Considerations for Large Scale Analytics in Production

by Yuan Tang
@terrytangyuan

Agenda

- Accelerate large scale ML
- Complex and unexpected data characteristics
- Model lifecycle management
- Visualization on ML experiments

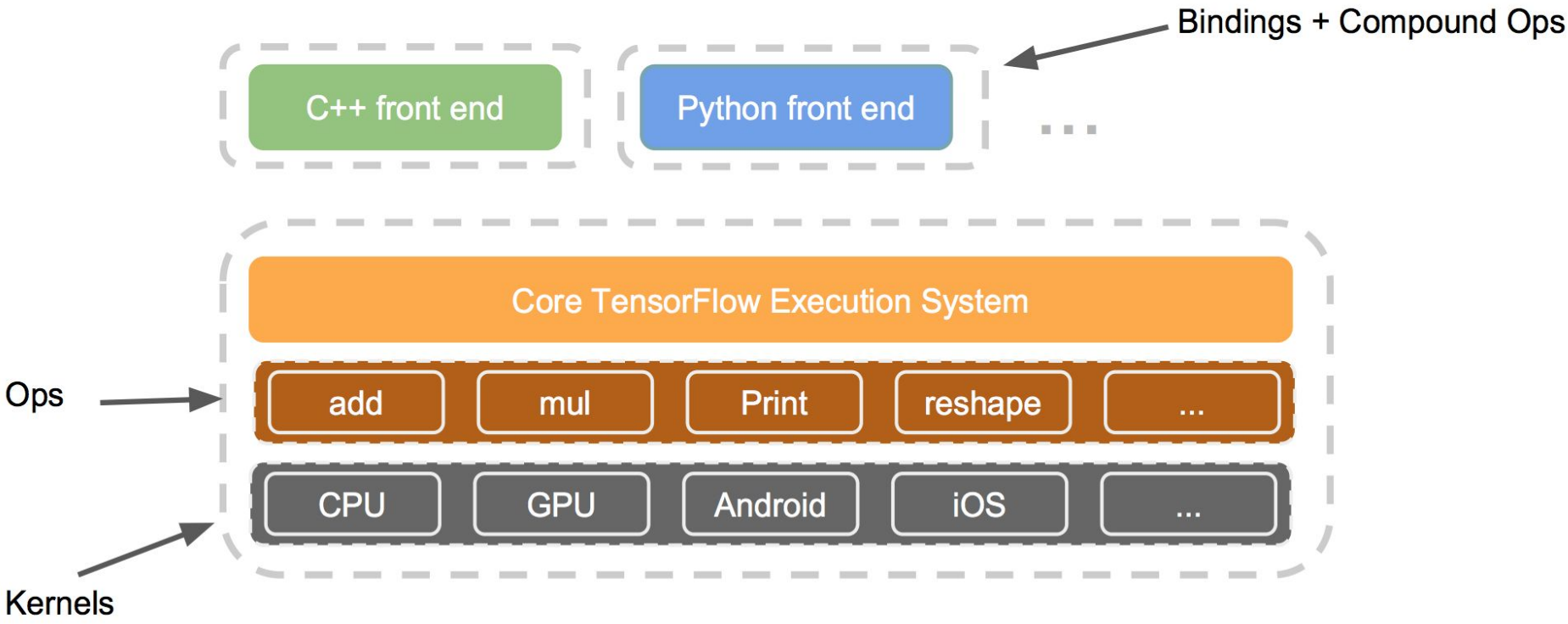
Hardware Accelerators

- GPUs
  - NVDIA cuDNN (training)
  - NVDIA TensorRT (inference)
- Virtual Cloud TPUs
  - Google Cloud (inference)


TESLA
S1070
24
RESET
23
Microway
TESLA
S1070
22
RESET
Microway
21

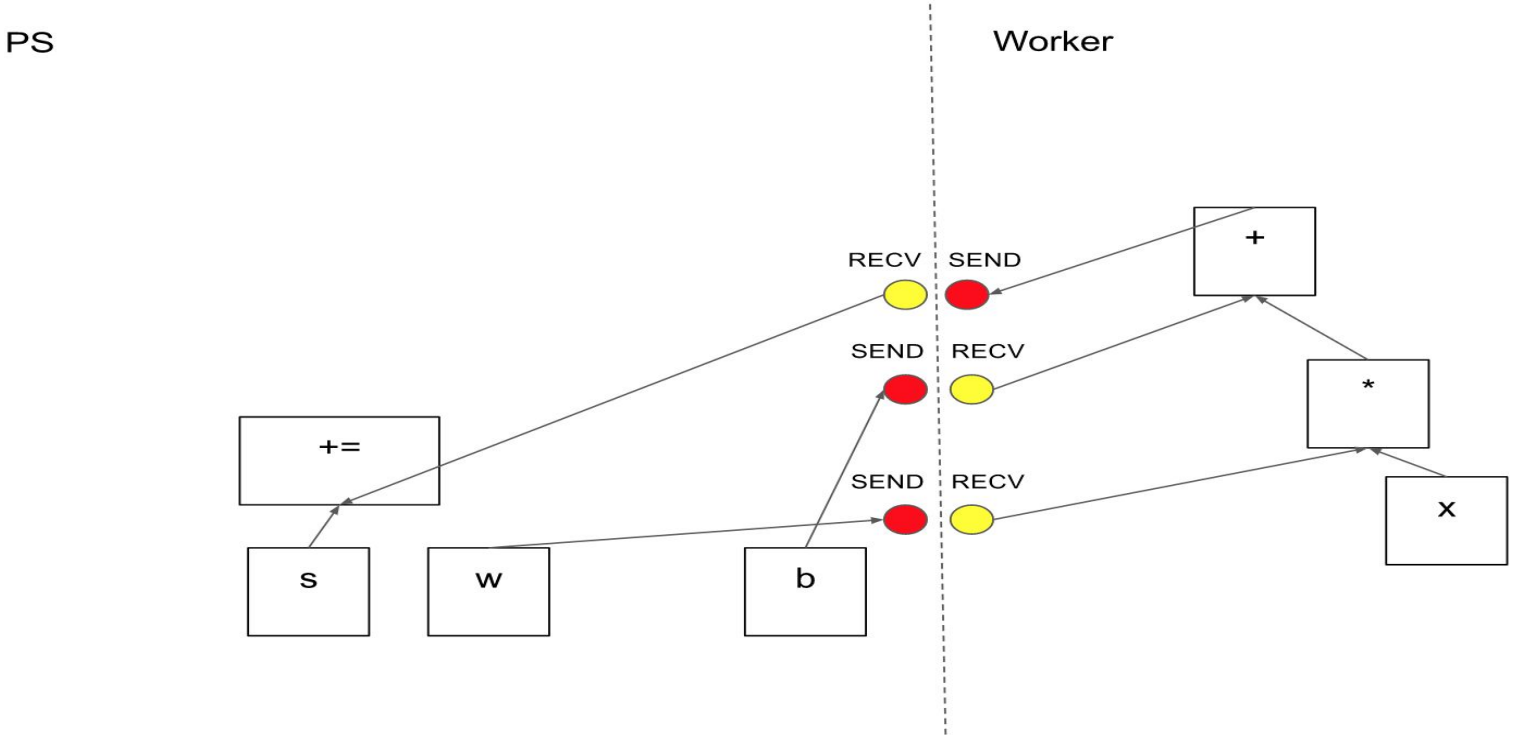

Titan X
VS
GTX 980
VS
Tesla M40
VS
Tesla K80
TITAN
GEFORCE GTX
GTX 980
TESLA

TensorFlow Architecture

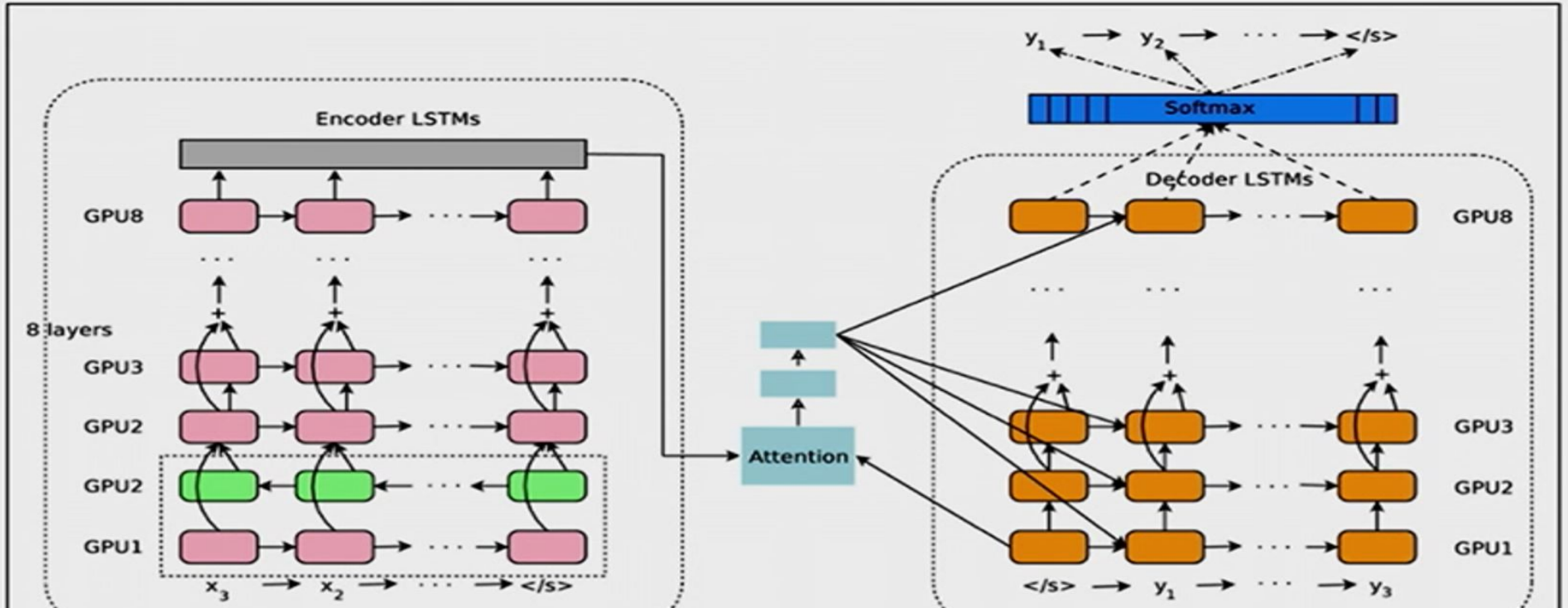


Distributed Training

Data Parallelism



Model Parallelism



Large Dataset

Large Dataset

- Out-of-core training
- Distributed file systems and dataset representations
- In-database training
- Mini-batches/streaming

Model Deployment

Model Deployment

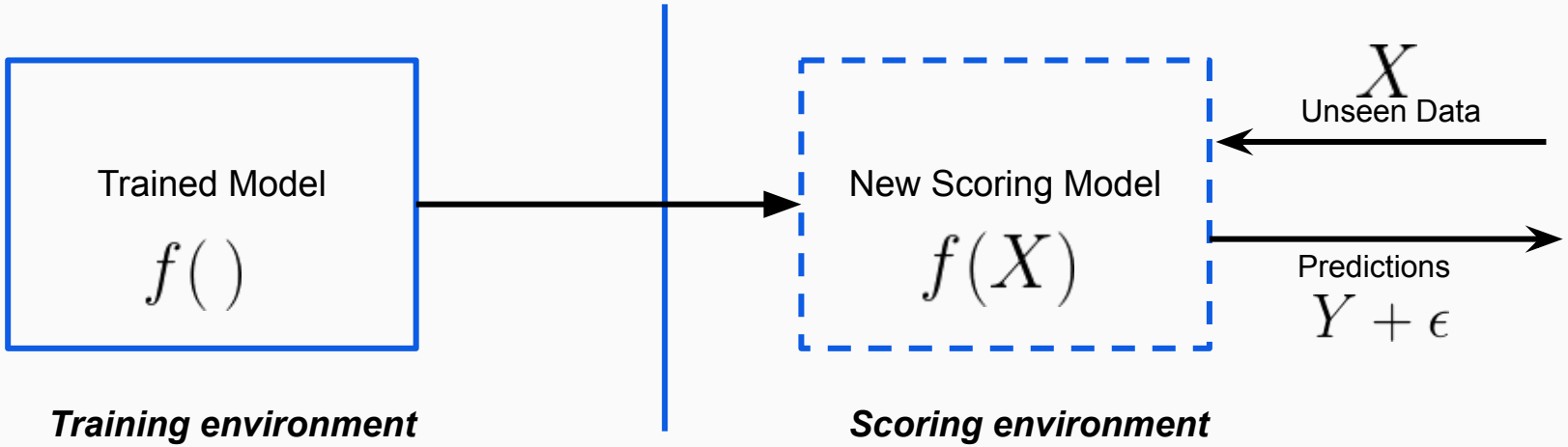
During training generally we're looking to approximate a target **Y** by finding ***f*** to solve the following…

$$Y = f(X) + \epsilon$$

Often ***f*** is highly complex and nonlinear

where error ϵ is out of our control and minimized

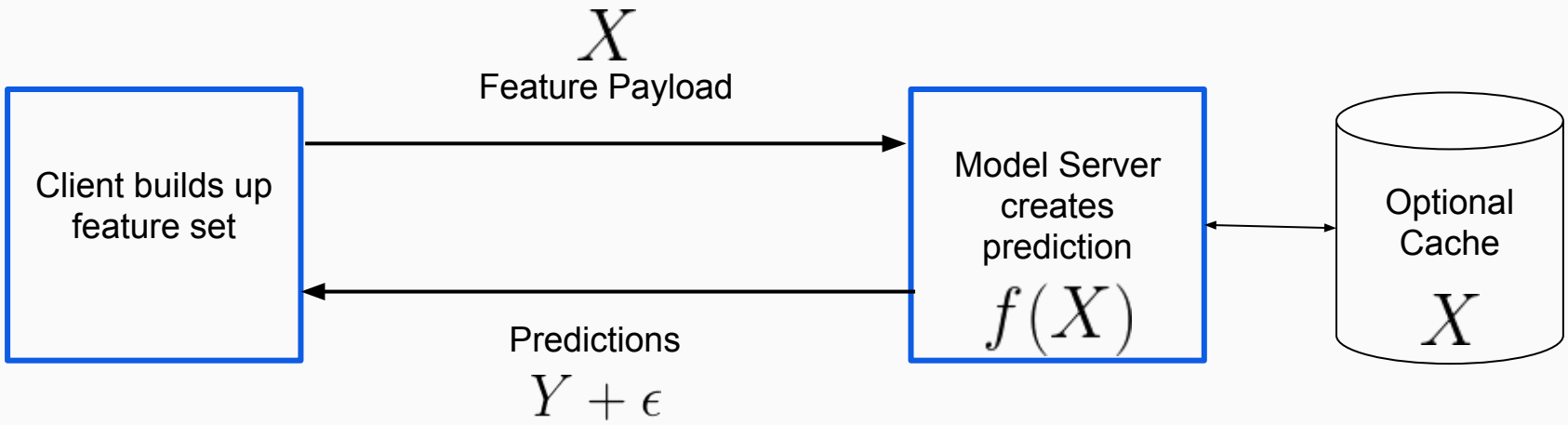
Model Deployment



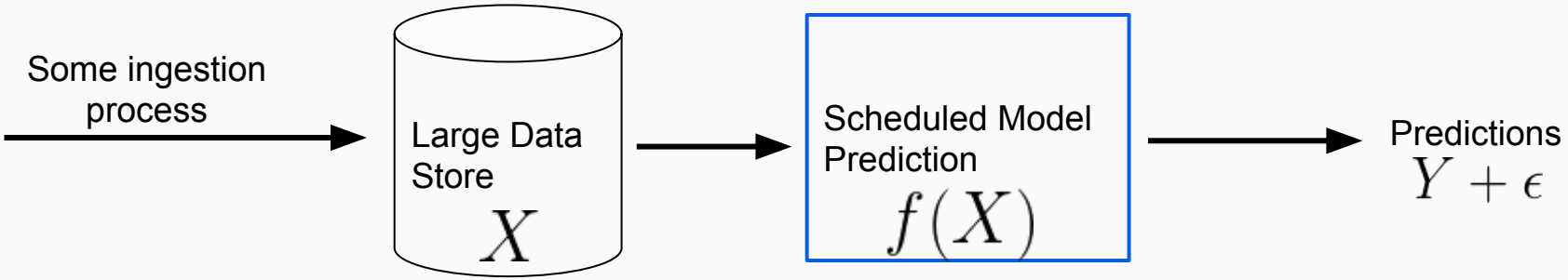
Model Deployment - Types of Artifacts

- Predictive Model Markup Language (PMML)
- Plain Old Java Object (POJO) or a Model Object, Optimized (MOJO)
- Portable Format for Analytics (PFA)
- TensorFlow 's SavedModel (mobile optimized version - TensorFlow Lite)
- Open Neural Network Exchange (ONNX) - a standard format for models built using different frameworks (e.g. TensorFlow, MXNet, PyTorch, etc.)

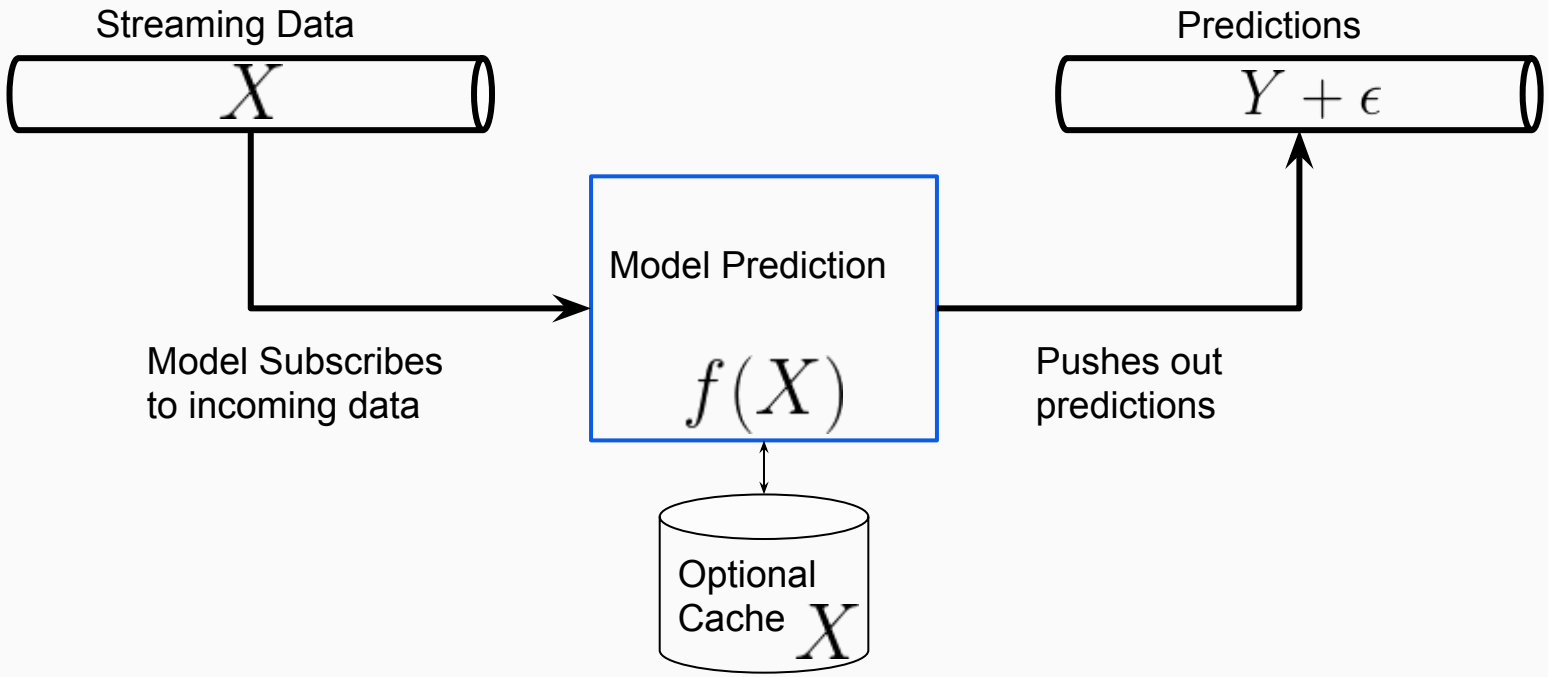
Model Deployment - REST APIs



Model Deployment - Batch



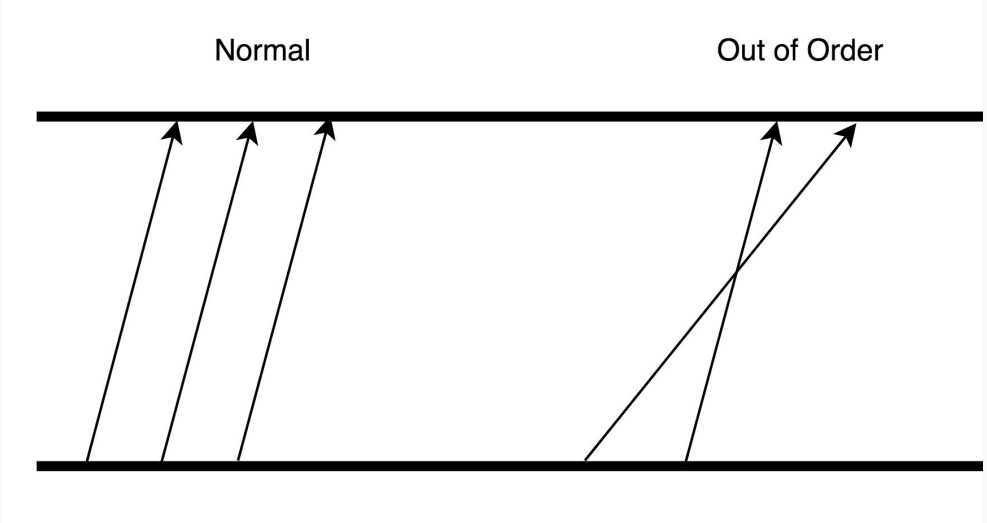
Model Deployment - Streaming



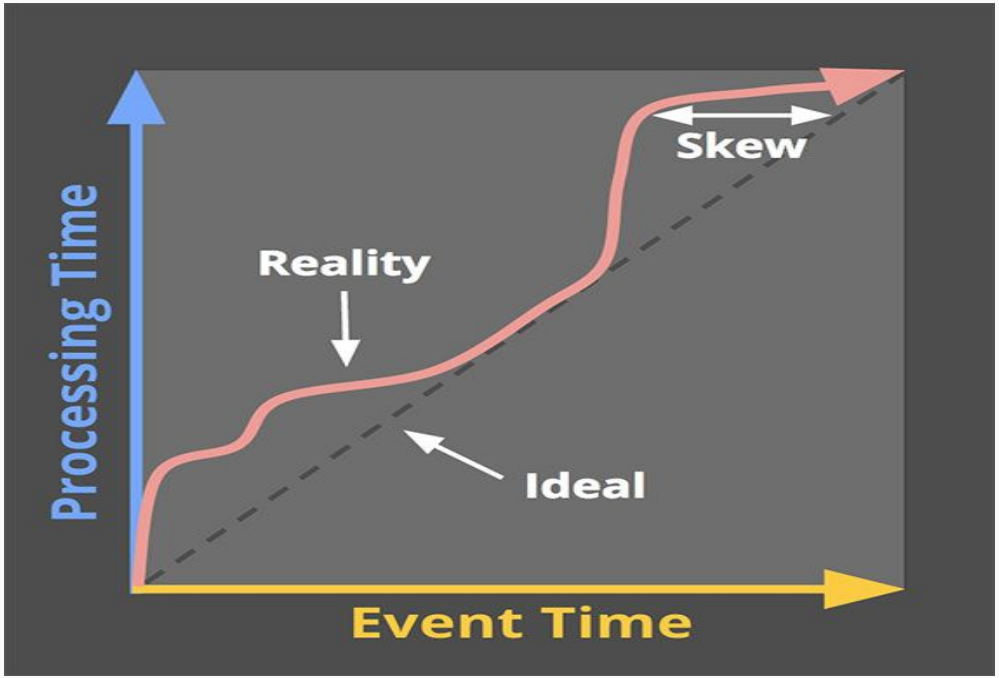
Time Sensitive and Streaming Data

- Event time
- Processing time

Time Sensitive and Streaming Data



Time Sensitive and Streaming Data



Concept Drift

- Condition monitoring on industry assets
  - Taking actions based on the predictions
  - Sensors are malfunctioning
- Retraining
  - Batch retraining
  - Online learning

Data Validation

- Anomalies detection
  - Summary statistics
  - Schema changes
  - Missing values
- Rolling changes to production models
  - Custom data validation rules

Train-serve Skew

- Differences in:
  - Statistical distributions
  - Processing topologies
  - Programming languages
  - ML frameworks

Adversarial Attacks

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

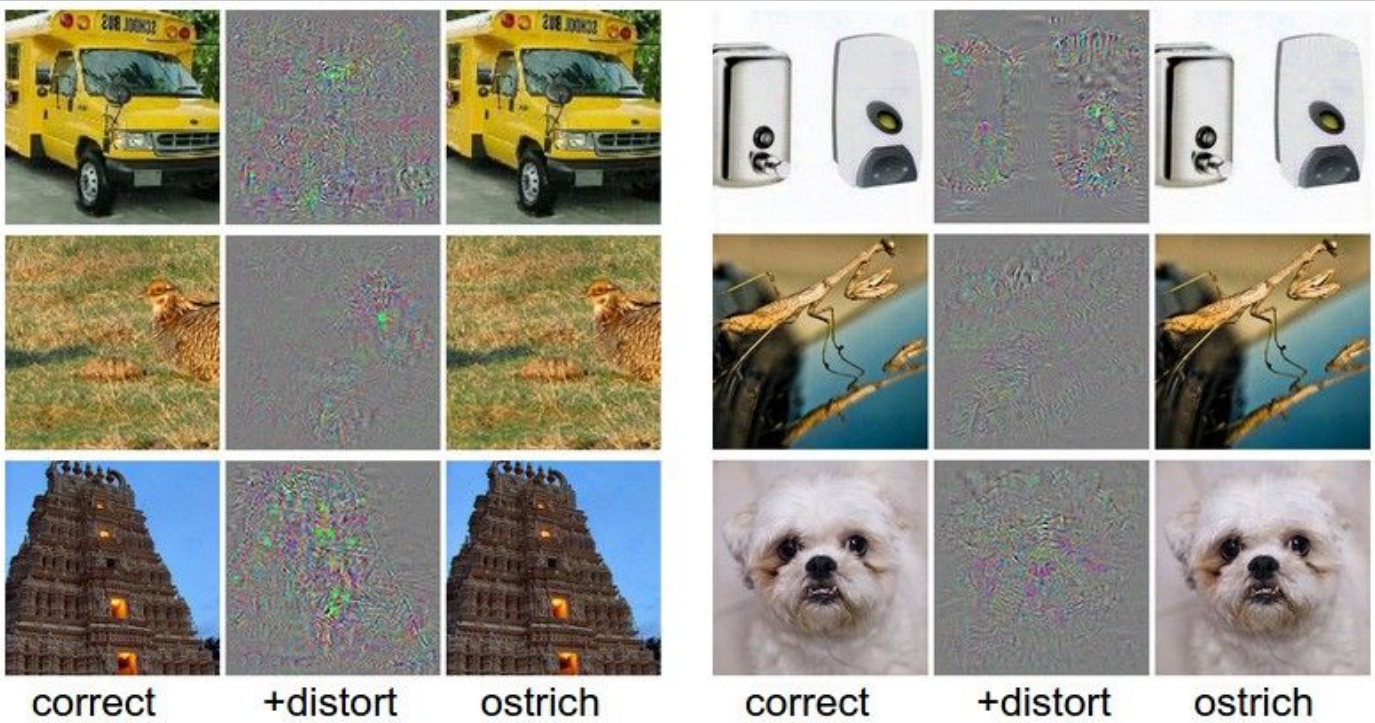
"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Adversarial Attacks

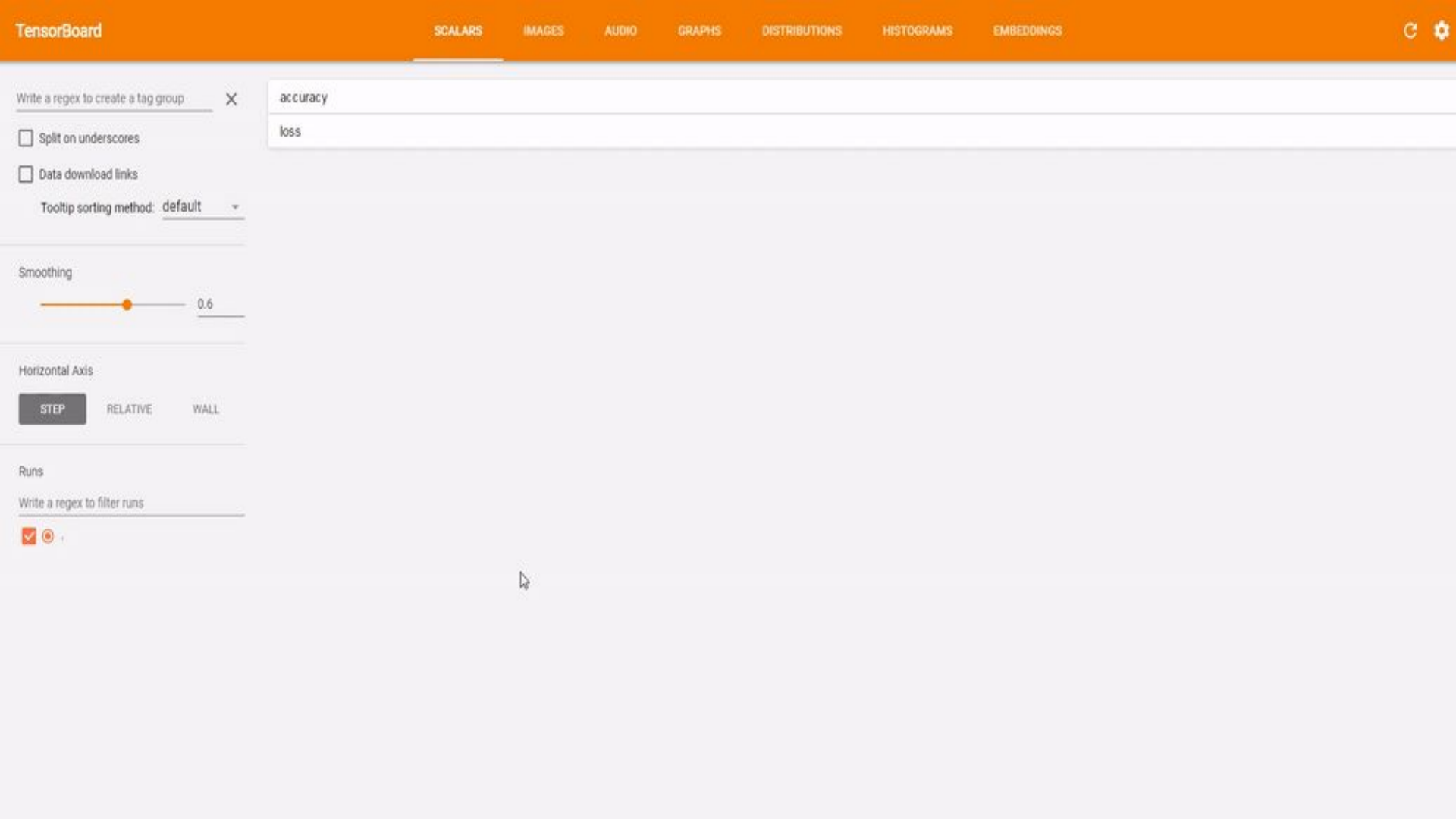


Model Management

Model Management

- Access and permission controls
- Status control for different environments
- Model versioning
- Model monitoring

Visualizations


TensorBoard
SCALARS
IMAGES
AUDIO
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
EMBEDDINGS
Write a regex to create a tag group
Split on underscores
Data download links
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
accuracy
loss

TensorBoard

SCALARS IMAGES AUDIO GRAPHS

Write a regex to create a tag group

Split on underscores

Data download links

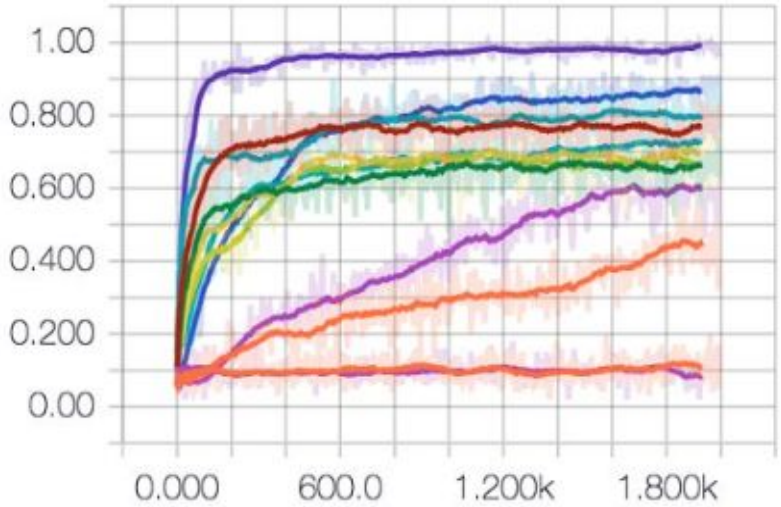
Tooltip sorting method: default

Smoothing

0.6

accuracy

accuracy/accuracy



TensorBoard

EVENTS IMAGES GRAPH HISTOGRAMS

Fit to screen

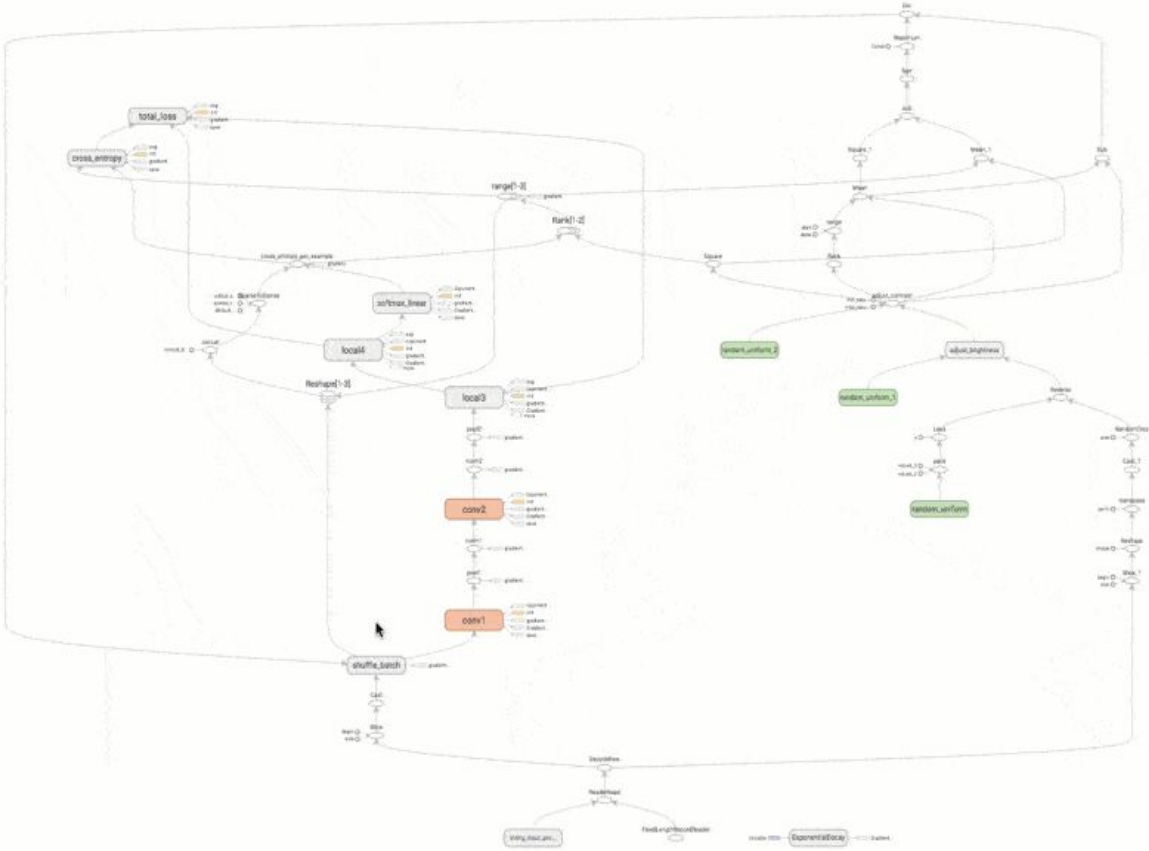
Run cifar-train

Upload Choose File

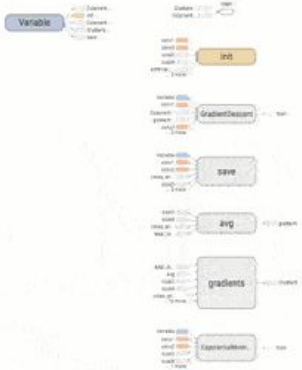
Color Structure

color: same substructure
gray: unique substructure

Main Graph



Auxiliary nodes



Graph (* = expandable)

- Namespace*
- OpNode
- Unconnected series*
- Connected series*
- Constant
- Summary
- Dataflow edge
- Control dependency edge
- Reference edge

DATA

8 tensors found

Mnist with images 10K

Color by

label

0	980
1	1135
2	1032
3	1010
4	982
5	892
6	958
7	1028
8	974

T-SNE PCA CUSTOM

Dimension 2D 3D

Perplexity 25

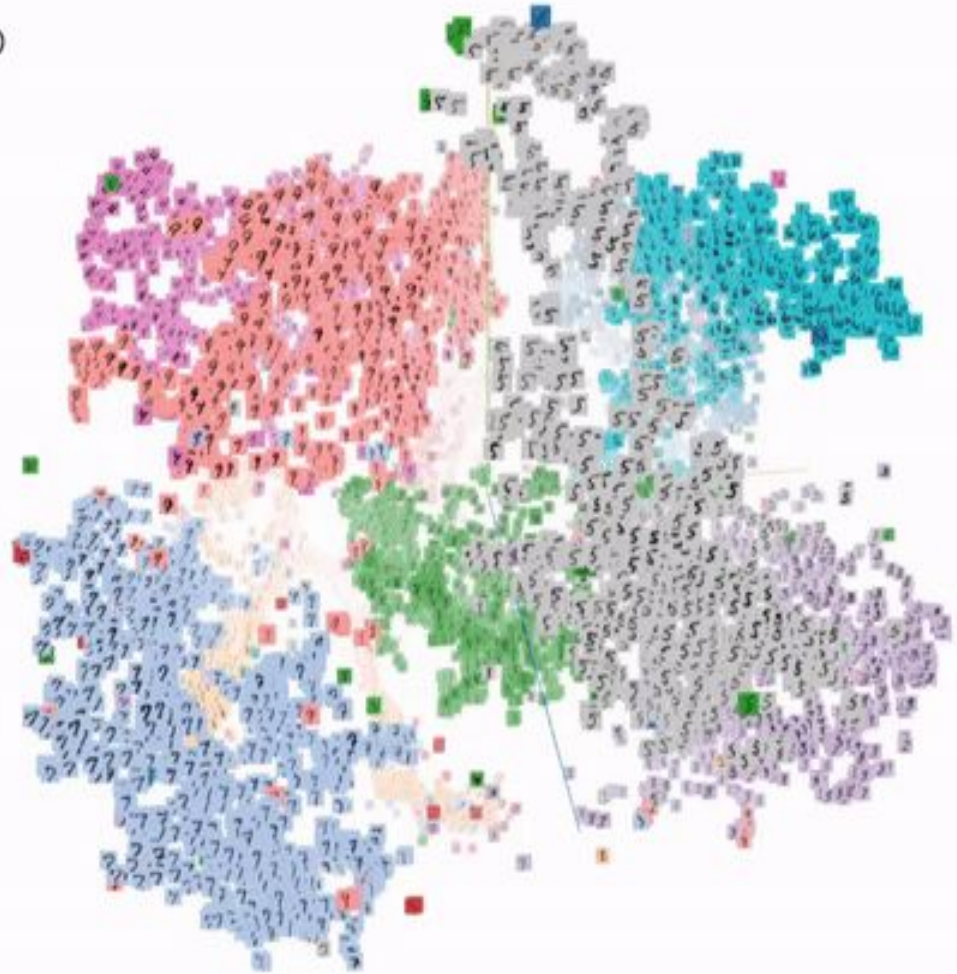
Learning rate 10

Re-run Stop

Iteration: 438

How to use t-SNE effectively.

Points: 10000 | Dimension: 784

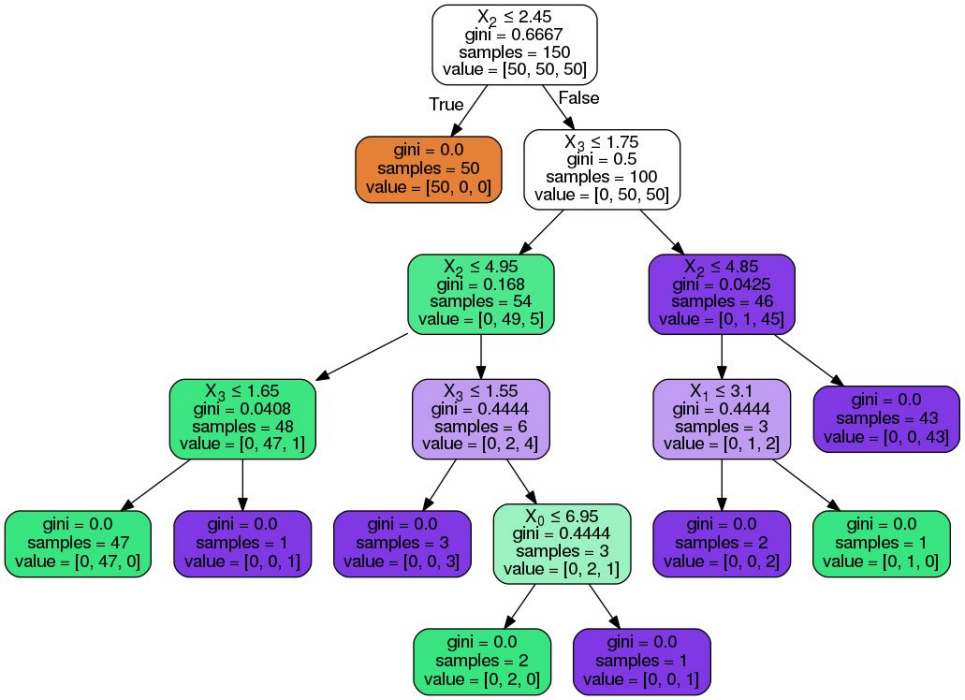


Show All Data | Isolate 101 points | Clear selection

Search by label

BOOKMARKS (0)

Visualization - Decision Trees



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

"Why Should I Trust You?": Explaining the Predictions of Any Classifier

Thank you! Any questions?

@terrytangyuan

Acknowledgement

- Some of the images are adopted from:
  - https://www.tensorflow.org/
  - https://medium.com/tensorflow/
  - Model Deployment Error
  - Breaking Linear Classifier on ImageNet
  - "Why Should I Trust You?": Explaining the Predictions of Any Classifier