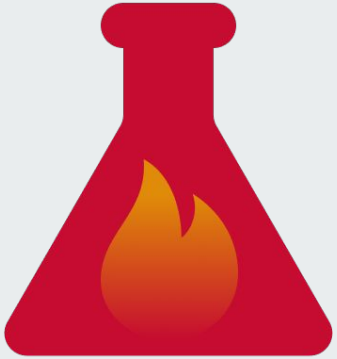
Deep dive into Catalyst

Tezikov Roman
ML-REPA #2



What is Catalyst?

High-level utils for PyTorch DL & RL research. Developed with focus on reproducibility, fast experimentation and code/ideas reusing.

- Train/inference loop
- Configuration files for model/data hyperparameters
- Easy customization
- Predefined DL pipelines

github.com/catalyst-team/catalyst

Typical train-loop

```
for epoch in epochs:
    for data_source in data_sources:
        for batch in batches:
            output = model(batch)
            metrics = fn(batch, output)
            optimize(metrics)
```

Reproducibility first

```python
from catalyst.utils import set_global_seed
set_global_seed(seed=42)

# or even
set_global_seed(seed=42)
prepare_cudnn(deterministic=True)
```

```python
def set_global_seed(seed: int) -> None:
    """Sets random seed into PyTorch, TensorFlow, Numpy and Random...."""

    try:
        import torch
    except ImportError:
        pass
    else:
        torch.manual_seed(seed)
        torch.cuda.manual_seed_all(seed)
    try:
        import tensorflow as tf
    except ImportError:
        pass
    else:
        tf.set_random_seed(seed)
    random.seed(seed)
    np.random.seed(seed)
```

```python
def prepare_cudnn(
    deterministic: bool = None,
    benchmark: bool = None
) -> None:
    """ ... """
    if torch.cuda.is_available():
        # CuDNN reproducibility
        # https://pytorch.org/docs/stable/notes/randomness.html#cudnn
        if deterministic is None:
            deterministic = \
                os.environ.get("CUDNN_DETERMINISTIC", "False") == "True"
        cudnn.deterministic = deterministic

        # https://discuss.pytorch.org/t/how-should-i-disable-using-cudnn-in-m
        if benchmark is None:
            benchmark = os.environ.get("CUDNN_BENCHMARK", "True") == "True"
        cudnn.benchmark = benchmark
```

Deterministic CuDNN

cudnnPoolingBackward

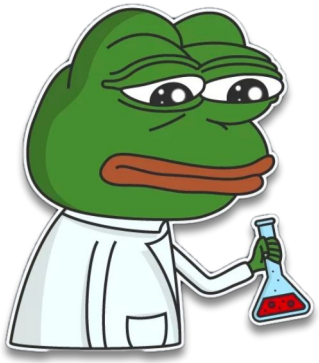
cudnnConvolutionBackwardFilter

cudnnConvolutionBackwardData

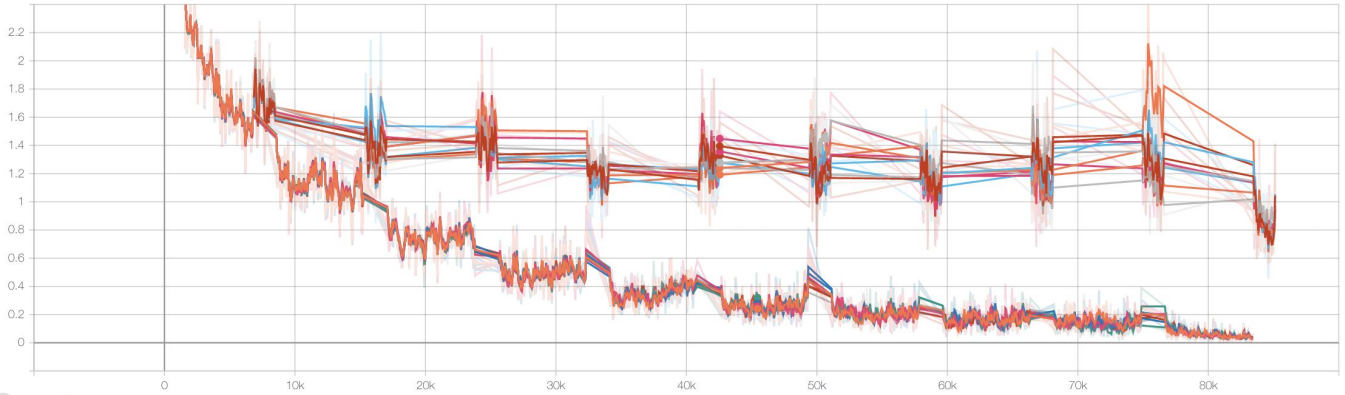
Deterministic CuDNN

- Same config and seed
- Several runs (10)

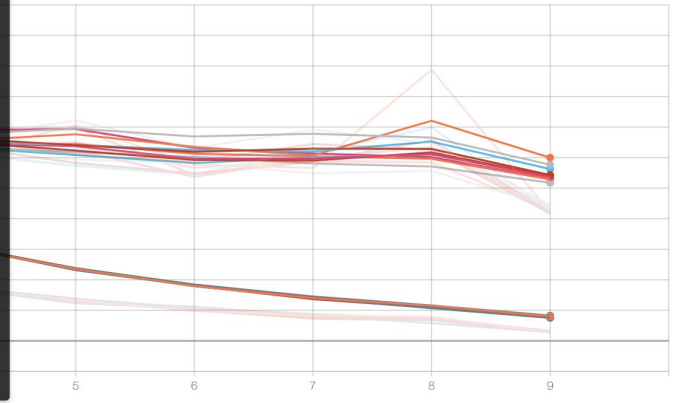
```
deterministic=False
```



batch
tag: loss/batch

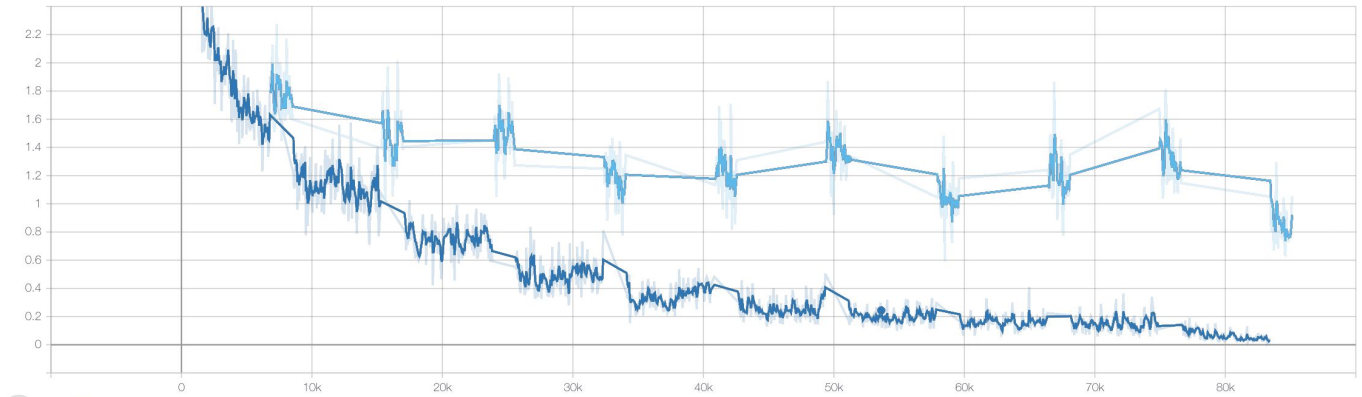


Name	Smoothed	Value	Step	Time	Relative
classification_tutorial_BENCHMARK_0/train_log	0.2412	0.1693	45.44k	Wed Aug 14, 14:42:29	3m 28s
classification_tutorial_BENCHMARK_0/valid_log	1.395	1.563	42.56k	Wed Aug 14, 14:42:14	2m 43s
classification_tutorial_BENCHMARK_1/train_log	0.2592	0.1867	45.44k	Wed Aug 14, 14:49:01	3m 28s
classification_tutorial_BENCHMARK_1/valid_log	1.277	1.328	42.56k	Wed Aug 14, 14:48:47	2m 43s
classification_tutorial_BENCHMARK_2/train_log	0.2775	0.2757	45.44k	Wed Aug 14, 14:55:34	3m 29s
classification_tutorial_BENCHMARK_2/valid_log	1.277	1.447	42.56k	Wed Aug 14, 14:55:20	2m 44s
classification_tutorial_BENCHMARK_3/train_log	0.3198	0.2583	45.44k	Wed Aug 14, 15:02:09	3m 30s
classification_tutorial_BENCHMARK_3/valid_log	1.24	1.199	42.56k	Wed Aug 14, 15:01:55	2m 44s
classification_tutorial_BENCHMARK_4/train_log	0.3065	0.2683	45.44k	Wed Aug 14, 15:08:44	3m 29s
classification_tutorial_BENCHMARK_4/valid_log	1.357	1.487	42.56k	Wed Aug 14, 15:08:30	2m 44s
classification_tutorial_BENCHMARK_5/train_log	0.2702	0.1959	45.44k	Wed Aug 14, 15:15:18	3m 28s
classification_tutorial_BENCHMARK_5/valid_log	1.328	1.448	42.56k	Wed Aug 14, 15:15:04	2m 43s
classification_tutorial_BENCHMARK_6/train_log	0.2429	0.1962	45.44k	Wed Aug 14, 15:21:51	3m 28s
classification_tutorial_BENCHMARK_6/valid_log	1.239	1.31	42.56k	Wed Aug 14, 15:21:37	2m 43s
classification_tutorial_BENCHMARK_7/train_log	0.2471	0.2004	45.44k	Wed Aug 14, 15:28:24	3m 28s
classification_tutorial_BENCHMARK_7/valid_log	1.277	1.38	42.56k	Wed Aug 14, 15:28:10	2m 43s
classification_tutorial_BENCHMARK_8/train_log	0.2583	0.2237	45.44k	Wed Aug 14, 15:35:00	3m 30s
classification_tutorial_BENCHMARK_8/valid_log	1.192	1.177	42.56k	Wed Aug 14, 15:34:45	2m 45s
classification_tutorial_BENCHMARK_9/train_log	0.2792	0.2533	45.44k	Wed Aug 14, 15:41:35	3m 29s
classification_tutorial_BENCHMARK_9/valid_log	1.448	1.641	42.56k	Wed Aug 14, 15:41:21	2m 44s

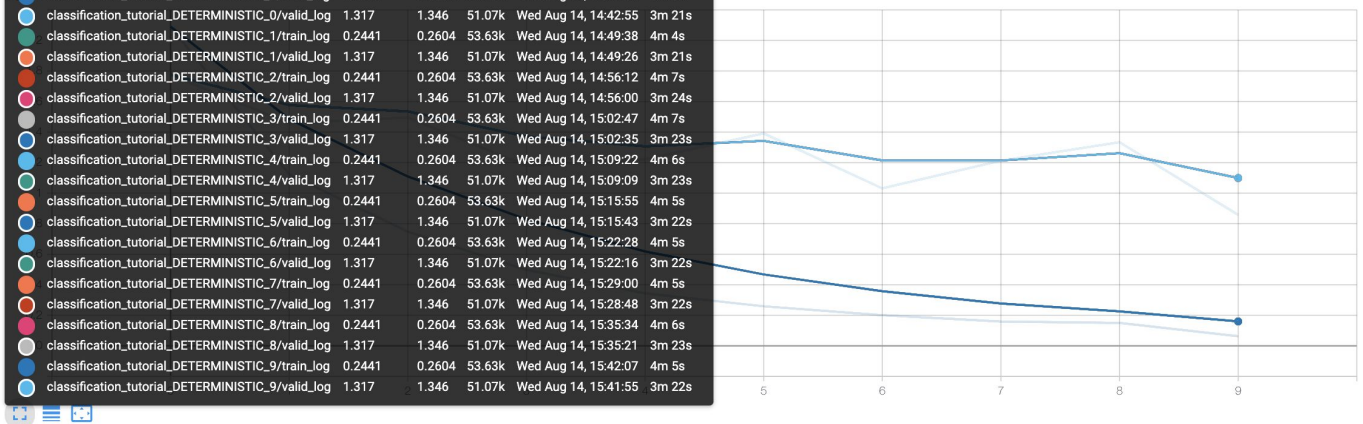


```
deterministic=True
```

batch
tag: loss/batch



Name	Smoothed	Value	Step	Time	Relative
classification_tutorial_DETERMINISTIC_0/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 14:43:07	4m 4s
classification_tutorial_DETERMINISTIC_0/valid_log	1.317	1.346	51.07k	Wed Aug 14, 14:42:55	3m 21s
classification_tutorial_DETERMINISTIC_1/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 14:49:38	4m 4s
classification_tutorial_DETERMINISTIC_1/valid_log	1.317	1.346	51.07k	Wed Aug 14, 14:49:26	3m 21s
classification_tutorial_DETERMINISTIC_2/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 14:56:12	4m 7s
classification_tutorial_DETERMINISTIC_2/valid_log	1.317	1.346	51.07k	Wed Aug 14, 14:56:00	3m 24s
classification_tutorial_DETERMINISTIC_3/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:02:47	4m 7s
classification_tutorial_DETERMINISTIC_3/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:02:35	3m 23s
classification_tutorial_DETERMINISTIC_4/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:09:22	4m 6s
classification_tutorial_DETERMINISTIC_4/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:09:09	3m 23s
classification_tutorial_DETERMINISTIC_5/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:15:55	4m 5s
classification_tutorial_DETERMINISTIC_5/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:15:43	3m 22s
classification_tutorial_DETERMINISTIC_6/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:22:28	4m 5s
classification_tutorial_DETERMINISTIC_6/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:22:16	3m 22s
classification_tutorial_DETERMINISTIC_7/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:29:00	4m 5s
classification_tutorial_DETERMINISTIC_7/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:28:48	3m 22s
classification_tutorial_DETERMINISTIC_8/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:35:34	4m 6s
classification_tutorial_DETERMINISTIC_8/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:35:21	3m 23s
classification_tutorial_DETERMINISTIC_9/train_log	0.2441	0.2604	53.63k	Wed Aug 14, 15:42:07	4m 5s
classification_tutorial_DETERMINISTIC_9/valid_log	1.317	1.346	51.07k	Wed Aug 14, 15:41:55	3m 22s



Use case: image classification Ants vs. bees

github.com/catalyst-team/catalyst/blob/master/examples/notebooks/classification-tutorial.ipynb

colab.research.google.com/github/catalyst-team/catalyst/blob/master/examples/notebooks/classification-tutorial.ipynb

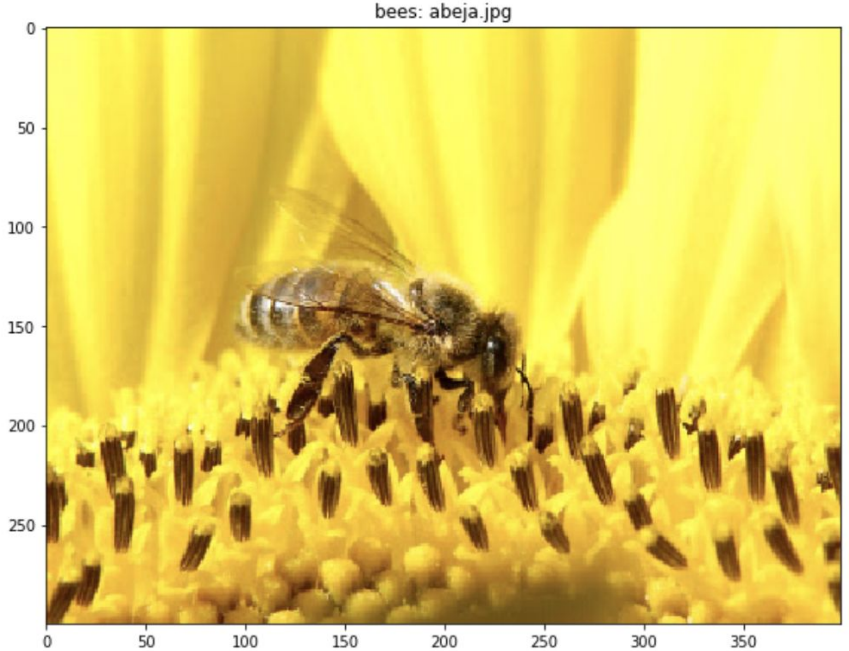
Dataset analysis & preprocessing

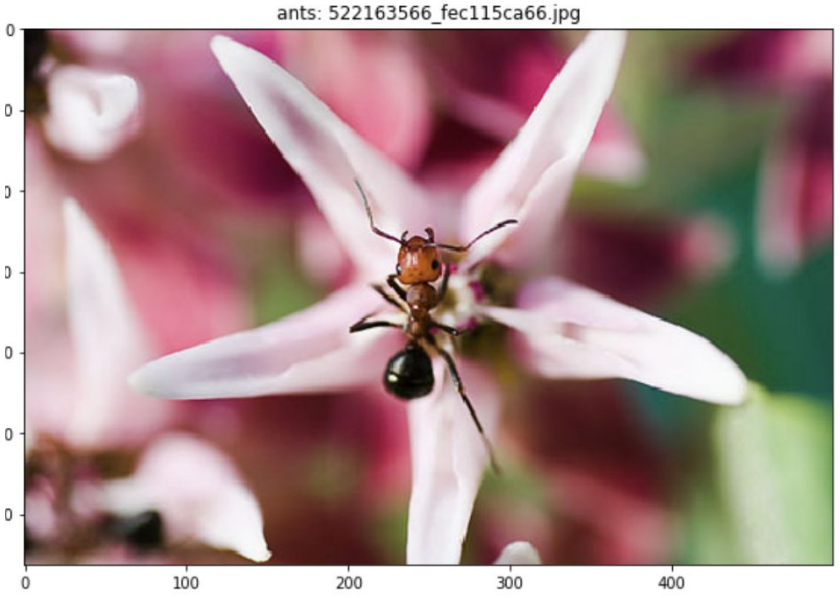
- ~400 images
- 2 classes

structure

```
dataset/
      ants/
            *.jpg
            ...
      bees/
            *.jpg
            ...
```


ants: 2255445811_dabcdf7258.jpg
0
50
00
50
00
50
00
0
100
200
300
400


bees: abeja.jpg
0
50
100
150
200
250
0
50
100
150
200
250
300
350


ants: 522163566_fec115ca66.jpg
0
100
200
300
400


bees: 353266603_d3eac7e9a0.jpg
0
50
100
150
200
250
300
350
0
100
200
300
400

Creating dataframe from data

```
from catalyst.utils.dataset import create_dataset, create_dataframe, prepare_dataset_labeling

dataset = create_dataset(dirs=f"{ROOT}/*", extension="*.jpg")
```

`create_dataset` function goes through a given directory and creates a dictionary `Dict[class_name, List[image]]`

```
df = create_dataframe(dataset, columns=["class", "filepath"])
df.head()
```

	class	filepath
0	ants	/home/tez_romach/Images/ants/0013035.jpg
1	ants	/home/tez_romach/Images/ants/1030023514_aad5c6...
2	ants	/home/tez_romach/Images/ants/10308379_1b6c72e1...
3	ants	/home/tez_romach/Images/ants/1053149811_f62a34...
4	ants	/home/tez_romach/Images/ants/1073564163_225a64...

finally `prepare_dataset_labeling` creates a numerical label for each unique class name

```
tag_to_label = prepare_dataset_labeling(df, "class")
tag_to_label
```

```
{'ants': 0, 'bees': 1}
```

Let's add a column with a numerical label value to the DataFrame. It can be easily done with `map_dataframe` function.

```
from catalyst.utils.pandas import map_dataframe

df_with_labels = map_dataframe(df, tag_column="class", class_column="label", tag2class=tag_to_label, verbose=True)
df_with_labels.head()
```

```
100% 396/396 [00:00<00:00, 16364.31it/s]
```

	class	filepath	label
0	ants	/home/tez_romach/Images/ants/0013035.jpg	0
1	ants	/home/tez_romach/Images/ants/1030023514_aad5c6...	0
2	ants	/home/tez_romach/Images/ants/10308379_1b6c72e1...	0
3	ants	/home/tez_romach/Images/ants/1053149811_f62a34...	0
4	ants	/home/tez_romach/Images/ants/1073564163_225a64...	0

Catalyst begins

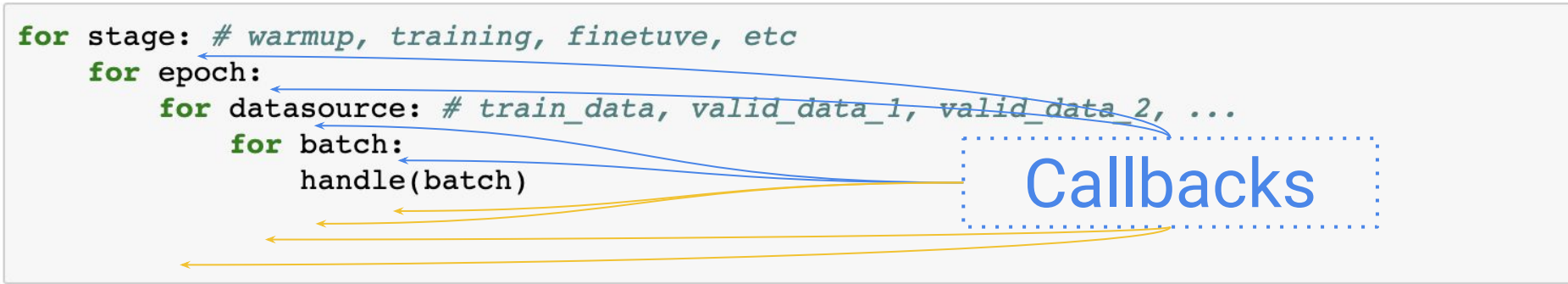
Why should I write this one more time?

Most common DL tasks do the same actions in the train-loop. The difference between them is in the Loss and Metrics.

```
for stage: # warmup, training, finetuve, etc
    for epoch:
        for datasource: # train_data, valid_data_1, valid_data_2, ...
            for batch:
                handle(batch)
```

Why should I write this one more time?

Most common DL tasks do the same actions in the train-loop. The difference between them is in the Loss and Metrics.



Callbacks

```
on_stage_start
     on_epoch_start  # one epoch - one run of every loader
          on_loader_start
               on_batch_start
               # ...
               on_batch_end
          on_loader_end
     on_epoch_end
on_stage_end

on_exception  # if an Exception was raised
```

Callbacks

List of standard callbacks here includes callbacks such as

- `CheckpointCallback`. Saves N best models to logdir
- `TensorboardLogger`. Logs all metrics to tensorboard
- `EarlyStoppingCallback`. Early training stop if metrics do not improve for the patience of epochs
- `ConfusionMatrixCallback`. Plots ConfusionMatrix per epoch in tensorboard

and many other callbacks, like `LRFinder` and `MixupCallback`

Callbacks for metrics

Many different metrics for classification

- `AccuracyCallback`
- `MapKCallback`
- `AUCCallback`
- `F1ScoreCallback`

- `ConfusionMatrixCallback`

and segmentation

- `DiceCallback`
- `IouCallback`
- `LovaszCallback`

Runner

An abstraction that contains logic **HOW** to run an experiment.

There is `SupervisedRunner` for standard tasks

```
from catalyst.dl.runner import SupervisedRunner

runner = SupervisedRunner()
```

```python
def train(
    self,
    model: _Model,
    criterion: _Criterion,
    optimizer: _Optimizer,
    loaders: "OrderedDict[str, DataLoader]",
    logdir: str,
    callbacks: "Union[List[Callback], Ordere
    scheduler: _Scheduler = None,
    num_epochs: int = 1,
    valid_loader: str = "valid",
    main_metric: str = "loss",
    minimize_metric: bool = True,
    verbose: bool = False,
    state_kwargs: Dict = None,
    checkpoint_data: Dict = None,
    fp16: Union[Dict, bool] = None,
    check: bool = False,
): ...
```

```python
def infer(
    self,
    model: _Model,
    loaders: "OrderedDict[str, DataLoader]",
    callbacks: "Union[List[Callback], Ordered
    verbose: bool = False,
    state_kwargs: Dict = None,
    fp16: Union[Dict, bool] = None,
    check: bool = False,
): ...

def predict_loader(
    self,
    loader: DataLoader,
    resume: str = None,
    verbose: bool = False,
    state_kwargs: Dict = None,
    fp16: Union[Dict, bool] = None,
    check: bool = False,
): ...
```

Model training

Function `runner.train` starts training process with specified model/criterion/optimizer/scheduler

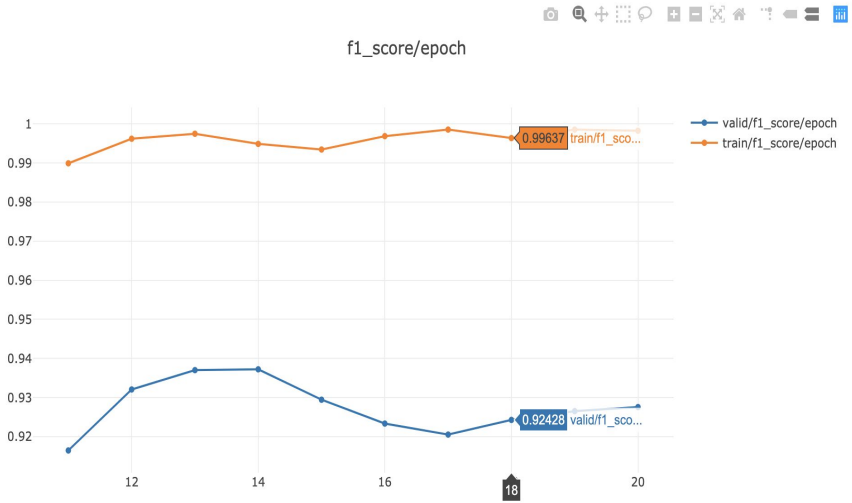
```
runner.train(
    model=model,
    logdir=logdir,
    criterion=criterion,
    optimizer=optimizer,
    scheduler=scheduler,
    loaders=loaders,

    callbacks=[
        AccuracyCallback(num_classes=num_classes),
        AUCCallback(
            num_classes=num_classes,
            input_key="targets_one_hot",
            class_names=class_names
        ),
        F1ScoreCallback(
            input_key="targets_one_hot",
            activation="Softmax"
        ),
        ConfusionMatrixCallback(
            num_classes=num_classes,
            class_names=class_names
        )
    ],
    num_epochs=NUM_EPOCHS,
    verbose=True
)
```

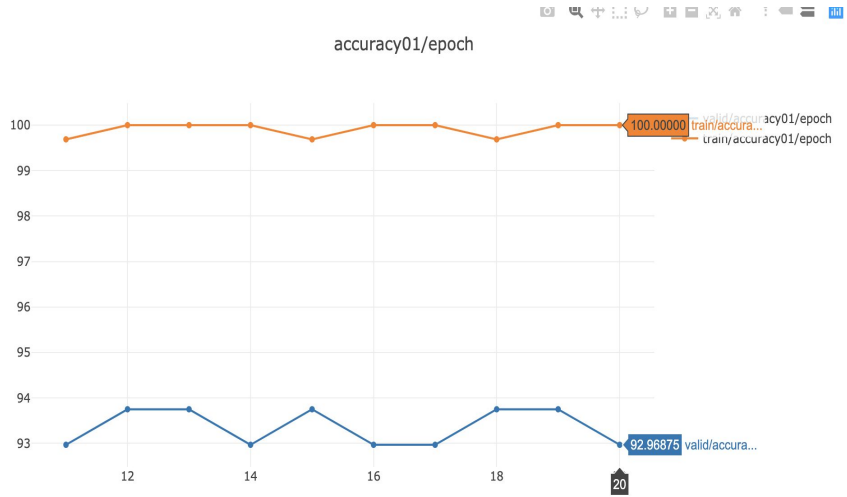
Training logs

```
[2019-08-14 03:14:53,134]
9/10 * Epoch 9 (train): _base/lr=2.700000E-05 | _base/momentum=0.9000 | _timers/_fps=865.5324 | _timers/batch_time=0.
2496 | _timers/data_time=0.1982 | _timers/model_time=0.0513 | accuracy01=99.6667 | auc/_mean=0.9998 | auc/class_ants=
0.9997 | auc/class_bees=0.9999 | f1_score=0.9918 | loss=0.0140
9/10 * Epoch 9 (valid): _base/lr=2.700000E-05 | _base/momentum=0.9000 | _timers/_fps=4147.2678 | _timers/batch_time=
0.3738 | _timers/data_time=0.3637 | _timers/model_time=0.0100 | accuracy01=88.2812 | auc/_mean=0.9813 | auc/class_ant
s=0.9800 | auc/class_bees=0.9826 | f1_score=0.8910 | loss=0.3229
```

```
Top best models:
logs/classification_tutorial_0/checkpoints//train.0.pth 0.1310
logs/classification_tutorial_0/checkpoints//train.6.pth 0.2501
logs/classification_tutorial_0/checkpoints//train.7.pth 0.2679
```

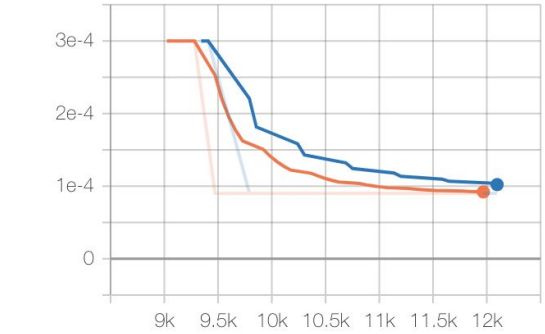
Notebook logs



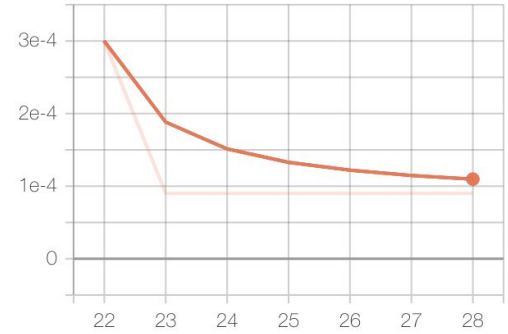


_base 4

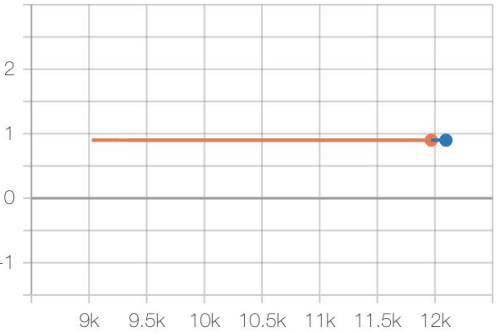
lr/batch
tag: _base/lr/batch



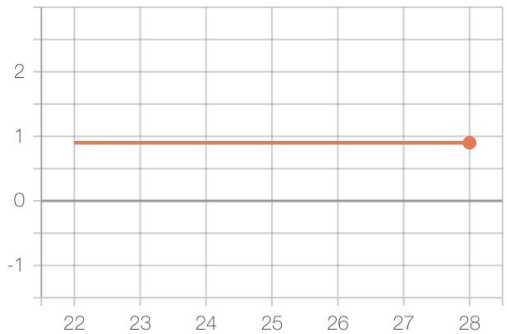
lr/epoch
tag: _base/lr/epoch



momentum/batch
tag: _base/momentum/batch



momentum/epoch
tag: _base/momentum/epoch



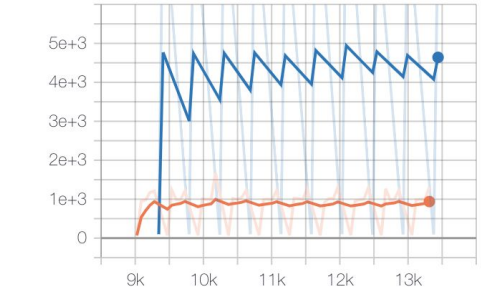
Runs

Write a regex to filter r...

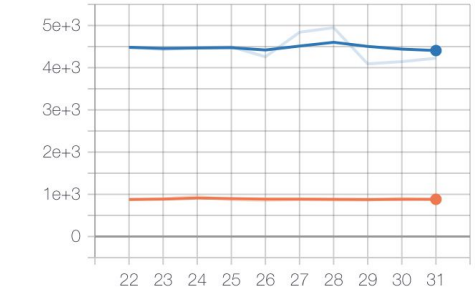
classification_tutorial/train_log

classification_tutorial/valid_log

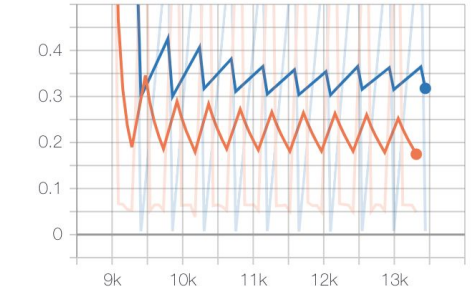
_fps/batch
tag: _timers/_fps/batch



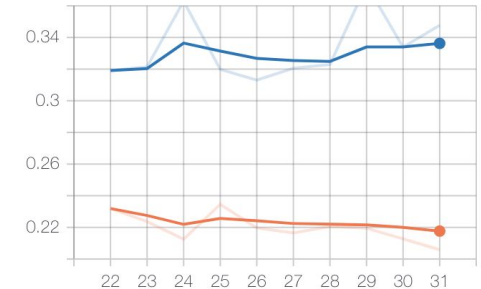
_fps/epoch
tag: _timers/_fps/epoch



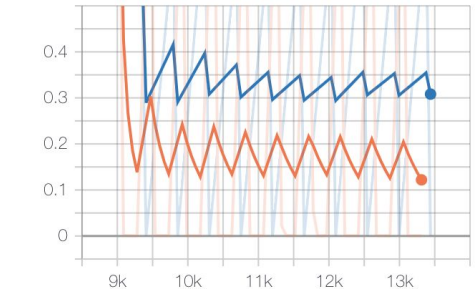
batch_time/batch
tag: _timers/batch_time/batch



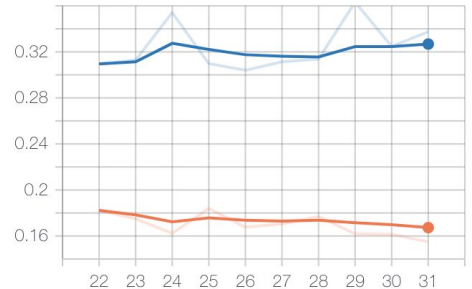
batch_time/epoch
tag: _timers/batch_time/epoch



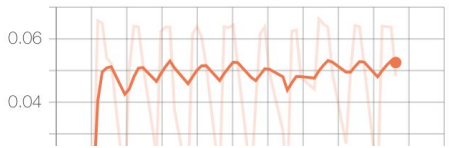
data_time/batch
tag: _timers/data_time/batch



data_time/epoch
tag: _timers/data_time/epoch



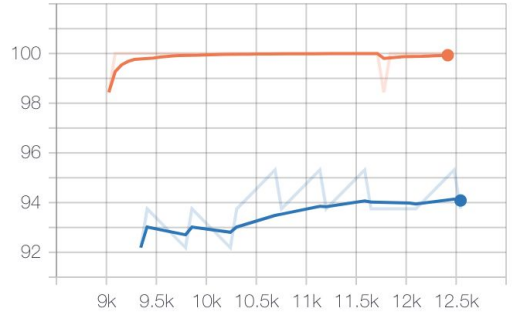
model_time/batch
tag: _timers/model_time/batch



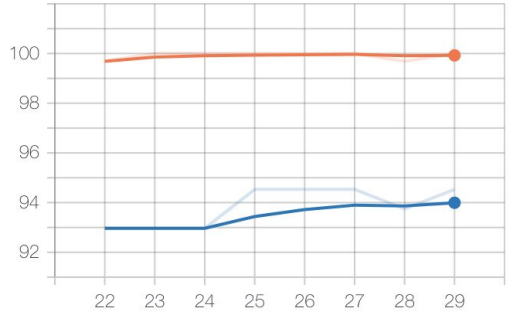
model_time/epoch
tag: _timers/model_time/epoch

accuracy01 2

batch
tag: accuracy01/batch

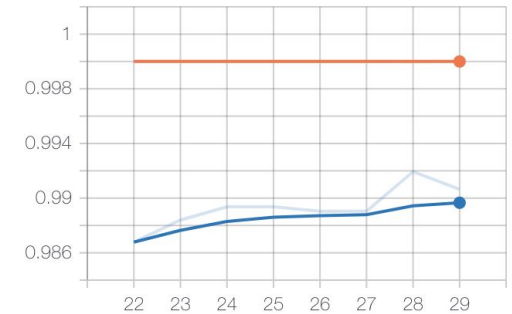


epoch
tag: accuracy01/epoch

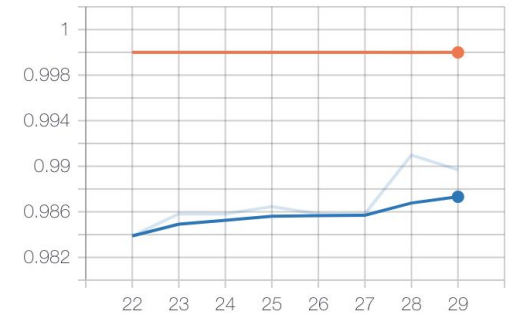


auc 3

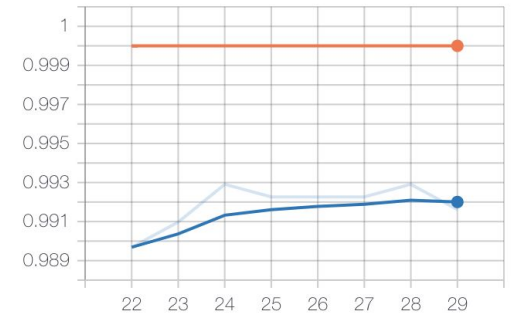
_mean/epoch
tag: auc/_mean/epoch



class_ants/epoch
tag: auc/class_ants/epoch

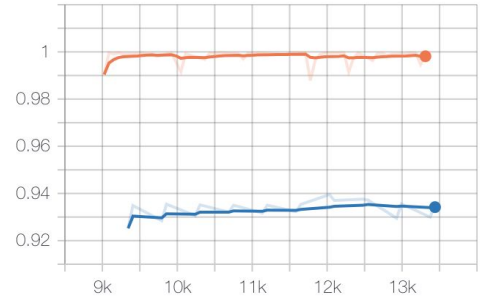


class_bees/epoch
tag: auc/class_bees/epoch

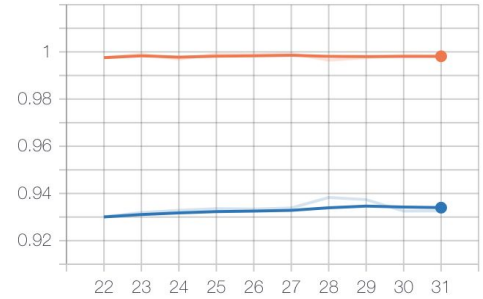


f1_score 2

batch
tag: f1_score/batch

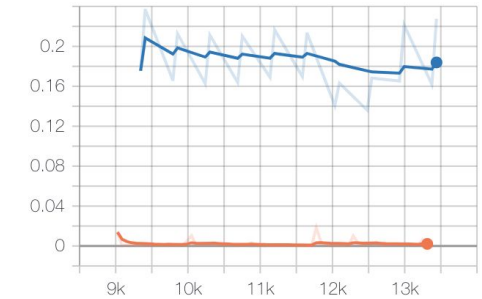


epoch
tag: f1_score/epoch

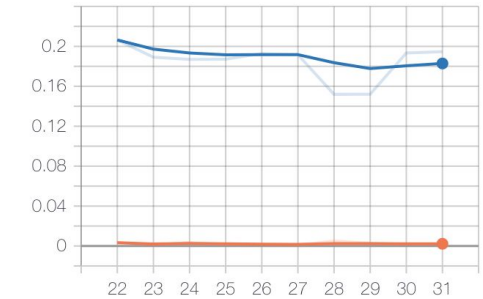


loss 2

batch
tag: loss/batch

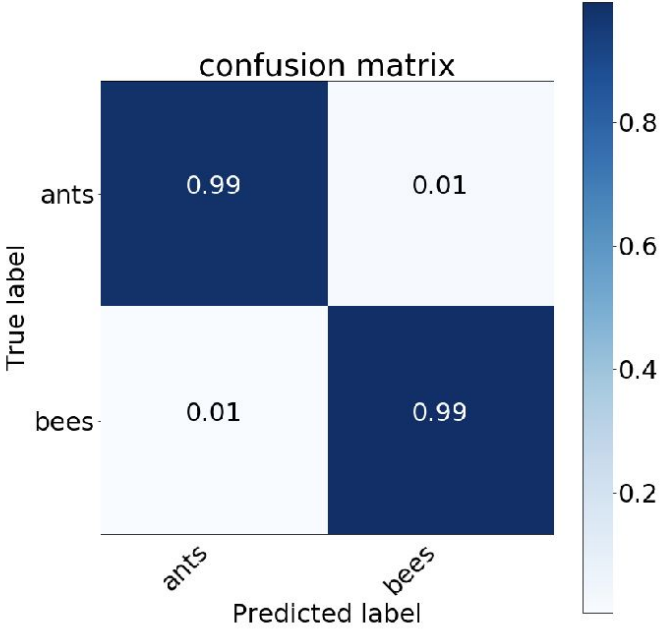


epoch
tag: loss/epoch

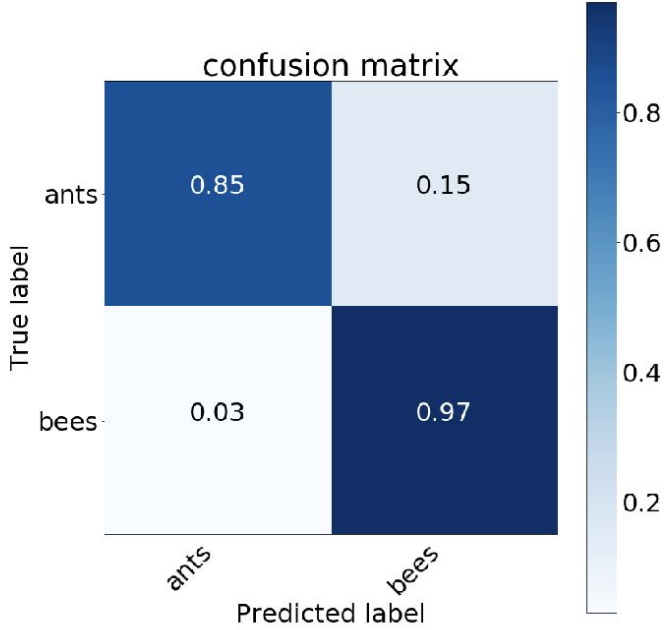


confusion_matrix

confusion_matrix/epoch
classification_tutorial/train_log
step **9**
Thu Aug 15 2019 10:38:39 GMT+0300 (Москва, стандартное время)



confusion_matrix/epoch
classification_tutorial/valid_log
step **9**
Thu Aug 15 2019 10:38:40 GMT+0300 (Москва, стандартное время)



Model Inference

```
predictions = runner.predict_loader(
    loaders["valid"], resume=f"{logdir}/checkpoints/best.pth", verbose=True
)
```

```
=> loading checkpoint ./logs/classification_tutorial/checkpoints/best.pth
loaded checkpoint ./logs/classification_tutorial/checkpoints/best.pth (epoch 2)

0/1 * Epoch (infer):   0% 0/2 [00:00<?, ?it/s]
0/1 * Epoch (infer):   0% 0/2 [00:05<?, ?it/s, _timers/_fps=12.294]
0/1 * Epoch (infer):  50% 1/2 [00:05<00:05,  5.21s/it, _timers/_fps=12.294]
0/1 * Epoch (infer):  50% 1/2 [00:06<00:05,  5.21s/it, _timers/_fps=52.531]
0/1 * Epoch (infer): 100% 2/2 [00:06<00:00,  4.01s/it, _timers/_fps=52.531]Top best models:
```

The resulting object has shape = (number of elements in the loader, output shape from the model)

```
print("loader", len(loaders["valid"].dataset))
print("predictions", predictions.shape)
```

```
loader 80
predictions (80, 2)
```

Classification pipeline

Config API

```
# in experiment.py
from catalyst.dl import ConfigExperiment

class Experiment(ConfigExperiment):
    ...
```

```
# in __init__.py
from catalyst.dl import SupervisedRunner as Runner
from .experiment import Experiment
from .model import SomeModel  # you choose
```

+ YAML config

Writing Experiment

```python
class Experiment(ConfigExperiment):
    @staticmethod
    def get_transforms(stage: str = None, mode: str = None): ...

    def get_datasets(self, stage: str, **kwargs): ...
```

Registry

Storage for various factories by name and make them visible through YAML config

```
from catalyst.dl import registry

registry.Callback(SomeCallback)
registry.Model(MyModel)
registry.Criterion(GreatLoss)
registry.Optimizer(NewAwesomeOptimizer)
registry.Scheduler(EvenScheduler)
```

Model

Common `torch.nn.Module` with registry

```python
@registry.Model
class SimpleNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 16 * 5 * 5)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Model

Common `torch.nn.Module`
with registry

You can also register in `__init__.py`

```
from catalyst.dl import SupervisedRunner as Runner
from .experiment import Experiment
from .model import SimpleModel
from catalyst.dl import registry

registry.Model(SimpleModel)
```

```
class SimpleNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 16 * 5 * 5)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

YAML config

Contains info about

- Model and model params
- Logdir and experiment directory
- Training stages
  - Criterion
  - Optimizer
  - Scheduler
  - Callbacks
  - Data params

```yaml
model_params:
  model: SimpleNet

args:
  expdir: "expdir"
  logdir: "./logs/cifar"

stages:
  data_params:
    batch_size: 64
    num_workers: 1

  state_params:
    num_epochs: 2
    main_metric: &reduce_metric accuracy01
    minimize_metric: False

  criterion_params:
    criterion: CrossEntropyLoss

  scheduler_params:
    scheduler: MultiStepLR
    milestones: [10]
    gamma: 0.3

  callbacks_params:
    loss:
      callback: CriterionCallback
    optimizer:
      callback: OptimizerCallback
    accuracy:
      callback: AccuracyCallback
```

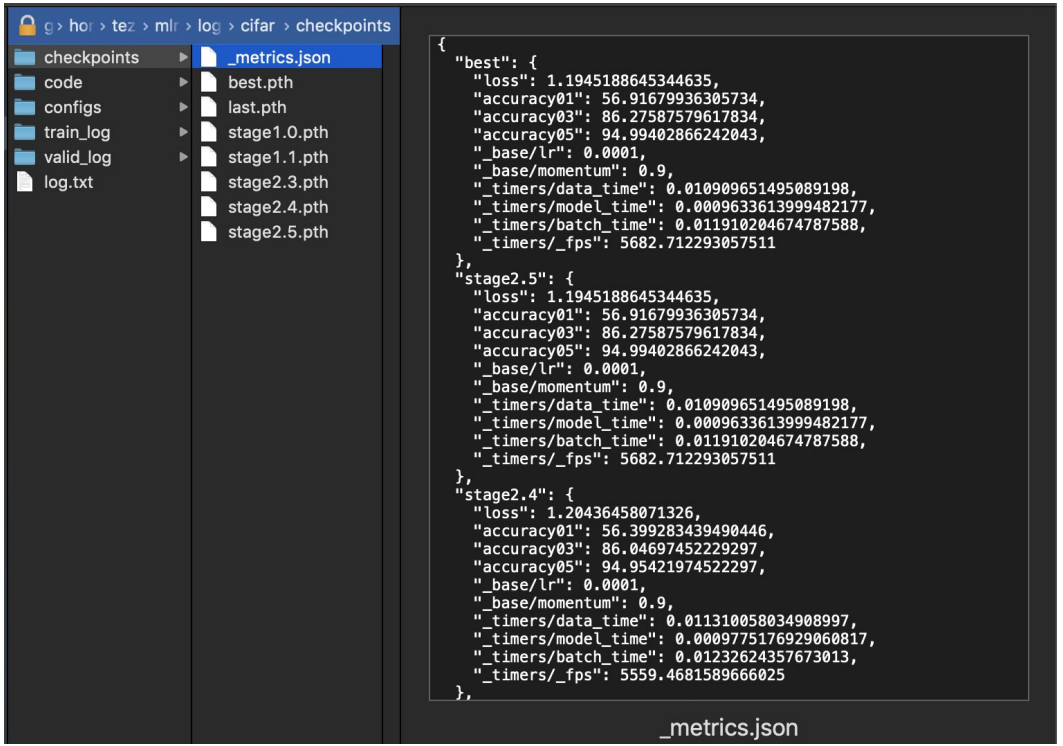
Easy run

```
catalyst-dl run --config ./config.yml --verbose
```

```
Files already downloaded and verified
Files already downloaded and verified
0/2 * Epoch (train): 100% 782/782 [00:12<00:00, 61.69it/s, _timers/_fps=11740.529, accuracy01=50.000, accuracy03=68.750, accuracy05=81.250, loss=1.824]
0/2 * Epoch (valid): 100% 157/157 [00:02<00:00, 60.66it/s, _timers/_fps=20216.558, accuracy01=37.500, accuracy03=68.750, accuracy05=87.500, loss=1.781]
[2019-08-15 10:46:47,212]
0/2 * Epoch 0 (train): _base/lr=0.0010 | _base/momentum=0.9000 | _timers/_fps=6110.9260 | _timers/batch_time=0.0110 | _timers/data_time=0.0100 | _timers/model_time=0.0009 | accuracy01=40.4492 | accuracy
03=74.3426 | accuracy05=88.6149 | loss=1.6314
0/2 * Epoch 0 (valid): _base/lr=0.0010 | _base/momentum=0.9000 | _timers/_fps=4845.5522 | _timers/batch_time=0.0141 | _timers/data_time=0.0131 | _timers/model_time=0.0009 | accuracy01=48.5470 | accuracy
03=80.5732 | accuracy05=92.2074 | loss=1.4308
1/2 * Epoch (train): 100% 782/782 [00:13<00:00, 59.75it/s, _timers/_fps=59718.678, accuracy01=37.500, accuracy03=75.000, accuracy05=87.500, loss=1.600]
1/2 * Epoch (valid): 100% 157/157 [00:02<00:00, 60.16it/s, _timers/_fps=9280.075, accuracy01=37.500, accuracy03=75.000, accuracy05=93.750, loss=1.530]
[2019-08-15 10:47:02,925]
1/2 * Epoch 1 (train): _base/lr=0.0010 | _base/momentum=0.9000 | _timers/_fps=5897.1136 | _timers/batch_time=0.0114 | _timers/data_time=0.0105 | _timers/model_time=0.0009 | accuracy01=51.7044 | accuracy
03=83.2261 | accuracy05=93.8859 | loss=1.3352
1/2 * Epoch 1 (valid): _base/lr=0.0010 | _base/momentum=0.9000 | _timers/_fps=4710.4297 | _timers/batch_time=0.0142 | _timers/data_time=0.0132 | _timers/model_time=0.0010 | accuracy01=53.6525 | accuracy
03=84.6636 | accuracy05=94.4467 | loss=1.2677
Top best models:
logs/cifar/checkpoints//stage1.1.pth    53.6525
logs/cifar/checkpoints//stage1.0.pth    48.5470
```

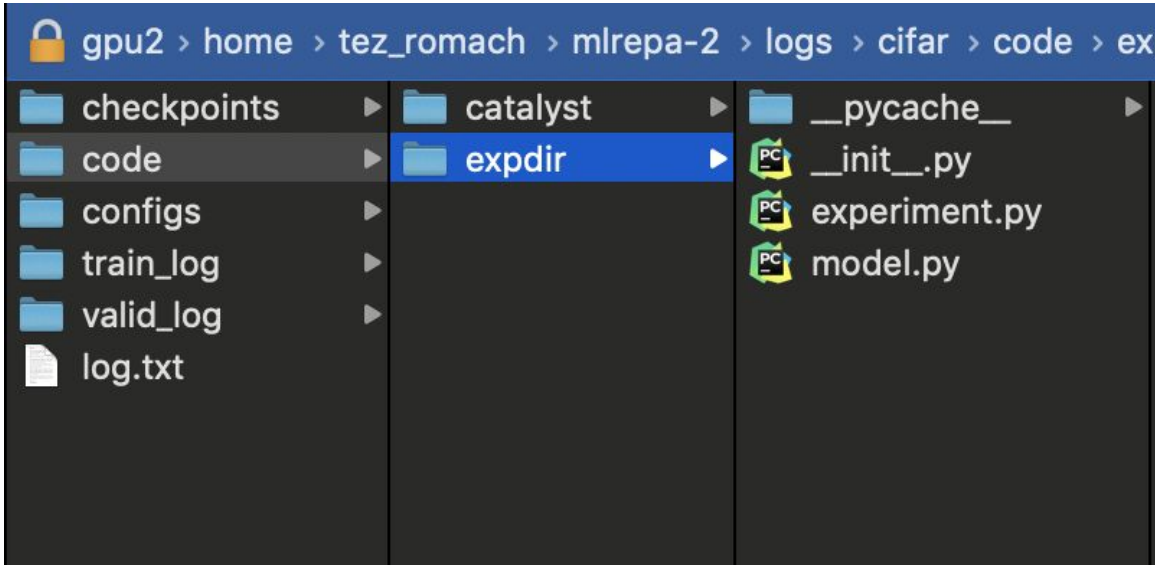
What's in LOGDIR

- Top best checkpoints with metrics



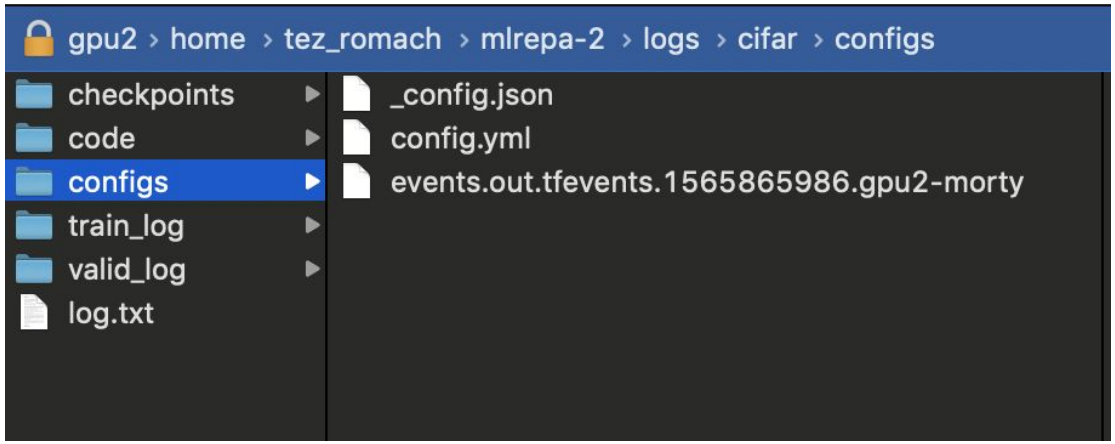
What's in LOGDIR

- Top best checkpoints with metrics
- Dumped code (experiment + catalyst)



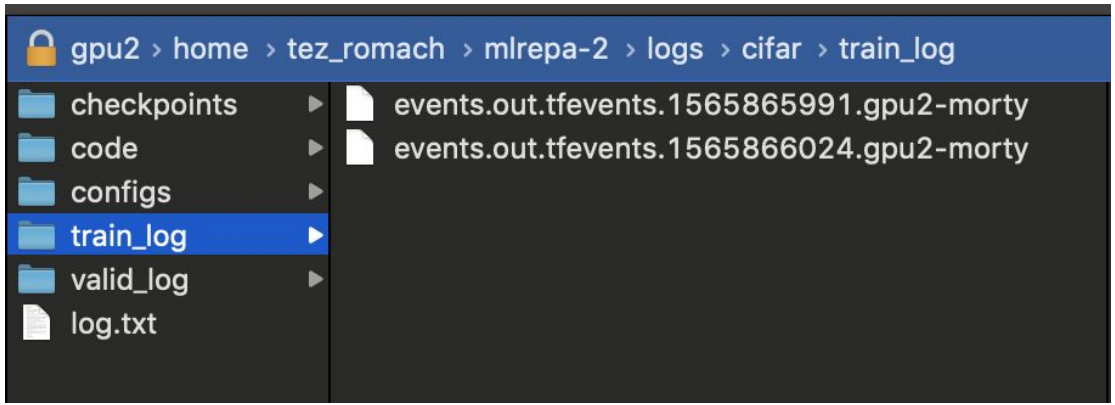
What's in LOGDIR

- Top best checkpoints with metrics
- Dumped code (experiment + catalyst)
- Configs



What's in LOGDIR

- Top best checkpoints with metrics
- Dumped code (experiment + catalyst)
- Configs
- Tensorboard logs



Easy grid-search with Config API

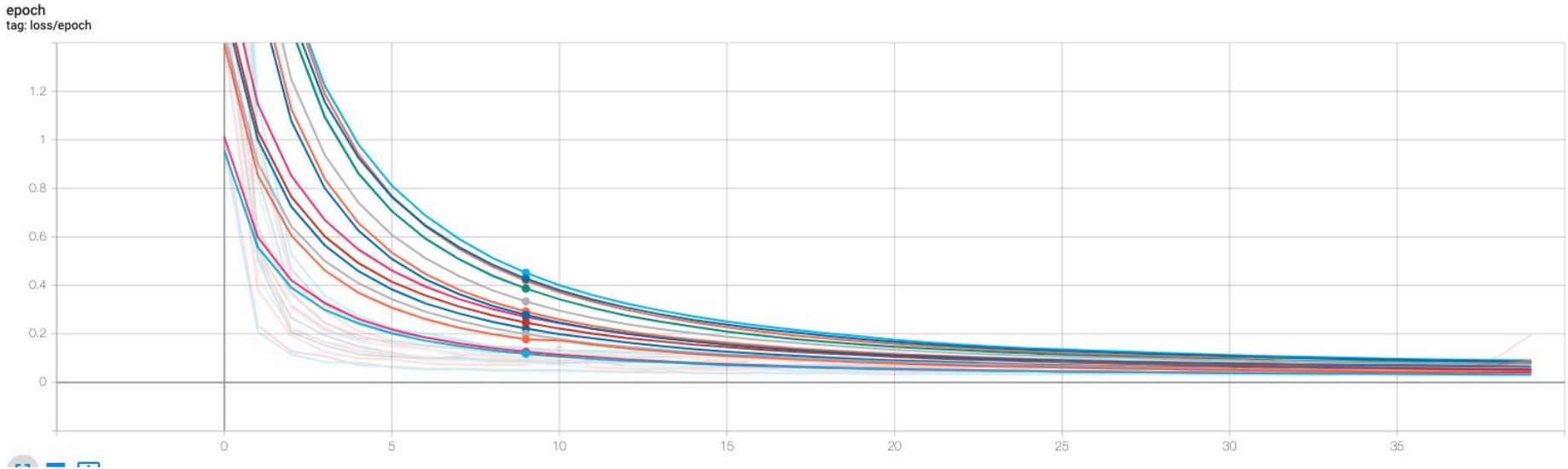
Create text file with runs
(for example `runs.sh`)

```
catalyst-dl run --config ./configs/config1.yml --verbose
catalyst-dl run --config ./configs/config2.yml --verbose
catalyst-dl run --config ./configs/config3.yml --verbose
catalyst-dl run --config ./configs/config4.yml --verbose
catalyst-dl run --config ./configs/config5.yml --verbose
catalyst-dl run --config ./configs/config6.yml --verbose
catalyst-dl run --config ./configs/config7.yml --verbose
catalyst-dl run --config ./configs/config8.yml --verbose
```

Run with gpu-parallel

```
cat ./runs.sh | catalyst-parallel-run 3 ./logs
```

Enjoy



Catalyst

github.com/catalyst-team/catalyst

pip install -U catalyst

#tool_catalyst

Thank you

telegram: @TezRomacH
ods: @tez.romach