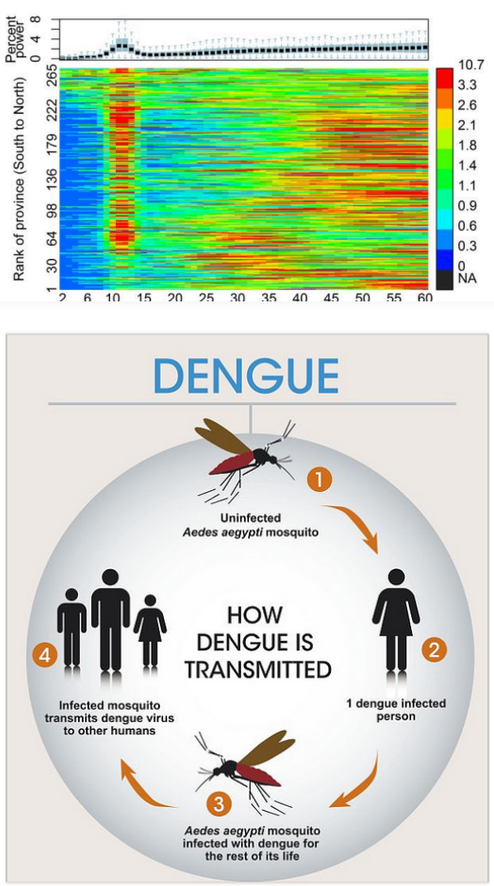
Towards Machine Learning in Pharo with TensorFlow

Dr. Serge Stinckwich

SergeStinckwich

UMMISCO, an International Joint Research Unit about Modelling and Simulation of Complex System

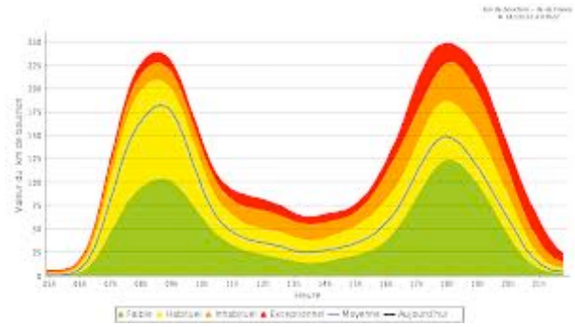
UMMISCO build and develop tools&methods to be applied to southern countries issues

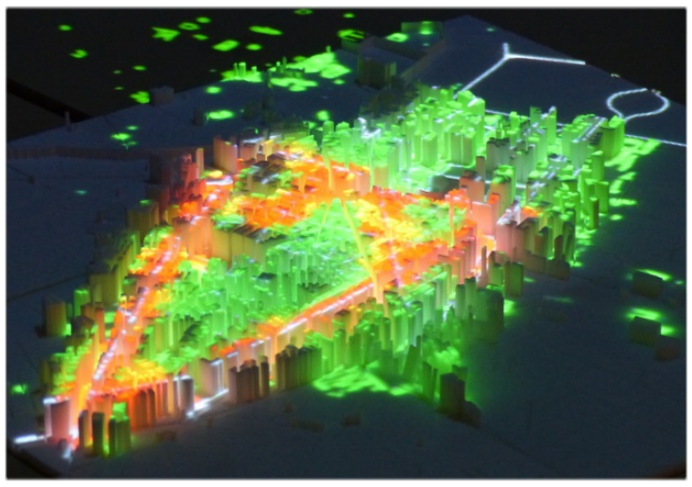


390 million dengue infections/year

Epidemic control

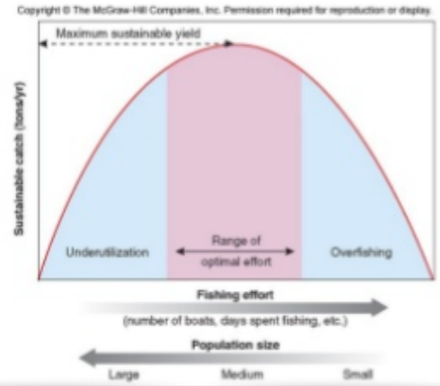
System Biology

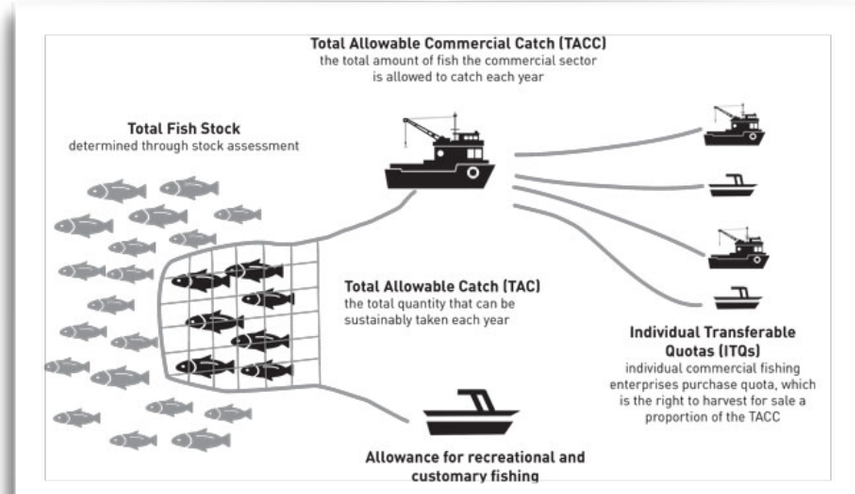




Cities Pollution

System Sociology



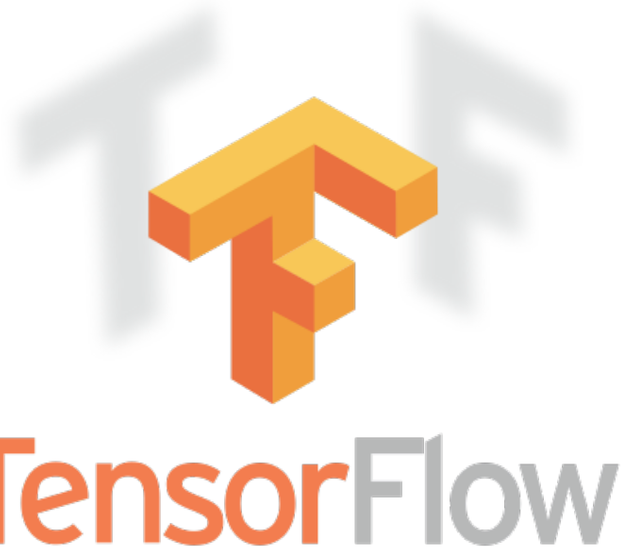


Sustainable fishing effort

System Ecology

What is TensorFlow ?

- A general purpose numerical computing library
- Open-source software
- Developed originally by researchers from Google Brain in 2015
- Written in Python, C++
- Hardware independent: CPU (Eigen/BLAS), GPU (CUDA/CuDNN), …



TensorFlow basics

- Tensors
- Data Flows
- Runtime execution

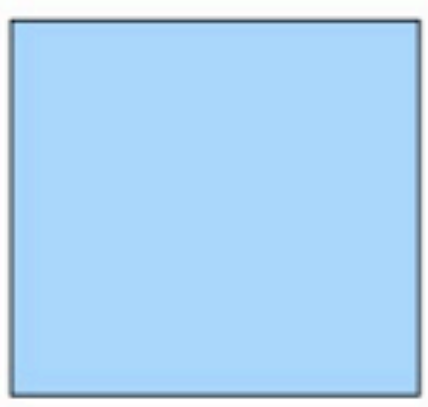
What are Tensors ?

- Tensors are multi-dimensional arrays

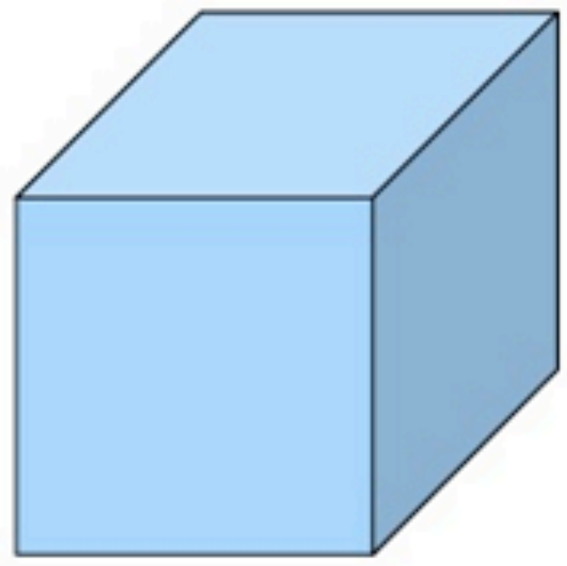
- TensorFlow supports: float16, float32, float64, bfloat16, complex64, complex128, int8, uint8, uint16, uint32, uint64, int16, int32, int64, bool, string, qint8, quint8, qint16, quint16, qint32

Rank	Math entity
0	Scalar
1	Vector
2	Matrix (2-Tensor)
3	3-Tensor
n	n-Tensor

1d-tensor

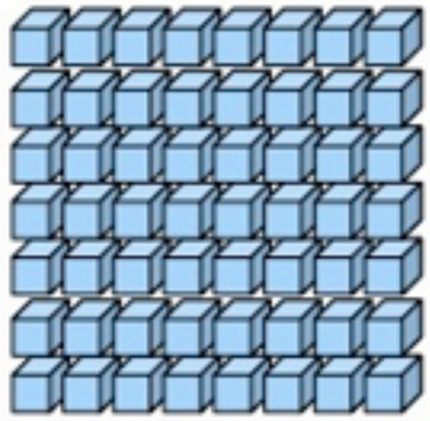


2d-tensor

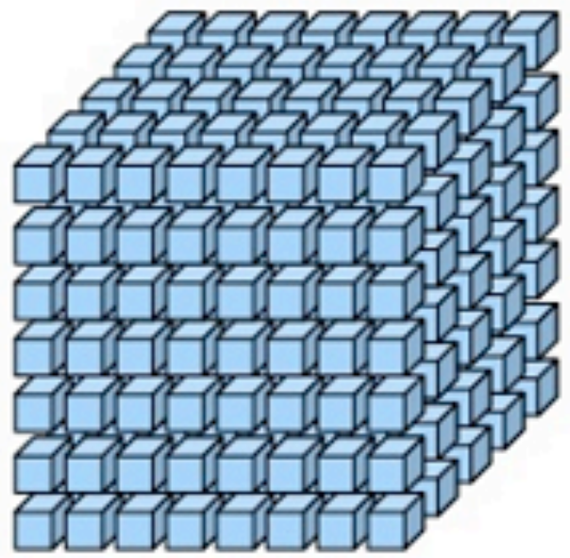


3d-tensor

4d-tensor



5d-tensor



6d-tensor

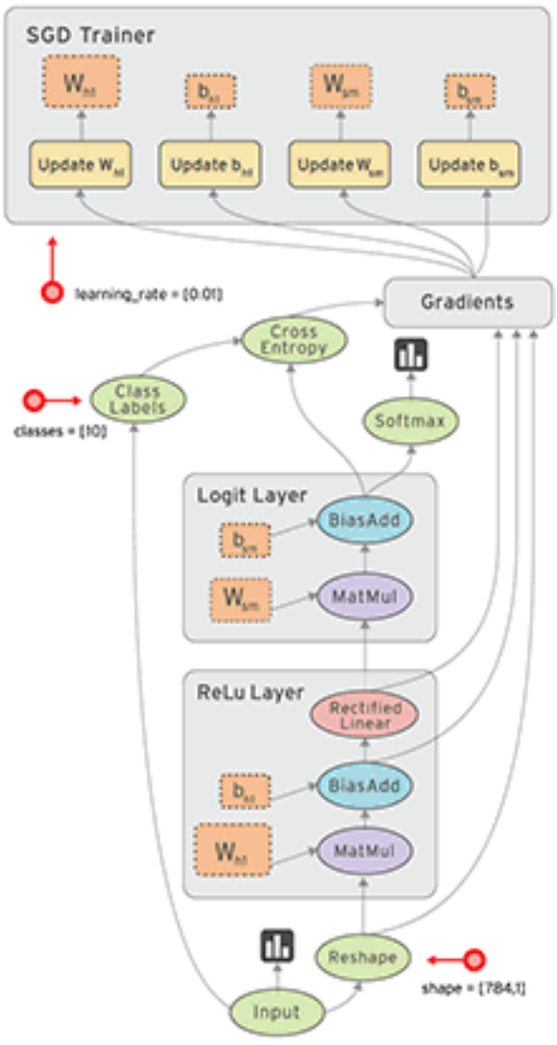
Applications of TensorFlow

- Originally: Quantum Physics
- Mainly: AI, Machine/Deep Learning
- but not only: Data analysis, image processing, big data, simulation, BioInformatics, computational neurosciences

Why Pharo should take care about TensorFlow?

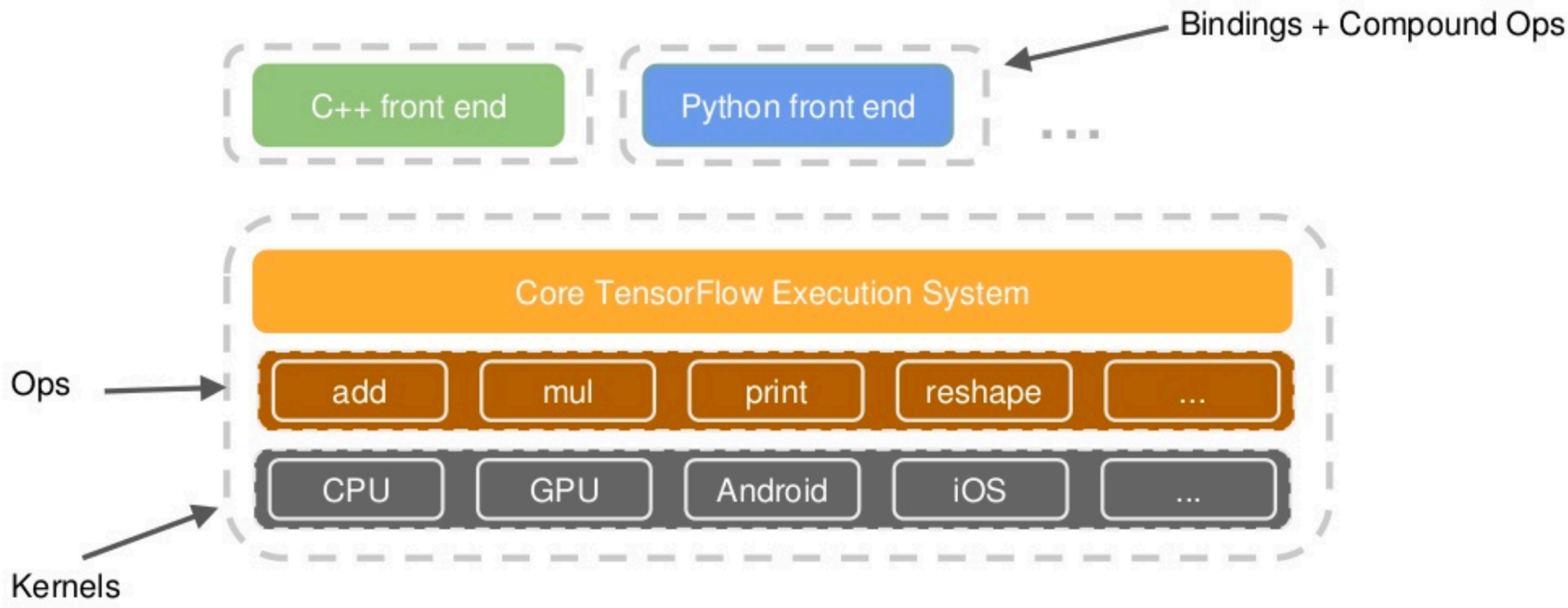
- Multi-platform library
- Distributed execution (CPU, GPU, TPU)
- Robust foundation for doing machine learning and deep learning frameworks (but not only)
- TF already support: **C++, Python**, Java, R, Go, JavaScript, Scala, …
- Will be more fun to build live TF code.

What is the meaning of flow in TensorFlow ?



- nodes = operations: math functions, constants (initializing values), logging data, …
- edges = data (tensors) flows between nodes
- Dataflows are defined in Pharo, compiled in memory with TensorFlow and then executed on devices (CPU, GPU, …)

TensorFlow Architecture



libTensorFlow C API

- TF main functionalities are exposed through a C layer
- We use UFFI (Unified Foreign Function Interface) to connect Pharo to TensorFlow

- https://github.com/tensorflow/tensorflow/blob/r1.8/tensorflow/c/c_api.h

How to use C API

TensorFlowCAPI>>version

"TF_Version returns a string describing version information of the TensorFlow library. TensorFlow using semantic versioning."

"TF_CAPI_EXPORT extern const char* TF_Version();"

^ self ffiCall: #(String TF_Version #()) module: TensorFlowCAPI

TensorFlowCAPI uniqueInstance version. => ‘1.10.1’

libTensorFlow-bindings-Pharo

- Fork from CUIS TensorFlow bindings (FFI not exactly the same). Thank you **Javier Burroni&Gerardo Richarte**.

- Part of **PolyMath** project: https://github.com/PolyMathOrg/libtensorflow-pharo-bindings

- Works on TF 1.10 on MacOS, Linux and Windows.

- More than 100 green unit tests

- Crash until recently ! Still finalization issues

Build Tensors

- Float pi asTensor.
- TF_Tensor fromFloats: #(1 2 3 4).
- TF_Tensor fromFloats: #((1 2 3 4 5)(6 7 8 9 10)).

Ranks, shapes

- pisTensor := TF_Tensor fromFloats: #(3.14 3.1415 3.141516).
- pisTensor rank. => 1
- pisTensor shape. => #(3)
- (TF_Tensor fromFloats: #((1 2 3 4 5)(6 7 8 9 10))) shape.

Graphs

- A Graph contains a set of Operation objects, which represent units of computation; and Tensor objects, which represent the units of data that flow between operations.

- Graph can be serializable (as protobuff) and can be exchanged between platforms.

- TF_Graph create.

Operations

- Operation is a node in a TF Graph that takes 0, 1 or n tensors as inputs and produces 0, 1 or n tensors as outputs.

- Example of TF operations :
  - Arithmetic operators: Add, Multiply, Mod, etc …
  - Mathematic functions: Sin, Exp, Pow, etc …
  - Matrix math functions: Transpose, Inverse, Norm, …
  - Reduction dimensions&Segmentation

Runtime execution

- After the data flow has been defined, the graph is executed within a session and inside a device (CPU, GPU, …).

- Distributed execution: graph can be splitted on many devices

- Portable: dataflow graph is a language-independant representation of the code (could be re-used with another language)

Demos

3+4 in TensorFlow

```
graph := TF_Graph create.
c1 := graph const: 'c1' value: 3.0 asTensor.
c2 := graph const: 'c2' value: 4.0 asTensor.
sum := c1 + c2.
session := TF_Session on: graph.
result := session runOutput: (sum output: 0).
result asNumbers
```

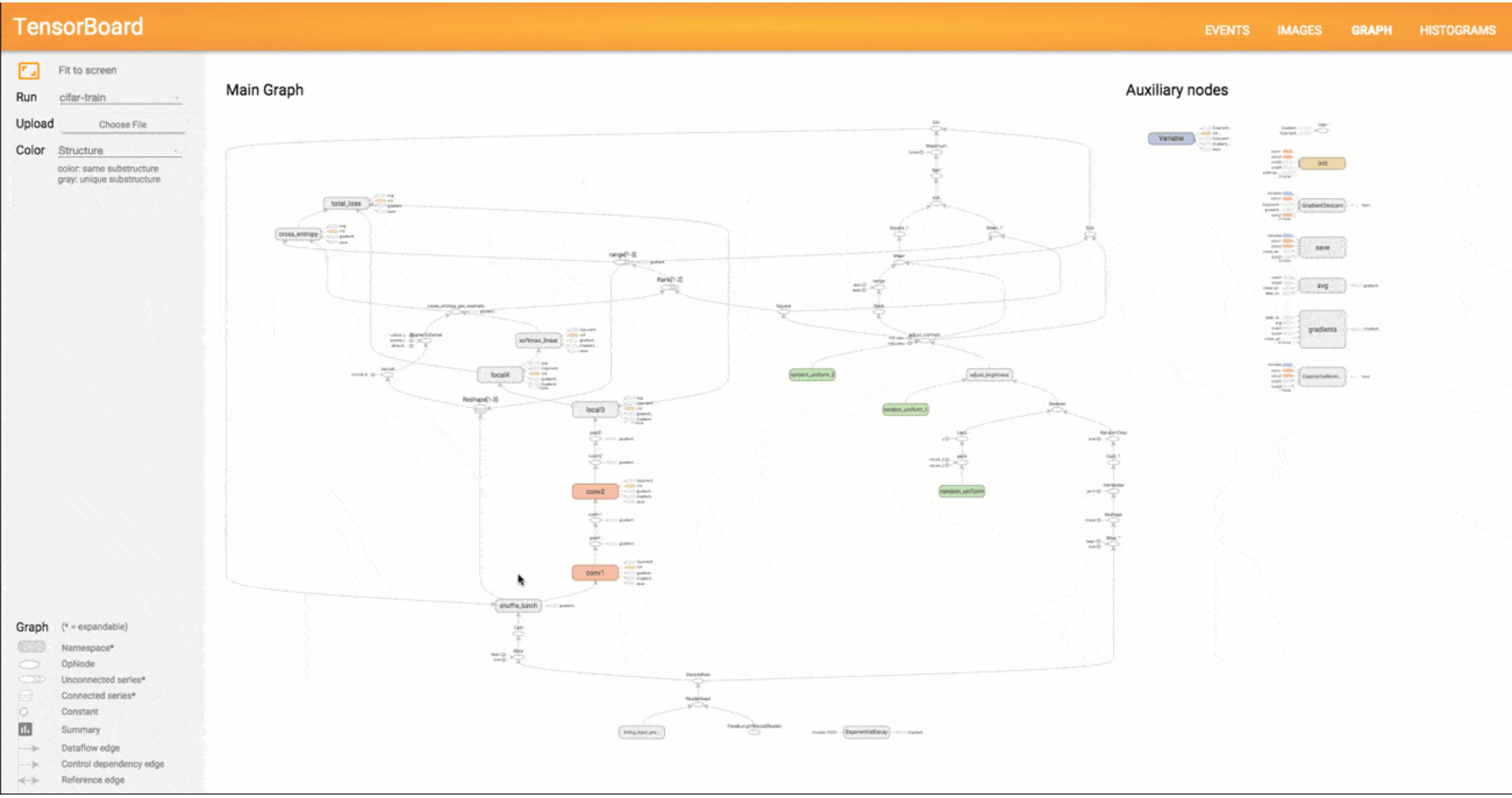
Multiply two matrices

```
graph := TF_Graph create.
t1 := TF_Tensor fromFloats: #((1 2)(3 4)).
t2 := TF_Tensor fromFloats: #((5 6)(7 8)).
c1 := graph const: 'c1' value: t1.
c2 := graph const: 'c2' value: t2.
mult := c1 * c2.
session := TF_Session on: graph.
result := session runOutput: (mult output: 0).
result asNumbers
```

Neural Networks

Demos

TensorBoard



Conclusion

- First alpha version of the TF bindings available on github
- TF dataflows are difficult to debug because when they are executed this is outside Pharo (TensorFlow board)
- Still some random crashes from time to time (mostly due to finalization)

Perspectives

- More complex visualisations of data flows with Roassal (ongoing work of a M2 student)
- Build a DSL on top of TF bindings to ease ML&DP models building (like Keras)
- Use GlamourousToolkit & Moldable inspectors
- Use TF bindings for applications (PolyMath, DataFrame, Kendrick, etc …)

Don’t miss

- Thursday, 14pm30 : CORMAS, a participatory and interdisciplinary modeling platform, **P.Bommel, E.Delay**, C. Le Page, H. Morales, N. Becu, B. Bonte, N.Papoulias, S.Stinckwich, CORMAS Team

- Friday, 11am : PolyMath, **O. Zaytsev, S.Stinckwich**