DeepLearning on the JVM - ApacheSystemML and DeepLearning4J on ApacheSpark

DeepLearning frameworks are popping up at very high frequency but only a few of them are suitable to run on clusters, use GPUs and supporting topologies beyond Feed-Forward at the same time. DeepLearning4J features all this without forcing you to learn new exotic programming languages and in addition also scales-out on well established infrastructures like ApacheSpark and Hadoop/YARN.

In this talk we will introduce DeepLearning4J on top of ApacheSpark with an example to create an anomaly detector for IoT sensor data with a LSTM auto encoder neural network.

Unless stated otherwise all images are taken from wikipedia.org or openclipart.org

IBM Disclaimer

- THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
- WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN

THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

- IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO,

THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

- NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:
  - CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
  - ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

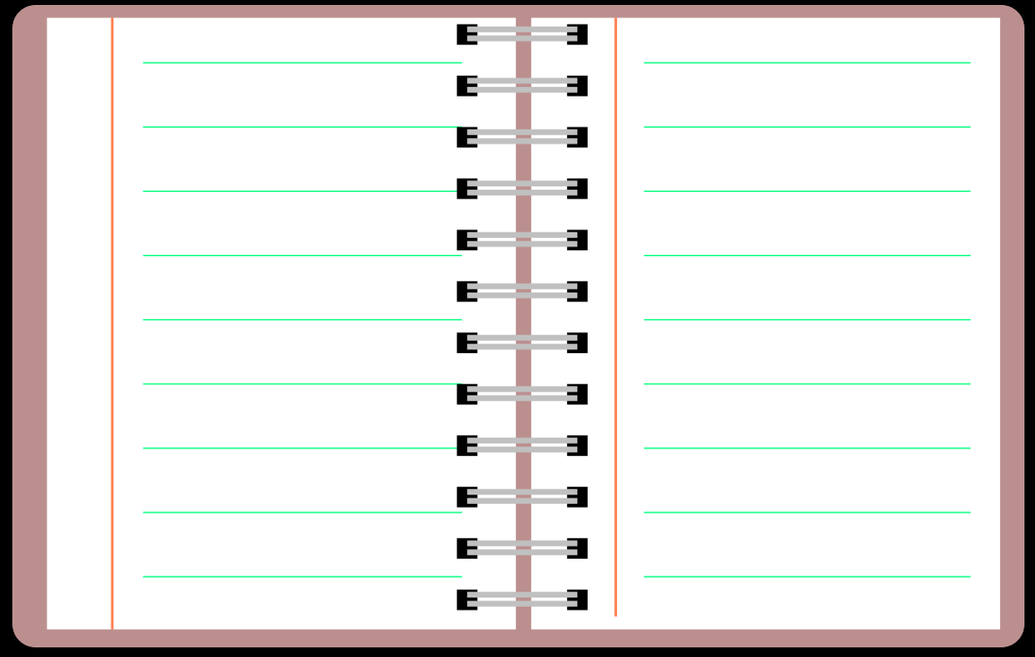
We are looking forward to hearing your suggestions and feedback so that we can improve our solutions and products.

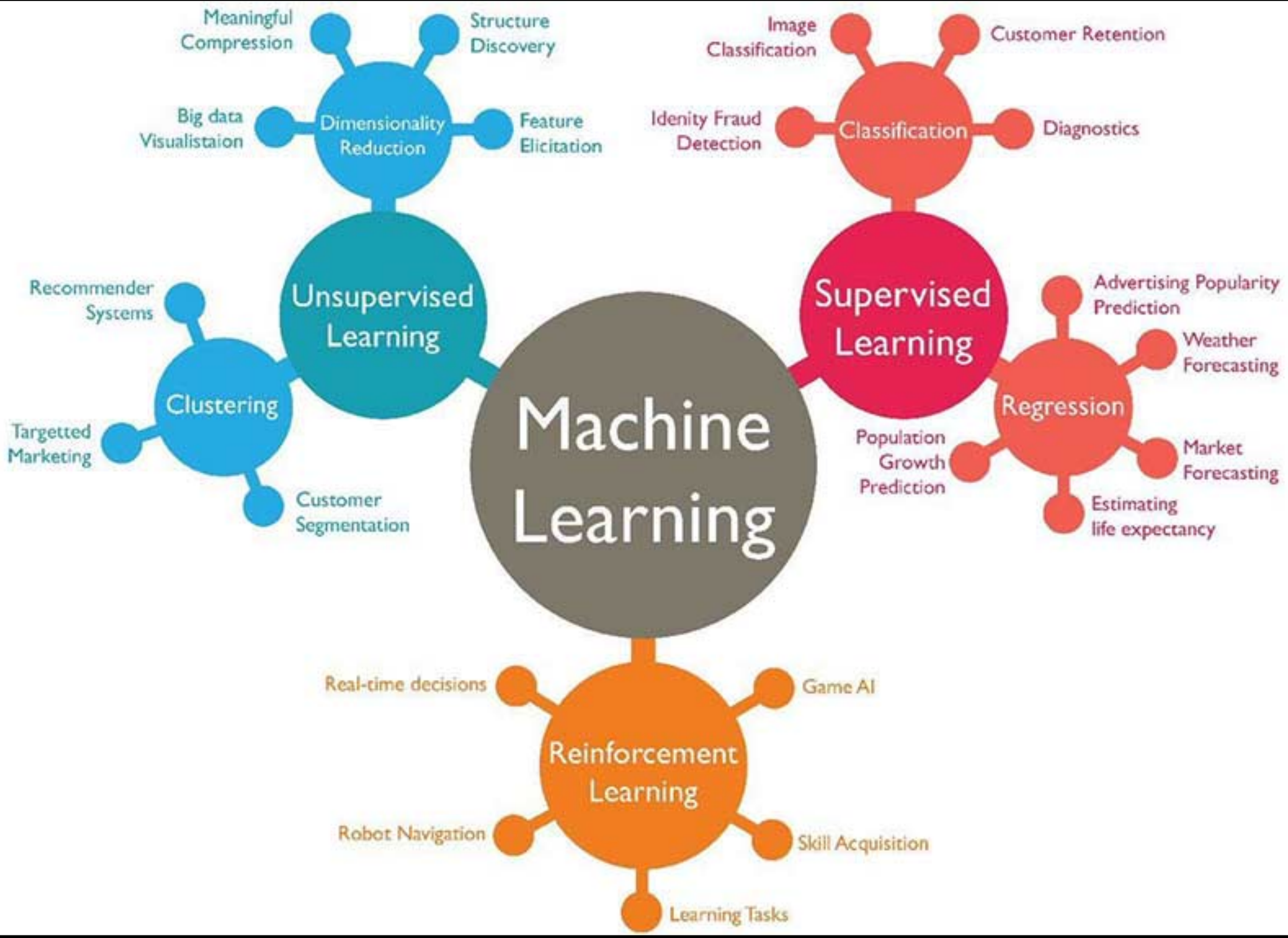
By joining this web conference, or clicking the “Join" link to join the ITM vNext Open Beta Community, you agree to the following terms, in addition to those set forth in the developerWorks terms of use:

IBM shall be free to use for any purpose and without restriction any oral or written suggestions or feedback that you provide to IBM. By providing IBM with any information or material, you grant IBM an unrestricted, irrevocable license to copy, reproduce, publish, upload, post, transmit, distribute, publicly display, perform, modify, create derivative works from, and otherwise freely use, those materials or information. You also agree that IBM is free to use any ideas, concepts, know-how, or techniques that you provide us for any purpose.

Agenda

- Introduction to DeepLearning ✓
- Introduction to ApacheSpark
- Introduction to DL4J
- Introduction to SystemML
- Demos





Source: https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/

Neural Networks

Part No.	Max Temp. 1	Min Temp 1	Max Vibration 1	Outcome
100	35	35	12	1
101	46	35	21	1
130	56	46	3412	0

Neural Networks

Part No.	Max Temp. 1	Min Temp 1	Max Vibration 1	Outcome
100	35	35	12	1
101	46	35	21	1
130	56	46	3412	0

```
def predict(datapoint):
    if datapoint.MaxVibration1>100:
        return 0
    else:
        return 1
```

Neural Networks

Part No.	Max Temp. 1	Min Temp 1	Max Vibration 1	Outcome
100	35	35	12	1
101	46	35	21	1
130	56	46	3412	0

```
def predict(dp):
    return a+b*dp.MaxTemp1+c*dp.MinTemp1+d*dp.MaxVibration1
```

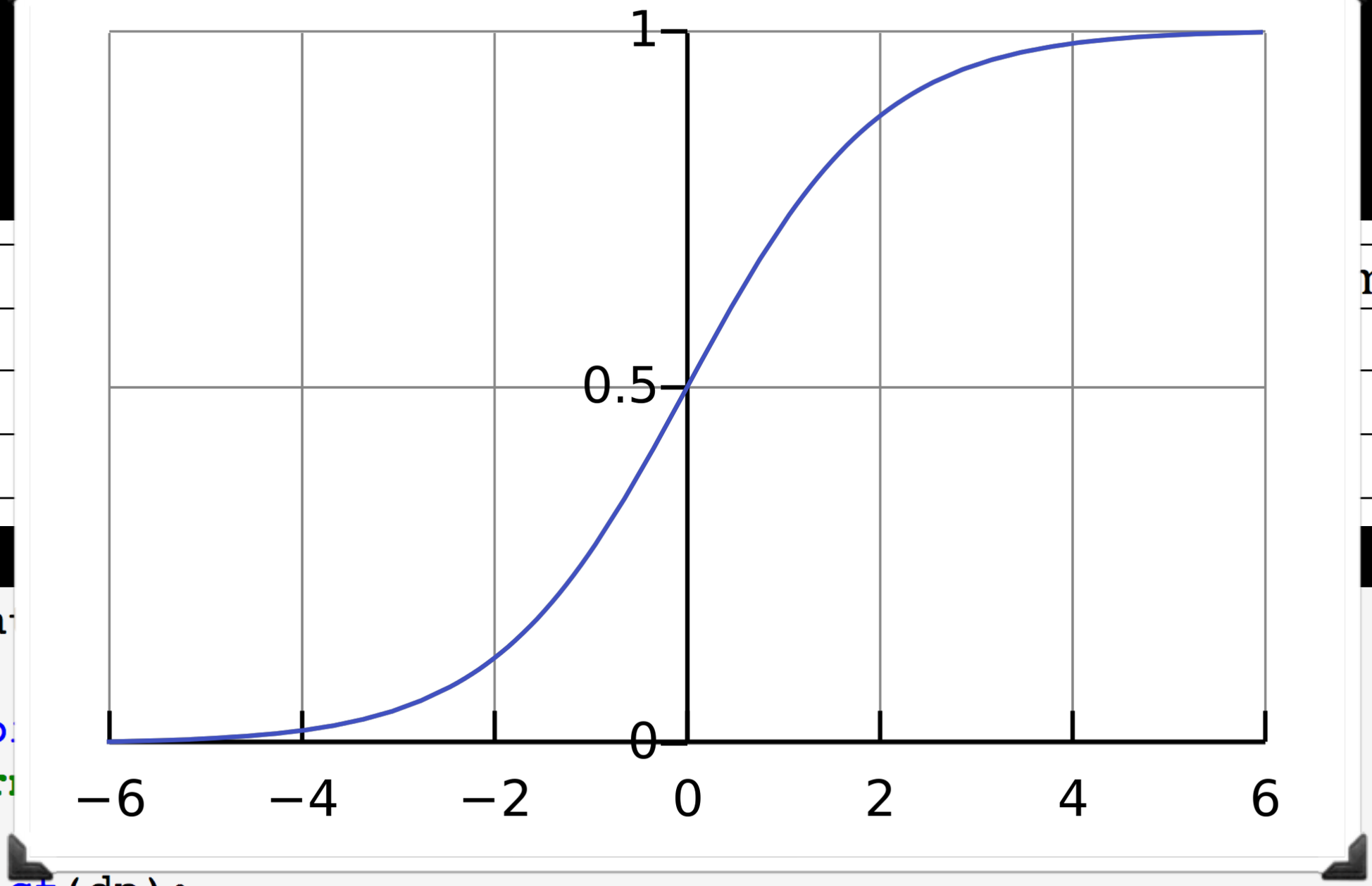
Neural Networks

Part No.	Max Temp. 1	Min Temp 1	Max Vibration 1	Outcome
100	35	35	12	1
101	46	35	21	1
130	56	46	3412	0

```
import math

def sigmoid(x):
    return 1 / (1 + math.exp(-x))

def predict(dp):
    return sigmoid(a+b*dp.MaxTemp1+c*dp.MinTemp1+d*dp.MaxVibration1)
```

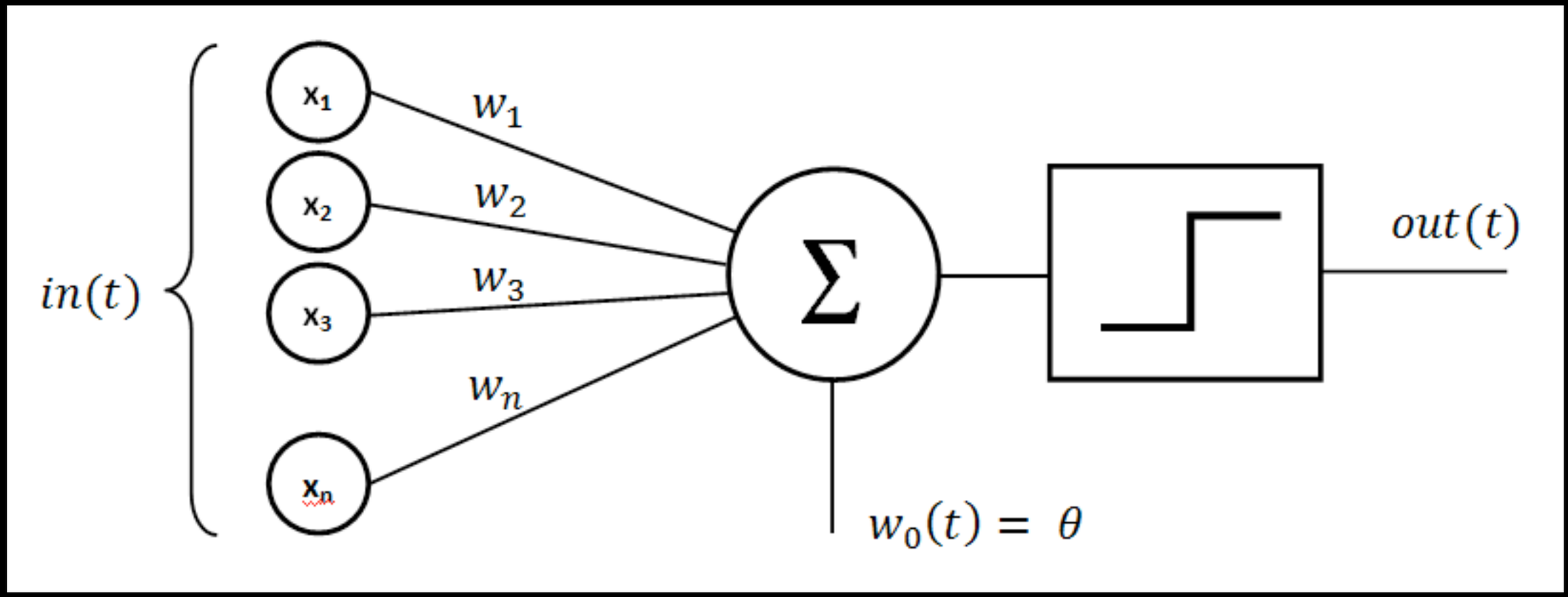
Neural Networks

Part No.								ne
100								
101								
130								

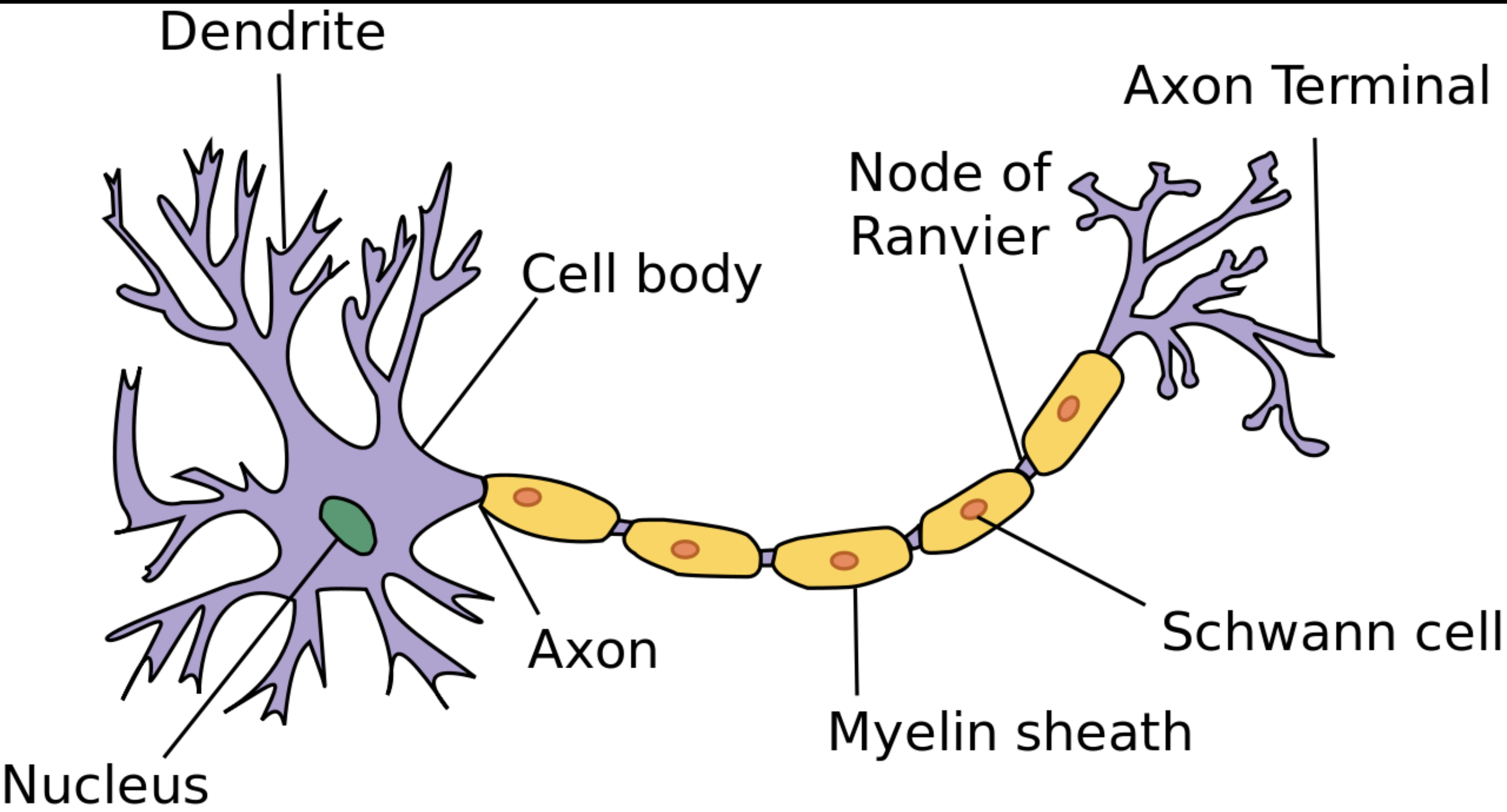



```
import ma

def sigmo
    retur

def predict(dp):
    return sigmoid(a+b*dp.MaxTemp1+c*dp.MinTemp1+d*dp.MaxVibration1)
```

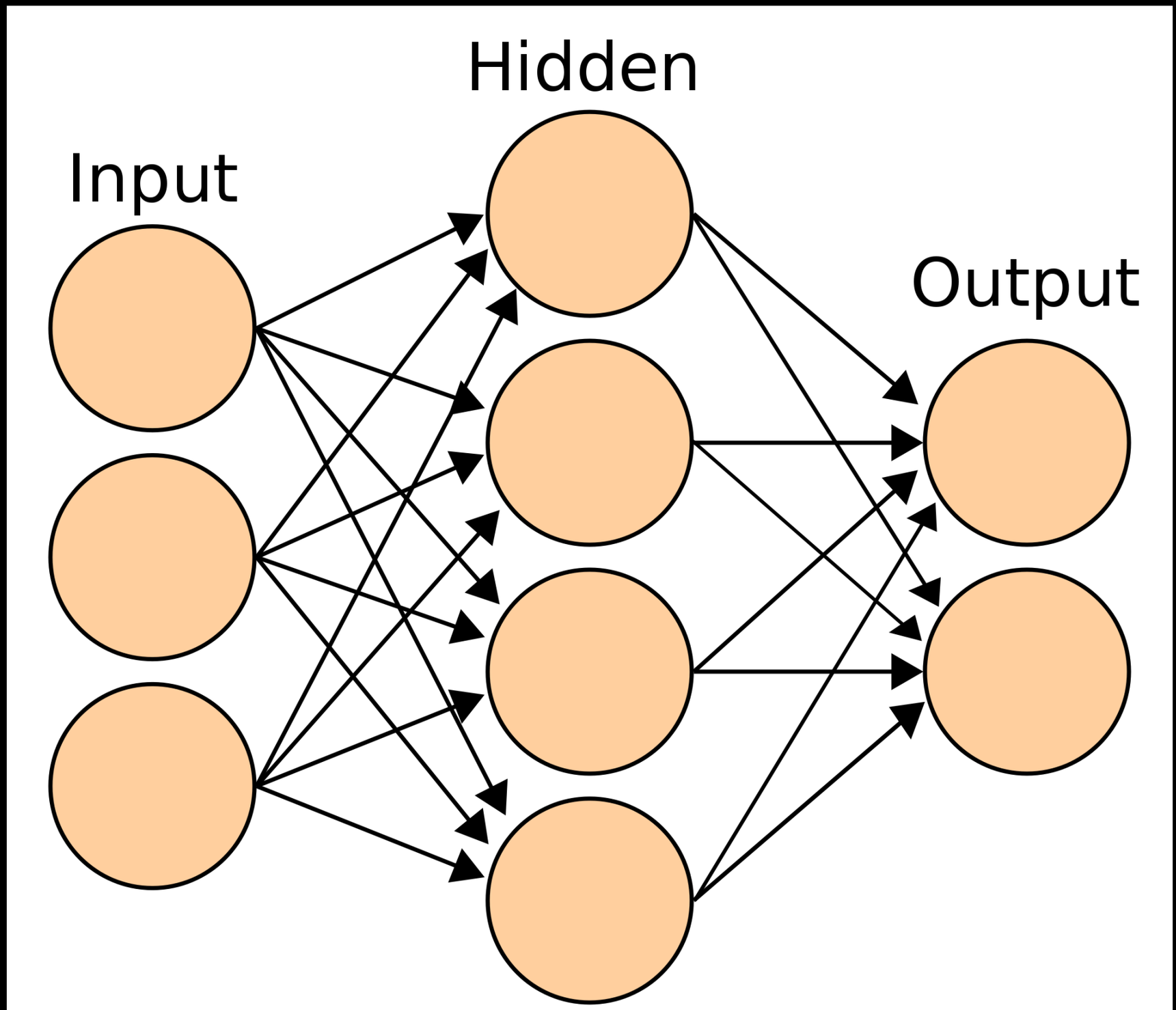
Neural Networks



Neural Networks

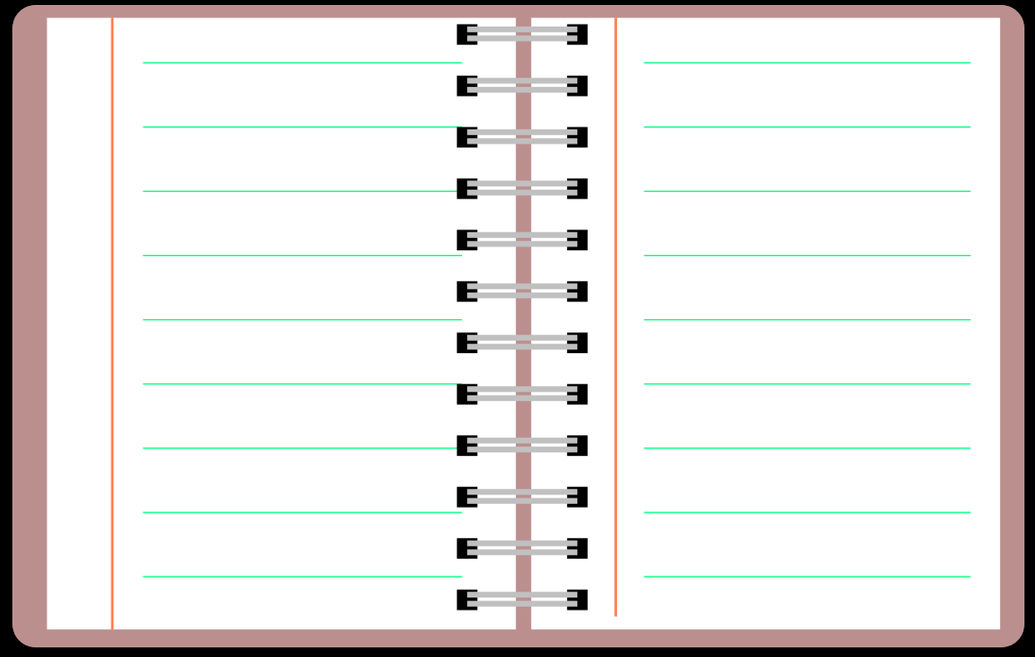


Neural Networks



Agenda

- Introduction to DeepLearning ✓
- Introduction to ApacheSpark
- Introduction to DL4J
- Introduction to SystemML
- Demos



JVM
Process

Driver JVM

Compute Node

Driver JVM

Compute Node

Executor JVM

Driver JVM

Compute Node

Executor JVM

Executor JVM

Executor JVM

Driver JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Driver JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Driver JVM

Compute Node

Executor JVM

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Executor JVM

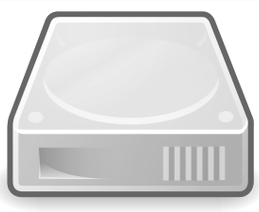
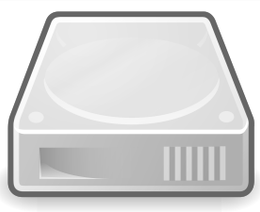
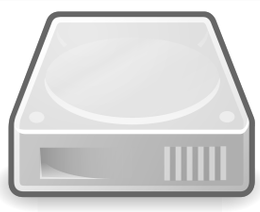
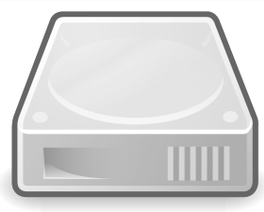
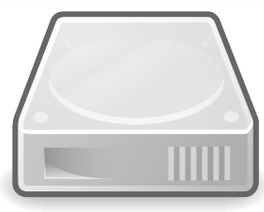
Compute Node

Executor JVM

Executor JVM

Executor JVM

HDFS

Compute Node	Compute Node	Compute Node	Compute Node	Compute Node
Executor JVM	Executor JVM	Executor JVM	Executor JVM	Executor JVM
Executor JVM	Executor JVM	Executor JVM	Executor JVM	Executor JVM
Executor JVM	Executor JVM	Executor JVM	Executor JVM	Executor JVM
Executor JVM	Executor JVM	Executor JVM	Executor JVM	Executor JVM
				
Part of File	Part of File	Part of File	Part of File	Part of File

HDFS

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM

Compute Node

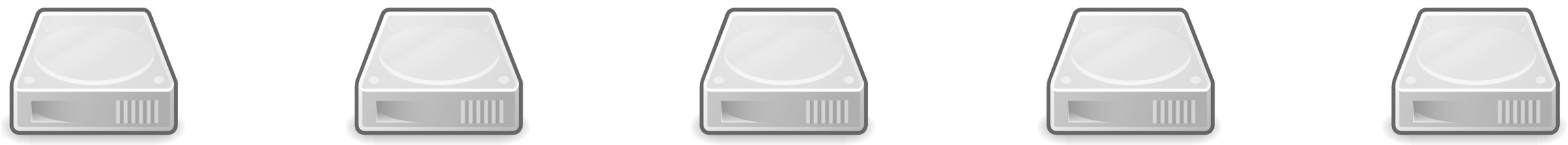
Executor JVM

Executor JVM

Compute Node

Executor JVM

Executor JVM



Part of File | Part of File | Part of File | Part of File | Part of File

Virtual File

RDD

- “Resilient Distributed Dataset“
- Distributed, immutable collection of data
- Created from HDFS, ObjectStore, Cloudant NoSQL, dashDB SQL, simple files, …
- In-memory, but spillable to disk
- lazy

Summary

- ApacheSpark programs are implicitly parallel
- Same code can process 1 KB or 1 PB
- RDD central API
- Data and task distribution transparent

Programming Language options on ApacheSpark

Summary

	Scala	Java	R	Python
Spark API support	complete	complete	very limited	limited
ease of use	low	very low	high	very high
Speed	very high	high	very low	low
3rd party libraries	few	few	many	many

Functional Programming basics

Functional Programming (FP)

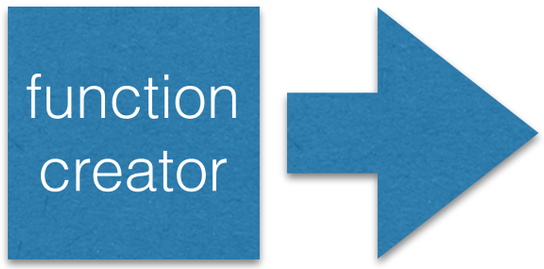
- mathematical concept of “Lambda Calculus
- 1st implementation was LISP in the 50s
- Haskell is part of every computer science curriculum since the 90s
- Scala most recent representative
- Python, R and Java also rudimentary support FP

Idea

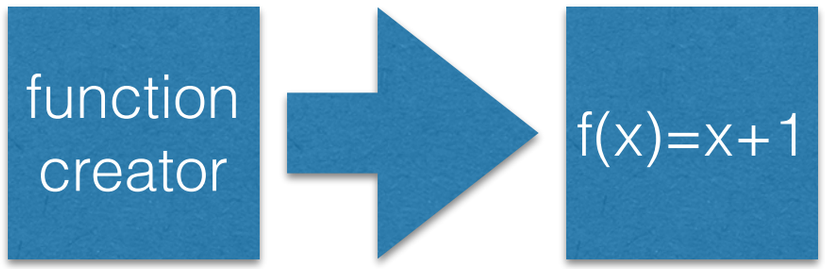
Idea

function creator

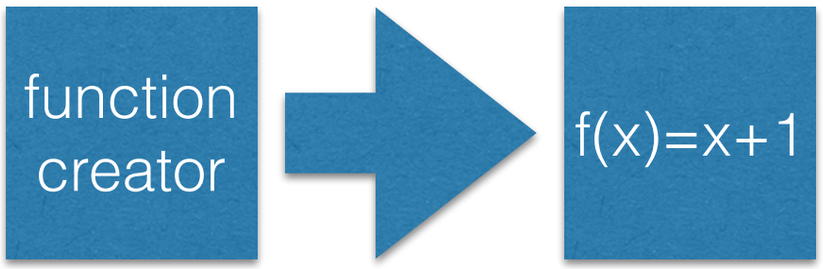
Idea

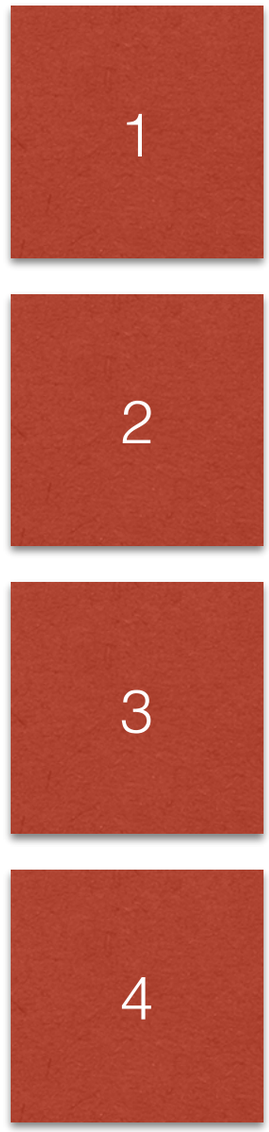


Idea

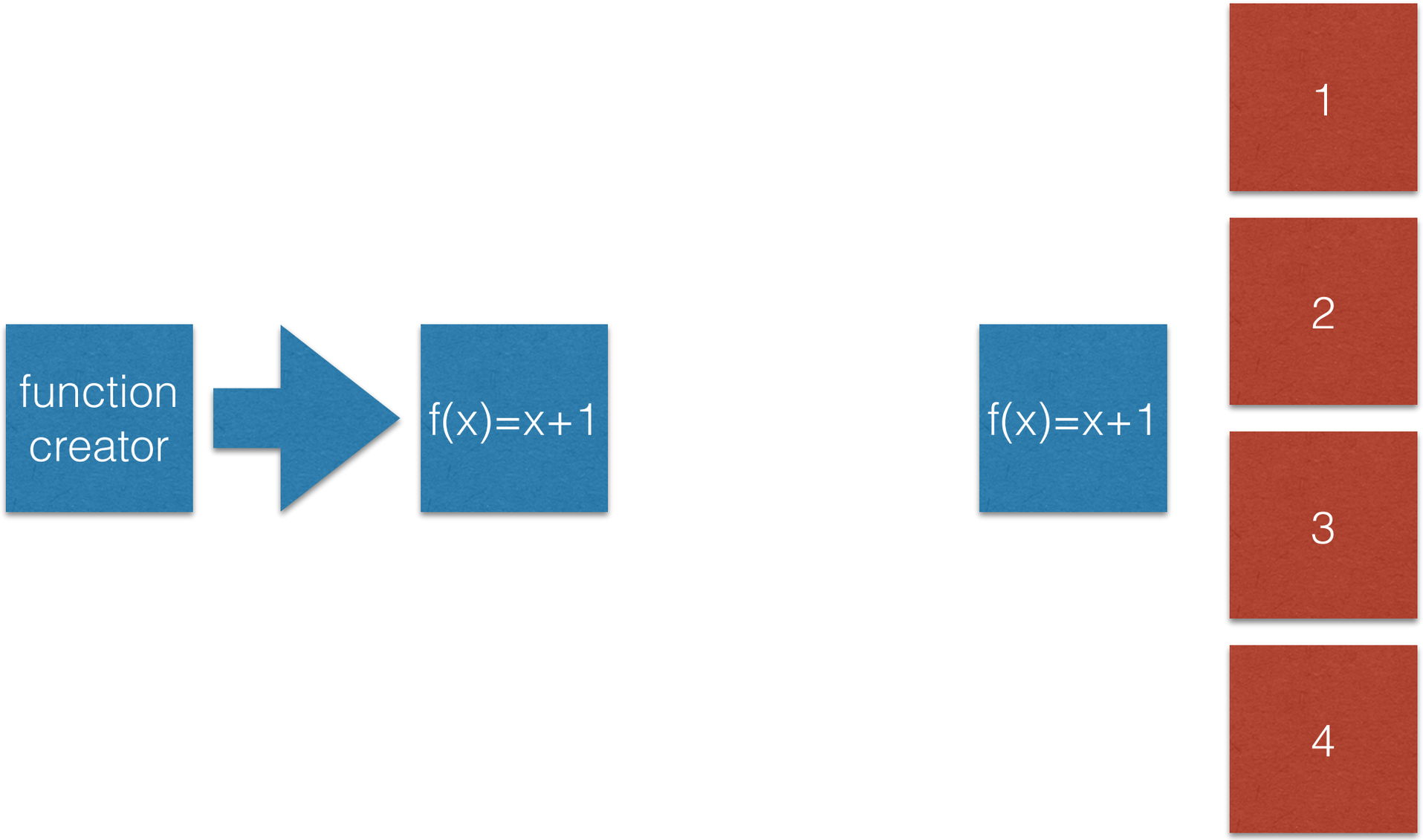


Idea

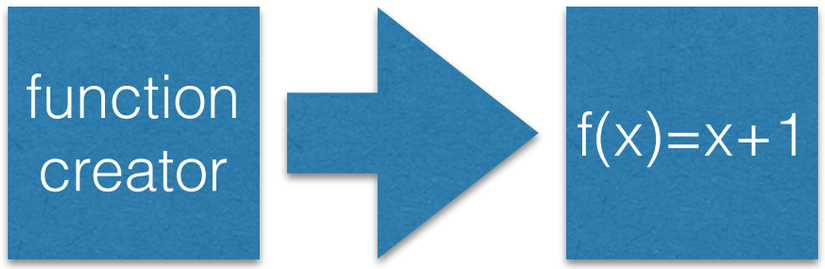


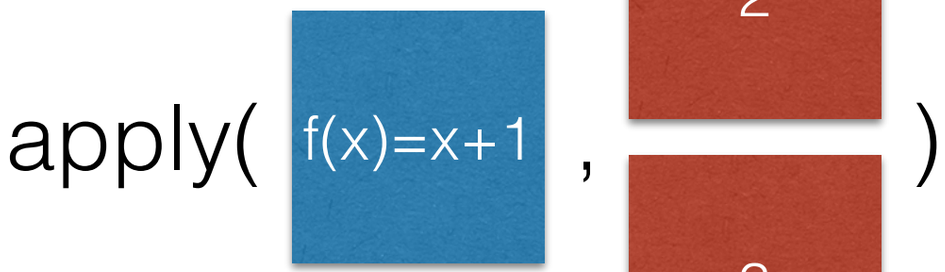


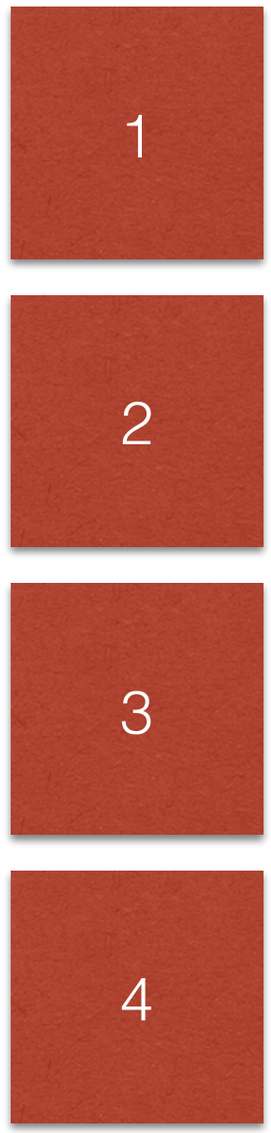
Idea



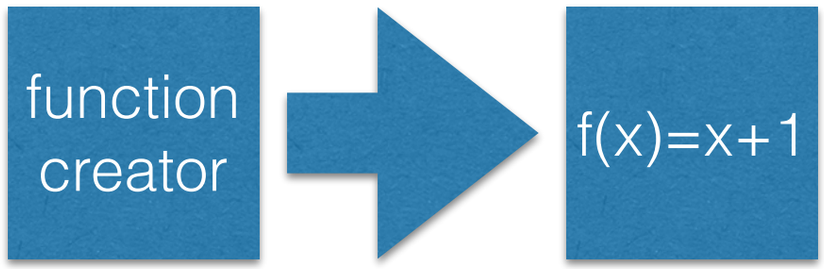
Idea

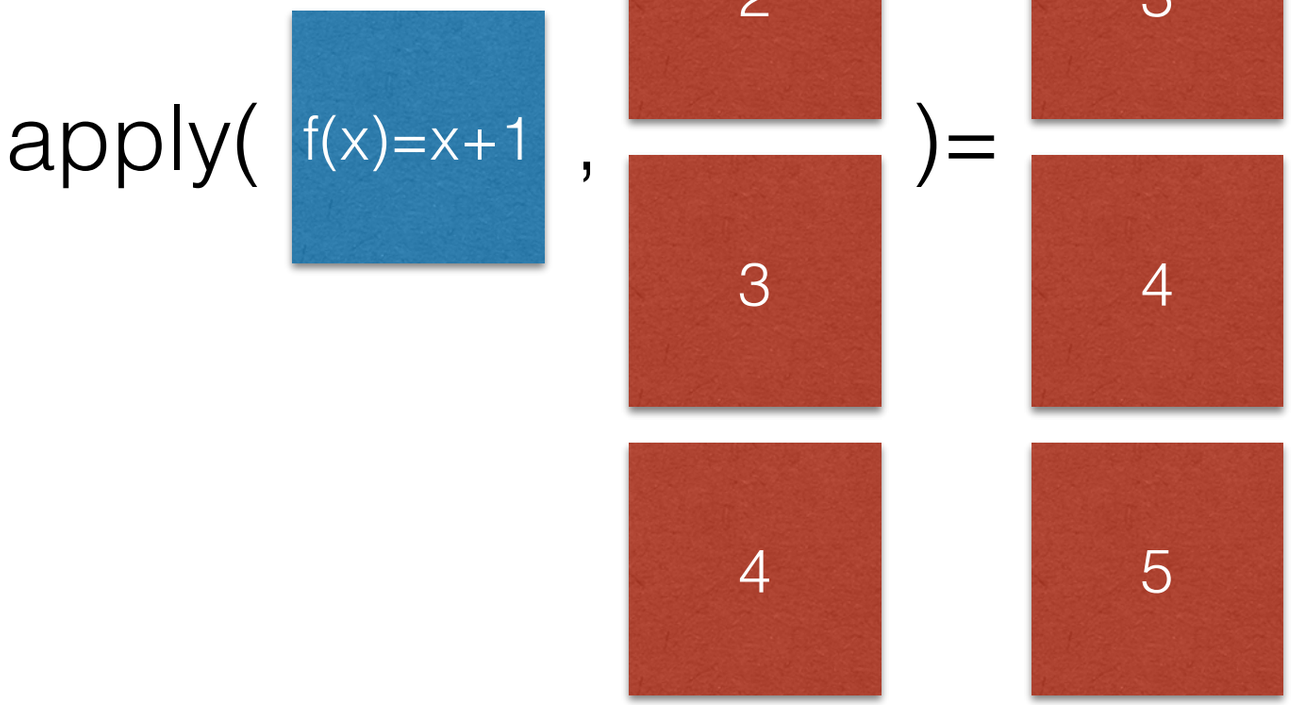






Idea





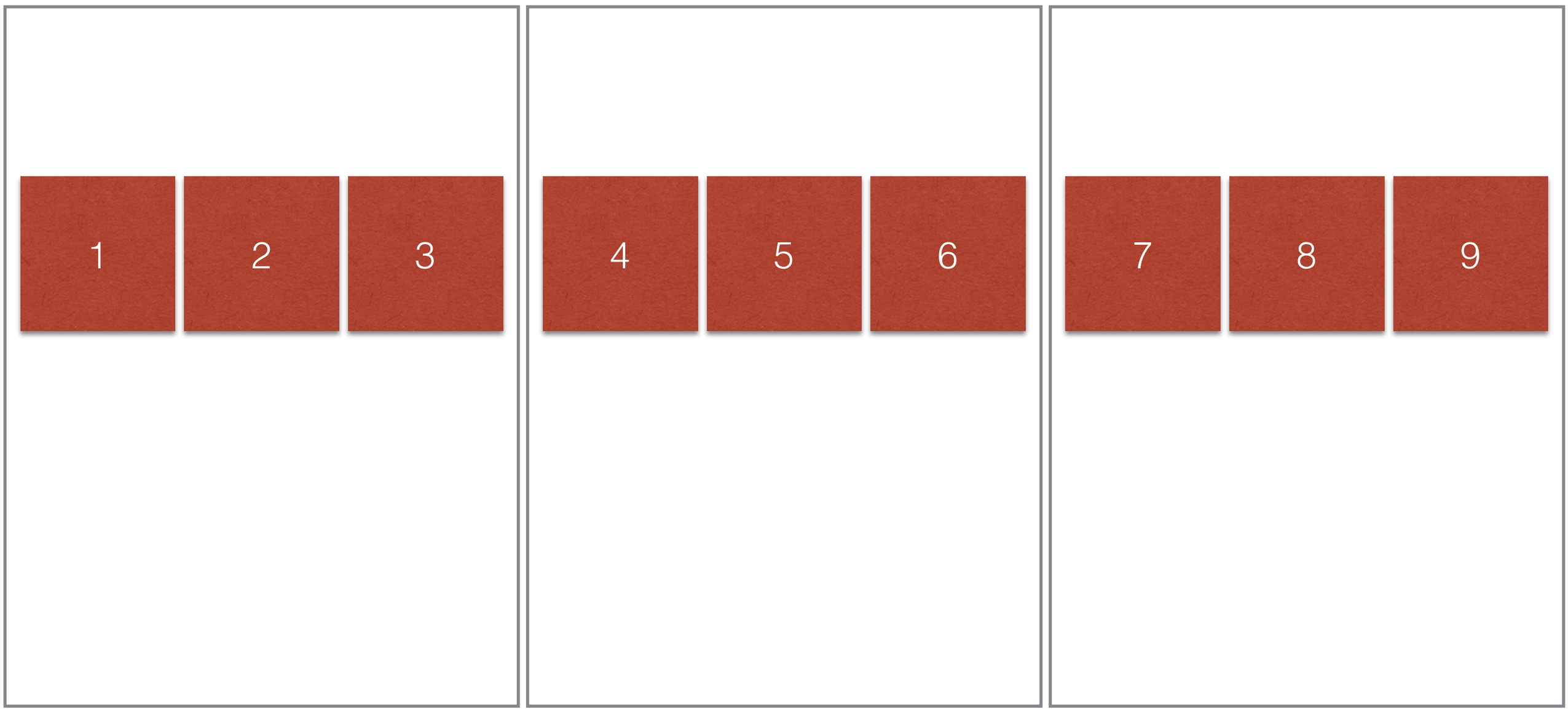
Parallelisation

f(x)=x+1

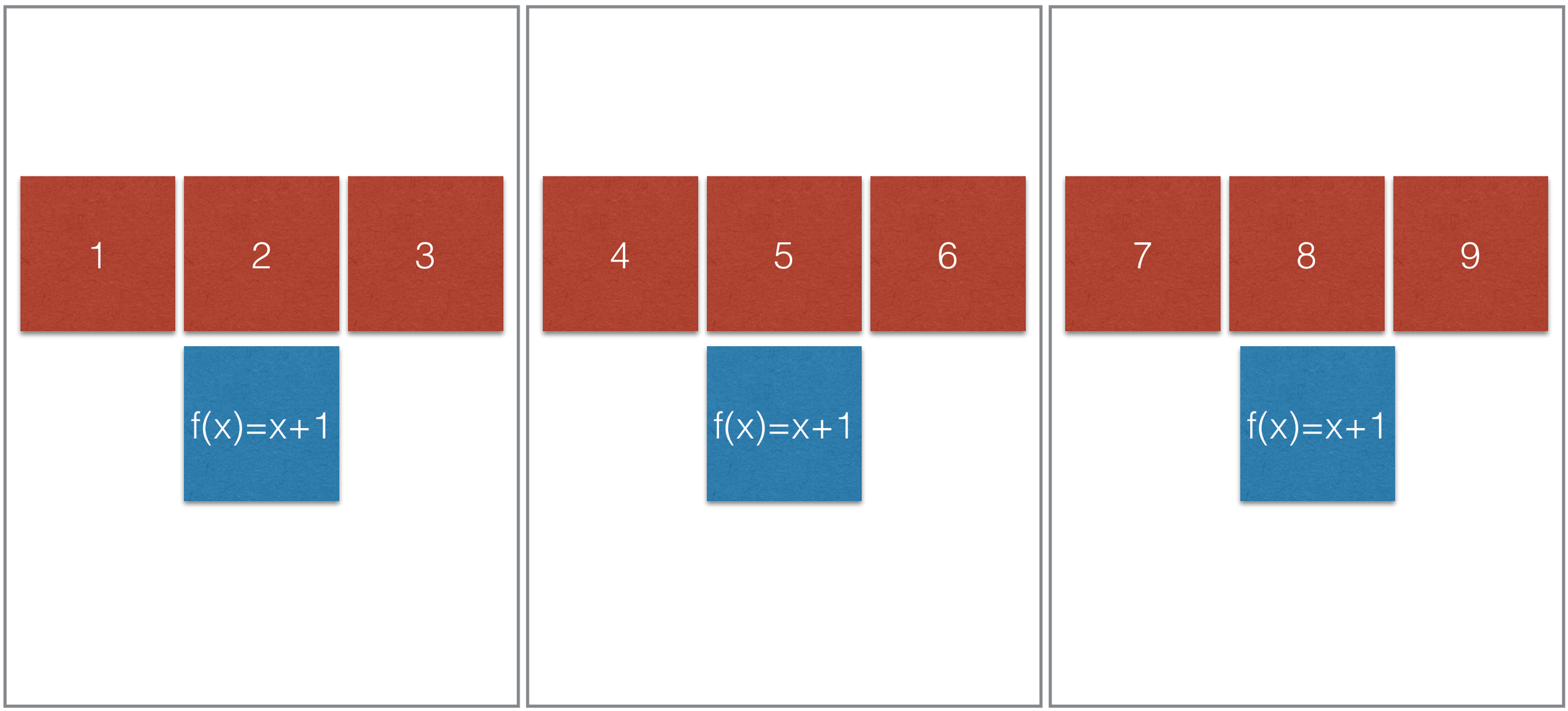
Parallelisation

f(x)=x+1

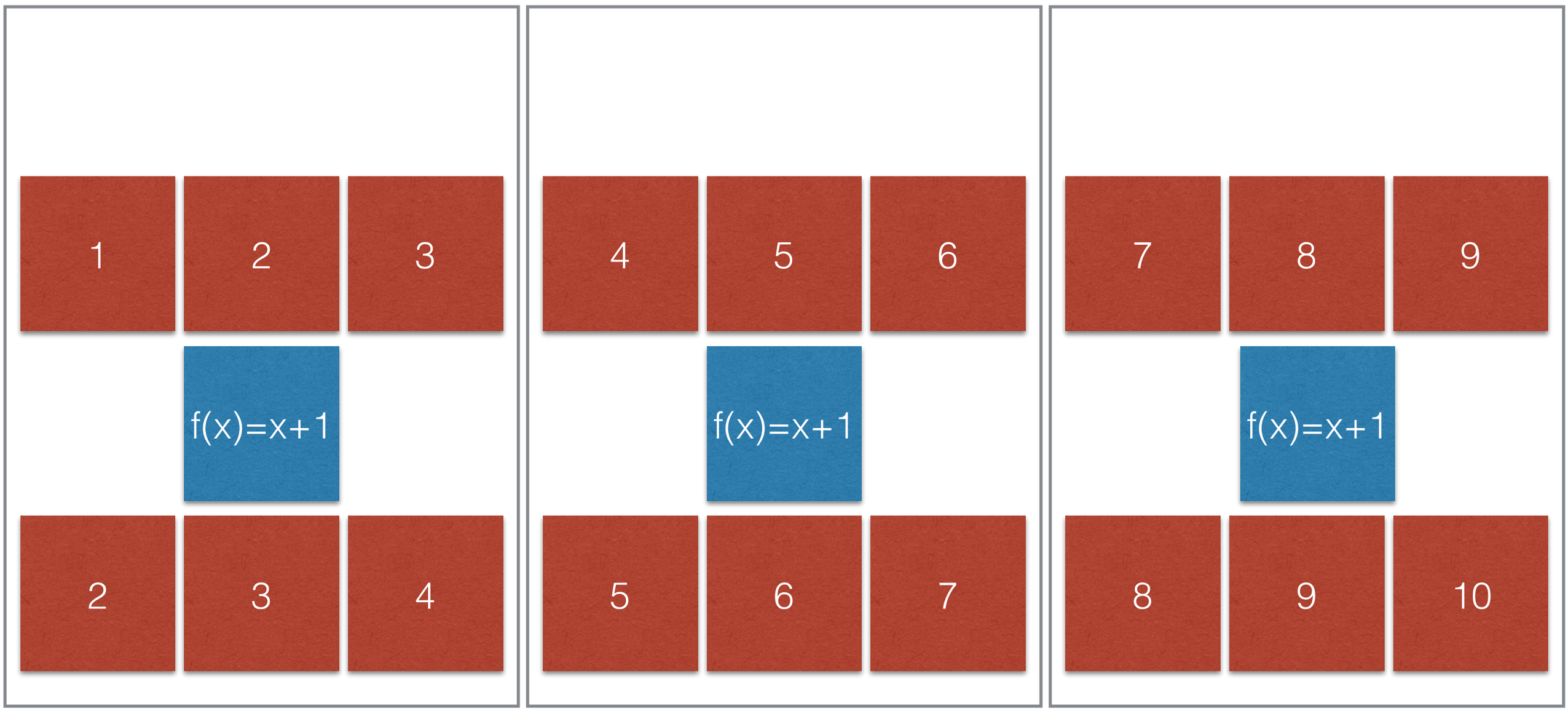
Parallelisation



Parallelisation



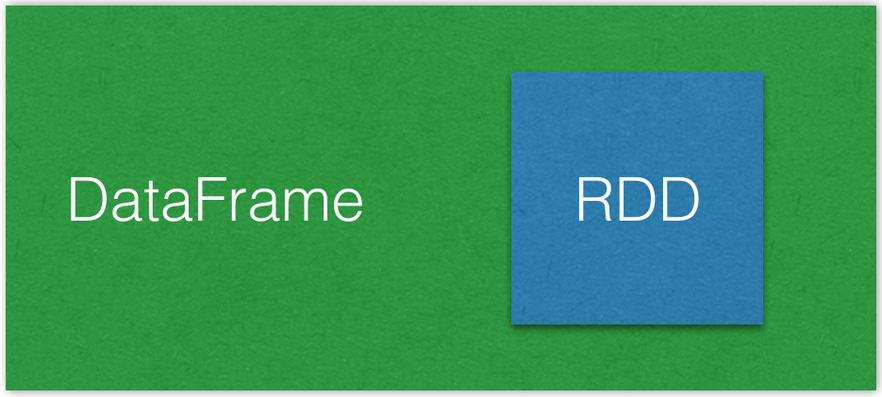
Parallelisation



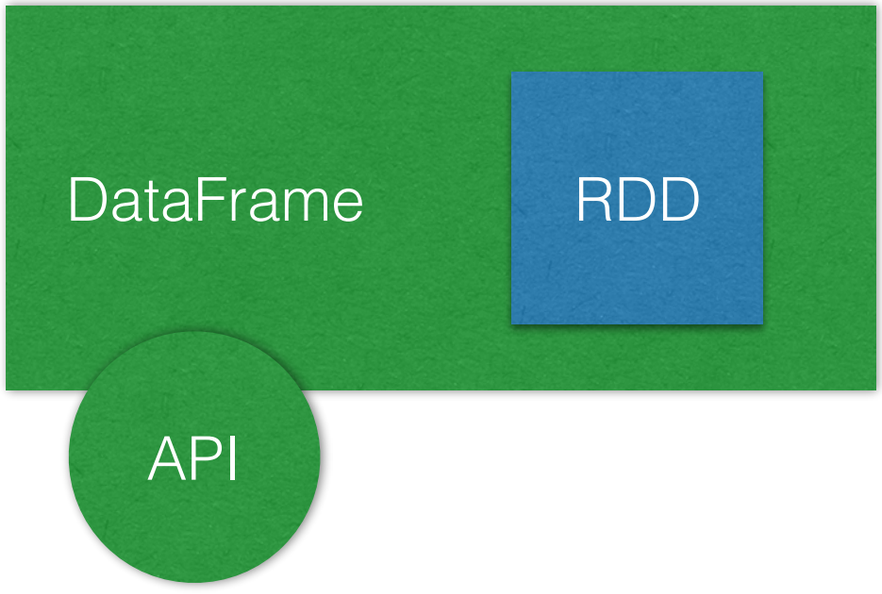
ApacheSparkSQL

ApacheSparkSQL

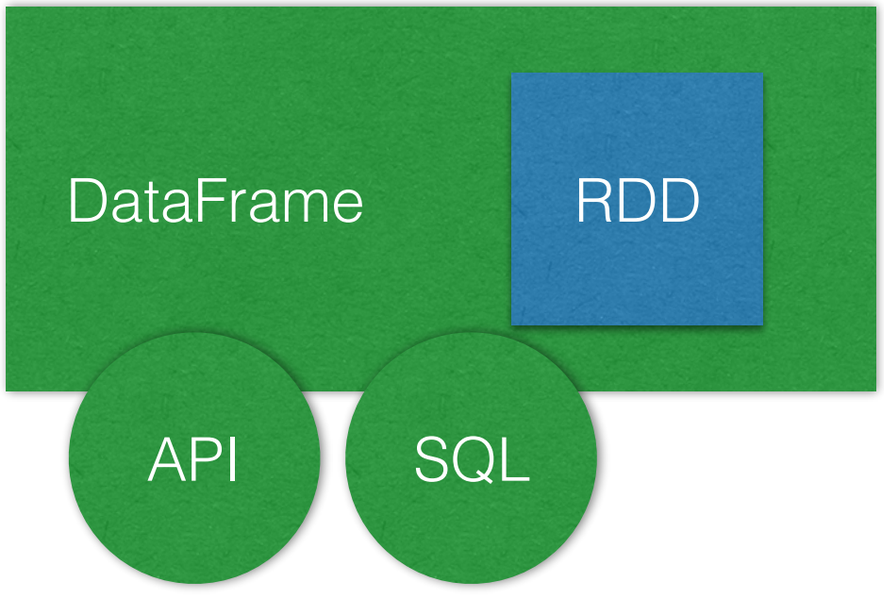
ApacheSparkSQL



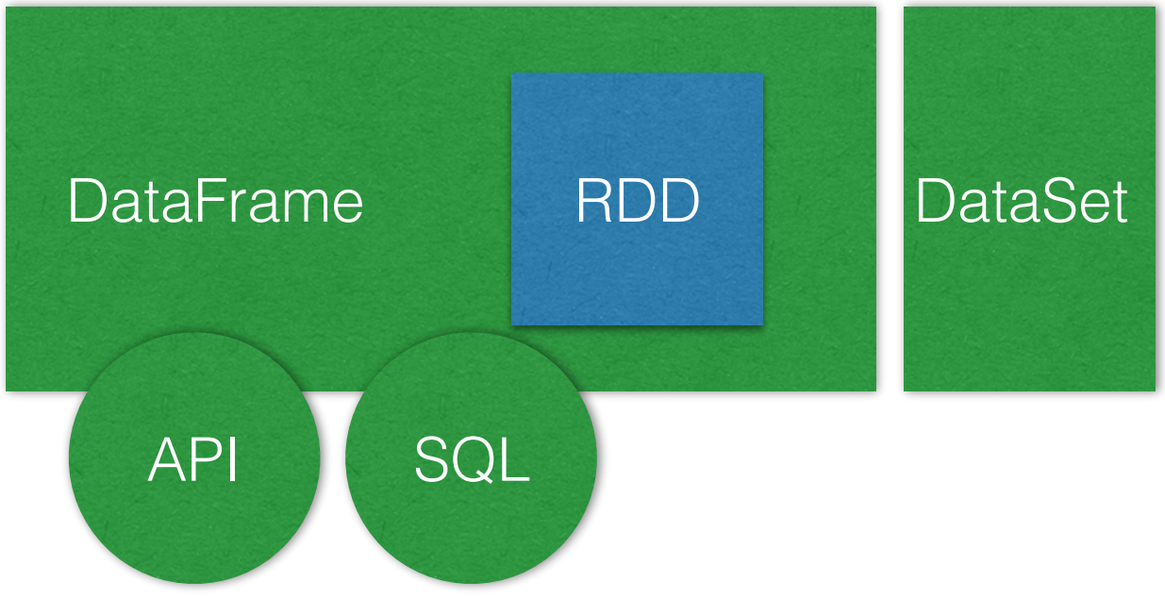
ApacheSparkSQL



ApacheSparkSQL



ApacheSparkSQL



Schemas

- RDDs are schema less (schema on read)
- DataFrames have a schema
  - lazy, inferred
  - explicitly defined

The “Catalyst”

- Creates “logical execution plan” (LEP) from SQL
- Optimises LEP to “physical execution plans” (PEPs)
  - based on statistics chooses best PEP to execute
  - similar to cost based optimisers in RDBMs

Project Tungsten

- Java Virtual Machine (JVM) is an art piece
- General purpose byte code execution engine
- JVM objects & Garbage Collection (GC) overhead
  - 4 byte string is 48 byte on the JVM
  - GC optimises on object life time estimation
  - Spark knows this better than JVM

Project Tungsten

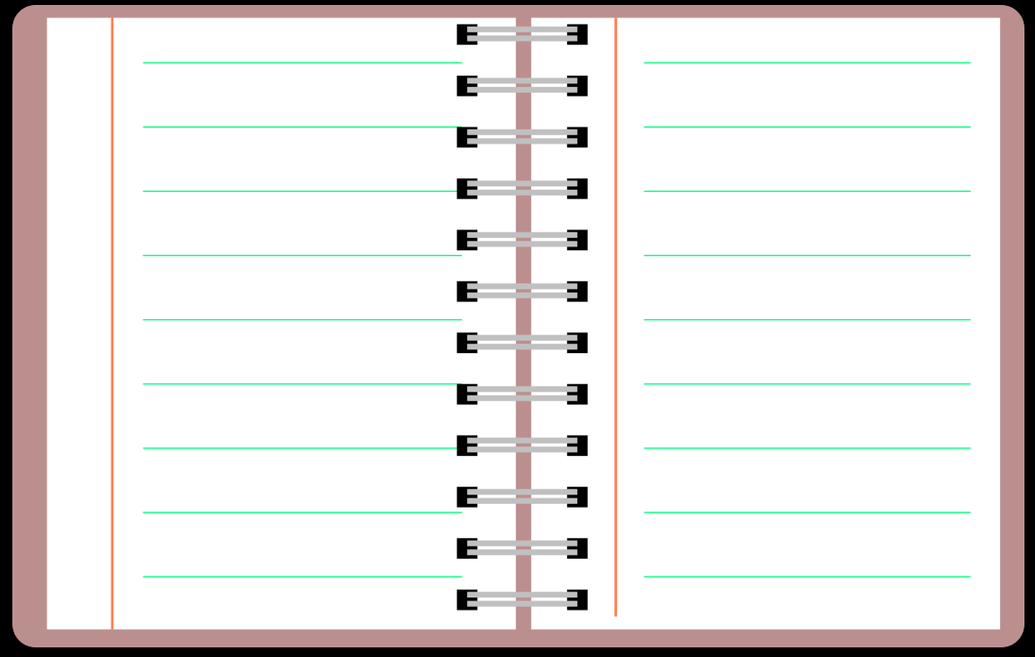
- L1/L2/L3 Cache friendly data structures
- Code generation to remove
  - boxing of primitive types
  - polymorphic function dispatching

Summary

- ApacheSpark supports SQL via data frame API
- Internally still RDDs are used
- Makes writing ApacheSpark jobs easier
- Performance benefits through Catalyst & Tungsten

Agenda

- Introduction to DeepLearning ✓
- Introduction to ApacheSpark ✓
- Introduction to DL4J
- Introduction to SystemML
- Demos



DeepLearning4J

Components

- DeepLearning4J
  Enterprise Grade DeepLearning Library

- DataVec
  CSV/Audio/Video/Image/… => Vector

- ND4J / ND4S (NumPy for the JVM)

ND4J

- Tensor support (Linear Buffer + Stride)
- Multiple implementations, one interface
  - vectorized c++ code (JavaCPP), off-heap data storage, BLAS (OpenBLAS, Intel MKL, cuBLAS)
  - GPU (CUDA 7.5)

turn on GPU

```
    <name>DeepLearning4j Examples Parent</name>
    <description>Examples of training different data sets</description>
    <properties>
        <nd4j.backend>nd4j-native-platform</nd4j.backend>
```

```
    <nd4j.backend>nd4j-cuda-7.5-platform</<nd4j.backend>
```

DL4J parallelisation

- TensorFlow on ApacheSpark =>
  - Scoring
  - Multi-model hyper-parameter tuning
  - Parallel training since V r0.8
- DeepLearning4J =>
  - Scoring, Multi-model hyper-parameter tuning
  - Parallel training
    “Jeff Dean style parameter averaging”

"Code local vs spark"

```
new MultiLayerNetwork(conf);
```

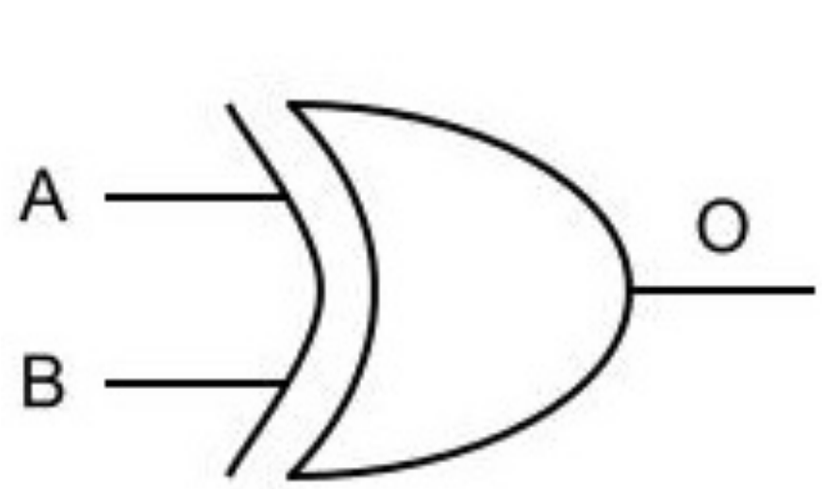
vs.

```
new SparkDl4jMultiLayer(sc, conf, tm);
```

```
TrainingMaster tm = new ParameterAveragingTrainingMaster.Builder(int dataSetObjectSize)
        ... (your configuration here)
        .build();
```

XOR

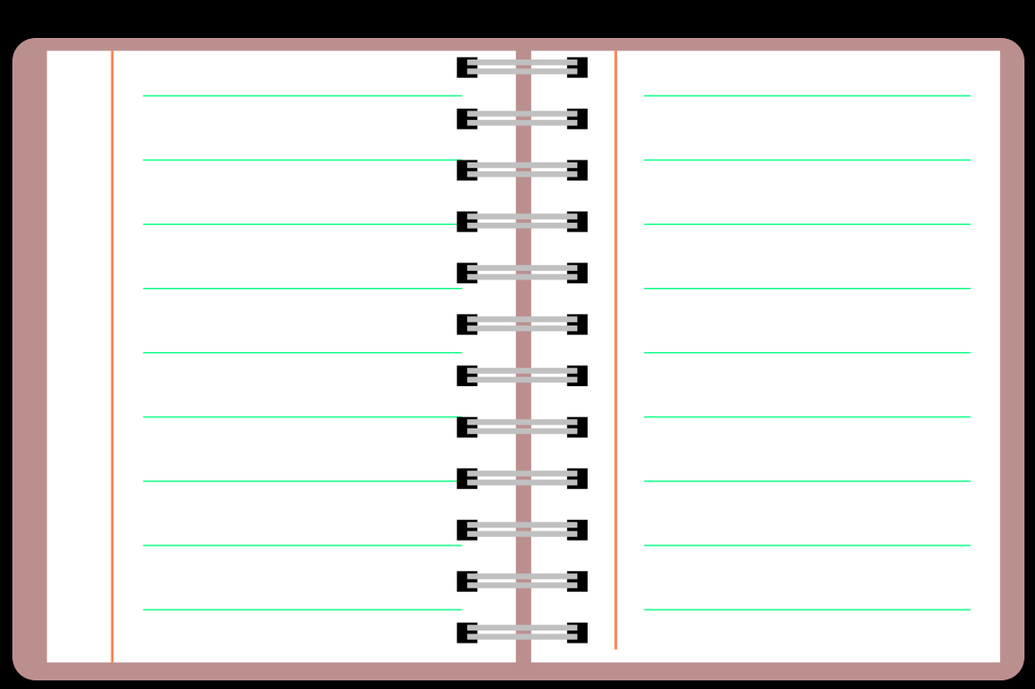
https://github.com/romeokienzler/dl4j-examples/blob/master/dl4j-examples-scala/src/main/scala/org/deeplearning4j/examples/feedforward/xor/XOrExampleScala.scala



A	B	O
0	0	0
0	1	1
1	0	1
1	1	0

Agenda

- Introduction to DeepLearning ✓
- Introduction to ApacheSpark ✓
- Introduction to DL4J ✓
- Introduction to SystemML
- Demos



Why another lib?

- Custom machine learning algorithms
- Declarative ML
- Transparent distribution on data-parallel framework
  - Scale-up
  - Scale-out
- Cost-based optimiser generates low level execution plans

Why on Spark?

- Unification of SQL, Graph, Stream, ML
- Common RDD structure
- General DAG execution engine
  - lazy evaluation
  - distributed in-memory caching

2007-2008: Multiple projects at IBM Research – Almaden involving machine learning on Hadoop.

2009: We form a **dedicated team** for scalable ML

2009-2010: Through engagements with customers, we observe how data scientists create **ML solutions**.

2007 2008 2009 2010

Research

2011 2012 2013 2014

June 2015: IBM Announces open-source SystemML

September 2015: Code available on Github

November 2015: SystemML enters Apache incubation

February 2016: First release (0.9) of Apache SystemML

June 2016: Second Apache release (0.10)

2015

2016

SystemML at IBM Watson Health

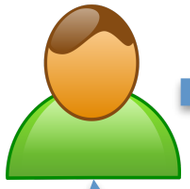
Moved from Hadoop MapReduce to Spark

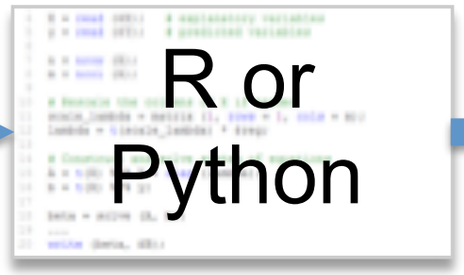
SystemML supports both frameworks

Exact same code

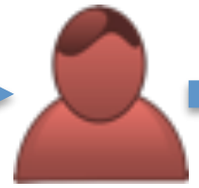
300X faster on 1/40th as many nodes

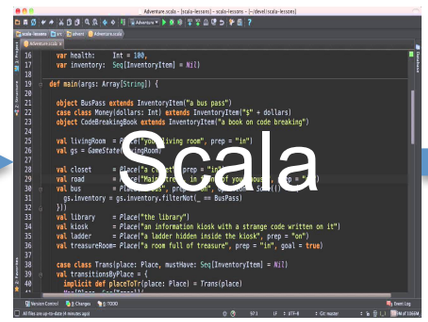
Data Scientist

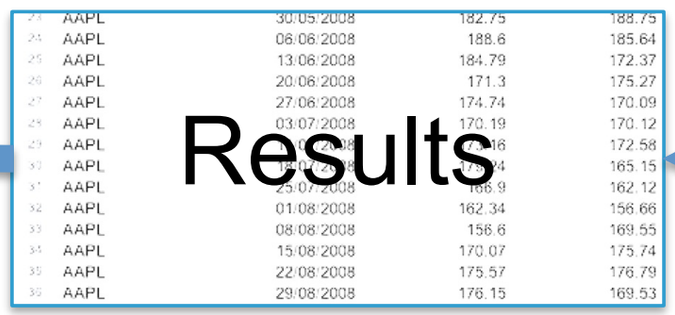




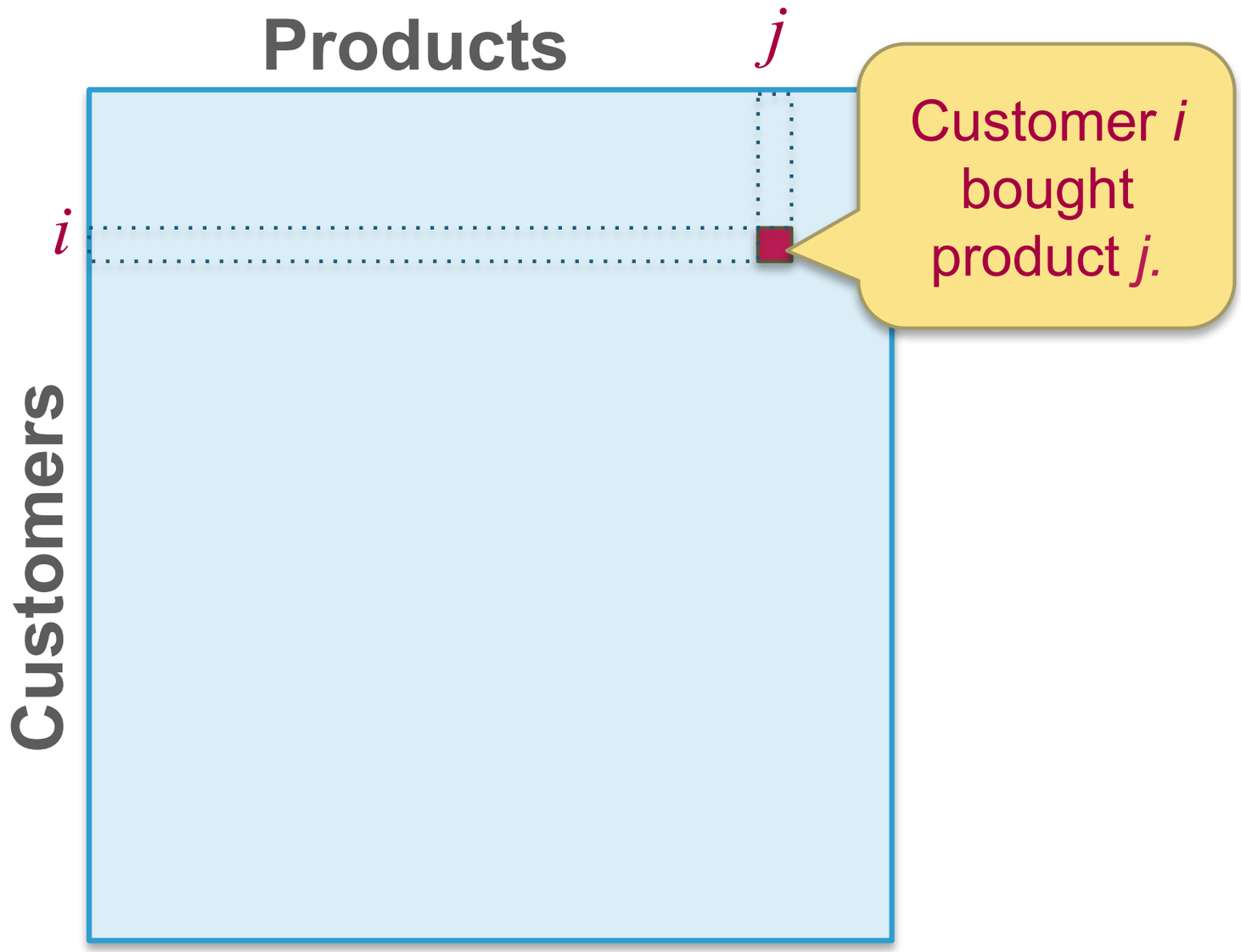
Systems Programmer



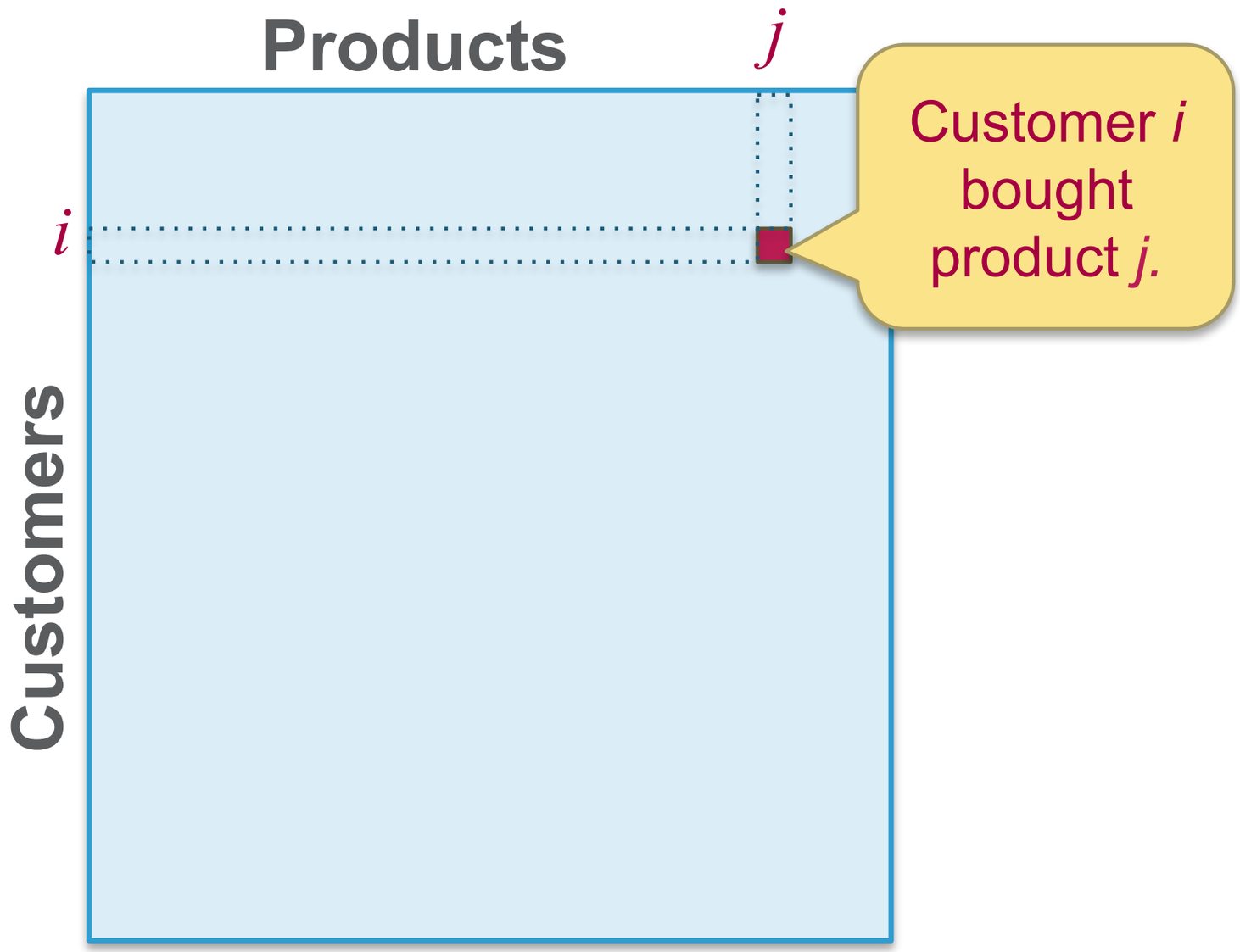




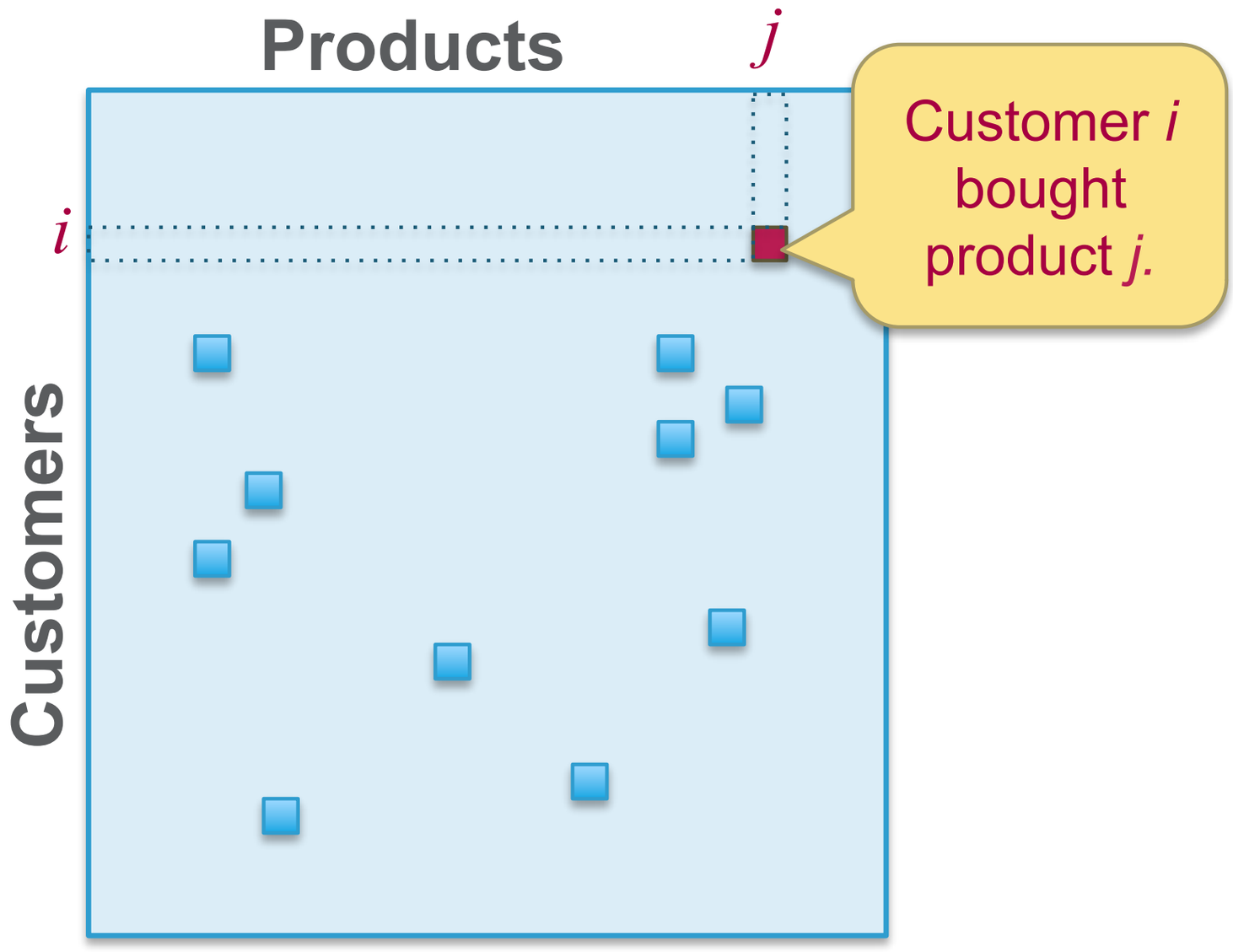
Alternating Least Squares



Alternating Least Squares



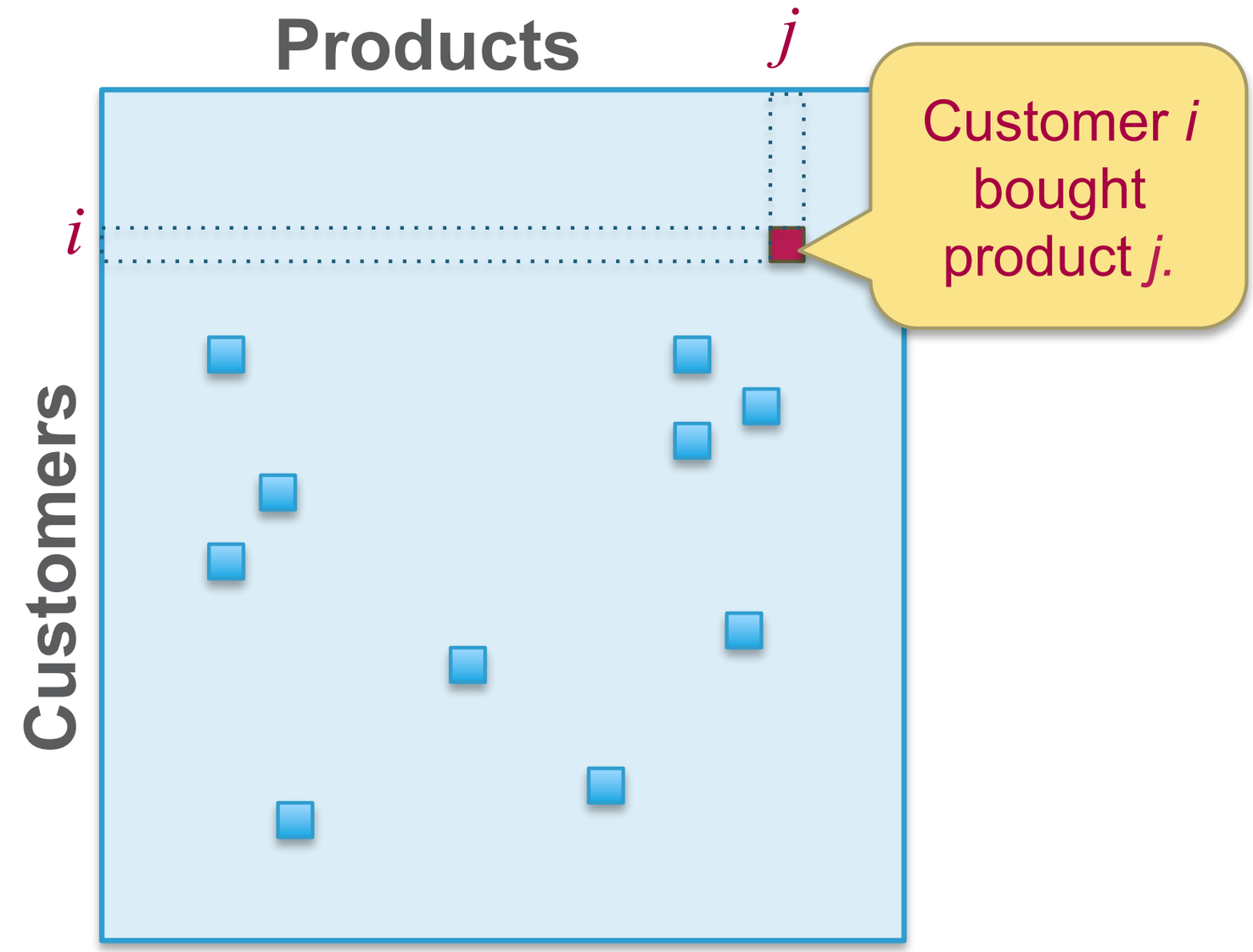
Alternating Least Squares



Products Factor

Customers Factor

p_u



Products Factor

q_i

Customers Factor

p_u

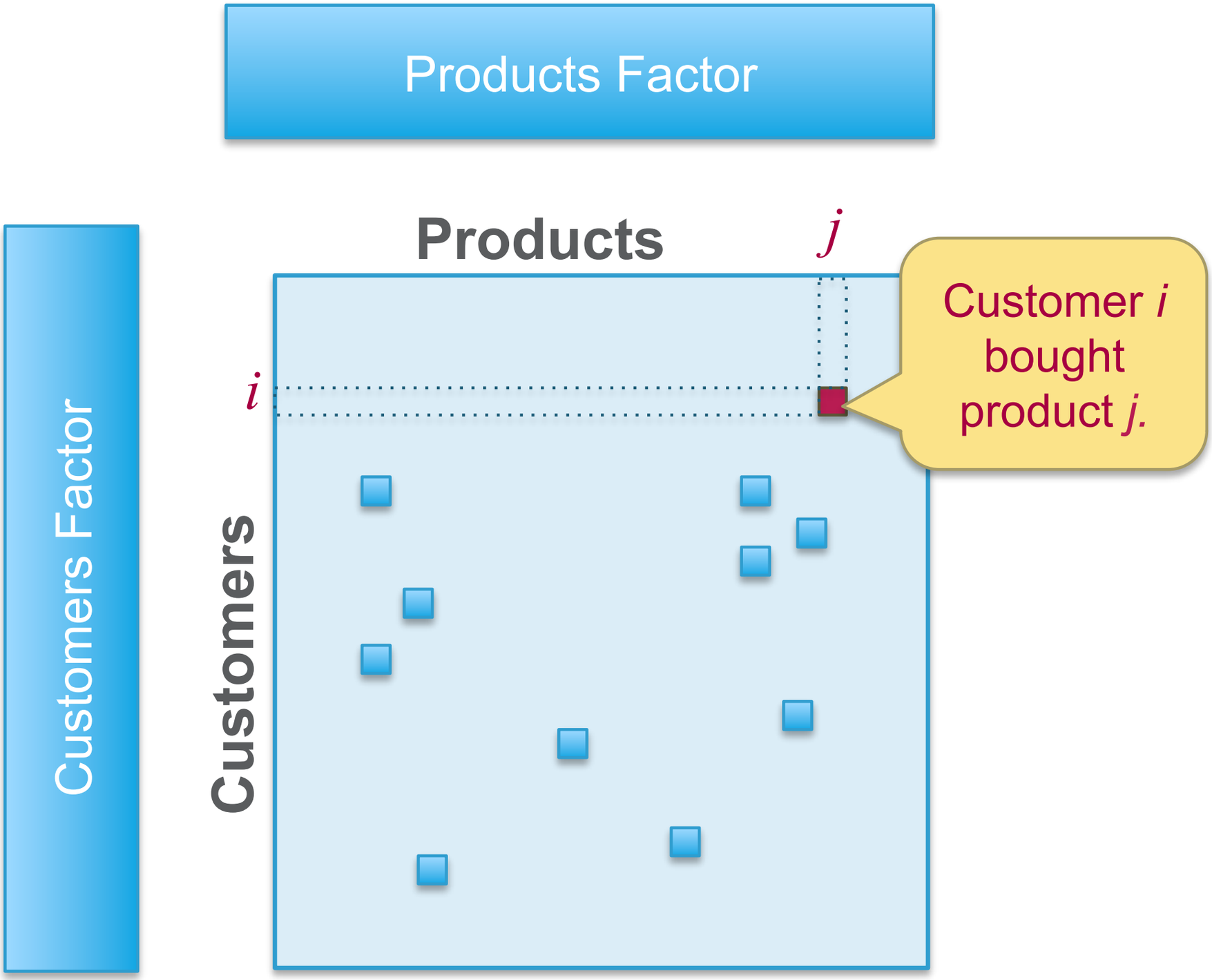
Products

Customers

j

i

Customer i bought product j.



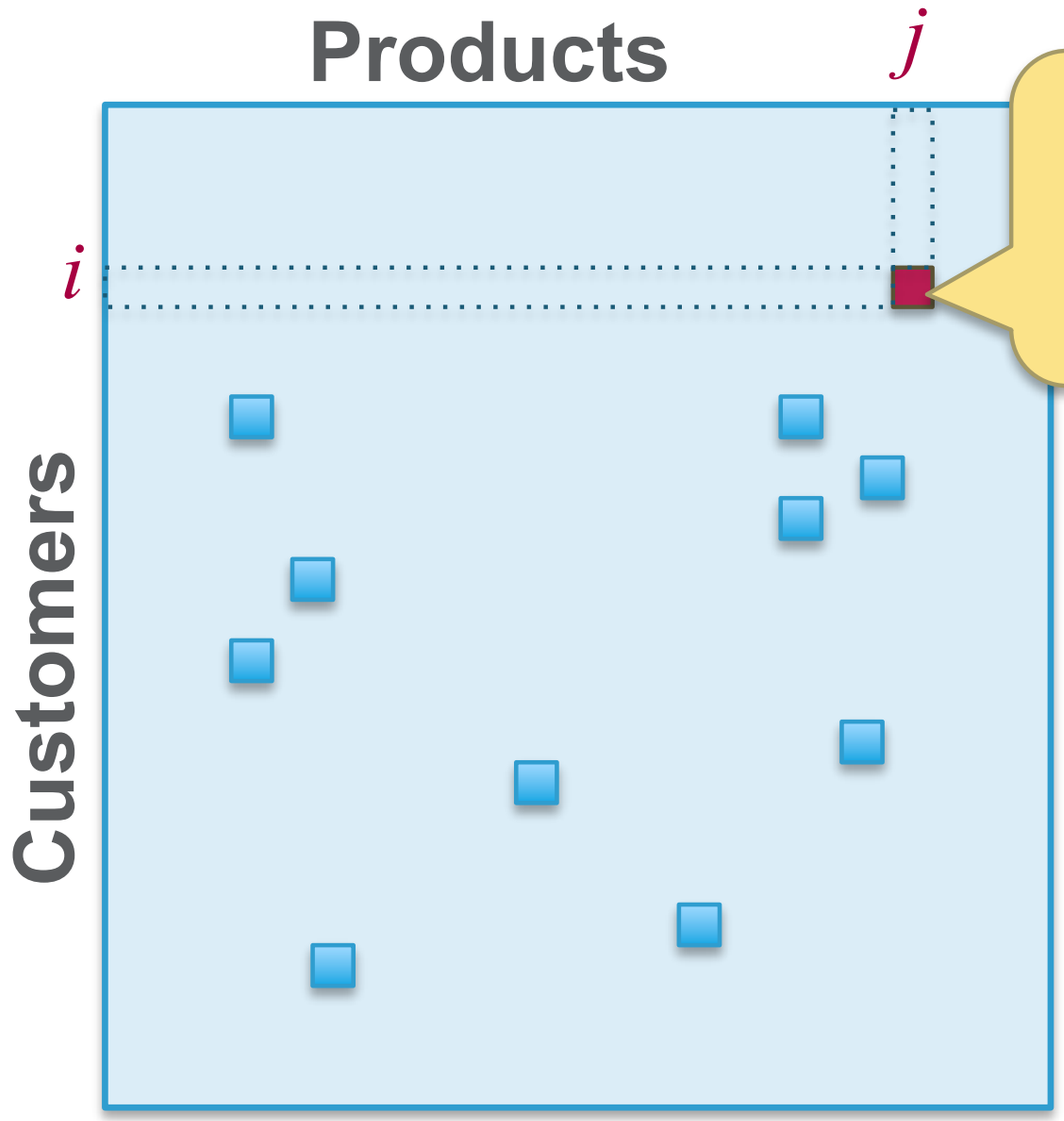
$$min_{q,p} \sum_{u,i} (r_{ui} - p_u^T q_i)^2$$

Multiply these two factors to produce a less-sparse matrix.

×

Products Factor

Customers Factor



$$r'_{ui} = p_u^T q_i$$

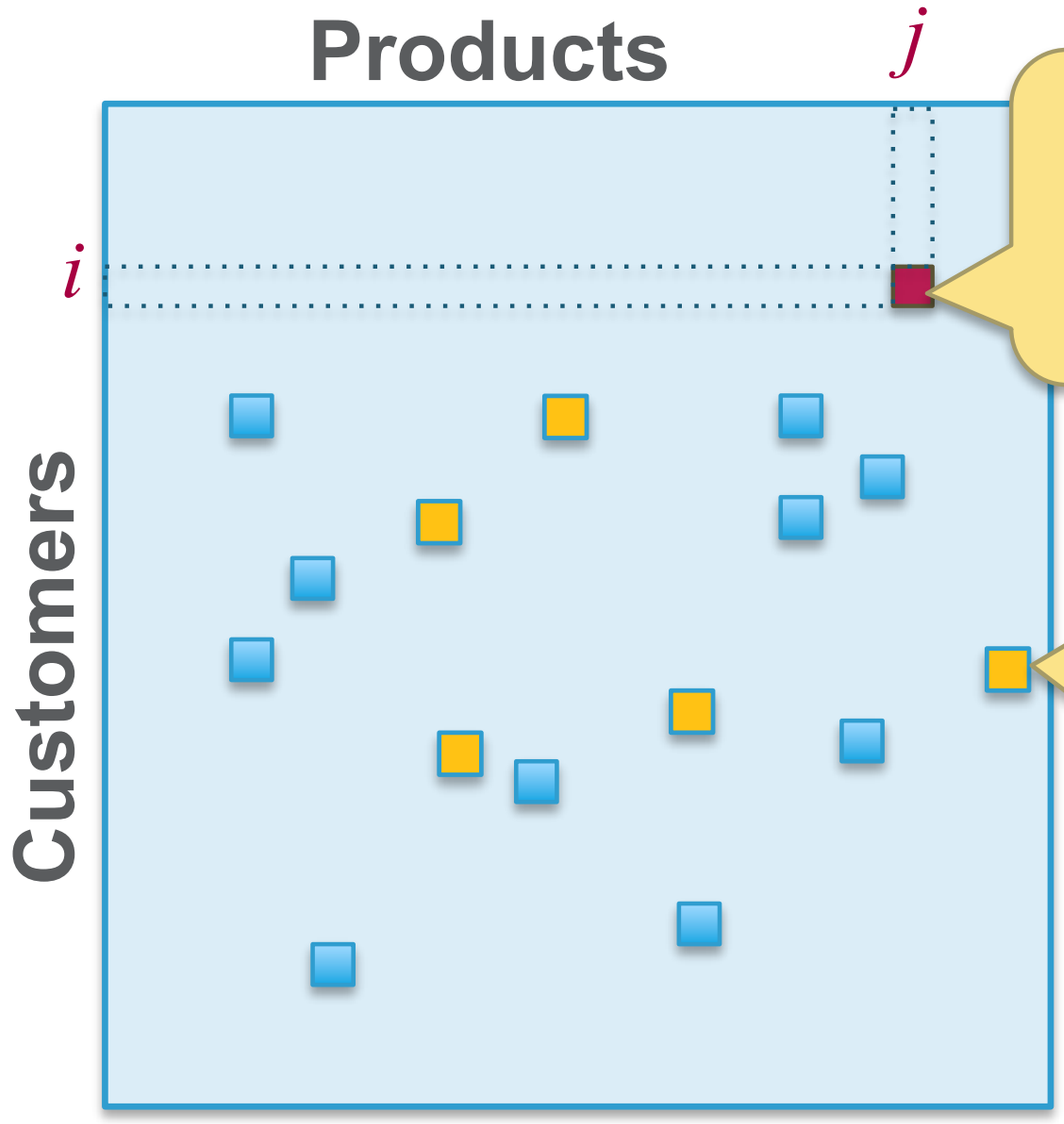
$$min_{q,p} \sum_{u,i} (r_{ui} - p_u^T q_i)^2$$

Multiply these two factors to produce a less-sparse matrix.

×

Products Factor

Customers Factor



Customer *i* bought product *j*.

New nonzero values become product suggestions.

$$r'_{ui} = p_u^T q_i$$

```
val model = ALS.train(ratings, rank, numIterations, 0.01)
```

```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

Every line has a clear purpose!

https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/recommendation/ALS.scala

https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/recommendation/ALS.scala

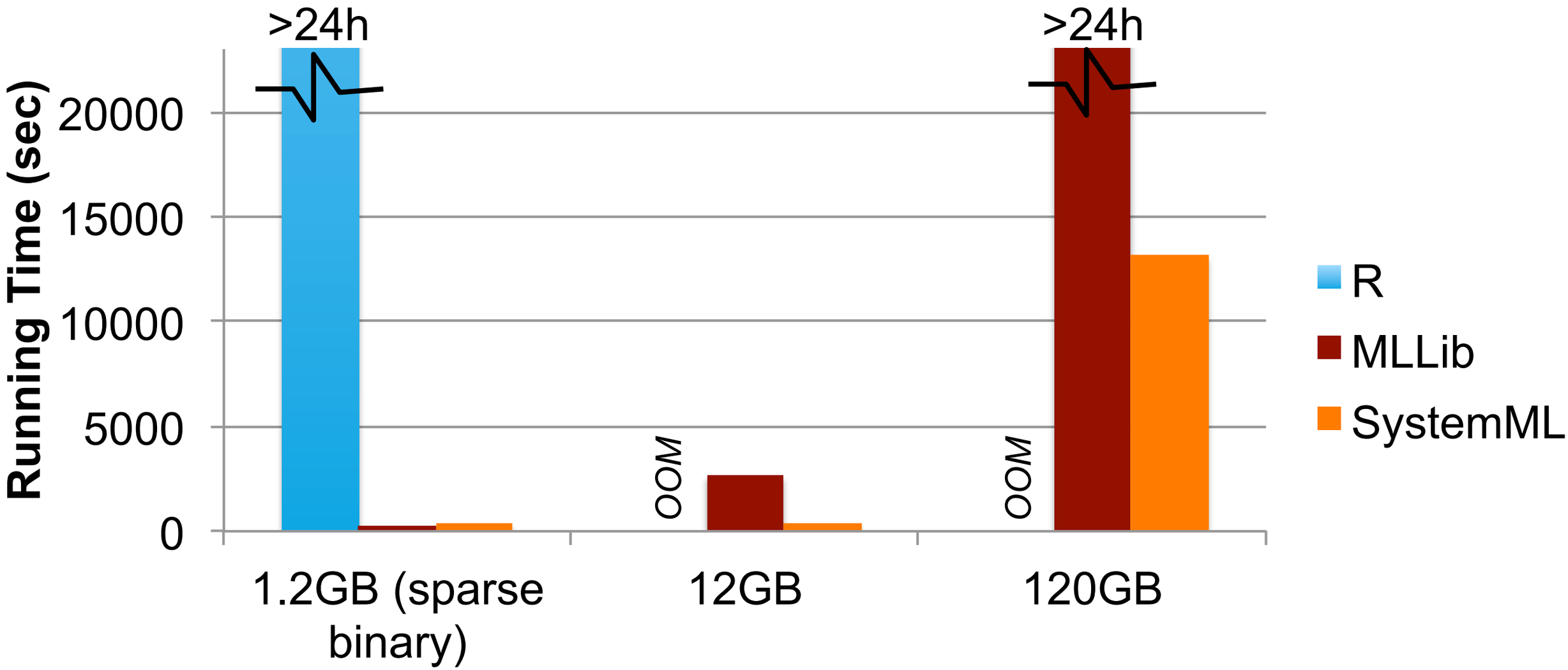
25 lines' worth of algorithm…

…mixed with 800 lines of performance code

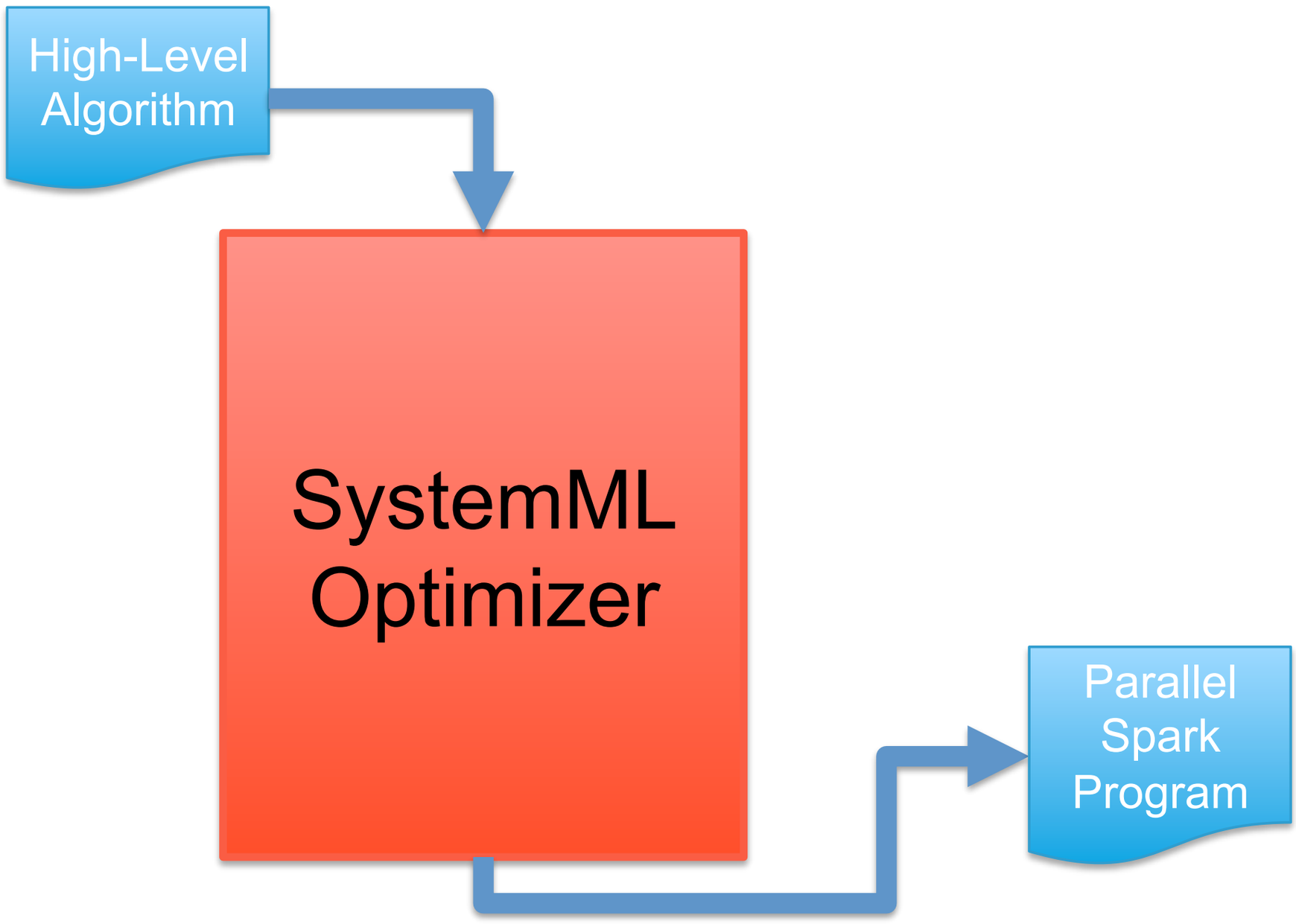
```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

```
U = rand(nrow(X), r, min = -1.0, max = 1.0);
V = rand(r, ncol(X), min = -1.0, max = 1.0);
while(i < mi) {
   i = i + 1; ii = 1;
   if (is_U)
      G = (W * (U %*% V - X)) %*% t(V) + lambda * U;
   else
      G = t(U) %*% (W * (U %*% V - X)) + lambda * V;
   norm_G2 = sum(G ^ 2); norm_R2 = norm_G2;
   R = -G; S = R;
   while(norm_R2 > 10E-9 * norm_G2 & ii <= mii) {
     if (is_U) {
       HS = (W * (S %*% V)) %*% t(V) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       U = U + alpha * S;
     } else {
       HS = t(U) %*% (W * (U %*% S)) + lambda * S;
       alpha = norm_R2 / sum (S * HS);
       V = V + alpha * S;
     }
     R = R - alpha * HS;
     old_norm_R2 = norm_R2; norm_R2 = sum(R ^ 2);
     S = R + (norm_R2 / old_norm_R2) * S;
     ii = ii + 1;
   }
   is_U = ! is_U;
}
```

SystemML:
compile and run at scale
no performance code needed!


>24h
>24h
Running Time (sec)
20000
15000
10000
5000
0
OOM
OOM
1.2GB (sparse binary)
12GB
120GB
R
MLLib
SystemML

Architecture



Architecture

High-level language front-ends

Cost Based Optimizer

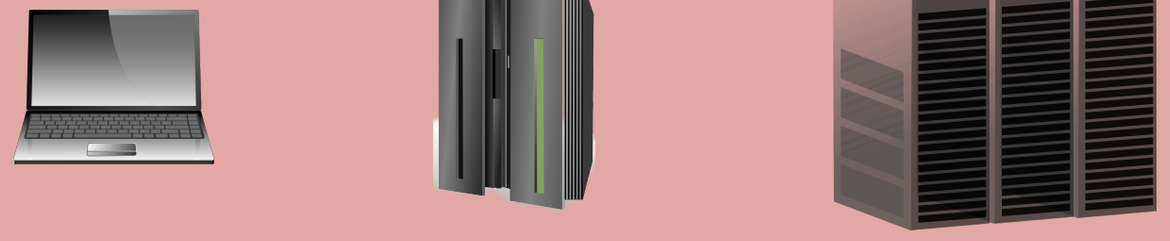
High-Level Operations (HOPs)

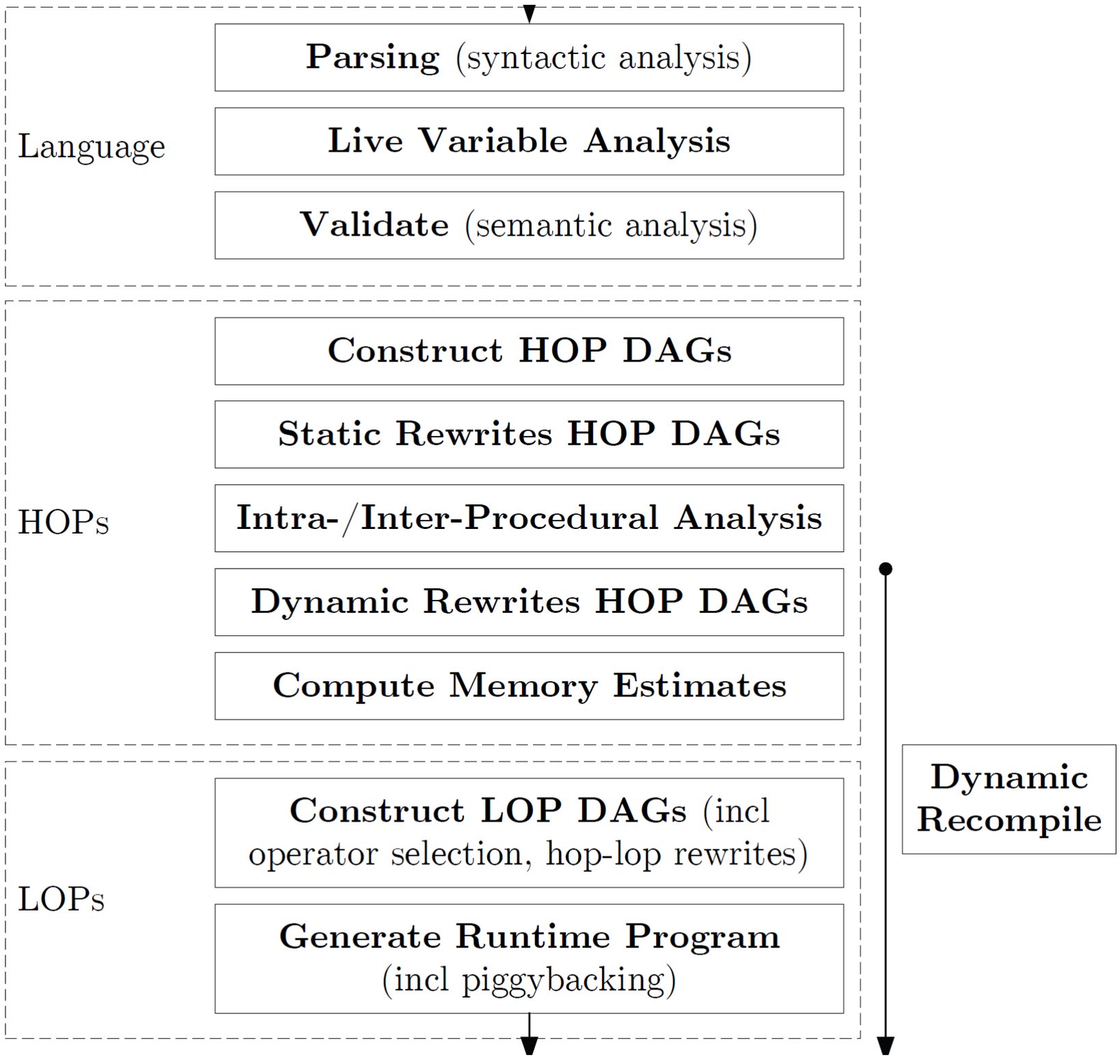
General representation of statements in the data analysis language

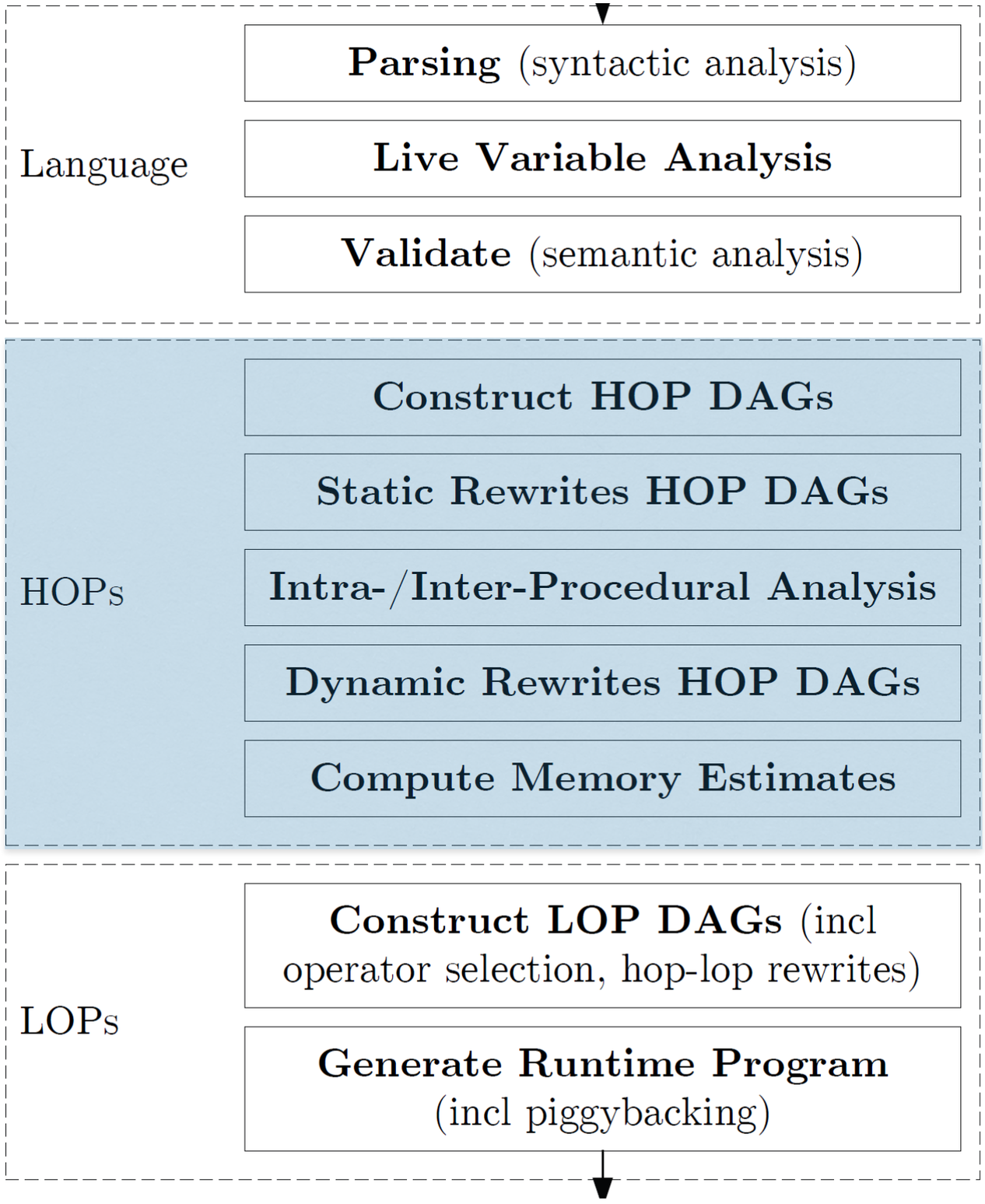
Low-Level Operations (LOPs)

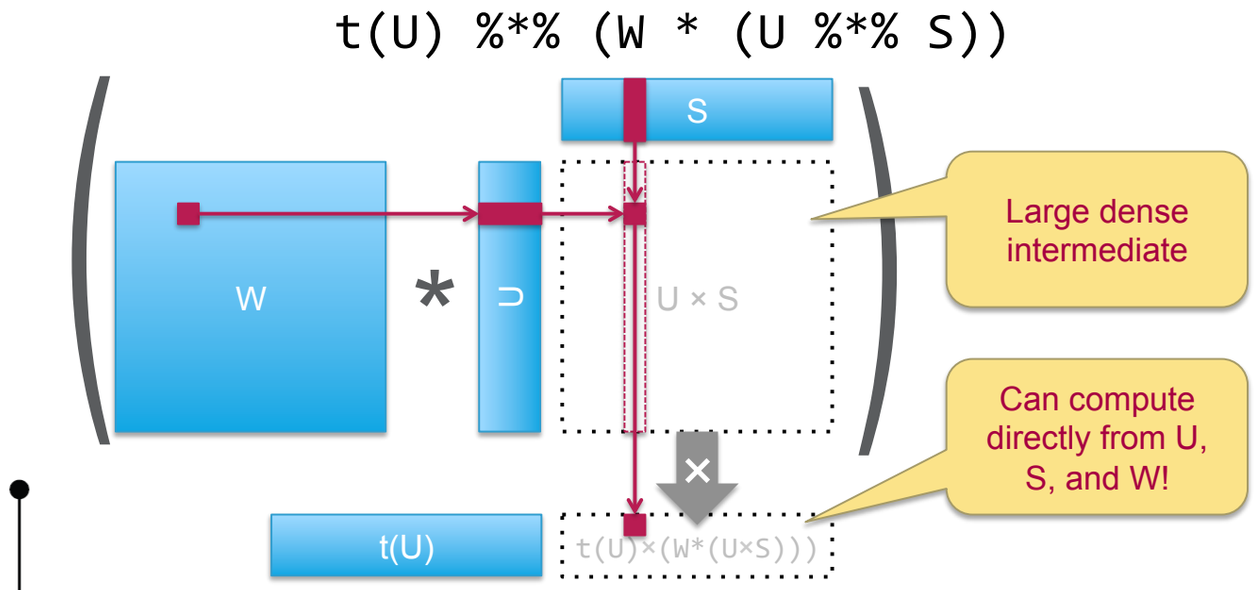
General representation of operations in the runtime framework

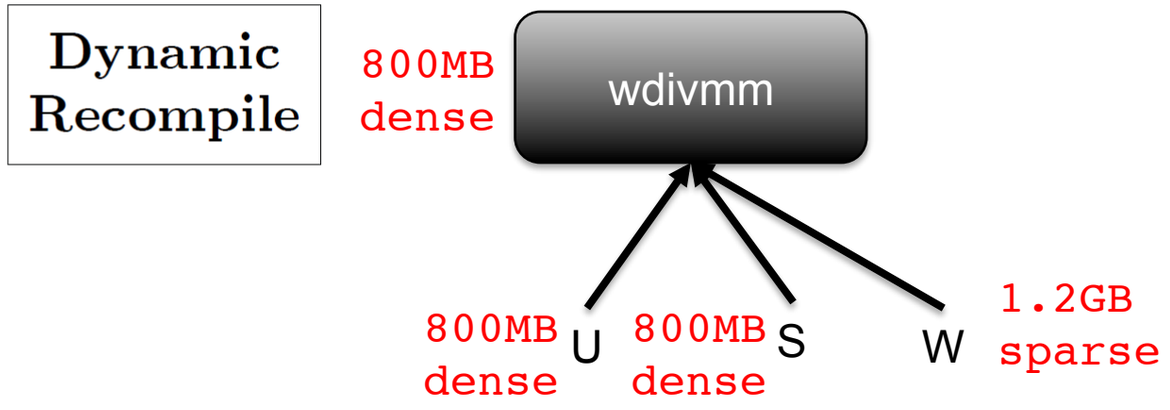
Multiple execution environments




Language
Parsing (syntactic analysis)
Live Variable Analysis
Validate (semantic analysis)
HOPs
Construct HOP DAGs
Static Rewrites HOP DAGs
Intra-/Inter-Procedural Analysis
Dynamic Rewrites HOP DAGs
Compute Memory Estimates
LOPs
Construct LOP DAGs (incl operator selection, hop-lop rewrites)
Generate Runtime Program (incl piggybacking)
Dynamic Recompile







(weighted divide matrix multiplication)

Language

Parsing (syntactic analysis)

Live Variable Analysis

Validate (semantic analysis)

HOPs

Construct HOP DAGs

Static Rewrites HOP DAGs

Intra-/Inter-Procedural Analysis

Dynamic Rewrites HOP DAGs

Compute Memory Estimates

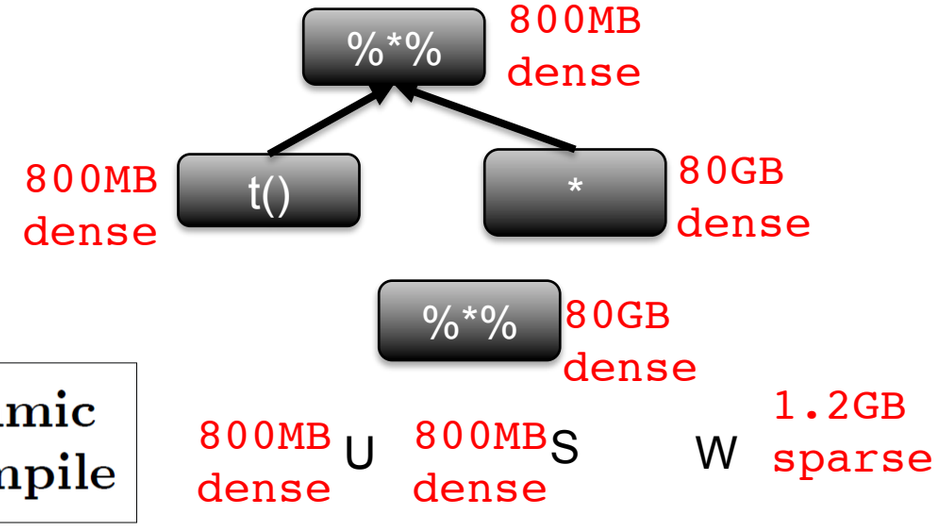
LOPs

Construct LOP DAGs (incl operator selection, hop-lop rewrites)

Generate Runtime Program (incl piggybacking)

Dynamic Recompile

All operands fit into heap → **use one node**



WDivMM

(MapWDivMM)

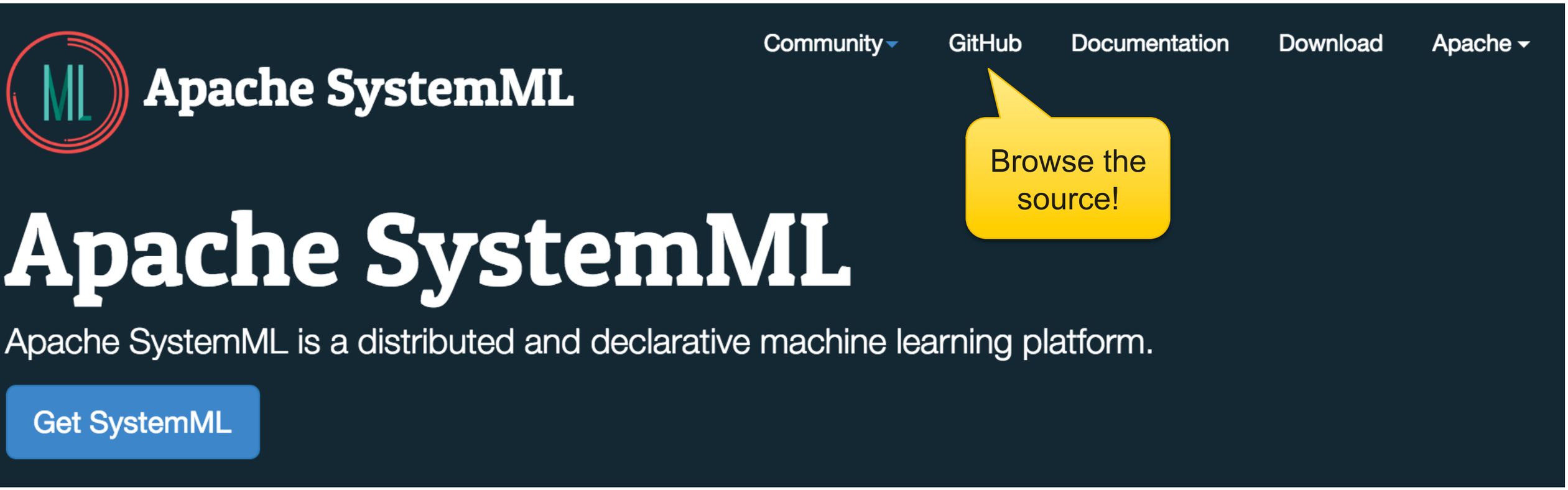
Apache SystemML

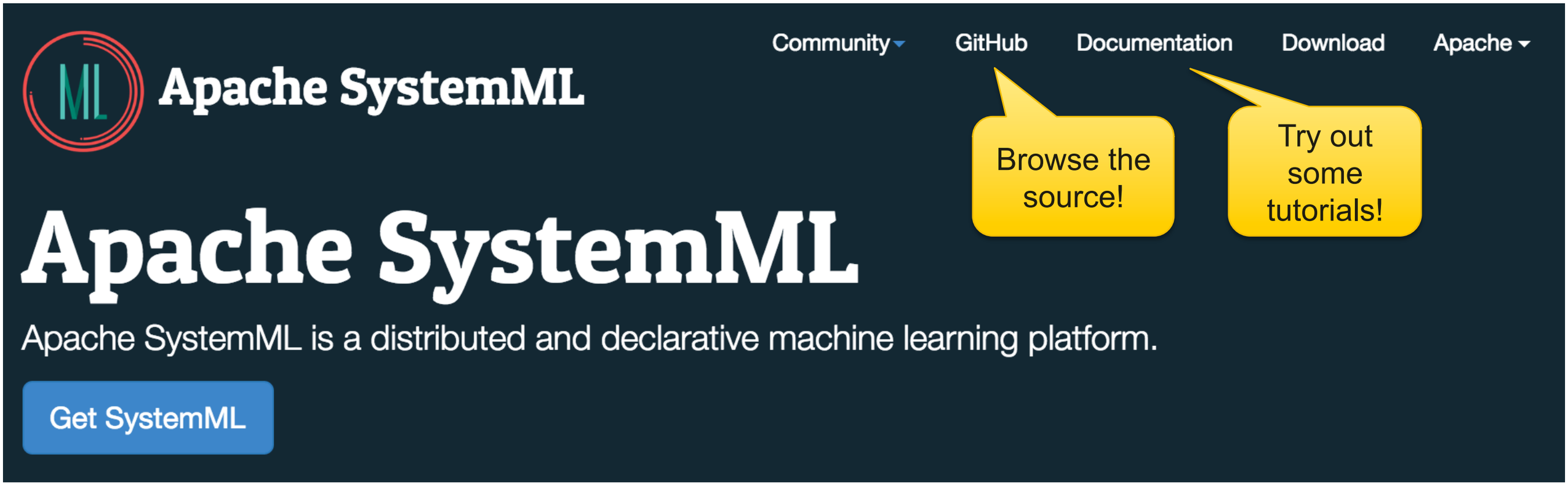
Community GitHub Documentation Download Apache

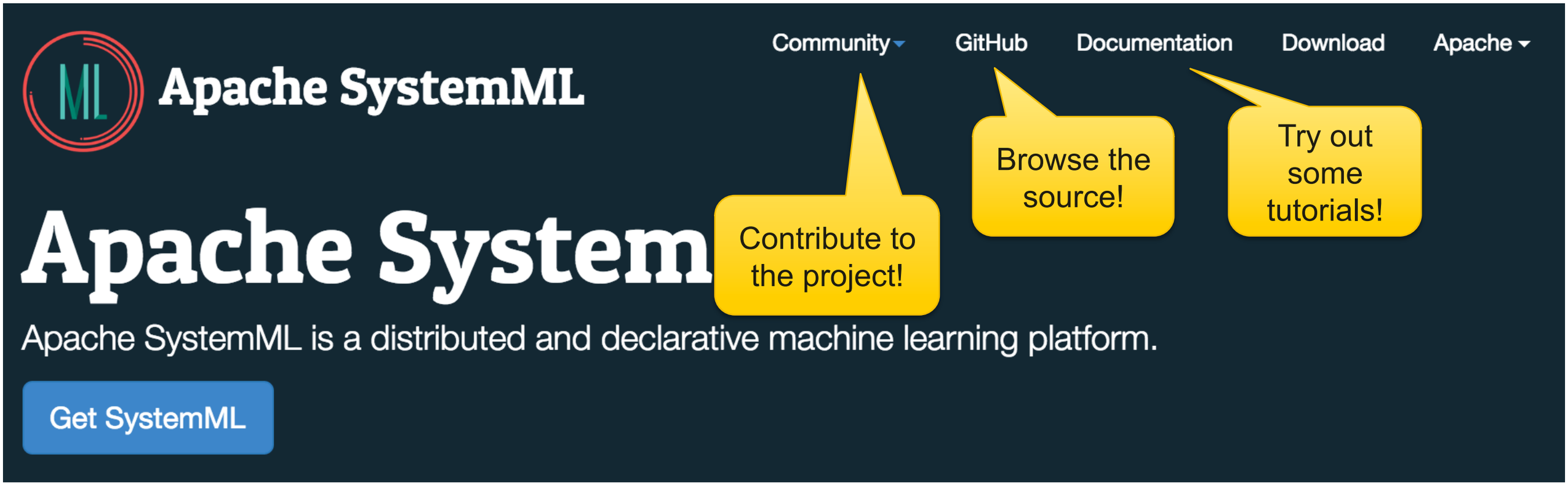
Apache SystemML

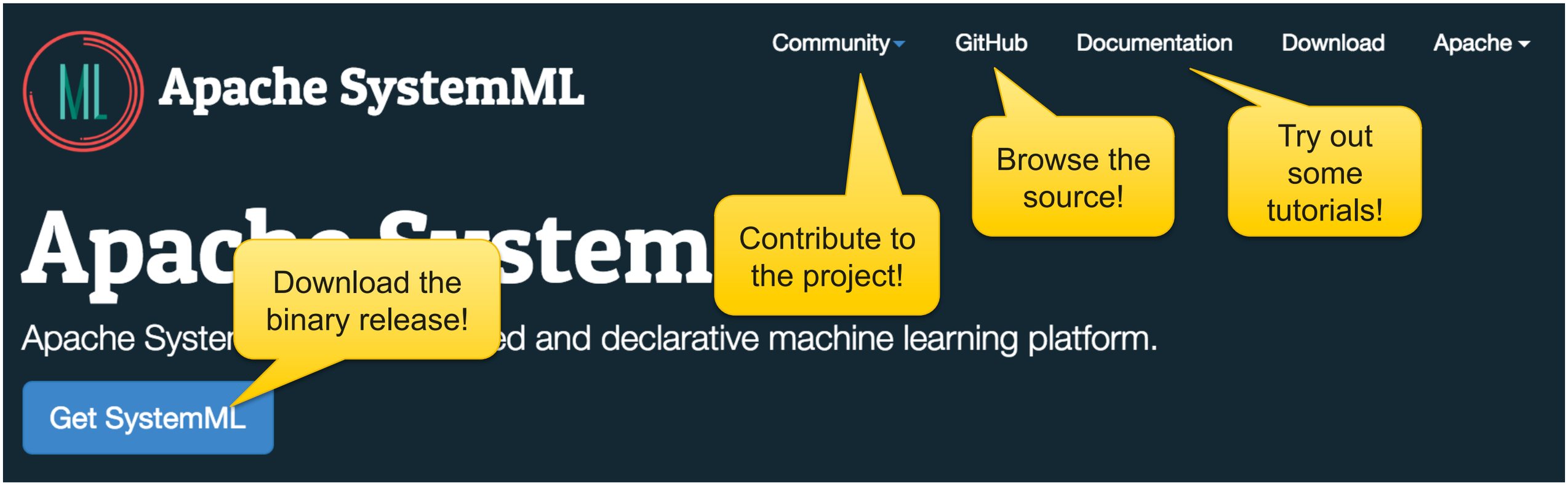
Apache SystemML is a distributed and declarative machine learning platform.

Get SystemML


Apache SystemML
Community
GitHub
Documentation
Download
Apache
Browse the source!
Apache SystemML
Apache SystemML is a distributed and declarative machine learning platform.
Get SystemML

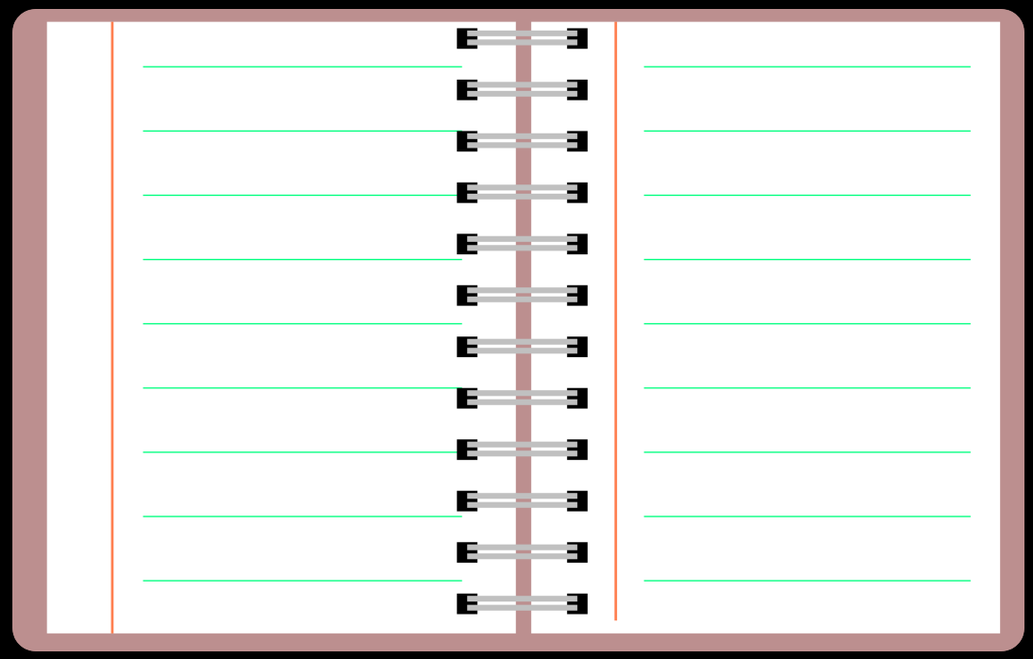

Apache SystemML
Community
GitHub
Documentation
Download
Apache
Browse the source!
Try out some tutorials!
Apache SystemML
Apache SystemML is a distributed and declarative machine learning platform.
Get SystemML


Apache SystemML
Community
GitHub
Documentation
Download
Apache
Apache System
Contribute to the project!
Browse the source!
Try out some tutorials!
Apache SystemML is a distributed and declarative machine learning platform.
Get SystemML


Apache SystemML
Community
GitHub
Documentation
Download
Apache
Apac stem
Apache Syste ed and declarative machine learning platform.
Get SystemML
Download the binary release!
Contribute to the project!
Browse the source!
Try out some tutorials!

Agenda

- Introduction to DeepLearning ✓
- Introduction to ApacheSpark ✓
- Introduction to DL4J ✓
- Introduction to SystemML ✓
- Demos



- 6000+ clients
- $3B investment
- Partners including
  - Avnet, BNP Paribas, EEBus, Capgemini, Tech Mahindra, Vodafone, BMW, Visa, Bosch, Indiegogo, French national railway SNCF,Arrow Electronics, Intel, Cisco

OPEN CONNECTIVITY FOUNDATION™

industrial internet® CONSORTIUM

MQTT.ORG

LoRa™ Alliance
Wide Area Networks for IoT

JS Foundation

one M2M

AIOTI
ALLIANCE FOR INTERNET OF THINGS INNOVATION

THE LINUX FOUNDATION

HYPERLEDGER

Spark

eclipse

openstack

Node-RED

CLOUD FOUNDRY

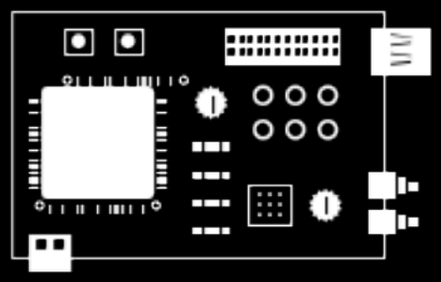
ALLSEEN ALLIANCE

Why IoT (now) ?

- 15 Billion connected devices in 2015
- 40 Billion connected devices in 2020
- World population 7.4 Billion in 2016

Now you can believe the hype

OASIS makes MQTT gold standard for Internet of Things



Your device or gateway

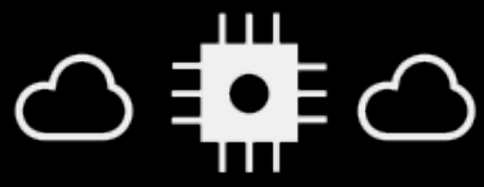
We start with your device, be it a sensor, a gateway or something else.
To find out how to get it connected, search our recipes.

MQTT & HTTP

Your device data is sent securely up to the cloud using
the open, lightweight MQTT messaging protocol or HTTP.

REST & Real-time APIs

Use our secure APIs to connect your apps
with the data coming from your devices.

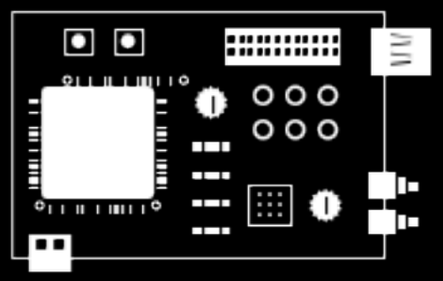


IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can
setup and manage your connected devices so that your
apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud,
or your own servers to interpret the data you now have access to!



Your device or gateway

We start with your device, be it a sensor, a gateway or something else.
To find out how to get it connected, search our recipes.

MQTT & HTTP

Your device data is sent securely up to the cloud using
the open, lightweight MQTT messaging protocol or HTTP.

api

REST & Real-time APIs

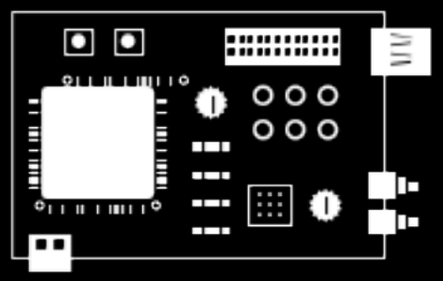
Use our secure APIs to connect your apps
with the data coming from your devices.

IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can
setup and manage your connected devices so that your
apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud,
or your own servers to interpret the data you now have access to!



Your device or gateway

We start with your device, be it a sensor, a gateway or something else.
To find out how to get it connected, search our recipes.

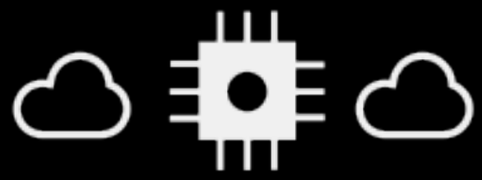
MQTT & HTTP

Your device data is sent securely up to the cloud using
the open, lightweight MQTT messaging protocol or HTTP.

api

REST & Real-time APIs

Use our secure APIs to connect your apps
with the data coming from your devices.

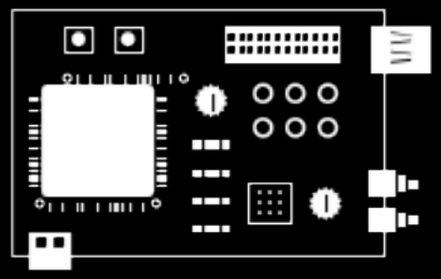


IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can
setup and manage your connected devices so that your
apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud,
or your own servers to interpret the data you now have access to!



Your device or gateway

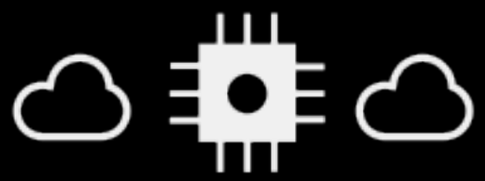
We start with your device, be it a sensor, a gateway or something else.
To find out how to get it connected, search our recipes.

MQTT & HTTP

Your device data is sent securely up to the cloud using
the open, lightweight MQTT messaging protocol or HTTP.

REST & Real-time APIs

Use our secure APIs to connect your apps
with the data coming from your devices.

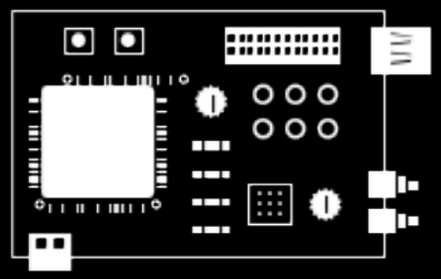


IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can
setup and manage your connected devices so that your
apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud,
or your own servers to interpret the data you now have access to!



Your device or gateway

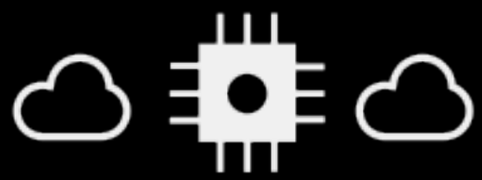
We start with your device, be it a sensor, a gateway or something else.
To find out how to get it connected, search our recipes.

MQTT & HTTP

Your device data is sent securely up to the cloud using the open, lightweight MQTT messaging protocol or HTTP.

REST & Real-time APIs

Use our secure APIs to connect your apps with the data coming from your devices.

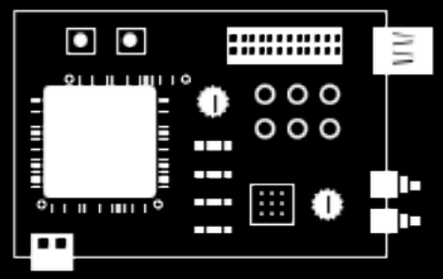


IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can setup and manage your connected devices so that your apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud, or your own servers to interpret the data you now have access to!



Your device or gateway

We start with your device, be it a sensor, a gateway or something else.
To find out how to get it connected, search our recipes.

MQTT & HTTP

Your device data is sent securely up to the cloud using the open, lightweight MQTT messaging protocol or HTTP.

REST & Real-time APIs

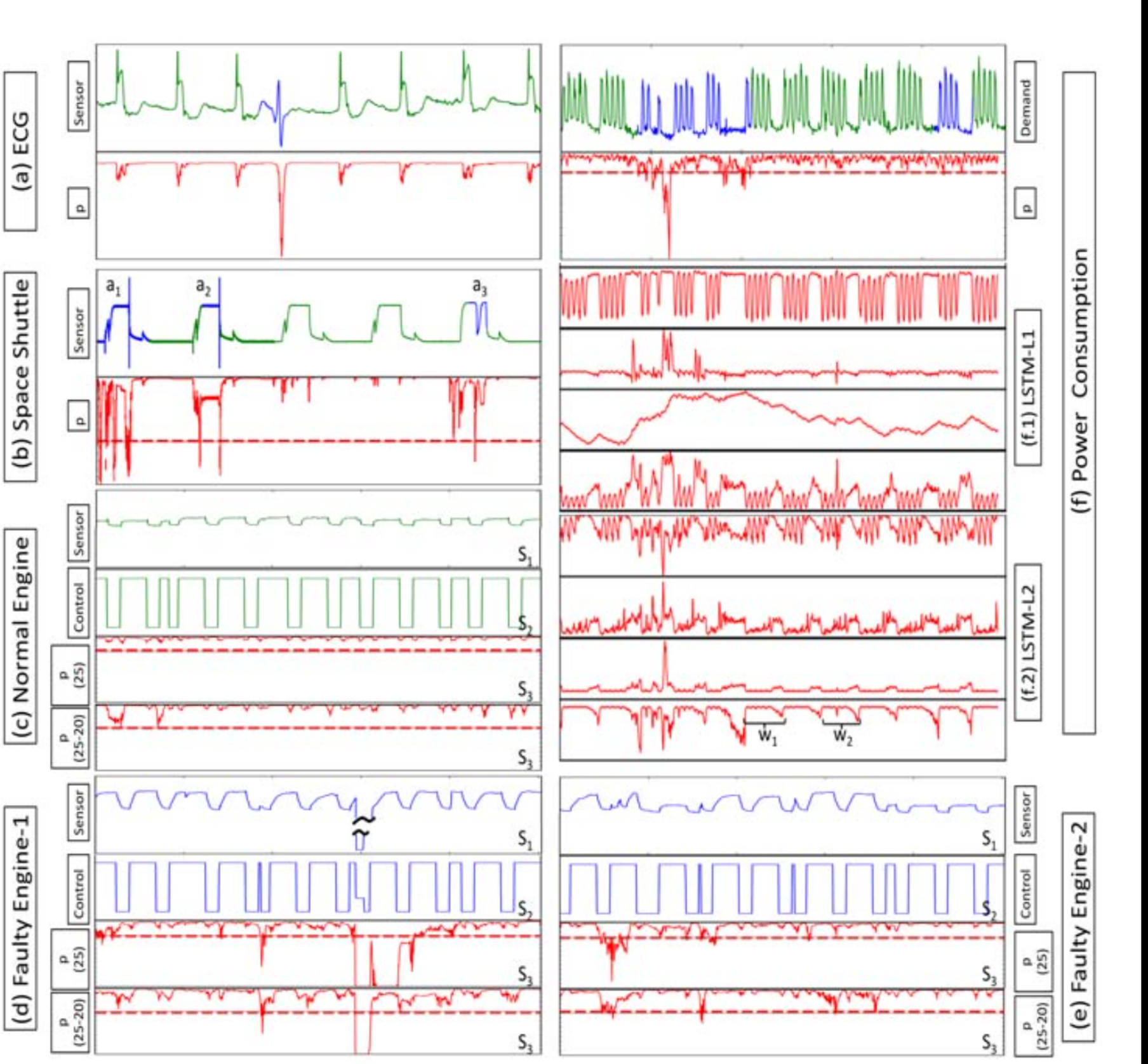
Use our secure APIs to connect your apps with the data coming from your devices.

IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can setup and manage your connected devices so that your apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud, or your own servers to interpret the data you now have access to!

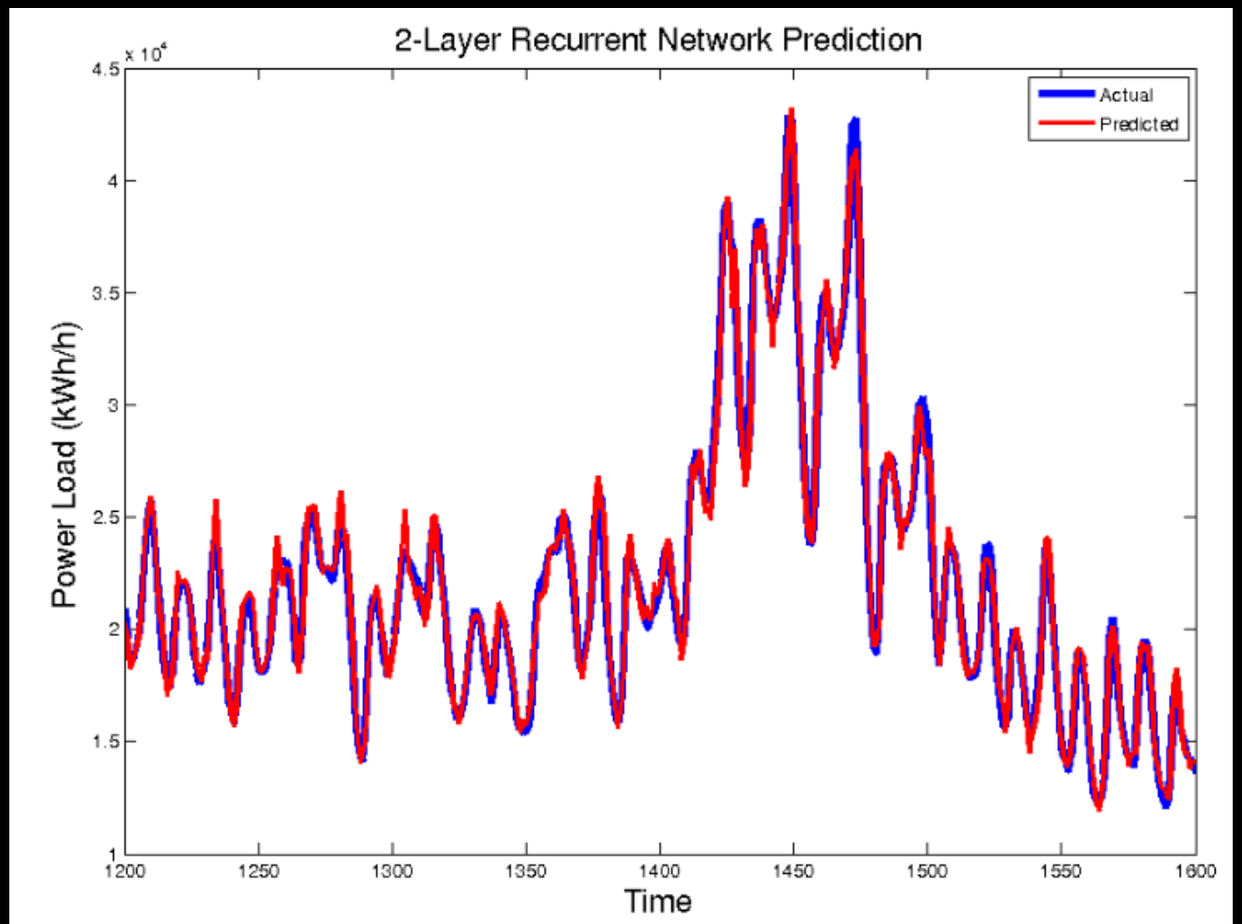


ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

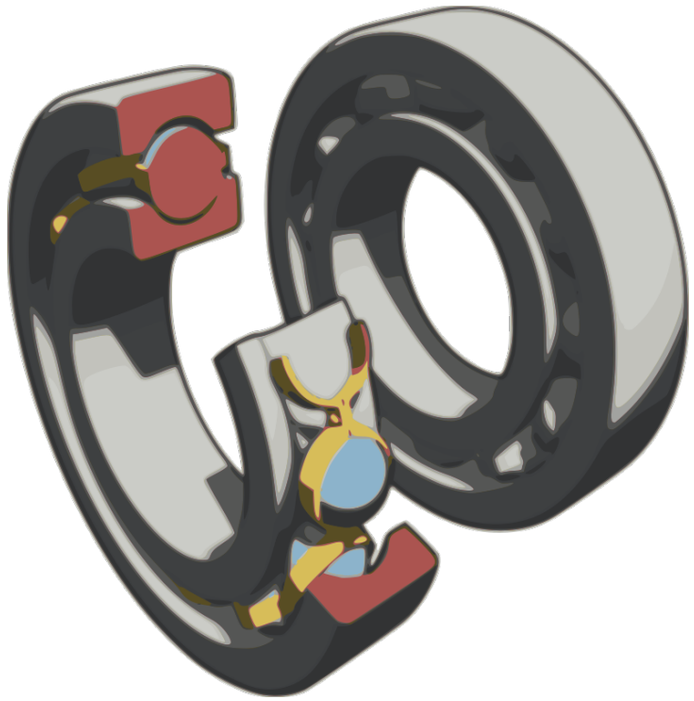
Deep Learning for Time Series Modeling

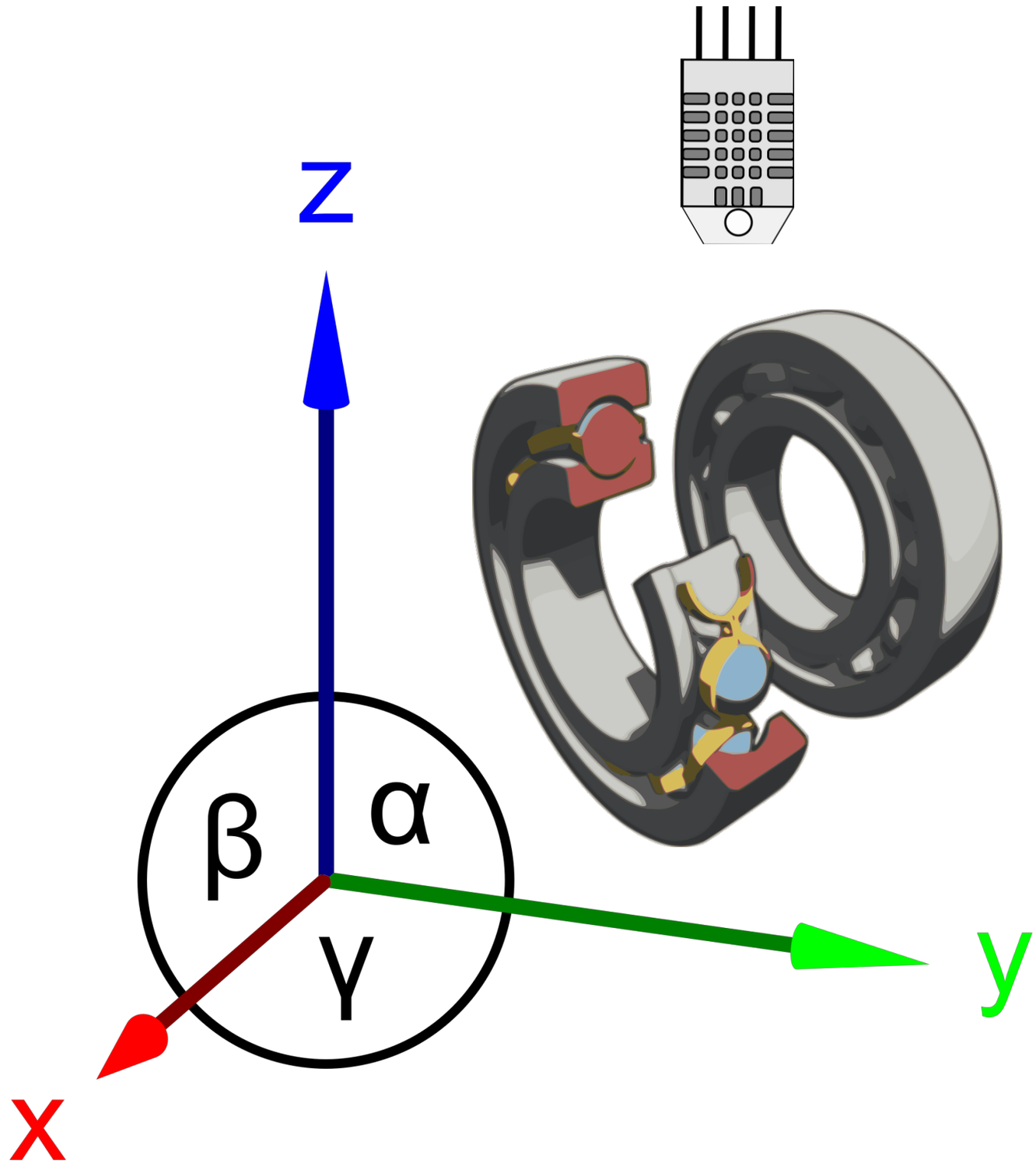
CS 229 Final Project Report

Enzo Busseti, Ian Osband, Scott Wong



Learning Method	RMSE	% RMSE
Kernelized Regression	1,540	8.3%
Frequency NN	1,251	6.7%
Deep Feedforward NN	1,103	5.9%
Deep Recurrent NN	530	2.8%





Node-RED

Deploy

filter nodes

Flow 1 | Flow 2 | Flow 3 | Flow 4 | Flow 5 | Flow 6 | Flow 7

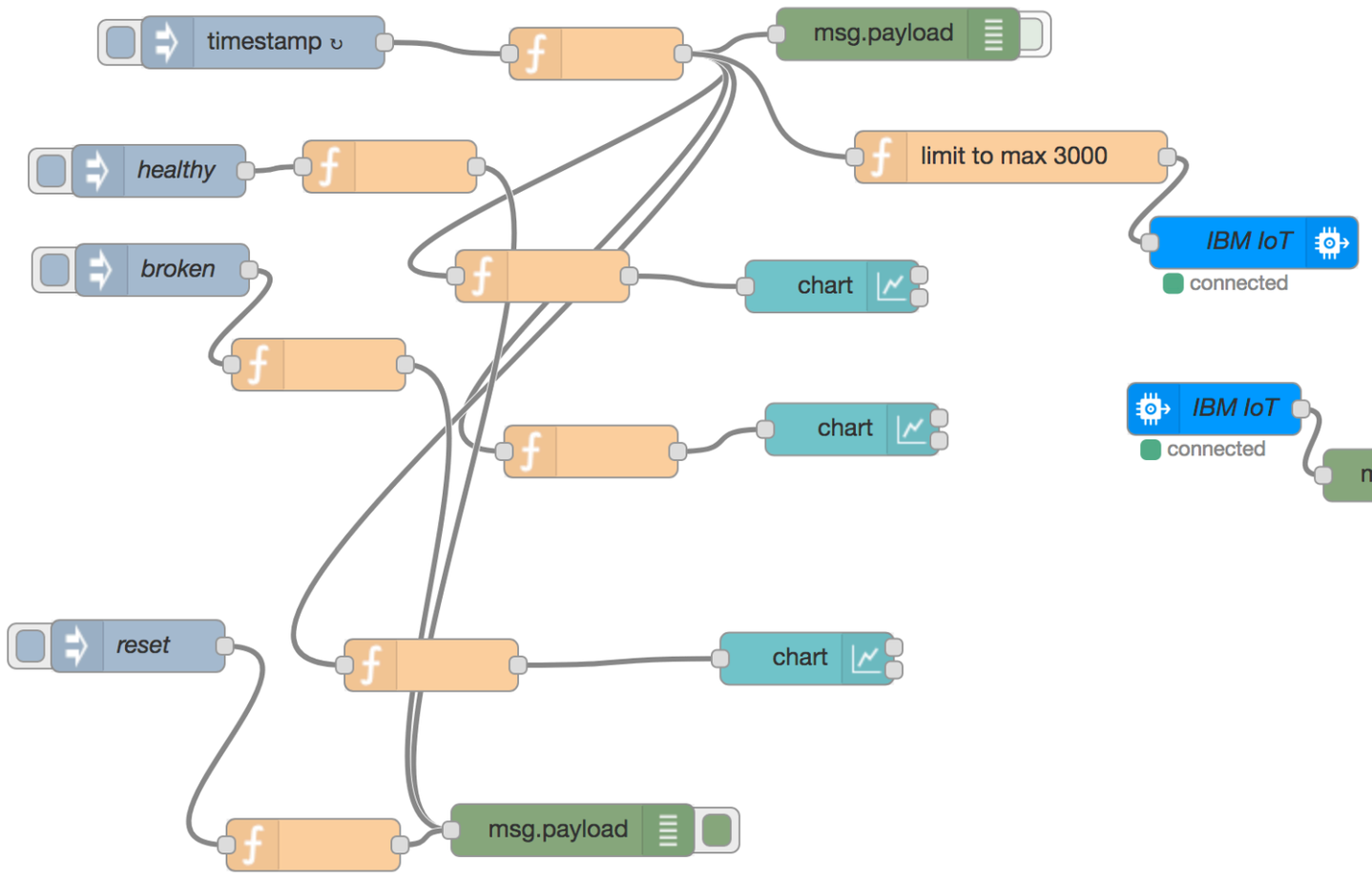
info | debug | dash

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- mqlight
- ibmiot

output

- debug
- link
- mqtt
- http response



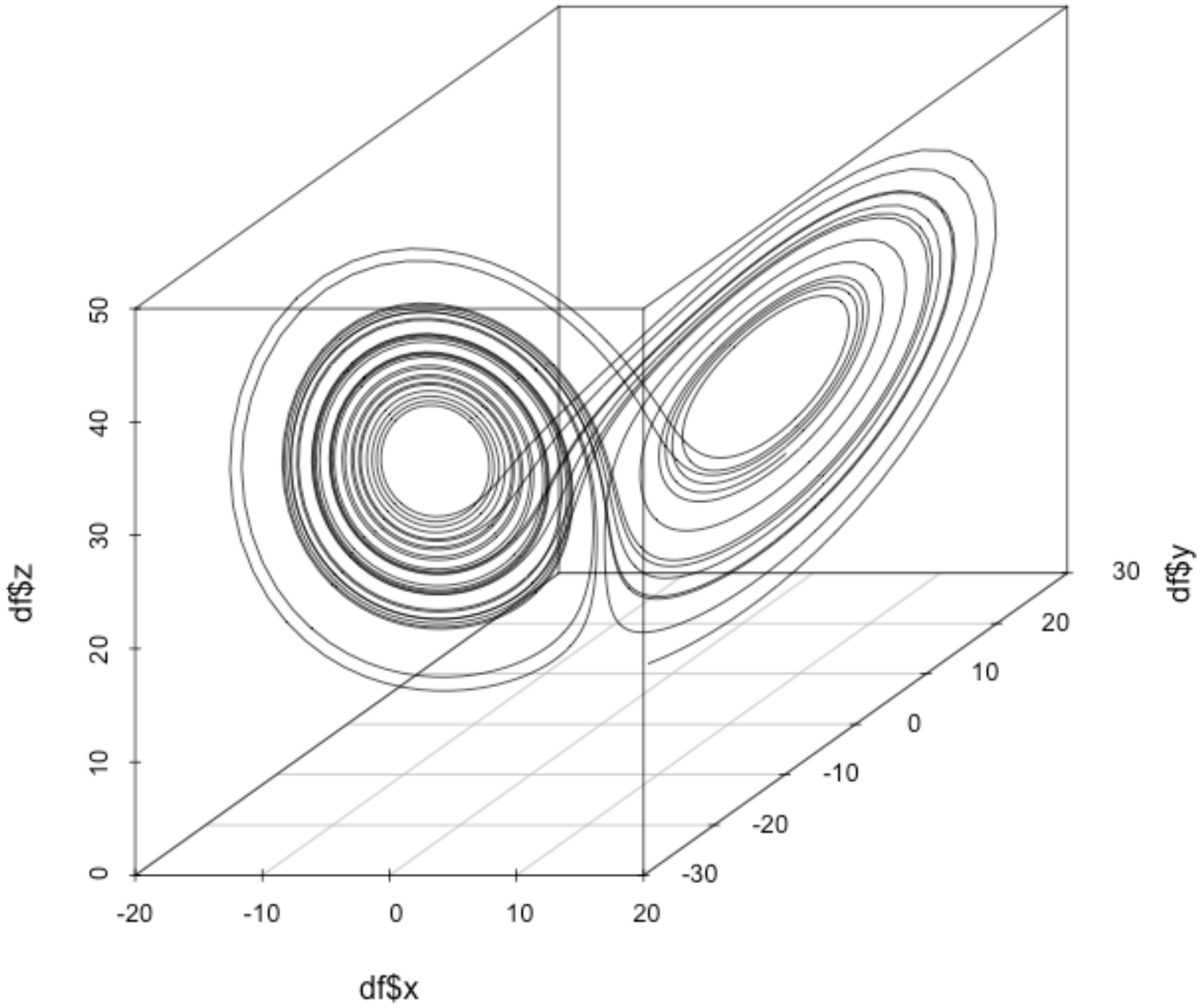
86.41824617650815 }

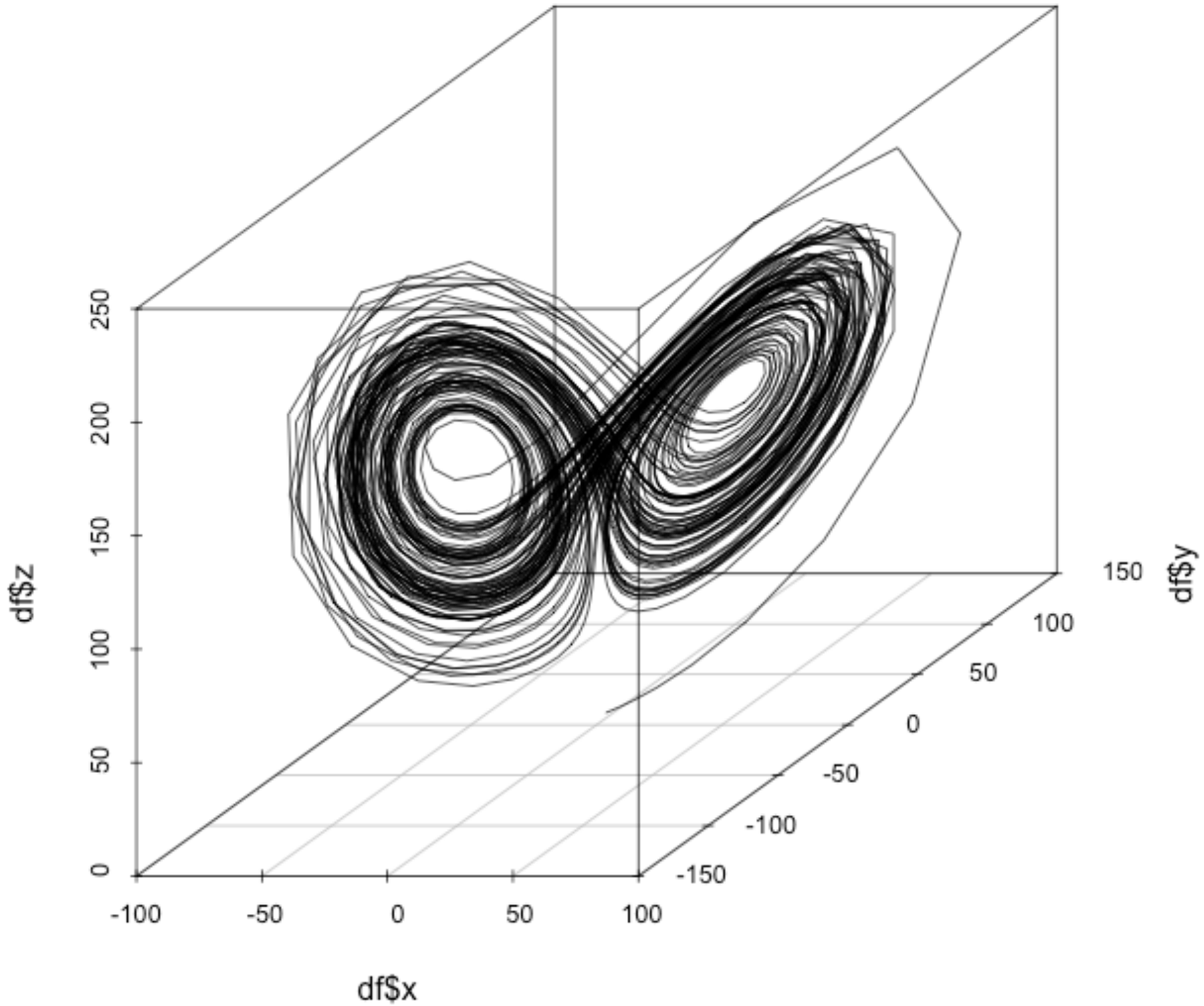
10/19/2017, 1:20:56 PM node: 9531d29.99a72b
iot-2/type/0.16.2/id/lorenz/evt/osc/fmt/json
: msg.payload : Object
{ x: -46.16269554882608, y: -93.7406140756372, z: 114.584239220398 }

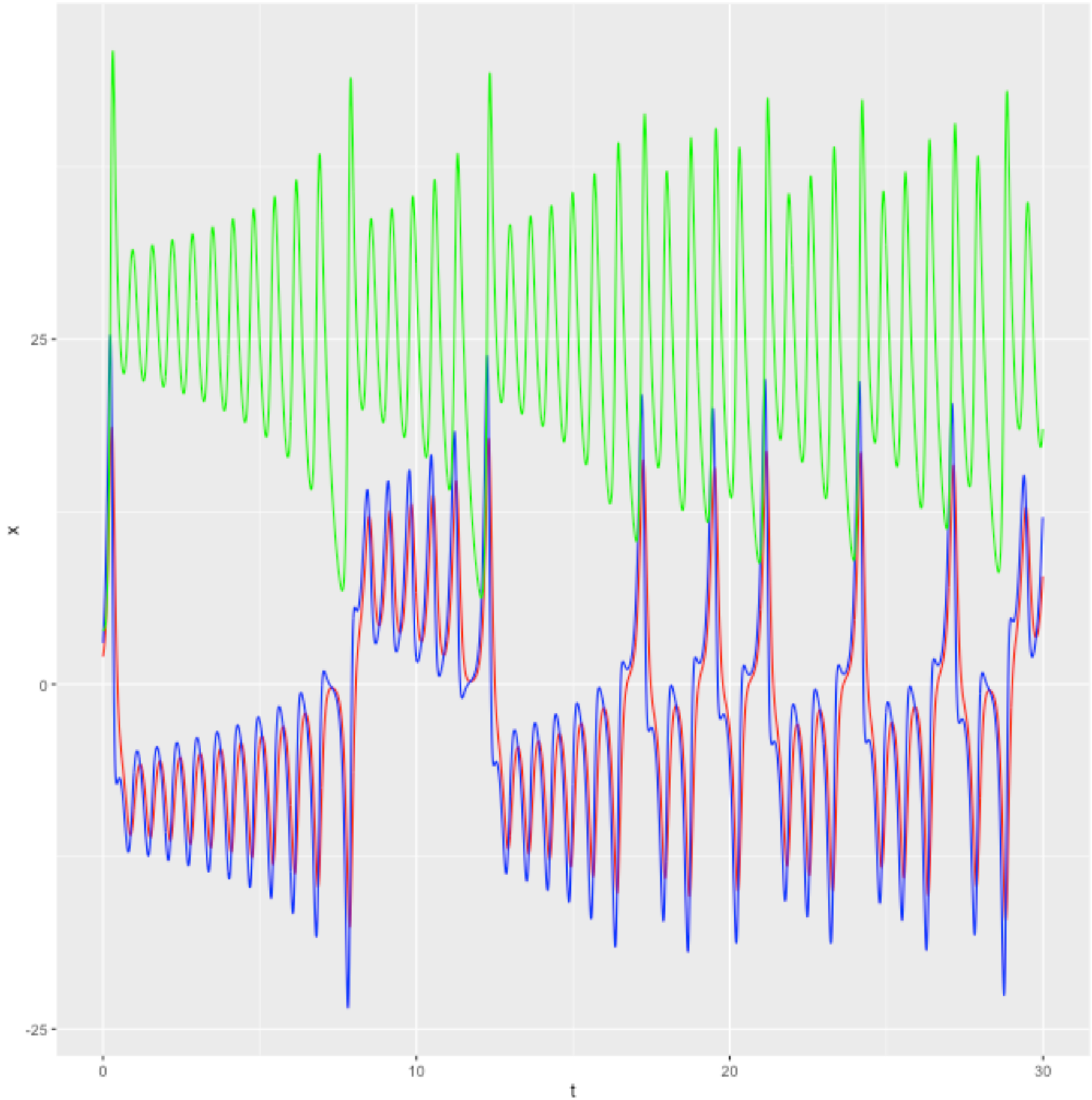
10/19/2017, 1:20:56 PM node: 9531d29.99a72b
iot-2/type/0.16.2/id/lorenz/evt/osc/fmt/json
: msg.payload : Object
{ x: -57.58139599526075, y: -99.1706750352557, z: 151.71170330785398 }

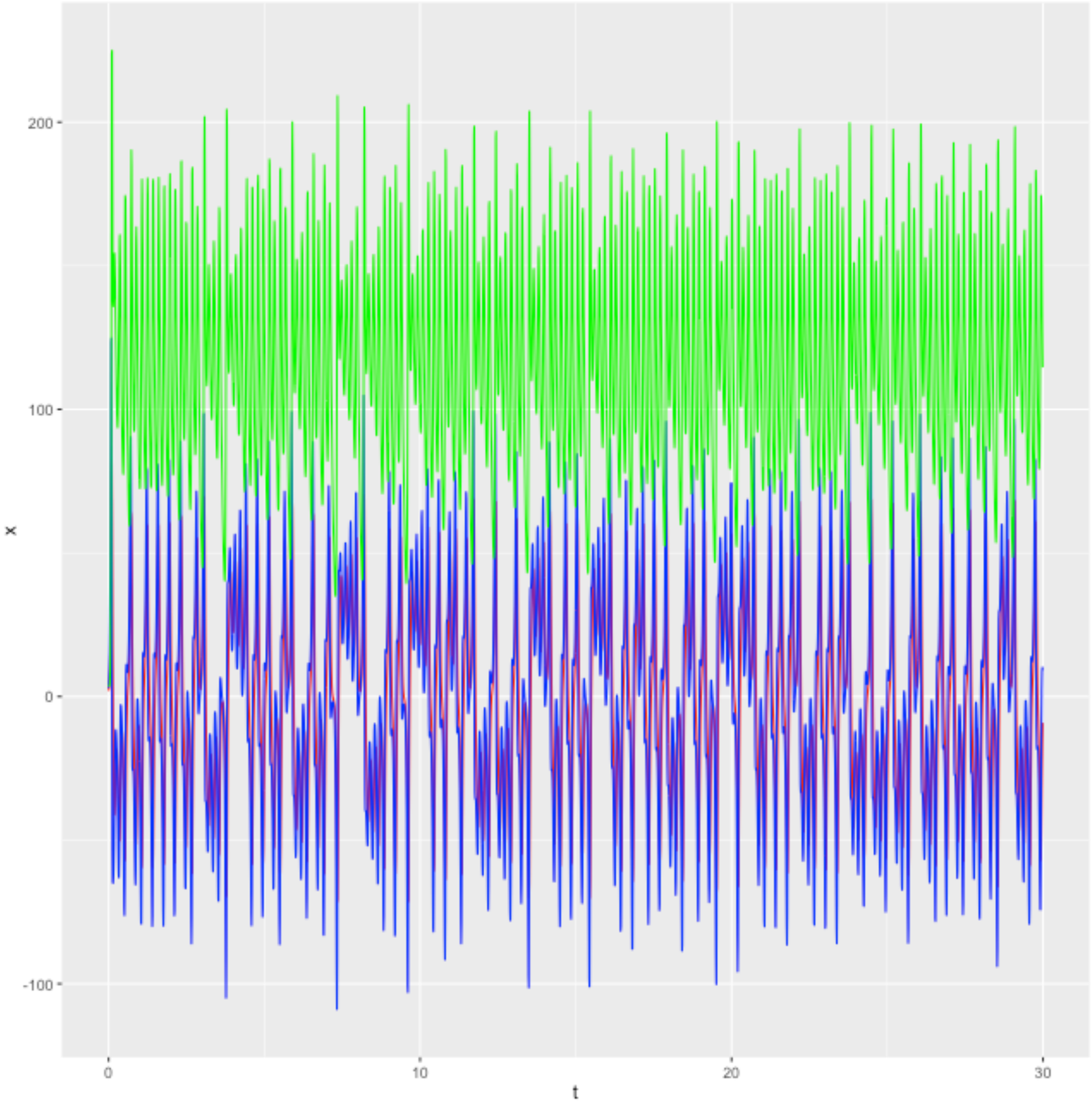
10/19/2017, 1:20:56 PM node: 9531d29.99a72b
iot-2/type/0.16.2/id/lorenz/evt/osc/fmt/json
: msg.payload : Object
{ x: -67.56282296485954, y: -85.56107273270315, z: 186.62987700531897 }

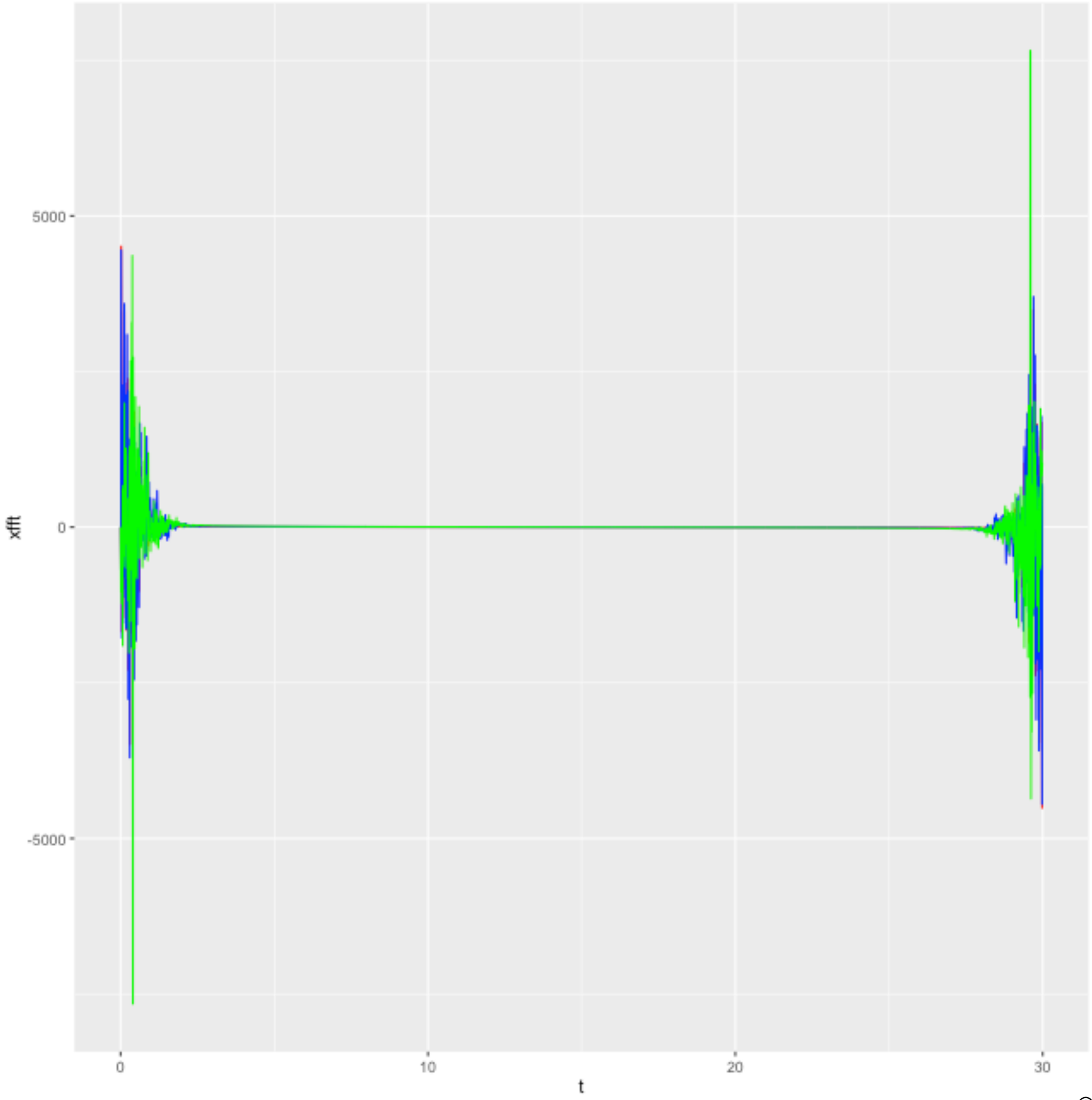
10/19/2017, 1:20:56 PM node: 9531d29.99a72b
iot-2/type/0.16.2/id/lorenz/evt/osc/fmt/json
: msg.payload : Object
{ x: -71.882402909142, y: -51.1609326195633, z: 202.11541236305715 }

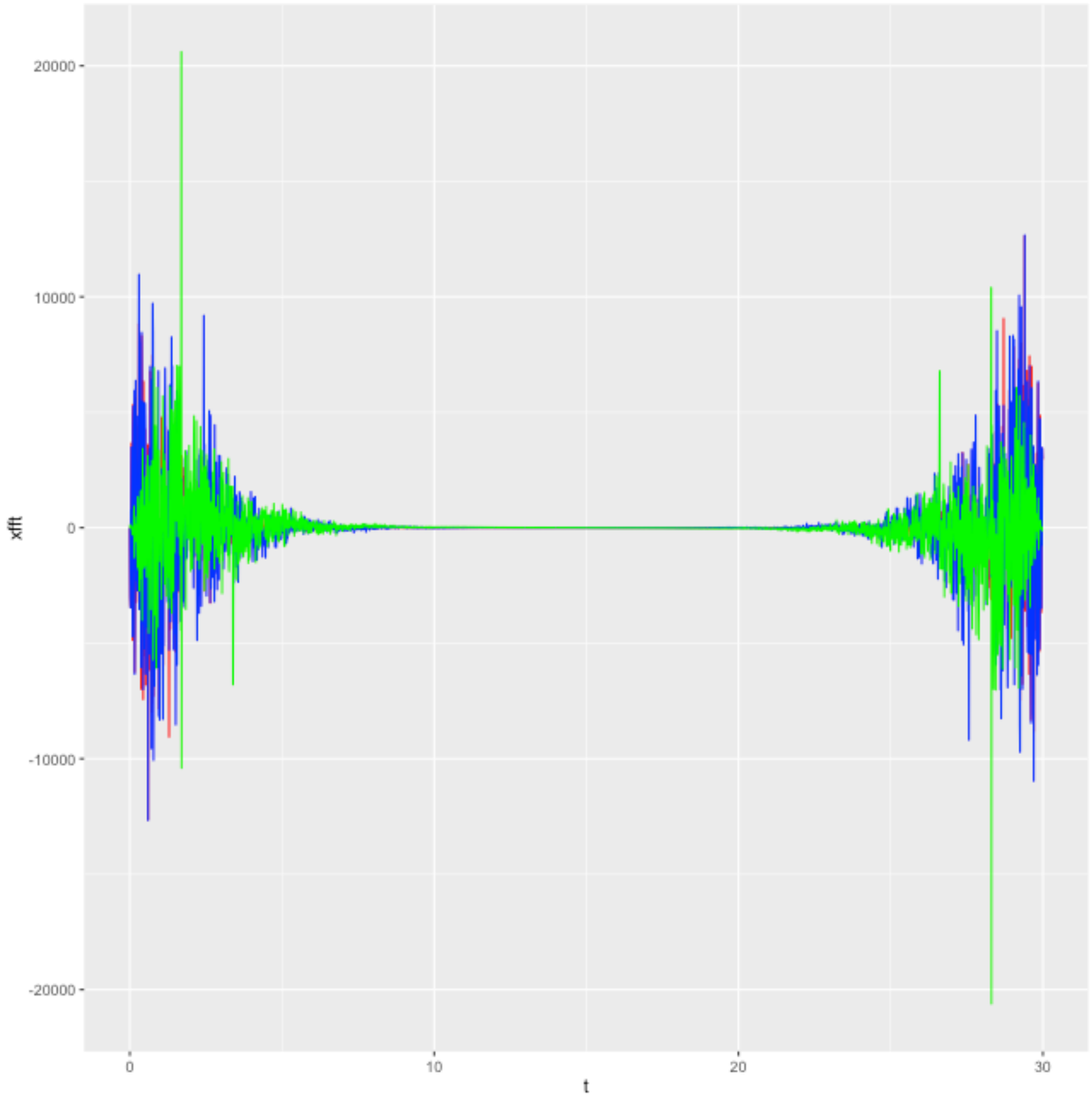












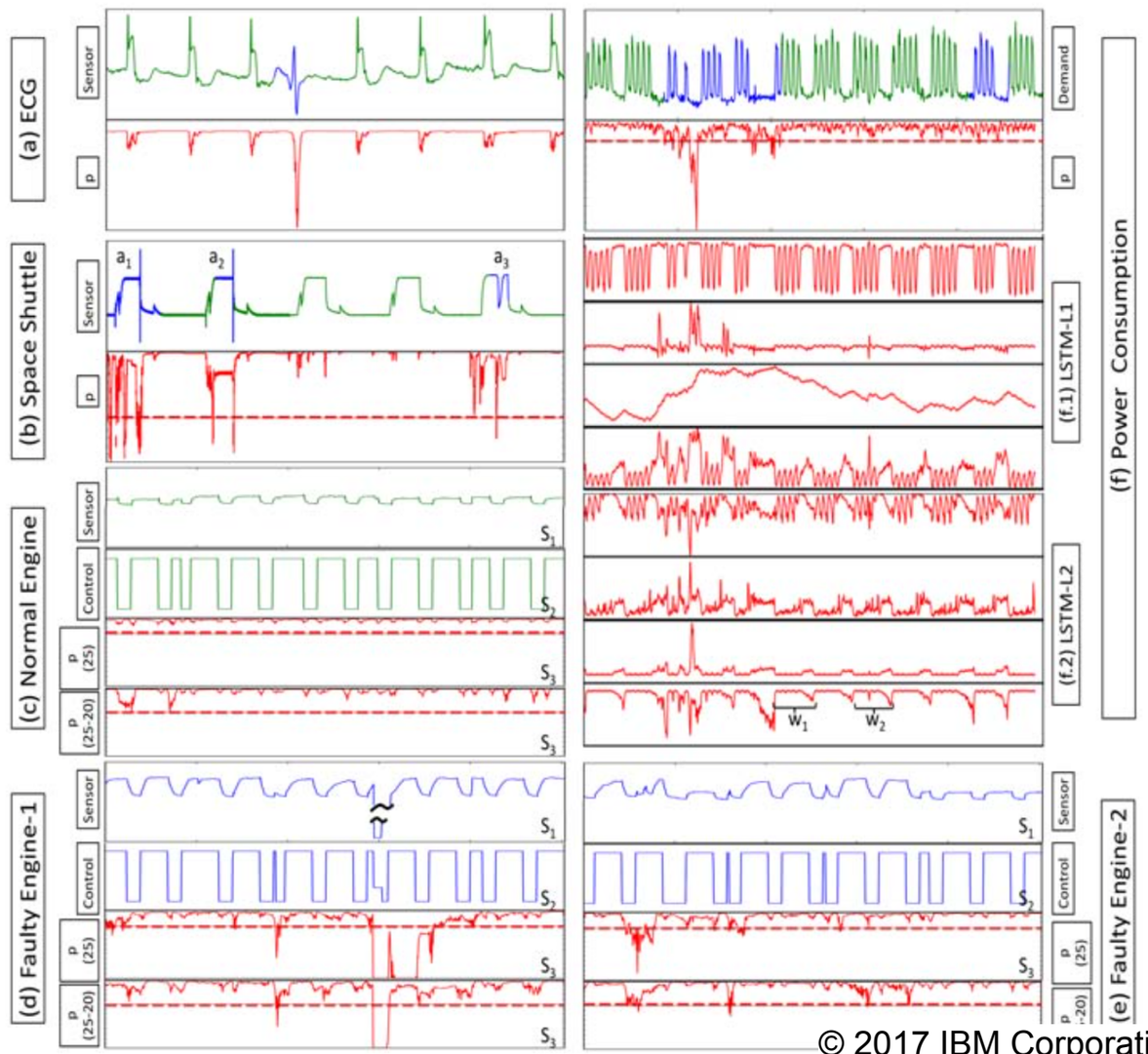
Long Short Term Memory Networks for Anomaly Detection in Time Series

Pankaj Malhotra[1], Lovekesh Vig[2], Gautam Shroff[1], Puneet Agarwal[1]

1- TCS Research, Delhi, India

2- Jawaharlal Nehru University, New Delhi, India

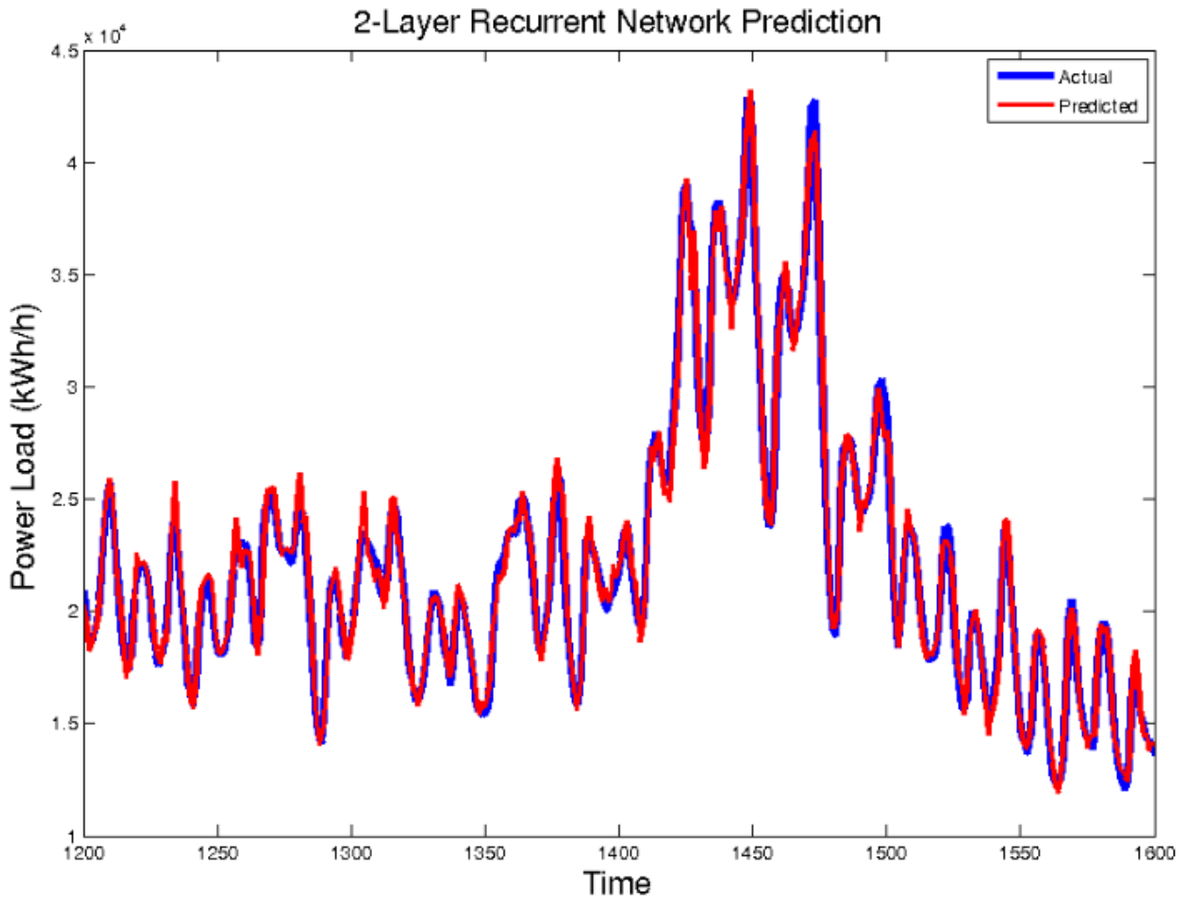
- Outperformed traditional methods, such as
  - cumulative sum (CUSUM)
  - exponentially weighted moving average (EWMA)
  - Hidden Markov Models (HMM)
- Learned what “Normal” is
- Raised error if time series pattern haven't been seen before



Deep Learning for Time Series Modeling

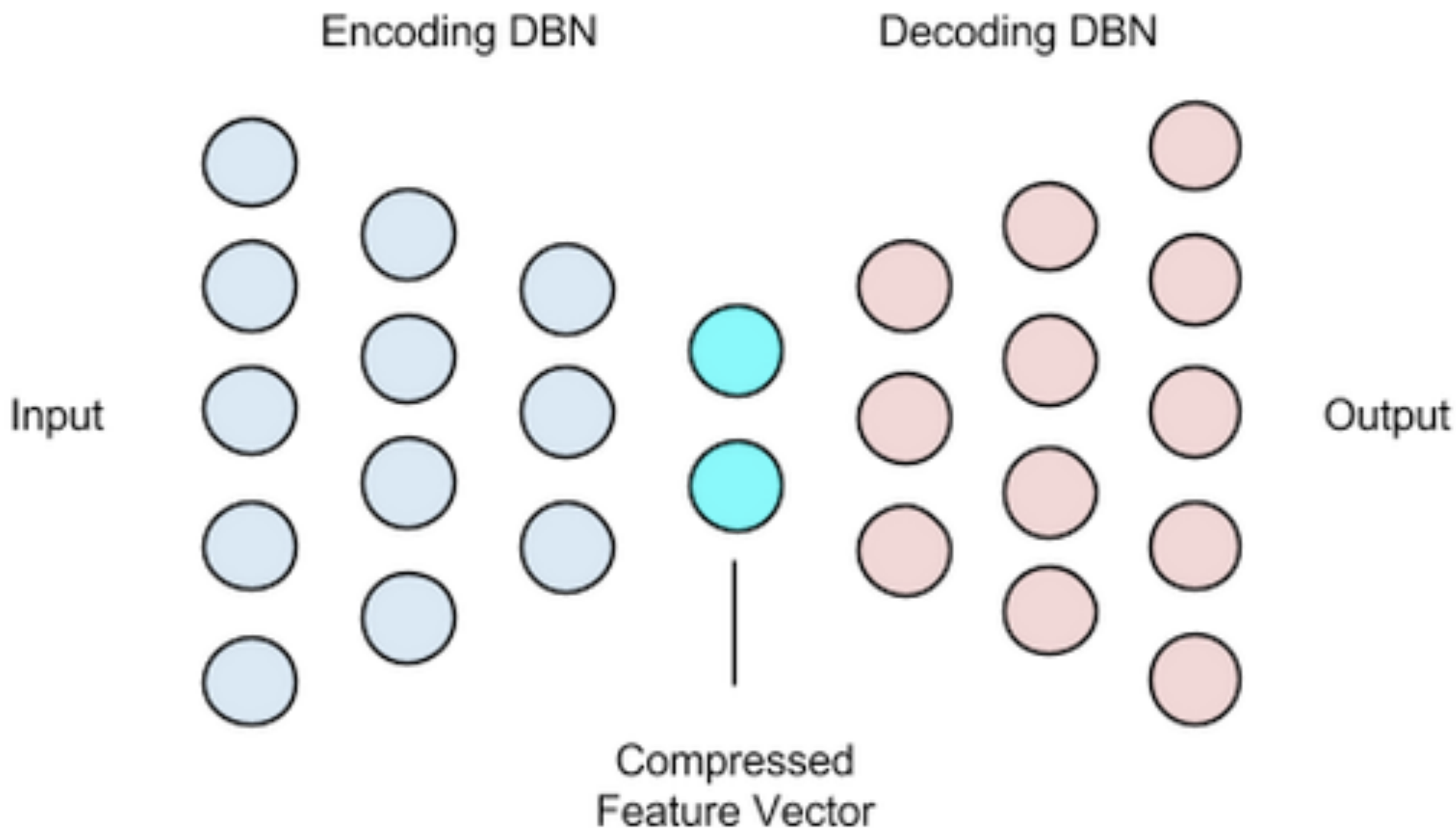
CS 229 Final Project Report

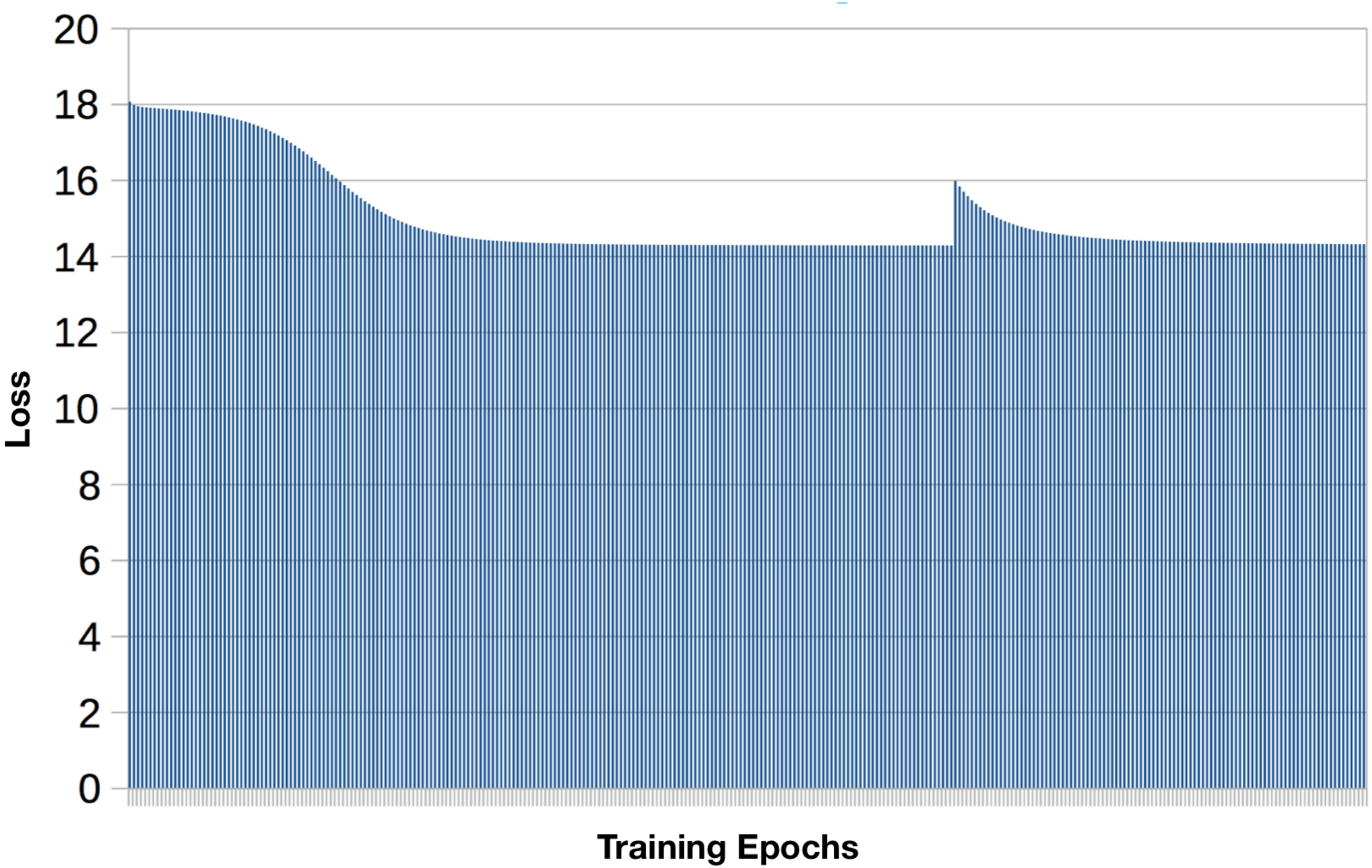
Enzo Busseti, Ian Osband, Scott Wong

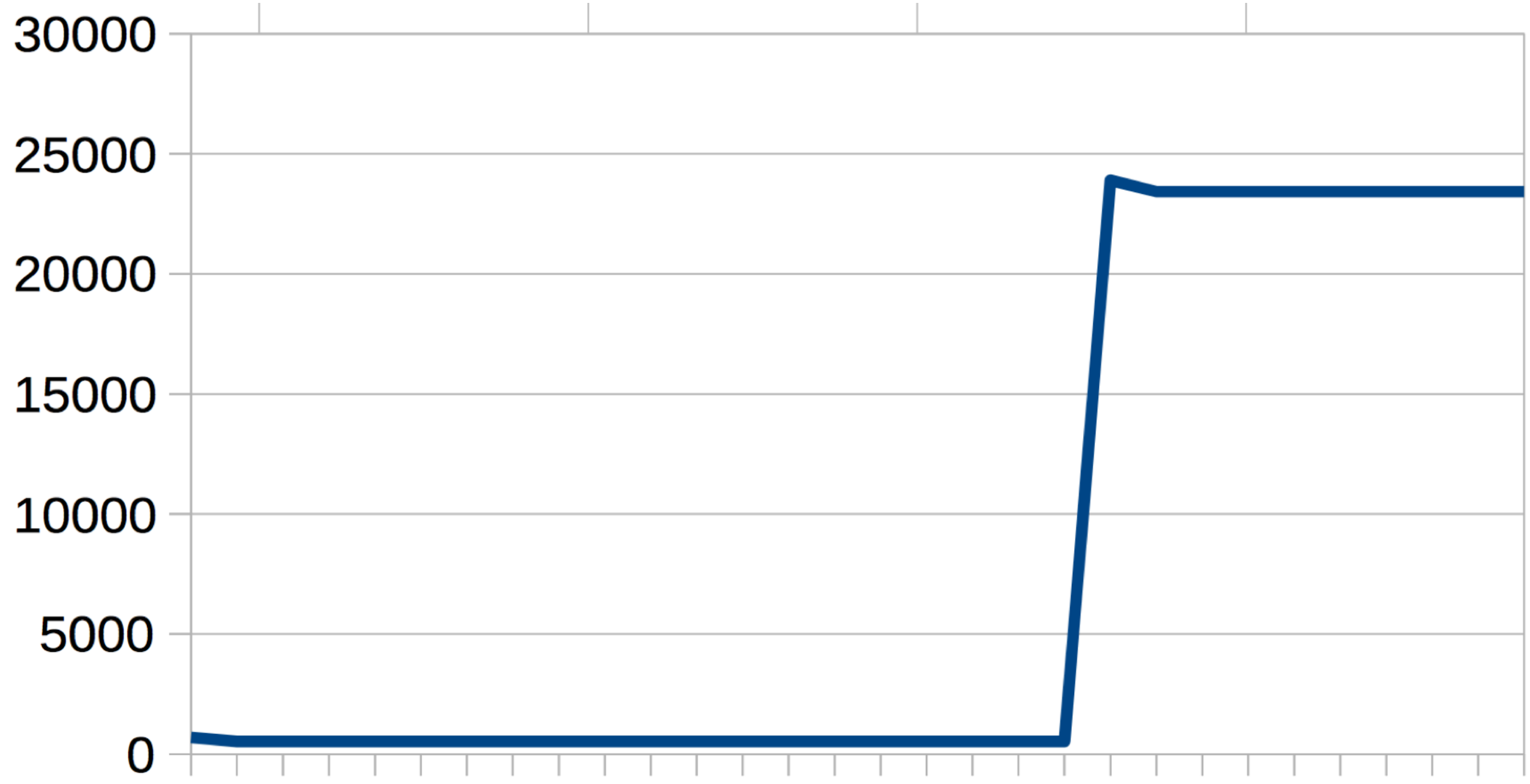


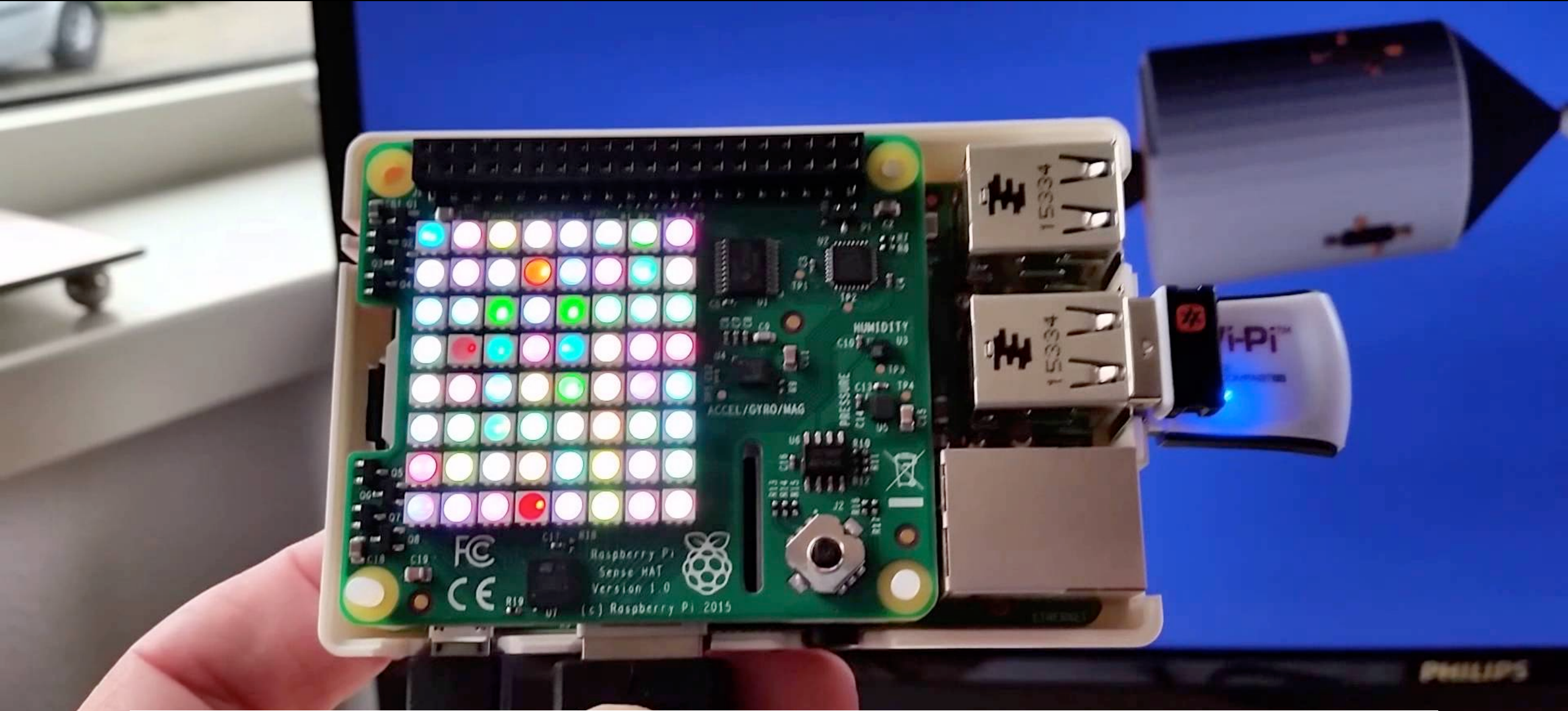
Learning Method	RMSE	% RMSE
Kernelized Regression	1,540	8.3%
Frequency NN	1,251	6.7%
Deep Feedforward NN	1,103	5.9%
Deep Recurrent NN	530	2.8%

Deep Autoencoder

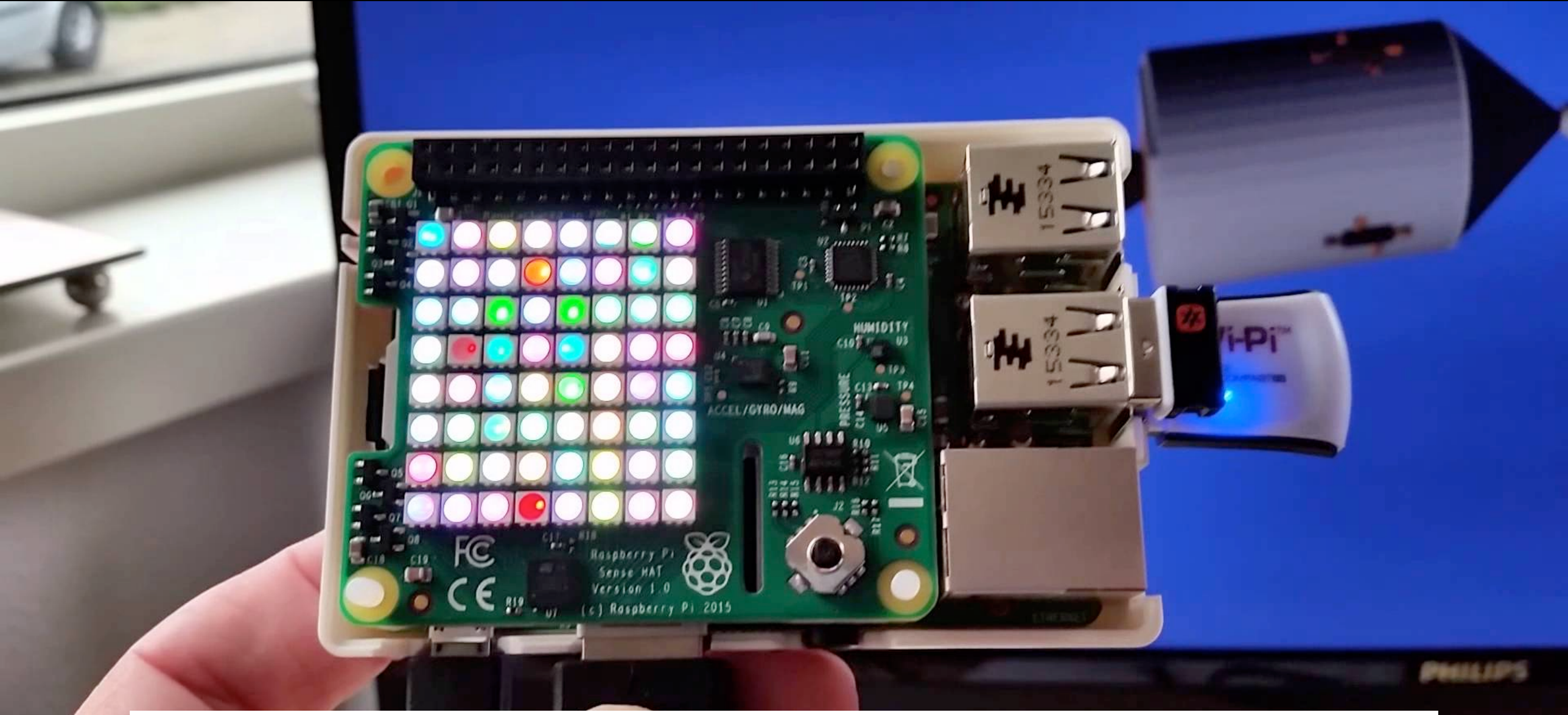



20
18
16
14
12
10
8
6
4
2
0
Loss
Training Epochs


30000
25000
20000
15000
10000
5000
0




```
import ibmiotf.gateway
from sense_hat import SenseHat
gatewayOptions = {"auth-method": "token", "auth-token": "s*)3uX"}
gatewayCli = ibmiotf.gateway.Client(gatewayOptions)
gatewayCli.connect()
sense = SenseHat()

temp = sense.get_temperature()
myData = {'temp' : temp}
gatewayCli.publishDeviceEvent(myData,qos=2)
```




```
import ibmiotf.gateway
from sense_hat import SenseHat
gatewayOptions = {"auth-method": "token", "auth-token": "s*)3uX"}
gatewayCli = ibmiotf.gateway.Client(gatewayOptions)
gatewayCli.connect()
sense = SenseHat()
```

```
at Run$MyEventCallback$2$.processEvent(Run.scala:43)
at com.ibm.iotf.client.app.ApplicationClient.messageArrived(ApplicationClient.java:973)
at org.eclipse.paho.client.mqttv3.internal.CommsCallback.deliverMessage(CommsCallback.java:477)
```

A developer's guide to the Internet of Things (IoT)

IBM

coursera

A developer's guide to the Internet of Things (IoT)

Enroll Now

Starting May 2016

A developer's guide to Exploring and Visualizing IoT Data

Created by: IBM

A developer's guide to Exploring and Visualizing IoT Data

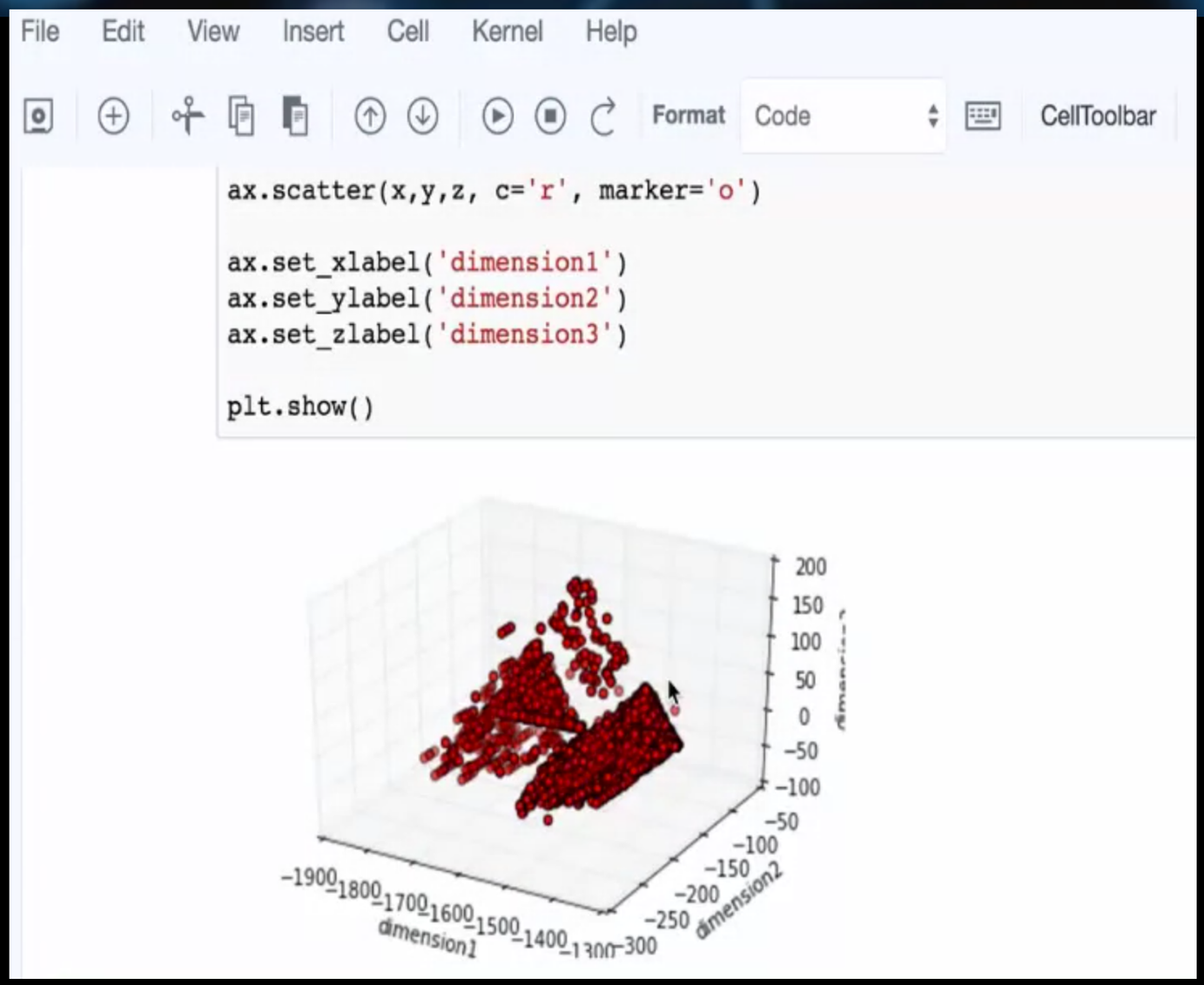
Enroll Now

Starting January 9, 2017

Taught by: Romeo Kienzler, Chief Data Scientist

IBM Watson IoT

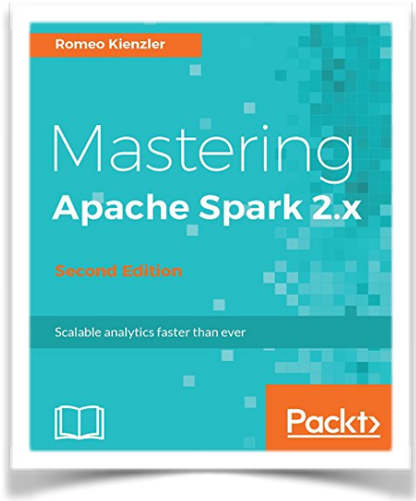
A developer's guide to Exploring and Visualizing IoT Data



Disclaimer: This is a beta talk - don't kill me - Disclaimer: This is a beta talk - don't kill me - Disclaimer: This is a beta talk - don't kill me

YouTube Romeo Kienzler 447 subscribers **http://nopanic.ai/tv**

http://nopanic.ai/slides



A developer's guide to Exploring and Visualizing IoT Data

http://nopanic.ai

IBM Bluemix **http://bluemix.net**

http://datascience.ibm.com

Created by: IBM

IBM

Taught by: Romeo Kienzler, Chief Data Scientist

IBM Watson IoT

?