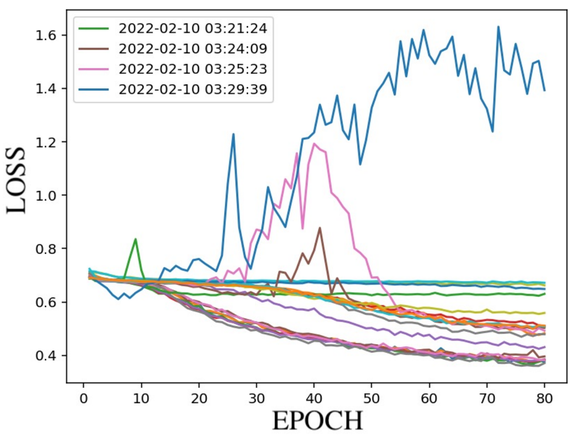
FLORDB: Retroactive Query Evaluation for Iterative AI/ML

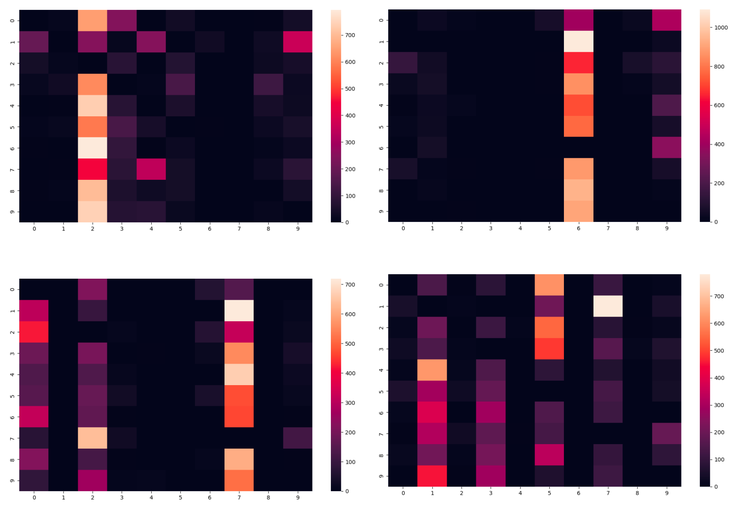
Rolando Garcia, Anusha Dandamudi, Gabriel Matute,
Joseph Gonzalez, Koushik Sen, Joseph Hellerstein

rogarcia@berkeley.edu

EPIC DATA lab
UC Berkeley

Iterative ML & High Velocity Experimentation





	runid	tstamp	cf_matrix	hidden_size	lr	batch_size	accuracy
0	firstRun	2023-03-16 19:05:52	NaN	500	0.001	100	NaN
30	smallerModel	2023-03-16 19:14:29	NaN	250	0.001	100	NaN
60	smallerModel	2023-03-16 19:15:07	NaN	125	0.001	100	NaN
90	increaseLR	2023-03-16 19:16:23	NaN	125	0.01	100	NaN
120	smallerBatchSize	2023-03-16 19:17:16	NaN	125	0.01	64	NaN
165	EpicDemo	2023-04-13 03:33:16	/home/rogarcia/.flor/ml_tutorial_flor.shadow/d...	NaN	NaN	NaN	97.83

	runid	tstamp	cf_matrix	hidden_size	lr	batch_size	accuracy
0	firstRun	2023-03-16 19:05:52	/home/rogarcia/.flor/ml_tutorial_flor.shadow.c...	500	0.001	100	10.52
30	smallerModel	2023-03-16 19:14:29	/home/rogarcia/.flor/ml_tutorial_flor.shadow.c...	250	0.001	100	6.52
60	smallerModel	2023-03-16 19:15:07	/home/rogarcia/.flor/ml_tutorial_flor.shadow.c...	125	0.001	100	10.62
90	increaseLR	2023-03-16 19:16:23	/home/rogarcia/.flor/ml_tutorial_flor.shadow.c...	125	0.01	100	6.4
120	smallerBatchSize	2023-03-16 19:17:16	/home/rogarcia/.flor/ml_tutorial_flor.shadow.c...	125	0.01	64	8.62
165	EpicDemo	2023-04-13 03:33:16	/home/rogarcia/.flor/ml_tutorial_flor.shadow/d...	NaN	NaN	NaN	97.83

Retroactive Query Evaluation

```
git checkout -b flor.shadow
```

```
Switched to a new branch 'flor.shadow'
```

```
python main.py --flor EpicDemo
```

```
...
Epoch [5/5], Step [500/600], Loss: 0.0208
Epoch [5/5], Step [600/600], Loss: 0.0550
Flor wrote log records locally.
```

	runid	tstamp	cf_matrix	hidden_size	lr	batch_size	accuracy
0	firstRun	2023-03-16 19:05:52	NaN	500	0.001	100	NaN
30	smallerModel	2023-03-16 19:14:29	NaN	250	0.001	100	NaN
60	smallerModel	2023-03-16 19:15:07	NaN	125	0.001	100	NaN
90	increaseLR	2023-03-16 19:16:23	NaN	125	0.01	100	NaN
120	smallerBatchSize	2023-03-16 19:17:16	NaN	125	0.01	64	NaN
165	EpicDemo	2023-04-13 03:33:16	/home/rogarcia/.flor/ml_tutorial_flor.shadow/d...	NaN	NaN	NaN	97.83

```
flor.replay(['cf_matrix'], 'cf_matrix.isna()')
```

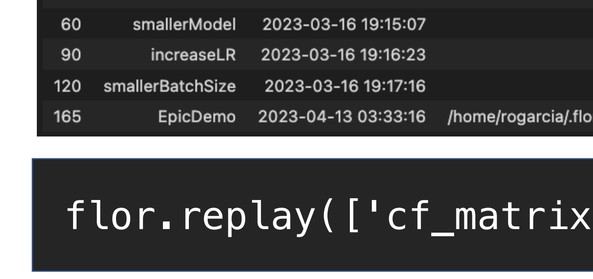
	projid	runid	tstamp	vid	seconds
0	ml_tutorial_flor.shadow.compressed	firstRun	2023-03-16 19:05:52	adef6a3ecb5b81729be132bffb1b77cafcbabbf0	1.592322
1	ml_tutorial_flor.shadow.compressed	smallerModel	2023-03-16 19:14:29	fd588e17c39bac20dfd7612a3a6ddba8b8d32cb3	1.589842
2	ml_tutorial_flor.shadow.compressed	smallerModel	2023-03-16 19:15:07	80b771bd0ae66538a6fa4d149a09ab34361d540b	1.585685
3	ml_tutorial_flor.shadow.compressed	increaseLR	2023-03-16 19:16:23	e273531a28bb6f16fa24b54056e23e7c1b9cdc9d	1.577927
4	ml_tutorial_flor.shadow.compressed	smallerBatchSize	2023-03-16 19:17:16	ae0355650b4c358baa113c1c75821c778091ee9e	1.582226

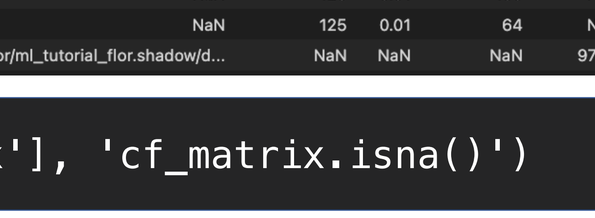
Flor Instrumentation

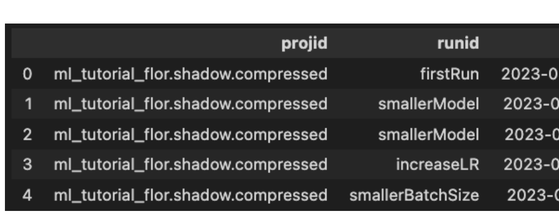
```
from flor import MTK as Flor
import torch

trainloader: torch.utils.data.DataLoader
testloader:  torch.utils.data.DataLoader
optimizer:   torch.optim.Optimizer
net:         torch.nn.Module
criterion:   torch.nn._Loss

Flor.checkpoints(net, optimizer)
for epoch in Flor.loop(range(...)):
    for data in Flor.loop(trainloader):
        inputs, labels = data
        optimizer.zero_grad()
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
    eval(net, testloader)
```

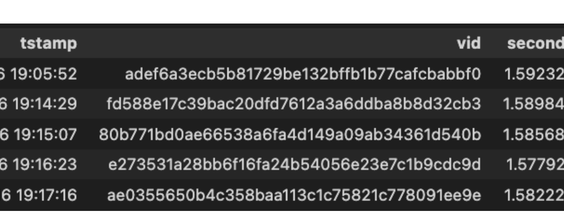
Evaluation

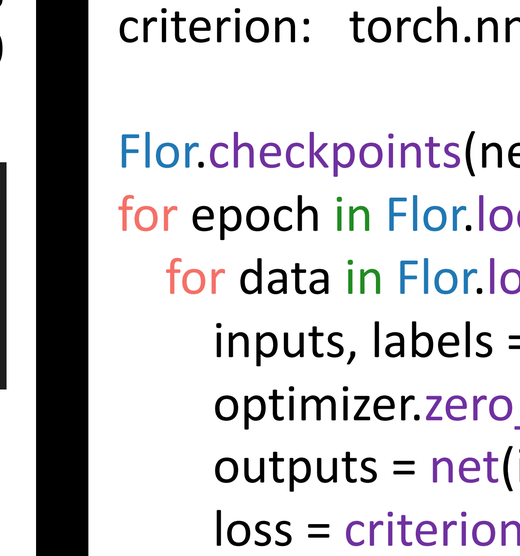
Model	Model Size	Data	Data Size	Objective	Evaluation	Application
ResNet-152	242 MB	ImageNet-1k	156 GB	image classification	accuracy	computer vision
BERT	440 MB	Wikipedia	40.8 GB	masked language modeling	accuracy	natural language processing
GPT-2	548 MB	WebText	40 GB	text generation	perplexity	natural language processing
LayoutLMv3	501 MB	FUNSD	20.8 MB	form understanding	F1-score	document intelligence
Yolos-tiny	26.1 MB	COCO	18 GB	object detection	avg precision	computer vision
Clip-VIT-large	1.71 GB	YFCC100M	15 GB	multimodal embedding	accuracy	computer vision
TAPAS-base	443 MB	Wikipedia	40.8 GB	table question answering	accuracy	document intelligence
CartPole-v0	142 KB	-	-	proximal policy optimization	reward	reinforcement learning

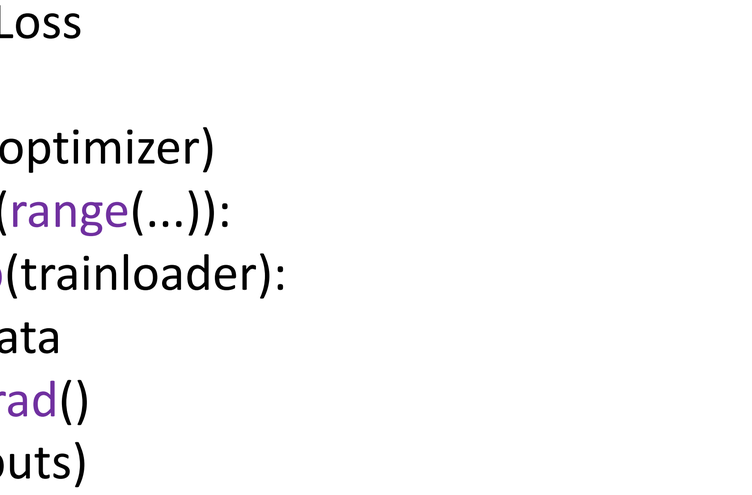












github.com/ucbrise/flor
pip install flordb