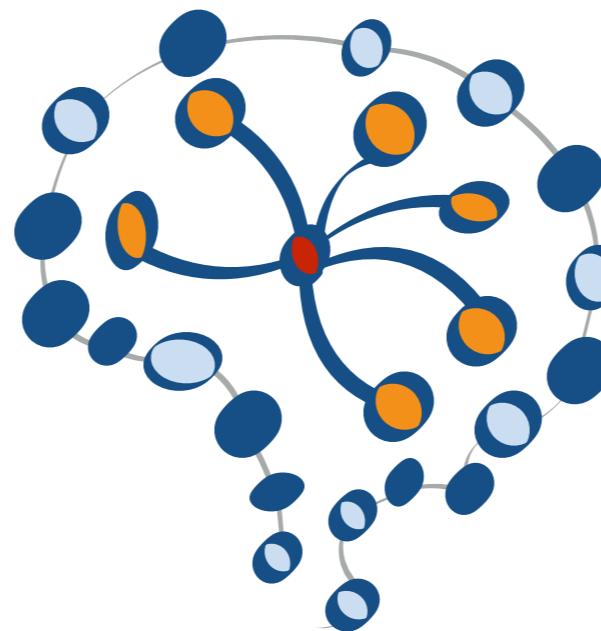


STAT 453: Introduction to Deep Learning and Generative Models

Sebastian Raschka

<http://stat.wisc.edu/~sraschka>



Deep Learning & AI News #9

Interesting Things Related to Deep Learning

Mar 27th, 2021

THE ROBOT BRAINS PODCAST

with Pieter Abbeel

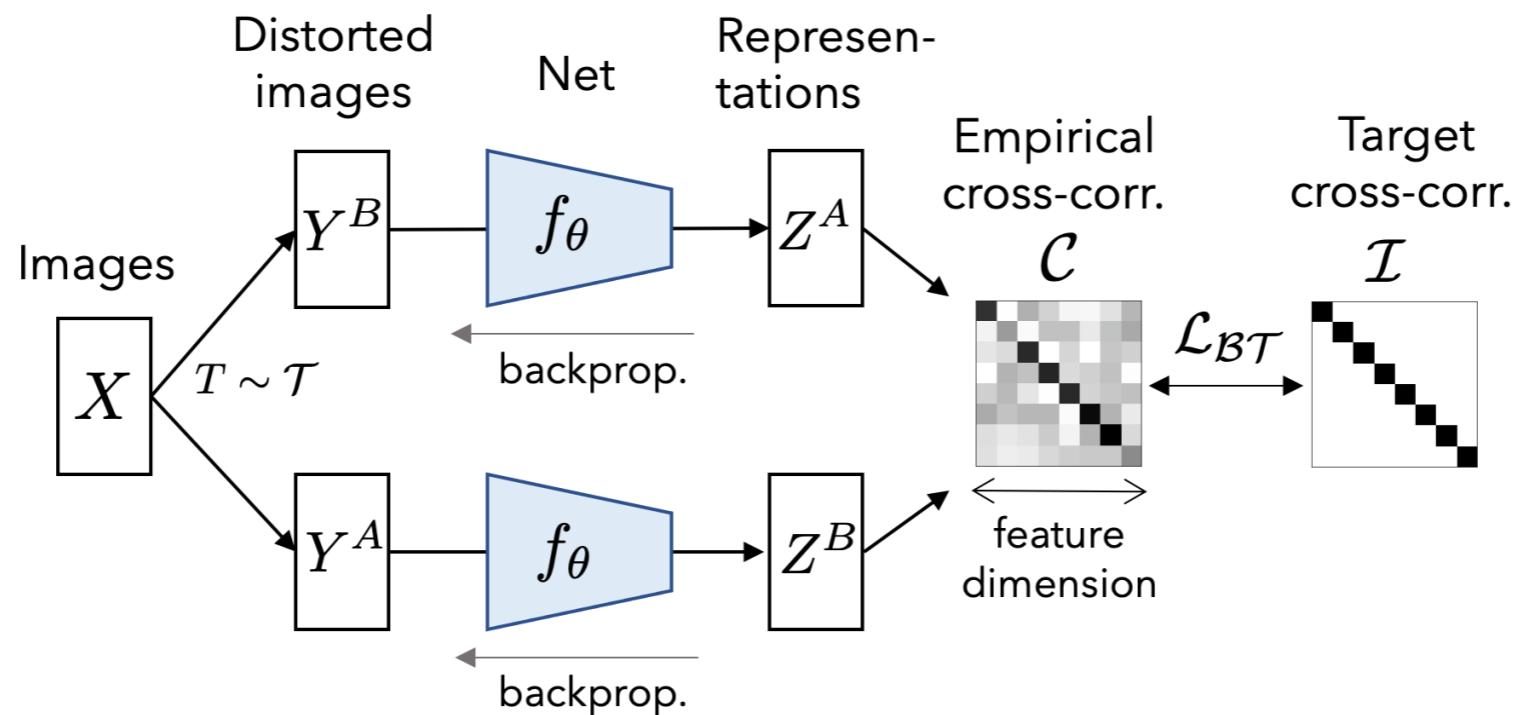


<https://www.therobotbrains.ai>

Barlow Twins: Self-Supervised Learning via Redundancy Reduction

Jure Zbontar ^{* 1} Li Jing ^{* 1} Ishan Misra ¹ Yann LeCun ^{1 2} Stéphane Deny ¹

<https://arxiv.org/pdf/2103.03230.pdf>



1. Run original and distorted image through same network
2. Compute correlation matrix
3. Add objective to make the correlation matrix close to identity matrix



Forces representation vectors of similar samples to be similar

Barlow Twins: Self-Supervised Learning via Redundancy Reduction

Jure Zbontar ^{* 1} Li Jing ^{* 1} Ishan Misra ¹ Yann LeCun ^{1 2} Stéphane Deny ¹

<https://arxiv.org/pdf/2103.03230.pdf>

Algorithm 1 PyTorch-style pseudocode for Barlow Twins.

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the representation
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute representations
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

Table 1. Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet. All models use a ResNet-50 encoder. Top-3 best self-supervised methods are underlined.

Method	Top-1	Top-5
Supervised	76.5	
MoCo	60.6	
PIRL	63.6	-
SIMCLR	69.3	89.0
MoCo v2	71.1	90.1
SIMSIAM	71.3	-
SWAV	71.8	-
BYOL	<u>74.3</u>	91.6
SWAV (w/ multi-crop)	<u>75.3</u>	-
BARLOW TWINS (ours)	<u>73.2</u>	91.0

Shared Task on Hateful Memes at WOAH 2021

Source: <https://www.workshopononlineabuse.com/cfp/shared-task-on-hateful-memes>

Task A (multi-label): For each meme, detect the protected category. Protected categories are: *race, disability, religion, nationality, sex*. If the meme is *not_hateful* the protected category is: *pc_empty*.

Task B (multi-label): For each meme, detect the attack type. Attack types are: Attack types are: *contempt, mocking, inferiority, slurs, exclusion, dehumanizing, inciting_violence*. If the meme is *not_hateful* the protected category is: *attack_empty*.

Important dates

- March 19th: Shared task data is available.
- March 25th: MMF setup for getting started, with initial baselines and pre-trained models released
- May 19th 23:59 (AOE): Predictions due
- May 31, 23:59 (AOE): Shared task paper submissions due
- June 14, 23:59 (AOE): Camera-ready papers due
- August 5th - 6th: Workshop day!



[Submitted on 22 Feb 2021]

Towards Causal Representation Learning

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, Yoshua Bengio

<https://arxiv.org/abs/2102.11107>

Don't have to drive a car off a cliff to learn what happens

- Deep learning is currently largely based on statistical correlations from i.i.d. data
- Learning causal relationships can make models more robust to unexpected situations
- Can make training cheaper -- fewer examples like objects from different angles required
- Enable transfer learning beyond fine-tuning

The challenges:

Does the data reveal causal relationships?

How do we infer abstract causal variables?

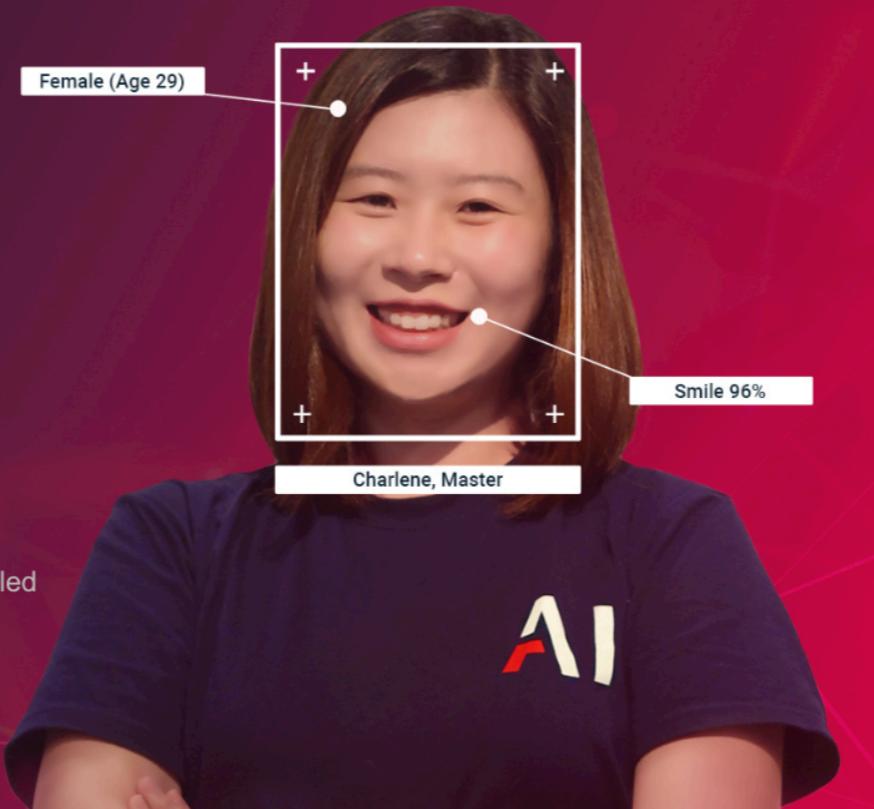
Star 58

"REIMAGINE THE FUTURE WITH DATA ANNOTATION"

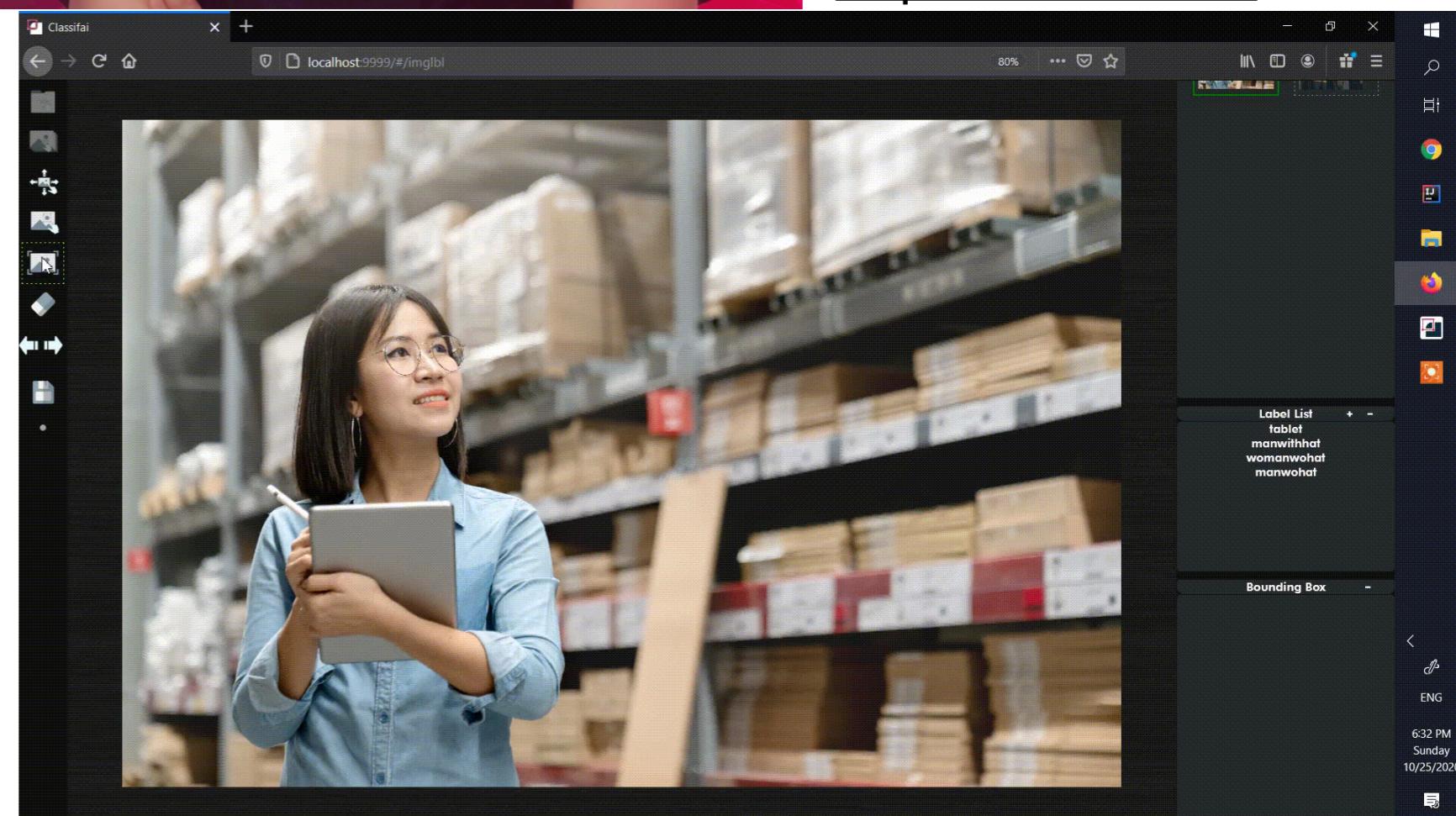
Classifai is one the most comprehensive open-source data annotation platform.

It supports the labelling of various data types with multi labelled outputs forms for AI model training.

[Get Started](#)



<https://classifai.ai>





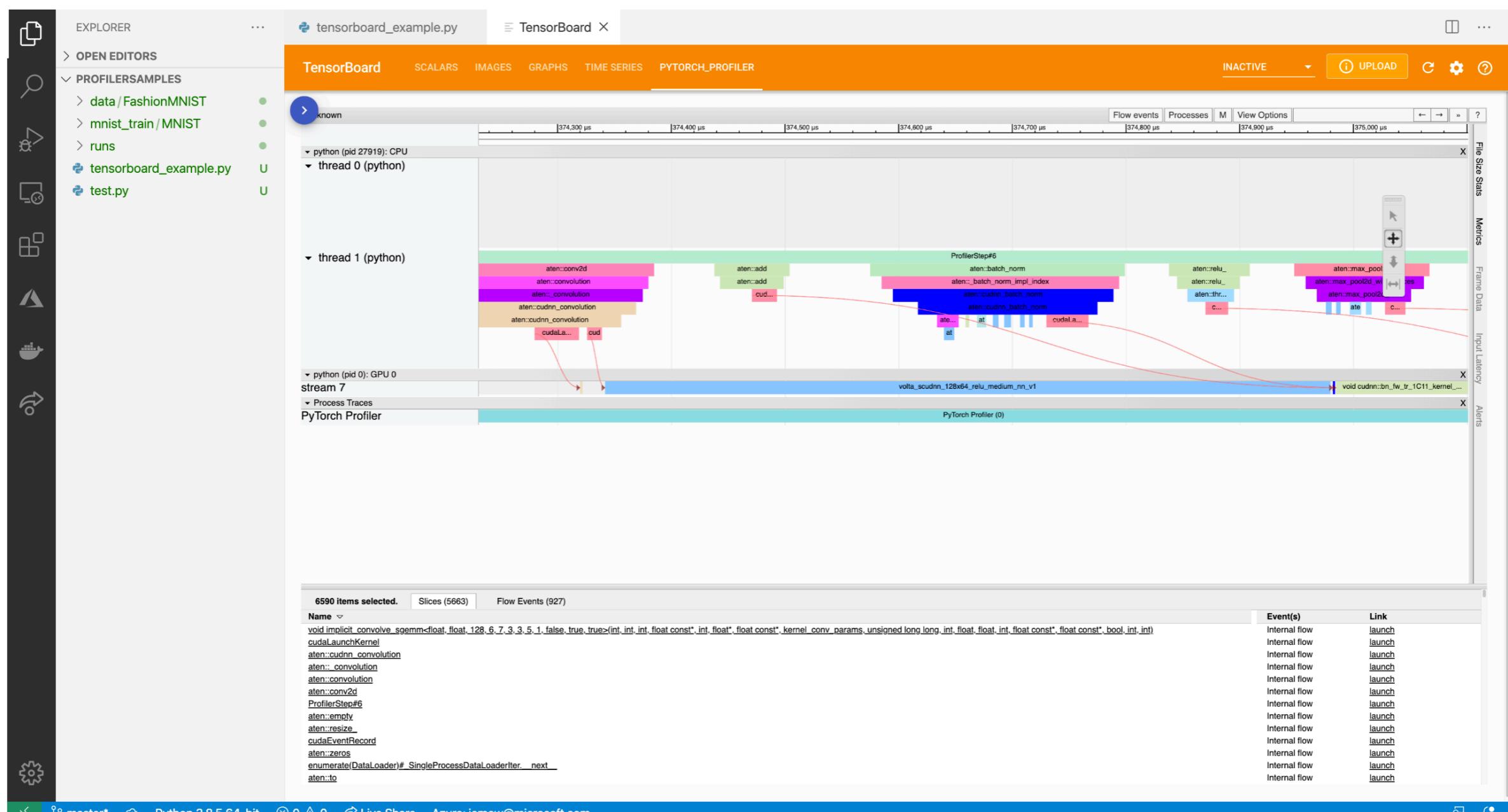
```
valid_mae, valid_mse = compute_mae_and_mse(model, valid_loader,
                                             device=device,
                                             which_model=which_model)
train_mae, train_mse = compute_mae_and_mse(model, train_loader,
                                             device=device,
                                             which_model=which_model)
train_rmse = torch.sqrt(train_mse)
valid_rmse = torch.sqrt(valid_mse)

valid_acc = compute_accuracy(model, valid_loader,
                             device=device,
                             which_model=which_model)
train_acc = compute_accuracy(model, train_loader,
                             device=device,
                             which_model=which_model)
```

March 25, 2021

Introducing PyTorch Profiler - the new and improved performance tool

<https://pytorch.org/blog/introducing-pytorch-profiler-the-new-and-improved-performance-tool/>



VisualStudio Code TensorBoard Integration

<https://devblogs.microsoft.com/python/python-in-visual-studio-code-february-2021-release/>

The screenshot illustrates the integration of TensorBoard within Visual Studio Code. On the left, the file structure shows `test.py` and `tensorboard_example.py`. The `tensorboard_example.py` file contains Python code for a PyTorch model, including imports for `matplotlib.pyplot`, `numpy`, `torch`, `torchvision`, `transforms`, `nn`, `functional`, and `optim`. It also includes code for setting environment variables, defining transforms, datasets (trainset and testset), and dataloaders.

The central part of the interface is the TensorBoard extension. It displays a graph of operations. A red box highlights the main graph area, which shows the flow from input through various layers (Conv2d, MaxPool2d, Linear) to the final output. The graph legend indicates that orange nodes represent operations like `Linear[fc3]` and `Linear[fc2]`, while green nodes represent `Conv2d[con...]` and `MaxPool2d[p...]`. The sidebar on the right provides details about the `Net` subgraph, which consists of 25 nodes. It lists one input node (`input/input.1`) and one output node (`output/output.1`). The status bar at the bottom shows the current Python version (Python 3.8.5 64-bit ('profiler': conda)), the number of errors (0), the number of warnings (0), and the user's Azure account (jemew@microsoft.com).

Aim  989

Compare 1000s of AI experiments at once

- > Get started
- > Get involved

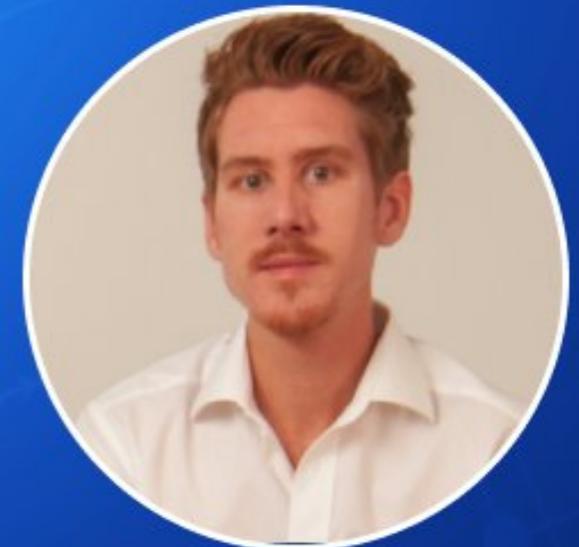


<https://aimstack.io/>

TALK

MACHINE LEARNING

Machine Learning and AI in 2021: Recent Trends, Technologies, and Challenges



Sebastian Raschka, PhD

Professor, Researcher, Author of 'Python Machine Learning'
University of Wisconsin-Madison

ODSC[®] EAST
VIRTUAL TRAINING CONFERENCE

March 30th – April 1st

"Machine Learning and AI in 2021: Recent Trends, Technologies, and Challenges" **March 31, 2021. 10:40 am CT**

<https://odsc.com/boston/>