



C14200: FROM ZERO TO
ONE - DEEP LEARNING
WITH PYTORCH

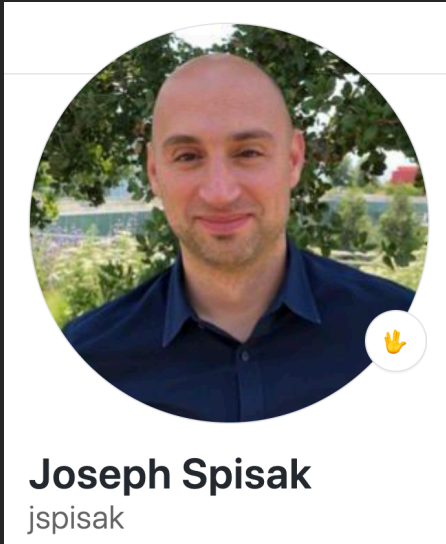
JOE SPISAK
PRODUCT MANAGER

FRANCISCO MASSA
RESEARCH ENGINEER





WHO AM I?



CURRENT: PRODUCT LEAD - PYTORCH

PREVIOUS:

- AI PRODUCT & PARTNERSHIPS LEAD
@AMAZON

- DIR OF ML STRATEGY @INTEL

*PASSIONATE ABOUT THE INTERSECTION
OF OSS, AI AND COMMUNITY BUILDING*



PYTORCH HUB FOR RESEARCHERS

Explore and extend models from the latest cutting edge research.

- All
- Audio
- Generative
- Nlp
- Scriptable
- Vision



Sort 



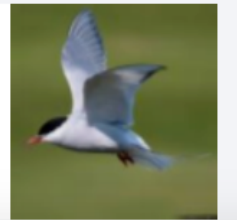
MiDaS  456

The MiDaS v2 model for computing relative depth from a single image.



ntsnet  7

classify birds using this fine-grained image classifier

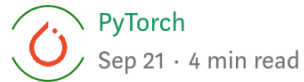




INDUSTRY USAGE

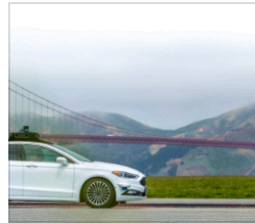
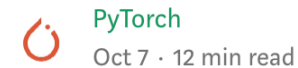
PyTorch and Mars Petcare: Formulas, photos & feces

At Mars Petcare, we are inspired every day by our Purpose: A Better World for Pets.



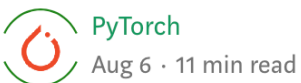
How Lyft Uses PyTorch to Power Machine Learning for Their Self-Driving Cars

Reducing the median job training time for heavy production jobs such as 2D and 3D detectors and segmenters to just 1 hour



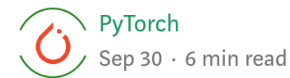
AI for AG: Production machine learning for agriculture

Blue River Technology builds an intelligent sprayer that leverages PyTorch to target weeds without harming crops.



Speeding up drug discovery with advanced machine learning

AstraZeneca Biological Insights Knowledge Graph (BIKG) group uses PyTorch and Microsoft Azure Machine Learning to speed up drug discovery



<https://medium.com/pytorch>







CV

NLP

SPEECH



RL

Semantic Segmentation

 Semantic Segmentation 66 benchmarks 1295 papers with code	 Tumor Segmentation 1 benchmark 58 papers with code
--	---



[▶ See all 19 tasks](#)

Machine Translation



 Machine Translation 49 benchmarks 901 papers with code	 Transliteration 22 papers with code
---	--

[▶ See all 7 tasks](#)

Speech Synthesis



 Speech Synthesis 2 benchmarks 72 papers with code	 Expressive Speech Synthesis 3 papers with code
--	--

Video Games

 Atari Games 62 benchmarks 155 papers with code	 Starcraft II 2 benchmarks 21 papers with code
---	--



[▶ See all 9 tasks](#)

Image Classification



 Image Classification 158 benchmarks 1129 papers with code	 Few-Shot Image Classification 48 benchmarks 67 papers with code
--	--

[▶ See all 19 tasks](#)



Language Modelling

 Language Modelling 15 benchmarks 857 papers with code	 Sentence Pair Modeling 5 benchmarks 5 papers with code
--	--



Speech Enhancement

 Speech Enhancement 9 benchmarks 57 papers with code	 Speech Dereverberation 2 benchmarks 5 papers with code
--	---



Atari Games

 Atari Games 62 benchmarks 155 papers with code	 Montezuma's Revenge 15 papers with code
---	---



Object Detection

 Object Detection 140 benchmarks 929 papers with code	 3D Object Detection 35 benchmarks 88 papers with code
---	--



Question Answering

 Question Answering 59 benchmarks 751 papers with code	 Open-Domain Question Answering 5 benchmarks 41 papers with code
--	---

Speaker Verification

 Speaker Verification 40 papers with code	 Text-Independent Speaker Verification 7 papers with code
--	--

Continuous Control

 Continuous Control 52 benchmarks 153 papers with code	 Car Racing 10 papers with code
--	--

Object Navigation



"Go to toilet"





WHAT IS PYTORCH?



<https://github.com/pytorch/pytorch>

pytorch / pytorch

Watch 1.4k Star 38.7k Fork 9.9k

Code Issues 4,338 Pull requests 1,456 Actions Projects 5 Wiki Security 0 Insights

Tensors and Dynamic neural networks in Python with strong GPU acceleration <https://pytorch.org>

neural-network autograd gpu numpy deep-learning tensor python machine-learning

26,759 commits 3,791 branches 0 packages 38 releases 1,399 contributors View license

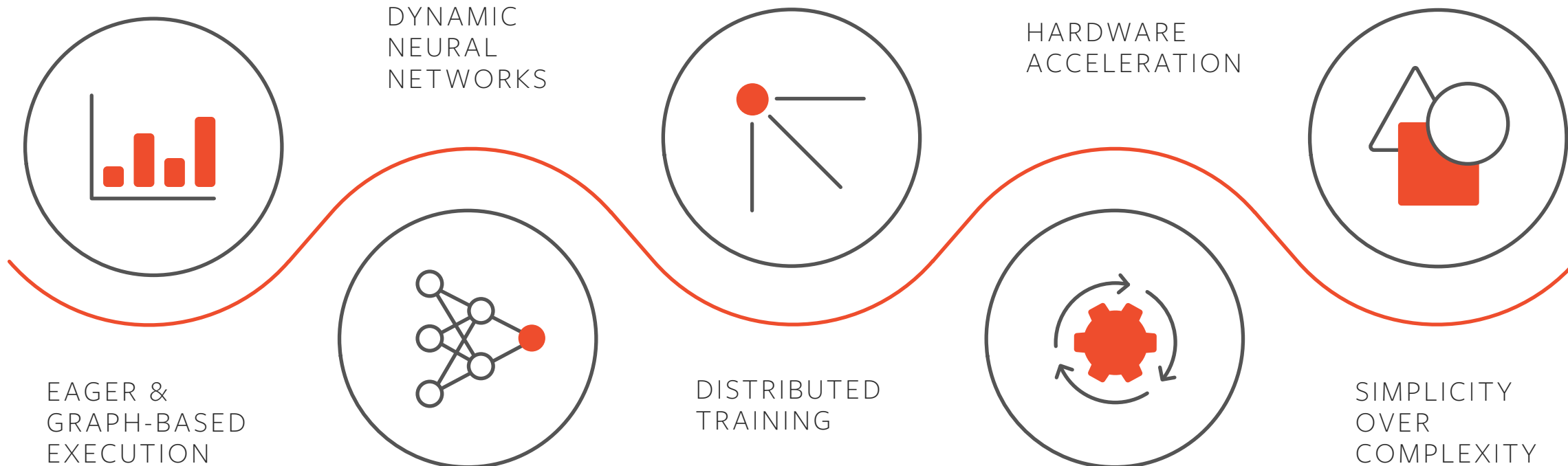
Branch: master New pull request Find file Clone or download

Nirav Mehta and facebook-github-bot Adding support for manifold files in DBReader (#37727) Latest commit acacad2 1 hour ago

.circleci	Revert D21585458: [pytorch][PR] [RELAND] .circleci: Improve docker im...	14 hours ago
.ctags.d	Add a .ctags.d/ toplevel directory (#18827)	14 months ago
.github	Revert D21585458: [pytorch][PR] [RELAND] .circleci: Improve docker im...	14 hours ago



WHAT IS PYTORCH?



EAGER &
GRAPH-BASED
EXECUTION

DYNAMIC
NEURAL
NETWORKS

DISTRIBUTED
TRAINING

HARDWARE
ACCELERATION

SIMPLICITY
OVER
COMPLEXITY



PYTORCH

RESEARCH
PROTOTYPING

+

PRODUCTION
DEPLOYMENT



Alfredo Canziani
@alfcnz

Atcold/pytorch-Deep-Learning-Minicourse

ing with PyTorch. Contribute to
rning-Minicourse development
GitHub.



Smerity
@smerity

True to their mission, the **@PyTorch** community focused
on solving the issues of eager mode w/o impacting

erability?
Want
ll without

2 Oct 2018

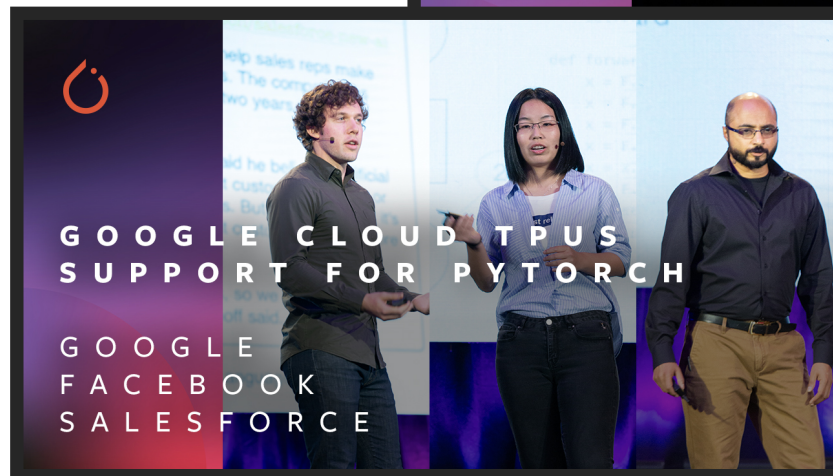
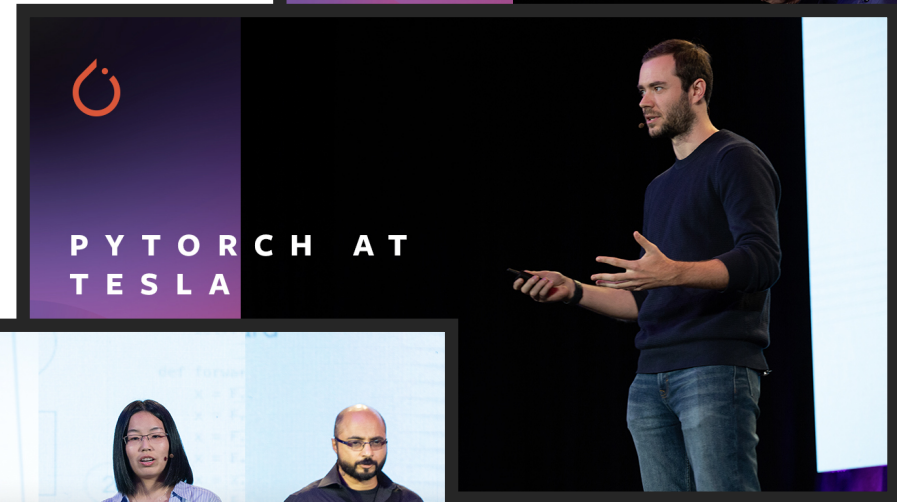


Jeremy Howard
@jeremyphoward

At the **@PyTorch** developer conference,
I was part of a fascinating panel with
@clattner_llvm, Yangqing Jia, and Noah
Goodman, Expertly moderated by
@soumithchintala. Here it is!



10 Oct 2018



GOOGLE CLOUD TPUS
SUPPORT FOR PYTORCH

GOOGLE
FACEBOOK
SALESFORCE



~1,619

CONTRIBUTORS

50%+

YOY GROWTH

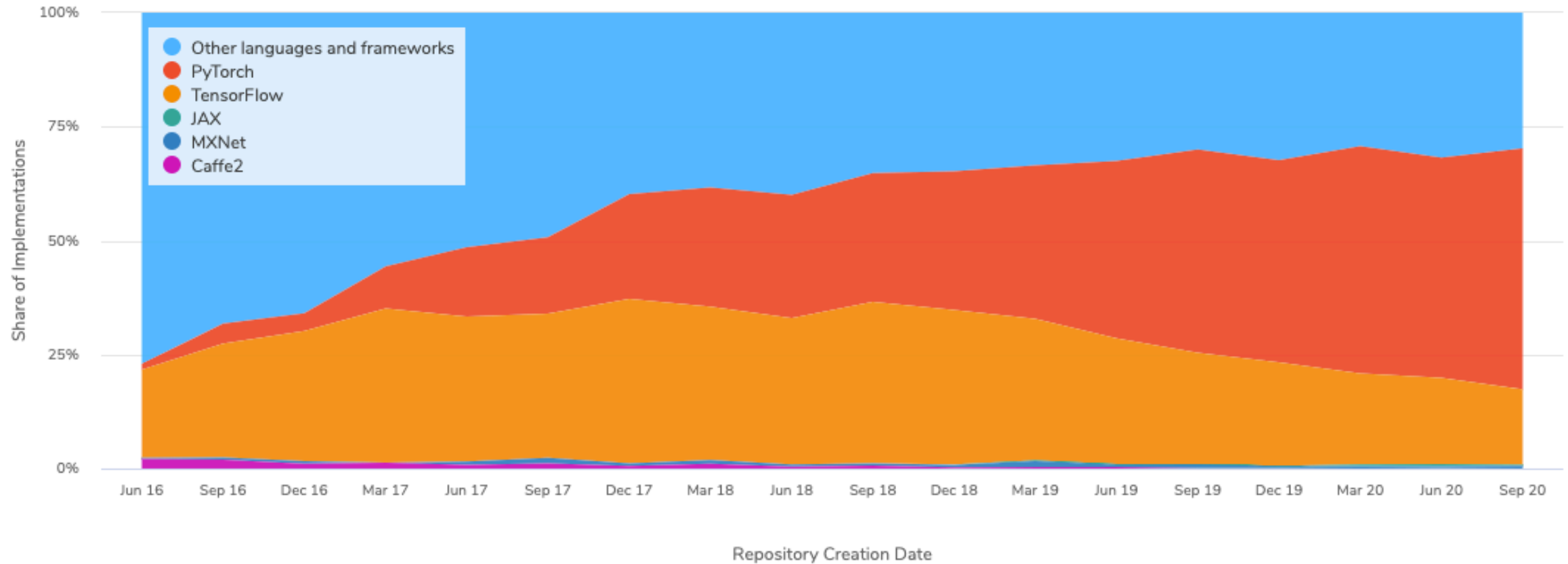
34K+

PYTORCH FORUM USERS



GROWING USAGE IN OPEN SOURCE

Paper Implementations grouped by framework



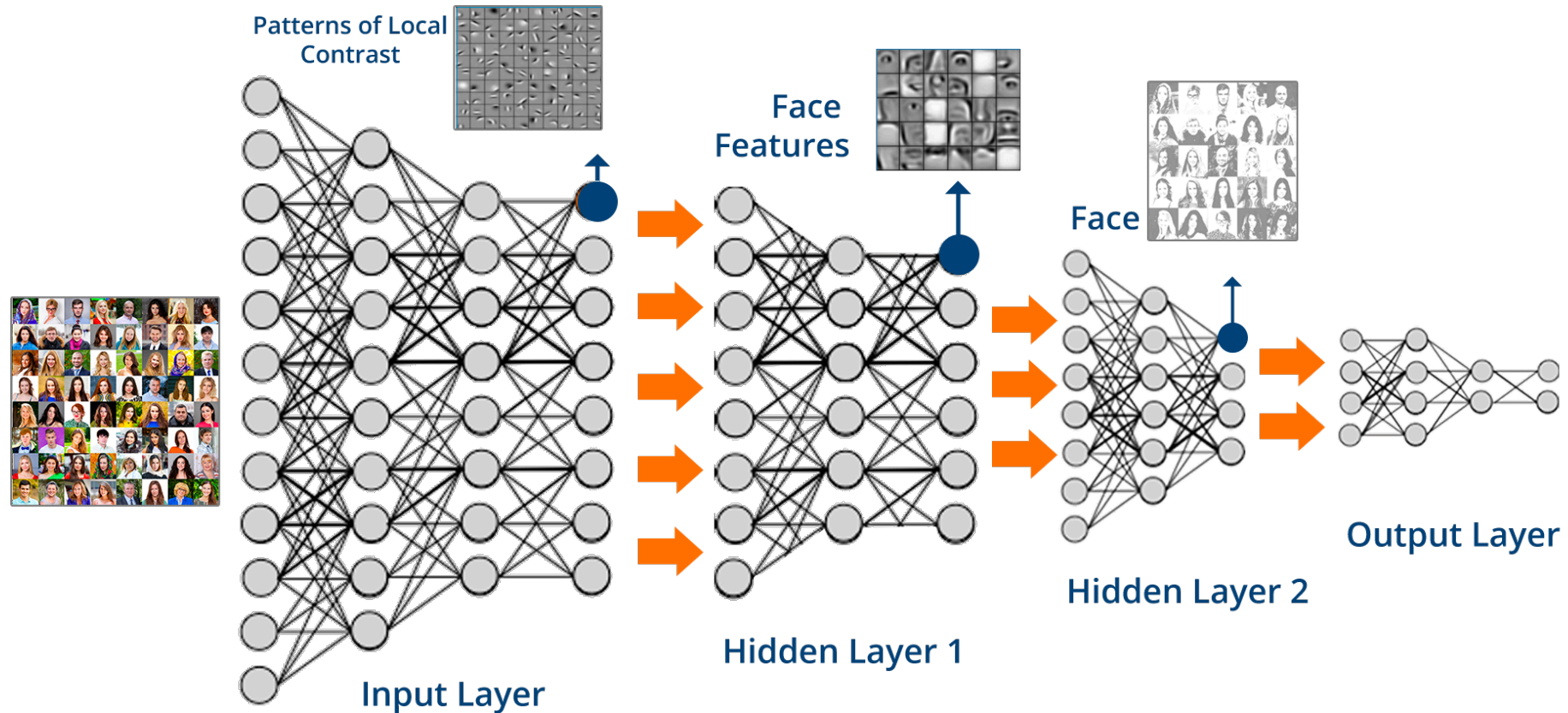


WHAT IS DEEP LEARNING?



WHAT IS DEEP LEARNING?

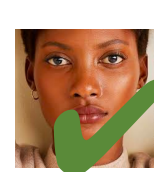
Deep learning algorithms attempt to draw similar conclusions as humans would by continually analyzing data with a given logical structure. To achieve this, deep learning uses a multi-layered structure of algorithms called neural networks.



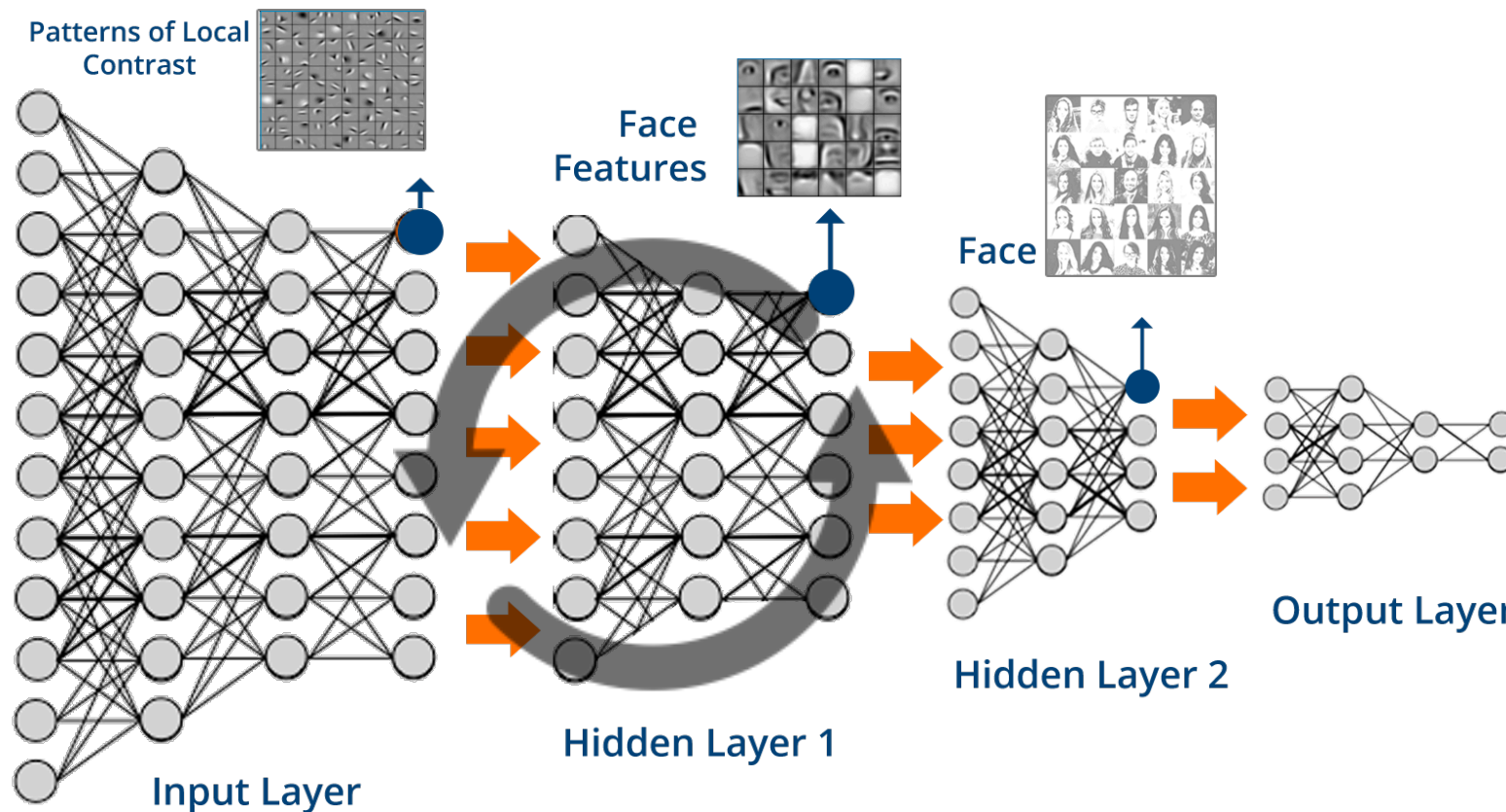


HOW DO NEURAL NETWORKS LEARN? BACKPROPOGATION

Training Data: Known correct examples



Input: Images



Loss: How are we doing?

Output: New Prediction



SOFTWARE 2.0 | PROGRAMMING WITH DATA

SOFTWARE 1.0

Explicit programming

1. It consists of explicit instructions to the computer written by a programmer.
2. By writing each line of code, the programmer identifies a specific point in program space with some desirable behavior.
3. Written directly in C++, Python,..

SOFTWARE 2.0

Optimization based - learned from data

1. Can be written in much more abstract, human unfriendly language, such as the weights of a neural network.
2. No human is involved in writing this code because there are a lot of weights (typical networks might have millions), and coding directly in weights is basically impossible
3. Weights are instead learned from iterating on data to learn a function - in a process known as training



JUPYTER + PYTORCH = ❤️



PYTORCH HAS NATIVE SUPPORT FOR COLAB

The screenshot shows the Google Colaboratory interface. At the top, there's a header with the Colab logo, the text "Welcome To Colaboratory", and a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side of the header, there are "Share", "Settings", and a user profile icon with the letter "J".

Below the header, there's a sidebar on the left with a "Table of contents" section. It lists several items: "Getting started", "Data science", "Machine learning", "More Resources", "Machine Learning Examples", and a "Section" with a plus icon.

The main content area is titled "What is Colaboratory?". It starts with the Colab logo and the heading "What is Colaboratory?". Below this, there's a paragraph: "Colaboratory, or 'Colab' for short, allows you to write and execute Python in your browser, with". This is followed by a bulleted list:

- Zero configuration required
- Free access to GPUs
- Easy sharing

Next is another paragraph: "Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!".

Below this is a section header "Getting started" with a downward arrow. The text under this section says: "The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code."

Then it says: "For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:"

The code cell contains the following Python code:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

Below the code, there's a small blue circle icon and the number "86400".

At the bottom, there's a paragraph: "To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut 'Command/Ctrl+Enter'. To edit the code, just click the cell and start editing."



OPTION 1: LOADING A NB FROM GITHUB

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

Examples Recent Google Drive **GitHub** Upload

Enter a GitHub URL or search by organization or user Include private repos

https://github.com/pytorch/workshops/blob/master/CS7643/Module1/tensorboard_with_pytorch.ipynb 🔍

Repository: Branch:

Path

CS7643/Module1/tensorboard_with_pytorch.ipynb

NEW NOTEBOOK CANCEL

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.



OPTION 2: CLICK THROUGH ON PYTORCH.ORG

The screenshot shows the PyTorch website interface. At the top, the PyTorch logo is on the left, and navigation links for 'Get Started', 'Ecosystem', 'Mobile', 'Blog', 'Tutorials', 'Docs', 'Resources', and 'Github' are on the right. The 'Tutorials' link is highlighted with a red dot. Below the navigation, the breadcrumb path 'Tutorials > (beta) Dynamic Quantization on an LSTM Word Language Model' is shown. A search bar is present on the left. The main content area features three buttons: 'Run in Google Colab' (highlighted with a red box), 'Download Notebook', and 'View on GitHub'. The title of the tutorial is '(BETA) DYNAMIC QUANTIZATION ON AN LSTM WORD LANGUAGE MODEL'. Below the title, the author is listed as James Reed and the editor as Seth Weidman. The 'Introduction' section begins with a paragraph explaining quantization and its benefits. A code block at the bottom shows the start of the Python code with imports for 'os', 'io', and 'time'.

1.5.1

Search Tutorials

PyTorch Recipes

See All Recipes

Learning PyTorch

Deep Learning with PyTorch: A 60 Minute Blitz

Learning PyTorch with Examples

What is *torch.nn* really?

Visualizing Models, Data, and Training with TensorBoard

Image/Video

TorchVision Object Detection Finetuning Tutorial

Transfer Learning for Computer Vision Tutorial

Adversarial Example Generation

DCGAN Tutorial

Audio

Tutorials > (beta) Dynamic Quantization on an LSTM Word Language Model

Shortcuts

Run in Google Colab

Download Notebook

View on GitHub

(BETA) DYNAMIC QUANTIZATION ON AN LSTM WORD LANGUAGE MODEL

Author: James Reed

Edited by: Seth Weidman

Introduction

Quantization involves converting the weights and activations of your model from float to int, which can result in smaller model size and faster inference with only a small hit to accuracy.

In this tutorial, we'll apply the easiest form of quantization - **dynamic quantization** - to an LSTM-based next word-prediction model, closely following the **word language model** from the PyTorch examples.

```
# imports
import os
from io import open
import time
```



PYTORCH ON JUPYTER VIA COLAB

The screenshot shows a Google Colab notebook interface. At the top, the notebook title is "tensorboard_with_pytorch.ipynb". Below the title is a menu bar with options: File, Edit, View, Insert, Runtime, Tools, Help. On the right side of the menu bar, there are icons for "Share", "Settings", and a user profile icon labeled "J". Below the menu bar, there are tabs for "+ Code" and "+ Text", and a "Copy to Drive" button. On the right side of this bar, there are "Connect", "Editing", and a scroll-up arrow. The main content area shows a code cell with the command `%matplotlib inline`. Below the code cell, there is a section titled "How to use TensorBoard with PyTorch" with a downward arrow. The text in this section describes TensorBoard as a visualization toolkit for machine learning experimentation, allowing tracking and visualizing metrics such as loss and accuracy, visualizing the model graph, viewing histograms, displaying images and much more. It states that in this tutorial, they will cover TensorBoard installation, basic usage with PyTorch, and how to visualize data logged in TensorBoard UI. Below this text is a sub-section titled "Installation". The text in this section states that PyTorch should be installed to log models and metrics into TensorBoard log directory. It provides the following command to install PyTorch 1.4+ via Anaconda (recommended):

```
::  
$ conda install pytorch torchvision -c pytorch
```

It also provides an alternative command using pip:

```
::  
$ pip install torch torchvision
```

Below the installation section is another section titled "Using TensorBoard in PyTorch" with a downward arrow. The text in this section starts with "Let's now try using TensorBoard with PyTorch! Before logging anything, we need to create a `SummaryWriter` instance."



CHANGE YOUR RUNTIME TO GPU OR TPU.. :),
SHIFT-ENTER AND YOU'RE OFF!!

The screenshot shows the Google Colab interface for a notebook titled 'tensorboard_with_pytorch.ipynb'. The top menu includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu, there are options for '+ Code', '+ Text', and 'Copy to Drive'. The main content area displays a code cell with the command `%matplotlib inline` and a section titled 'How to use TensorBoard with PyTorch'. A 'Notebook settings' dialog box is open in the foreground, showing the 'Hardware accelerator' dropdown menu with 'None' selected, 'GPU' highlighted, and 'TPU' as an option. The dialog also includes a help icon, 'CANCEL', and 'SAVE' buttons.

tensorboard_with_pytorch.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

Connect Editing

`%matplotlib inline`

How to use TensorBoard with PyTorch

TensorBoard is a visualization toolkit for machine learning experiments, providing a visual interface for monitoring training metrics such as loss and accuracy, visualizing the model graph, viewing histograms, displaying scalar values, and more. This guide covers TensorBoard installation, basic usage with PyTorch, and how to visualize training metrics.

Installation

PyTorch should be installed to log models and metrics into TensorBoard. Anaconda (recommended):

```
::  
$ conda install pytorch torchvision -c pytorch
```

or pip

```
::  
$ pip install torch torchvision
```

Using TensorBoard in PyTorch

Let's now try using TensorBoard with PyTorch! Before logging anything, we need to create a `SummaryWriter` instance.



SHALL WE LOOK AT SOME CODE?
(IN A JUPYTER NB OF COURSE.. :)