TensorFlow Tools and Techniques

Joao Pereira
19354106

Hello!

I am Joao Pereira

I will be presenting TensorFLow Tools and Techniques and all of its wonder’s within this lecture.

Contact me at joao.pereira2@mail.dcu.ie

What is TensorFlow?

First of all what even is TensorFlow?

1

Open-source library for machine learning and artificial intelligence

TensorFlow's primary focus is directed towards training and inference of 'neural networks'. Primarily used for developing machine learning applications.

Neural Network?

- Simplified models of brains used to recognize patterns
- Decision Tree
- Algorithm designed to map a set of inputs to a range of possible outputs.

Form of artificial intelligence consisting of groups of neurons (just like the human brain) used to recognize patterns and clutter by interpreting data and then classifying that information

Example of TensorFlow

Type something into the Google search bar.

Notice that when you're typing, Google supplies you with autofill's on what you could be looking for?

Created by Google Brain

Released in 2015 under Apache License 2.0 in 2015

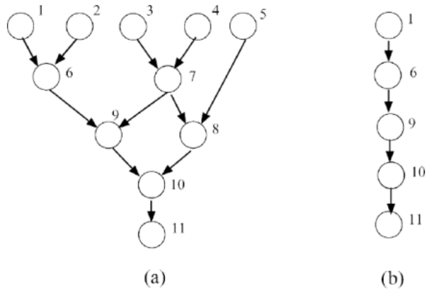
In 2019 it got the full release with the name of TensorFlow

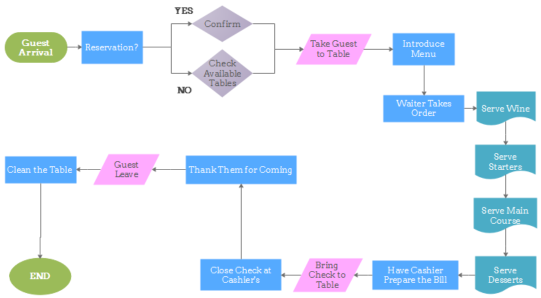
How does TensorFlow work?

2

How it works:

- Developers create Dataflow graphs
- By accepting a multi-dimensional array called Tensor
- Flow chart of operations
- Developed in C++
- Accessed and controlled by Python





Guest Arrival

Reservation?

YES

Confirm

NO

Check Available Tables

Take Guest to Table

Introduce Menu

Waiter Takes Order

Serve Wine

Serve Starters

Serve Main Course

Serve Desserts

Have Cashier Prepare the Bill

Bring Check to Table

Close Check at Cashier's

Thank Them for Coming

Guest Leave

Clean the Table

END

“

> A tensor is a generalization of vectors and matrices and is easily understood as a multidimensional array. A vector is a one-dimensional or first-order tensor and a matrix is a two-dimensional or second-order tensor

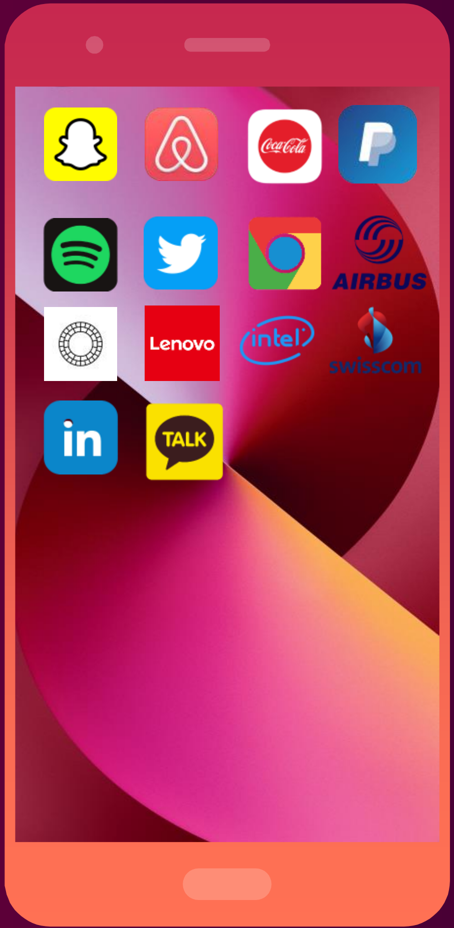
- Introduction To Tensors for Machine Learning

Usage of TensorFlow
Around the world

3

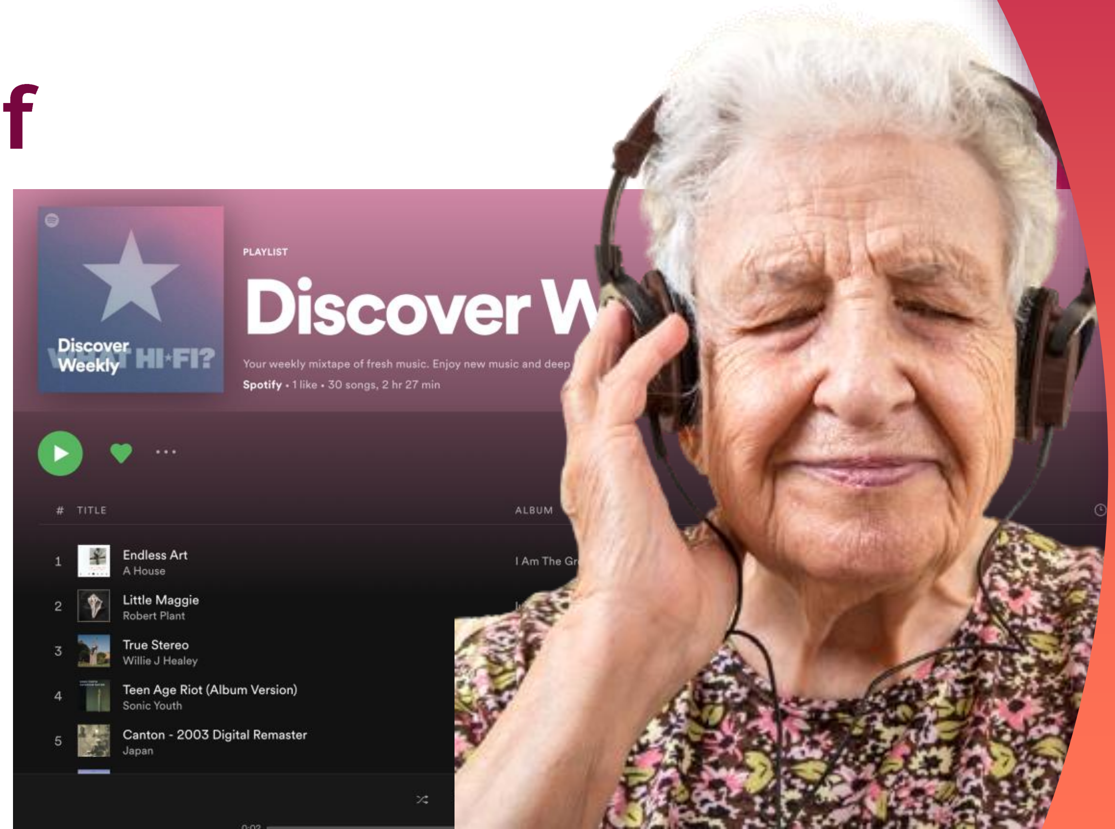
Famous applications currently using TensorFlow

Companies like Spotify, Airbnb, Chrome and PayPal rely a lot on TensorFlow to operate their most popular features



Spotify's use of TensorFlow

One of Spotify's most crucial feature is it's display of created playlists and recommendations for user's to listen to.



PLAYLIST

Discover W

Your weekly mixtape of fresh music. Enjoy new music and deep

Spotify • 1 like • 30 songs, 2 hr 27 min

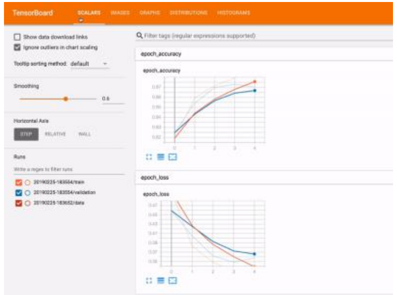
#	TITLE	ALBUM
1	Endless Art A House	I Am The Gr
2	Little Maggie Robert Plant	
3	True Stereo Willie J Healey	
4	Teen Age Riot (Album Version) Sonic Youth	
5	Canton - 2003 Digital Remaster Japan	

TensorFlow Tools

3

List of Tools:

- Colab
- TensorBoard
- What-If Tool
- ML Perf
- XLA
- TensorFlow Playground
- TPU Research Cloud
- MLIR



□ Show data download links
☑ Ignore outliers in chart scaling
Tooltip sorting method: default

Smoothing 0.6

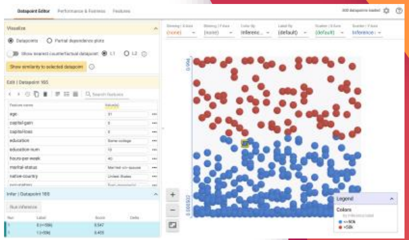
Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ 20190225-183554/train
☑ ○ 20190225-183554/validation
☑ ○ 20190225-183652/data

Filter tags (regular expressions supported)



● Datapoints ○ Partial dependence plots

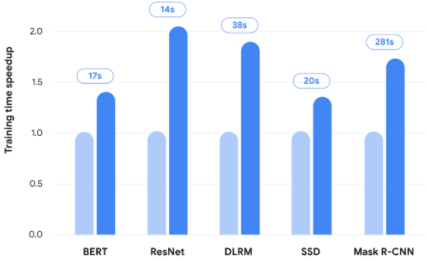
Show nearest counterfactual datapoint ● L1 ○ L2

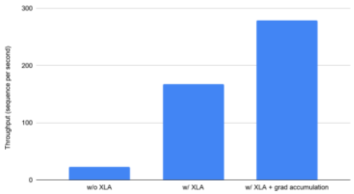
Show similarity to selected datapoint

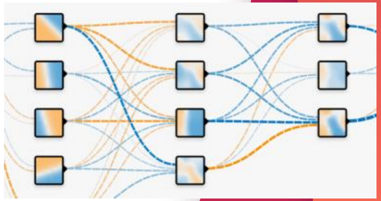
Inference: (none) (none) Inference... (default) (default) Inference

Feature name	Value(s)
age	[illegible]
capital-gain	0
capital-loss	0
education	Some-college
education-num	10
hours-per-week	40
marital-status	Married-civ-spouse
native-country	United-States

Run	Label	Score	Delta
1	0 (<=50k)	0.547	
1	1 (>50k)	0.453	







colab

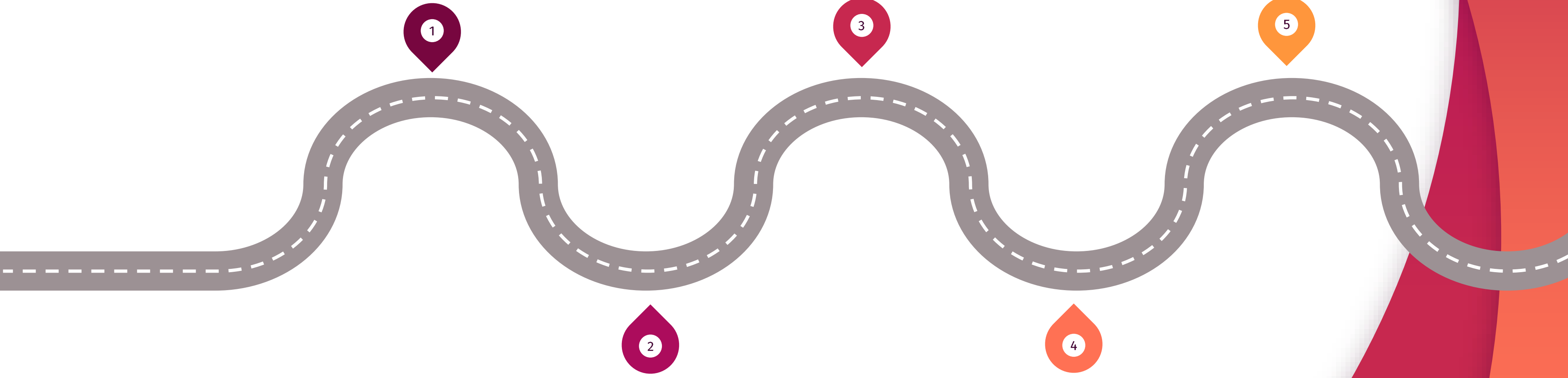
- Sharing
- Free access to GPU’s
- Zero Configuration

Composed of cells, each of which can be consisted of code, images and more

Connected to a cloud-based run time

Filled with code snippets to produce interactive visualizations

colab Steps



1 Create or import a New Notebook

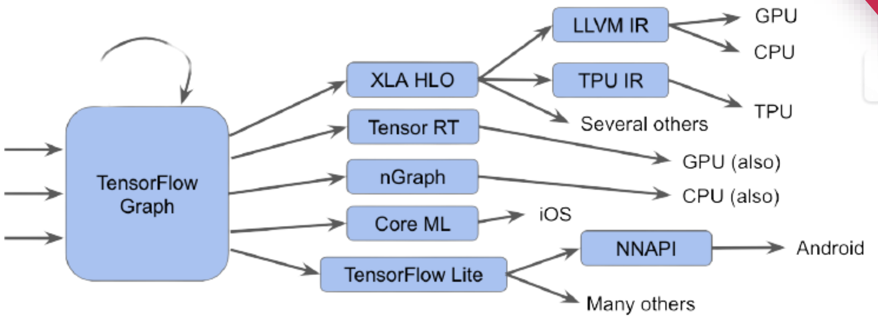
2 Set up your Free GPU

3 Begin coding and testing

4 Mount it to your google drive and import libraries

5 Save and share your code with other users

MLIR

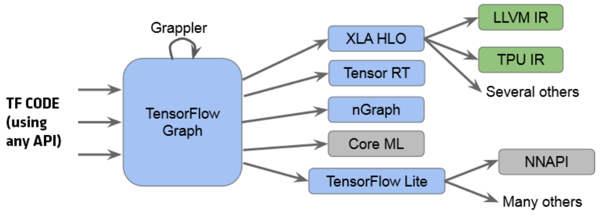


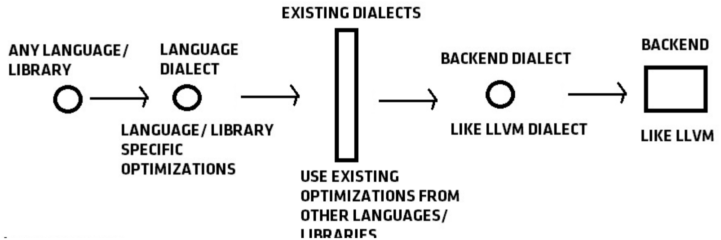
Multi-Level Intermediate Representation is responsible for creating extensible and reusable compiler infrastructure. Intended to be a hybrid IR

- Dataflow Graphs
- Optimizations and transformations on graphs
- High-performance-computing-style loop optimizations across Kernel
- Represent specified target operations
- Quantization and other graph transformations

Intermediate Representation between a language (like Python) or library and the compiler backend (E.g., LLVM).

Demonstration of MLIR





- Start off with a flow chart
- Issue with this flow chart is that developers will have to re-implement all optimizations after each path such as XLA HLO and TensorFlow Lite
- MLIR solves this issue
- Same compiler infrastructure + support for a lot of backends with different IRs with no code re-use
- Increases code reuse so hardware backends are supported
- Captures information with flow chart and provides for better design, reuse of LLVM for code generation and it has better source code tracking

https://iq.opengenus.org/mlir-compiler-infrastructure/

XLA (Accelerated Linear Algebra)

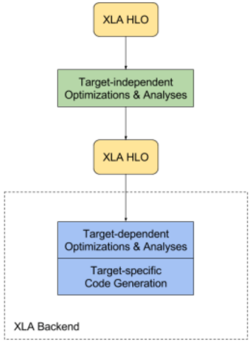
- Domain Specific compiler for linear algebra that speeds up performance within TensorFlow Models
- No source code changes required
- Performance is improvised through speed and memory management
- Performed either by Just-in-Time (JIT) or Ahead-of-Time (AOT)
- JIT – performed to optimize graph at runtime
- AOT used to generate binaries for specific architecture so that runtime + JIT compilation is not necessary

How XLA operates

Run TensorFlow program -> operations executed individually by executor -> each operation has a precompiled GPU Kernel that it must dispatch to

XLA as an alternative method -> compilers graph into computation kernels -> developed specifically for given model -> exploit model-specific information for optimization.

Demonstration of XLA


XLA HLO
Target-independent Optimizations & Analyses
XLA HLO
Target-dependent Optimizations & Analyses
Target-specific Code Generation
XLA Backend

Above is the XLA Compilation process:

- XLA is run through target-independent optimizations and analyses
- Sends HLO to backend
- Backend performs once again target-independent optimizations and analyses
- Passed through target-specific code generation

```
def model_fn(x, y, z):
 return tf.reduce_sum(x + y * z)
```

Ran through three separate kernels:

- Addition
- Multiplication
- Reduction

Whilst if to use XLA it would run by one kernel

Does this by “fusion”

- Fusion – Avoids extra cycles therefore reduces memory usage and improves performance speeds.

Thank you for Listening

Hope you enjoyed, have a great day!

If any issues are concerned, please feel free to contact me at joao.pereira2@mail.dcu.ie