



PPCES: Machine and Deep Learning

Hands-on Exercises



DAS KOMPETENZNETZWERK FÜR HOCHLEISTUNGSRECHNEN.

- **Hands-on material and slide decks can be found on the PPCES website**
 - <https://blog.rwth-aachen.de/itc-events/en/event/ppces-2026/>
- **Download and extract material on desired machine**
 - Separate folders for scikit-learn and PyTorch
 - scikit-learn available as regular Python (.py) and Jupyter Notebooks (.ipynb)

```
# change to the directory where you want to save your material
cd <working directory>

# download material and examples
wget https://hpc.rwth-aachen.de/ppces/ppces2026-ML-DL-labs.tar.gz

# unpack the tar file
tar -xzvf ppces2026-ML-DL-labs.tar.gz
```

3 Ways to Access the HPC Cluster



Classic / Batch Mode

Login via SSH
SLURM Workload
Manager

Module System
Apptainer (Singularity)



Remote Desktop

Login via FastX
Graphical Interface

Rest is the same as in
Batch Mode



Interactive Mode

JupyterHub/Lab
Web Interface

Predefined Profiles

- **Dedicated hardware (and accounting) for the workshop**
 - 7 CLAIX-2023-ML nodes – each with 4x H100

Example in batch script file:

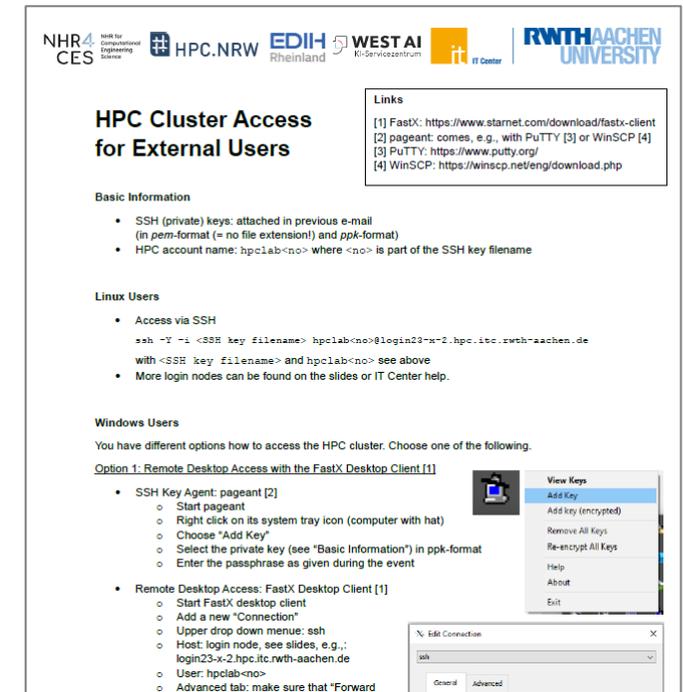
Advanced reservation	ppces
Compute time project	lect0174

```
#SBATCH --reservation=ppces  
#SBATCH --account=lect0174
```

**ONLY relevant for
PyTorch exercise**

- Login nodes (full list [here](#))
 - login23-1.hpc.itc.rwth-aachen.de (SSH)
 - login23-2.hpc.itc.rwth-aachen.de (SSH)
 - login23-x-1.hpc.itc.rwth-aachen.de (X-Server, remote desktop sessions, [web access available](#))
 - login23-x-2.hpc.itc.rwth-aachen.de (X-Server, remote desktop sessions, [web access available](#))

- **Members of RWTH (or affiliated persons)**
 - HPC account + Two-factor-authentication (2FA) required:
 - <https://regapp.itc.rwth-aachen.de/>
 - [Instructions](#) (Step 3 – Step 5)
 - If provided during survey, all permissions for dedicated hardware have already been granted
 - If not, please contact the IT center staff during the lab session!
- **External participants**
 - Temporary account and SSH key was sent to you by e-mail
 - Account: hpclab[01-20]
 - Key: Linux format (pem) or PuTTY/pageant format (ppk)
 - If you didn't get an e-mail, please contact the IT Center staff
 - Password: will be provided during the lab sessions
 - Hpclab accounts have all permissions for dedicated hardware
 - **See Handout: HPC Cluster Access** for further details



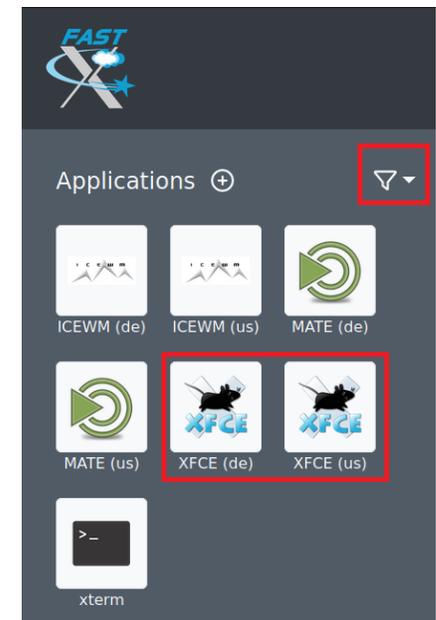
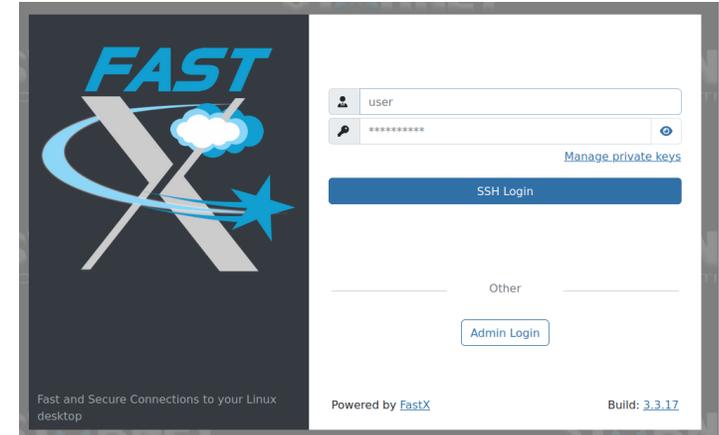
– Access via SSH

- Access from within the RWTH network / eduroam via ssh:
 - `ssh -Y ab123456@login23-2.hpc.itc.rwth-aachen.de`
- Tools:
 - Linux: Bash/Zsh/... + SSH
 - PuTTY: <https://www.putty.software/>
 - GitBash for Windows: <https://git-scm.com/download/win>

Accessing CLAIX – Access (FastX)

– FastX3 (desktop client or browser)

- Browser client: <https://login23-x-2.hpc.itc.rwth-aachen.de:3300>
- Desktop client: login nodes (see slides before)
- Select the correct application
 - Recommended: XFCE
 - May need to select “all applications” in filter at the top right corner of the Application box
- Start the session
 - Click on the session (may have to click twice)
- Open up a terminal
 - Bottom of the screen
 - Click the “terminal” icon



Part 1: Hands-on: scikit-learn

- **Large Jobs / high demand** → JupyterHub HPC
 - Special JupyterHub instance
 - Starts containers on HPC cluster nodes
 - <https://jupyterhub.hpc.itc.rwth-aachen.de:9651>
- **Alternatives**
 - Another RWTH JupyterHub
 - Cheaper standard servers under the hood
 - <https://jupyter.rwth-aachen.de/hub/login>
 - Profiles / kernles might differ
 - **Install JupyterLab in your local virtual environment**
 - Execution happens on your local machine
 - Visual Studio Code integration available

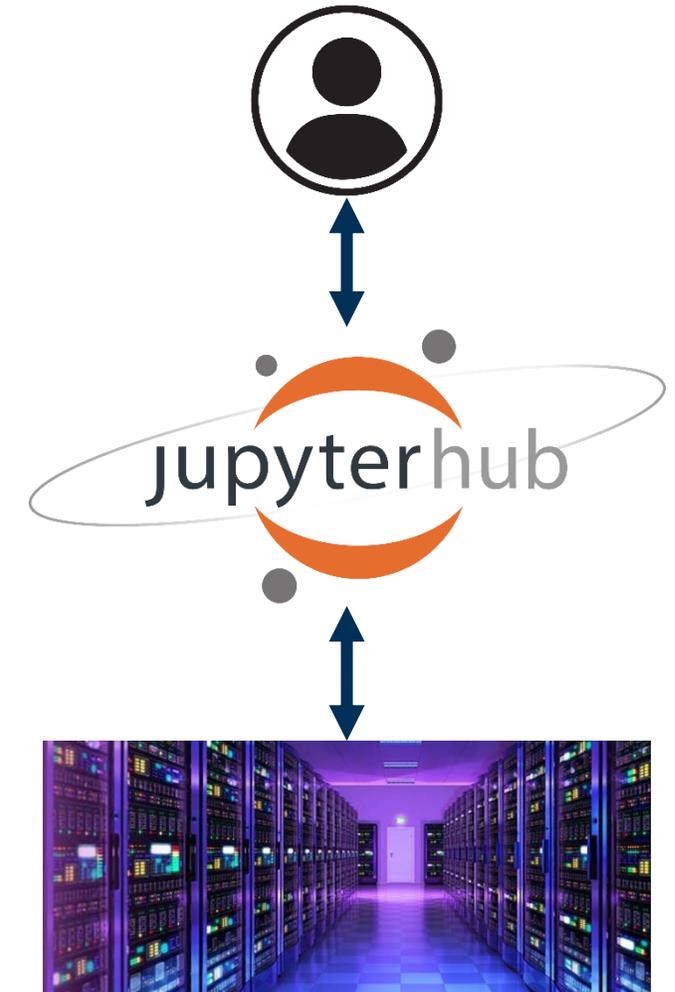
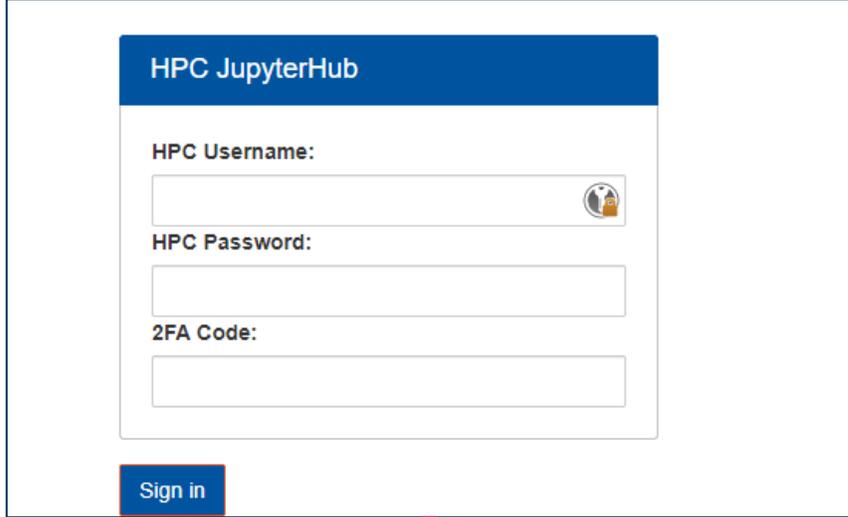


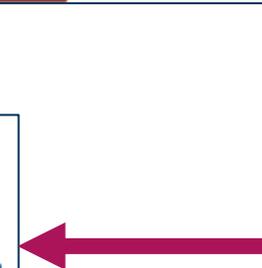
Image: scientific-computing.com

- **Step 1:** Login to JupyterHub HPC
 - Similar to SSH or FastX login
 - You need your HPC 2FA again here

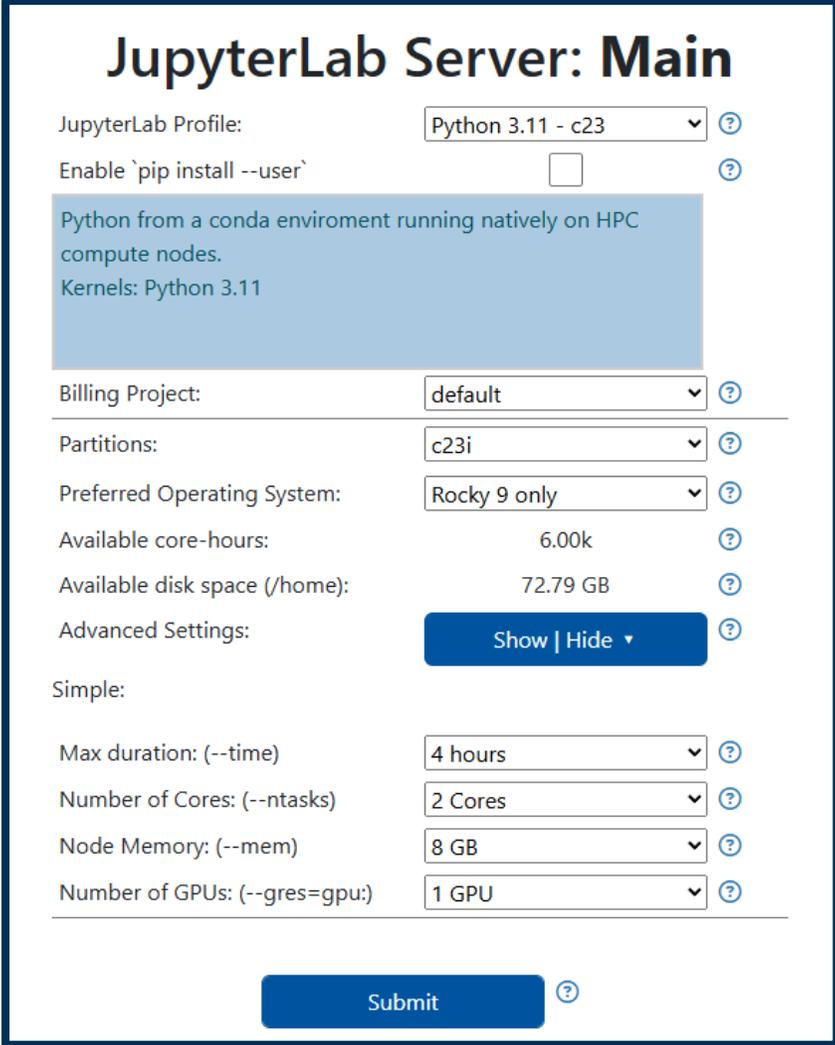


The screenshot shows a login form titled "HPC JupyterHub". It contains three input fields: "HPC Username:" with a search icon, "HPC Password:" with a password icon, and "2FA Code:". A blue "Sign in" button is located at the bottom left of the form.

Main JupyterLab Server
You can start your main JupyterLab server here:



- **Step 2:** Configure and start your JupyterLab server
 - Request resources on the HPC cluster
 - **General Settings**
 - JupyterLab Profile Python 3.11 – c23
 - Billing Project default
 - Partition c23i
 - **Simple Settings**
 - Number of Cores 2
 - Memory per Node: 8 GB
 - Max. duration depending on how long you want to test



JupyterLab Server: Main

JupyterLab Profile: Python 3.11 - c23 ⓘ

Enable `pip install --user` ⓘ

Python from a conda environment running natively on HPC compute nodes.
Kernels: Python 3.11

Billing Project: default ⓘ

Partitions: c23i ⓘ

Preferred Operating System: Rocky 9 only ⓘ

Available core-hours: 6.00k ⓘ

Available disk space (/home): 72.79 GB ⓘ

Advanced Settings: [Show | Hide](#) ⓘ

Simple:

Max duration: (--time) 4 hours ⓘ

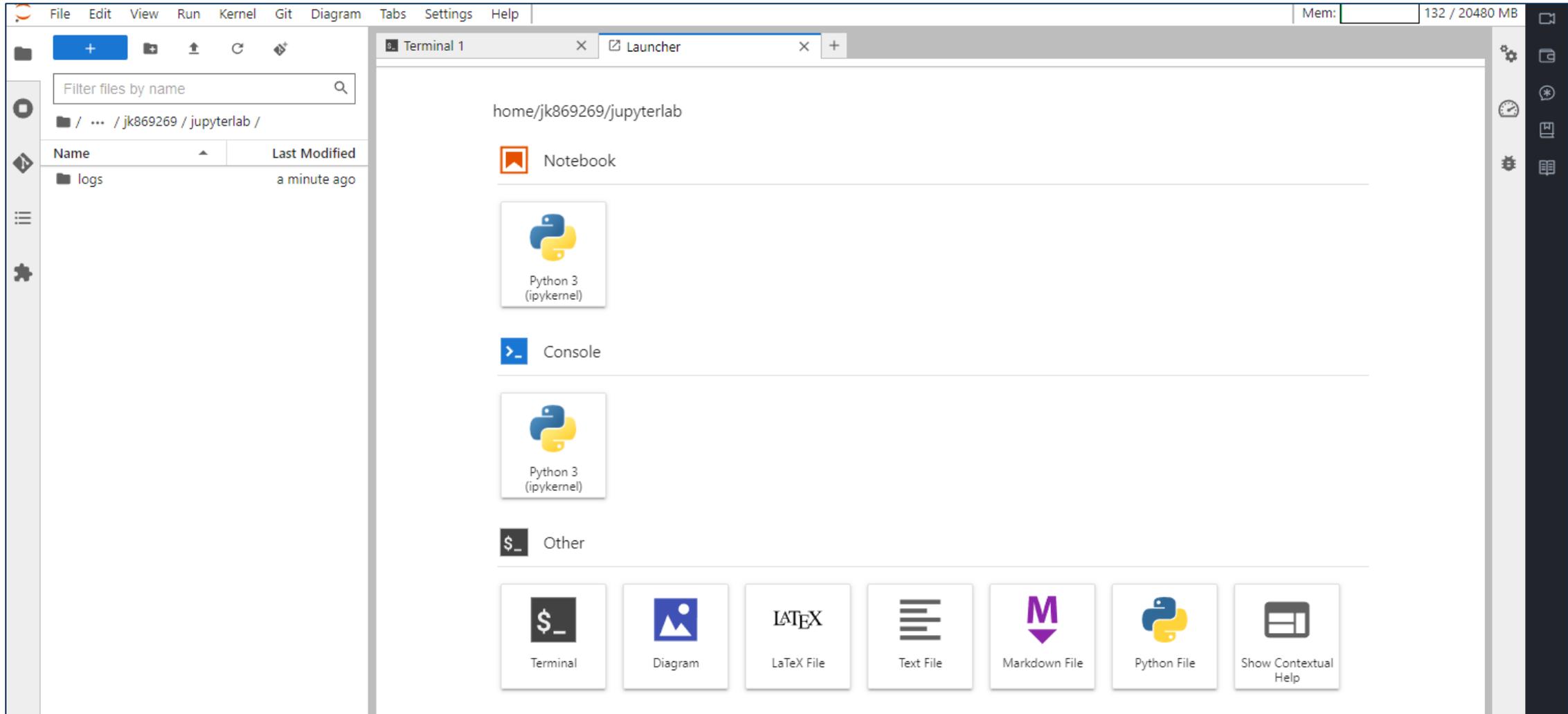
Number of Cores: (--ntasks) 2 Cores ⓘ

Node Memory: (--mem) 8 GB ⓘ

Number of GPUs: (--gres=gpu:) 1 GPU ⓘ

[Submit](#) ⓘ

Hands-on: scikit-learn – Getting Started with JupyterHub

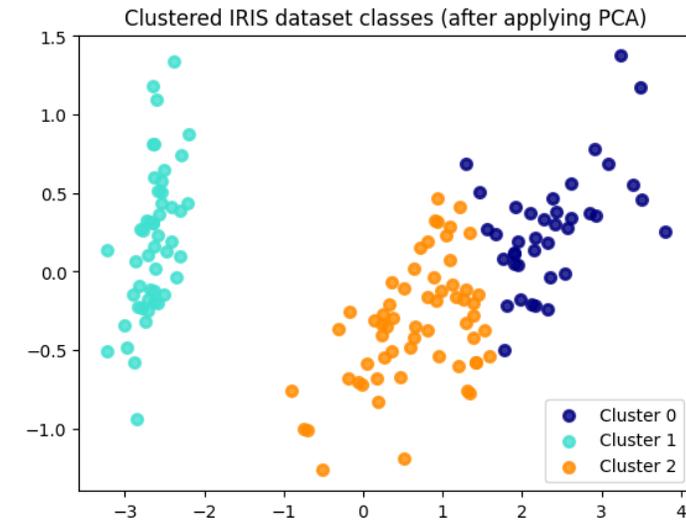


The screenshot displays the JupyterLab web interface. At the top, a menu bar includes 'File', 'Edit', 'View', 'Run', 'Kernel', 'Git', 'Diagram', 'Tabs', 'Settings', and 'Help'. The top right corner shows 'Mem: 132 / 20480 MB'. The interface is divided into three main sections:

- File Browser (Left):** Contains a search bar 'Filter files by name', a breadcrumb path '/ ... / jk869269 / jupyterlab /', and a table with columns 'Name' and 'Last Modified'. A 'logs' folder is listed with a modification time of 'a minute ago'.
- Terminal 1 (Top Middle):** Shows the current directory path 'home/jk869269/jupyterlab'.
- Launcher (Main Area):** Offers options to create a 'Notebook' or 'Console', both using a 'Python 3 (ipykernel)' environment. Below these are 'Other' options represented by icons for Terminal, Diagram, LaTeX File, Text File, Markdown File, Python File, and Show Contextual Help.

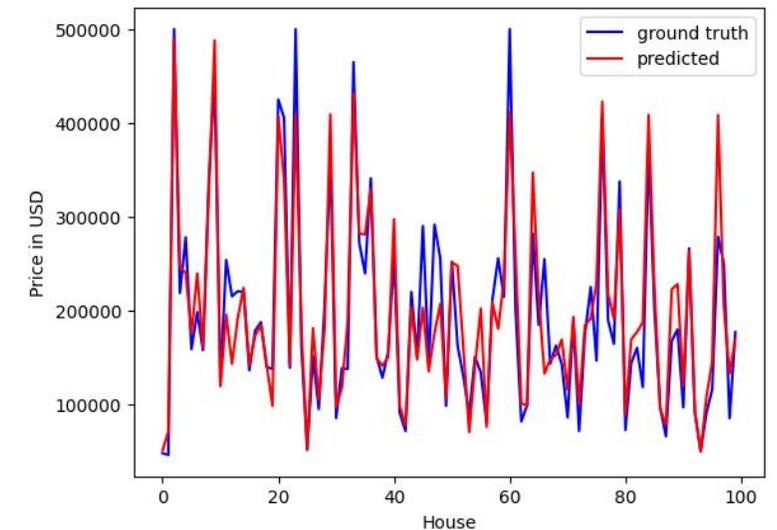
– Exercise 1: Clustering with Iris Dataset

- Load dataset
- Train clustering model (here: Kmeans clustering)
- Apply dimensionality reduction (PCA)
- Visualize and compare against true labels



– Exercise 2: Regression with California Housing Dataset

- Load dataset
- Apply preprocessing techniques (Standardization)
- Train regression model (here: RandomForest)
- Model performance evaluation and visualization



Part 2: Hands-on: PyTorch

– Exercise: Train ResNet model with CIFAR-10 dataset

– Model: ResNet

- Popular model for image classification
- Winner of ILSVRC 2015 (ImageNet Large Scale Visual Recognition Challenge)
- Tackles vanishing gradient problem

– Dataset: CIFAR-10

- Several images of 10 different classes
- Airplane, automobile, bird, cat, dog, frog, ...

– Single GPU (Task 1), Distributed (Task 2)

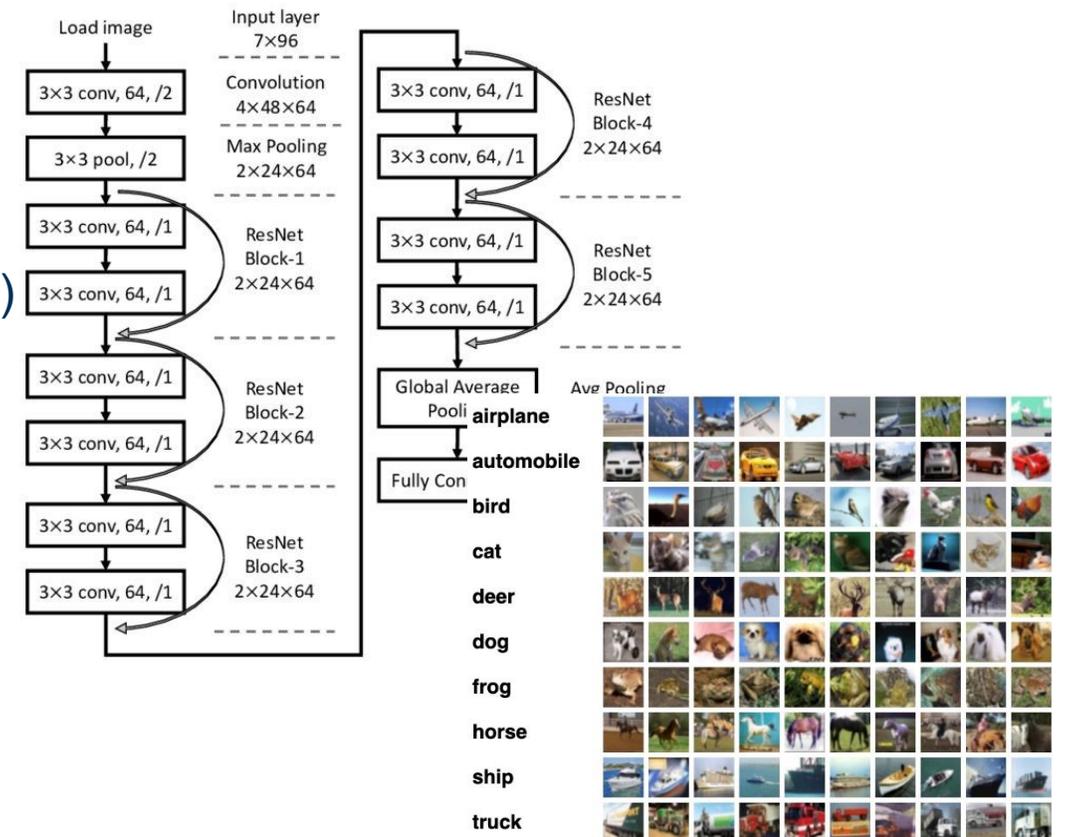


Diagram: Choi, Hyungeun & Ryu, Seunghyoung & Kim, Hongseok. (2018). Short-Term Load Forecasting based on ResNet and LSTM

- **Note:** We will use a container for this exercise!

	File	Description
Provided by us	set_vars.sh	Shell script that sets environment variables for DDP and container
	submit_job.sh	Submits a batch job to SLURM, which loads the required container module and executes the Python code
Your job	train_model.py	Python code that is responsible for training and testing the model

- **About: submit_job.sh**

- Option to run with 1 GPU or 2 GPUs (distributed)
- Option to enable explicit monitoring (nvidia-smi)

```
# execute with a single GPU
sbatch submit_job.sh

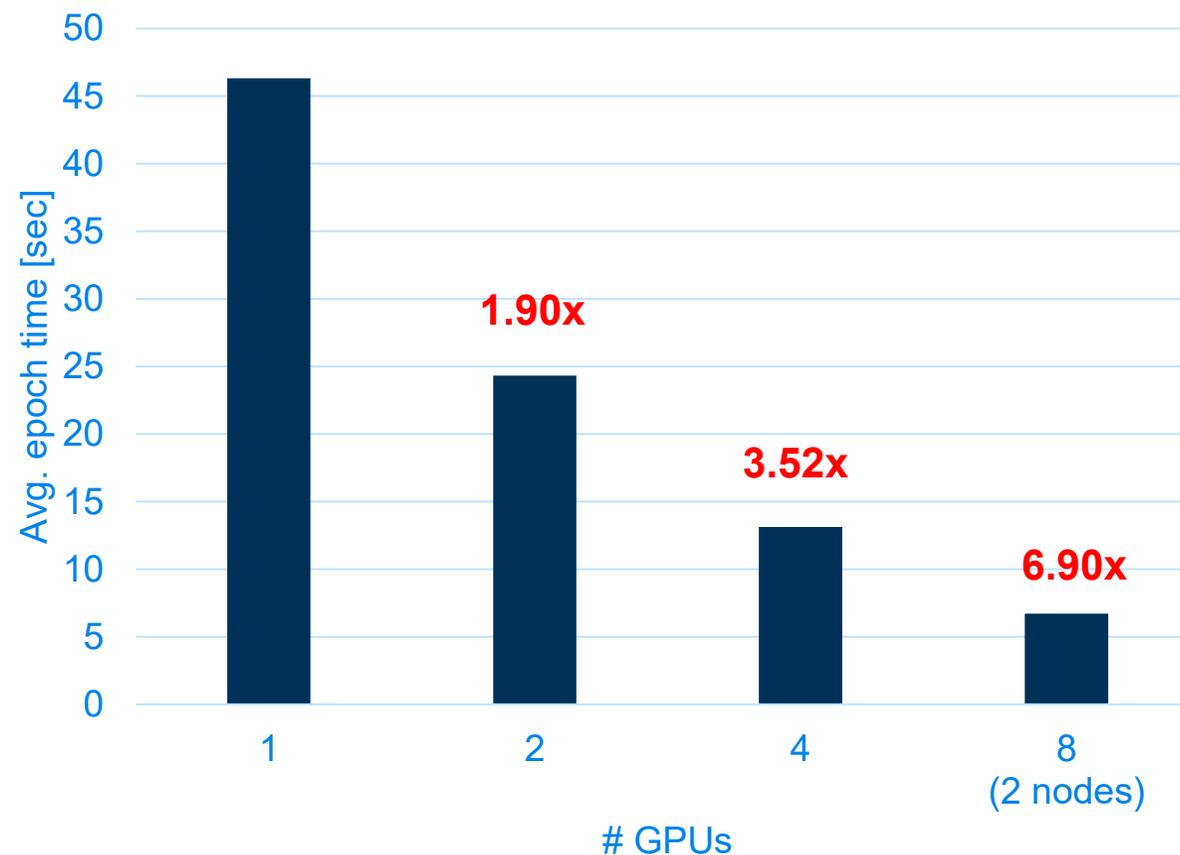
# execute with nvidia-smi monitoring
sbatch --export=ENABLE_MONITORING=1 submit_job.sh

# execute with 2 GPUs (distributed)
sbatch --ntasks-per-node=2 --gres=gpu:2 submit_job.sh

# check for status
squeue --me -a
```

– Output and scaling behavior

```
...  
Epoch 2/2 Step 0 / 391  
Epoch 2/2 Step 20 / 391  
Epoch 2/2 Step 40 / 391  
Epoch 2/2 Step 60 / 391  
Epoch 2/2 Step 80 / 391  
Epoch 2/2 Step 100 / 391  
Epoch 2/2 Step 120 / 391  
Epoch 2/2 Step 140 / 391  
Epoch 2/2 Step 160 / 391  
Epoch 2/2 Step 180 / 391  
Epoch 2/2 Step 200 / 391  
Epoch 2/2 Step 220 / 391  
Epoch 2/2 Step 240 / 391  
Epoch 2/2 Step 260 / 391  
Epoch 2/2 Step 280 / 391  
Epoch 2/2 Step 300 / 391  
Epoch 2/2 Step 320 / 391  
Epoch 2/2 Step 340 / 391  
Epoch 2/2 Step 360 / 391  
Epoch 2/2 Step 380 / 391  
Epoch 2/2 Elapsed: 47.857 sec Acc: 0.642  
Epoch 2/2 Test Acc: 0.648
```



- **Problem:** PyTorch throws errors when executing distributed variant
 - “Port already in use”
 - **Reasons:** You will potentially be working on a shared machine (e.g., with only 2 of 4 GPUs) and other users are occupying that port on the machine.
- **Solution:** Select a different port

- **Problem:** Be careful with using `$ (pwd)` or `$PWD` inside a container
 - This might return `/rwthfs/...` which is not mounted by default
 - What's interesting:
 - `/work/<user-id>/` is mounted by default
 - `/rwthfs/rz/cluster/work/<user-id>/` is not although pointing to the same path
 - **Cause:** Can occur when working with multiplexers like `tmux` on the host system
- **Solution:**
 - Map the directories that you want to use in the container
 - Also see hands-on examples