), multiplexers, and a variety of metropolitan area network (MAN) and wide area network (WAN) access devices

Design Review MXNet

- RecordIO (image processing)
- solution to cumbersome existing method of cloning repo, for creating dataset using CLI
- Thread vs Worker
 - worker (master - slave, distributed)
 - thread - parallel computation, on 1 machine

D file

The **D** programming language is similar to the C++ programming language, but combines C++ with the power of other modern programming languages like Python and Ruby.

Many build systems add automatically detected make dependencies into the .d file. In particular, for C/C++ source files they determine what `#include` files are required and automatically generate that information into the .d file.

The .d files are then included by the makefile so make is aware of that information. If you look at the contents of those files they'll be make prerequisite statements, like:

```
foo.o : foo.h bar.h biz.h
```

setting up random seed

```
import mxnet as mx
mx.random.seed()
```

Sparse Matrix

2 main types of format

Support modifications - Dictionary of keys (DOK), List of Lists (LiL), COO (Coordinate Lists)

Support access & matrix operations - CSR - compressed sparse row / compressed row storage (CRS),

Row major order / Column major order

Template specialization

Template metaprogramming

Why? - generic, reusable code

entire STL (std template library)

```
template <typename T>
T byte_swap( T value ) {
}
```

Issue with basic template programming - no control over what type T gets passed in

Hence “partial specialization”

```
template <>
double byte_swap( double value ) {
    assert( false && "Illegal to swap doubles" );
    return value;
}

template <>
char byte_swap( char value ) {
    assert( false && "Illegal to swap chars" );
    return value;
}
```

Now, what about float, pointer, etc? Many other data-types to be handled, separately and individually.

simple function would balloon out into many specializations, all to protect the caller

Solution - Type traits

a way for you to get information about the types passed in as **template arguments**, at **compile time**, so you can make more intelligent decisions

LWTA - Local Winner Take All

```
In member function 'int64_t mxnet::common::random::RandGenerator<mshadow::gpu, DType>:
include/mxnet/././random_generator.h:148:44: error: left shift count >= width of type
    int64_t result = (curand(&state_) << 32) + curand(&state_);
```

FFNN - Feed forward neural network

DNN - deep NN

RNN - Recurrent NN

Diff b/w FFNN and RNN

activations from previous time steps

- act as input to network for deciding the current input
- stored in internal state of network
- provide indefinite temporal contextual info (unlike fixed contextual windows in FFNN)
 - i.e. use dynamically changing all sequence history (rather than static fixed window over sequence)

Hence RNN better for sequence modelling tasks viz. seq prediction, seq labeling

RNN with gradient based BPTT suffers from exploding and vanishing gradient