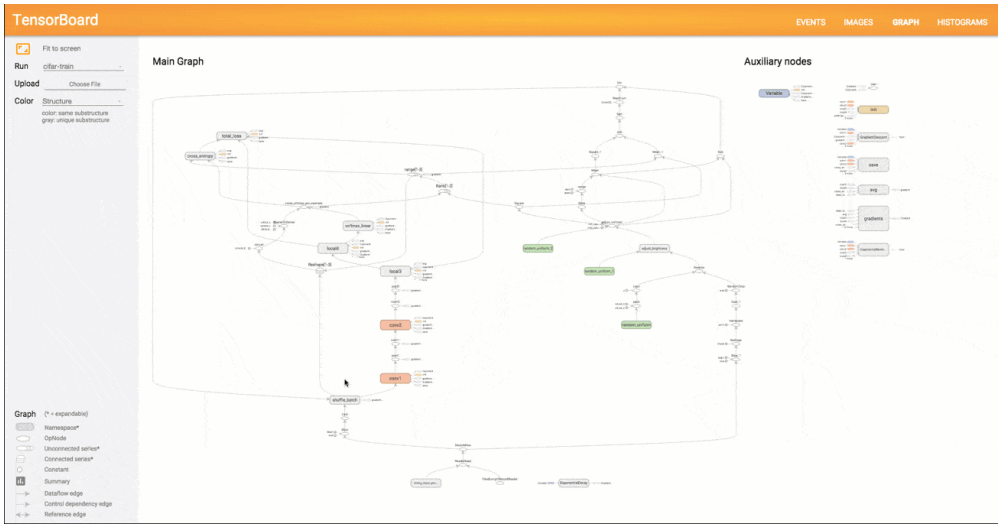
POLYTECH NANTES

UNIVERSITÉ DE NANTES



TensorFlow-Graph Visualizer

Mohammed Azaz & Philipp Behrendt

Nantes, 18th December 2018

Outline

The structure of our presentation

- Journal and Authors
- Structure of the paper
- Summary of key points:
  - Context
  - Graph Visualization
- Personal remarks

Publication information

- Paper title: Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow
- Authors: Wongsuphasawat K, Smilkov D, Wexler J, Wilson J, Mane D, Fritz D, Krishnan D, Viegas FB, Wattenberg M
- Published 2017, IEEE Transactions on Visualization and Computer Graphics
- Journal: this journal mostly contain computer graphics, information and scientific visualization.

About Authors

- Worked together for google research and at University of Washington
- Many common research projects on neural network and visulaziation
- They mostly graduated from top university (Standford, MIT, Berkely) and/or worked for big „players“ Google, IBM

Structure of the paper

Design Study of Deep Learning Model Visualization

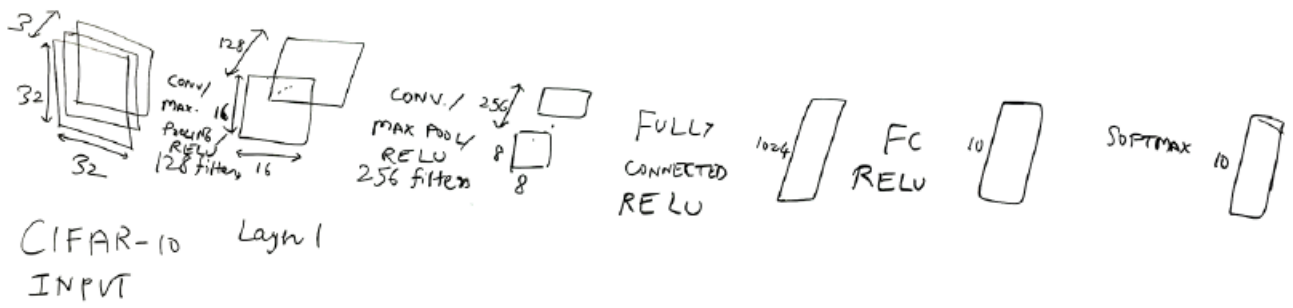
- Introduction
- Tasks and Challenges
- Design
- Usecase
- Feedback
- Discussion

The Motivation

Lack of Dataflow viz

- Many High-Level APIs for Deep Learning (Theano, Torch, TensorFlow, …) with **many** underlying Low-Level operations

➢ But **no** High-Level Model Visualization



The Motivation

Standard flow-layouts on Low-Level fail

- Displaying heterogenous Low-Level Operations is not helping

Tasks

The viz should …

- Give an immediate overview
- Indicate similarities and differences
- Examine nested structures
- Allow to inspect details
- Give summaries

Challenges

Why standard flow layouts perform poorly …

- Missmatch between graph topolgy and semantics
- Graph heterogenity
- Interconnected nodes

New Approach

How to adress tasks and challenges ?

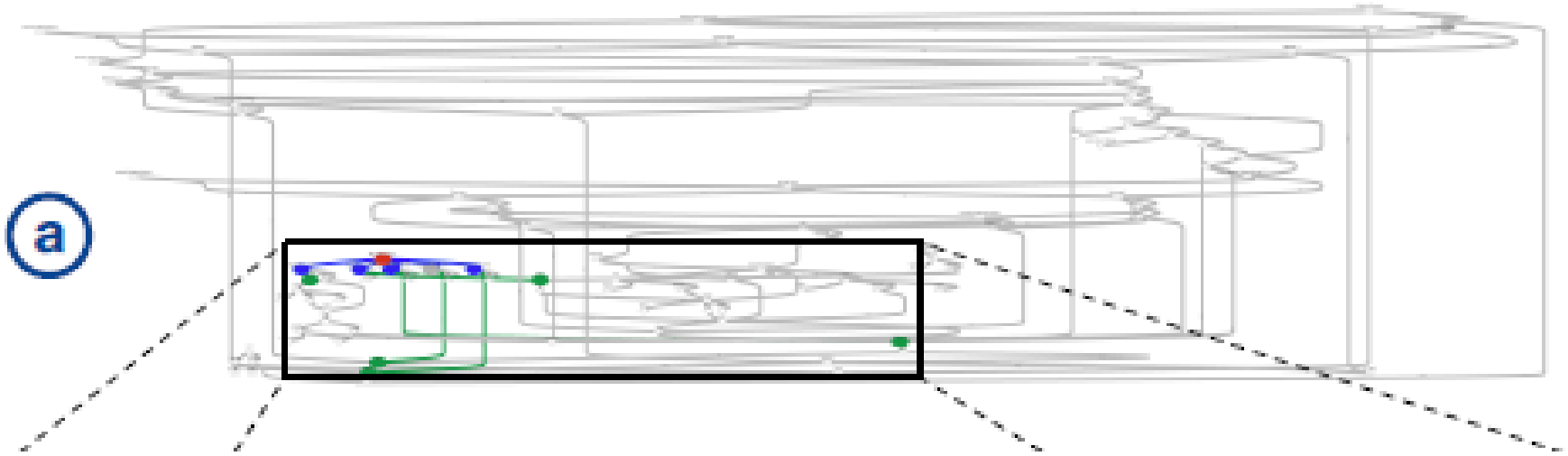
- General Bottom up flow*:*

- Extract Non-Critical Operations
- Build a Clustered Graph
- Bundled Edges
- Extract Auxiliary Nodes

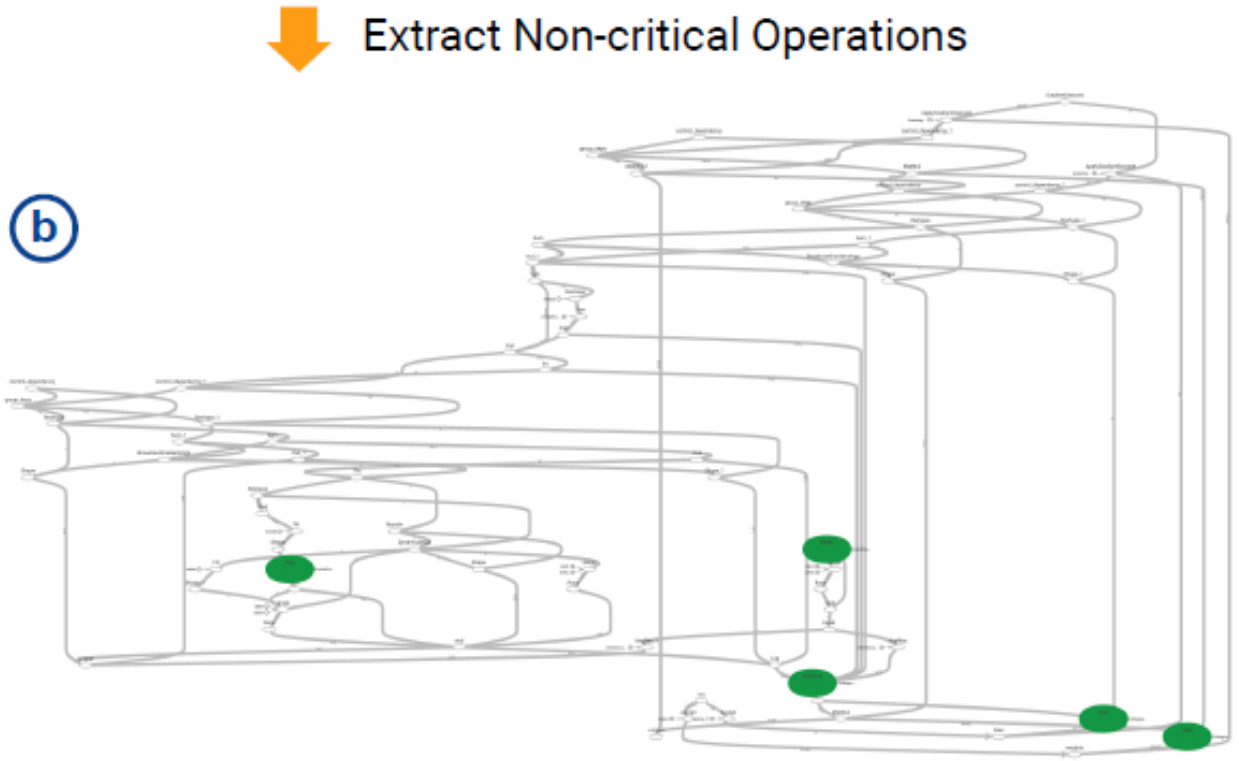
Extracting Non-Criticals

- Shrink the graph
- Remove Constants and Summaries

Starting Point – Low-Level Operations



Extract non Critical Operations



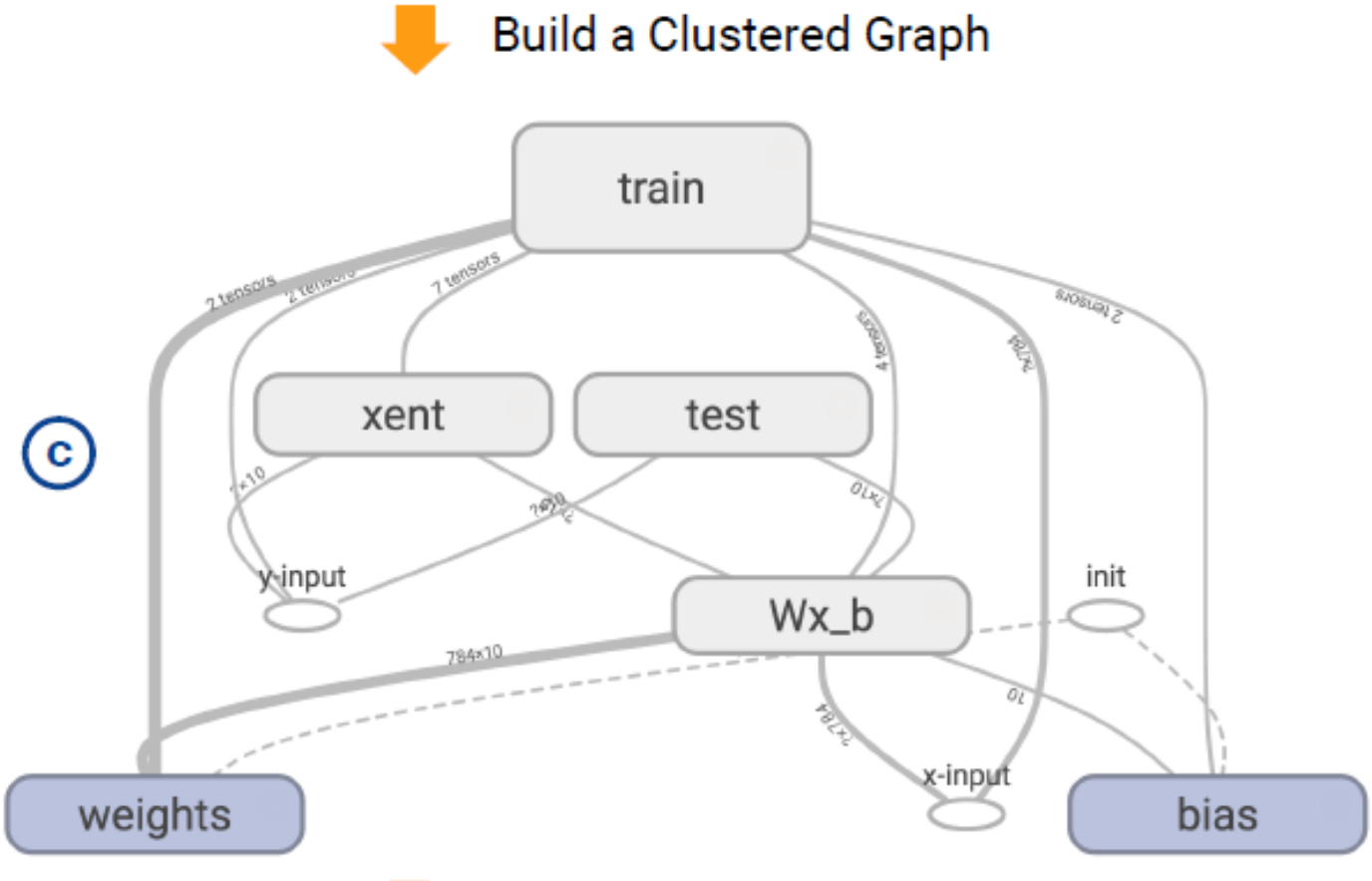
Build Clustered Subgraphs

- „Summarize“ different operations into a *group nodes*
- Clustered by namespace
- Builds a hierachy similar to a folder structure
- The bigger a node, the more operations

Bundle Edges

- Avoid messy and cluttered layouts
- Draw only edges within the same subgraph / in the same hierachy
- Edges can be recursively inferred
- Advantage: Re-Rendering only subgraphs after zooming
- Drawback: Harder to follow a trace

Clustered Subgraphs



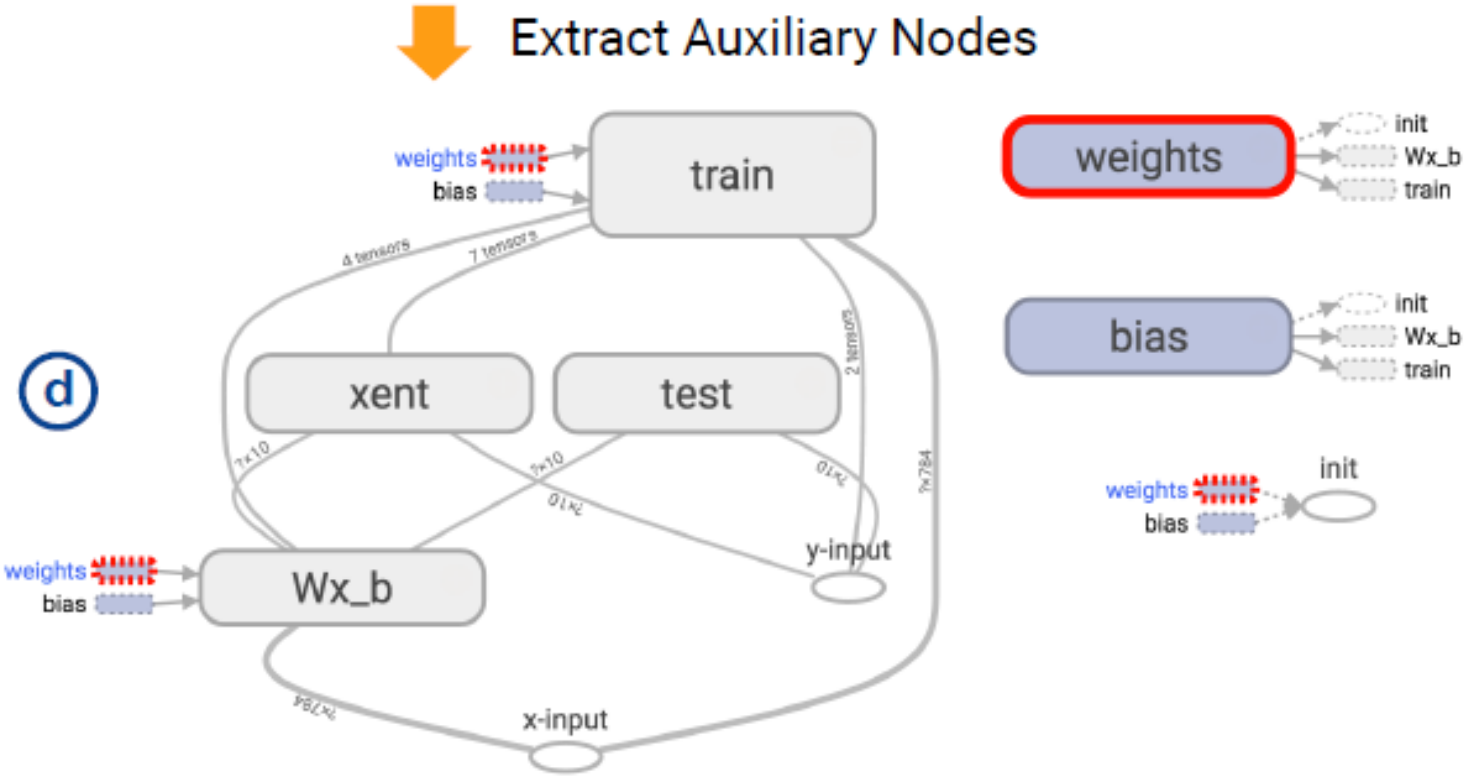
Extracting Auxiliary Nodes

- Some operations are conducted for many group

➢ Intertwinning edges and clutter layout

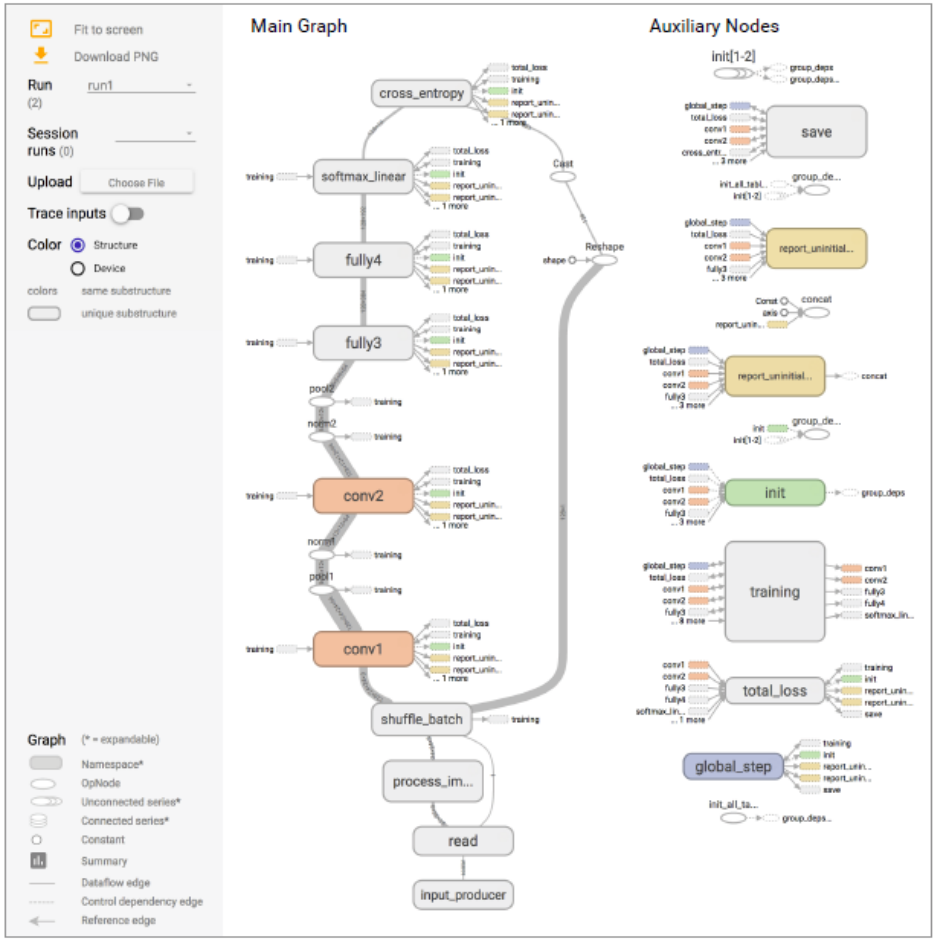
- Put these auxiliary nodes apart and make them accessible through icon

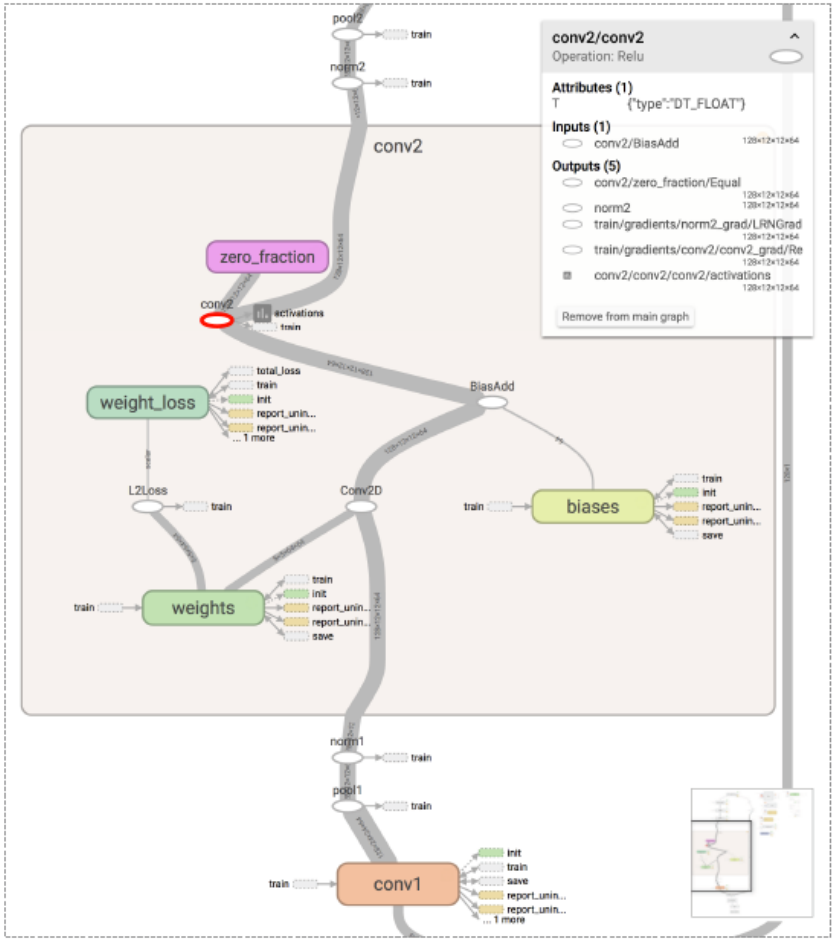
Extracting Auxiliary Nodes



Additional features

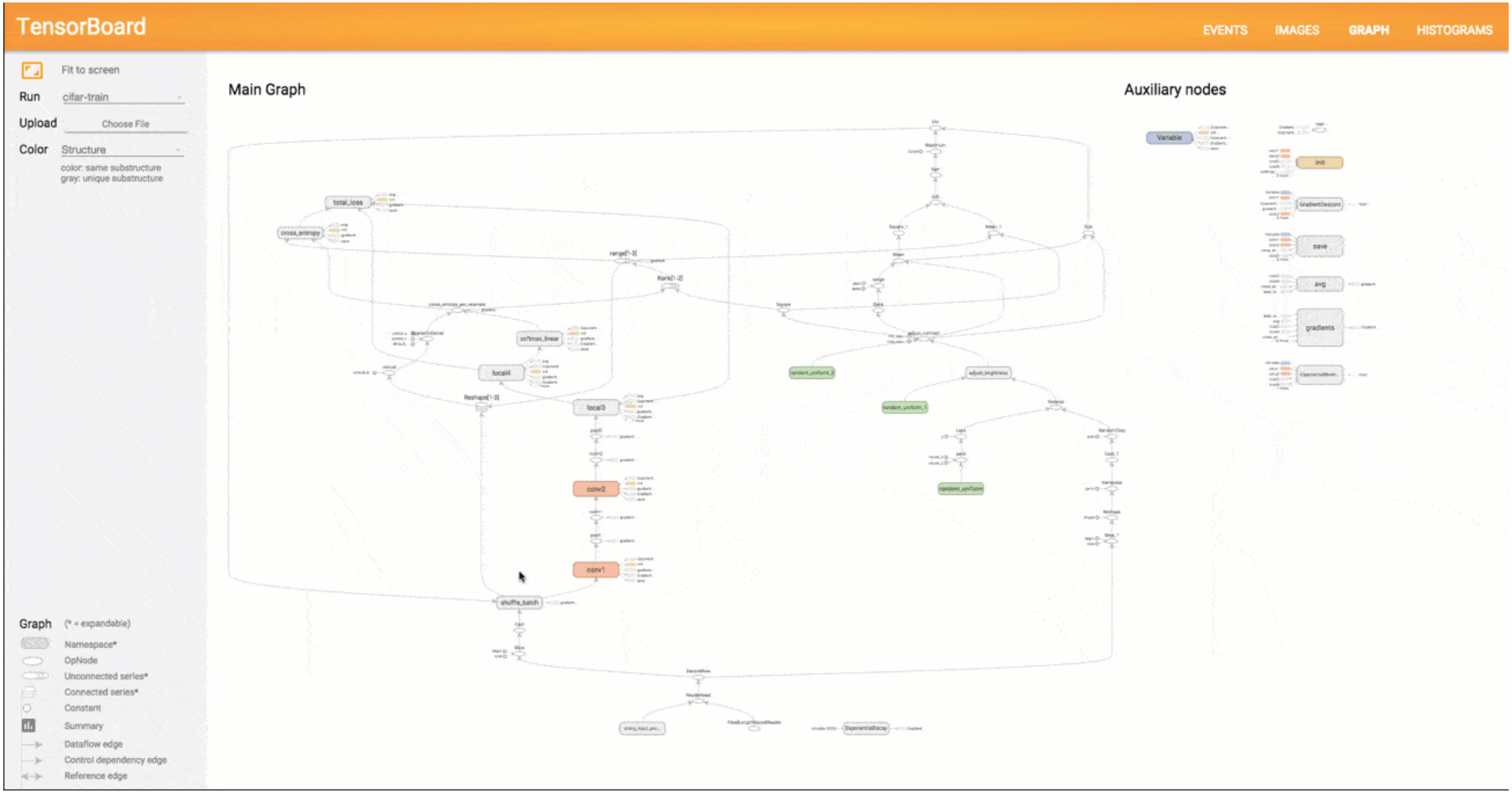
- Check for subgraph similarity – map same color
- Indicate type of relation
  (data relation → line, control relation → dotted, …)
- Provide dimensions of tensors
- Further mappings (round shapes → single operation,
  thickness of edges → size of data flow, …)





TensorFlow Graph Visualizer - Usecase

Applying the Graph-Visualizer on a Deep Learning Script for MNIST-Data Digit Classification



TensorFlow Graph Visualizer - Example

Applying the Graph-Visualizer on a Deep Learning Script cifar – Dataset (https://www.tensorflow.org/guide/graph_viz)

Evaluation

- Internal Feedback with questionaire : Positive results, high usability
- External Feedback collected from online forums: „Step in the right direction“

Discussion

- Meaningful contribution to interactive exploration of the dataflow models in machine learning

- Theoretical Implications: Extraction of non-critical nodes can be applied to other domains
  - Allowing High-Level Models from Low-Level Operations

- Practical open questions: Comparing Models, direct editing

- Clustering by namespace requires “good coding – meta annotation”

Personal Remarks

- Nice and handy tool to display model structure and also to debug code
- Profund knowledge in graph theory for the different reduction steps
- Feedback report is not too valid/reliable

POLYTECH NANTES

UNIVERSITÉ DE NANTES

Thank you for your attention!

Appendix

- Namespace explanation

```
import tensorflow as tf

with tf.name_scope('hidden') as scope:
  a = tf.constant(5, name='alpha')
  W = tf.Variable(tf.random_uniform([1, 2], -1.0, 1.0), name='weights')
  b = tf.Variable(tf.zeros([1]), name='biases')
```