Recitation: Software Tools

Jiawei Zhao

Outline

1. Computational Resources
2. Modeling Softwares
3. Reporting Tools
4. Useful Repositories
5. Showcase

Computational Resources

Determining what kinds of resources you need given your algorithms:

CPU heavy? GPU heavy? Memory constrained?

Available Options:

1. Google Colab (free)
2. Your own desktop
3. Amazon AWS

Google Colab

1. An interactive coding platform powered by Google
2. **Free** GPU provided
3. Similar to Jupyter Notebook
4. Codes and data are stored in Google Drive

Try it: https://colab.research.google.com/

Introduction: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Google Colab

Mounting your google drive (store your own data in the drive)

Checking RAM and CPU info

```
!cat /proc/meminfo
```

```
!cat /proc/cpuinfo
```

Common libraries are already installed, such as tensorflow and pytorch

Modeling Softwares for Machine Learning

Classical Machine Learning:

1. Scikit-Learn
2. Numpy
3. Pandas

Deep Learning (automatic differentiation library):

1. PyTorch
2. TensorBoard
3. Keras

PyTorch

Popular machine learning framework

1. Efficient computation with automatic differentiation
2. High flexibility: from research to production deployment
3. Extensive features with rich community

Libraries:

1. torch: main library for building and training models
2. torchvision: storing a list of stat-of-the-art models and datasets

PyTorch

Pipeline - how to train a deep neural network in pytorch:

1. Building your model using torch.nn.module
2. Loading your dataset through torch.utils.data.DataLoader
3. Defining your optimizer and loss function
4. Training model
5. Evaluating model

A beginning tutorial: https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

Advanced features in PyTorch

torch.cuda.amp
Automatic Mixed Precision (AMP) Training, FP32 -> FP16
Largely improve training efficiency while reducing GPU memory requirement
https://pytorch.org/docs/stable/amp.html

torch.profiler
Collecting performance metrics during training and inference
Analyzing what model operators are the most expensive
https://pytorch.org/docs/stable/profiler.html

Advanced features in PyTorch

torch.distributed

Deploy training across gpus and machines

Usually refer to data parallel training (i.e., increasing batch size)

torch.nn.DataParallel: for multi-gpus within a single machine

torch.nn.parallel.DistributedDataParallel: for multi-gpus within multi machines

Reporting Tools

Summarizing and analyzing your models

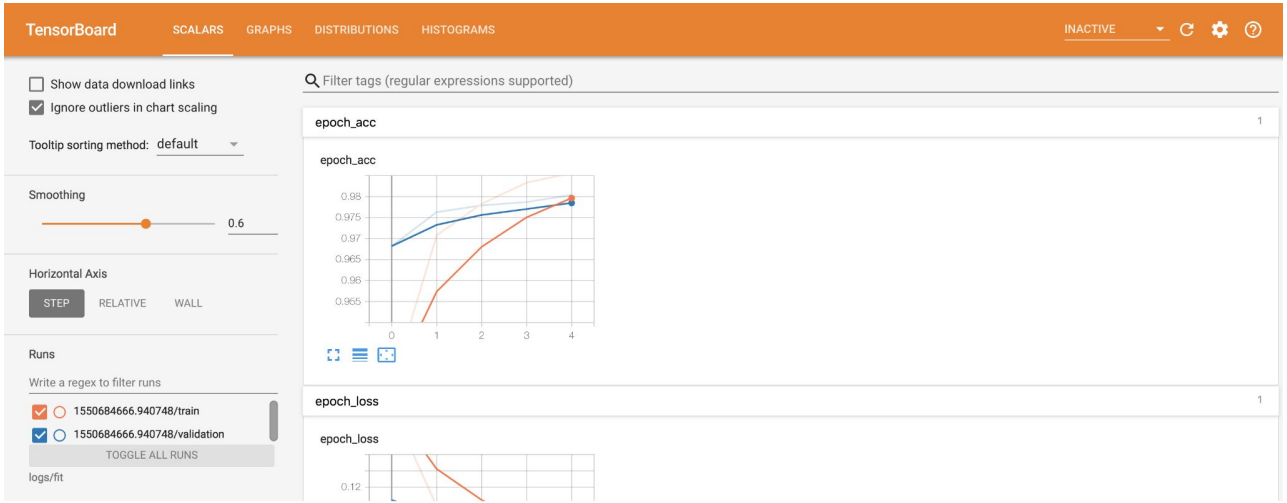
E.g., accuracy plots, weight distributions, and graphs of networks

Popular tools:

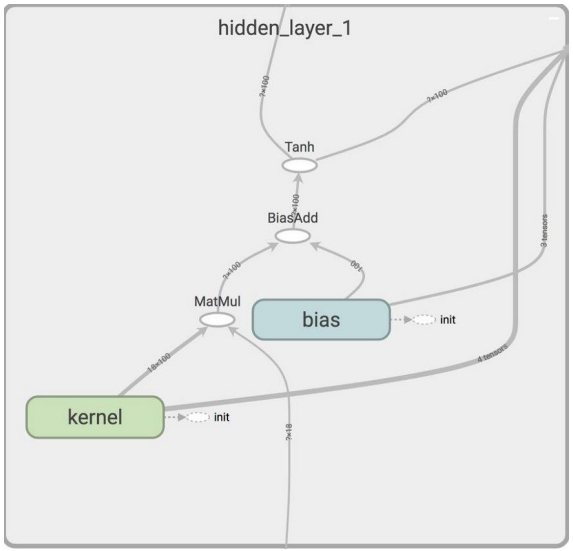
1. PyPlot
2. Tensorboard
3. Weight & Biases (WandB)

Tensorboard

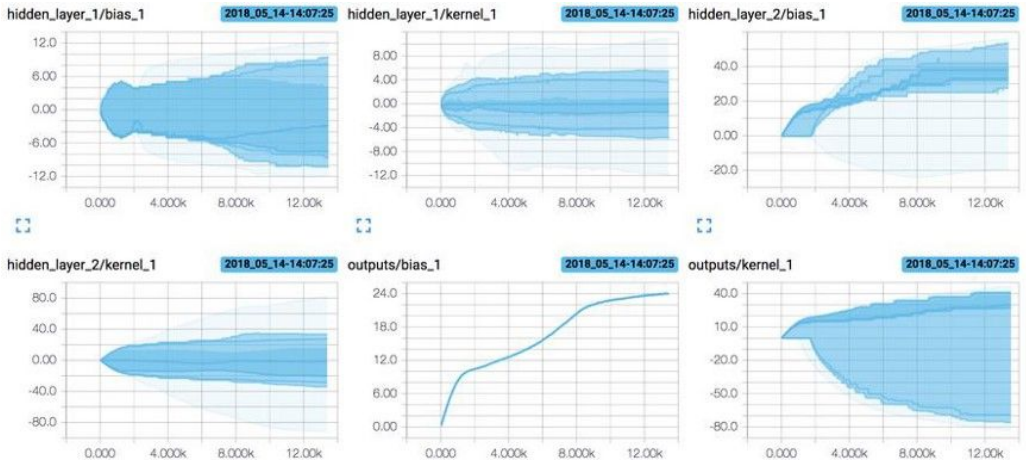
1. Light visualizing software, supported by both tensorflow and pytorch
2. Store data locally, deploy visualization locally.
3. Introduction: https://medium.com/@kkoehncke/tensorboard-for-beginners-c4709998628b



Tensorboard



Network Graph



Distribution

Weights & Biases

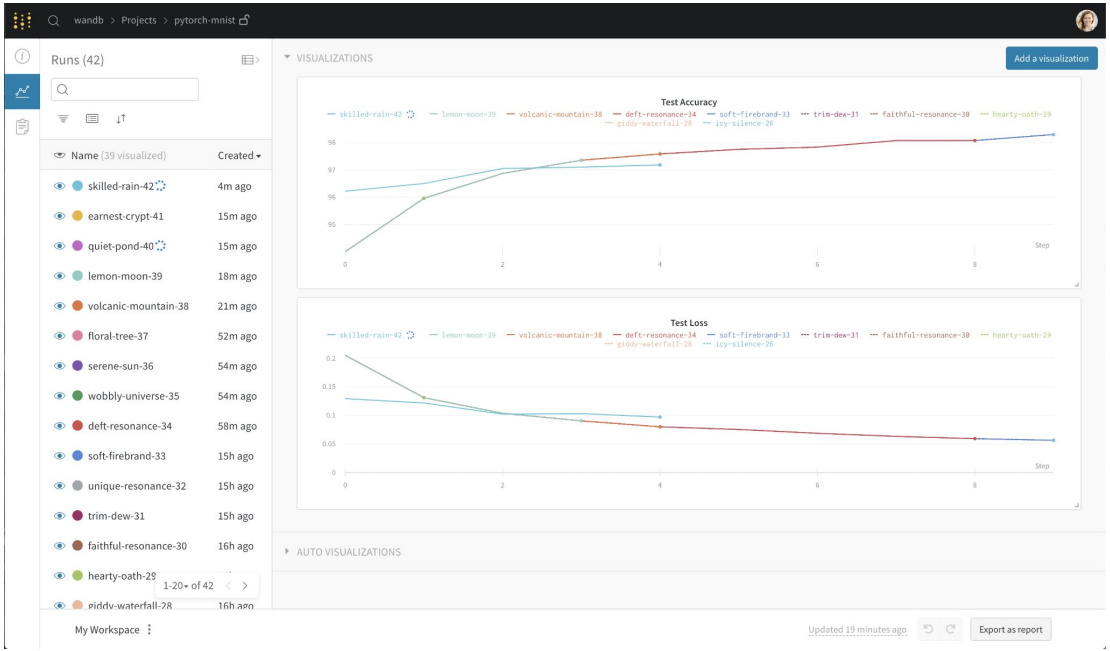
1. Advanced reporting platform for machine learning
2. Store data and view visualization both in the cloud
3. Not only experiment tracking, but includes:
    a. Hyperparameter Tuning
    b. Data + Model Versioning
    c. Collaborative Reports

A quickstart: https://docs.wandb.ai/quickstart

Features in WandB

Experiment tracking

Similar to Tensorboard, log model metrics and visualize experiment results



Features in WandB

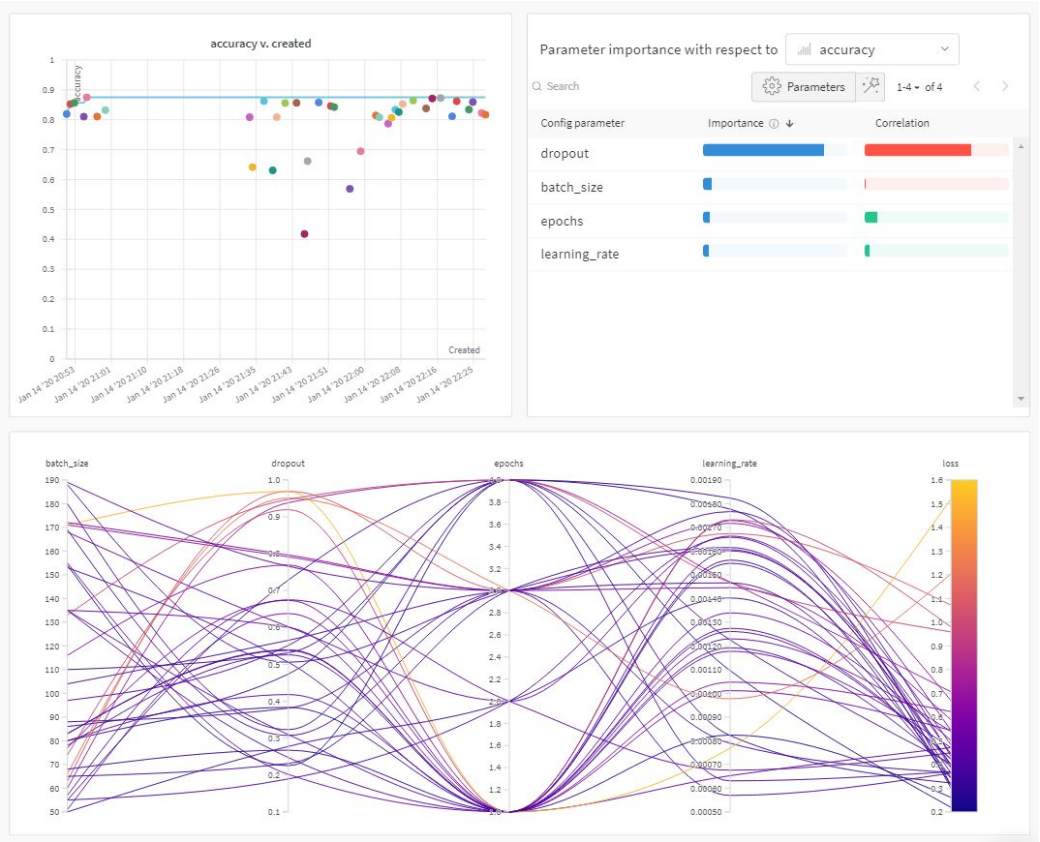
Automatic Hyperparameter Tuning

wandb.sweep

Automatically assign parameters for testing given provided training script

Support grid search, random search, and bayesian search

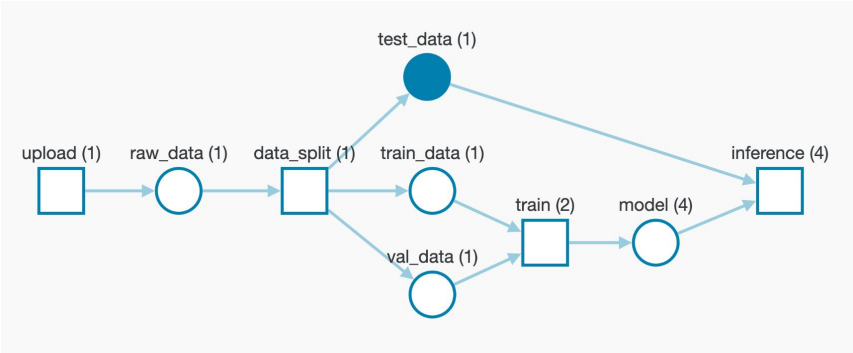
https://docs.wandb.ai/guides/sweeps/quickstart



Features in WandB

Data + Model Versioning

git version control tailored to machine learning experiments



Features in WandB

Collaborative Reports

Organize visualization, describe your findings, and share updates with collaborators

Support figures, Markdown languages, and math equations

A report example: https://wandb.ai/stacey/estuary/reports/When-Inception-ResNet-V2-is-too-slow--VmlldzO3MDcxMA

Useful Repositories for deep learning

1. PyTorch Template: https://github.com/victoresque/pytorch-template
2. PyTorch Lighting: https://github.com/PyTorchLightning/pytorch-lightning
3. Apex - Mixed Precision Training: https://github.com/NVIDIA/apex
4. Netron - Visualizing Deep Models: https://github.com/lutzroeder/netron
5. Baseline Training on CIFAR-10: https://github.com/kuangliu/pytorch-cifar

Finding more interesting repos in:
https://github.com/josephmisiti/awesome-machine-learning

Showcase

Training in PyTorch + Reporting in WandB + Deploying in Colab

https://colab.research.google.com/drive/1swXWUgjS7CBlHPRR5Vvv4d96H2E0DSe9?usp=sharing

Add-on: PyTorch Lightning

Simplified PyTorch: https://www.pytorchlightning.ai/

Benefits:

1. Reduce repeated PyTorch training codes
2. Various built-in functions to support your experiments

Watch tutorials: how to transfer PyTorch codes into Lightning codes:
https://www.pytorchlightning.ai/tutorials

Add-on: Hydra

A framework configuring complex experiments: https://hydra.cc/

Features:

1. Hierarchical configurations
2. Multi-run, parallel launchers, auto sweepers

On-boarding doc: https://hydra.cc/docs/intro/

Add-on: advanced training framework template

Template link: https://github.com/ashleve/lightning-hydra-template

PyTorch Lightning + Hydra

Ultimate solution storing all experiments across tasks

Benefits:

1. Test your algo across different tasks and models easily
2. All benefits from PyTorch Lightning and Hydra