DL4J

DEEP LEARNING
FOR ENTERPRISE

Deep Learning - why now?

Confluence of Algorithms Compute Data

Goals for today: Enable you

- The ten thousand foot view on “Why dl4j”
- Basics of the api: Some nuts and bolts

Skymind?

We are a distributed team living and working in every major time zone

WE BUILT DEEPLEARNING4J

WHY THE JVM?

"We are stuck with technology when what we really want is just stuff that works."

DOUGLAS ADAMS

The Economist Espresso

ENTERPRISE NEEDS

Data Gravity

• We need to process the data in workflows where the data lives • *If you move data you don't have big data*

• Even if the data is not "big" we still want simpler workflows

Integration Issues

• Ingest, ETL, Vectorization, Modeling, Evaluation, and Deployment issues, security etc

The JVM is great at network I/O and data access. Also great at streaming infrastructure. BUT…

When considering HPC on the JVM

Scientific computing & the JVM

- Vectorization
- Array indexing, 32 bit address space

FULLY native backend

https://github.com/deeplearning4j/libnd4j

JAVACPP OpenMP CUDA

JAVACPP

- Auto generate JNI bindings for C++ by parsing classes

- Allows for easy maintenance and deployment of C++ binaries in java

- Write efficient ETL pipelines for images via opencv (javacv)

- 64 bit pointers (wasn’t possible before)

ND4J

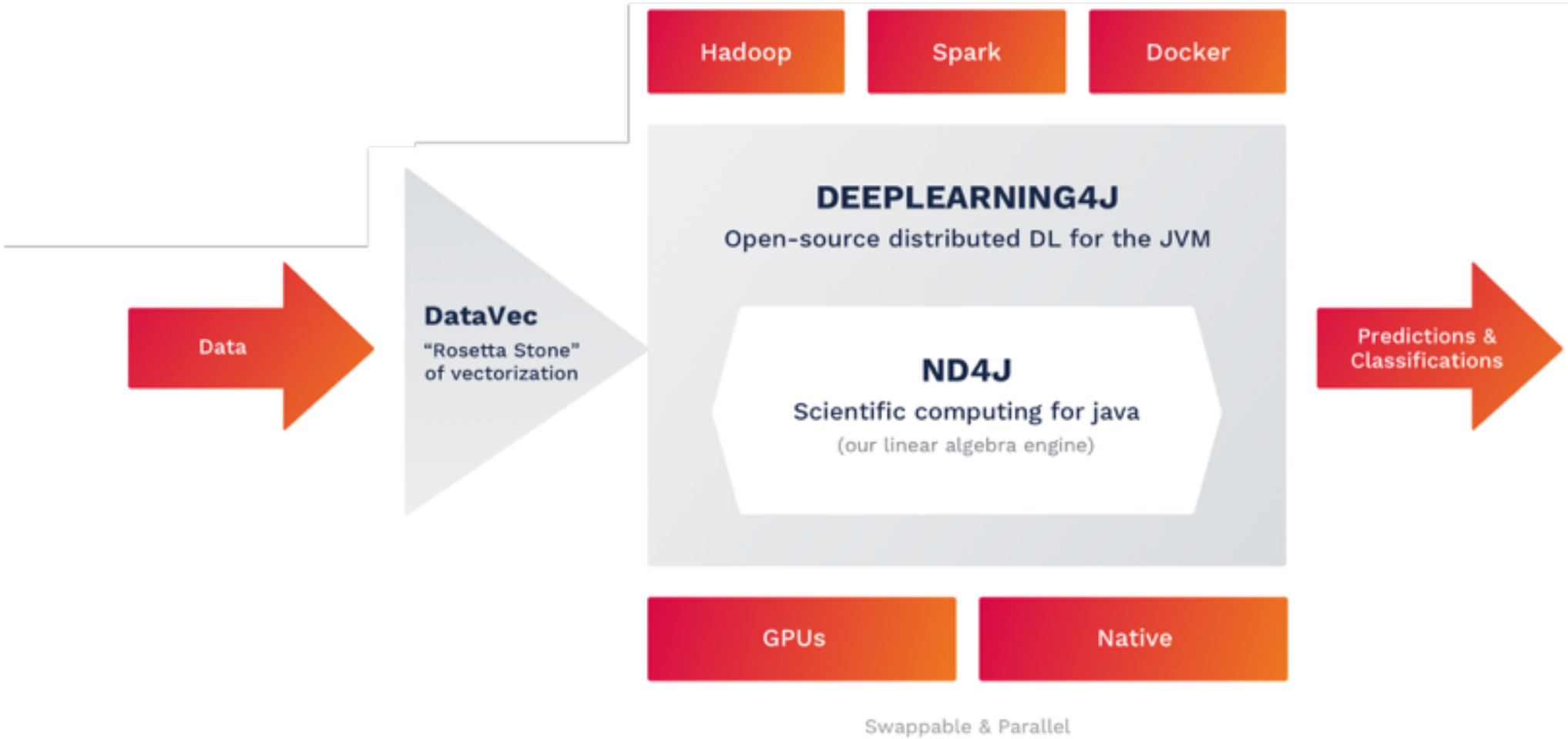
- Heterogenous codebase
- Supports cuda, x86 and Power
- Shared indexing logic for writing ndarray routines
- Memory management in java (even cuda memory!)

BOTTOMLINE:

When in production, what problems are you spending your time on? How do they align with your end goal?

Are you rearranging deck chairs on the titanic…

SCHEMATIC OVERVIEW



From the NeedForSpeed to Distributed Deep Learning

Switching to GPUs

<dependency>

<groupId>org.nd4j</groupId> <artifactId>**nd4j-cuda-7.5**</artifactId> <version>

${nd4j.version}</version>

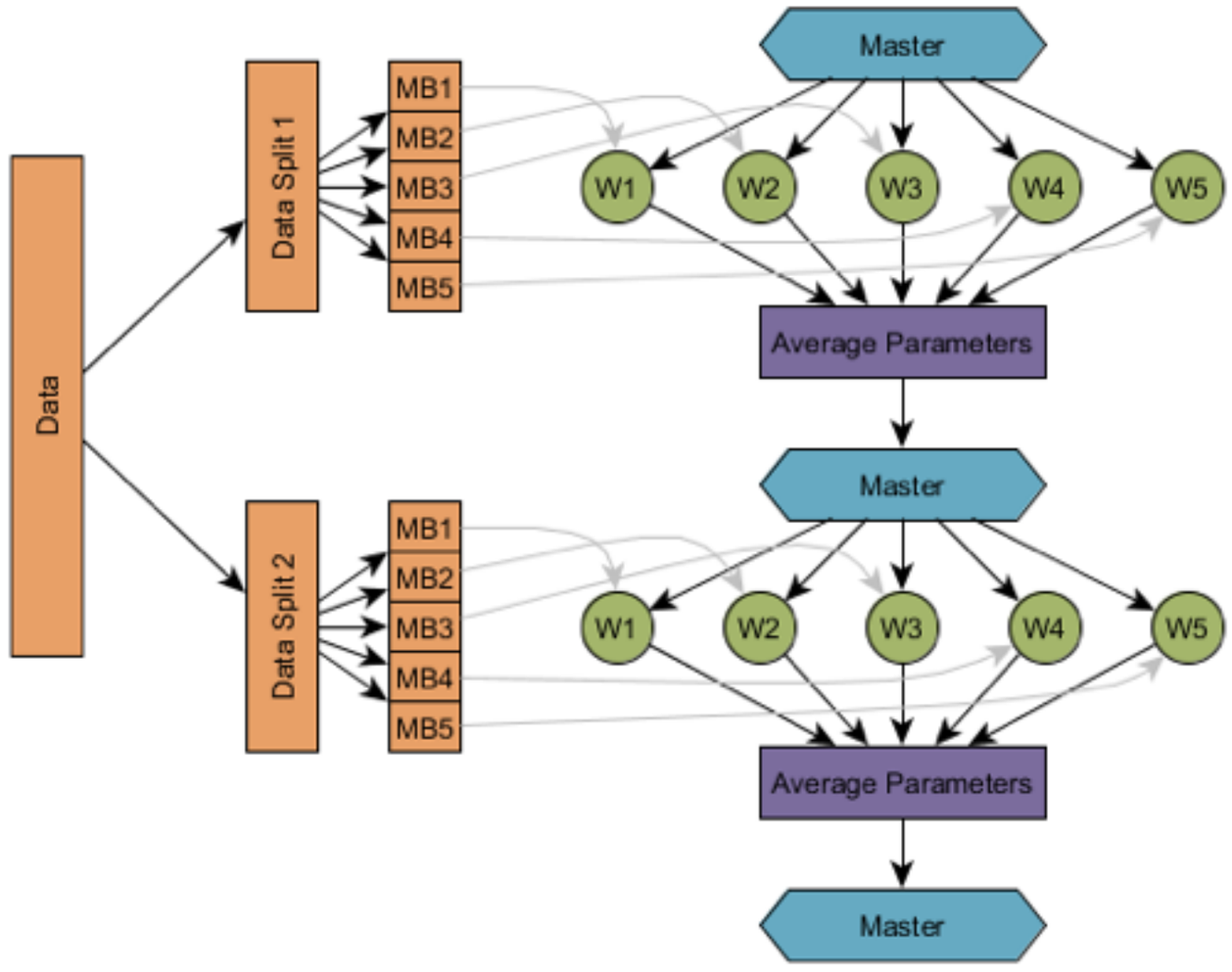
</dependency>

Onto multi-GPUs

```
ParallelWrapper wrapper =
    new
ParallelWrapper.Builder(YourExistingModel)
                        .prefetchBuffer(24)
                            .workers(4)
                        .averagingFrequency(1)
                    .reportScoreAfterAveraging(true)
                        .useLegacyAveraging(false)
                            .build();
```

http://deeplearning4j.org/gpu

Deep Learning with Parameter Averaging



http://engineering.skymind.io/distributed-deep-learning-part-1-an-introduction-to-distributed-training-of-neural-networks

Onto Spark…

```
//Create the TrainingMaster instance
int examplesPerDataSetObject = 1;
TrainingMaster trainingMaster =
new ParameterAveragingTrainingMaster.
Builder(examplesPerDataSetObject).
(other configuration options ).build();

//Create the SparkDl4jMultiLayer instance
SparkDl4jMultiLayer sparkNetwork = new SparkDl4jMultiLayer(sc,net
workConfig,trainingMaster);

//Fit the network using the training data:
sparkNetwork.fit(trainingData);
```

http://deeplearning4j.org/spark

Spark with a GPU cluster

Distributed GPUs With Spark…

`http://deeplearning4j.org/spark-gpus`

Tune appropriately and keep gpu memory into consideration

Double performance gains?

Huge gains on memory-intensive operations

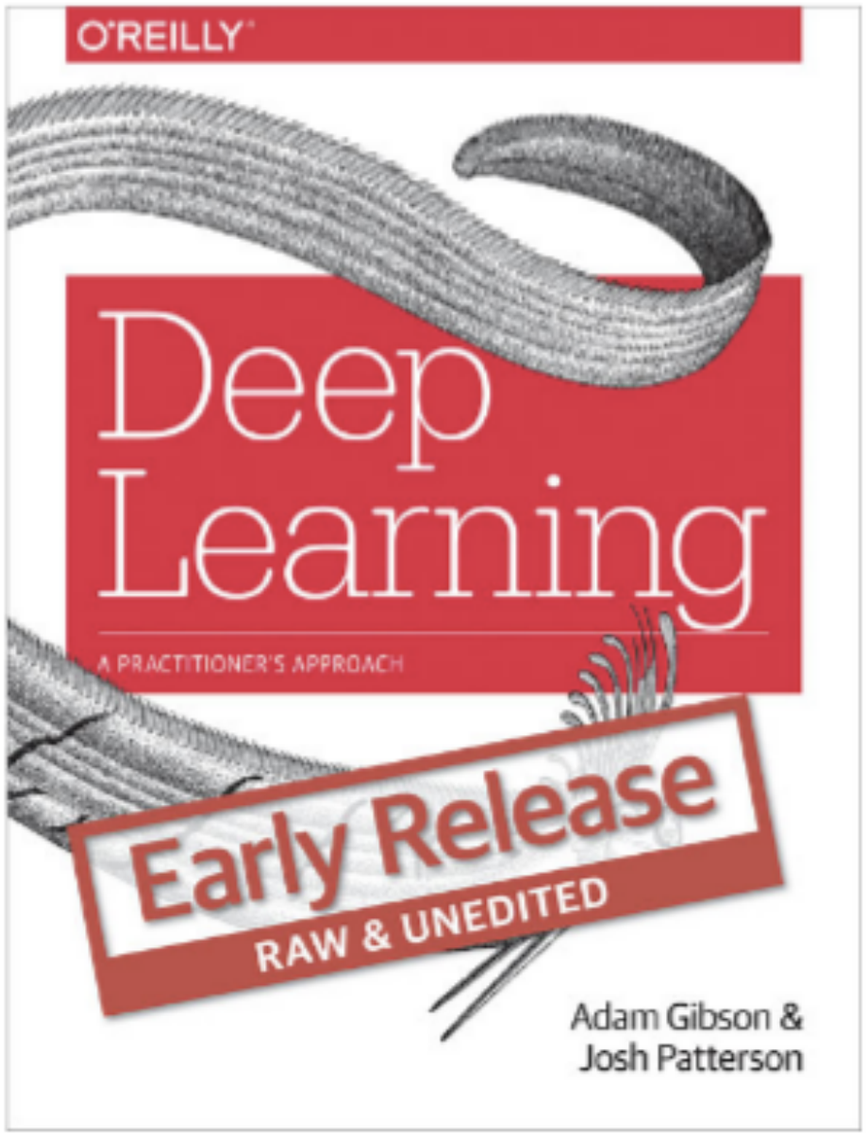
if you are willing to sacrifice some precision

```
DataTypeUtil.setDTypeForContext(DataBuffer.
Type.HALF);
```

DEEP LEARNING:
A PRACTITIONER'S APPROACH

From the perspective of the practitioner -

Written by Skymind's Adam Gibson and Josh Patterson.



Exercises

QUICK START

http://deeplearning4j.org/quickstart

Get support on Gitter.

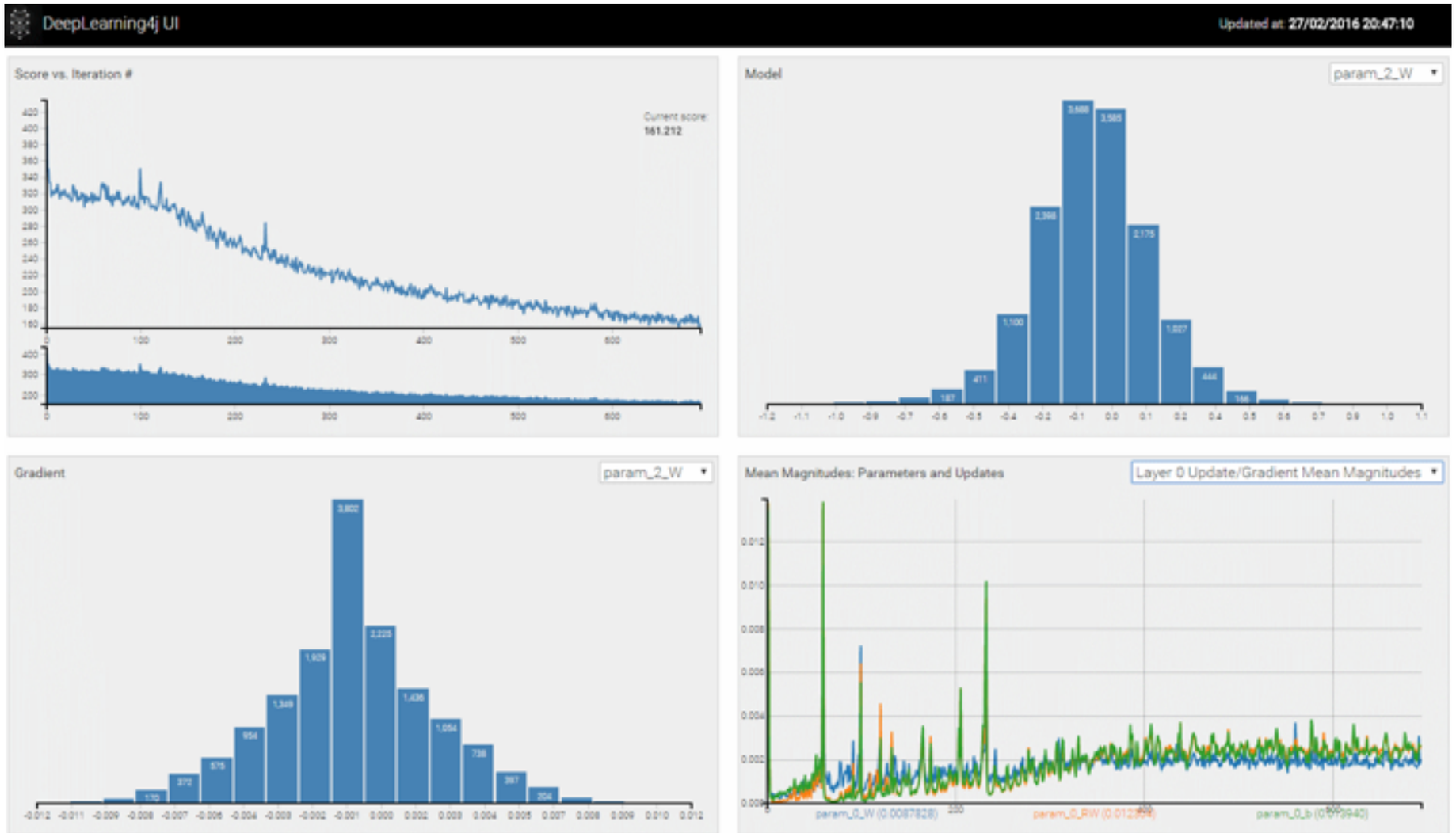
https://gitter.im/deeplearning4j/deeplearning4j

Example Repo

https://github.com/deeplearning4j/dl4j-examples

Inside the black box…

Use the UI

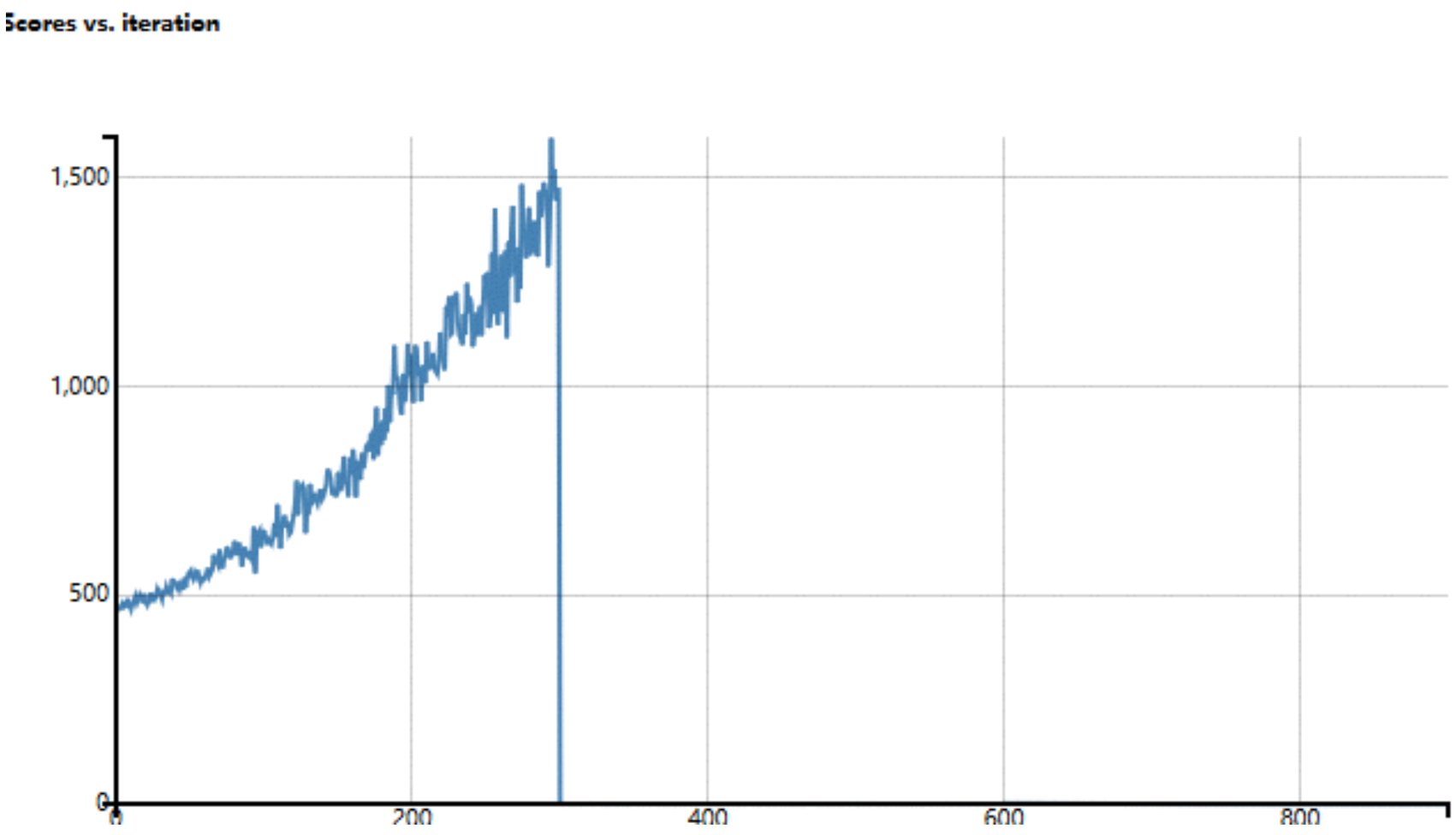

DeepLearning4j UI
Updated at: 27/02/2016 20:47:10
Score vs. Iteration #
Current score:
161.212
Model
param_2_W
Gradient
param_2_W
Mean Magnitudes: Parameters and Updates
Layer 0 Update/Gradient Mean Magnitudes

All this and more…

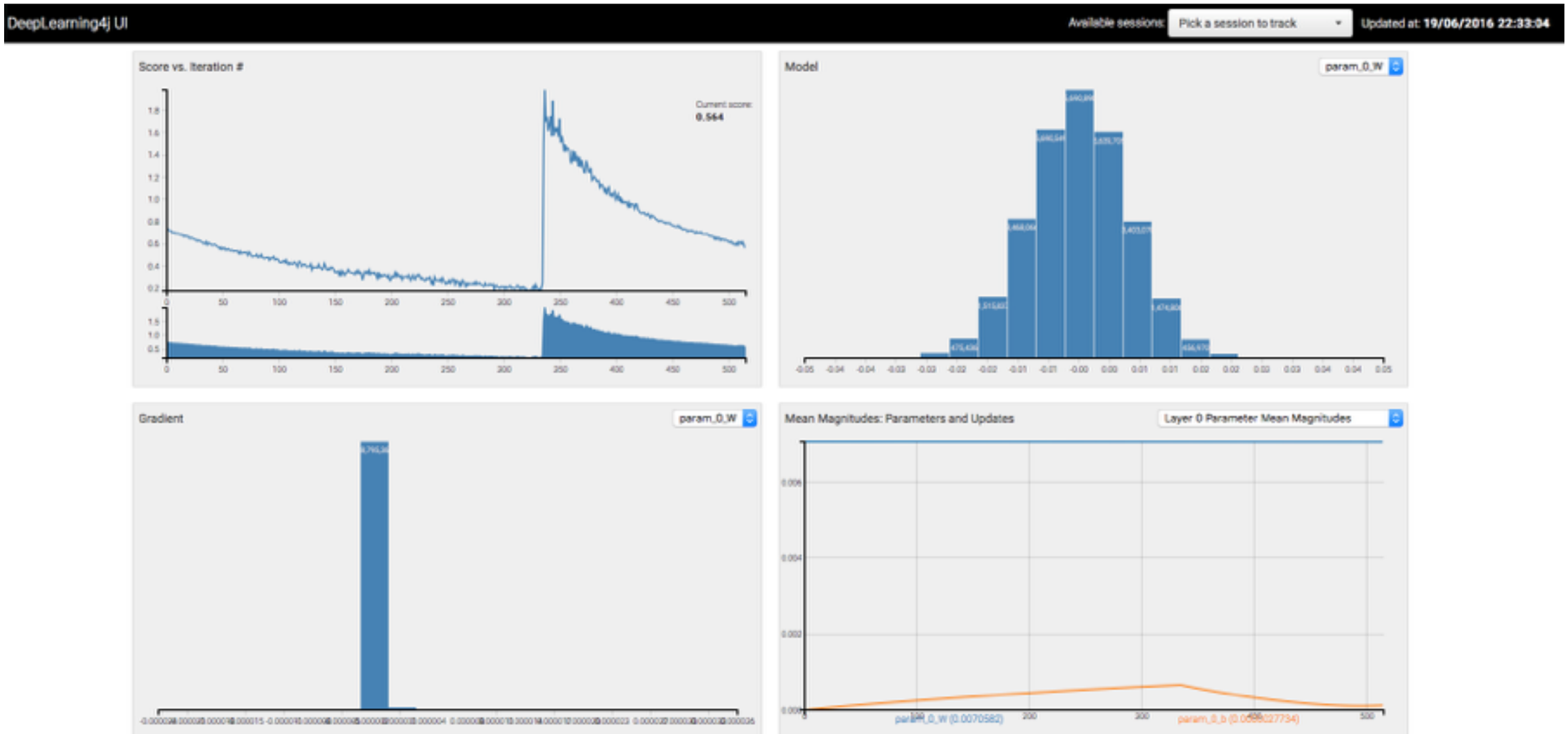
http://
deeplearning4j.org/
visualization

What happened here?

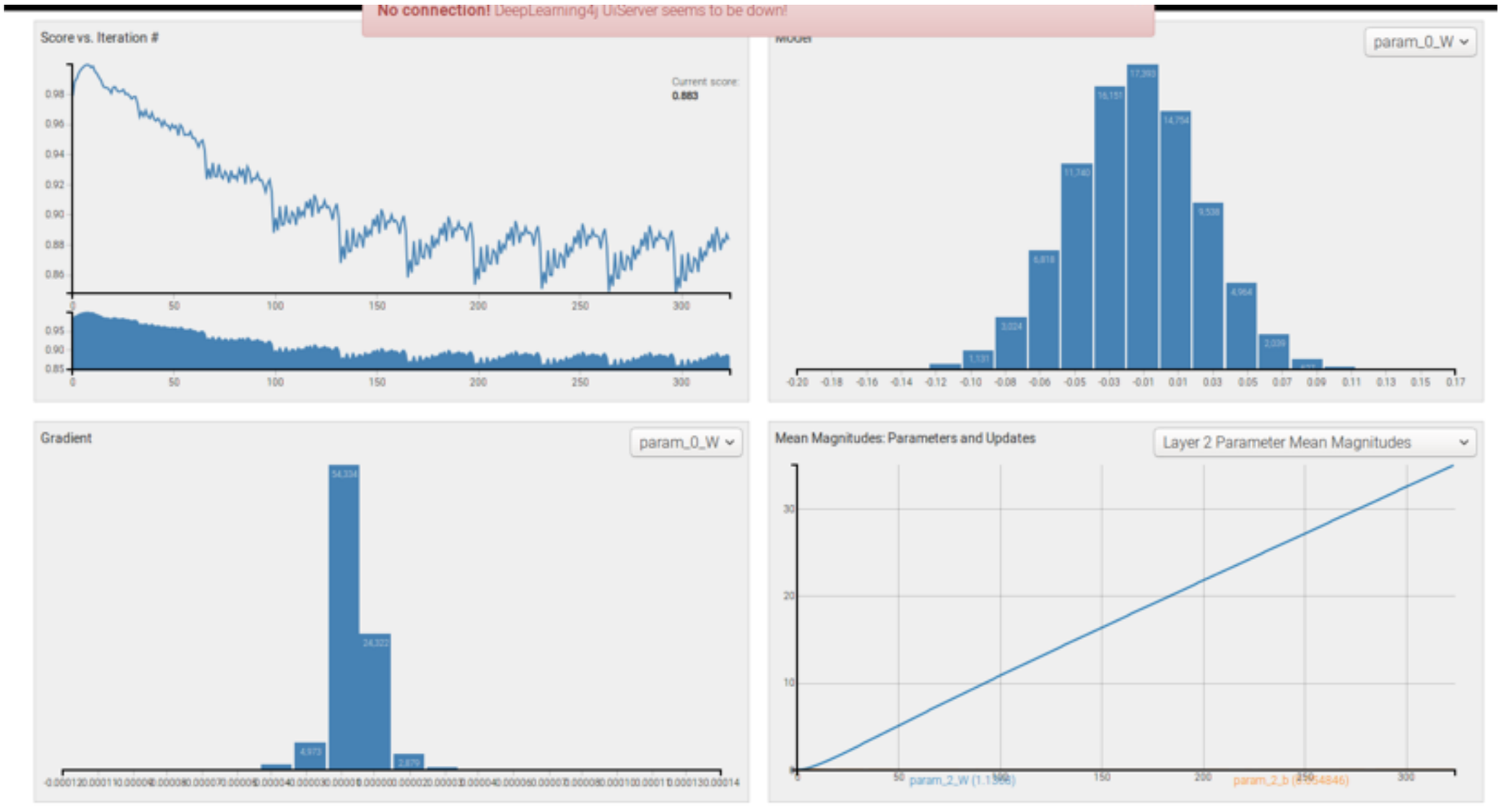
Here?



Here?


DeepLearning4j UI
Available sessions:
Pick a session to track
Updated at: 19/06/2016 22:33:04
Score vs. Iteration #
Current score:
0.564
Model
param_0_W
Gradient
param_0_W
Mean Magnitudes: Parameters and Updates
Layer 0 Parameter Mean Magnitudes

Here?


No connection! DeepLearning4j UiServer seems to be down!
Score vs. Iteration #
Current score:
0.883
param_0_W
Gradient
param_0_W
Mean Magnitudes: Parameters and Updates
Layer 2 Parameter Mean Magnitudes
param_2_W (1.1366)
param_2_b (0.0854846)

I don’t understand why I get nans?!

The saga of the nans

- Not a saga - any numerical software/ scientific computing can/will run into this. Default data type is float, could change it to double.

- Most often - **It’s a tuning issue.** If it's diverging just keep adjusting your parameters.

Nd4j exercises (take home)

- Main gotcha - The idea of views. Reference here

Simple MLP example

- Example for reference here

- Build your own example - fizz buzz.

CNN example

- Example for reference here, LeNet

RNN example

- Example for reference here, UCI classification

ADDITION as SeqToSeq

73+45=118

37+54=91

&#\$%/!!*

#&\%$/(!Z

What is the random probability of getting an addition right?

Example here

BackUp

LET’s TALK TUNING!

Learning Rate, updaters, batch size, loss functions,...

Normalize your data!

- continuous values
- discrete classes
- repeat the exact same method for training and test

Weight Initialization

- Xavier weight initialization
- RELU weight initialization

Learning Rate, Epochs, Iterations, Minibatch size

- A learning rate range typical range: 0.1 to 1e-6
- Multiple epochs and one iteration
- 16 to 128 minibatch, with a max 1k

Activation Functions, Loss Functions

- Regression/Classification: MSE (MAE,MAPE..)/ XENT (KLD..) + softmax
- relu, leakyrelu, identity, tanh, softsign

Remember overfitting and the bias/variance trade off?

Regularization

- l1, l2 regularization. Common values for l2 regularization are 1e-3 to 1e-6.
- Dropout,
- Dropconnect
- Restrict network size
- Early stopping

Updaters, Optimization Algorithms

- Updaters: momentum, RMSProp, adagrad etc
- Optimization Algorithms: SGD with line search, conjugate gradient and LBFGS optimization algorithms

Exploding and Vanishing…

Gradient Normalization

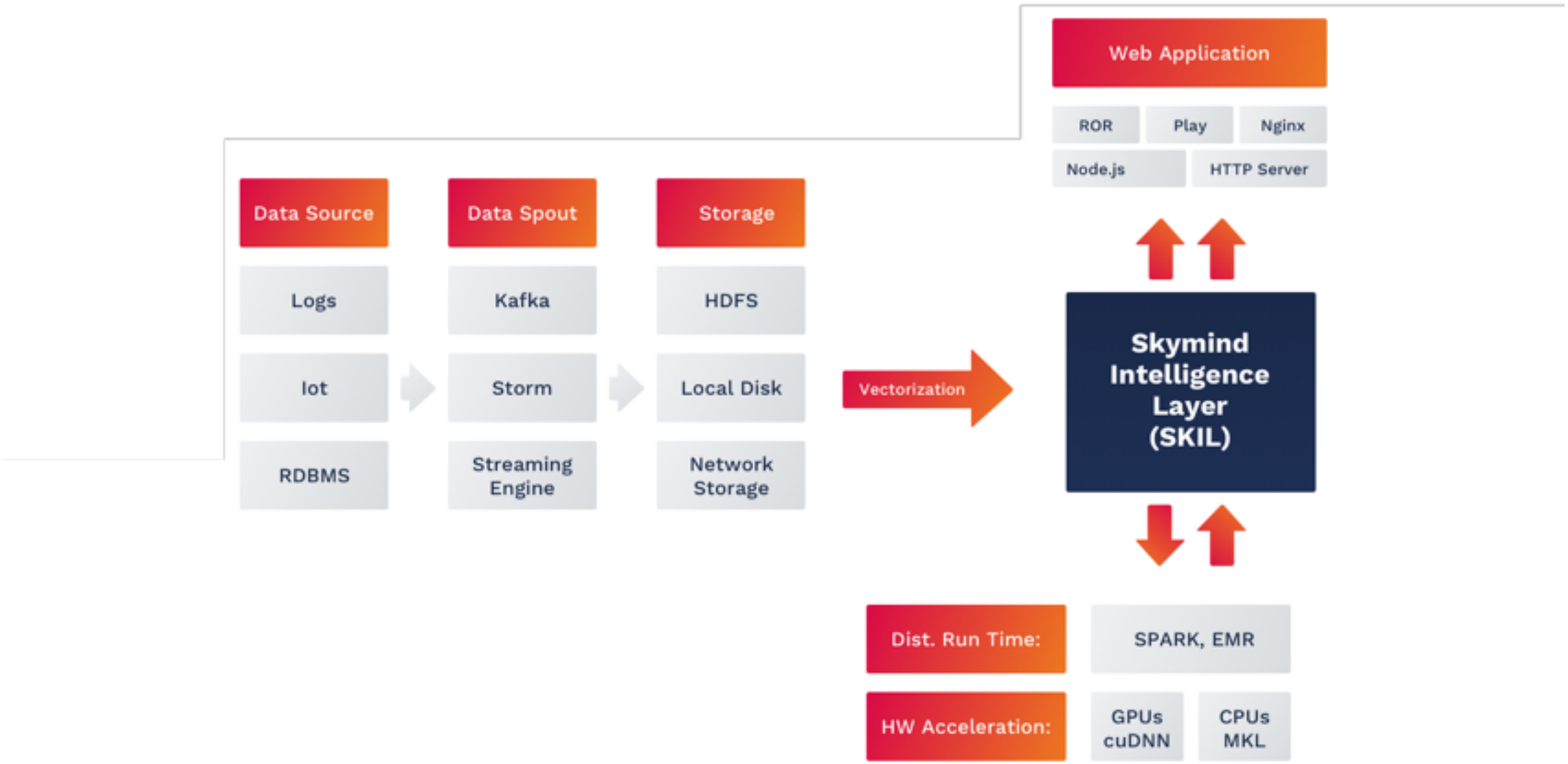
- gradientNormalization
- gradientNormalizationThreshold

All this and more…

http://deeplearning4j.org/troubleshootingneuralnets

SKYMIND INTELLIGENCE LAYER (SKIL)

Reference Architecture



SKIL-DC/OS


skil-dcos
52.53.175.184
Dashboard
Services
Jobs
Nodes
Network
Universe
System
chris@skymind.io
CPU Allocation
50%
22 of 44 Shares
Memory Allocation
41%
61 GiB of 151 GiB
Disk Allocation
42%
162 GiB of 382 GiB
100%
50%
0%
-60s
-30s
0
Services Health
cassandra Healthy
hdfs Healthy
marathon-user Healthy
spark Idle
Tasks
17
Total Tasks
Component Health
Admin Router Agent Healthy
Admin Router Master Healthy
Admin Router Reloa... Healthy
Admin Router Reloa... Healthy
Admin Router Reloa... Healthy

SKIL / DL
DOCKER
DCOS + SPARK
MESOS
MULTI-GPUS

skymind

DEEP LEARNING
FOR ENTERPRISE