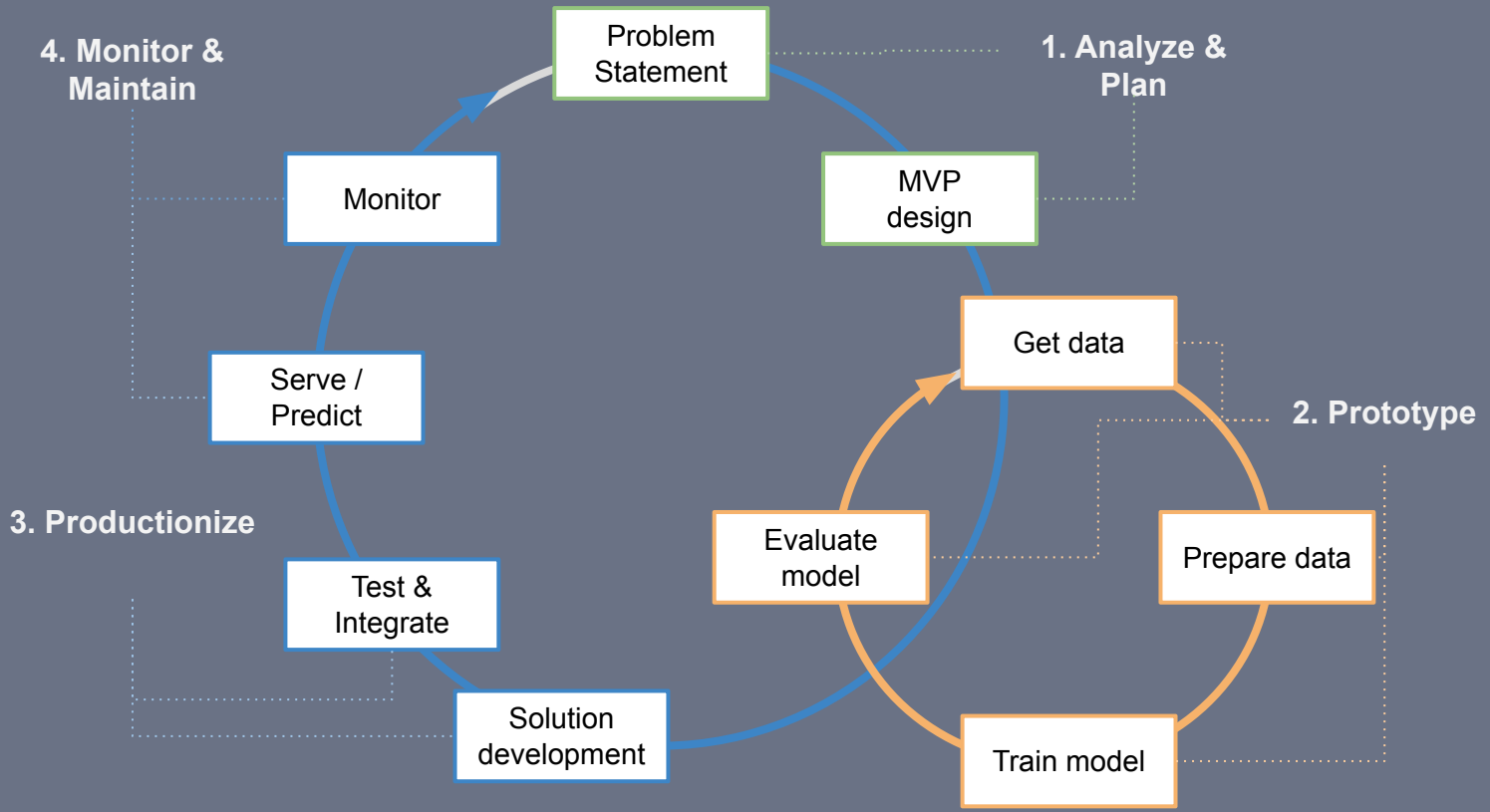
Reproducibility in CV

Overview

Tezikov Roman
REPA meetup #1

Why do I need it?



Specific problems

Running CUDA and cuDNN

Bitwise reproducibility is not guaranteed across cuDNN versions, as the implementation of a given routine may change.

Functions without guarantee of reproducibility:

- cudnnConvolutionBackwardFilter
- cudnnConvolutionBackwardData
- cudnnPoolingBackward
- cudnnSpatialTfSamplerBackward

Running CUDA and cuDNN

cudnnPoolingBackward

cudnnConvolutionBackwardFilter

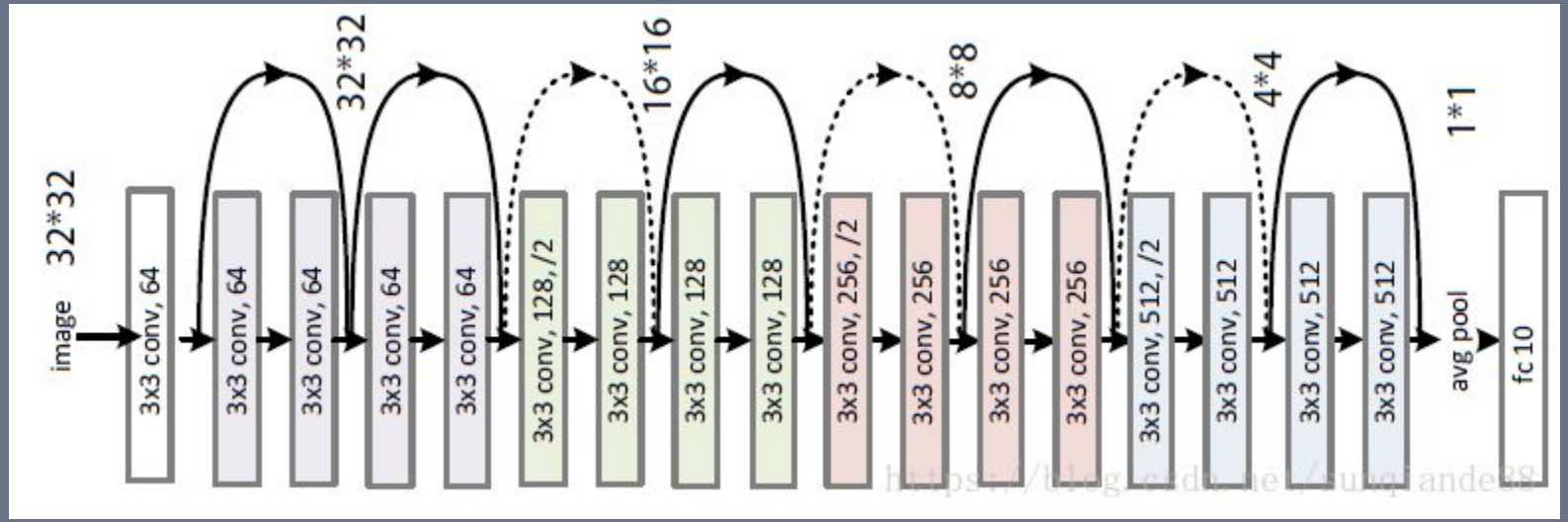
cudnnConvolutionBackwardData

Running CUDA and cuDNN

A number of operations have backwards that use `atomicAdd`

- CTCLoss
- Many forms of pooling, padding, and sampling

Inaccuracies accumulate with deeper networks



ResNet-18

Completely reproducible results are not guaranteed!

Is there any solution?

Make requirements.txt and Dockerfile with frozen dependencies

requirements.txt

```
albumentations==0.2.2
catalyst==19.6.1
imageio==2.5.0
numpy==1.16.3
opencv-python==4.1.0.25
PyYAML==5.1
safitty==1.2.0
tensorboardX==1.6
torch==1.1.0
torchvision==0.2.2.post3
tqdm==4.31.1
```

Make requirements.txt and Dockerfile with frozen dependencies

Dockerfile

```
FROM nvidia/cuda:10.1-devel-ubuntu18.04

# Set up locale to prevent bugs with encoding
ENV LANG=C.UTF-8 LC_ALL=C.UTF-8

RUN apt-get update && \
    apt-get install -y apt-utils && \
    apt-get install -y \
        wget \
        curl \
        python3 \
        python3-pip && \
    rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*

COPY ./requirements.txt /workspace/requirements.txt
RUN pip3 install -r /workspace/requirements.txt
```

Store your model & training params as YAML/JSON

```yaml
model_params:
  model: ResNeXt50_32x4d
args:
  logdir: "./logs/cifar_stages"

data_params:
  batch_size: 64
  num_workers: 1
state_params:
  num_epochs: 10
  main_metric: &reduce_metric accuracy01
  minimize_metric: False

optimizer_params:
  optimizer: Adam
  lr: 0.001
  weight_decay: 0.0001

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

Set a seed to all frameworks you use.

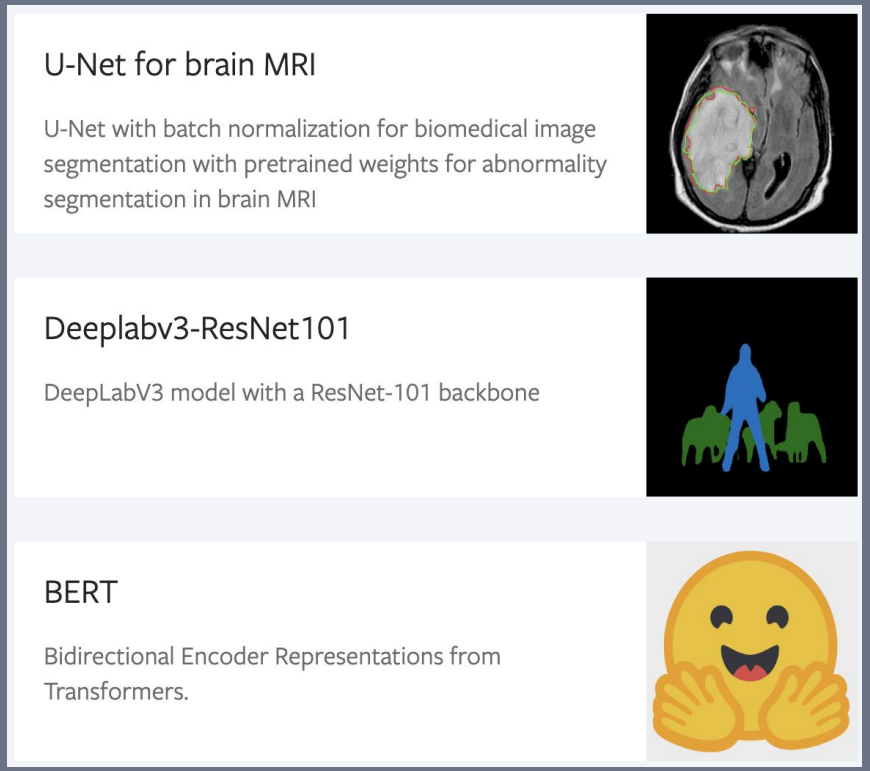
```python
def set_global_seed(seed: int) -> None:
    """
    Sets random seed into PyTorch, TensorFlow, Numpy and Random

    Args:
        seed: random seed
    """
    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)
```

Change cuDNN mode to deterministic

```
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
```

Pytorch Hub



- Easy to load and use models
- 20+ different models
- Integration with PapersWithCode

```
In[2]: import torch
In[3]: torch.hub.list('pytorch/vision')
Using cache found in /Users/romach/.cache/torch/hub/pytorch_vision_master
Out[3]:
['alexnet',
 'deeplabv3_resnet101',
 'densenet121',
 'densenet161',
 'densenet169',
 'densenet201',
 'fcn_resnet101',
 'googlenet',
 'inception_v3',
 'mobilenet_v2',
```

Frameworks

Common

DL

Teasers

REPA meetup #2

- **Владислав Грозин**
  «MLFlow и Sacred. Контроль экспериментов в DL»
- **Александр Бельских**
  Тема уточняется
- **Тезиков Роман**
  «Управление моделями с Catalyst. Workshop»

Questions?