Experiment reproduction

PyTorch BIVL²ab

Universidad Industrial de Santander

Biomedical Imaging, Vision and Learning Laboratory

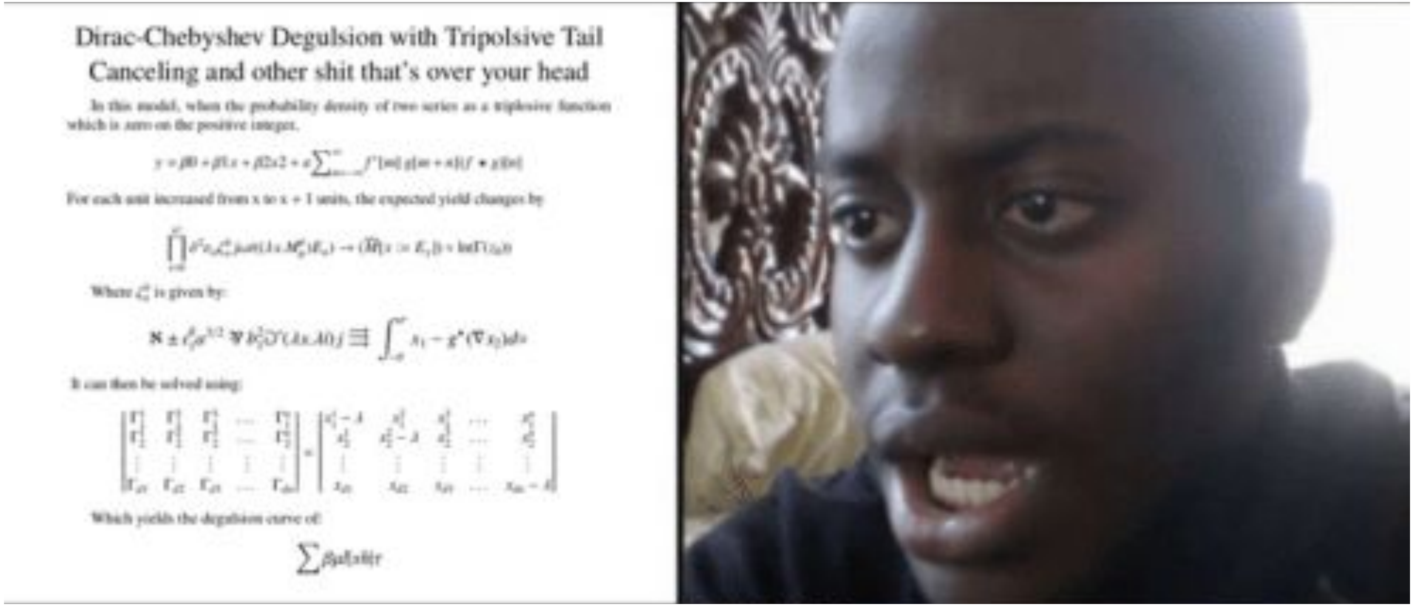
Agenda

- Why should I even care?
- Good practices for experiment reproducibility
- Demo

Why should I even care?

Inspiration

Inspiration



Inspiration

Inspiration

The most used **tools are easy to use and easy to access**.

PyTorch

- AlexNet
- ConvNeXt
- DenseNet
- EfficientNet
- EfficientNetV2
- GoogLeNet
- Inception V3
- MaxVit
- MNASNet
- MobileNet V2
- MobileNet V3
- RegNet
- ResNet
- ResNeXt
- ShuffleNet V2
- SqueezeNet
- SwinTransformer
- VGG
- VisionTransformer
- Wide ResNet

TensorFlow

There are lots of researchers using older architectures instead of newer ones just because there are plenty of implementations on the internet.

Inspiration

Even though you may find a paper excellent, you may not choose to use it just because the barriers preventing you from using it. Some of these barriers could be:

- No codebase
- Codebase, executes on your machine, dependencies failed → Environment not specified
- “We use 80% to train and 20% for test” → Dataset splits are not specified

Inspiration

Even though you may find a paper excellent, you may not choose to use it just because the barriers preventing you from using it. Some of these barriers could be:

- No codebase
- Codebase, executes on your machine, dependencies failed → Environment not specified
- “We use 80% to train and 20% for test” → Dataset splits are not specified

And it’s fine to not put everything on the paper, at the end, a scientific paper purpose is to present novel ideas. Nevertheless, nowadays exist many tools to record all the little details that go into creating your work.

Inspiration

Also, having things organized and being able to reproduce your results is cool

- Someone asks for your code? Share your github repo with clear instructions.
- Want to test your method on other datasets? Fine, create a dataloader and run the experiment.
- Your model weights were magically deleted? No problem, run the experiment again with the same configuration.

Good practices for experiment reproducibility

Examples

- ConvNeXt

Results and Pre-trained Models

ImageNet-1K trained models

name	resolution	acc@1	#params	FLOPs	model
ConvNeXt-T	224x224	82.1	28M	4.5G	model
ConvNeXt-S	224x224	83.1	50M	8.7G	model
ConvNeXt-B	224x224	83.8	89M	15.4G	model
ConvNeXt-B	384x384	85.1	89M	45.0G	model
ConvNeXt-L	224x224	84.3	198M	34.4G	model
ConvNeXt-L	384x384	85.5	198M	101.0G	model

Examples

- ConvNeXt

Installation

We provide installation instructions for ImageNet classification experiments here.

Dependency Setup

Create an new conda virtual environment

```
conda create -n convnext python=3.8 -y
conda activate convnext
```

Install Pytorch>=1.8.0, torchvision>=0.9.0 following official instructions. For example:

```
pip install torch==1.8.0+cu111 torchvision==0.9.0+cu111 -f https://download.pytorch.org/whl/torch_stable.html
```

Clone this repo and install required packages:

```
git clone https://github.com/facebookresearch/ConvNeXt
pip install timm==0.3.2 tensorboardX six
```

Examples

- ConvNeXt
- Vision Transformer

Installation

Make sure you have `Python>=3.6` installed on your machine.

Install JAX and python dependencies by running:

```
# If using GPU:
pip install -r vit_jax/requirements.txt

# If using TPU:
pip install -r vit_jax/requirements-tpu.txt
```

Examples

- ConvNeXt
- Vision Transformer

Fine-tuning a model

You can run fine-tuning of the downloaded model on your dataset of interest. All models share the same command line interface.

For example for fine-tuning a ViT-B/16 (pre-trained on imagenet21k) on CIFAR10 (note how we specify `b16,cifar10` as arguments to the config, and how we instruct the code to access the models directly from a GCS bucket instead of first downloading them into the local directory):

```
python -m vit_jax.main --workdir=/tmp/vit-$(date +%s) \
    --config=$(pwd)/vit_jax/configs/vit.py:b16,cifar10 \
    --config.pretrained_dir='gs://vit_models/imagenet21k'
```

Demo

Thank You!

Universidad Industrial de Santander

BIVL^{2}ab

Biomedical Imaging, Vision and Learning Laboratory