iterative

Tools for Deep Learning Scenarios

Lesson 7

DVC Tools for Data Scientists & Analysts

2021

Course lessons

Lesson 1. Course Introduction

Lesson 2. Practices and Tools for Efficient Collaboration in ML projects

Lesson 3. Pipelines Automation and Configuration Management

Lesson 4. Versioning Data and Models

Lesson 5. Visualize Metrics & Compare Experiments with DVC and Studio

Lesson 6. Experiments Management and Collaboration

Lesson 7. Tools for Deep Learning Scenarios

Lesson 8. Review Advanced Topics and Use Cases

Lesson Outline

- DVCLive and checkpoints in deep learning scenarios
- Setting up checkpoints
- Experimenting and collaboration

DVCLive and Checkpoints in Deep Learning

DVC tools for Deep Learning

- Save your model weights at checkpoints
- Log metrics and results
- Track all code and data changes
  - update params
  - queueing experiments
  - comparing many experiments
  - persisting experiments
  - sharing experiments

```
$ dvc exp show --no-timestamp \
               --include-params train.n_est,train.min_split
```

Experiment	avg_prec	roc_auc	train.n_est	train.min_split
workspace	0.56191	0.93345	50	2
master	0.55259	0.91536	50	2
├── exp-bfe64	0.57833	0.95555	50	8
├── exp-b8082	0.59806	0.95287	50	64
├── exp-c7250	0.58876	0.94524	100	2
├── exp-b9cd4	0.57953	0.95732	100	8
├── exp-98a96	0.60405	0.9608	100	64
└── exp-ad5b1	0.56191	0.93345	50	2

Setting up a DVC pipeline: dvc stage add

Defines the checkpoint file or directory

```
dvc stage add --name train
              --deps data/MNIST --deps train.py \
              --checkpoints model.pt \
              --plots-no-cache predictions.json \
              --params seed,lr,weight_decay \
              --live dvclive \
              python train.py
```

Enables **DVCLive** logger, which helps to register checkpoints from your code.

Setting up a DVC pipeline: update dvc.yaml

```yaml
stages:
  train:
    cmd: python train.py
    deps:
      - data/MNIST
      - train.py
    params:
      - lr
      - seed
      - weight_decay
    outs:
      - model.pt:
          checkpoint: true
    live:
      dvclive:
        summary: true
          html: true
```

Or, update dvc.yaml accordingly

Registering checkpoints in Python code

```
import dvclive
...

# Iterate over training epochs.
for i in range(1, EPOCHS+1):
    # Train model
    ...
    # Save model
    torch.save(model.state_dict(), "model.pt")

    # Evaluate and checkpoint.
    metrics = evaluate(model, x_test, y_test)
    for k, v in metrics.items():
        print('Epoch %s: %s=%s'%(i, k, v))
        dvclive.log(name=k, val=v)
    dvclive.next_step()
```

Log metrics and save model for each training epoch

Running experiments

run command

```
$ dvc exp run
```

output

```
Epoch 1: loss=1.9428282976150513
Epoch 1: acc=0.5715
Generating lock file 'dvc.lock'
Updating lock file 'dvc.lock'
Checkpoint experiment iteration 'd99d81c'.
file:///Users/milecia/Repos/checkpoints-tutorial/dvclive.html

Epoch 2: loss=1.25374174118042
Epoch 2: acc=0.7738
...
```

DVC saves checkpoints and publishes metrics updates in dvclive.html file for each epochs

Running experiments & metrics tracking

run command

```
$ dvc exp show
```

Experiment	Created	step	acc	loss
workspace	-	4	0.92623	0.19567
main	01:58 PM	-	-	-
╓ bf81637 [exp-a1f53]	02:05 PM	4	0.92623	0.19567
╟ 9ca3fb8	02:04 PM	3	0.89344	0.27423
╟ a34ead1	02:03 PM	2	0.87295	0.29018
╟ ae382c7	02:02 PM	1	0.89754	0.26993
╙ a95260d	02:01 PM	0	0.73361	0.5271

Live code example

Setting up DVCLive and Checkpoints

Experimenting and collaboration with Studio

Live code example

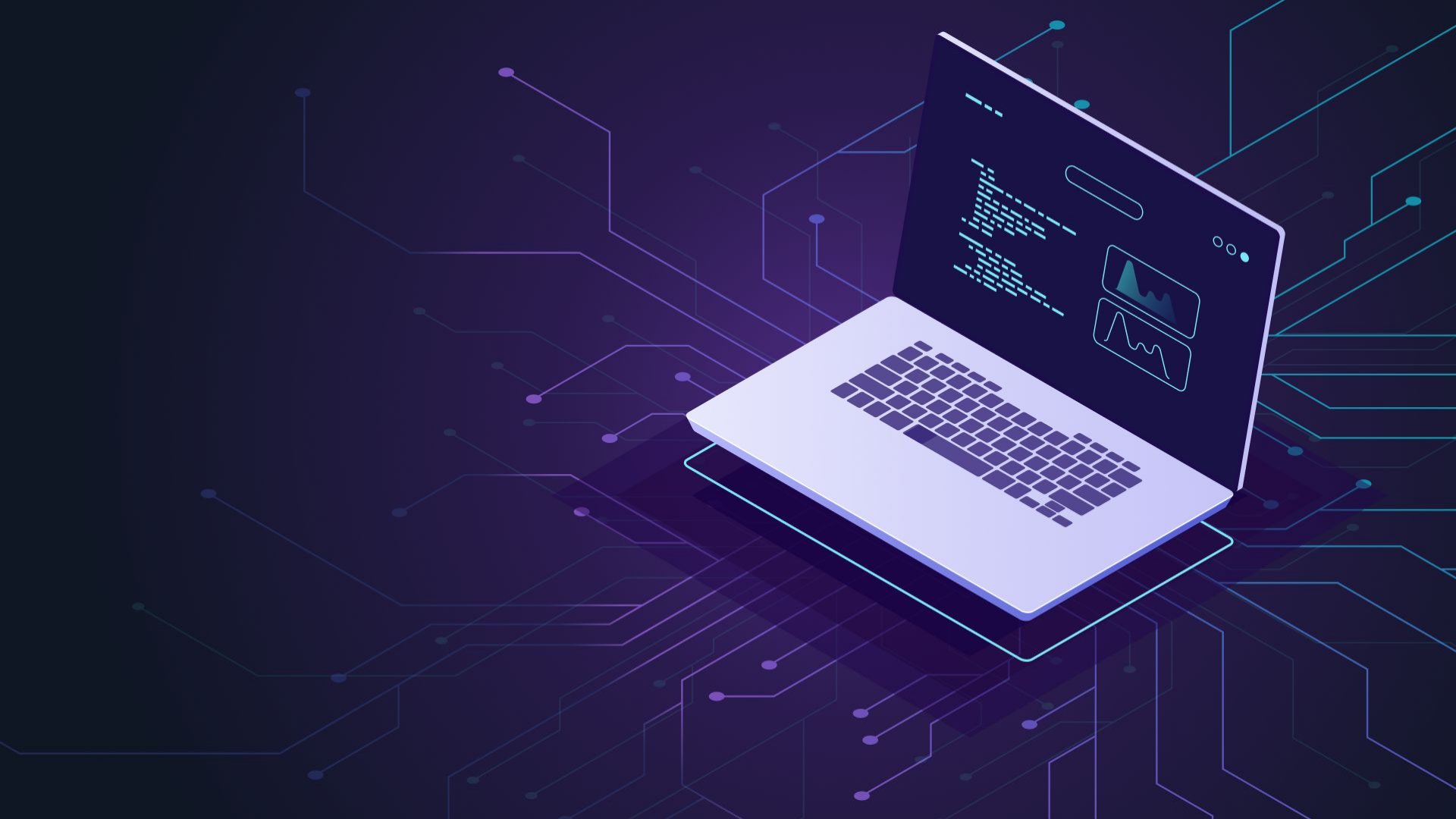
Experimenting and collaboration

with Studio

What have we learned?

What have we learned?

1. DVCLive and checkpoints in deep learning scenarios
2. Setting up checkpoints
3. Experimenting and collaboration



Links

- Data Science blueprint
  https://data-science-blueprint.readthedocs.io/en/latest/presentation/schema.html
- Post: Using Experiments for Transfer Learning
  https://dvc.org/blog/transfer-learning-experiments
- Post: Adding Data to Build a More Generic Model
  https://dvc.org/blog/adding-data-to-build-a-more-generic-model
- Docs: Checkpoints
  https://dvc.org/doc/user-guide/experiment-management/checkpoints