

Machine Learning Systems Design

Lecture 7: Versioning & experiment tracking





PSET collaboration policy



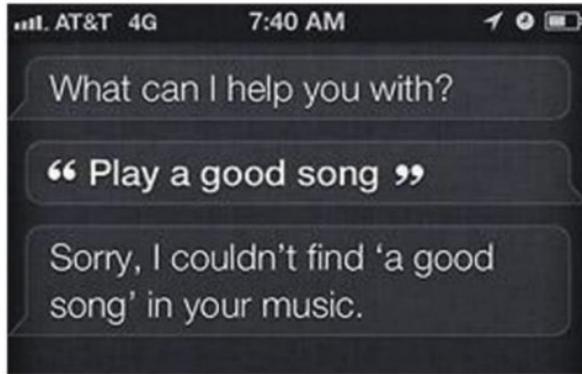
You can work in a group of 2 people, but

EACH MUST WRITE ANSWERS INDIVIDUALLY



PSET 1 highlights

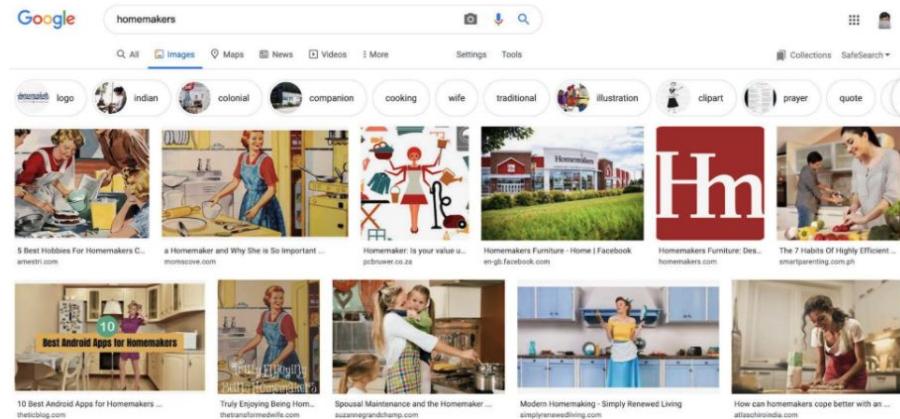
- Spam filter for search \neq spam filter for emails
- Fraud detection for online transaction (interesting [read](#))
 - False positives: immediate loss
 - False negatives: delayed loss



When Siri thinks your music is trash

Google Translate: The term / idiom "Hip hip hooray" is commonly used as a celebration / congratulation. However, its translated Korean counterpart "엉덩이 엉덩이 만세" literally means "Buttocks buttocks hooray".

| 영어 ▾ | ↔ | 한국어 ▾ |
|----------------------------------|---|---|
| Hip hip hooray hip hip hooray | × | 엉덩이 엉덩이 만세 eongdeong-i eongdeong-i manse |



The image shows a Google search results page with the query "cats like hitler". The top result is a link to "cats like hitler - Google Search". Below it are several other search suggestions: "cats like", "cats like a good party rizzi", "cats like us", "cats like dogs", "cats like earwax", "cats like milk", "cats like music", "cats like cheese", and "cats like plain crisps". To the right of the search bar, there is a microphone icon. Below the search bar, there are four images arranged in a 2x2 grid. The top-left image is a black and white profile of Adolf Hitler looking upwards. The top-right image is a black and white portrait of Hitler from the chest up. The bottom-left image is a close-up of a white cat with a black patch on its head. The bottom-right image is another black and white portrait of Hitler.

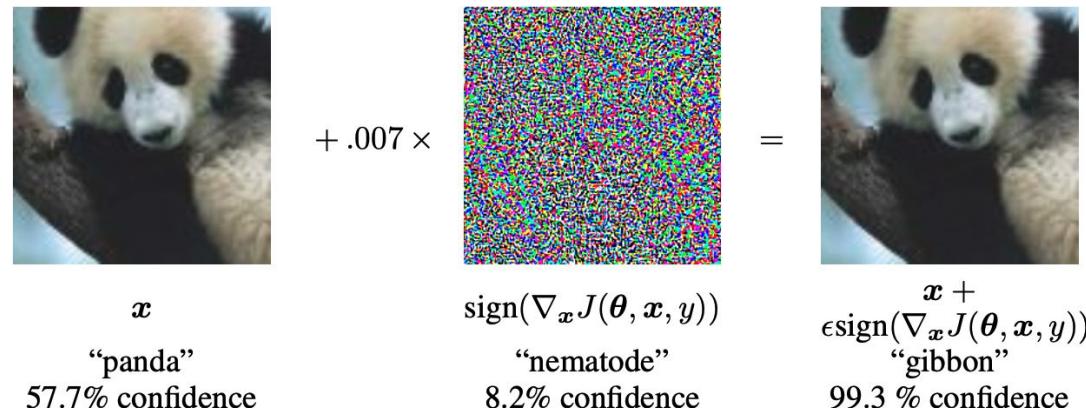
Perturbation evaluation

Perturbation evaluation

- Randomly add small noise to data to see how much outputs change
 - User inputs might contain noise
 - Speech recognition: background noise
 - Object detection: different lighting
 - Text inputs: typos, intentional changes (e.g. loooooooooong)
 - Do small changes in inputs lead to big changes in outputs?

Perturbation evaluation

- Randomly add noise to data to see how much outputs change
- The more sensitive the model is to noise:
 - The harder it is to maintain
 - The more vulnerable the model is to adversarial attacks



Invariance tests

- Some input changes shouldn't lead to changes in outputs
 - Changing race/gender info shouldn't change predicted approval outcome
 - Changing name shouldn't affect resume screening results

The Berkeley study found that both face-to-face and online lenders rejected a total of 1.3 million creditworthy black and Latino applicants between 2008 and 2015. Researchers said they believe the applicants "would have been accepted had the applicant not been in these minority groups." That's because when they used the income and credit scores of the rejected applications but deleted the race identifiers, the mortgage application was accepted.

Directional expectation tests

- Some changes to inputs should cause predictable changes in outputs
 - E.g. when predicting housing prices:
 - Increasing lot size shouldn't decrease the predicted price
 - Decreasing square footage shouldn't increase the predicted price

Ablation study

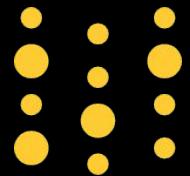
- Systematically removing parts of your model to see which ones are relevant to the model's performance
 - Features
 - Model components: e.g. regularizations

Short-term results ≠ long-term results

- Identical results short-term don't mean identical results long-term
- Superior models now don't mean superior models a week from now on

Agenda

1. Weights & Biases tutorials
2. DVC tutorials



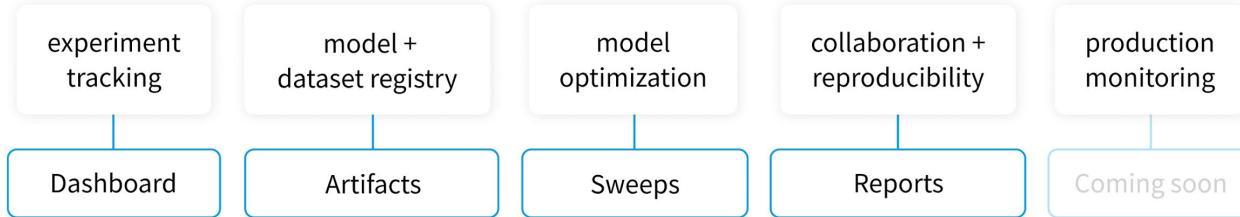
Weights & Biases

Developer tools for machine learning

ML Developer Tools for the entire ML workflow



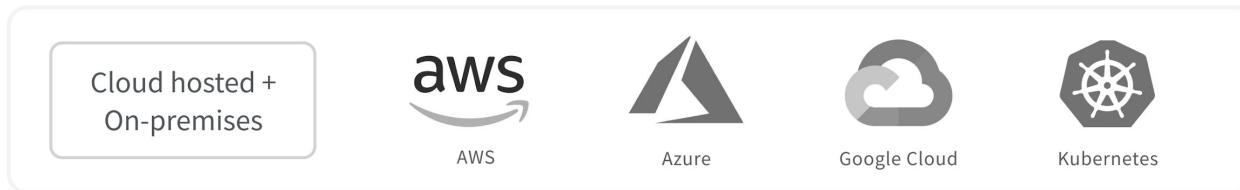
MODULAR TOOLS



FRAMEWORK AGNOSTIC



ENVIRONMENT AGNOSTIC





PROBLEM

Massive developer tools gap for ML

SOFTWARE 1.0 - WRITE CODE

| Design | Version Code | Code & Collaborate | Deploy to Infra | CI/CD | Production Monitoring |
|-----------------|------------------|--------------------|---------------------|---------------------|-----------------------|
| Figma Sketch | GitHub GitLab | Jira VS Code | HashiCorp puppet | circleci Jenkins | PagerDuty DATADOG |
| | | | | | |

SOFTWARE 2.0 - TRAIN MODELS

| Prep & Visualize Data | Version Data & Models | Experiment Tracking | Manage Model Pipeline | Model CI/CD | Production monitoring |
|--------------------------|-----------------------|---------------------------|-----------------------|----------------|-----------------------|
| Custom apps Notebooks | Files in S3 | Text files Screenshots | Text files | Custom scripts | Nothing |
| | | | | | |



W&B Company



Chris Van Pelt
Co-Founder, CVP
Powerset, Figure Eight



Lukas Biewald
Co-Founder, CEO
Powerset, Figure Eight, OpenAI



Shawn Lewis
Co-Founder, CTO
Google

Customers

US Federal



Pacific Northwest
NATIONAL LABORATORY

Enterprise



Academia



Investors

COATUE

Bloomberg
BETA

Trinity
VENTURES



Richard Socher
Former Chief
Scientist, Salesforce



Nat Friedman
CEO, GitHub



Tom
Preston-Werner
Co-Founder, GitHub



Greg Brockman
CTO, OpenAI



Vladlen Koltun
Chief Scientist, Intel



Jeff Hammerbacher
Founder, Cloudera

Weights & Biases

1. Install the lightweight library
2. Track metrics and artifacts
3. Compare experiments fast
4. Optimize hyperparameters easily
5. Collaborate on reproducible models

Light integration

Try W&B in 5 minutes:

- Intro notebook
bit.ly/intro-wb
- Quickstart docs
docs.wandb.com

PYTORCH

```
import torch  
model.train()
```

```
import torch  
import wandb  
  
wandb.watch(model)  
model.train()
```

TENSORFLOW

```
import tensorflow as tf  
  
classifier.train()
```

```
import tensorflow as tf  
import wandb  
  
wandb.init(sync_tensorboard=True)  
  
classifier.train()
```

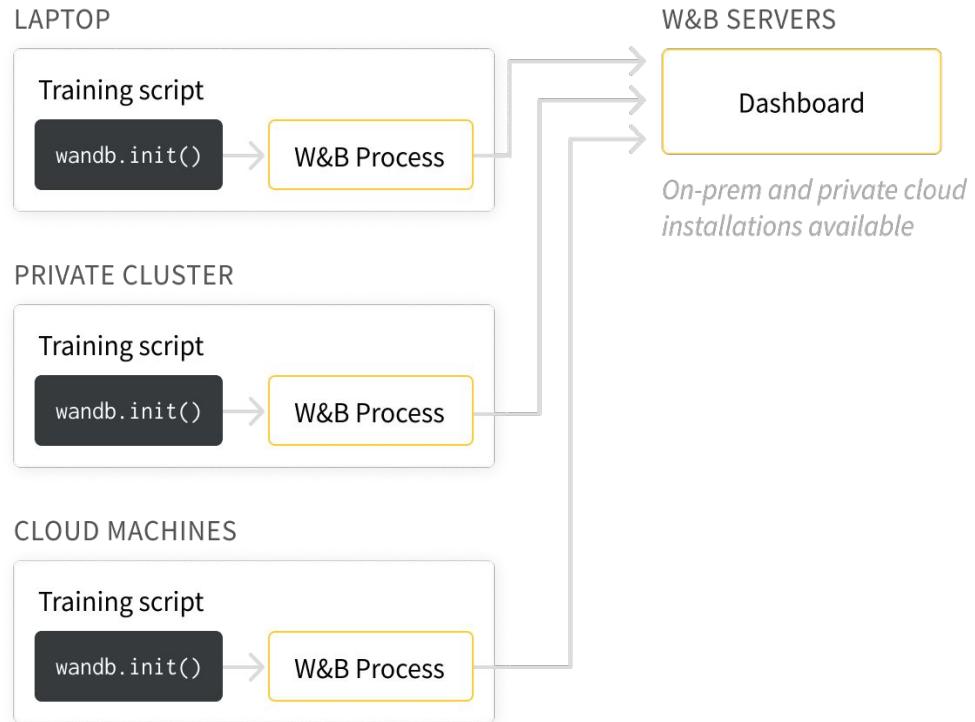
KERAS

```
from tensorflow import keras  
  
model.fit(X, y)
```

```
from tensorflow import keras  
import wandb  
  
model.fit(X, y,  
          callbacks=[WandbCallback()])
```

Infra agnostic

Get started quickly with W&B in our hosted cloud, private cloud, or on-prem



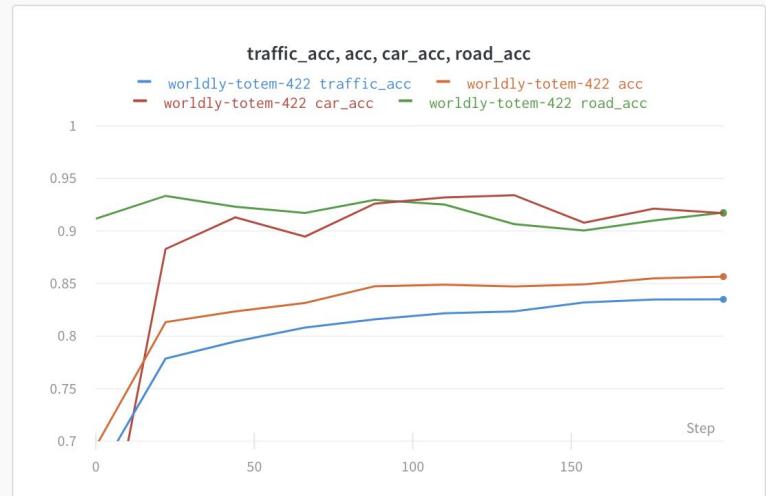
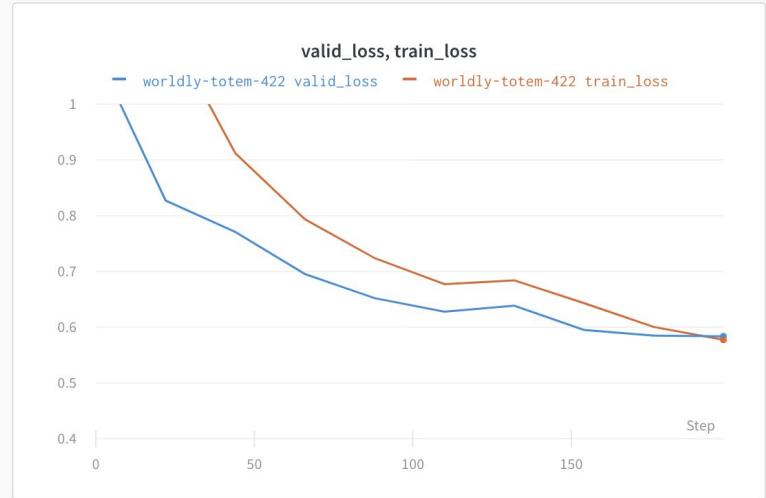
Weights & Biases

1. Install the lightweight library
2. Track metrics and artifacts
3. Compare experiments fast
4. Optimize hyperparameters easily
5. Collaborate on reproducible models

bit.ly/demo-run

Track experiments

See live updates on model performance, check for overfitting, and visualize how a model performs on different classes.



Track experiments – Before W&B

| Layer (type) | Output Shape | Param # | Connected to |
|--|---------------------|---------|--|
| <hr/> | | | |
| conv2d_1_input (InputLayer) | (None, 299, 299, 3) | 0 | |
| lambda_1 (Lambda) | (None, 299, 299, 3) | 0 | conv2d_1_input[0][0] |
| lambda_2 (Lambda) | (None, 299, 299, 3) | 0 | conv2d_1_input[0][0] |
| sequential_1 (Sequential) | (None, 10) | 910922 | lambda_1[0][0] lambda_2[0][0] |
| activation_7 (Concatenate) | (None, 10) | 0 | sequential_1[1][0] sequential_1[2][0] |
| <hr/> | | | |
| Total params: | 910,922 | | |
| Trainable params: | 910,922 | | |
| Non-trainable params: | 0 | | |
| <hr/> | | | |
| None | | | |
| Found 49999 images belonging to 10 classes. | | | |
| Found 8000 images belonging to 10 classes. | | | |
| Epoch 1/50 | | | |
| 39/39 [=====] - 270s 7s/step - loss: 2.3153 - acc: 0.1178 - val_loss: 2.2428 - val_acc: 0.1589 | | | |
| Epoch 2/50 | | | |
| 39/39 [=====] - 263s 7s/step - loss: 2.2588 - acc: 0.1538 - val_loss: 2.1890 - val_acc: 0.2174 | | | |
| Epoch 3/50 | | | |
| 39/39 [=====] - 262s 7s/step - loss: 2.2060 - acc: 0.1827 - val_loss: 2.1767 - val_acc: 0.2096 | | | |
| Epoch 4/50 | | | |
| 39/39 [=====] - 263s 7s/step - loss: 2.1640 - acc: 0.2107 - val_loss: 2.1133 - val_acc: 0.2357 | | | |
| Epoch 5/50 | | | |
| 39/39 [=====] - 261s 7s/step - loss: 2.0827 - acc: 0.2432 - val_loss: 2.1499 - val_acc: 0.2344 | | | |
| Epoch 6/50 | | | |
| 39/39 [=====] - 262s 7s/step - loss: 2.0810 - acc: 0.2508 - val_loss: 2.0289 - val_acc: 0.2604 | | | |
| Epoch 7/50 | | | |
| 39/39 [=====] - 262s 7s/step - loss: 2.0575 - acc: 0.2602 - val_loss: 1.9912 - val_acc: 0.2865 | | | |
| Epoch 8/50 | | | |
| 39/39 [=====] - 261s 7s/step - loss: 2.0355 - acc: 0.2734 - val_loss: 2.0319 - val_acc: 0.2409 | | | |
| Epoch 9/50 | | | |
| 39/39 [=====] - 263s 7s/step - loss: 2.0254 - acc: 0.2829 - val_loss: 1.9364 - val_acc: 0.3307 | | | |
| Epoch 10/50 | | | |
| 39/39 [=====] - 262s 7s/step - loss: 2.0056 - acc: 0.2804 - val_loss: 1.9934 - val_acc: 0.2773 | | | |
| Epoch 11/50 | | | |
| 39/39 [=====] - 256s 7s/step - loss: 1.9678 - acc: 0.3048 - val_loss: 1.9048 - val_acc: 0.3352 | | | |
| Epoch 12/50 | | | |
| 39/39 [=====] - 260s 7s/step - loss: 1.9634 - acc: 0.3109 - val_loss: 1.8758 - val_acc: 0.3568 | | | |

Table 3: Detection results on PASCAL VOC 2007 test set. The detector is Fast R-CNN and VGG-16. Training data: “07”: VOC 2007 trainval, “07+12”: union set of VOC 2007 trainval and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. [†]: this number was reported in [2]; using the repository provided by this paper, this result is higher (68.1).

| method | # proposals | data | mAP (%) |
|-------------------|-------------|------------|-------------------|
| SS | 2000 | 07 | 66.9 [†] |
| SS | 2000 | 07+12 | 70.0 |
| RPN+VGG, unshared | 300 | 07 | 68.5 |
| RPN+VGG, shared | 300 | 07 | 69.9 |
| RPN+VGG, shared | 300 | 07+12 | 73.2 |
| RPN+VGG, shared | 300 | COCO+07+12 | 78.8 |

Table 4: Detection results on PASCAL VOC 2012 test set. The detector is Fast R-CNN and VGG-16. Training data: “07”: VOC 2007 trainval, “07+12”: union set of VOC 2007 trainval+test and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. [†]: http://host.robots.ox.ac.uk:8080/anonymous/YNPLXB.html; [‡]: http://host.robots.ox.ac.uk:8080/anonymous/HZJ1QA.html; [§]: http://host.robots.ox.ac.uk:8080/anonymous/XEDH10.html.

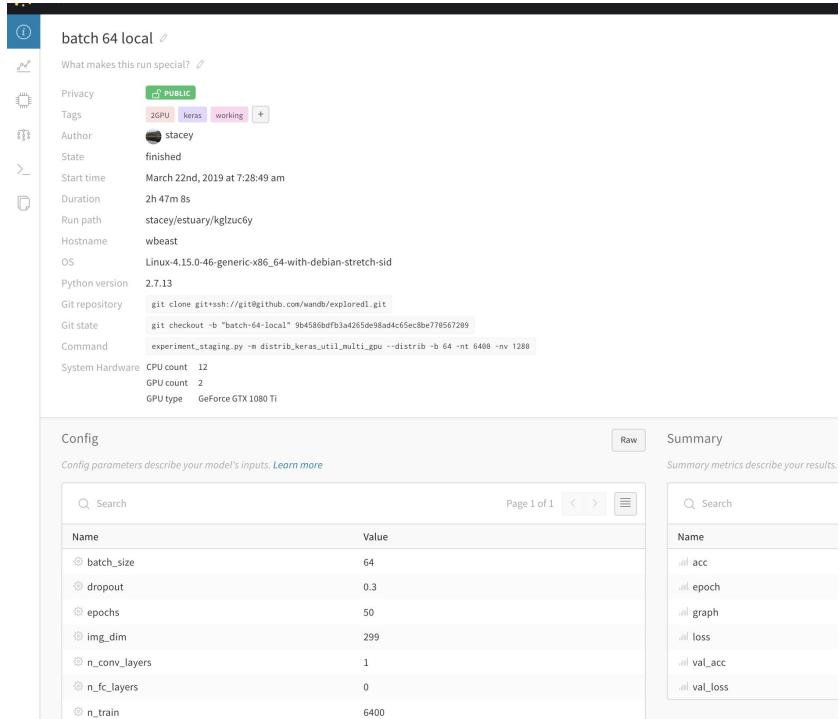
| method | # proposals | data | mAP (%) |
|------------------------------|-------------|------------|---------|
| SS | 2000 | 12 | 65.7 |
| SS | 2000 | 07+12 | 68.4 |
| RPN+VGG, shared [†] | 300 | 12 | 67.0 |
| RPN+VGG, shared [‡] | 300 | 07+12 | 70.4 |
| RPN+VGG, shared [§] | 300 | COCO+07+12 | 75.9 |

Table 5: Timing (ms) on a K40 GPU, except SS proposal is evaluated in a CPU. “Region-wise” includes NMS, pooling, fully-connected, and softmax layers. See our released code for the profiling of running time.

| model | system | conv | proposal | region-wise | total | rate |
|-------|------------------|------|----------|-------------|-------|---------|
| VGG | SS + Fast R-CNN | 146 | 1510 | 174 | 1830 | 0.5 fps |
| VGG | RPN + Fast R-CNN | 141 | 10 | 47 | 198 | 5 fps |
| ZF | RPN + Fast R-CNN | 31 | 3 | 25 | 59 | 17 fps |

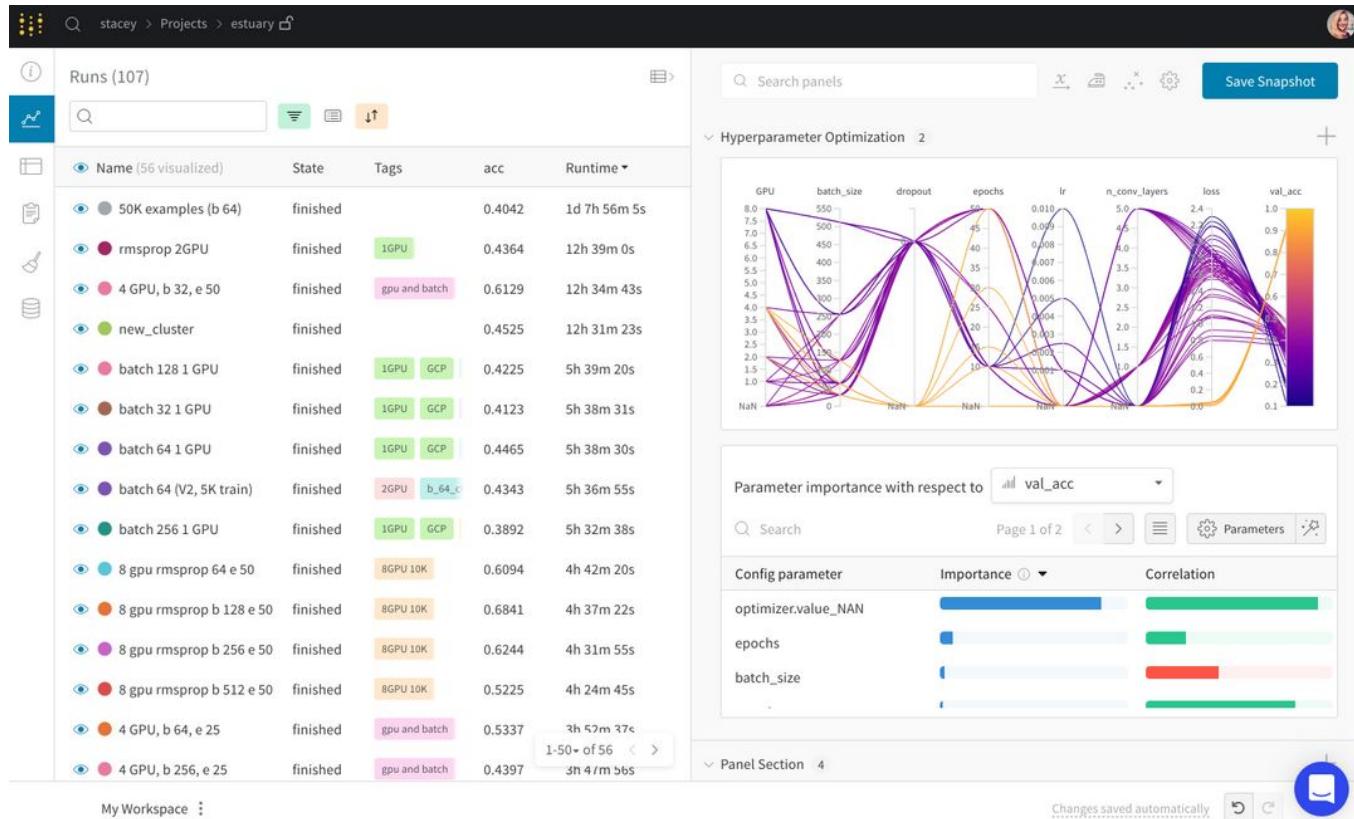
| | backbone | AP | AP ₅₀ | AP ₇₅ | AP _S | AP _M | AP _L |
|---------------------------|--------------------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
| <i>Two-stage methods</i> | | | | | | | |
| Faster R-CNN++ [5] | ResNet-101-C4 | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster R-CNN w FPN [8] | ResNet-101-FPN | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| Faster R-CNN by G-RMI [6] | Inception-ResNet-v2 [21] | 34.7 | 55.5 | 36.7 | 13.5 | 38.1 | 52.0 |
| Faster R-CNN w TDM [20] | Inception-ResNet-v2-TDM | 36.8 | 57.7 | 39.2 | 16.2 | 39.8 | 52.1 |
| <i>One-stage methods</i> | | | | | | | |
| YOLOv2 [15] | DarkNet-19 [15] | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD513 [11, 3] | ResNet-101-SSD | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| DSSD513 [3] | ResNet-101-DSSD | 33.2 | 53.3 | 35.2 | 13.0 | 35.4 | 51.1 |
| RetinaNet [9] | ResNet-101-FPN | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| RetinaNet [9] | ResNeXt-101-FPN | 40.8 | 61.1 | 44.1 | 24.1 | 44.2 | 51.2 |
| YOLOv3 608 × 608 | Darknet-53 | 33.0 | 57.9 | 34.4 | 18.3 | 35.4 | 41.9 |

Track experiments – After W&B



[Live Dashboard](#)

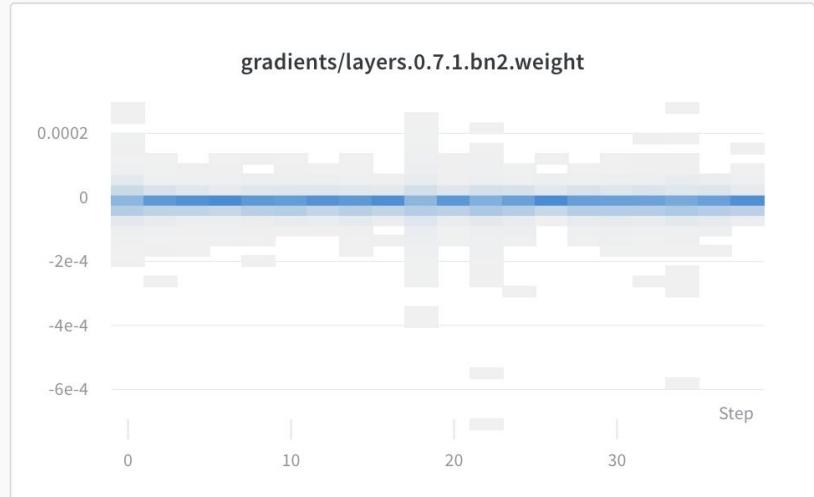
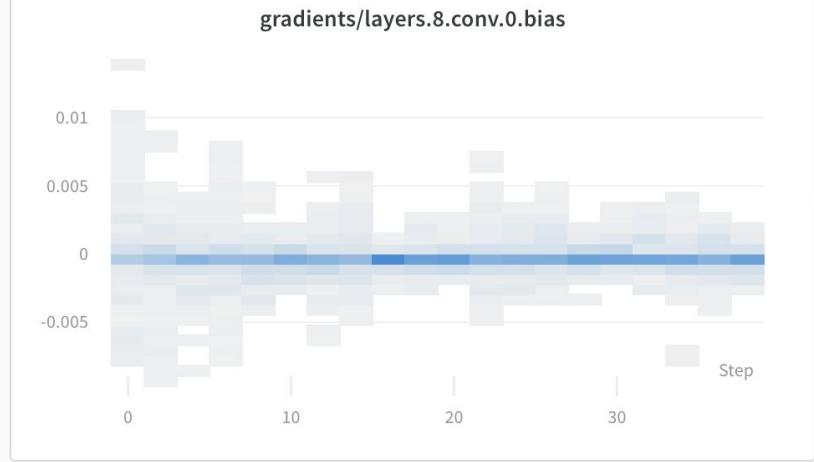
Track experiments – After W&B



bit.ly/demo-run

Visualize gradients

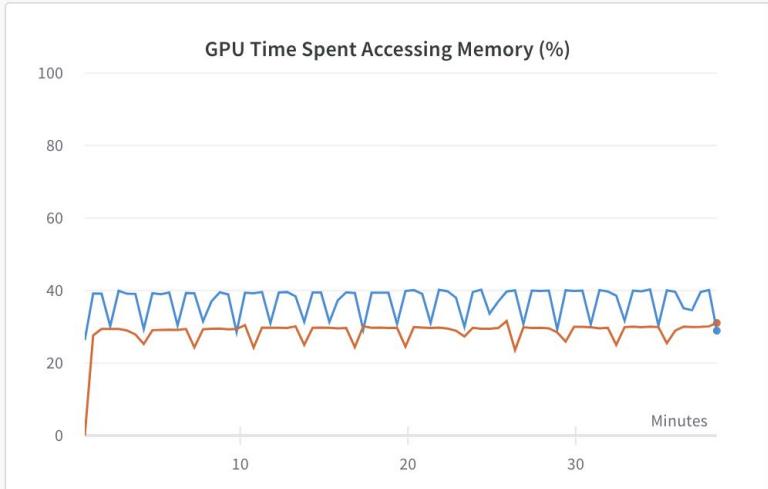
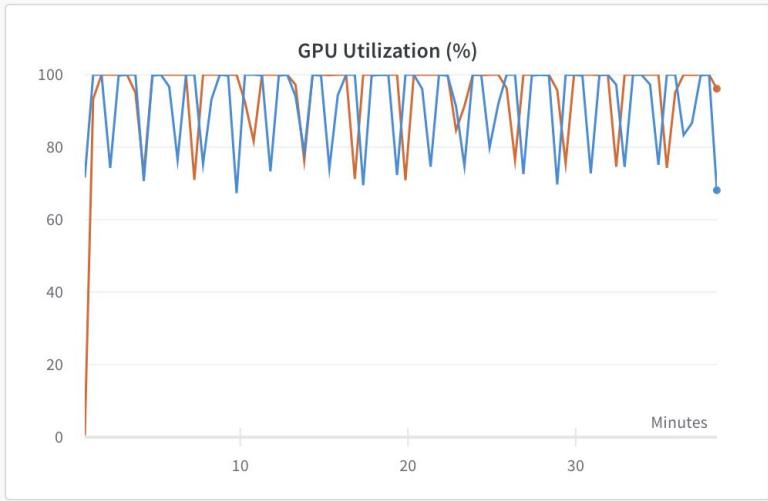
See the inner workings of your model, look for convergence during training, and check for exploding gradients.



bit.ly/demo-run

System metrics

Get the most out of your GPUs
and identify opportunities for
optimizing hardware utilization.



bit.ly/demo-run

Capture the code

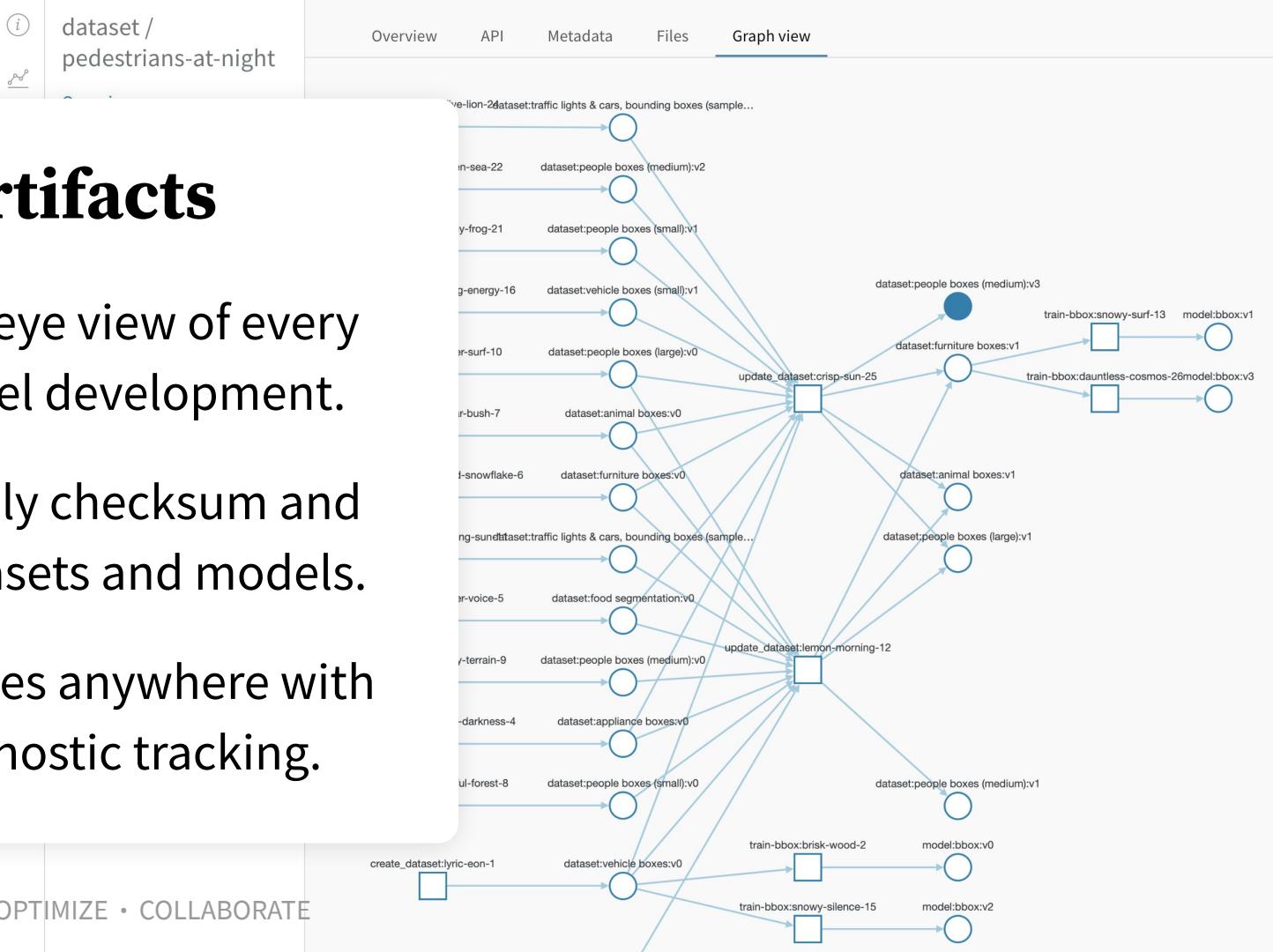
Save the most recent git commit,
the command and args, and
system hardware setup.

W&B also saves a patch file with
uncommitted changes so you
can reproduce the exact code
that trained the model.

dutiful-dragon-174 ↗

What makes this run special? ↗

| | | | | | |
|-------------------|--|-----------|---|----------|---------------------|
| Privacy |  PUBLIC | | | | |
| Tags | + | | | | |
| Author | stacey | | | | |
| State | finished | | | | |
| Start time | January 24th, 2020 at 1:30:16 pm | | | | |
| Duration | 38m 27s | | | | |
| Run path | stacey/deep-drive/6zsn8ltb | | | | |
| Hostname | wbrave | | | | |
| OS | Linux-5.0.0-37-generic-x86_64-with-debian-buster-sid | | | | |
| Python version | 3.7.2 | | | | |
| Python executable | /home/stacey/.pyenv/versions/sd/bin/python | | | | |
| Git repository | <code>git clone https://github.com/borisdayma/semantic-segmentation.git</code> | | | | |
| Git state | <code>git checkout -b "dutiful-dragon-174" f0a9494a5d152663d226e28e279c65</code> | | | | |
| Command | <code>train.py</code> | | | | |
| CPU count | 20 | | | | |
| System Hardware | <table><tr><td>GPU count</td><td>2</td></tr><tr><td>GPU type</td><td>GeForce RTX 2080 Ti</td></tr></table> | GPU count | 2 | GPU type | GeForce RTX 2080 Ti |
| GPU count | 2 | | | | |
| GPU type | GeForce RTX 2080 Ti | | | | |
| W&B CLI Version | 0.8.21 | | | | |



Track artifacts

Get a bird's eye view of every step of model development.

Automatically checksum and
version datasets and models.

Host your files anywhere with our infra-agnostic tracking.

Weights & Biases

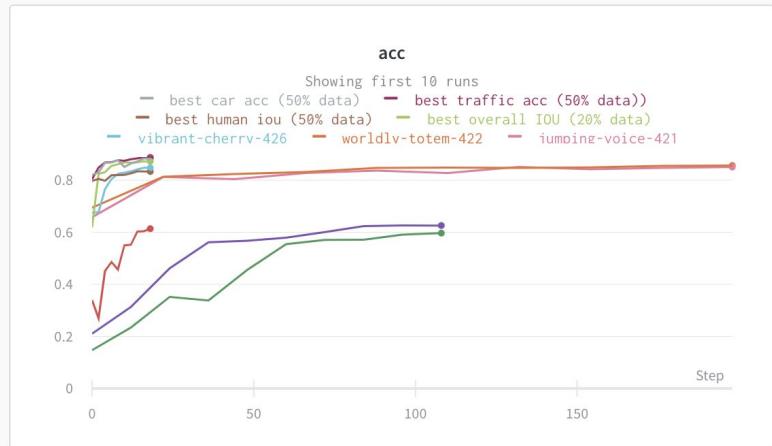
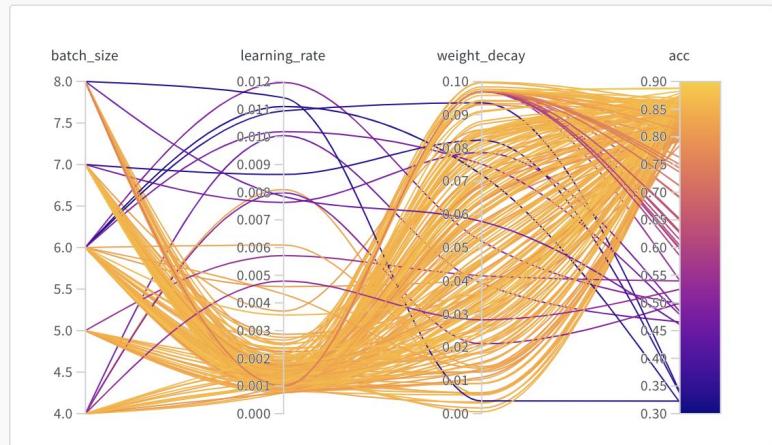
1. Install the lightweight library
2. Track metrics and artifacts
3. Compare experiments fast
4. Optimize hyperparameters easily
5. Collaborate on reproducible models

bit.ly/deep-drive

Iterate quickly

Use the interactive dashboard to spot issues in real time.

Stop underperforming runs early to optimize resource utilization.

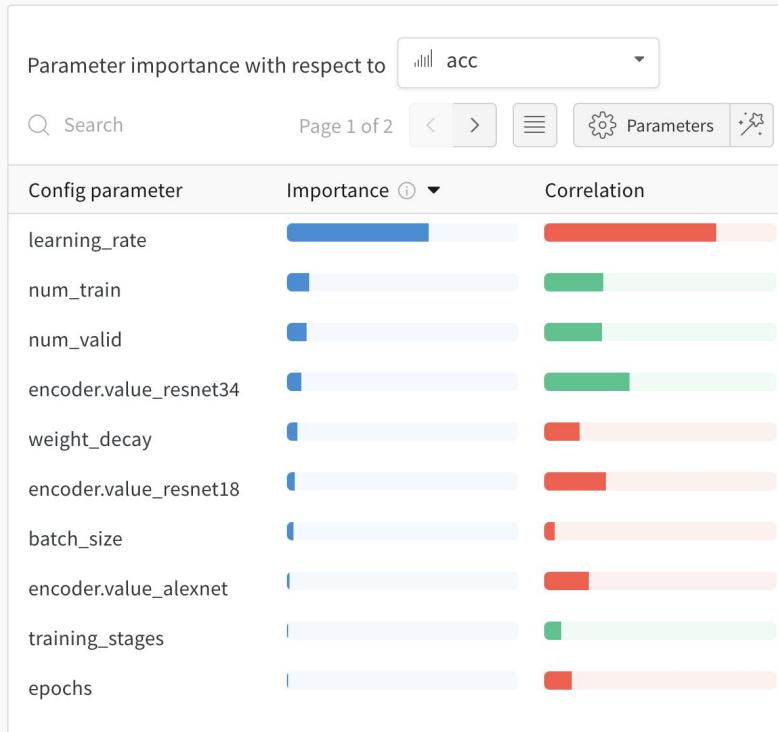


bit.ly/deep-drive

Compare runs

Visualize the relationships between hyperparameters and model metrics.

Explore the space of possible models quickly, without getting bogged down setting up manual visualizations.



bit.ly/deep-drive

Persistent system of record

Query and filter across thousands of runs, and easily keep projects organized.

| Runs (395) | | | | | | | | | | | | | | |
|--|-----------|---------|------------|----------|---------------|-----------|-----------|--------------|--------|------------|------------|--------|-------------|------|
| <input type="checkbox"/> Name (228 visualized) | Tags | Runtime | batch_size | encoder | learning_rate | num_train | num_valid | weight_decay | iou | train_loss | valid_loss | acc | traffic_acc | road |
| best car acc (50% data) | seg_masks | 47m 52s | 6 | resnet34 | 0.001311 | 3524 | 492 | 0.08173 | 0.7997 | 0.5375 | 0.4427 | 0.8823 | 0.8664 | 0.93 |
| best traffic acc (50% data) | seg_masks | 46m 42s | 8 | resnet18 | 0.001 | 3523 | 492 | 0.097 | 0.8073 | 0.4919 | 0.4203 | 0.888 | 0.8718 | 0.94 |
| best human iou (50% data) | seg_masks | 31m 34s | 7 | alexnet | 0.0009084 | 1405 | 190 | 0.097 | 0.716 | 0.6222 | 0.6259 | 0.8334 | 0.8042 | 0.9 |
| best overall IOU (20% data) | seg_masks | 20m 23s | 7 | resnet34 | 0.001367 | 1376 | 205 | 0.06731 | 0.7948 | 0.5096 | 0.4659 | 0.8726 | 0.8593 | 0.93 |
| vibrant-cherry-426 | | 6m 12s | 8 | resnet34 | 0.001 | 347 | 42 | 0.097 | 0.752 | 0.7114 | 0.6055 | 0.8474 | 0.8282 | 0.95 |
| worldly-totem-422 | | 12m 54s | 8 | resnet34 | 0.001 | 682 | 97 | 0.097 | 0.7523 | 0.5774 | 0.5836 | 0.8566 | 0.8349 | 0.91 |
| jumping-voice-421 | | 11m 59s | 8 | resnet34 | 0.001 | 725 | 92 | 0.097 | 0.7449 | 0.5633 | 0.5334 | 0.8504 | 0.8296 | 0.91 |
| logical-energy-420 | test_only | 2m 14s | 8 | resnet34 | 0.001 | 66 | 10 | 0.097 | 0.4297 | 1.459 | 1.221 | 0.626 | 0.5958 | 0.76 |

Weights & Biases

1. Install the lightweight library
2. Track metrics and artifacts
3. Compare experiments fast
4. Optimize hyperparameters easily
5. Collaborate on reproducible models

bit.ly/deep-drive-sweep

HYPERPARAMETER SWEEPS

Scale up training

Easily launch dozens of parallel experiments to quickly optimize models.



bit.ly/deep-drive-sweep

HYPERPARAMETER SWEEPS

Iterate + explore

Use manual experiments
and previous sweeps to
seed new searches.

Prior Runs

Select prior runs to feed into your sweep. Random and Grid search sweeps will skip these runs. Bayesian optimization will use these runs to inform where to search next. [See the documentation](#)

| Filter | Group | Sort | Tag | Move | trash | ↓ | Columns |
|-----------------------------|-------|---------|-----------|----------|--------|------------|----------|
| (396 visualized) | | Runtime | Notes | State | Tags | batch_size | encoder |
| curious-meadow-434 | | 50m 6s | Add notes | finished | | 6 | resnet34 |
| seed-galaxy-433 | | 33s | Add notes | crashed | | 6 | resnet34 |
| test car acc (50% data) | | 47m 52s | reprod... | finished | seg_ma | 6 | resnet34 |
| test traffic acc (50% data) | | 46m 42s | reprod... | finished | seg_ma | 8 | resnet18 |
| test human iou (50% data) | | 31m 34s | reprod... | finished | seg_ma | 7 | alexnet |
| test overall IOU (20% data) | | 20m 23s | reprod... | finished | seg_ma | 7 | resnet34 |
| useful-violet-428 | | 34s | Add notes | failed | | 8 | resnet34 |
| major-firefly-427 | | 8s | Add notes | failed | | 8 | resnet34 |
| vibrant-cherry-426 | | 6m 12s | Add notes | finished | | 8 | resnet34 |
| good-cosmos-425 | | 43s | Add notes | failed | | 8 | resnet34 |

bit.ly/deep-drive-sweep

[Sweeps docs](#)

HYPERPARAMETER SWEEPS

Fast setup

Use Bayesian sweeps and
early stopping without
writing custom code.

INSTALL • TRACK • COMPARE • **OPTIMIZE** • COLLABORATE

The screenshot shows the WandB interface with a sidebar on the left containing icons for project, sweeps, runs, and profiles. The main area displays a sweep configuration titled "max iou".

max iou (edit)

| | |
|--------------|--|
| ID | lnrd5iw8 |
| Privacy | PUBLIC |
| Created | 1/30/2020, 2:30:22 PM |
| Last updated | 1/30/2020, 2:30:22 PM |
| Launch agent | <code>\$ wandb agent stacey/deep-drive/lnrd5iw8</code> |
| Author | stacey |

Sweep Configuration

```
method: bayes
metric:
  goal: maximize
  name: iou
name: max iou
parameters:
  batch_size:
    values:
```

Weights & Biases

1. Install the lightweight library
2. Track metrics and artifacts
3. Compare experiments fast
4. Optimize hyperparameters easily
5. Collaborate on reproducible models

bit.ly/deep-drive-report

REPORTS

Annotate progress

Use reports to keep a work log and quickly pick up where you left off.

Reports

Last edited

Created by

The View from the Driver's Seat

segmentation for scene
Berkeley Deep Drive 100K



1 month ago

stacey

asks for Semantic tation

; and explore semantic
ion masks



1 month ago

stacey

c Segmentation Masks

; and explore semantic
ion masks



3 months ago

stacey

c Segmentation Demo

e segmentation model using
; car scenes. Click the Gear
interact with the scene.



3 months ago

nbaryd

[WIP] Semantic Segmentation from Dashcam

Ongoing notes, exploration, and
development



4 months ago

stacey

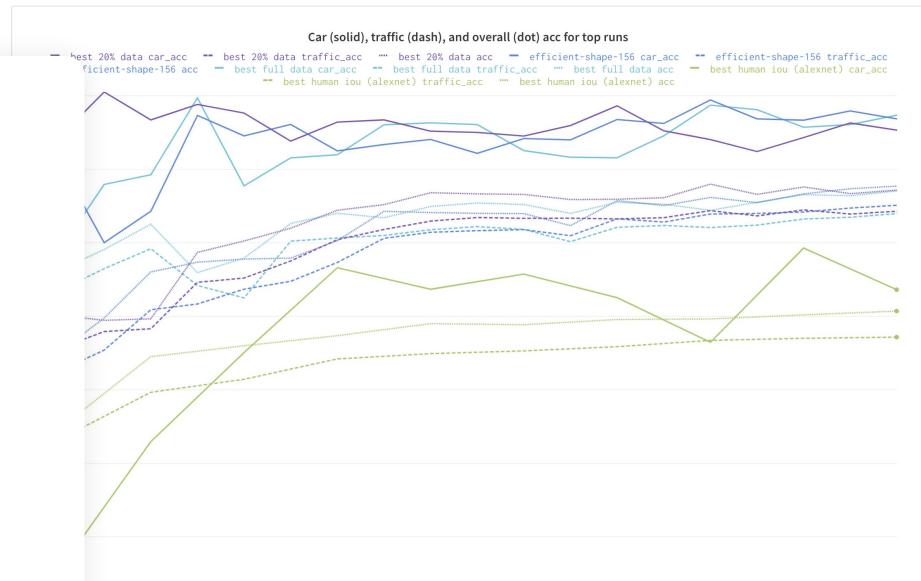
bit.ly/deep-drive-report

REPORTS

Live dashboards

Customize real-time views of model training and evaluation, and share auto-updating dashboards with your team to keep everyone in the loop.

Comparing per-class accuracies



top models 4

| (visualized) | acc ▾ | car_acc | traffic_acc | train_loss | valid_loss | encoder | learning_ra | num_train | dur |
|--------------------------|--------|---------|-------------|------------|------------|----------|-------------|-----------|-----|
| est 20% data | 0.8866 | 0.931 | 0.8724 | 0.4421 | 0.4284 | resnet18 | 0.001 | 1421 | 183 |
| efficient-shape-156 | 0.8863 | 0.9329 | 0.8739 | 0.4321 | 0.4394 | resnet18 | 0.001 | 1409 | 200 |
| best full data | 0.8848 | 0.9356 | 0.8697 | 0.5218 | 0.4242 | resnet18 | 0.001 | 7000 | 100 |
| best human iou (alexnet) | 0.8036 | 0.8181 | 0.7859 | 0.6373 | 0.708 | alexnet | 0.0009084 | 699 | 106 |

1-4 of 4 < >

bit.ly/deep-drive-report

REPORTS

Collaborate easily

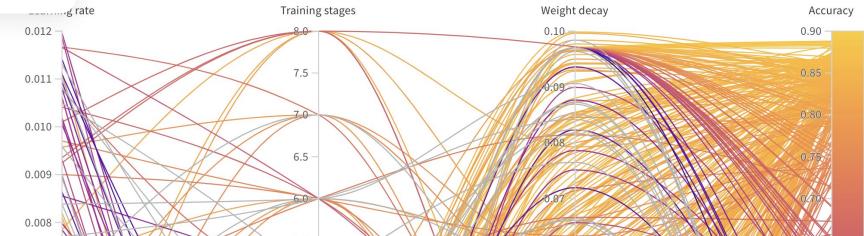
Give team members access to your exploratory results, sharing the context they need to quickly build on your work.

INSTALL • TRACK • COMPARE • OPTIMIZE • COLLABORATE

Weight decay: inconclusive

Initially increasing the weight decay 5X improved the accuracy by 9%. Increasing by 200X causes the same amount of improvement though.

Hyperparameter Sweep Insights



The Gallery

≡ 🔍

Gallery

All reports

FRAMEWORKS

- PyTorch
- Keras or TensorFlow
- Kaggle
- Scikit
- Fastai
- HuggingFace

APPLICATIONS

- Computer Vision
- Natural Language
- Reinforcement Learning

TOPICS

- Advice From Experts
- Debugging and Optimization
- Tools & Libraries
- Deep Learning Papers
- Interesting ML Techniques
- Tutorials
- Explainability
- Visualizations
- Custom Charts
- Papers on W&B

INTERNATIONAL

- Chinese
- Japanese
- Korean

Latest reports

JAN 30, 2021

Information Extraction From Documents

Tulasi Ram Laghumavarapu

Computer Vision, Natural Language, Interesting ML Te...

JAN 29, 2021

[Overview] X-Fields: Implicit Neural View-, Light- and Time-Image...

Ayush Thakur

Computer Vision, Interesting ML Techniques, Two Min...

JAN 26, 2021

Show and Tell

Aritra Roy Gosthipaty, Deviyoti Chakraborty

Keras or TensorFlow, Computer Vision, Natural Language

JAN 14, 2021

Finetuning DETR (Object Detection with Transformers) on Tensorflow - ...

Thibault Neveu

Computer Vision, Interesting ML Techniques, Tutorials,...

JAN 13, 2021

The Science of Debugging with W&B Reports

Sarah Jane

Advice From Experts, Computer Vision

DEC 22, 2020

Paper Summary: One Shot 3D Photography

Ayush Thakur

Computer Vision, Interesting ML Techniques, Two Min...

DEC 17, 2020

Introducing Quicksight with Wide Residual Networks

Saurav Maheshkar

Interesting ML Techniques, Computer Vision

Submit to gallery

ML Community

PODCASTS

- Richard Socher, Ex-Salesforce
- Joaquin Candela, Responsible AI at Facebook
- Jeremy Howard, Fastai
- Anthony Goldbloom, Kaggle
- Anantha Kancharla, Lyft Level 5

UPCOMING EVENTS

Fundamentals of DL with W&B
ft. Chris Van Pelt

February 11, 10 AM PST [RSVP →](#)

DEEP LEARNING SALON

- Andrew Ferlitsch, DL Design Patterns
- Hannes Hapke, Managing ML Pipelines
- Sara Hooker, Hardware Lottery

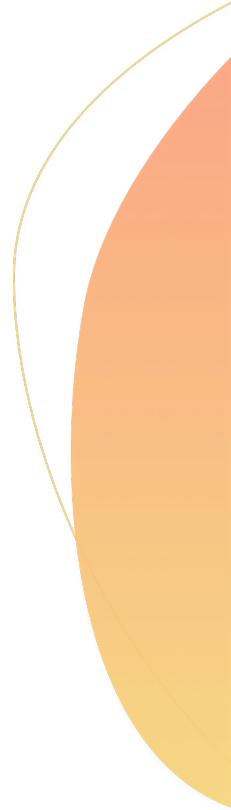
≡ 🔍

The Community: wandb.me/slack

The screenshot shows the WandB Slack interface. On the left, a sidebar lists various Slack channels, with **# ml-projects** currently selected and highlighted in blue. The main area displays a feed of messages from different users:

- December 11th, 2020**
 - Lavanya** 4:26 AM: Advanced (emoji: sun)
Expert (emoji: star)
Paper (emoji: scroll)
If you have any questions, please don't hesitate to ask [@Ayush Thakur](#) or me. (edited)
15 13 6
 - Lavanya** 4 replies Last reply 20 days ago
- December 23rd, 2020**
 - Gabriel Wai** 6:34 AM: Hi everyone, I'm having a little trouble with using IBM Watson API (Speech to Text); I'm getting this error: "ibm_cloud_sdk_core.api_exception.ApiException: Error: unable to transcode data stream audio/wav -> audio/x-float-array . Code: 400 , X-global-transaction-id: ad0494266c697eb676eacf384fce03eb". Does anyone know what I can do? I'm trying to convert my audio file into a text. Thank you
 - Gabriel Wai** 6:34 AM
- Monday, January 11th**
 - Luís Espírito Santo** 7:40 PM: Hey everyone, I'm developing a prototype (shouldn't take long time to develop) for a face recognition app that I want to be able to compile to android, IOS and windows desktop. The big question here is that I can't just use an API and run everything on a server, I need to run models (I'm currently using TF) on the edge device (android, iOS and Windows). I started implementing something in C++ but python is so much simpler for prototyping. How do you develop such kind of a prototype? Do you guys have any recommendation on courses or literature on developing ML for mobile? Any literature on how are facial filters are developed inside Instagram (assuming the models run on the device)?
 - Luís Espírito Santo** 7:40 PM

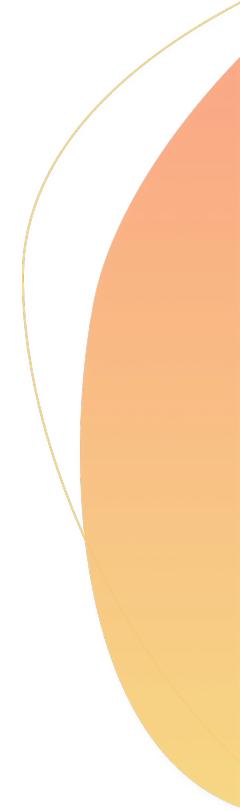
Experiment Tracking In Practice



LatentSpace

the first fully AI-rendered 3D engine

- Reports
 - Communicate results internally
 - Build on top of each others work iteratively
 - As mini research papers, which then become a paper
- Debugging
 - [Report](#)



OpenAI

- Reports
 - Every experiment the OpenAI team tries starts as a Report
 - “The ability to mix real data from experiments with context and commentary on the results was the primary selling point for us”
 - From Google docs with screenshots to interactive reports.
- Workflow
 - [Report](#)
 - Create a new report with text at the top explaining the context, hypotheses under test, and experimental plan.
 - Launch one or more experiments aimed at testing the hypotheses.
 - Monitor the experiments. If enough data has been collected after the first set of experiments, add a conclusion to the top and share with the team. Otherwise, update the hypotheses and experimental plan and return to step 2.

Wadhwani

AI for Pest Monitoring – Intelligent autonomous spraying

- Model pipeline tracking
 - [Blog Post](#)
 - **Data versioning:**
 - As datasets continue to evolve over time, train/val/test splits cannot be static. Explicitly version data to take snapshots of samples used to train any given experiment.
 - **Experiment versioning:**
 - We store all our hyperparameters for each experiment in a config file along with the Git commit ID and the specific seed.



John Deere

Production machine learning for agriculture

- Debugging through reproducible model pipelines
 - [Blog Post](#)
 - **Interactive plots for debugging**
 - Parameter importance, Custom plots, Gradients
 - Model predictions
 - **Reproducible model training pipelines with Artifacts**
 - **Efficiency:** “As we develop new models, we need to experiment quickly and share results. W&B gives us the visibility we need to debug and improve our models.”
 - **Performance:** “At the end of the day, we need to build the most accurate and fastest models for our field machines. We have full visibility and transparency in the development process with W&B, making it easy to identify the most performant models.”





Weights & Biases System of Record

Unified toolset for Machine Learning

Reproducibility

- Data lineage and provenance
- ML pipeline version control
- Maintain a knowledge base + central registry of all trained models

Visualization

- Live dashboards for fast experimentation
- Collaboration in unified workspaces and reports
- Manage visibility of machine learning projects across research teams, stakeholders, and academic community

Fast integration

- Modular solutions with lightweight setup
- Flexible deployment, with both cloud and on-prem solutions available
- Intuitive user experience built in partnership with leading research institutions

Trusted by [50,000+](#) machine learning practitioners at [200+](#) companies and research institutions

Experiment Tracking in practice

wandb.me/emotion

Weights & Biases

Docs docs.wandb.com

Instrument W&B in 5 mins github.com/wandb/examples

Gallery wandb.ai/gallery

Slack Community wandb.me/slack

Questions? twitter.com/lavanyaai

lavanya@wandb.com



Thank You!



DVC introduction

Dmitry Petrov, PhD

[@FullStackML](#)

dmitry@iterative.ai

Hello



Dmitry Petrov

PhD in Computer Science

Twitter: @FullStackML

Creator of
DVC.org project

❖ Co-Founder & CEO > Iterative.AI > San Francisco, USA

❖ ex-Data Scientist > Microsoft (BingAds) > Seattle, USA

❖ ex-Head of Lab > St. Petersburg Electrotechnical University > Russia

1

DVC principles

What DVC does?

1. Data versioning
2. ML models versioning
3. ML pipeline versioning
4. * Data & models transferring

DVC principles

- 1. Git is a foundation**
- 2. Use storage directly – speed!**
 - S3, Azure Blob, GCS, GDrive, FTP
- 3. No services**
 - Git server (like GitHub) + S3
 - Distributed (as Git)
- 4. Compatible with Git and Git-ecosystem**
 - GitHub/GitLab/BitBucket issue trackers
 - CI/CD

DVC users

1. ML researcher & ML engineer

- Create ML models
- Tune models

2. DevOps & Engineers

- Productionize models
- Manage data and labels

DVC first step

WebSite and Docs: <https://dvc.org>

Source code: <https://github.com/iterative/dvc/>

```
$ pip install dvc
```

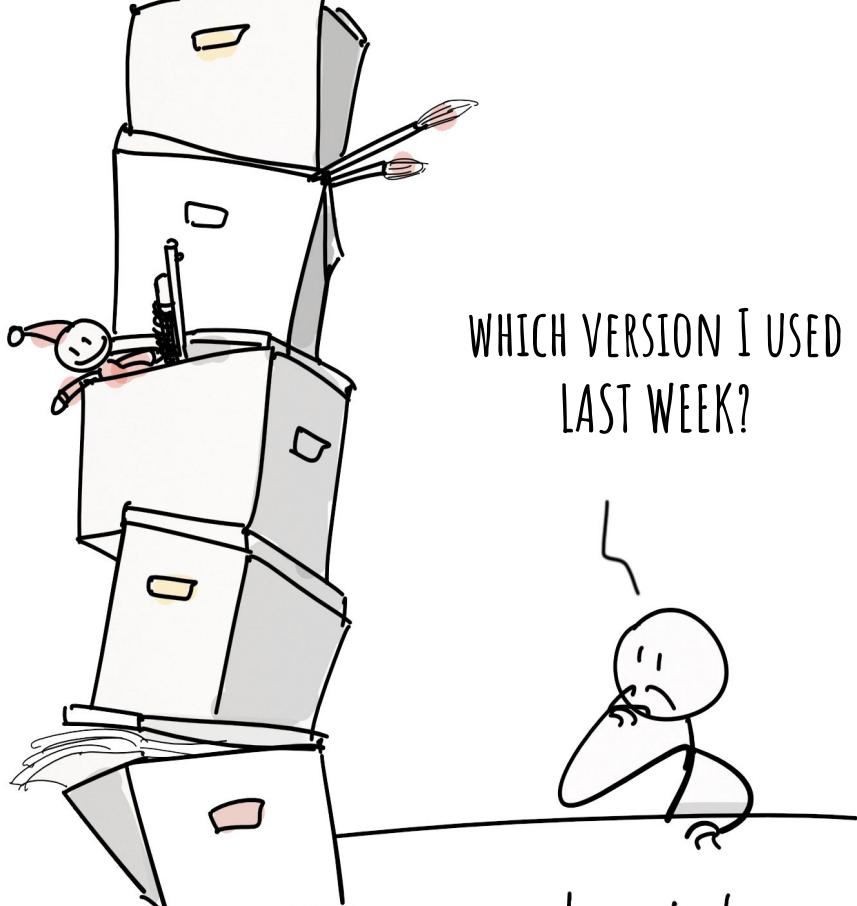
```
# Init in a project
```

```
$ cd my-project
```

```
$ dvc init
```

Define storage - data remote

```
$ dvc remote add -d mys3 s3://dmpetrov/test  
$ git add .dvc/config
```



2

Data versioning

Add data to your project

```
$ dvc add data/ # creates meta-file data.dvc
```

```
$ cat data.dvc
```

```
outs:
```

```
- md5: b8f4d5a78e55e88906d5f4aeaf43802e.dir
```

```
size: 41149064
```

```
nfiles: 1800
```

```
path: data
```

Commit meta-data

```
$ cat data.dvc
outs:
- md5: b8f4d5a78e55e88906d5f4aeaf43802e.dir
  size: 41149064
  nfiles: 1800
  path: data
```

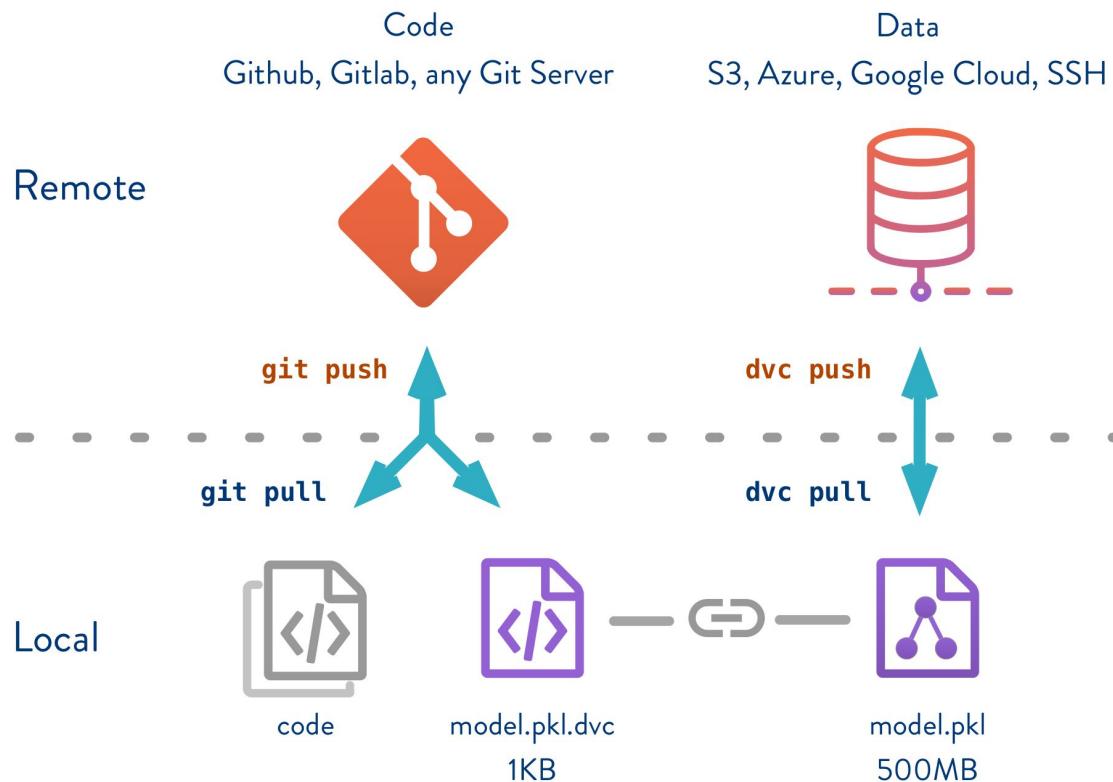
```
$ git add data.dvc
$ git commit -m "1st dataset"
$ dvc push # push dataset to the storage
```

Transfer projects

```
$ git clone https://github.com/dmpetrov/test
$ cd test
$ ls
...
data.dvc

$ dvc pull          # transfers data
$ du -sh data
43M      data.tsv
```

DVC: data meta-info



3

ML pipeline versioning

Define pipeline and metrics file

```
$ dvc run -n my_train          \
    -d data/ -d train.py      \ # dependencies
    -o model.h5              \ # model file
    -M metrics.json          \ # model metrics file
    python train.py          \ # command to run
```



```
$ cat metrics.json
{"step": 2, "loss": 2.142695903778076,
 "accuracy": 0.21351666748523712,
 "val_loss": 2.1023569107055664,
 "val_accuracy": 0.25060001015663147
}
```

Pipeline meta-file

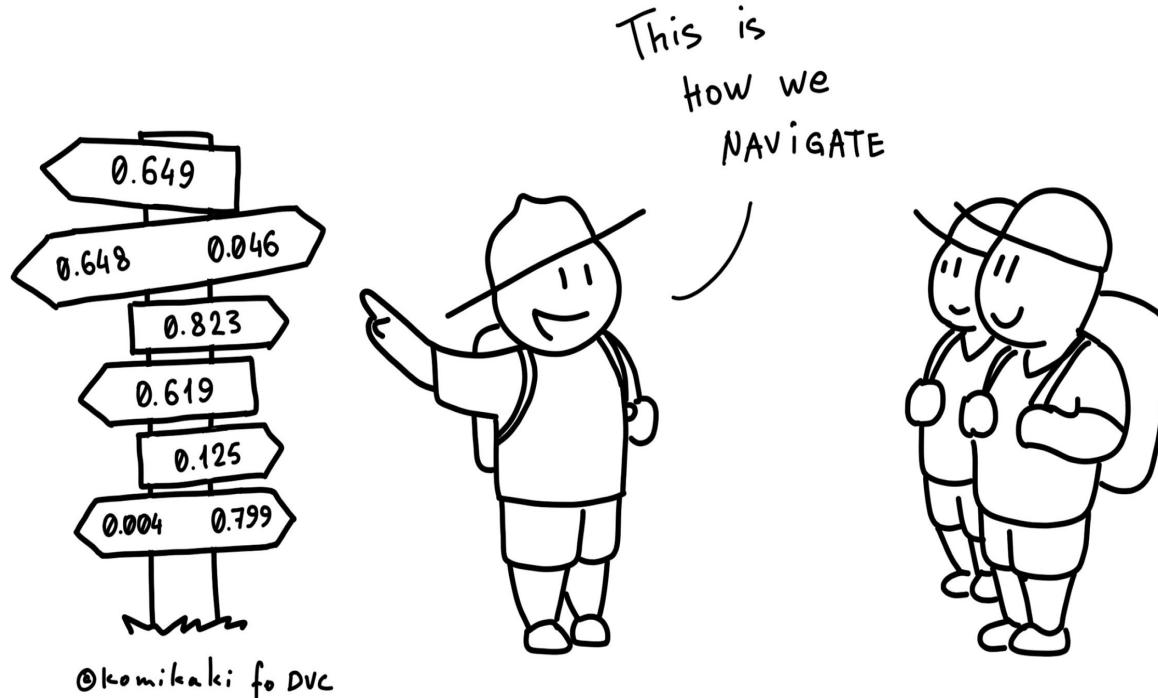
```
$ cat dvc.yaml
stages:
  my_train:
    cmd: python train.py
    deps:
      - data
      - train.py
    outs:
      - model.h5
    metrics:
      - metrics.json:
          cache: false
```

Reproduce pipeline

```
$ dvc repro
'data.dvc' didn't change, skipping
Stage 'cats-and-dogs' didn't change, skipping
Data and pipelines are up to date.
```

4

Model versioning



Metrics diff

```
$ dvc metrics diff
```

| Path | Metric | Value | Change |
|--------------|----------|----------|-----------|
| metrics.json | accuracy | 0.213516 | 0.023528 |
| metrics.json | loss | 2.142695 | -0.382401 |

Also:

```
$ dvc metrics diff HEAD^^
```

```
$ dvc metrics diff master
```

```
$ dvc metrics diff v2.1 v2.3
```

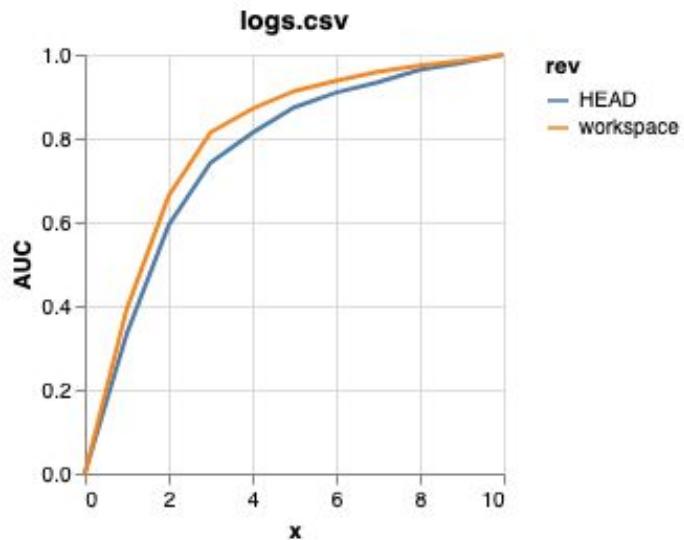
...

Plots file

```
$ dvc plots diff  
file:///Users/dmitry/  
src/exp4/plots.html →
```

Also:

```
$ dvc plots diff HEAD^^  
$ dvc plots diff master  
$ dvc plots diff v2.1 v2.3  
...
```



Getting ML model (to production)

```
$ dvc list https://github.com/iterative/myproject  
model.h5  
train.py  
...  
  
$ dvc get https://github.com/iterative/myproject model.h5  
$ du -sh model.h5  
4.7M  model.h5
```

OR in Python
Import dvc.api

Getting ML model (to production) - Python API

```
import dvc.api

with dvc.api.open(
    'get-started/data.xml',
    repo='https://github.com/iterative/dataset-registry'
) as fd:
    # ... fd is a file descriptor that can be processed normally.
```

5

Integration and compatibility CI/CD

CML – Continuous Machine Learning

CML is CI/CD for Machine Learning Projects.

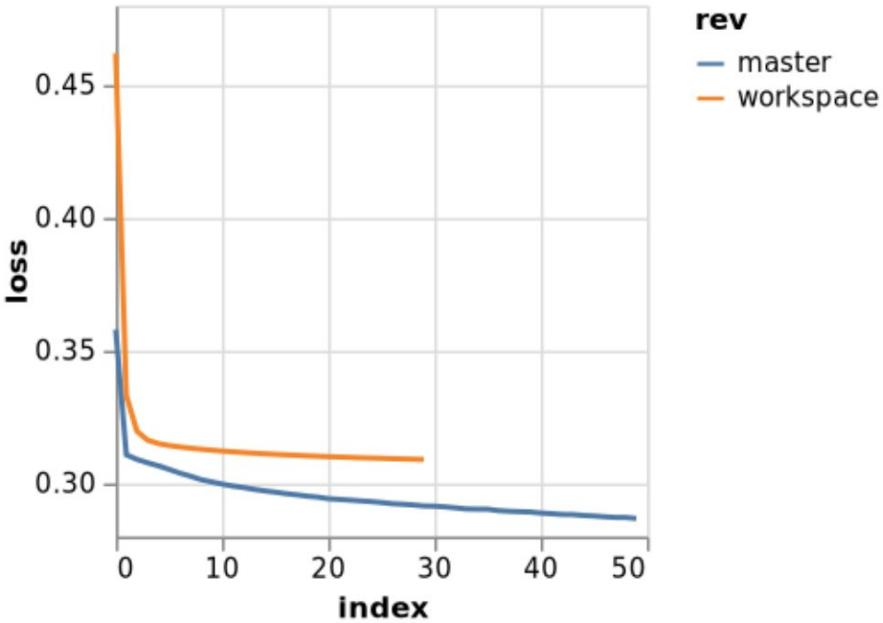
<https://cml.dev/>



github-actions bot commented on 591b626 on Jul 2, 2020

...

| Path | Metric | Value | Change |
|--------------|-----------|---------|----------|
| metrics.json | accuracy | 0.86944 | -0.00292 |
| metrics.json | precision | 0.86923 | 0.00504 |
| metrics.json | recall | 0.87185 | -0.01382 |



GitHub Action CI/CD workflow

```
dvc pull data
pip install -r requirements.txt
dvc repro

git fetch --prune
dvc metrics diff --show-md master >> report.md

dvc plots diff \
    --target loss.csv --show-vega master > vega.json
vl2png vega.json | cml-publish --md >> report.md
cml-send-comment report.md
```

Conclusion

Data codification, not versioning

DVC “codifies” data: data → meta-data

Git does the versioning (of meta-info).

Is Git version control system?

Data codification, not versioning

DVC “codifies” data: data → meta-data

Git does the versioning (of meta-info).

Is Git version control system?

```
$ man git
```

NAME

git - the stupid **content tracker**

SYNOPSIS

...

Important

1. Keep track of data, ML models and pipelines
2. Efficiently transfer data
3. Be a part of the entire software engineering stack



Thank you!

Dmitry Petrov, PhD

Twitter: [@FullStackML](https://twitter.com/@FullStackML)

Tutorial: dvc.org/doc/use-cases/versioning-data-and-model-files/tutorial

Machine Learning Systems Design

Next class: Deployment