

# INTRODUCTION TO WANDB



Weights & Biases

By: N.S.Akshara

# WHAT IS WANDB?

WandB (Weights and Biases) is a powerful ML Operations platform that helps data scientists and ML engineers track, visualize, and manage their experiments seamlessly.

## Track Everything

Metrics, hyperparameters, model versions, and system resources

## Visualize in Real-Time

Interactive dashboards updated during training

## Collaborate

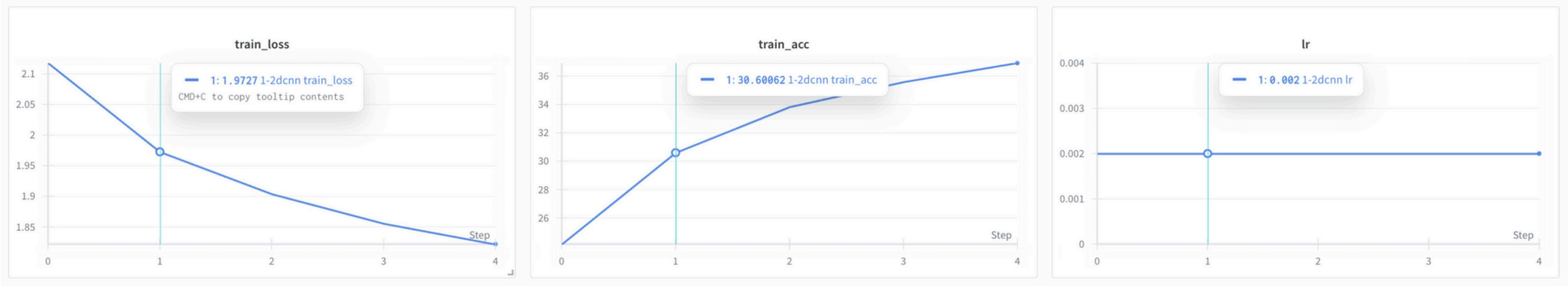
Share experiments with teammates easily

## Integrates Seamlessly

Works with PyTorch, TensorFlow, scikit-learn, and more

# KEY FEATURES OF WANDB

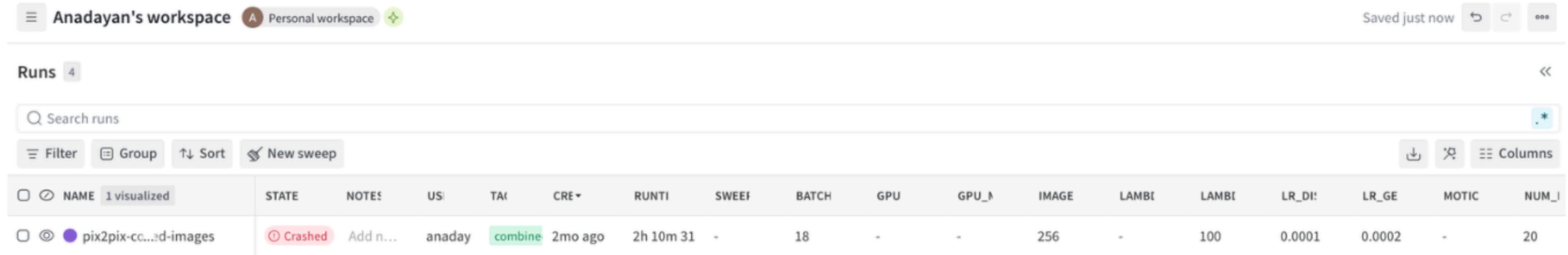
## Real-time Performance Logging



- Visualize metrics like Loss and Accuracy F1 in real-time as the model trains.
- Keep track of different models and training runs.
- Spot overfitting or exploding gradients instantly using interactive charts.

# KEY FEATURES OF WANDB

## *Crash Recovery & Resumption*



The screenshot shows the WandB interface for 'Anadayan's workspace'. At the top, it says 'Saved just now' with a refresh icon. Below that, it shows 'Runs 4'. A search bar and filter options ('Filter', 'Group', 'Sort', 'New sweep') are available. The main table lists four runs:

	NAME	STATE	NOTES	USI	TAT	CRE	RUNTI	SWEET	BATCH	GPU	GPU_N	IMAGE	LAMBDA	LAMBDA	LR_DIF	LR_GE	MOTIC	NUM_I
<input type="checkbox"/>	pix2pix-cc...d-images	Crashed	Add n...	anaday	combine	2mo ago	2h 10m 31	-	18	-	-	256	-	100	0.0001	0.0002	-	20

- WandB can save your model's checkpoints to the cloud.
- By passing the specific Run ID, you can download the weights and resume the epoch exactly where you left off.

# KEY FEATURES OF WANDB

## *Hyperparameter Tuning*

- Automating the search to find the best configuration (Learning Rate, Batch Size, Optimizer).
- Let WandB try different configurations while you sleep!

## Collaboration

- All team members log runs to the same "Project" bucket.
- See everyone's runs side-by-side to determine whose model is performing best.

# SET UP WANDB ACCOUNT

The AI developer platform to build AI agents, applications, and models with confidence

**W&B Weave: Build agentic AI applications**

```
1 import weave
2 weave.init("quickstart")
3 @weave.op()
4 def llm_app(prompt):
5     pass # Track LLM calls, document retrieval, agent steps
```

**W&B Models: Build AI models**

```
1 import wandb
2 run = wandb.init(project="my-model-training-project")
3 run.config = {"epochs": 1337, "learning_rate": 3e-4}
4 run.log({"metric": 42})
5 my_model_artifact = run.log_artifact("./my_model.pt", type="model")
```

[GET STARTED WITH WEAVE](#)

[REQUEST DEMO](#)

[GET STARTED WITH MODELS](#)

[SIGN UP >](#)

<https://wandb.ai/site/>

Sign up

Log in Sign up

Sign up with GitHub

Sign up with Google

Sign up with Microsoft

Hint: Use your work email to easily find and join your team.

name@work-email.com

your password

By signing up, you agree to our terms of service and privacy policy.

[SIGN UP >](#)

# SET UP WANDB API KEY

API keys

[Manage the API keys associated with your account.](#)

[Reveal](#)

- Your API key authenticates your machine/notebook so runs can upload to your WandB dashboard.
- You can find your key in the “API Keys” section in your WandB user settings.

---

**Never post your API key in GitHub, Piazza, screenshots, or shared notebooks!**

→ If it does get leaked, please rotate the key immediately in WandB settings.

# WANDB - NEW PROJECT

The screenshot shows the WANDB interface for the team 'anadayan-cmu'. The top navigation bar includes 'Overview', 'Reports', 'Projects', 'Users', 'Service Accounts', and 'Settings'. On the left, there's a sidebar with a user icon, the team name 'anadayan-cmu', and options like 'Team settings' and 'Registry'. Below the sidebar, under 'MEMBERS (1)', there's a link to 'Invite team members' and a profile for 'anadayan'. A 'Search by project name' bar is at the bottom.

**Overview**

**About**

Create your team's central resource  
Share context, instructions, and resources to help team members succeed.  
[Create](#)

**Projects**

Highlight your latest projects  
Make a project public to showcase here!  
[View my projects](#)

**Projects**

[+ New project](#)

### Create a new project

This project will be created inside your academic account.

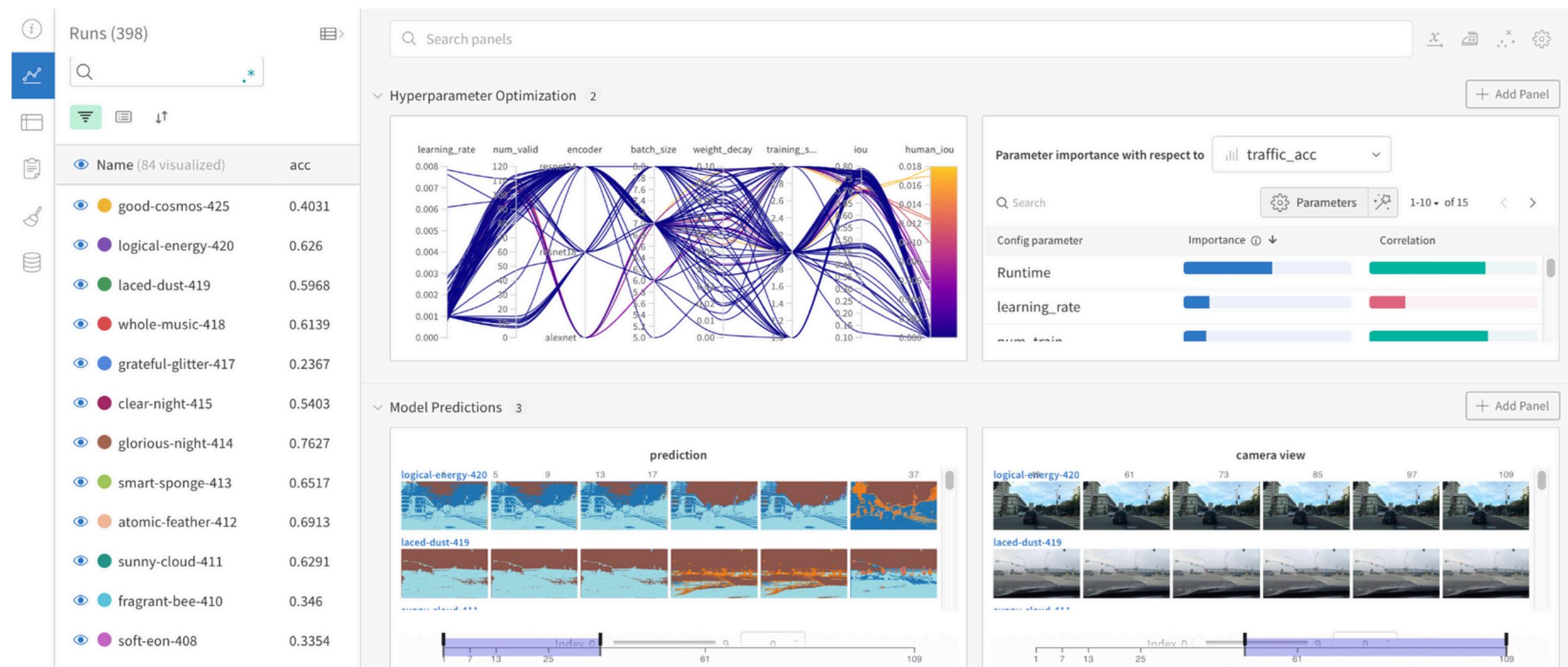
**Team**  
anadayan-cmu

**Name**  
Give your project a name

**Project visibility**  
Select the visibility for this project

**Description (optional)**

# WANDB DASHBOARD



# COLLABORATING ON WANDB

The screenshot shows the WandB team interface for 'av-team'. At the top, there's a navigation bar with tabs for Overview, All Reports, All Projects, and Members. Below the navigation is a section titled 'Intro' containing a brief introduction and three bullet points: 'Our interests include Perception, Planning & Control', 'We're currently building Autonomous Vehicles, of course!', and 'Let's collaborate on making vehicles drive themselves!'. To the left of the intro section is a sidebar with links for Team settings, Enterprise dashboard, and Model Registry. Below the sidebar are sections for 'WEEKLY MOST ACTIVE' and 'RUNS'. On the right side, there's a 'MEMBERS (18)' section with a list of team members: hamelsmu, admrswanberg, lavanya, morgan, ayut, ivangrov, and others. At the bottom, there's a 'Reports' section showing two reports: 'A System of Record for Autonomous Driving...' and 'Modeling Drivable Areas :: for Autonomous Vehicle...'. A 'Select reports to showcase' button is also present.

Source: WandB documentation

```
# "entity" = The Team Name  
wandb.init(entity="intro-dl-group-4",  
            project="final-project")
```

- Go to Settings > Create New Team. Invite members via email or username.
- Make sure to use the WandB init call used for the team when using it!

# COLLABORATING ON WANDB

## Roles

Add a new role to your organization with customized permissions.

### Viewer

View-Only members can view assets within the team such as runs, reports, and workspaces. They can follow and comment on reports, but they can not create, edit, or delete project overview, reports, or runs.

### Member

A regular member of the team. A team member is invited by email by the team admin. A team member cannot invite other members. Team members can only delete runs and sweep runs created by that member.

### Admin

Team admins can add and remove other admins or team members. They have permissions to modify all projects and full deletion permissions.

### Custom roles

Create a role by inheriting a pre-defined role and adding permissions to it.

+ Create a role

[Contact sales](#) to upgrade and access this feature.

# INSTALLATION & NOTEBOOK SET UP

## Wandb Installation for Python

### ▼ Installation and Libraries

```
[ ] ## Installing WandB  
!pip install wandb -qqq
```

## Notebook Setup

- Dataset: CIFAR10
- Neural Network: Convolutional Neural Network
- All functionalities are built in functional blocks for automated access.

*Credits to Fall 2024 TAs for the Notebook!*

# NOTEBOOK: WANDB LOGIN

```
[ ] import wandb, os  
os.environ['WANDB_API_KEY'] = "7328d336610dec777bdecff906dc57e2a464d0b4"#your key here  
wandb.login()  
  
[+] wandb: Using wandb-core as the SDK backend. Please refer to https://wandb.me/wandb-core for more information.  
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)  
wandb: You can find your API key in your browser here: https://wandb.ai/authorize  
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit: .....  
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc  
True
```

# NOTEBOOK: INITIALIZING PROJECT

```
run_config = {
    'model': '1-2dcnn',
    'optimizer': 'sgd',
    'lr': 2e-3,
    'batch_size': 64,
    'epochs': 5
}

train_loader, test_loader = build_data(run_config['batch_size'],
                                        data_train, data_test)

optimizer = get_optim(run_config['optimizer'], run_config['lr'], model)

criterion = nn.CrossEntropyLoss()

scaler = torch.cuda.amp.GradScaler()

run = wandb.init(
    #entity="wandb-starter",
    project="wandb-quickstart",
    #job_type="model-training",
    name=run_config['model'],
    config=run_config
)
```

---

Tracking run with wandb version 0.23.1

Run data is saved locally in /content/wandb/run-20251214\_062711-h4lcg101

Syncing run [1-2dcnn](#) to [Weights & Biases \(docs\)](#)

View project at <https://wandb.ai/anadayan-cmu/wandb-quickstart>

View run at <https://wandb.ai/anadayan-cmu/wandb-quickstart/runs/h4lcg101>

# NOTEBOOK: LOGGING & SAVING MODELS

```
metrics = {
    "train_loss":train_loss,
    "train_acc": train_acc,
    'lr': lr
}
```

*Logging metrics*

```
torch.save({
    'model_state_dict': model.state_dict(),
    'optimizer_state_dict': optimizer.state_dict()
}, "Model.pth")

# ALTERNATIVE 1: Saving Files as Artifacts
# Creating Artifact
model_artifact = wandb.Artifact(run_config['model'], type='model')

# Adding model file to Artifact
model_artifact.add_file("Model.pth")

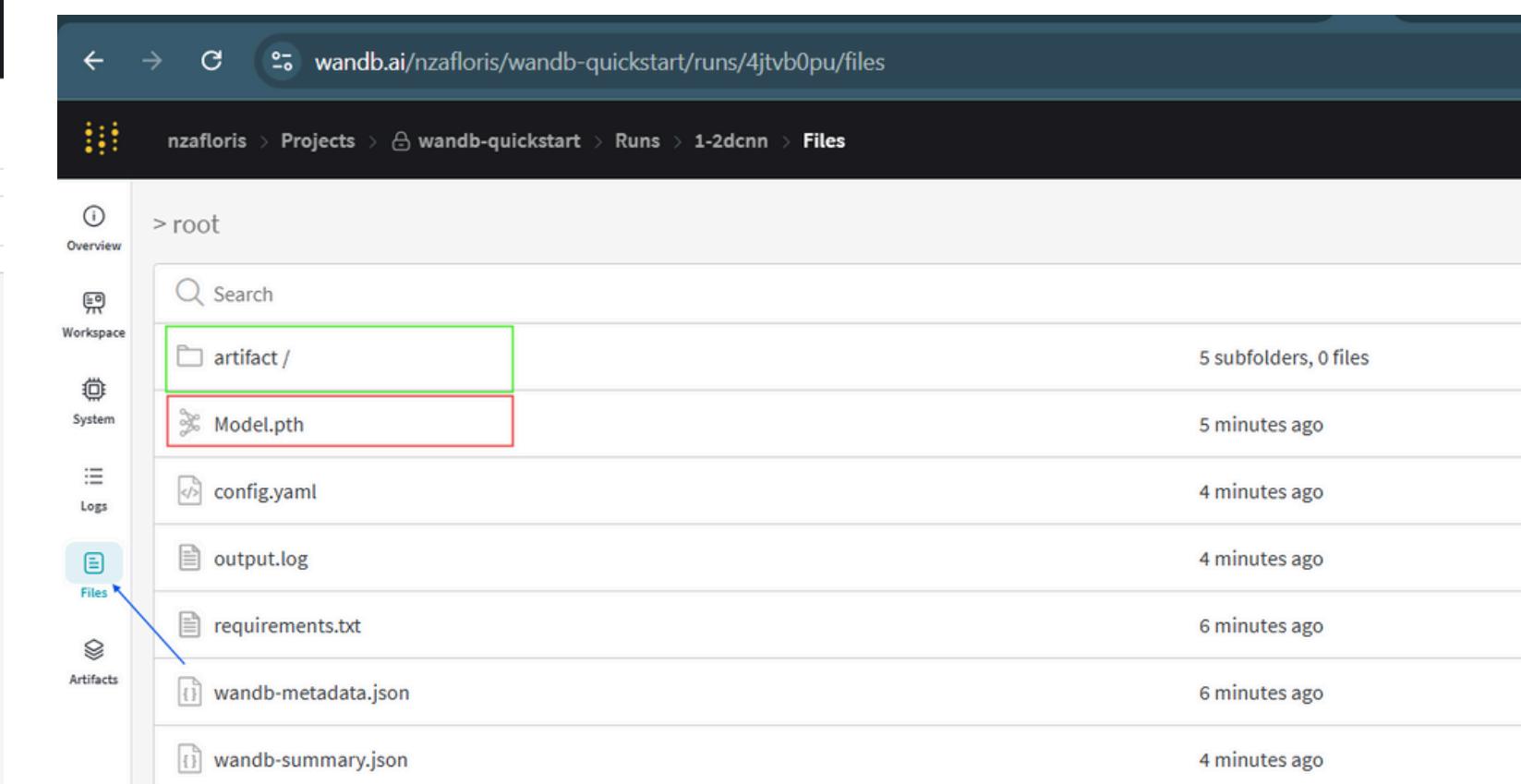
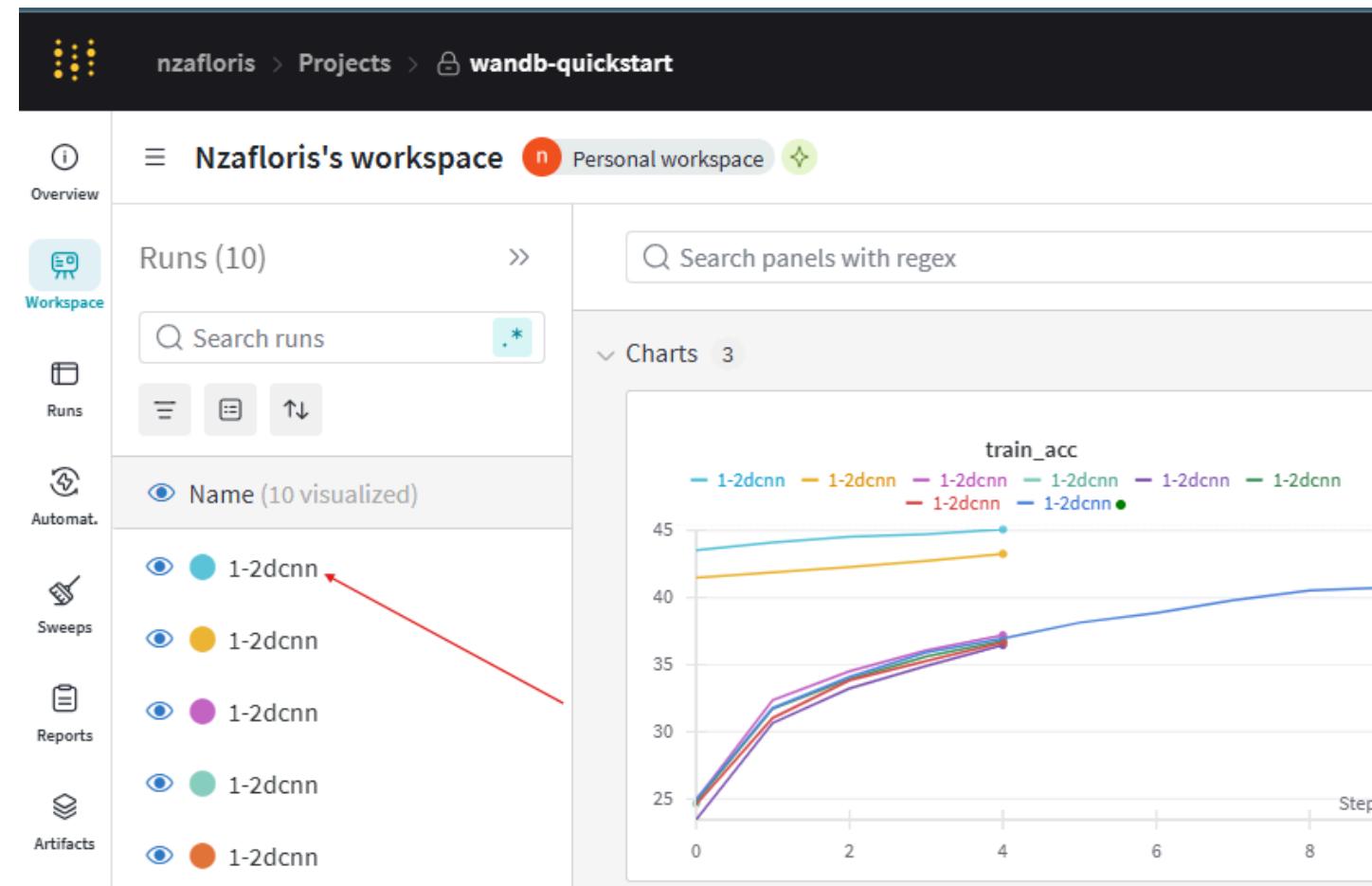
# Saving Artifact to WandB
run.log_artifact(model_artifact)

# ALTERNATIVE 2: Saving Files as Files
wandb.save("Model.pth")

if finish:
    wandb.finish()
```

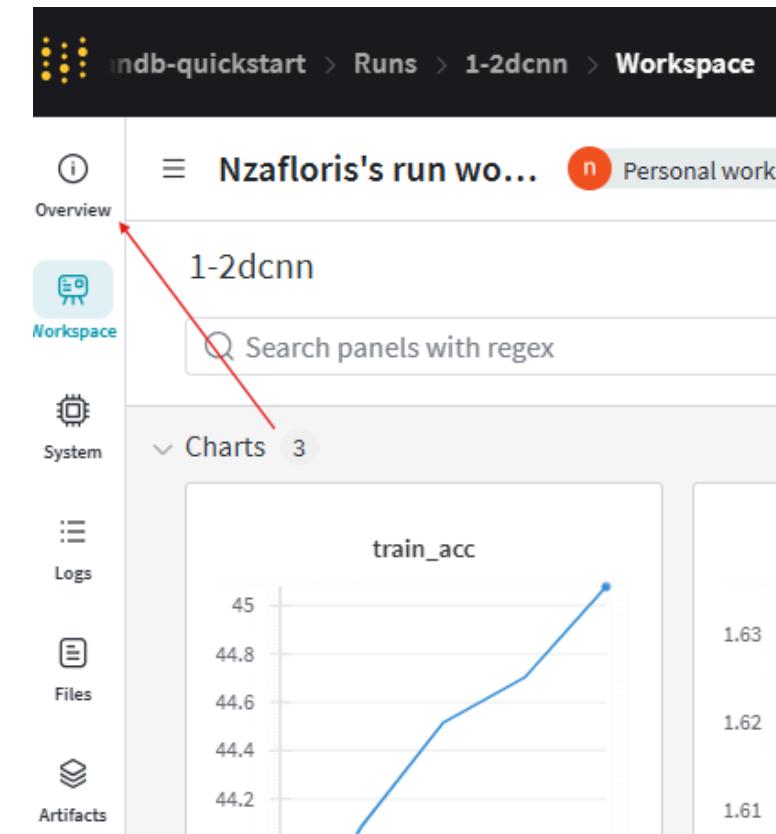
*Saving models*

# NOTEBOOK: LOGGING & SAVING MODELS



# NOTEBOOK: GET RUN ID

The screenshot shows the WandB interface for the project "wandb-quickstart". The left sidebar includes sections for Overview, Workspace (selected), Runs, Automat., Sweeps, Reports, and Artifacts. The main area displays "Nzafloris's workspace" with 10 runs. A search bar for runs is present. Below it, a chart titled "1-2dcnn" visualizes training accuracy over time.



The screenshot shows the "Overview" page for the "1-2dcnn" run. The top navigation bar shows the path: "wandb-quickstart > Runs > 1-2dcnn > Overview". The page includes sections for Overview, Tags, Author, State, Start time, Duration, Run path, Hostname, OS, Python version, Python executable, Colab link, Command, System Hardware, W&B CLI Version, and Job Type. The "Run path" field is highlighted with a red box. The "Colab link" field contains a URL: <https://colab.research.google.com/notebook#fileId=1eycSFUdDE92qMizHCo>.

# NOTEBOOK: RESUMING RUN

```
run_id = "4jtvb0pu" ### Replace with run id string
run = wandb.init(
    id      = run_id, ### Insert specific run id here if you want to resume a previous run
    resume = "must", ### You need this to resume previous runs, but comment out reinit = True when using this
    project = "wandb-quickstart", ### Project should be created in your wandb account
)
```

- We use resume='must' in the code. This is a safety feature which prevents you from accidentally starting a new run instead of resuming.
- An often noticed mistake is loading the model.load\_state\_dict but forgetting optimizer.load\_state\_dict. If you forget the optimizer, you lose your momentum and learning rate information!

# HYPERPARAMETER SWEEPING

Hyperparameters refer to parameters set before learning begins.  
They control the behavior of the training algorithm!

## Examples

- Learning Rate
- Batch Size: Samples processed before update.
- Optimizer: Adam, SGD, RMSProp.
- Dropout Rate: Probability of ignoring neurons.

# NOTEBOOK: HYPERPARAMETER SWEEPING

```
[ ] # Hyperparameters to work with

parameters_dict = [
    'optimizer':{
        'values': ['sgd', 'adam']
    },
    'learning_rate':{
        'distribution': 'uniform',
        'min': 2e-4,
        'max': 1e-1
    },
    'batch_size': {
        'distribution': 'q_log_uniform_values',
        'q': 4,
        'min': 16,
        'max': 128
    },
    'epochs':{
        'value': 5
    }
]
sweep_config['parameters'] = parameters_dict
```

```
[ ] # Initalizing the sweep

sweep_id = wandb.sweep(sweep_config, project="CIFAR-Sweep2")
```

```
▶ def train_sweep(config = None):
    with wandb.init(config=config) as run:
        run.name=f"Jeel_{wandb.config.learning_rate}_{wandb.config.batch_size}_{wandb.config.optimizer}"
        config = wandb.config

        train_loader, test_loader = build_data(config.batch_size, data_train, data_test)

        model = Network().to(device)

        optimizer = get_optim(config.optimizer, config.learning_rate, model)

        criterion = nn.CrossEntropyLoss()

        scaler = torch.cuda.amp.GradScaler()

        for epoch in range(config.epochs):

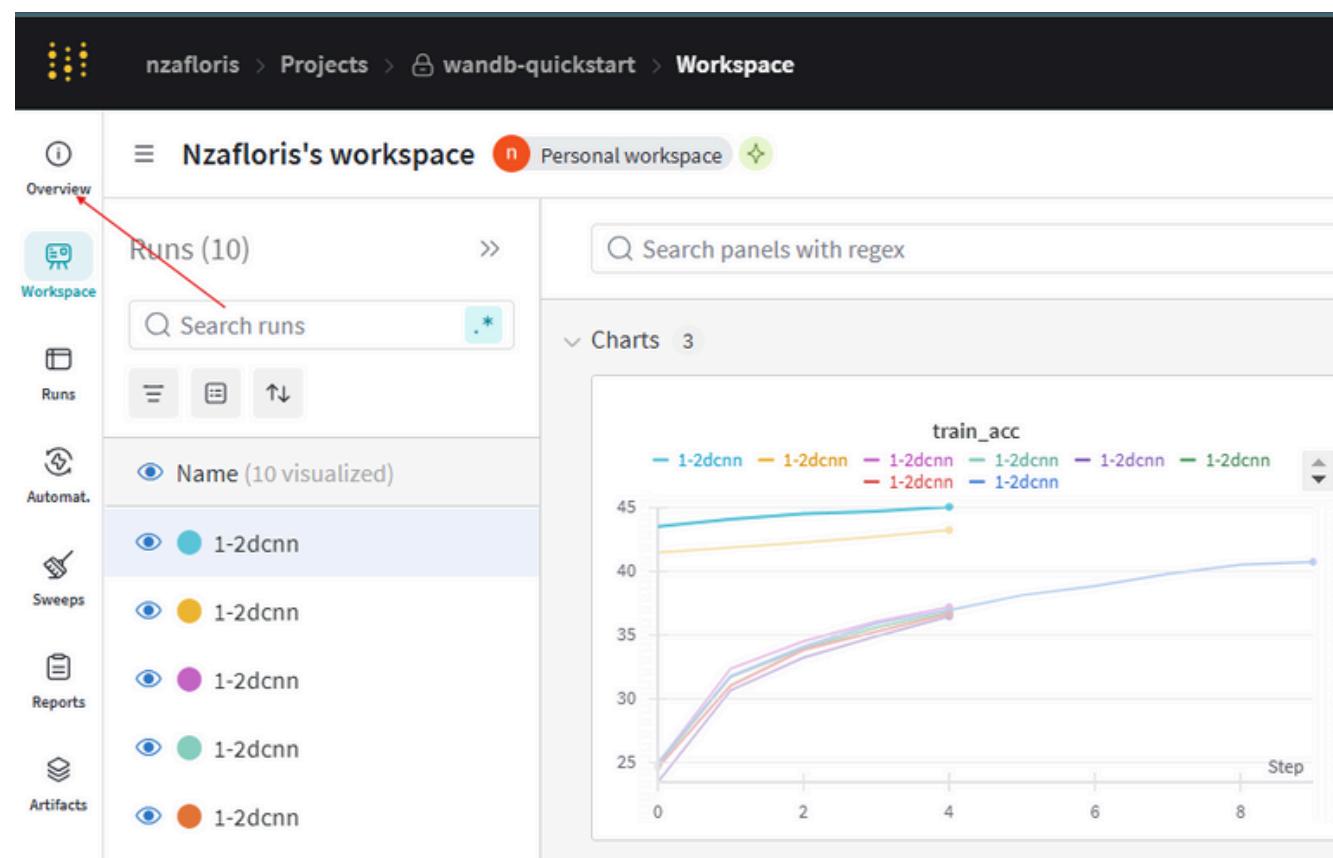
            model, loss = train_epoch(model, train_loader, optimizer, criterion, scaler)

            wandb.log({'loss': loss})
```

```
[ ] # Running the sweep

wandb.agent(sweep_id, train_sweep, count=2)
```

# MAKING YOUR WANDB PROJECT PUBLIC



The screenshot shows the WandB project overview and edit pages for the "wandb-quickstart" project. The top navigation bar includes "Projects", "wandb-quickstart", "Overview", "Invite teammates", and user information. The "Edit" button is highlighted with a red arrow. The project details include:

Project visibility	Private
Last active	12/19/2024, 6:20:18 PM
Owner	Floris Nzabakira
Contributors	1 user
Total runs	10
Total compute	17 minutes

The "Edit project" dialog box shows the current project name "wandb-quickstart" and the "Project visibility" dropdown set to "Private". Other options include "Public", "Open", and "Restricted".

# WANDB STORAGE FOR ACCOUNTS

Once your WandB free storage is full, you won't be able to access your runs!

- Individual Account:
  - Provides 100GB free
  - Can be accessed through <https://wandb.ai/subscriptions>
- Group Account:
  - Provides 5GB free per team
  - Can be accessed through:  
<https://wandb.ai/account-settings/<Team-Name>/billing>



**THANK YOU!**

