Deep Learning Toolkit *(Weights & Biases, HF Accelerate & other Useful Tools)*

Rowel Atienza, PhD
University of the Philippines
github.com/roatienza
2022

Outline

Environment, Code Editor

Python

Tensor libraries – numpy, einsum, einops

PyTorch, Timm

Hugging Face (HF), Gradio, Streamlit

W&B and HF Accelerate

git, GitHub, VMs

Weights and Biases (wandb)

https://wandb.ai/

Why wandb?

Track experiments

Dataset and model versioning

Visualization and sharing of results

Installing `wandb`

```
pip install wandb
```

Create an account at wandb.ai

Login and initialize `wandb`

```
wandb.login()
config = {
  "learning_rate": 0.1,
  "epochs": 100,
  "batch_size": 128,
  "dataset": "cifar10"
}
run = wandb.init(project="wandb-project",
                 entity="upeee",
                 config=config)
```

Watch the model gradients during training

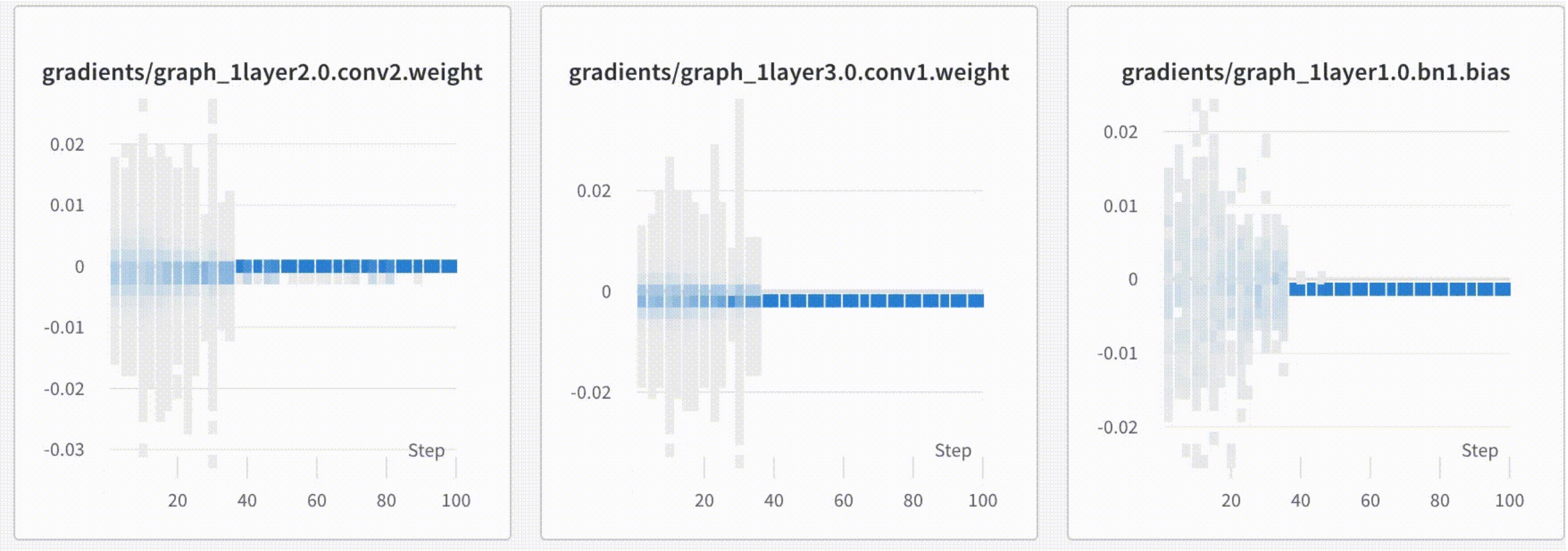
```
model =  torchvision.models.resnet18(pretrained=False,
                                     progress=True)
model.fc = torch.nn.Linear(model.fc.in_features, 10)


# watch model gradients during training
wandb.watch(model)
```

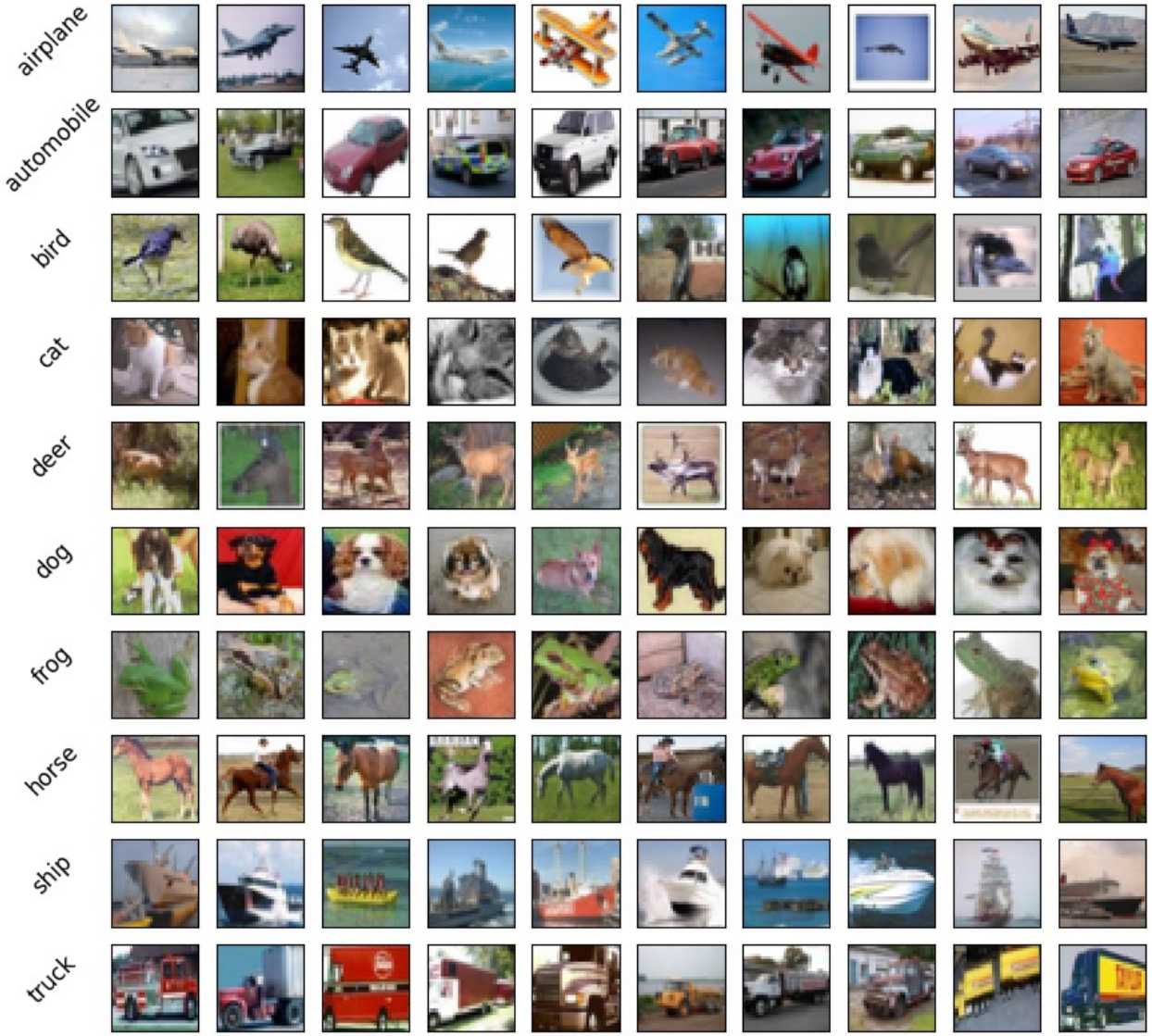
Replacing the Fully Connected (fc) Head

Original Network

New Network


gradients/graph_1layer2.0.conv2.weight
0.02
0.01
0
-0.01
-0.02
-0.03
Step
20
40
60
80
100
gradients/graph_1layer3.0.conv1.weight
0.02
0
-0.02
Step
20
40
60
80
100
gradients/graph_1layer1.0.bn1.bias
0.02
0.01
0
-0.01
-0.02
Step
20
40
60
80
100

CIFAR10 Dataset

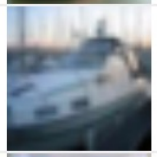
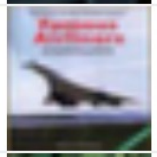
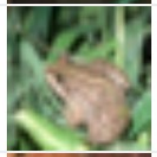
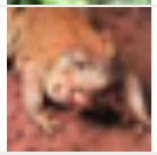


wanbd table for input viz

```
…
table_test = wandb.Table(columns=[
                         “Image”,
                         "Ground Truth",
                         "Initial Pred Label",])
…

for i in range(8):
  table_test.add_data(wandb.Image(image[i]),
  label_human[label[i]],
  label_human[pred[i]])
```

```
runs.summary["Test data"]
```

	Image	Ground Truth	Initial Pred Label	Final Pred Label
1		cat	horse	cat
2		ship	horse	ship
3		ship	horse	ship
4		airplane	horse	airplane
5		frog	horse	deer
6		frog	horse	frog

1 - 6 of 8

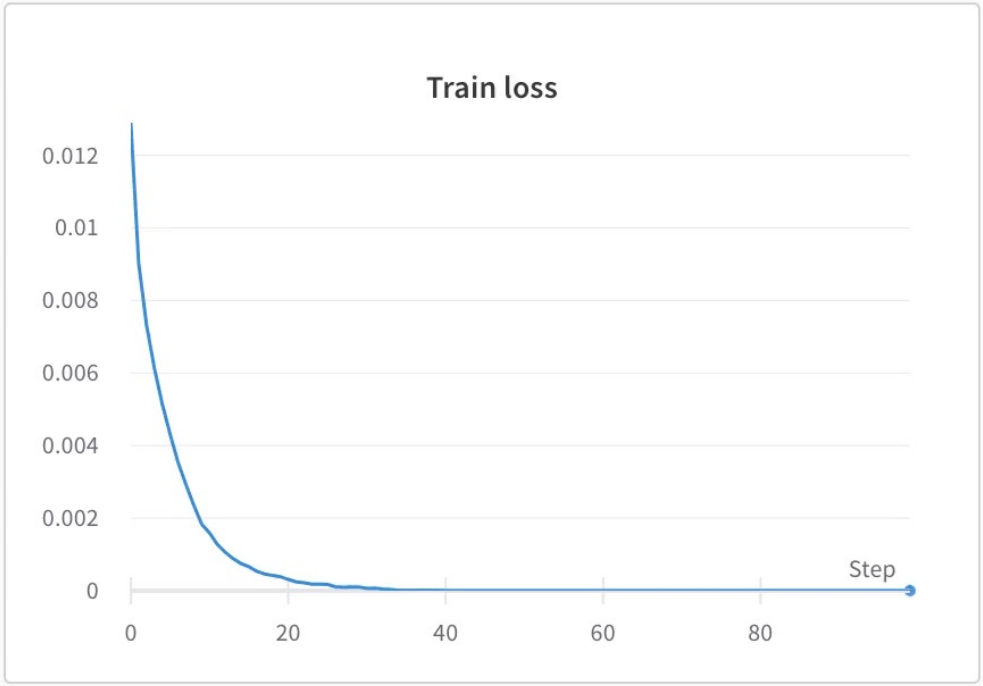
Reset Table

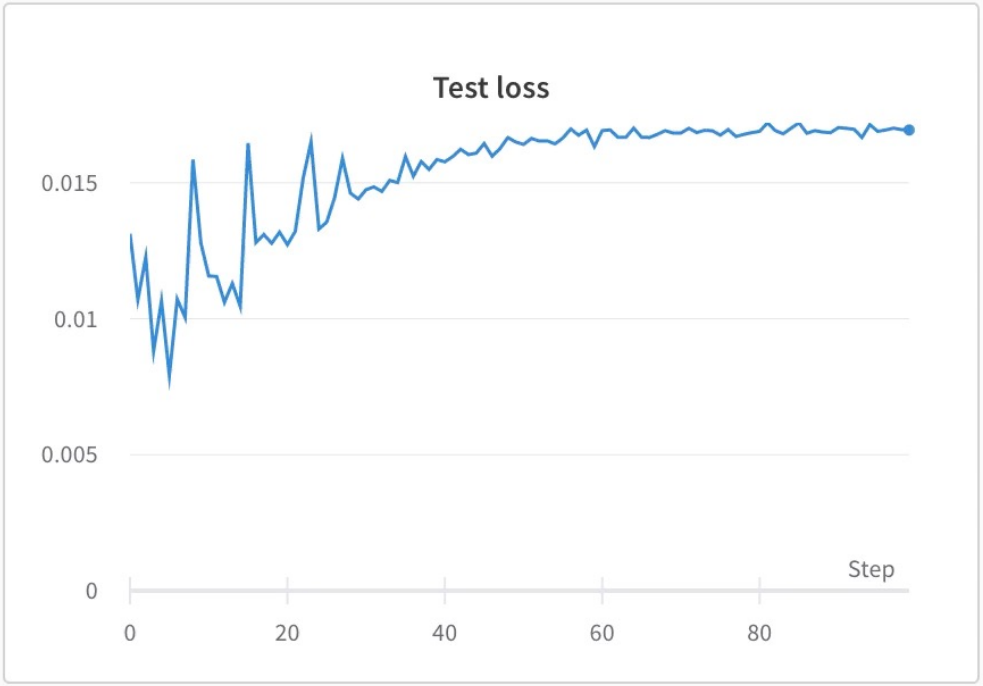
Track training with wanbd

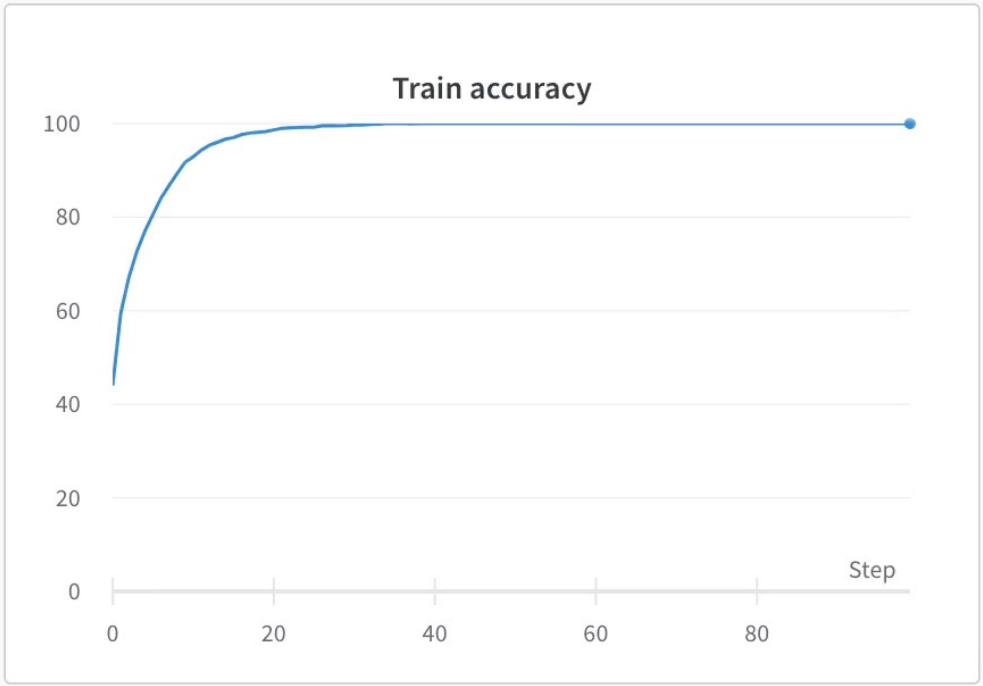
```
run.display(height=1000)
for epoch in range(wandb.config["epochs"]):
  train_acc, train_loss = train(epoch)
  test_acc, test_loss = test()
  if test_acc > best_acc:
    wandb.run.summary["Best accuracy"] = test_acc
    torch.save(model, "resnet18_best_acc.pth")
  wandb.log({"Train accuracy": train_acc,
             ”Test accuracy": test_acc,
             "Train loss": train_loss,
             "Test loss": test_loss,
             "Learning rate": optimizer.param_groups[0]['lr’]})
wandb.log({"Test data": table_test})
```

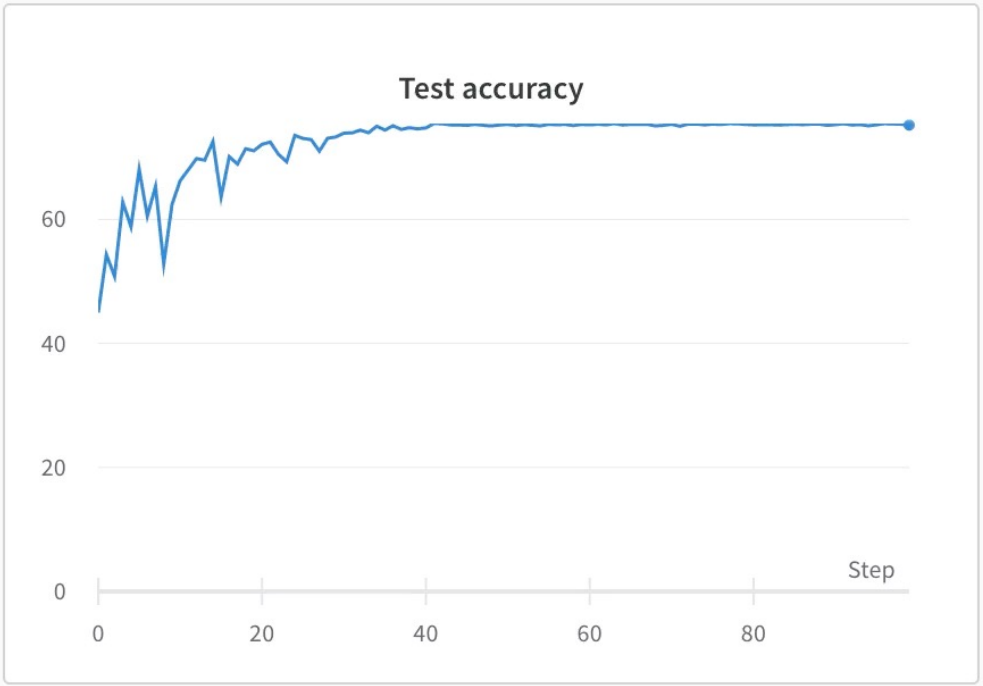
Closing `wanbd`

```
wandb.finish()
```


Train loss
0.012
0.01
0.008
0.006
0.004
0.002
0
Step
0
20
40
60
80


Test loss
0.015
0.01
0.005
0
Step
0
20
40
60
80


Train accuracy
100
80
60
40
20
0
Step
0
20
40
60
80


Test accuracy
60
40
20
0
Step
0
20
40
60
80

Share the report

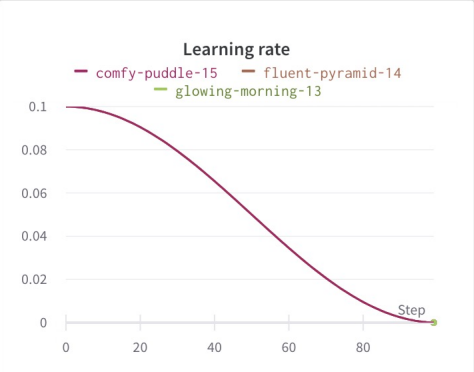
https://bit.ly/35MlkoO

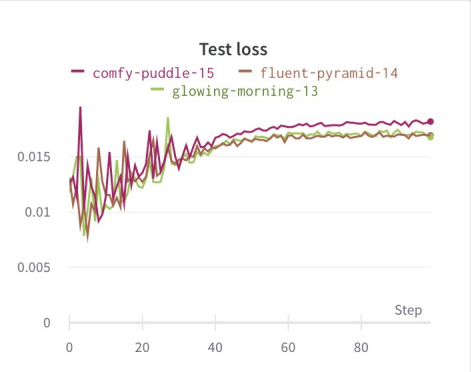
W&B and Accelerator

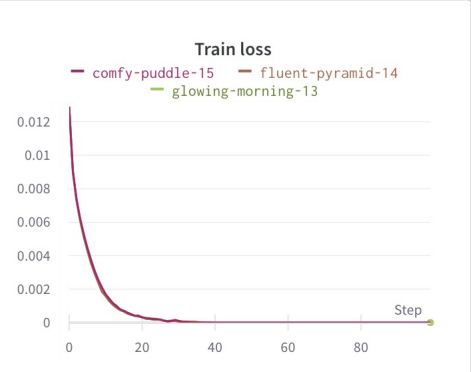
Shows wandb logs and accelerator options

Rowel Atienza

Section 1







Accelerate

https://github.com/huggingface/accelerate

Why Accelerate

Simplify model training on various hardware configs

Installing `accelerate`

```
pip install accelerate
```

Use:

```
from accelerate import Accelerator
```

Initialize `accelerate`

```
#device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
accelerator = Accelerator()

#model.to(device)
```

Wrap model, optimizer, scheduler and data loaders

```
# Accelerate API Wrapper
model = accelerator.prepare(model)
optimizer = accelerator.prepare(optimizer)
scheduler = accelerator.prepare(scheduler)
train_loader = accelerator.prepare(train_loader)
test_loader = accelerator.prepare(test_loader)
```

No need to manually transfer data to `device`

```
# Replaced by the Accelerate API.
#target = target.to(device)
#output = model(data.to(device))
output = model(data)
loss_value = loss(output, target)

# Replaced by the Accelerate API.
#loss_value.backward()
accelerator.backward(loss_value)
```

Support for fp16

```
accelerator = Accelerator(fp16=True)
```

CPU

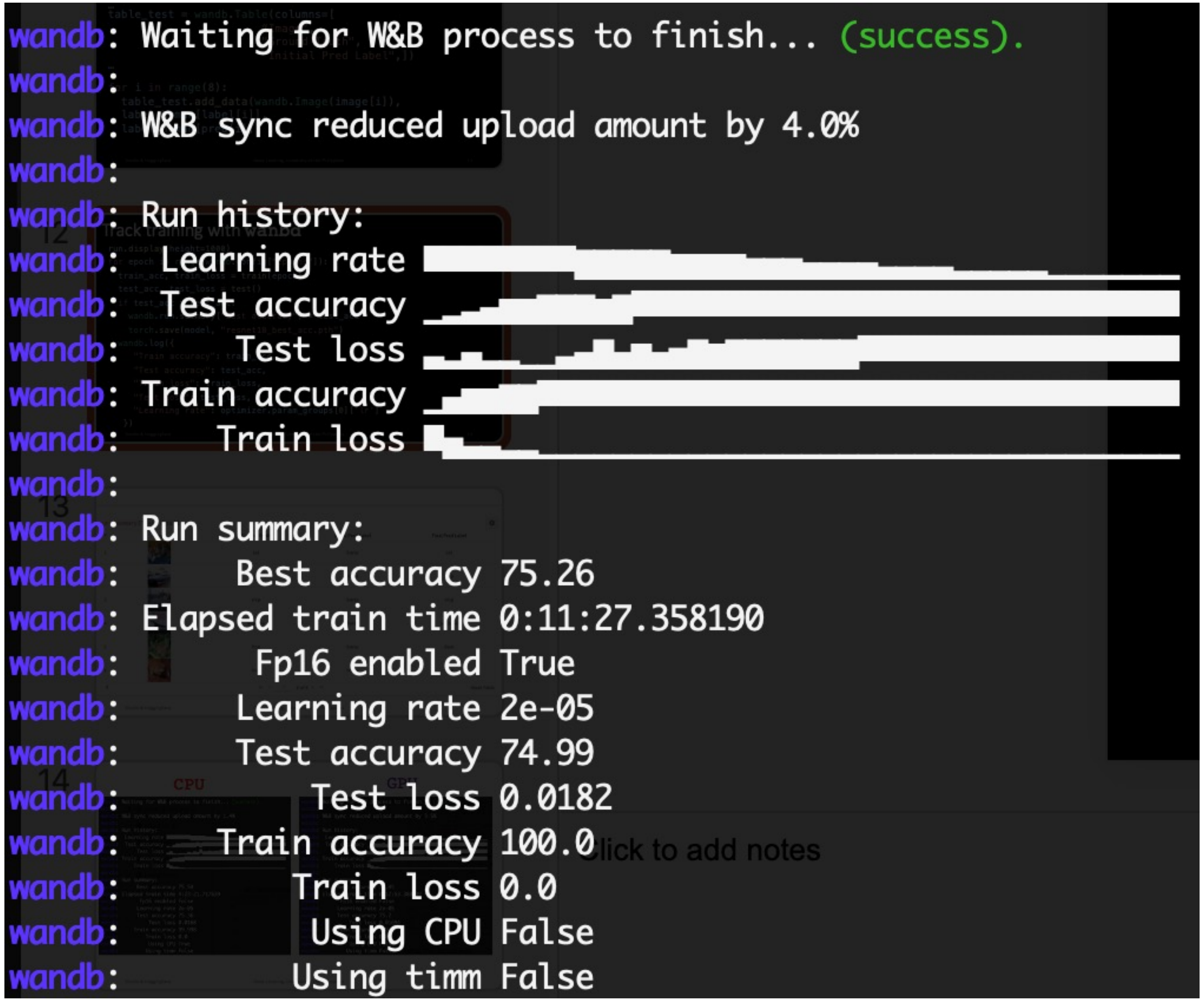
```
wandb: Waiting for W&B process to finish... (success).
wandb:
wandb: W&B sync reduced upload amount by 1.4%
wandb:
wandb: Run history:
wandb:  Learning rate
wandb:  Test accuracy
wandb:      Test loss
wandb: Train accuracy
wandb:     Train loss
wandb:
wandb: Run summary:
wandb:      Best accuracy 75.54
wandb: Elapsed train time 4:23:21.717639
wandb:       Fp16 enabled False
wandb:      Learning rate 2e-05
wandb:      Test accuracy 75.36
wandb:          Test loss 0.0168
wandb:     Train accuracy 99.998
wandb:         Train loss 0.0
wandb:          Using CPU True
wandb:         Using timm False
```

GPU

```
wandb: Waiting for W&B process to finish... (success).
wandb:
wandb: W&B sync reduced upload amount by 3.5%
wandb:
wandb: Run history:
wandb:  Learning rate
wandb:  Test accuracy
wandb:      Test loss
wandb: Train accuracy
wandb:     Train loss
wandb:
wandb: Run summary:
wandb:      Best accuracy 75.45
wandb: Elapsed train time 0:12:53.380139
wandb:       Fp16 enabled False
wandb:      Learning rate 2e-05
wandb:      Test accuracy 75.2
wandb:          Test loss 0.01694
wandb:     Train accuracy 100.0
wandb:         Train loss 0.0
wandb:          Using CPU False
wandb:         Using timm False
```

What. The. Actual. Hell.

GPU fp16



```
wandb: Waiting for W&B process to finish... (success).
wandb:
wandb: W&B sync reduced upload amount by 4.0%
wandb:
wandb: Run history:
wandb:  Learning rate
wandb:  Test accuracy
wandb:      Test loss
wandb: Train accuracy
wandb:     Train loss
wandb:
wandb: Run summary:
wandb:      Best accuracy 75.26
wandb: Elapsed train time 0:11:27.358190
wandb:       Fp16 enabled True
wandb:      Learning rate 2e-05
wandb:      Test accuracy 74.99
wandb:          Test loss 0.0182
wandb:     Train accuracy 100.0
wandb:         Train loss 0.0
wandb:          Using CPU False
wandb:         Using timm False
```

Other tools: GitHub

https://github.com/

Basic `git` Commands

```
git clone
git pull
git add
git commit
git push
git reset
git diff
```

Online Deep Learning VMs

VMs

Google Colab
Deepnote
Kaggle
Gradient
Google Cloud
Microsoft Azure
Amazon SageMaker

Other useful commands

```
tmux
nvidia-smi
```

Code demo is next

https://github.com/roatienza/Deep-Learning-Experiments/blob/master/versions/2022/tools/python/wandb_demo.ipynb

https://github.com/roatienza/Deep-Learning-Experiments/blob/master/versions/2022/tools/python/accelerate_demo.ipynb