

Deep Learning with GPU cores

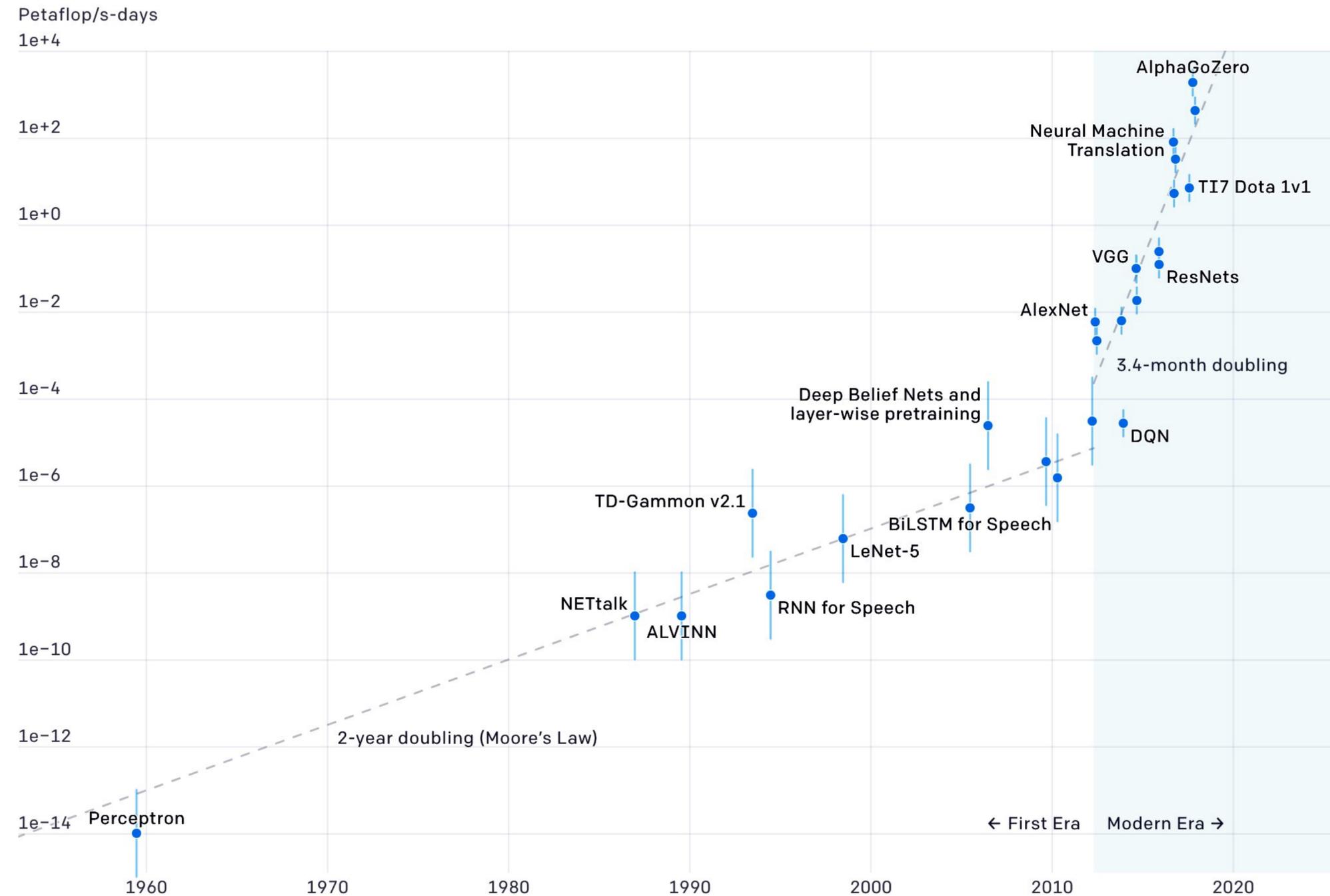
How to do more in less time

Mohammad Hossein Biniaz | GWDG | 22. August 2024

Credits: Dorothea Sommer

Why Deep Learning + GPUs?

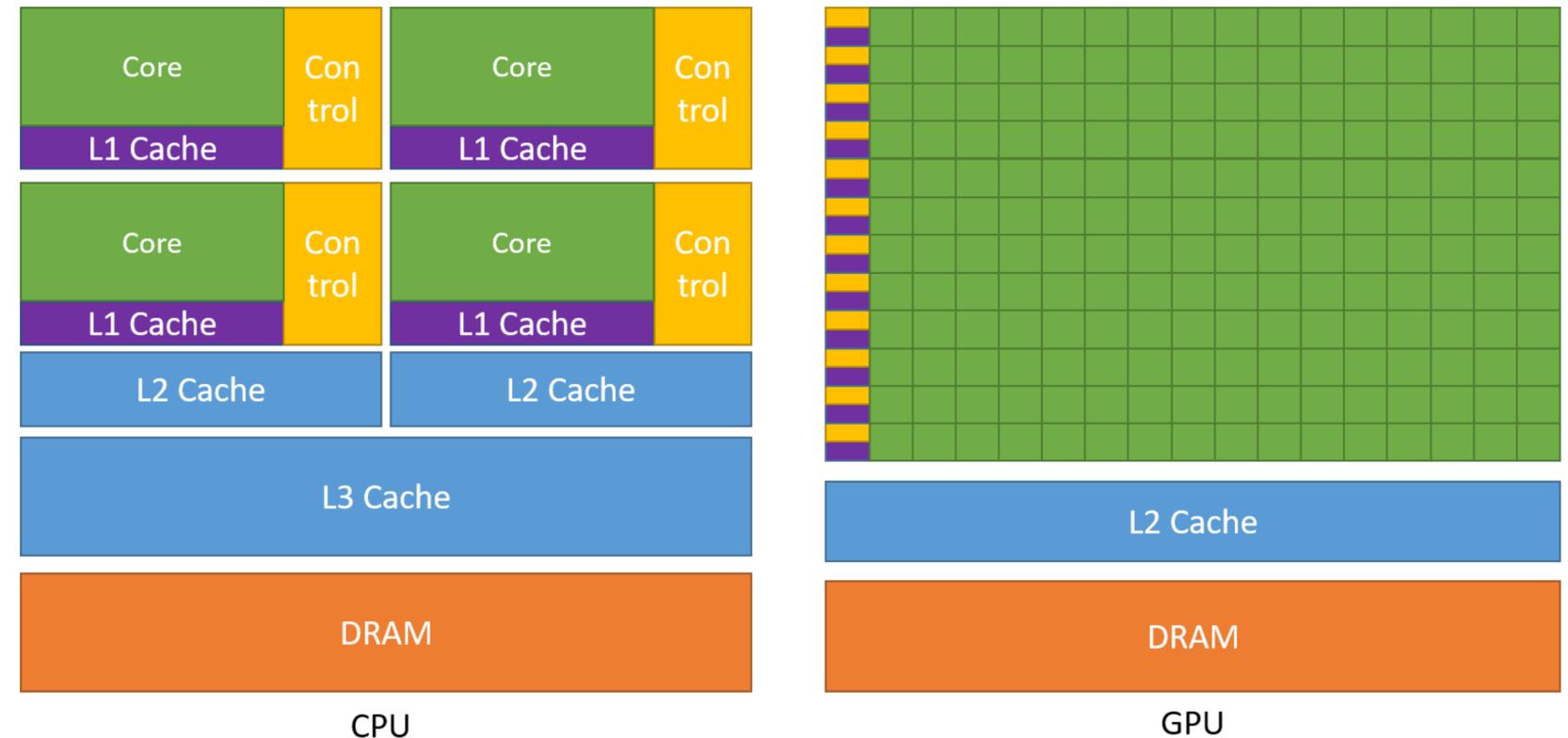
- Deep learning is bound by compute power.
- GPU enable efficient training of neural networks.



Monitoring

Basic GPU ideas: For what are GPUs efficient?

- **SIMD: single instruction multiple data**
- **GPUs are efficient at running the same operation on a large number of elements (i.e., running a lot of threads simultaneously).**



Internal comparison between a CPU and a GPU.

<https://docs.nvidia.com/cuda/cuda-c-programming-guide/>

What we'll do

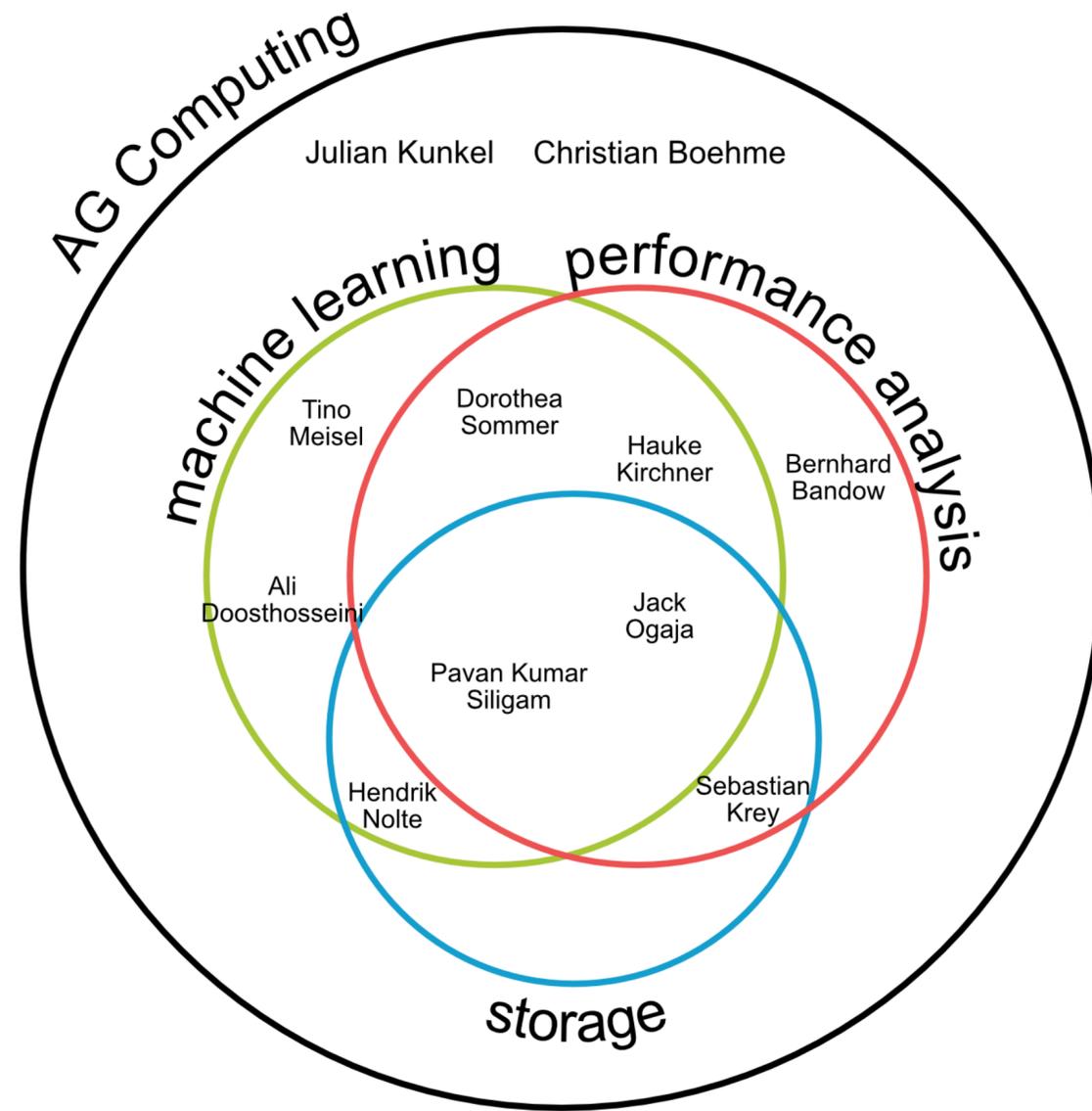
	Deep Learning with GPU cores	
09.30 - 09.45	Welcome	
09.45 - 10.15 (30 min)	Deep Learning and Infrastructure	Learn how to train a neural network with a GPU.
10.15 - 11.30 (60 min)	Practical: Working on the GPU	
11.30 - 11.45	Short break ☕	
11.45 - 12.00 (15 min)	Introduction to Containers	Learn how to use deep-learning containers, especially on the HPC
12.00 - 12.45 (45 min)	Practical: Containers	
12.45 - 13.00	General Q&A	

Why do we offer this course?



- **Working Group “Computing”** at the GWDG
- Work in conjunction with University Göttingen and Max Planck Society
- **Mission:** provide scalable solutions for resource-intensive applications
 - Independent research in the field of computer science
 - Planning, operation, hosting and housing of HPC systems
 - Training around the use of HPC systems

Working Group Computing



We are currently 40 persons.
You can become part of our team!

We are
... supervising theses.
... hiring.

Mohammad Hossein Biniiaz

- Data Scientist at AG Computing
- Experience
 - B.Sc. Mathematics & Applications (K. N. TU, Tehran, Iran)
 - M.Sc. Environmental Engineering (清华大学, Tsinghua, Beijing, China)
 - M.Sc. Mathematical Data Science (Göttingen)
 - Computational Optimal Transport Algorithm in one dimension for Single-Cell Gene Expression Analysis (Master's thesis)
- Research Interests
 - Scalable AI / MLOPs / Deep learning
 - Computational finance
 - Parallel programming
 - Software engineering for scientific software
 - Scientific algorithms



Hauke Kirchner

- Data Scientist at AG Computing
- Experience
 - B.Sc. Biology (Göttingen)
 - M.Sc. Forest Information Technology (Eberswalde, Warschau)
 - Tree species classification from airborne LiDAR using individual crowndelineation and machine learning (Master Thesis, UFZ)
- Research Interest
 - Machine Learning
 - Remote sensing
 - Forest science
 - Data management



Ali Doosthosseini

- AI & Data Scientist at AG Computing
- Experience
 - M. Sc. Applied Computer Science w/ Data Science Specialization
 - Synthetic Point Cloud Generation and Tree Segmentation (Master Thesis)
- Research Interests
 - AI & Machine Learning
 - Natural Language Processing
 - Computational Neuroscience



Who are you?

<https://take.supersurvey.com/QP06XTPPC>



What we'll do

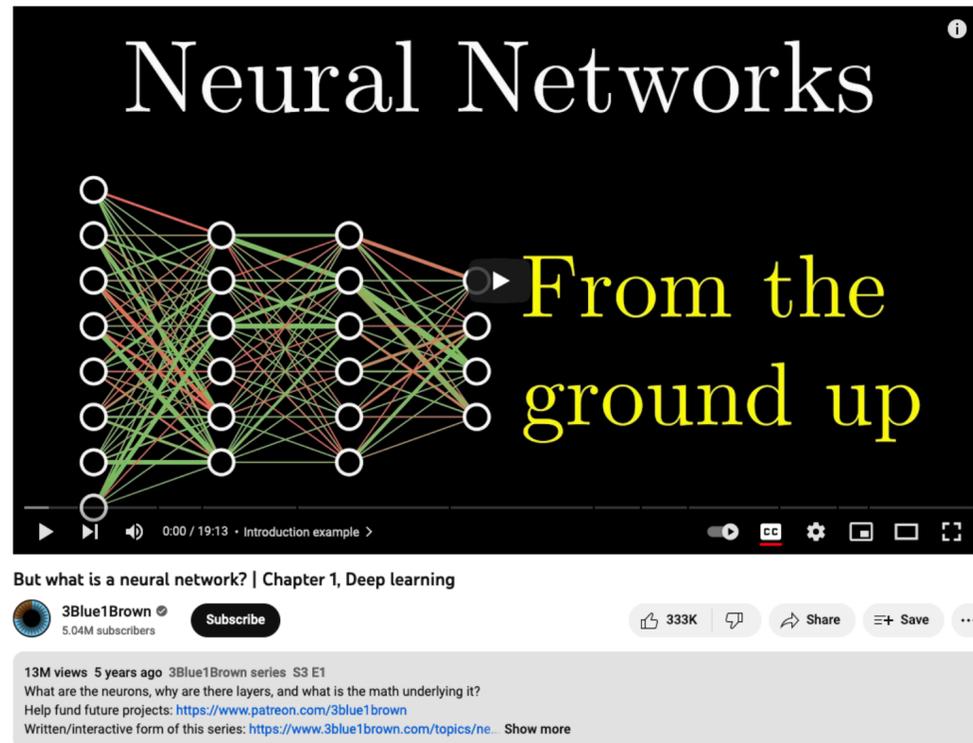
	Deep Learning with GPU cores	
09.30 - 09.45	Welcome	
09.45 - 10.15 (30 min)	Deep Learning and Infrastructure	Learn how to train a neural network with a GPU.
10.15 - 11.30 (60 min)	Practical: Working on the GPU	
11.30 - 11.45	Short break ☕	
11.45 - 12.00 (15 min)	Introduction to Profiling	Learn how to profile the training and training efficiently.
12.00 - 12.45 (45 min)	Practical: Profiling Jobs	
12.45 - 13.00	General Q&A	

Deep Learning and Infrastructure

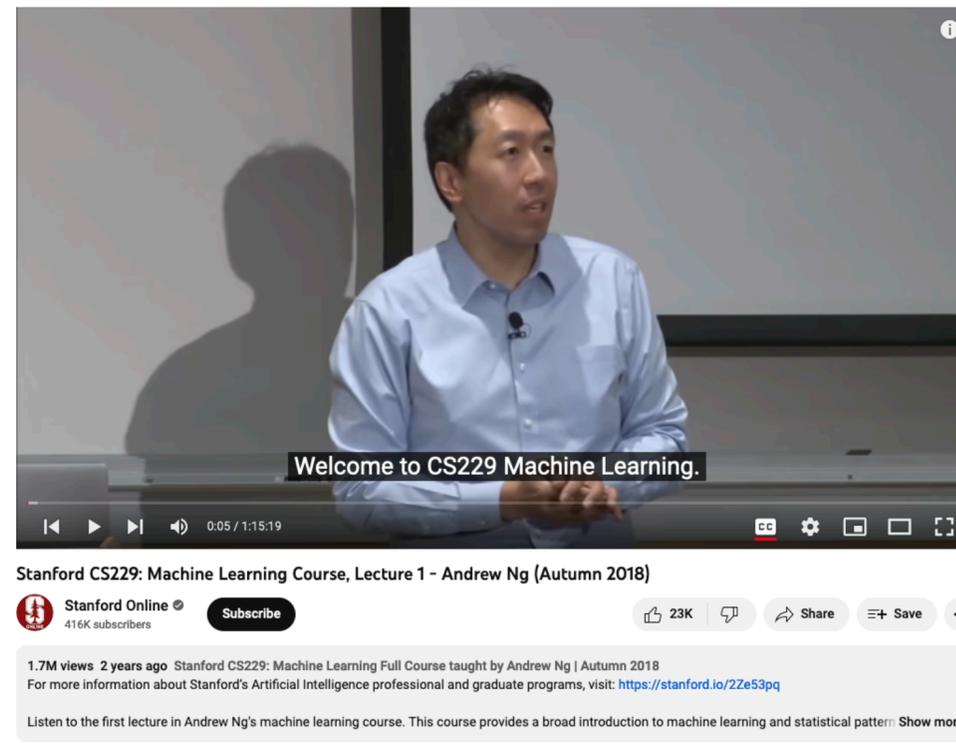
Deep Learning Example

Going further

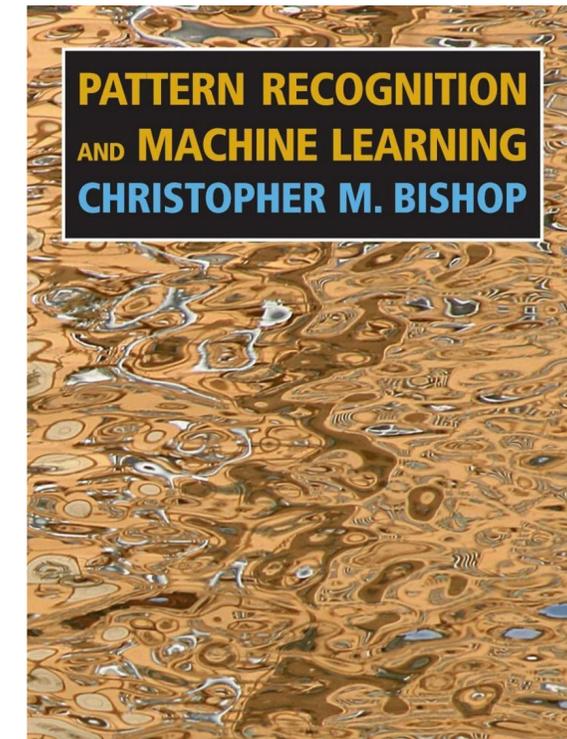
Requires more effort (time and math)



- YouTube series by **3Blue1Brown**:
- explains the math behind neural networks intuitively
 - amazing visualisations



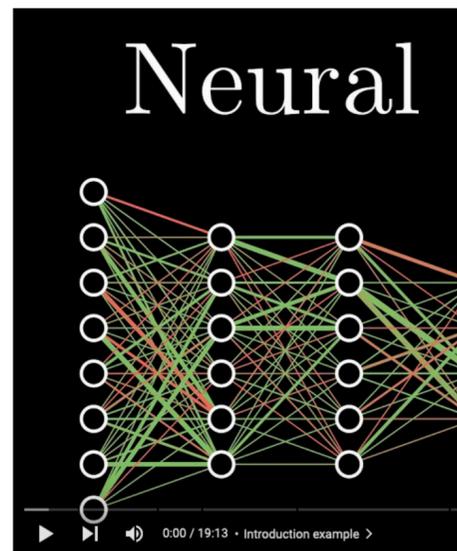
- YouTube series by **Andrew Ng**:
- clear explanation of concepts
 - involves basic math
 - use older series if you like more math



- Classic book by **Christopher Bishop**:
- good if you have a solid math background

Going further

Requires more effort (time and math)



But what is a neural network? | Chapter 1, Deep learning
3Blue1Brown
5,04M subscribers
13M views 5 years ago
3Blue1Brown series S3 E1
What are the neurons, why are there layers, and what is the math underlying...
Help fund future projects: <https://www.patreon.com/3blue1brown>
Written/interactive form of this series: <https://www.3blue1brown.com/topics>

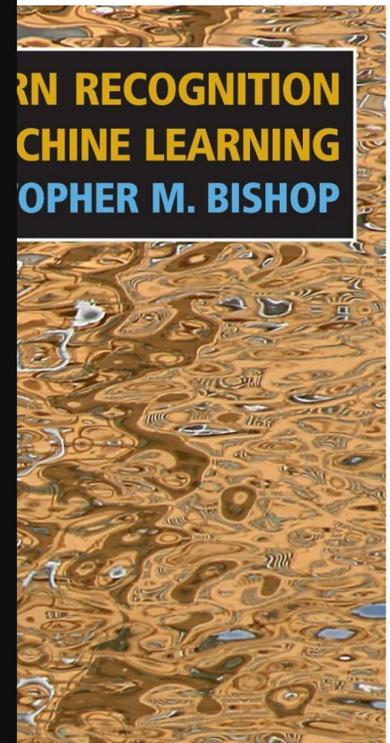
- explains the math of neural networks intuitively
- amazing visualizations

General Audience
videos for more general audience, no programming experience necessary.

[1hr Talk] Intro to Large Language Models
Andrej Karpathy • 1.7M views • 4 months ago
This is a 1 hour general-audience introduction to Large Language Models: the core technical component behind systems like ChatGPT, Claude, and Bard. What they are, where they are headed, comparison...

Neural Networks: Zero to Hero ▶ Play all

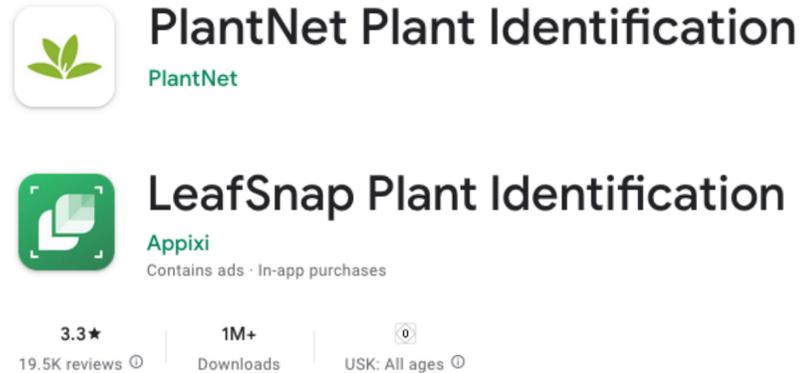
- The spelled-out intro to neural networks and...**
Andrej Karpathy
1.4M views • 1 year ago
- The spelled-out intro to language modeling: building...**
Andrej Karpathy
553K views • 1 year ago
- Building makemore Part 2: MLP**
Andrej Karpathy
258K views • 1 year ago
- Building makemore Part 3: Activations & Gradients,...**
Andrej Karpathy
229K views • 1 year ago



book by **Christopher Bishop**:
if you have a math background

Learning Paradigm

Let's build...



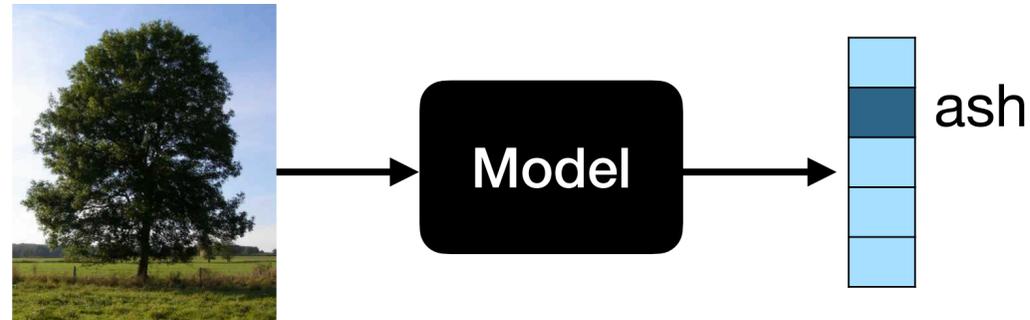
Snap picture of tree to identify species

Use Case

tree species classification

\underline{x}_i tree photo

y_i corresponding tree species



classification
predict tree species
given photo of the tree

Supervised Learning

"learning with teacher"

Observations $\underline{x}_1, \dots, \underline{x}_n$

Labels y_1, \dots, y_n



predict label of observation

regression, classification

Deep Learning

Let's build...



PlantNet Plant Identification

PlantNet



LeafSnap Plant Identification

Appixi

Contains ads - In-app purchases

3.3★

19.5K reviews

1M+

Downloads

USK: All ages

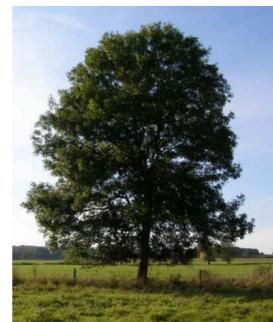
Snap picture of tree
to identify species

Use Case

tree species classification

\underline{x}_i tree photo

y_i corresponding tree species



Neural
Network

ash

classification

predict tree species

given photo of the tree

Supervised Learning

"learning with teacher"

Observations $\underline{x}_1, \dots, \underline{x}_n$

Labels y_1, \dots, y_n

\underline{x}_i

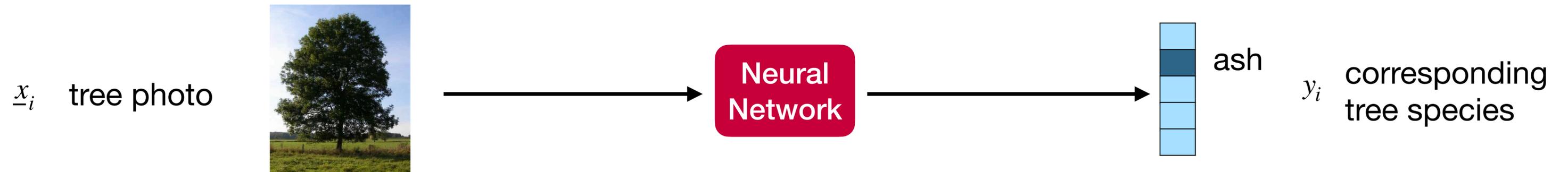
Neural
Network

y_i

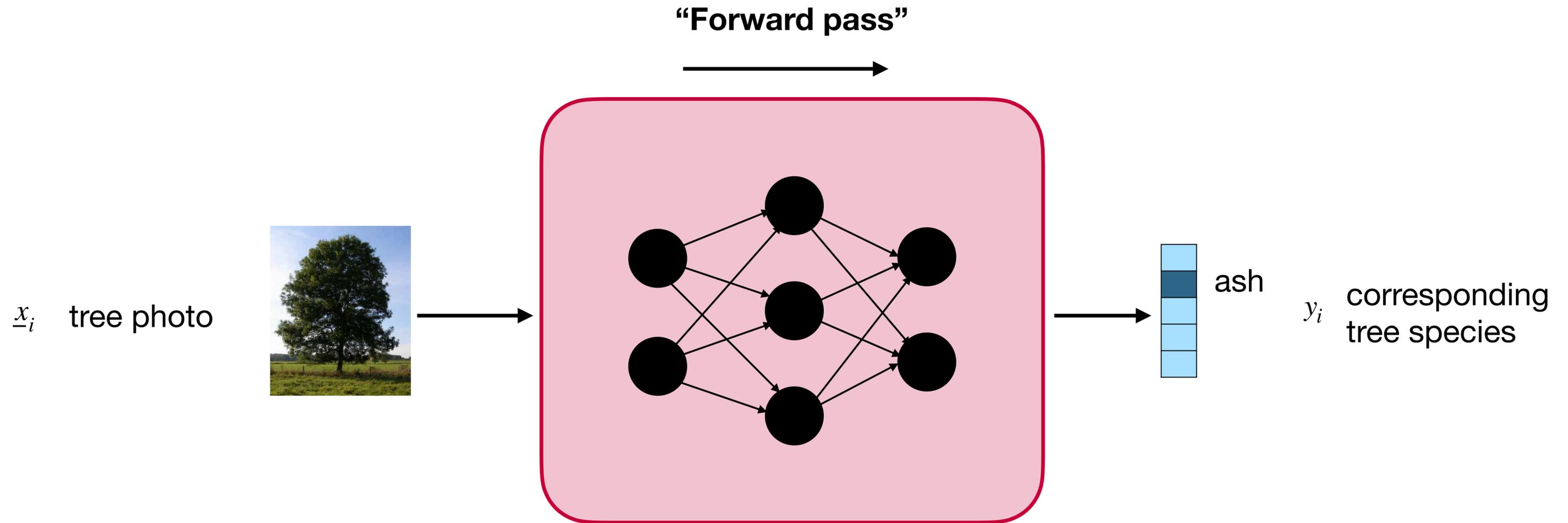
predict label of observation

regression, classification

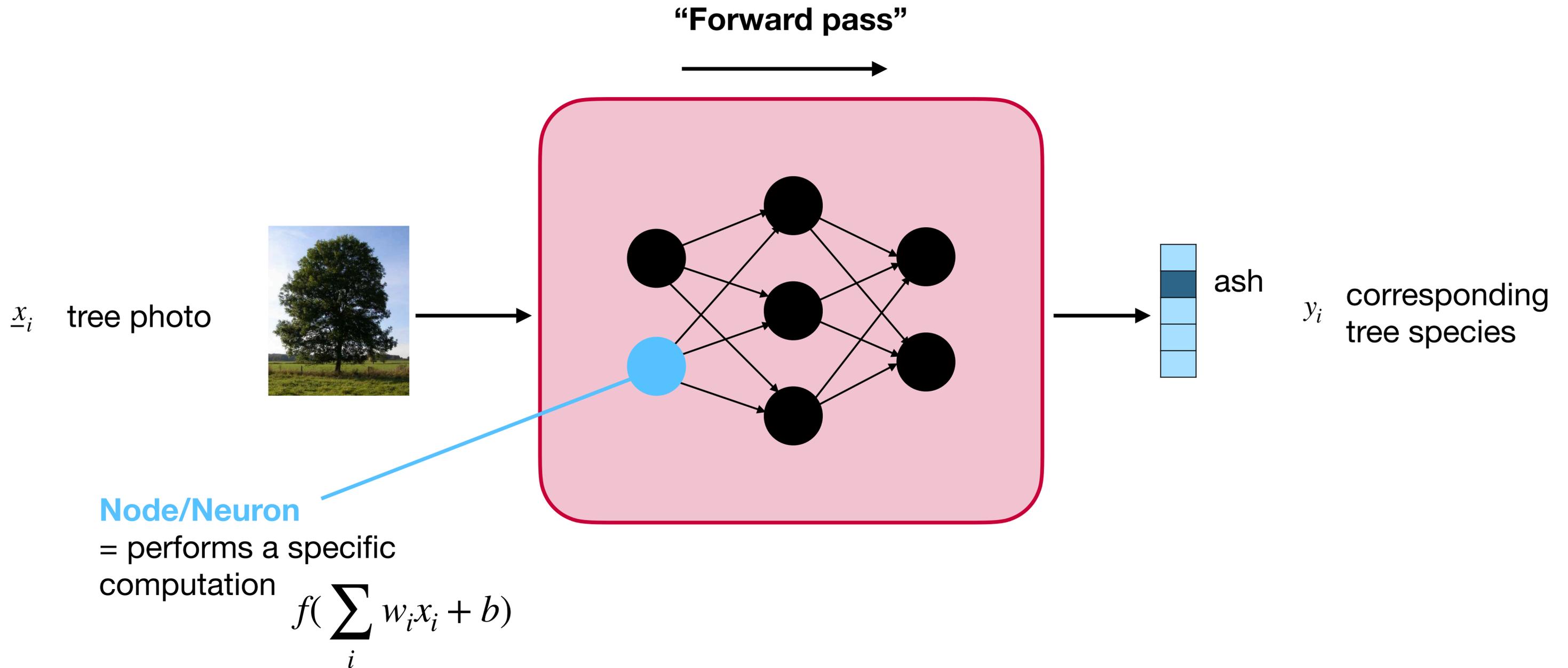
What happens in the network?



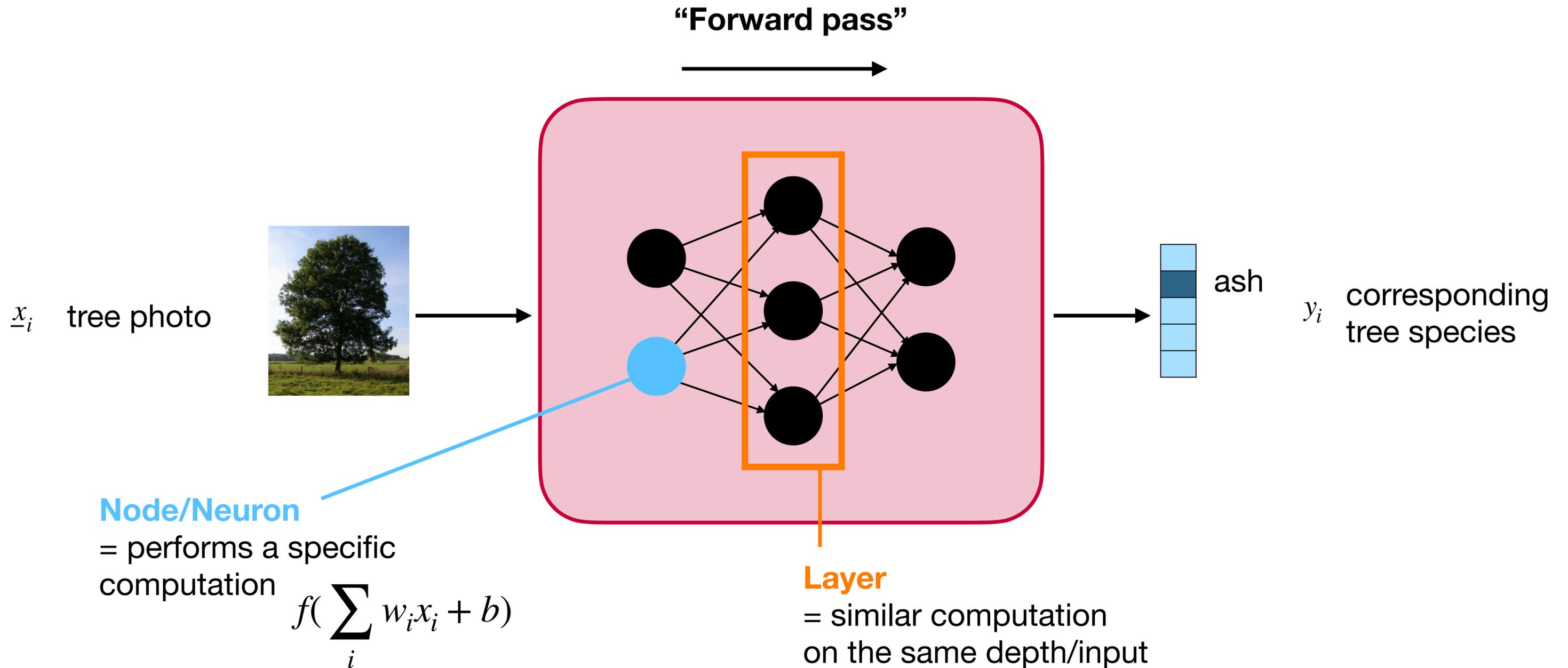
What happens in the network?



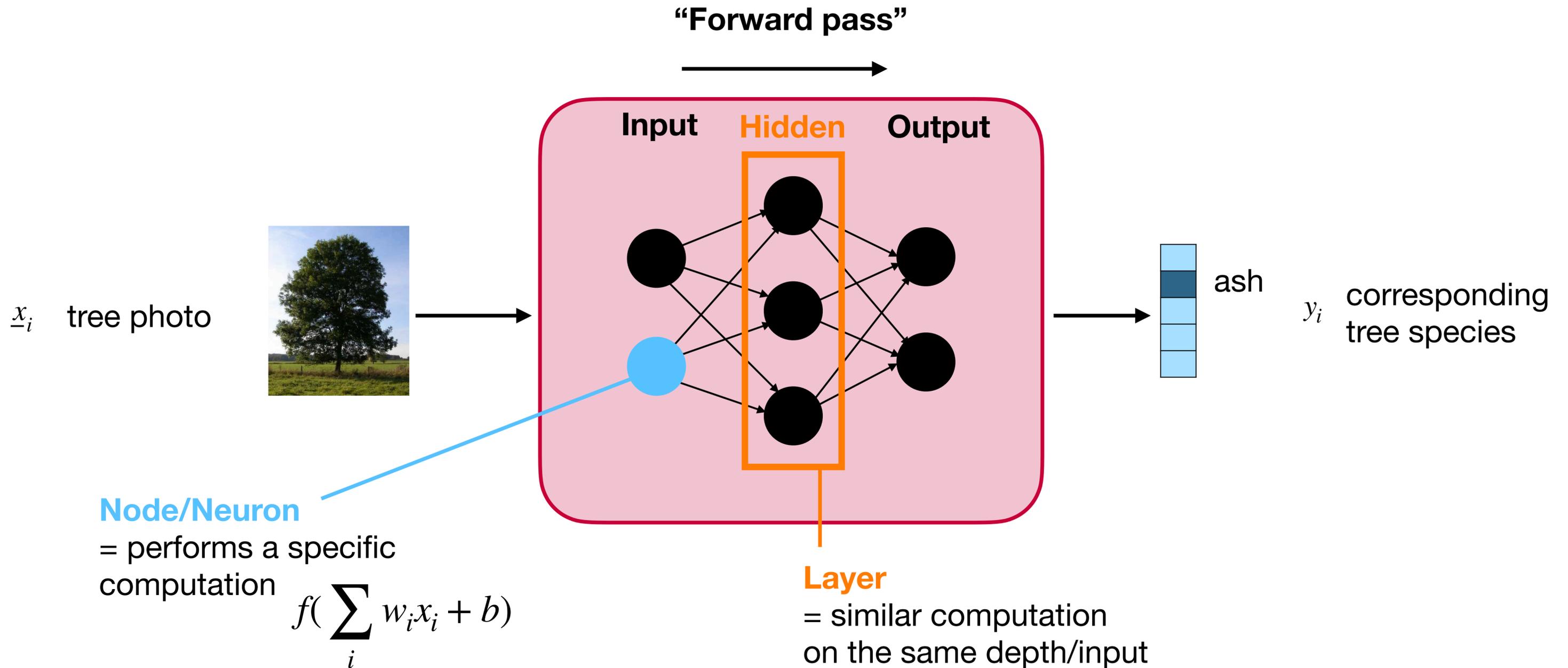
What happens in the network?



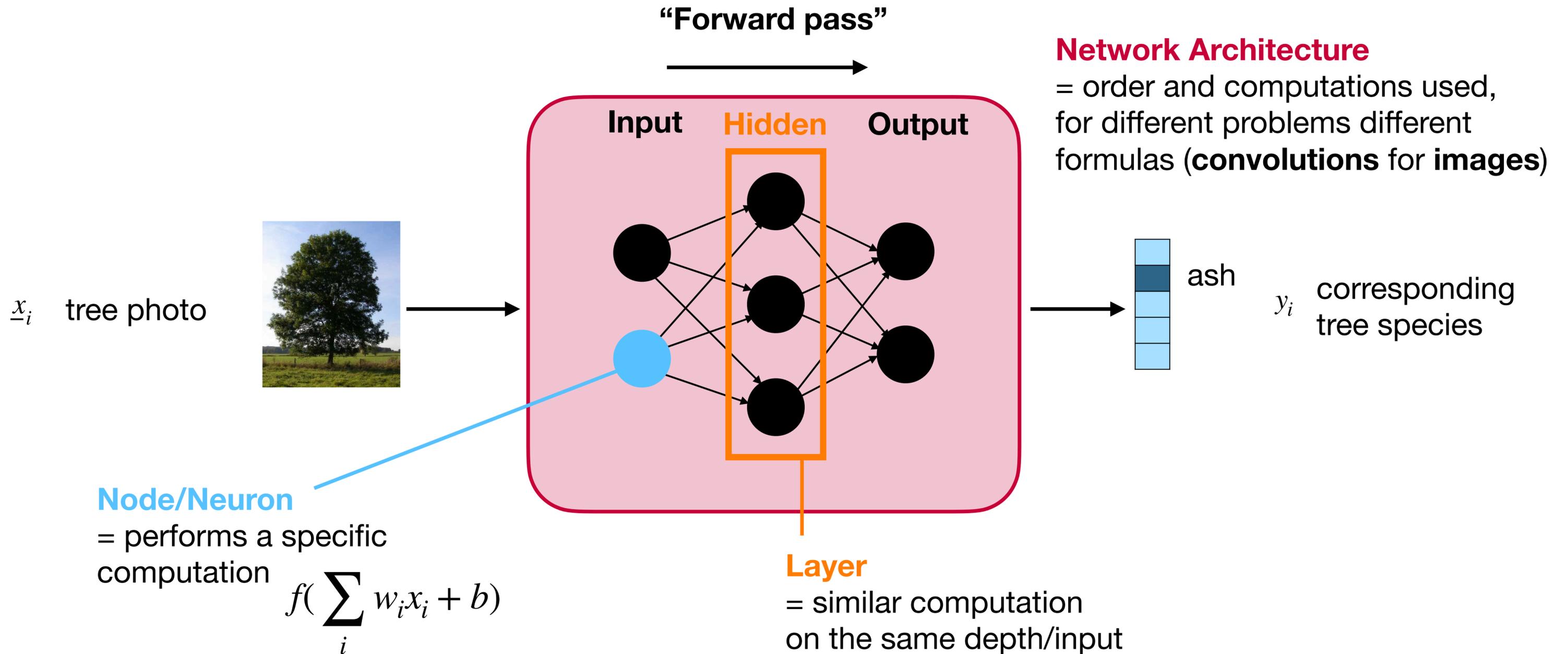
What happens in the network?



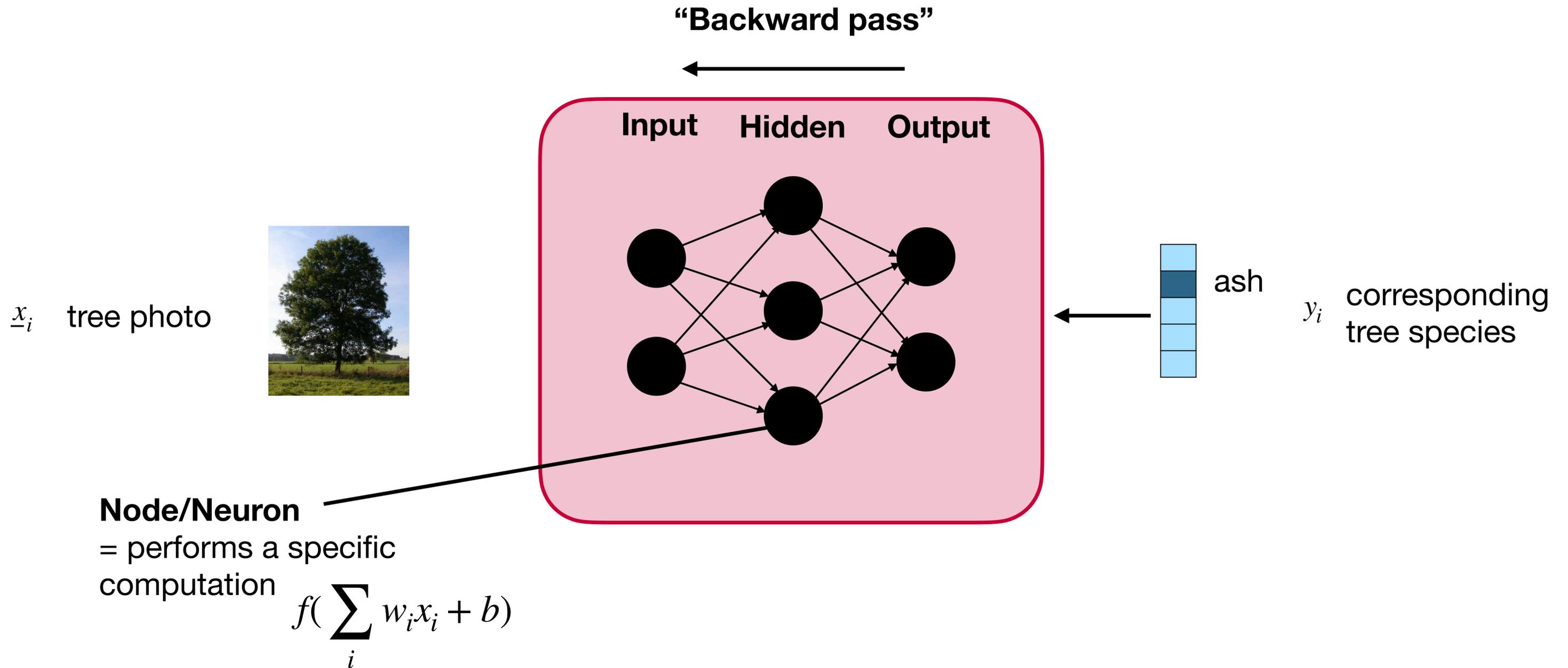
What happens in the network?



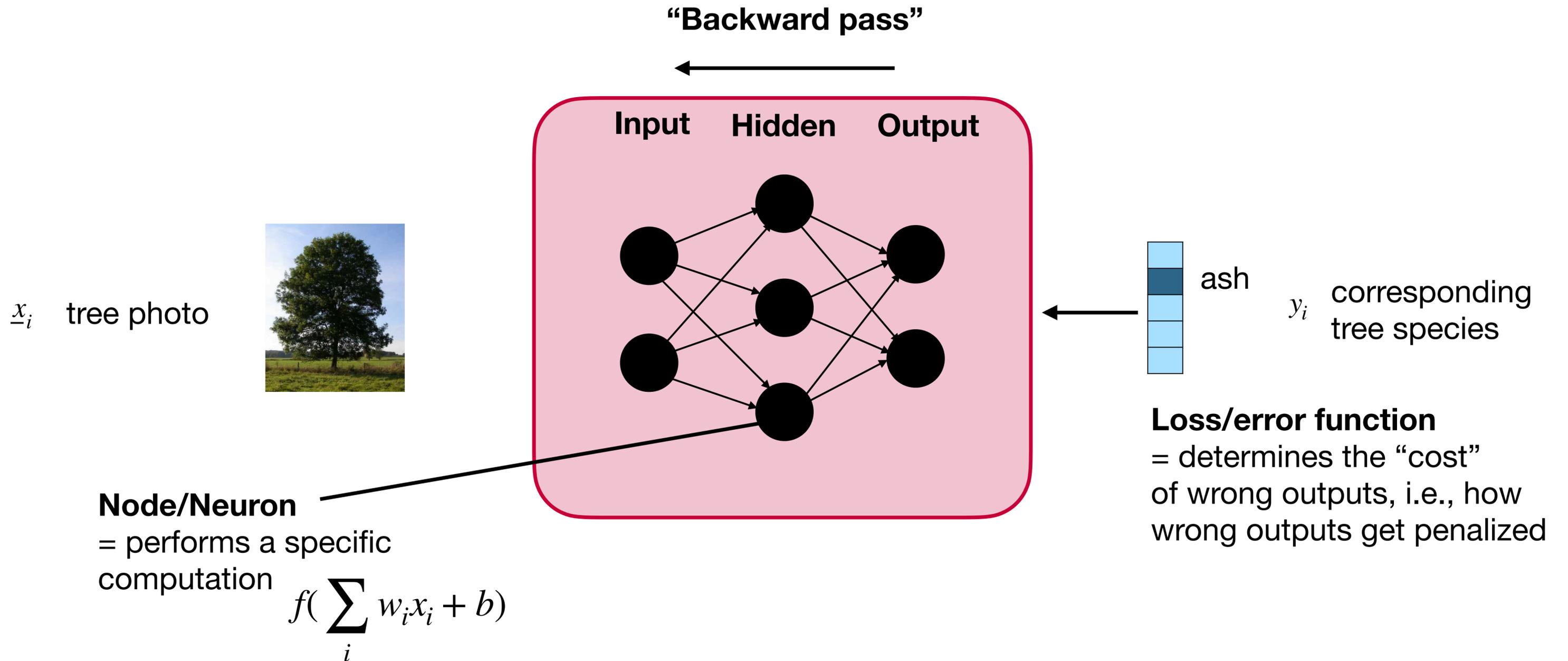
What happens in the network?



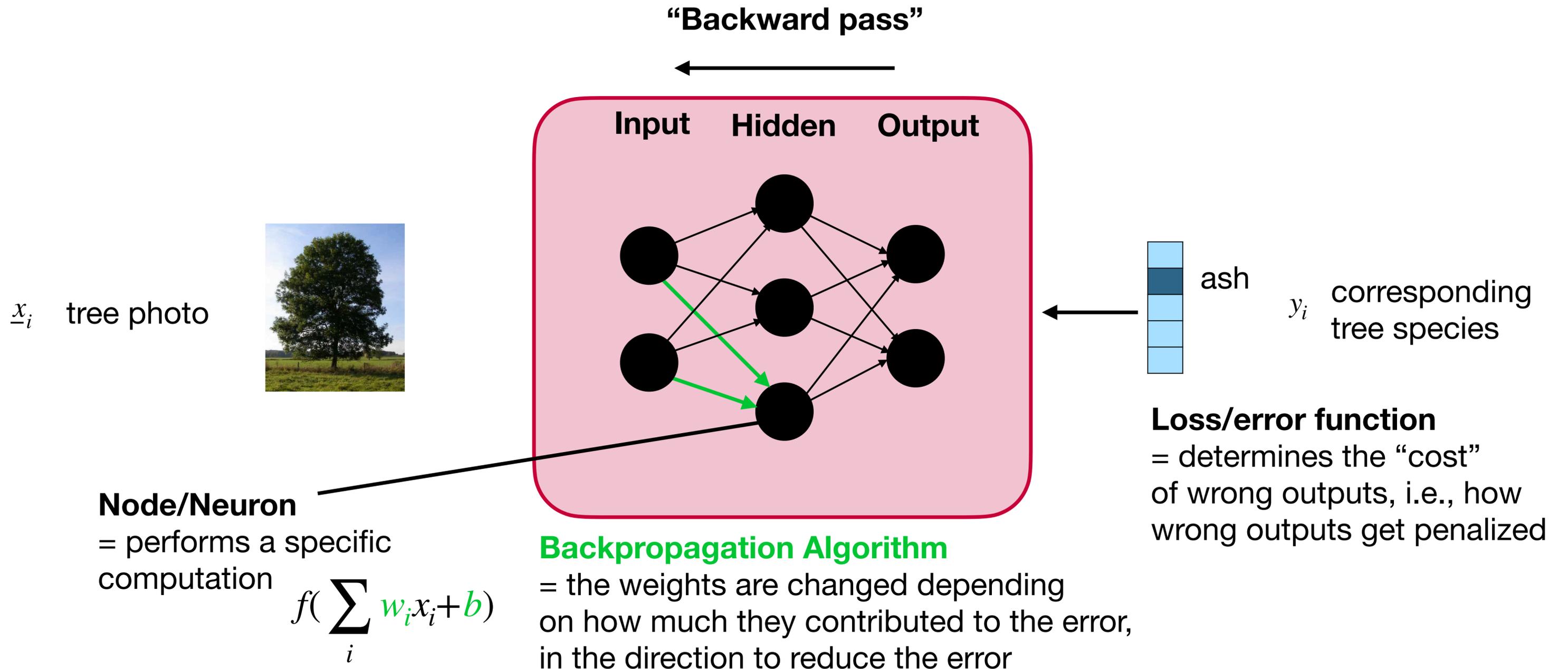
What happens in the network?



What happens in the network?

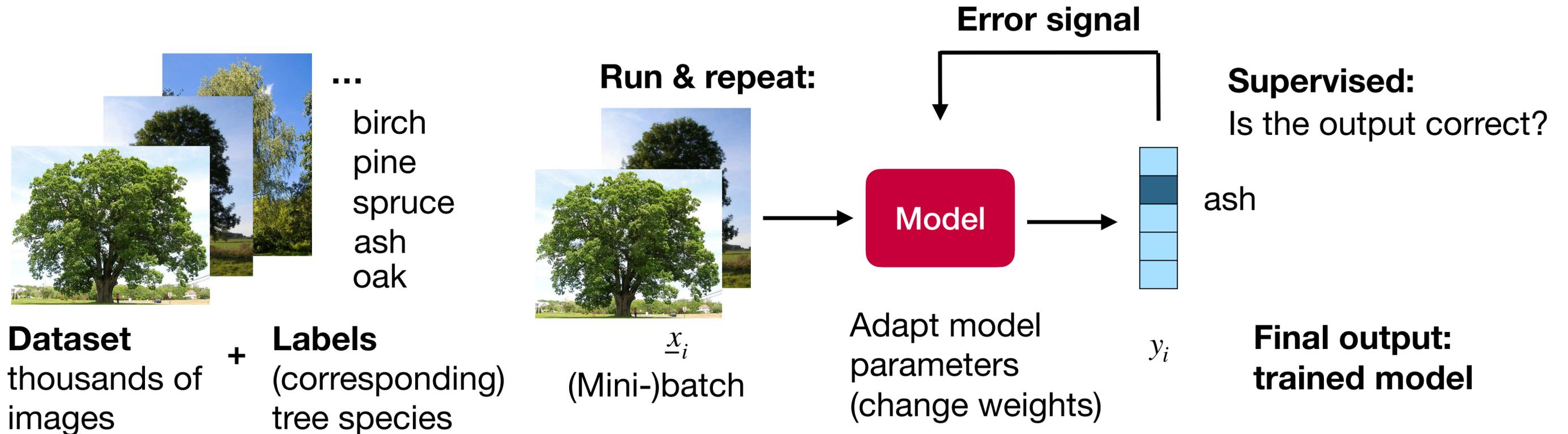


What happens in the network?

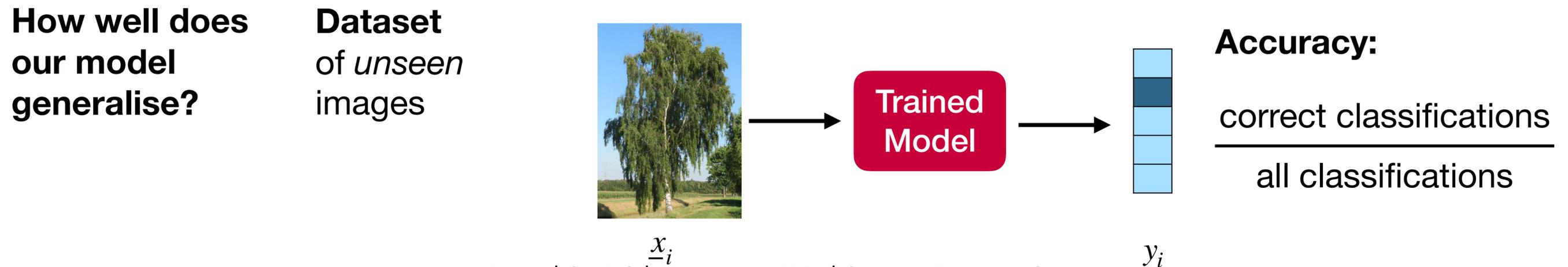


Deep Learning: Procedure

Training

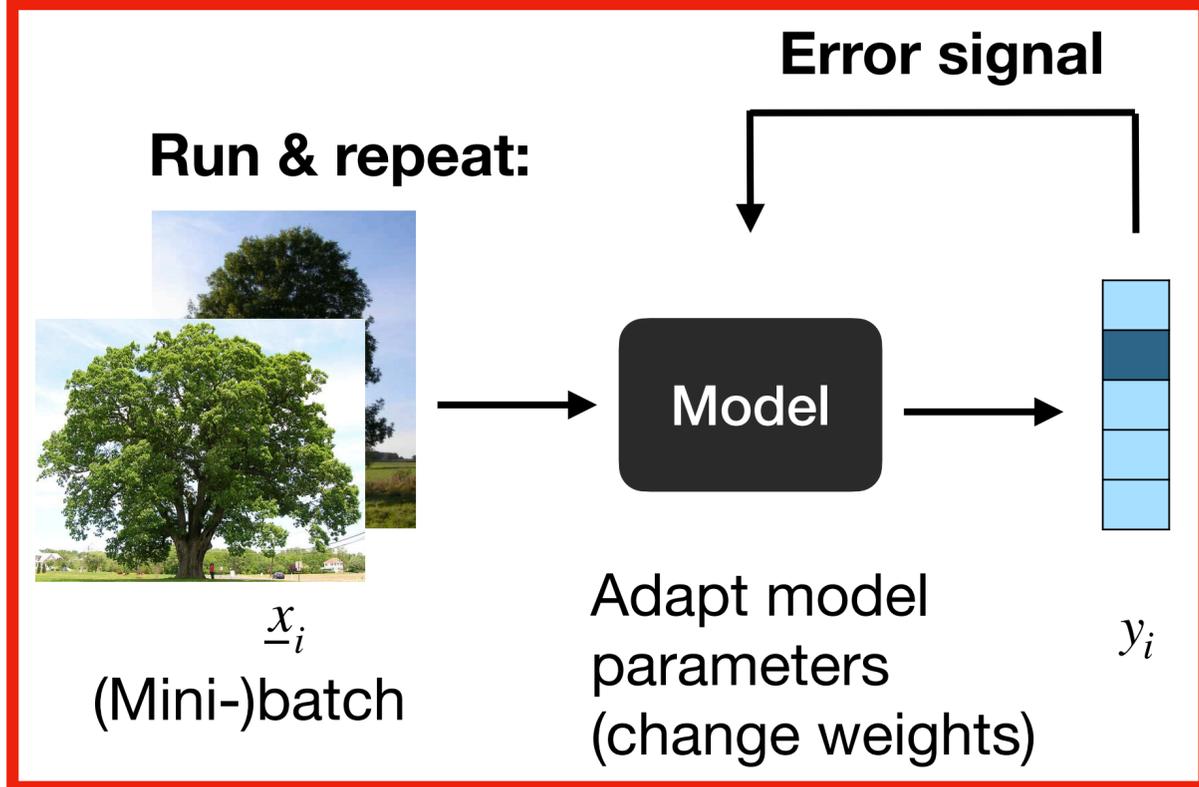
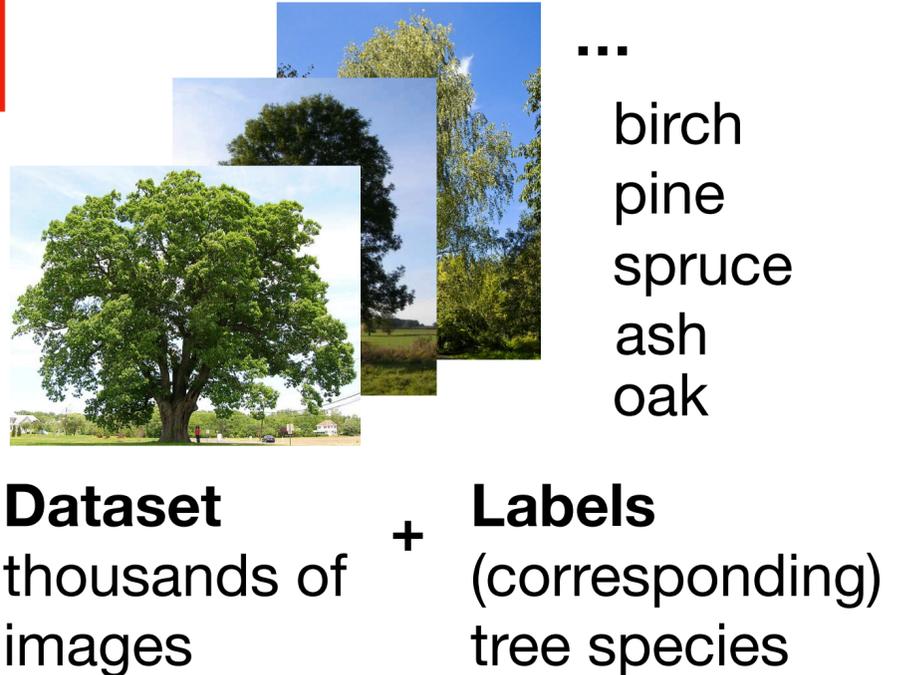


Testing



Where do we need GPUs?

Training



Supervised:
Is the output correct?

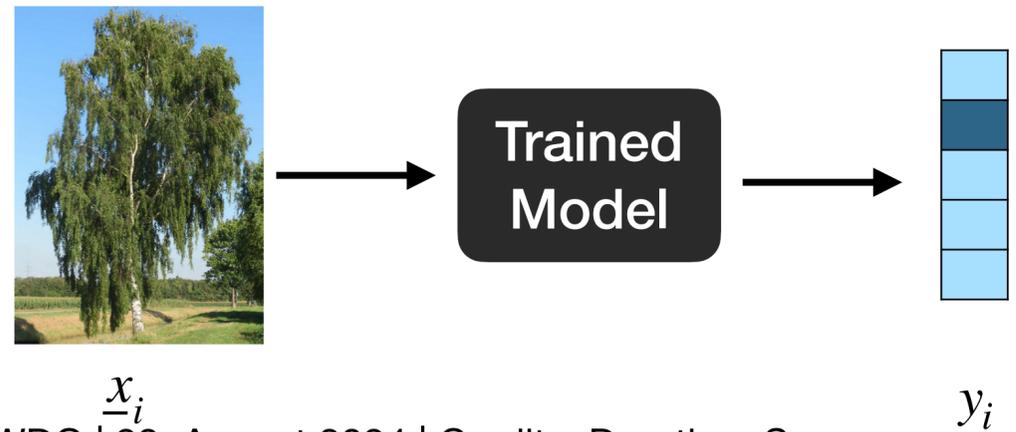
Final output:
trained model
(model parameters)

Testing

How well does our model generalise?

Dataset
of *unseen* images

Inference:



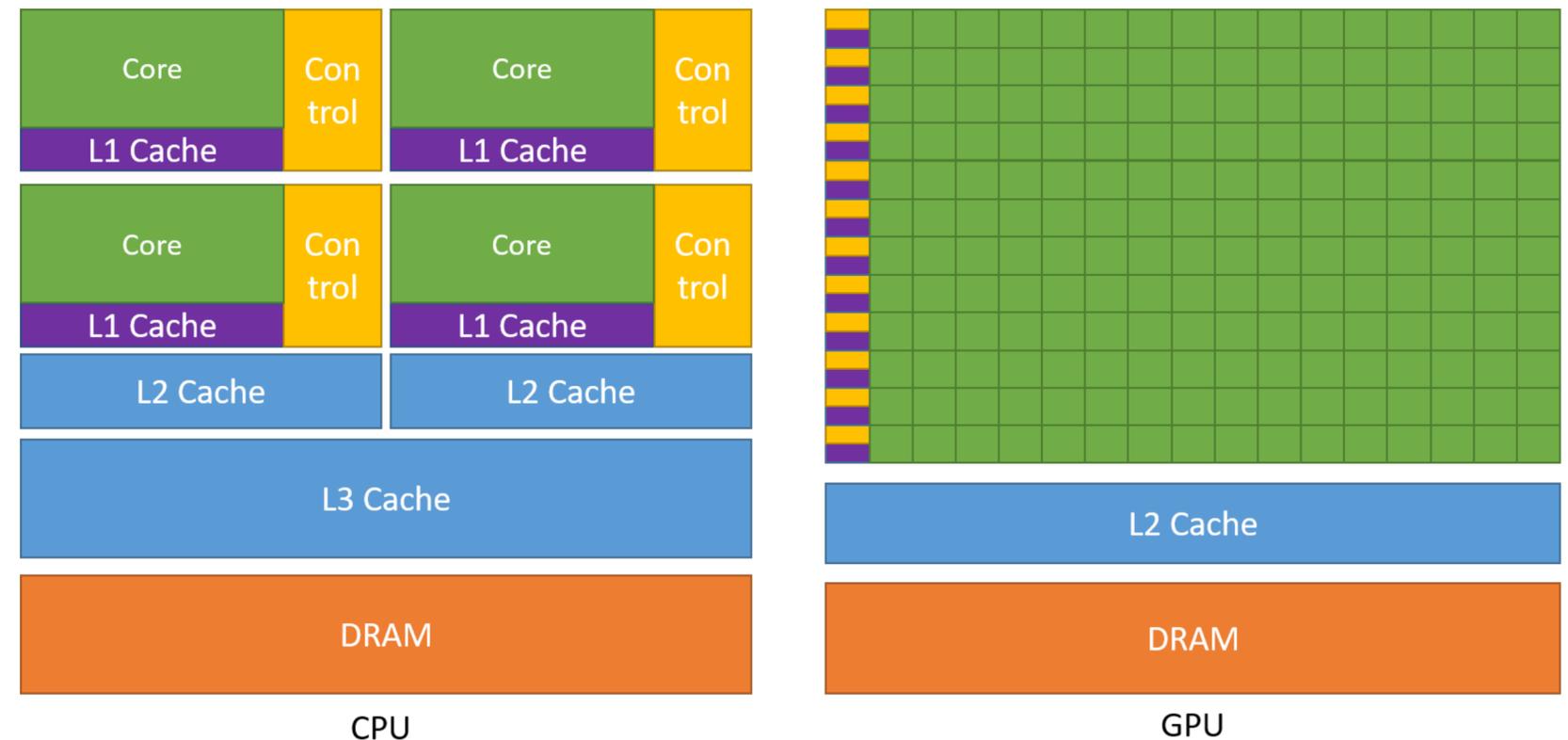
Accuracy:

$\frac{\text{correct classifications}}{\text{all classifications}}$

Monitoring

Basic GPU ideas: For what are GPUs efficient?

- GPUs are efficient at running the same operation on a large number of elements (i.e., running a lot of threads simultaneously).



Internal comparison between a CPU and a GPU.

<https://docs.nvidia.com/cuda/cuda-c-programming-guide/>

Deep Learning and Infrastructure

Checklist

Training Checklist

Tools & Infrastructure

- Place to train (access to cluster)
- Cluster essentials
- Data
- Code
- Monitoring & tracking

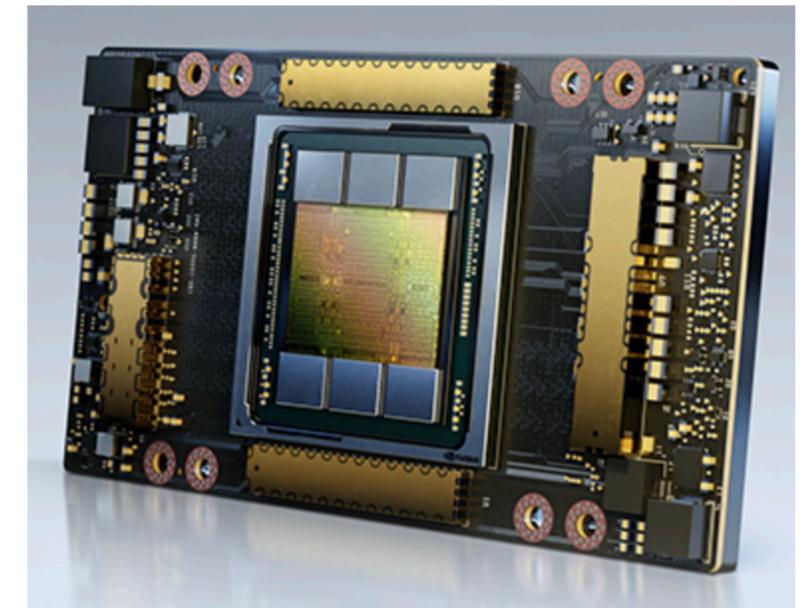
Where to train?

GPU System Grete

- Most energy efficient system in Germany (top 12 in the world)
- 144 NVIDIA A100 GPUs with 40GB memory
- Multi-Instance GPU: Splitting one GPU in more GPUs possible; we will be using *splits*
- Between nodes InfiniBand (2x200 GBit/s per node)
- GPU-to-Storage 130TiB local flash-based storage
- CUDA, the NVIDIA HPC SDK, and CUDA-enabled OpenMPI versions are available via the module system.



GPU System Grete



<https://www.nvidia.com/de-de/data-center/a100/>

GPU Cluster Grete: https://www.gwdg.de/documents/20182/27257/GN_3-2023_www.pdf

<https://news.ycombinator.com/item?id=35008694>

<https://info.gwdg.de/news/en/how-to-use-our-new-gpu-cluster-grete-for-hlrn-users/>

Mohammad Hossein Biniiaz | GWDG | 22. August 2024 | Credits: Dorothea Sommer

<https://nachrichten.idw-online.de/2022/11/23/germanys-most-energy-efficient-supercomputer>

GPU A100

Where to train?

Partition	Nodes	GPU + slices	VRAM each	CPU	RAM per node	Cores
grete	35	4 × Nvidia A100	40 GB	2 × Zen3 EPYC 7513	497 810 MB	64
	22	4 × Nvidia A100	80 GB	2 × Zen3 EPYC 7513	497 810 MB	64
grete:shared	54	4 × Nvidia A100	40 GB	2 × Zen3 EPYC 7513	497 810 MB	64
	2	4 × Nvidia A100	80 GB	2 × Zen3 EPYC 7513	1 013 000 MB	64
	2	8 × Nvidia A100	80 GB	2 × Zen2 EPYC 7662	1 013 620 MB	128
	3	4 × Nvidia V100	32 GB	2 × Skylake 6148	746 000 MB	40
grete:interactive	3	4 × Nvidia A100 (2g.10gb and 3g.20gb)	10/20 GB	2 × Zen3 EPYC 7513	497 810 MB	64
	3	4 × Nvidia V100	32 GB	2 × Skylake 6148	746 000 MB	40
grete:preemptible	3	4 × Nvidia A100 (2g.10gb and 3g.20gb)	10/20 GB	2 × Zen3 EPYC 7513	497 810 MB	64

https://docs.hpc.gwdg.de/how_to_use/compute_partitions/cpu_partitions/index.html

Mohammad Hossein Biniiaz | GWDG | 22. August 2024 | Credits: Dorothea Sommer

Where to train?

The GPUs, the options to request them, and some of their properties are given in the table below.

GPU	VRAM	FP64 cores	Tensor cores	-G option	-C option	Compute Cap.
Nvidia A100	40 GB	6912	432	A100		80
	80 GB	6912	432	A100	80gb	80
2g.10gb slice of Nvidia A100	10 GB	1728	108	2g.10gb 		80
3g.20gb slice of Nvidia A100	20 GB	2592	162	3g.20gb 		80
Nvidia V100	32 GB	5120	640	V100 (v100 SCC)		70
Nvidia Quadro RTX 5000	32 GB	3072	384	rtx5000 		75
Nvidia GeForce GTX 1080	8 GB	2560		gtx1080 		61
Nvidia GeForce GTX 1080	4 GB	2048		gtx980 		52

https://docs.hpc.gwdg.de/compute_partitions/gpu_partitions/index.html

Mohammad Hossein Biniiaz | GWDG | 22. August 2024 | Credits: Dorothea Sommer

Where to train?



Partitions

The partitions are listed in the table below without hardware details.

- Projects = More Compute

Cluster	Partition	OS	Shared	Max. walltime	Max. nodes per job	Core-hours per node
NHR	grete	Rocky 8		48:00:00	16	600
	grete:shared	Rocky 8	yes	48:00:00	1	150 per GPU
	grete:interactive	Rocky 8	yes	48:00:00	1	150/47 per GPU/slice
	grete:preemptible	Rocky 8	yes	48:00:00	1	150/47 per GPU/slice
	kisski	Rocky 8		48:00:00	16	600
	react	Rocky 8		48:00:00	16	600
	jupyter:gpu (jupyter)	Rocky 8	yes	24:00:00	1	150 per GPU
SCC	gpu	SL 7	yes	48:00:00	max	
	gpu-int (jupyter)	SL 7	yes	48:00:00	max	
	vis	SL 7	yes	48:00:00	max	

https://docs.hpc.gwdg.de/how_to_use/compute_partitions/gpu_partitions/index.html

Mohammad Hossein Biniiaz | GWDG | 22. August 2024 | Credits: Dorothea Sommer

Monitoring

Basic GPU ideas: What to monitor?

- 1 Start the program.
Define the network.



- 2 Copy model.



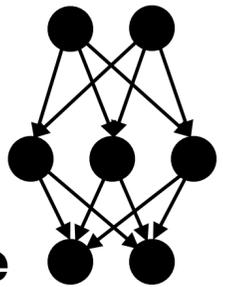
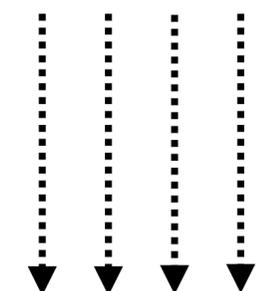
- 3 Copy data.



- 5 Copy result back.



- 4 Compute in threads.



“**Host**” = primary processor that manages the copying and controls the GPU

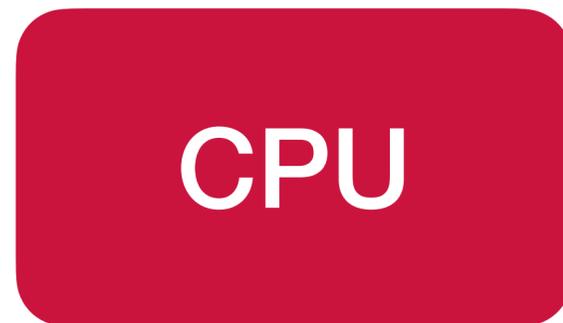
Computation is done in a **kernel function** that is executed in **parallel simultaneously** among many threads.

Same operation, just **different data** for the nodes.

Monitoring

Basic GPU ideas: What to monitor?

- 1 Start the program.
Define the network.



“**Host**” = primary processor that manages the copying and controls the GPU

- 2 Copy model.



- 3 Copy data.



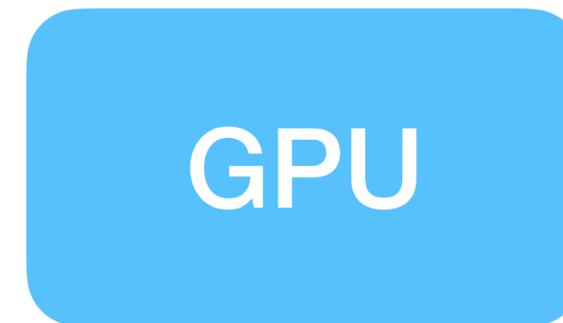
- 5 Copy result back.



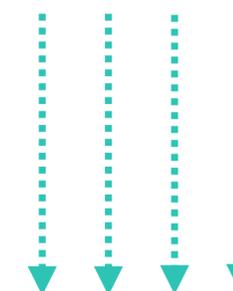
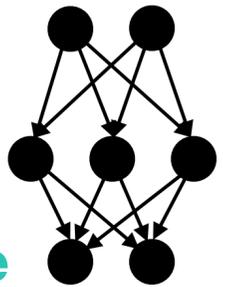
=> **Bandwidth matters!**

=> **Make sure to compute a lot, not only copy a lot!**

=> **Size (GB) matters!**



- 4 Compute in threads.

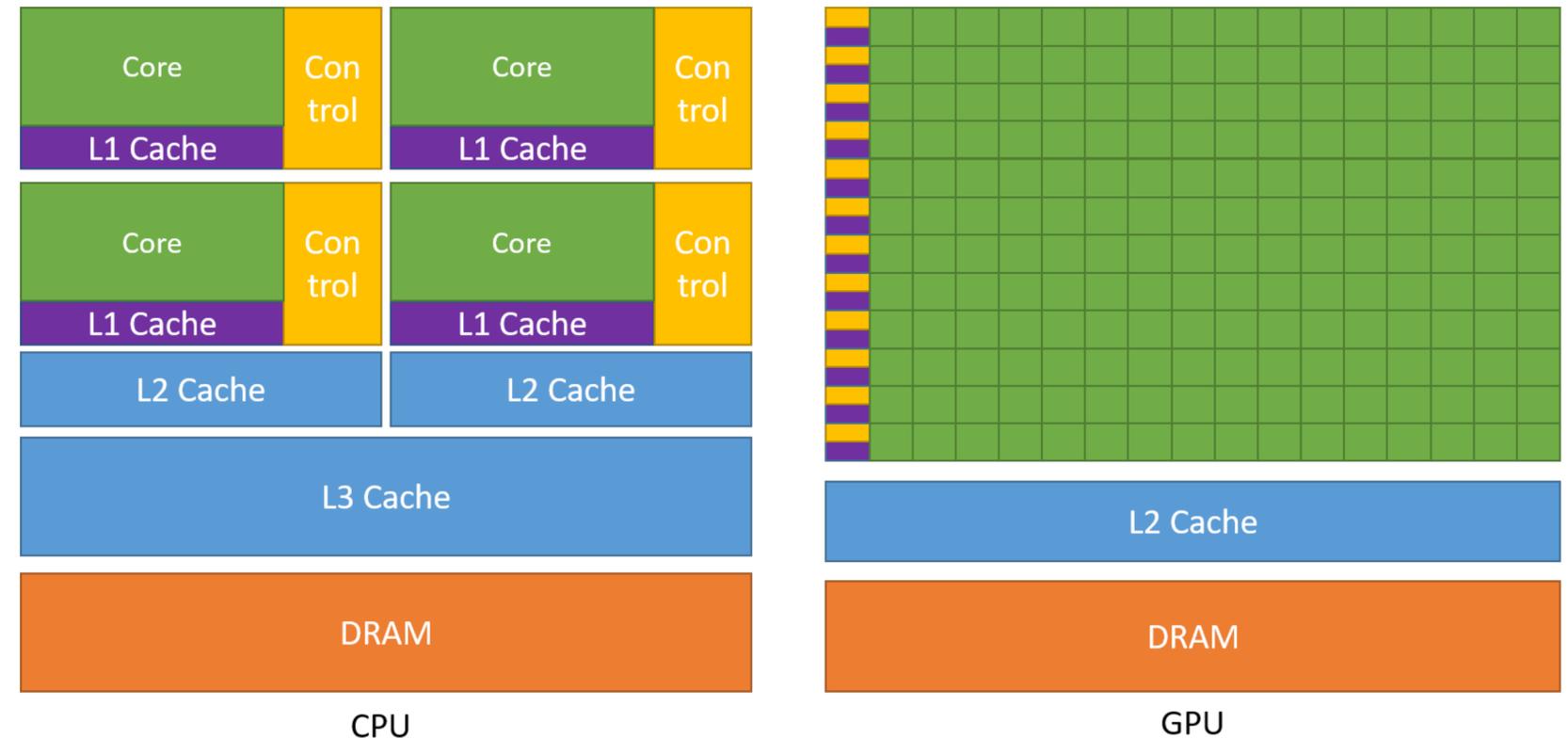


Computation is done in a **kernel function** that is executed in **parallel simultaneously** among many threads.

Monitoring

Basic GPU ideas: For what are GPUs efficient?

- Sequential operations are called a thread.
- GPUs are efficient at running the same operation on a large number of elements (i.e., running a lot of threads simultaneously).



Internal comparison between a CPU and a GPU.

<https://docs.nvidia.com/cuda/cuda-c-programming-guide/>

Cluster Access

Free User Account

- 1 Under <https://zulassung.hlrn.de/> fill out the application for a user account.

- 2 Print this form. Let your **university employer** (head of institute/supervisor) **sign this form** to confirm your position.

NHR@ZIB NHR@GÖTTINGEN

Login | Logout

deutsch | english

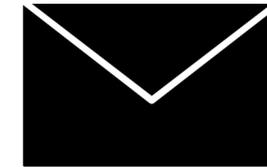
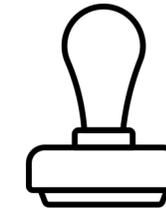
Joint Service Portal of NHR@ZIB and NHR@Göttingen (HLRN)

— Until the NHR-wide system (JARDS) is launched, please apply here for accounts and projects —

User Accounts

(General Information in a new window)

- Application for a user account**
Help in window on the right hand side
- Account information (retrieve and modify data)
(contact data, target hosts, query allocation and usage, password)
Help in window on the right hand side
- Manage keys for login to the HLRN
(parallel usage of multiple keys is possible)
Help in window on the right hand side



- 3 Send this signed form to gwdg@gwdg.de

Done! The GWDG assign you an adviser to help you with software questions and do the rest of the application process.

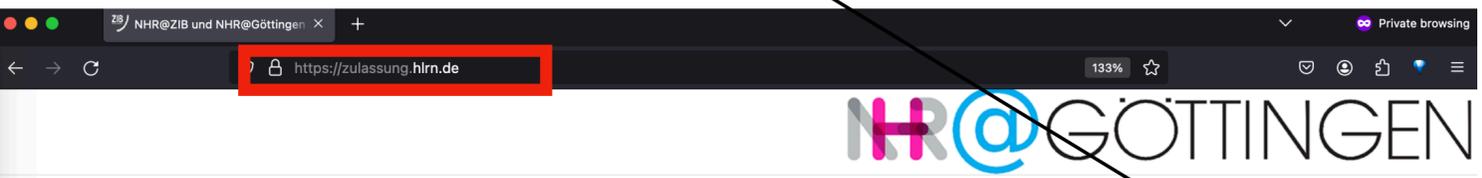
- 4 **You will receive an e-mail with your credentials.**

<https://zulassung.hlrn.de/>

Cluster Access

Free User Account

1 Under <https://zulassung.hlrn.de/> fill out the application for a user



Since 2021, the former HLRN provider sites in Berlin and Göttingen are part of the Germany-wide [NHR alliance \(www.nhr-verein.de\)](http://www.nhr-verein.de). Therefore, both centers offer information for users and about account/project application on their respective websites.
The former "service portal" (zulassung@hlrn.de) for a **shared management of both sites is no longer in production.**

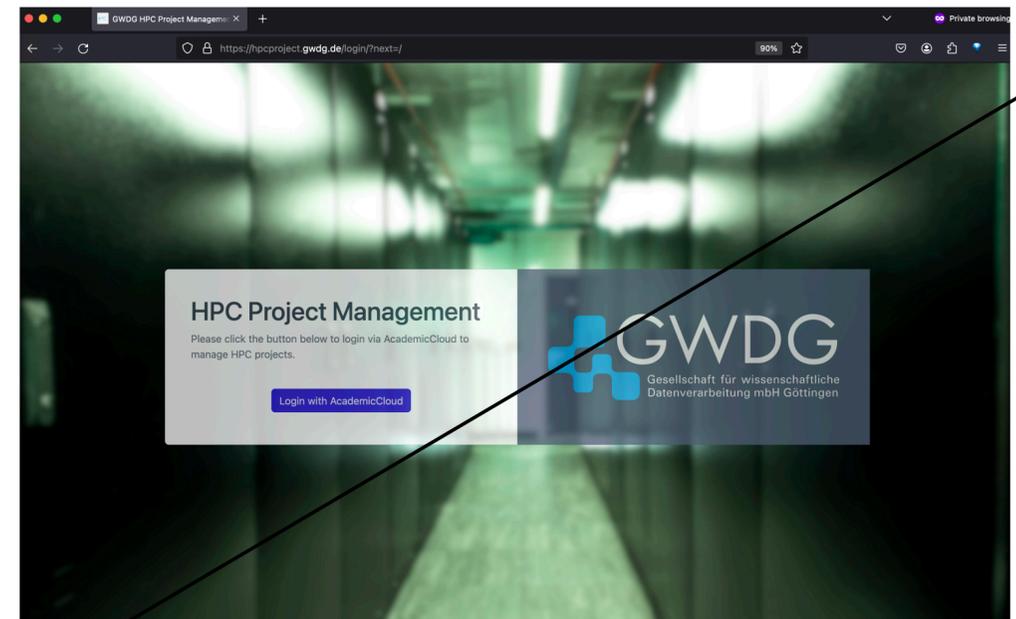
To access the portal of NHR@ZIB in Berlin please go to

<https://portal.nhr.zib.de/>

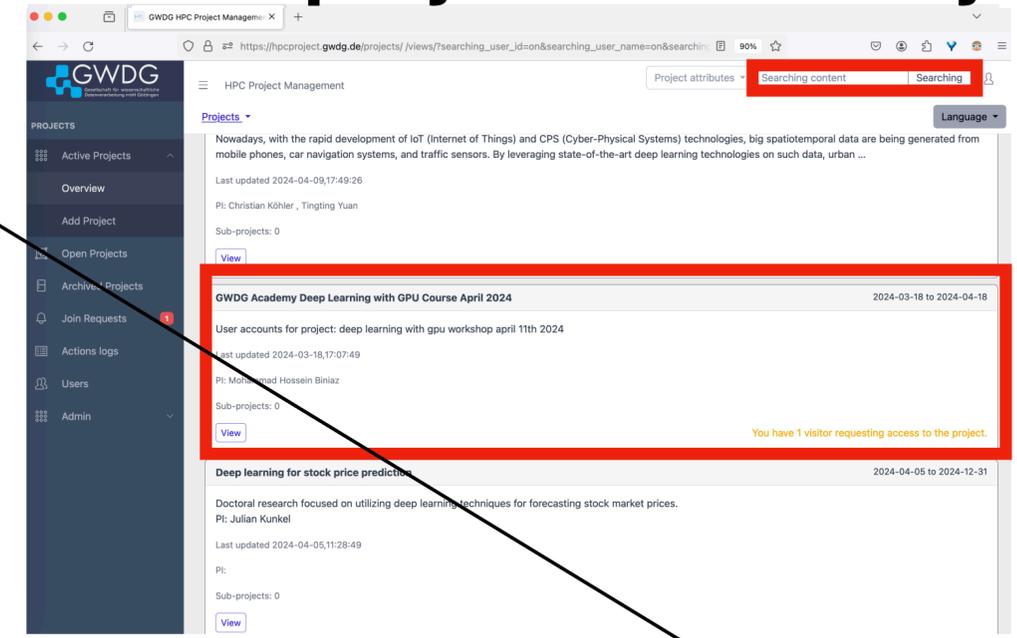
To access the portal of NHR-NORD@Göttingen please go to

<https://hpcproject.gwdg.de/>

2 Log in via AcademicCloud account



3 Search project and click join request



4 You will receive an e-mail with your credentials.

5 Alternatively: Write support support@gwdg.de

Cluster Access

Free User Account

1 Under <https://zulassung.hlrn.de/> fill out the application for a user account.



Since 2021, the former HLRN provider sites in Berlin and Göttingen are part of the Germany-wide [NHR alliance](http://www.nhr-verein.de). Therefore, both centers offer information for users and about account/project application on their respective websites.
The former "service portal" (zulassung@hln.de) for a shared management of both sites is no longer in production.



To access the portal of NHR@ZIB in Berlin please go to

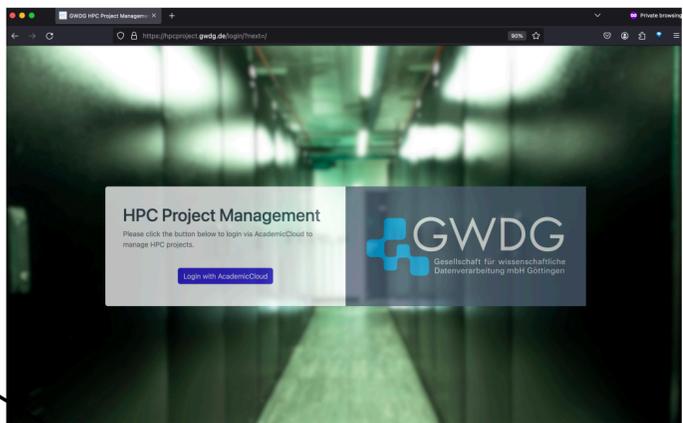
<https://portal.nhr.zib.de/>



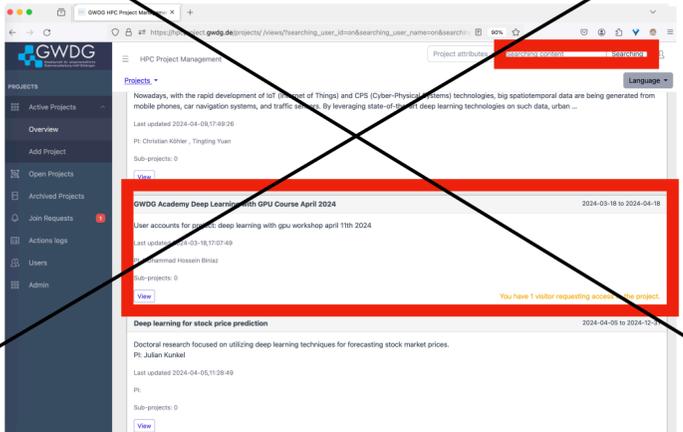
To access the portal of NHR-NORD@Göttingen please go to

<https://hpcproject.gwdg.de/>

2 Log in via AcademicCloud account



3 Search project and click join request

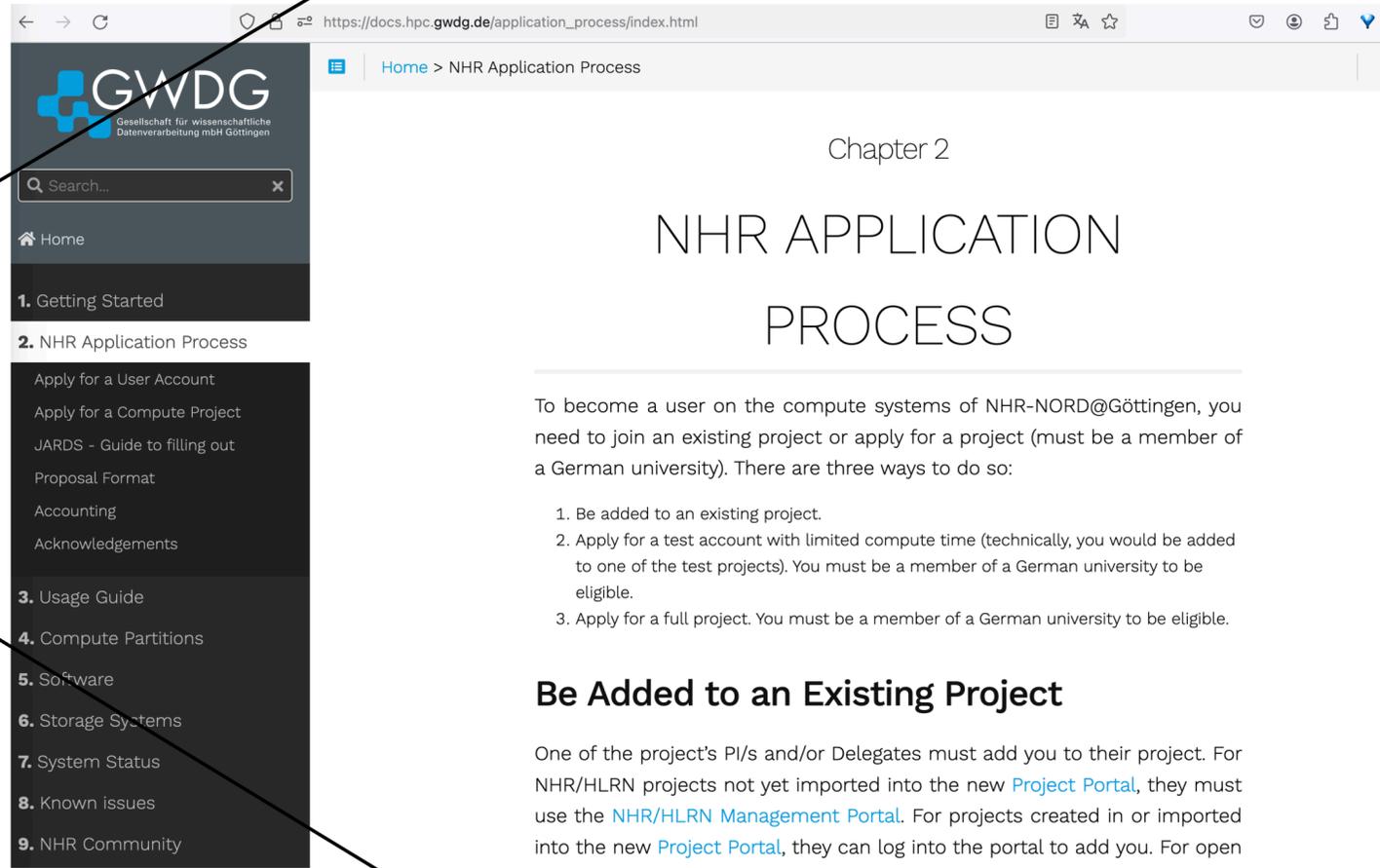


4 You will receive an e-mail with your credentials.

5 Write support

Read documentation

! https://docs.hpc.gwdg.de/application_process/index.html



Chapter 2

NHR APPLICATION PROCESS

To become a user on the compute systems of NHR-NORD@Göttingen, you need to join an existing project or apply for a project (must be a member of a German university). There are three ways to do so:

1. Be added to an existing project.
2. Apply for a test account with limited compute time (technically, you would be added to one of the test projects). You must be a member of a German university to be eligible.
3. Apply for a full project. You must be a member of a German university to be eligible.

Be Added to an Existing Project

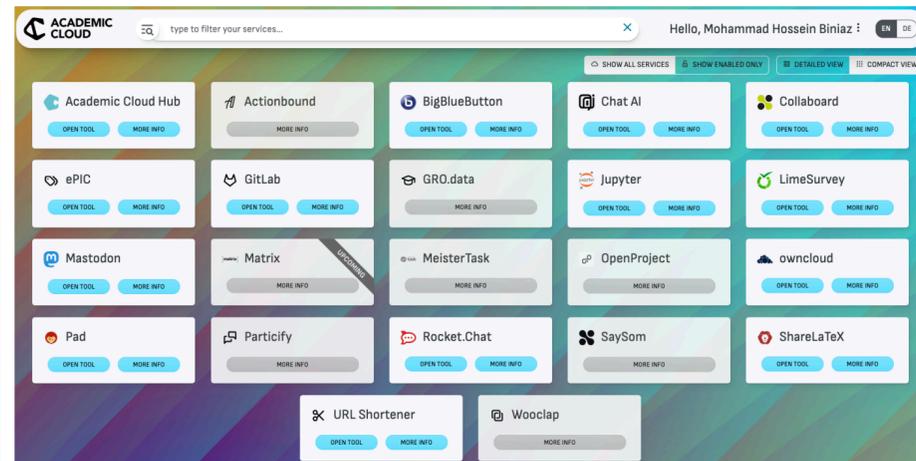
One of the project's PI/s and/or Delegates must add you to their project. For NHR/HLRN projects not yet imported into the new [Project Portal](#), they must use the [NHR/HLRN Management Portal](#). For projects created in or imported into the new [Project Portal](#), they can log into the portal to add you. For open

Cluster Access

Book training or Inference Service

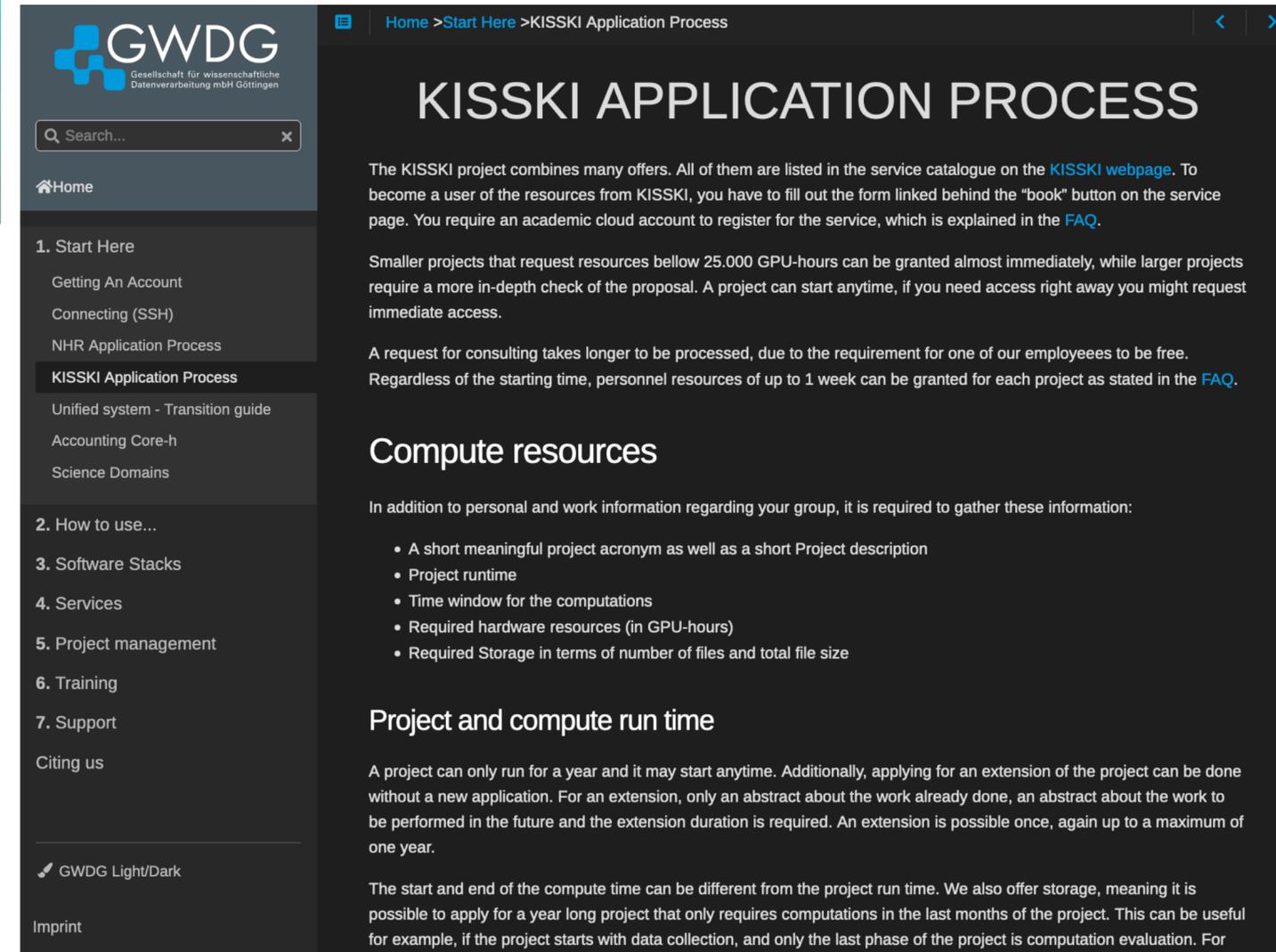
1 Submit the service booking

! You need: <https://academiccloud.de/services/>

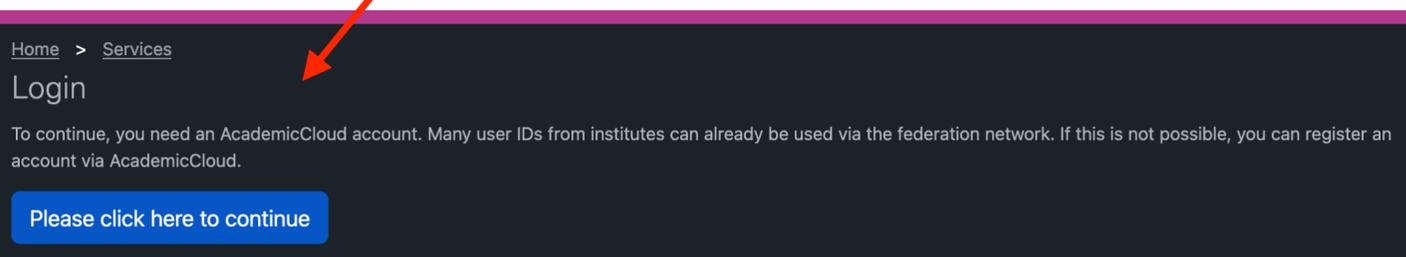
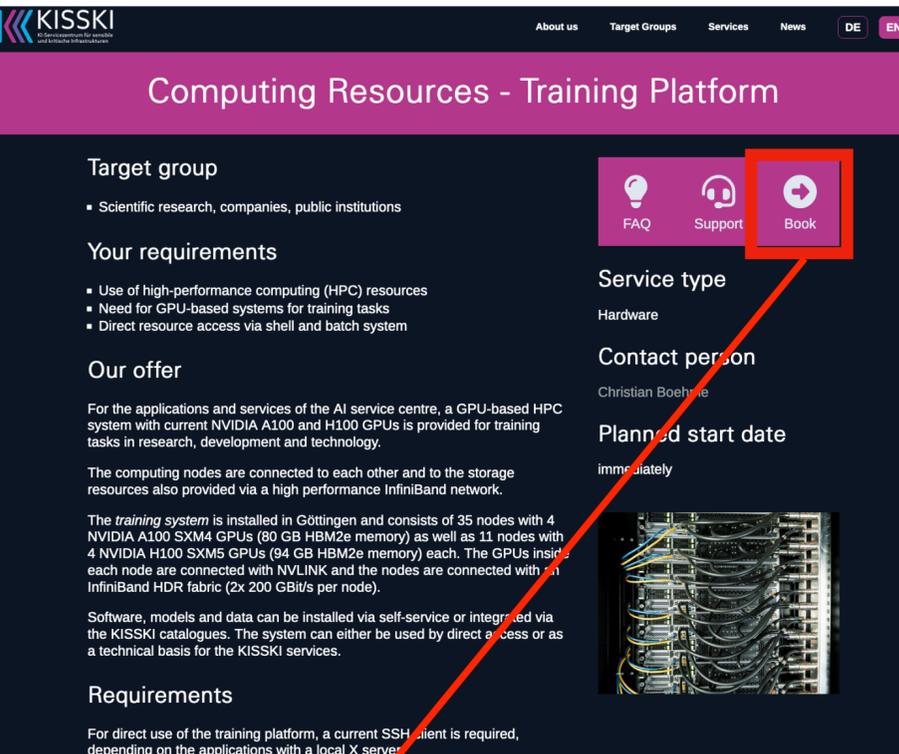


Read documentation

! https://docs.hpc.gwdg.de/start_here/kisski_application_process/



- 2 Wait for reply from GWDG: GWDG will create credentials ✓
- 3 Upload ssh-key 🗝️
- 4 Login via SSH & use the system



Booking resources

Links

- Kisski training platform booking https://kisski.gwdg.de/leistungen/2-01-01_trainingsplattform/
- Kisski inference platform booking https://kisski.gwdg.de/leistungen/2-01-02_inferenz/
- Kisski application process docs https://docs.hpc.gwdg.de/start_here/kisski_application_process/index.html
- NHR application process docs https://docs.hpc.gwdg.de/start_here/nhr_application_process/index.html

Extra help

Write support
support@gwdg.de

Training Checklist

Tools & Infrastructure

- **Place to train (access to cluster)**
 - Get free access for Emmy as a researcher in Germany.

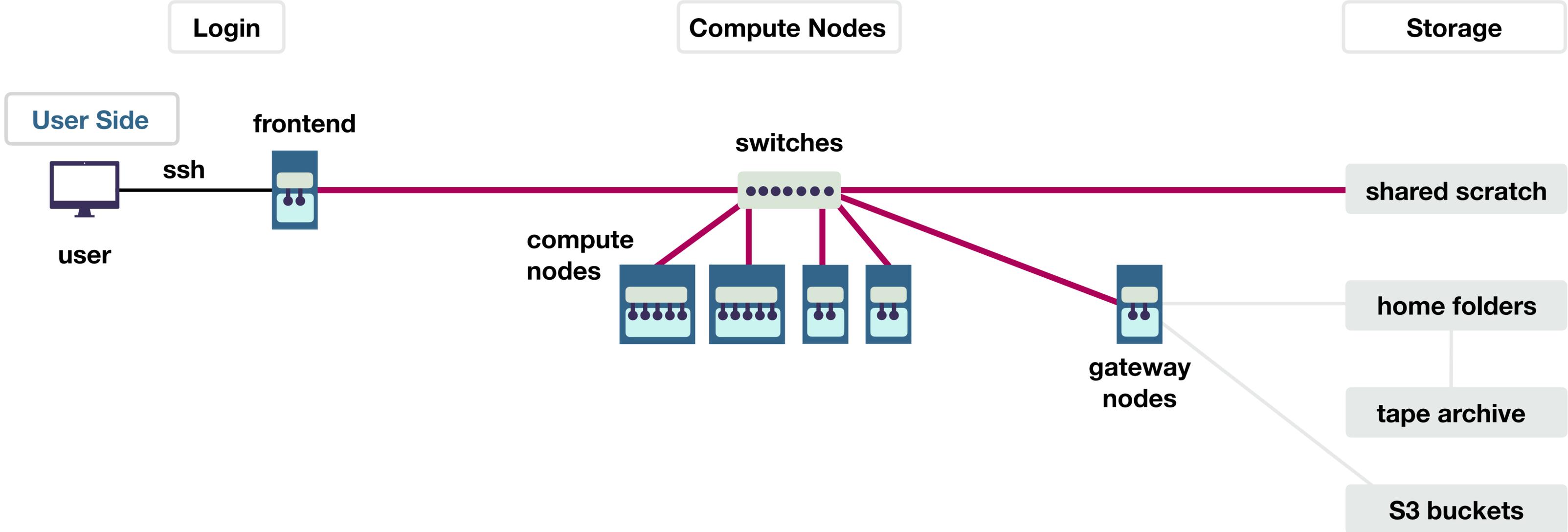
- **Cluster essentials**

- **Data**

- **Code**

- **Monitoring & tracking**

Simplified Network Overview



(fast) InfiniBand/Omni-Path
(slow) Ethernet

Simplified Network Overview

Login

Compute Nodes

Storage

User Side

frontend

switches

4 Put your data in the right place.



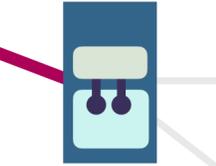
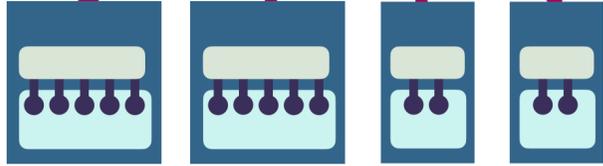
ssh



shared scratch

user

compute nodes



home folders

1 Log in on the frontend (glogin9).

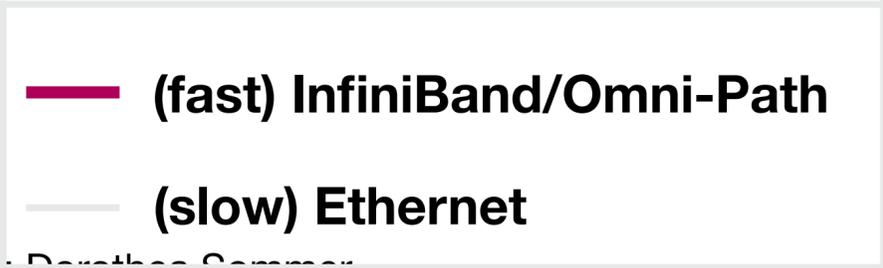
2 Do not directly run your jobs on the frontend!

3 Use our scheduler to run your programme.

gateway nodes

tape archive

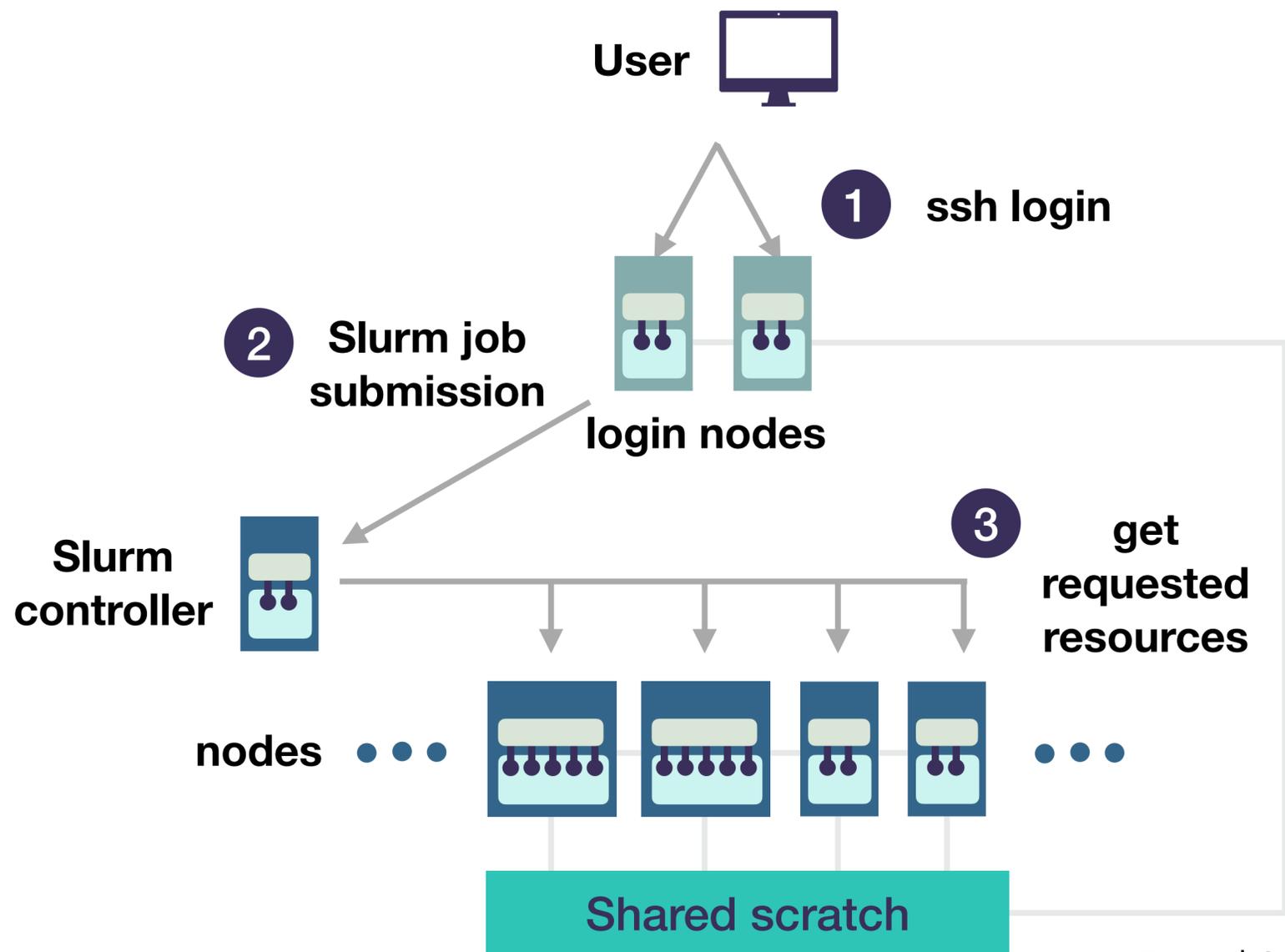
S3 buckets



Cluster Essentials

Slurm in 1 minute

Job Script Submission



resources you need

job.sh

```
#!/bin/bash
#SBATCH --job-name=custom-name

#SBATCH --nodes=2
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=2

#SBATCH --mem-per-cpu=4GB

#SBATCH --time=00:01:00
#SBATCH --mail-type=all
#SBATCH --mail-user=custom@mail.com
```

```
# Prepare the environment.
module load anaconda3/2021.05
source activate custom-env
```

Instead of anaconda on GWDG HPC, you can
Use miniconda without default channel
Or you can use miniforge3/CPython

```
# Run the script.
python custom-script.py
```

```
$ sbatch job.sh
```

submitting a job
running a Python programme on a CPU

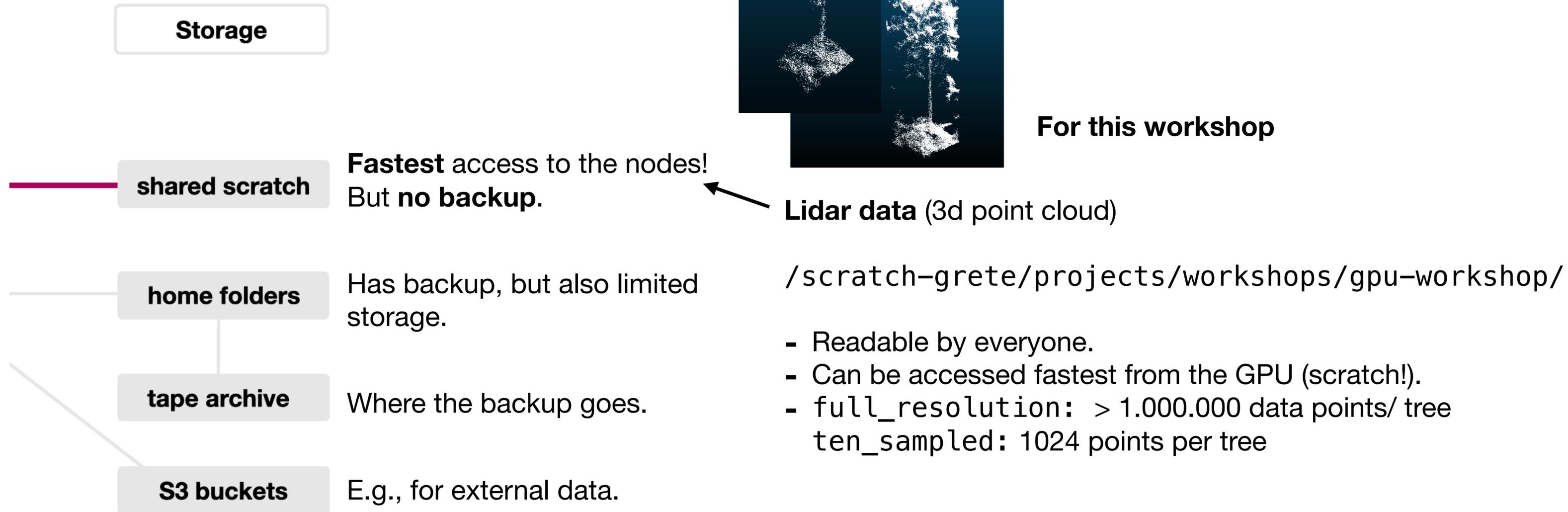
Training Checklist

Tools & Infrastructure

- **Place to train (access to cluster)**
 - Get free access for Emmy as a researcher in Germany.
- **Cluster essentials**
 - All jobs need to go through a scheduler (here: Slurm scheduler).
- **Data**
- **Code**
- **Monitoring & tracking**

Data

Where to store your data?



Training Checklist

Tools & Infrastructure

- **Place to train (access to cluster)**
 - Get free access for Emmy as a researcher in Germany.
- **Cluster essentials**
 - All jobs need to go through a scheduler (here: Slurm scheduler).
- **Data**
 - Use /scratch/ for the fastest access, but note that it has no backup.
- **Code**
- **Monitoring & tracking**

Frameworks

The frameworks are similar: define a model in Python code, optimised computations in the background



Josh Tobin
@josh_tobin_



Why do people always ask what ML framework to use?
It's easy:

- jax is for researchers
- pytorch is for engineers
- tensorflow is for boomers

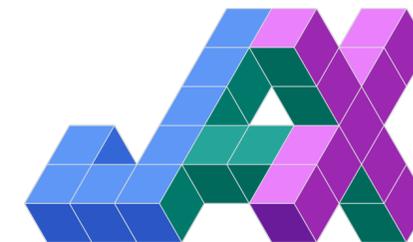
6:24 PM · Mar 11, 2021 · Twitter Web App



- From Google (2015)



- From Facebook, now part of Linux foundation
- Dominant: competitions (77% winners), models, number of papers



- Newest: from Google v.0.3.13 (2022)
- Auto-differentiation, vectorisation
- Deep learning: needs separate framework (Flax, Haiku)

Code



- 1 Log in to the frontend **glogin9**.
- 2 Clone the code you need (e.g. in your home directory).

```
utils.py  train.py 6 x  model.py  $ submit_train.sh
train.py > create_data_loader
112     return model
113
114     if __name__ == "__main__":
115         print("Start training")
116
117         ### LEARNING PARAMETERS ###
118
119         n_classes = len(TREE_SPECIES)
120         device = torch.device("cuda" if torch.cuda.is_available() else "cpu") # GPU available?
121         print(f"Training with: {device}")
122
123         saved_models_path = "./saved_models"
124         if not os.path.exists(saved_models_path):
125             os.makedirs(saved_models_path)
126             print(f"Created path for models in {saved_models_path}")
127
128         learning_rate = 0.001
129         batch_size = 32
130         num_training_epochs = 100
131
132
```

<https://gitlab-ce.gwdg.de/hpc-team-public/deep-learning-with-gpu-cores>

Environments

Why is environments is important?

- **Reproducibility!**
 - Projects require different Python and package versions.
 - On the cluster, everyone needs a different environment.
- 1 Create a new conda environment.
 - 2 Install all packages, either manually or from the `requirements.txt`
 - 3 Activate environment before running a job.



```
requirements.txt 2.01 KIB
1 anyio==3.6.2
2 argon2-cffi==21.3.0
3 argon2-cffi-bindings==21.2.0
4 asttokens==2.2.0
5 attrs==22.1.0
6 backcall==0.2.0
7 beautifulsoup4==4.11.1
8 bleach==5.0.1
9 certifi==2022.9.24
10 cffi==1.15.1
11 charset-normalizer==2.1.1
12 comm==0.1.1
13 contourpy==1.0.6
14 cycler==0.11.0
15 debugpy==1.6.4
16 decorator==5.1.1
```

Environ

Why is environme

- **Reproducibility**
- Projects require
- On the cluster, e

- 1
- 2
- 3

CONDA & PYTHON

For python and environment we offer the modules

- `miniforge3`
- `CPython`

The environment `base` is installed system wide and cannot be modified.

Warning

The old module `anaconda3` is no longer supported. Please switch to another module.

Warning

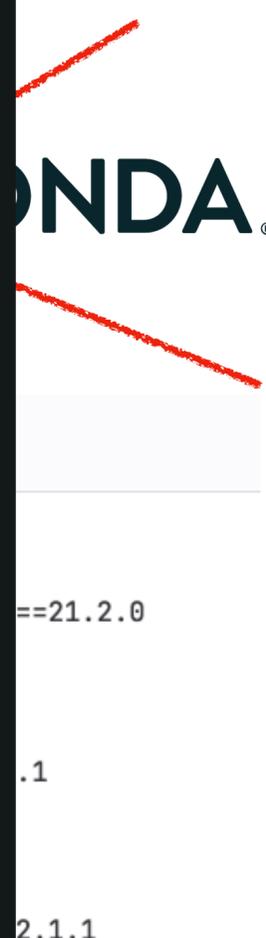
We discourage users from using the module `miniconda3` because the "default" channel requires a payed licence for more than 200 "employees" and outside of class room usage. You use this module at your own risk.

Loading the module and preparing an environment

Info

We recommend using `source activate` instead of `conda init`. This way, the `.bashrc` is not updated and the environment is not loaded at each login, reducing the load on the file system.

In order to use python and conda environments load the module `miniforge3` and create an environment. We strongly suggest you create this environment on the scratch file using the prefix option. This example load the module, creates a new environment called `myenv` on the scratch file system linked to the variable `$WORK`, loads `python3.12`, and acitvates it using the



Environments

Anaconda/miniconda not recommended :(



Conda the package manager CAN be used

- The default channel however cannot be!
- Change default channel or use miniforge/CPython

One does not need to purchase a license if they use `conda` package manager only with conda-forge channel. I received this answer from Anaconda customer support via email on February 27th, 2023.

This can be achieved by creating a file `~/.condarc` with the following contents:

```
channels:  
- conda-forge
```

You can check the active channels by running:

```
conda config --show channels
```

Share Follow

edited Mar 14, 2023 at 22:25

answered Feb 27, 2023 at 14:48

 tuomastik
4,809 ● 5 ● 41 ● 50

<https://stackoverflow.com/questions/74762863/are-conda-miniconda-and-anaconda-free-to-use-and-open-source/75581962#75581962>

tl;dr:

- The `conda` package manager is free to use; not all package channels are free to use for commercial purposes, such as the `anaconda` channel.
- Packages built by the Anaconda organization and hosted on the "anaconda" `conda` channel are not free to use commercially, but can be used for free for non-commercial purposes as per their [Terms of Service](#).
- Packages hosted on channels such as "`conda-forge`" and "`bioconda`" are free to use.
- some third-party conda channels might also be free or non-free to use, so be sure to check before using them in a manner that may violate the channel's Terms of Service.

The answer directly from the Anaconda User_Care_Team:

<https://community.anaconda.cloud/t/is-conda-cli-free-for-use/14303>

Hello.

The terms of service only applies to the packages in the Anaconda.com 21 (<http://anaconda.com/> 11) package repository, not to the conda software itself (which has an open source BSD license).

You are free to use conda with any other source of packages (conda-forge, their own packages, etc) without worrying about the Anaconda commercial terms of service.

Please let me know if this answers your question.

Kim

Training Checklist

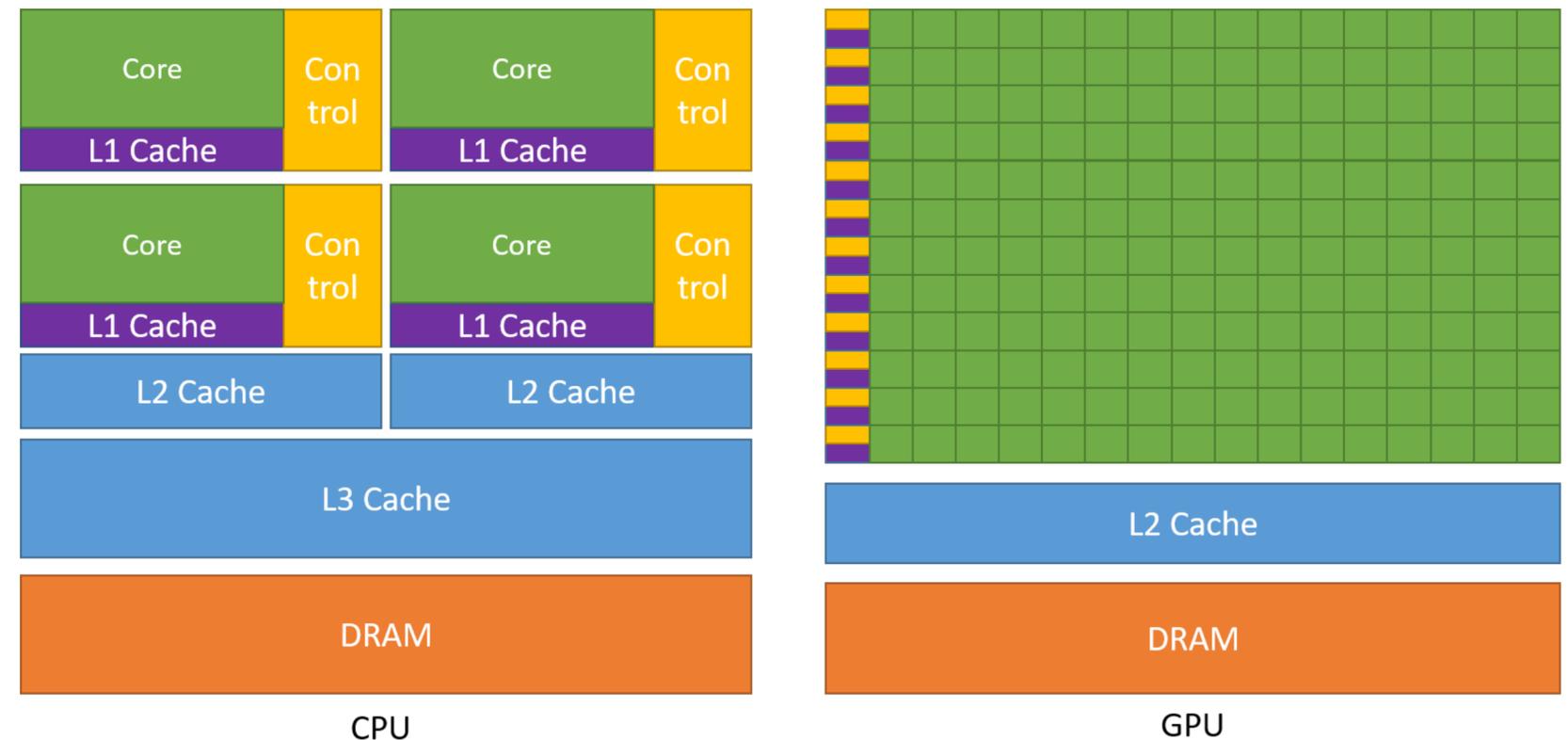
Tools & Infrastructure

- **Place to train (access to cluster)**
 - Get free access for Emmy as a researcher in Germany.
- **Cluster essentials**
 - All jobs need to go through a scheduler (here: Slurm scheduler).
- **Data**
 - Use /scratch/ for the fastest access, but note that it has no backup.
- **Code**
 - Choose framework (PyTorch).
 - Make or clone GitHub repository.
 - Create a conda environment with `requirements.txt`.
- **Monitoring & tracking**

Monitoring

Basic GPU ideas: For what are GPUs efficient?

- Sequential operations are called a thread.
- GPUs are efficient at running the same operation on a large number of elements (i.e., running a lot of threads simultaneously).



Internal comparison between a CPU and a GPU.

<https://docs.nvidia.com/cuda/cuda-c-programming-guide/>

Monitoring

Basic GPU ideas: What to monitor?

- 1 Start the program.
Define the network.



- 2 Copy model.



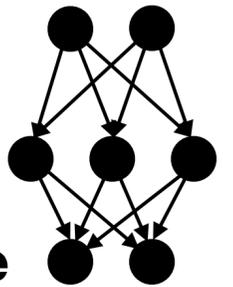
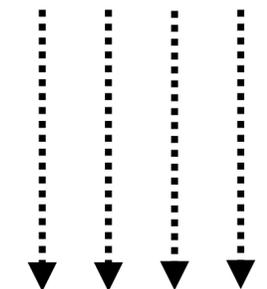
- 3 Copy data.



- 5 Copy result back.



- 4 Compute in threads.



“**Host**” = primary processor that manages the copying and controls the GPU

Computation is done in a **kernel function** that is executed in **parallel simultaneously** among many threads.

Same operation, just different data for the nodes.

Monitoring

Basic GPU ideas: What to monitor?

- 1 Start the program.
Define the network.



“**Host**” = primary processor that manages the copying and controls the GPU

- 2 Copy model.



- 3 Copy data.



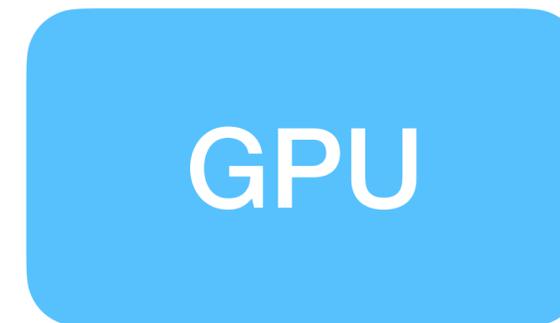
- 5 Copy result back.



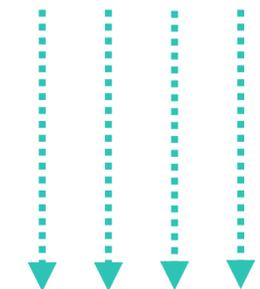
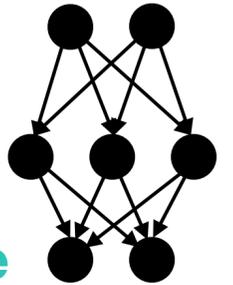
=> **Bandwidth matters!**

=> **Make sure to compute a lot, not only copy a lot!**

=> **Size (GB) matters!**



- 4 Compute in threads.



Computation is done in a **kernel function** that is executed in **parallel simultaneously** among many threads.

Training Checklist

Tools & Infrastructure

- **Place to train (access to cluster)**
 - Get free access for Emmy as a researcher in Germany.
- **Cluster essentials**
 - All jobs need to go through a scheduler (here: Slurm scheduler).
- **Data**
 - Use `/scratch/` for the fastest access, but note that it has no backup.
- **Code**
 - Choose framework (PyTorch).
 - Make or clone GitHub repository.
 - Create a conda environment with `requirements.txt`.
- **Monitoring & tracking**
 - Ensure that your jobs utilise the GPU well (computation and memory).

What we'll do

	Deep Learning with GPU cores	
09.30 - 09.45	Welcome	
09.45 - 10.15 (30 min)	Deep Learning and Infrastructure	Learn how to train a neural network with a GPU.
10.15 - 11.30 (60 min)	Practical: Working on the GPU	
11.30 - 11.45	Short break ☕	
11.45 - 12.00 (15 min)	Introduction to Profiling	Learn how to profile the training and training efficiently.
12.00 - 12.45 (45 min)	Practical: Profiling Jobs	
12.45 - 13.00	General Q&A	

Practical Part I

Let's switch to the code...

Additional Material

Machine Learning Paradigms

Supervised

learning with teacher

Unsupervised

learning representations

Reinforcement

learning behaviour

Data

Observations $\underline{x}_1, \dots, \underline{x}_n$

Labels y_1, \dots, y_n

Observations $\underline{x}_1, \dots, \underline{x}_n$

States $\underline{s}_1, \dots, \underline{s}_n$

Actions a_1, \dots, a_n

Rewards r_1, \dots, r_n

Aim



predict label of observation

regression, classification



extract relevant structures
for useful representation

**dimensionality reduction,
clustering**

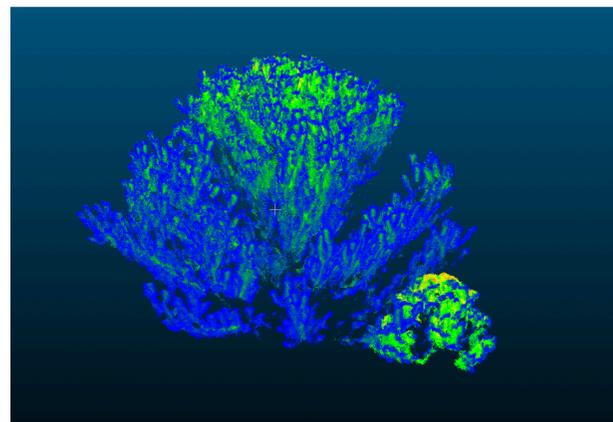


find best action in
every state

Unboxing the Model

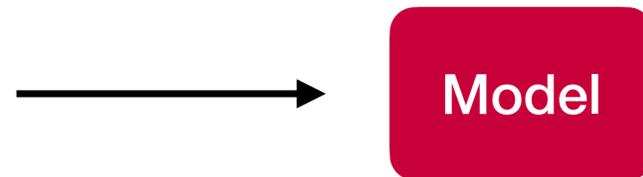
What do we know about our input data?

In **general**: the type of neural network depends on the **input data type**



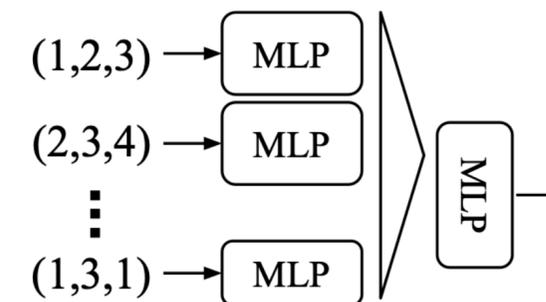
3D point clouds

$$f(x_1, \dots, x_n) \approx g(h(x_1), \dots, h(x_n))$$



$$h: \mathbb{R}^N \rightarrow \mathbb{R}^K \text{ neural network}$$

$$g: \underbrace{\mathbb{R}^K \times \dots \times \mathbb{R}^K}_n \rightarrow \mathbb{R} \text{ symmetric function (e.g., max pooling)}$$



symmetry function

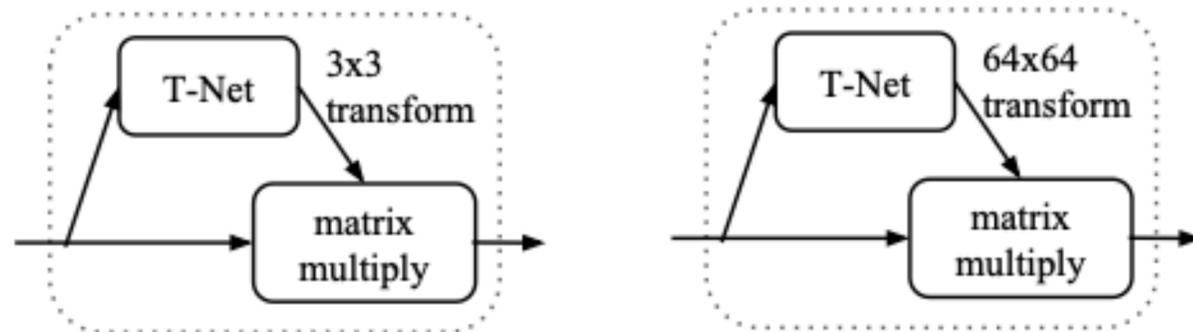
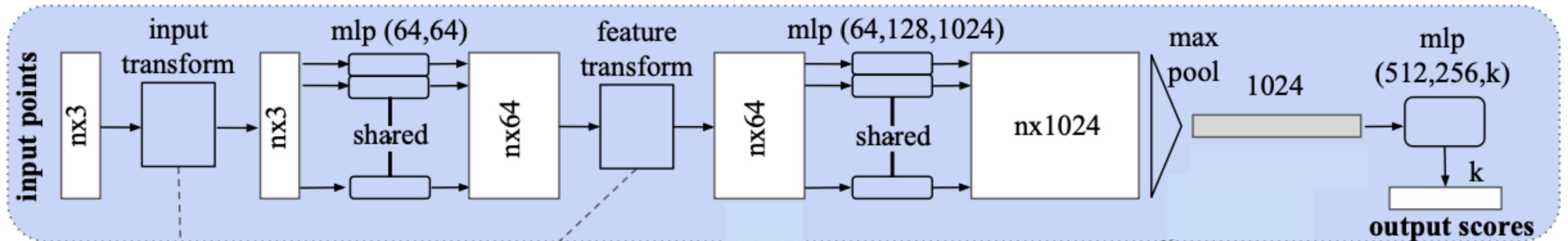
1. **unordered** set of points, a list of (x, y, z)
2. **invariance** under rigid transformations

Qi et al. (2017) **PointNet**: Deep Learning on Point Sets for 3D Classification and Segmentation

Unboxing the Model

PointNet Architecture

Classification Network



T-Net as a learned affine transformation matrix

Look into the paper for more details.

Qi et al. (2017) **PointNet**: Deep Learning on Point Sets for 3D Classification and Segmentation

Stack for Deep Learning

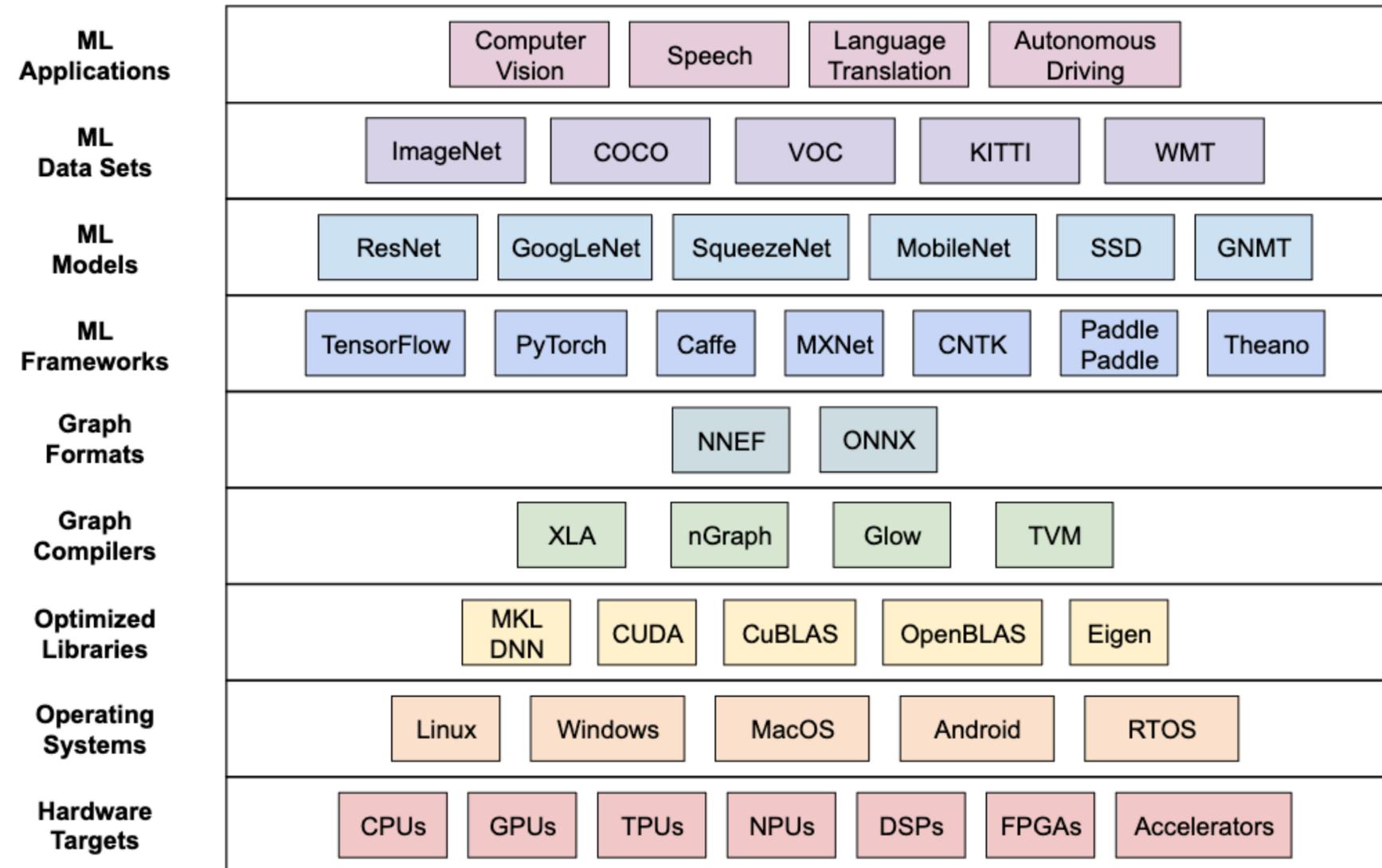
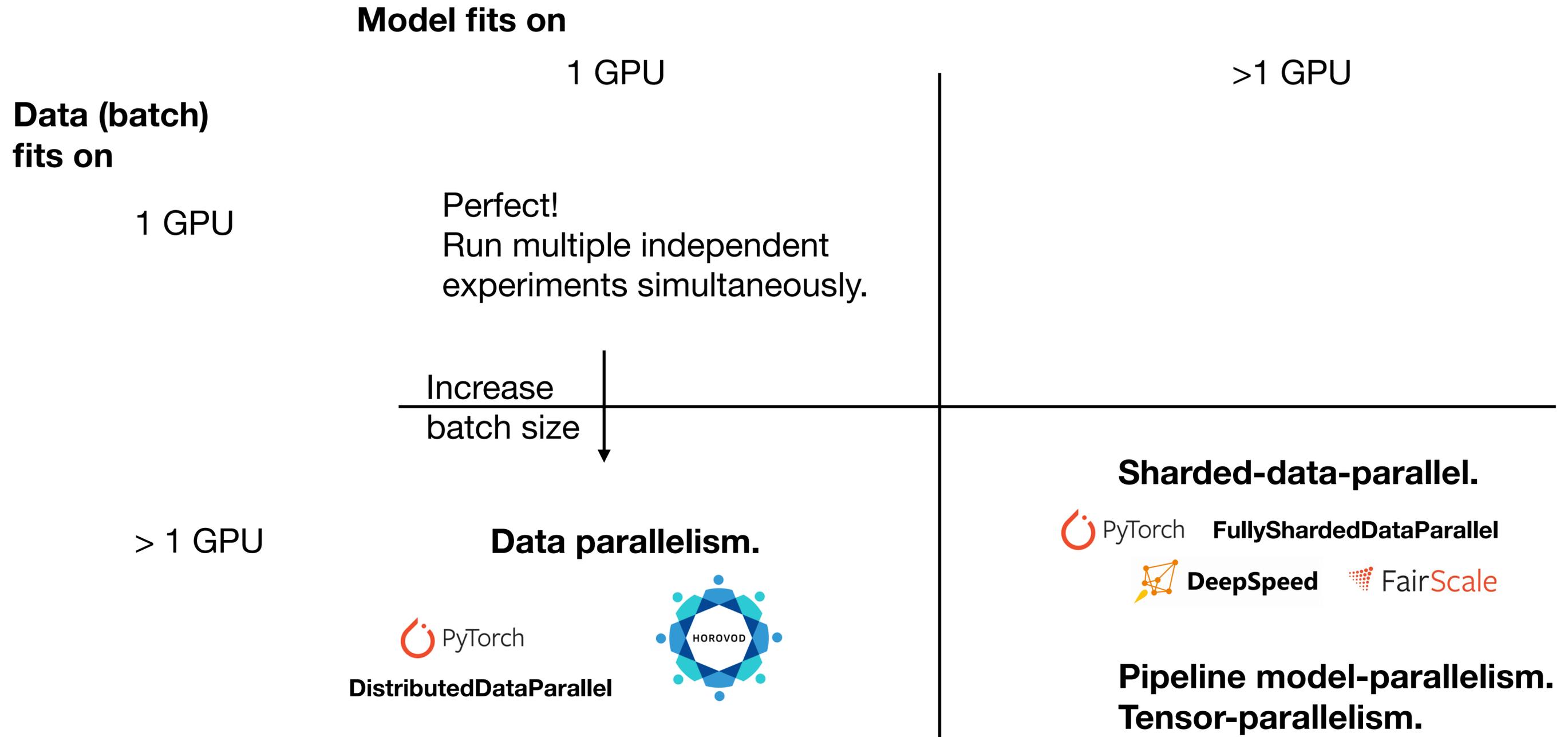
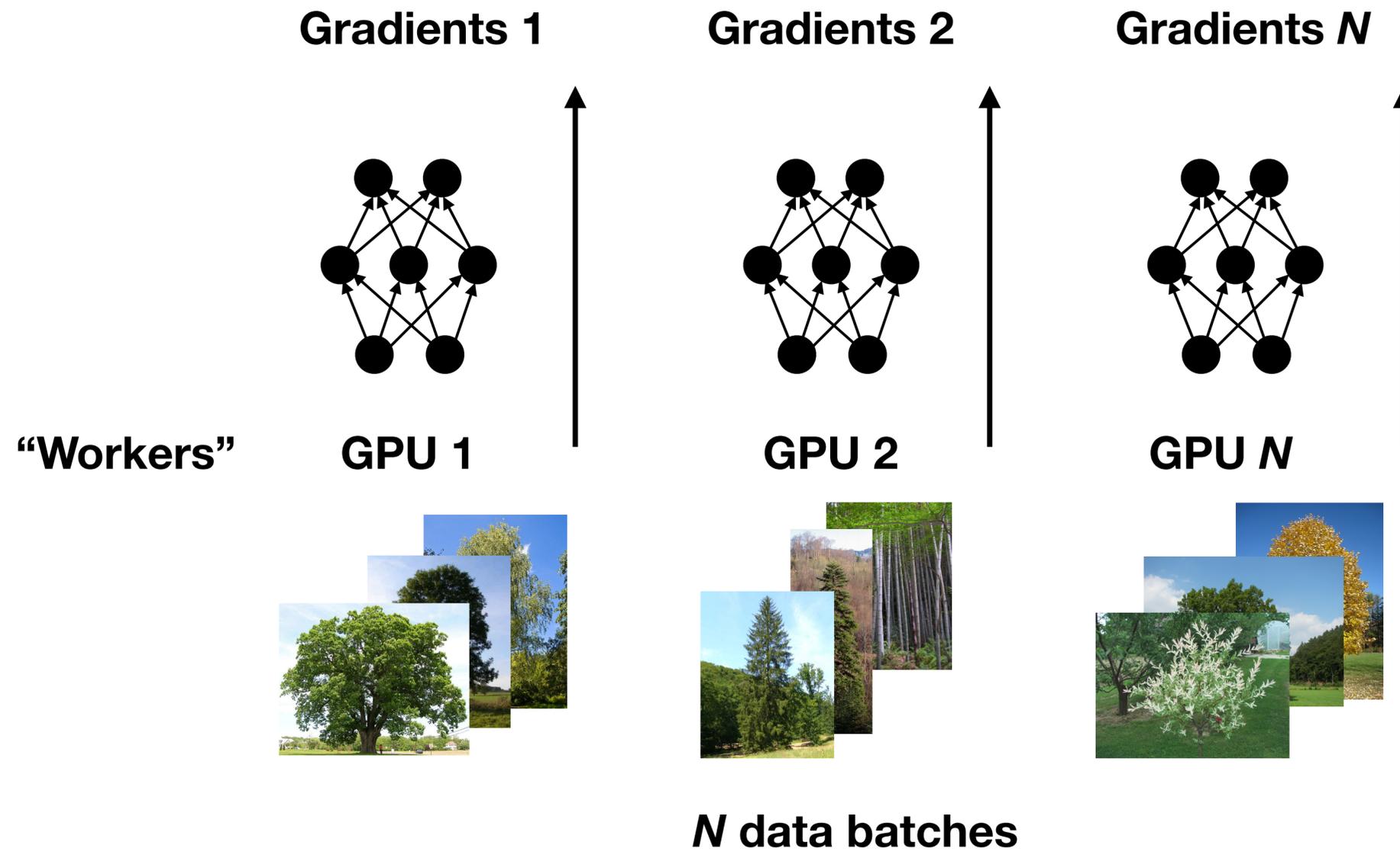


Fig. 2. The diversity of options at every level of the stack, along with the combinations across the layers, make benchmarking inference systems hard.

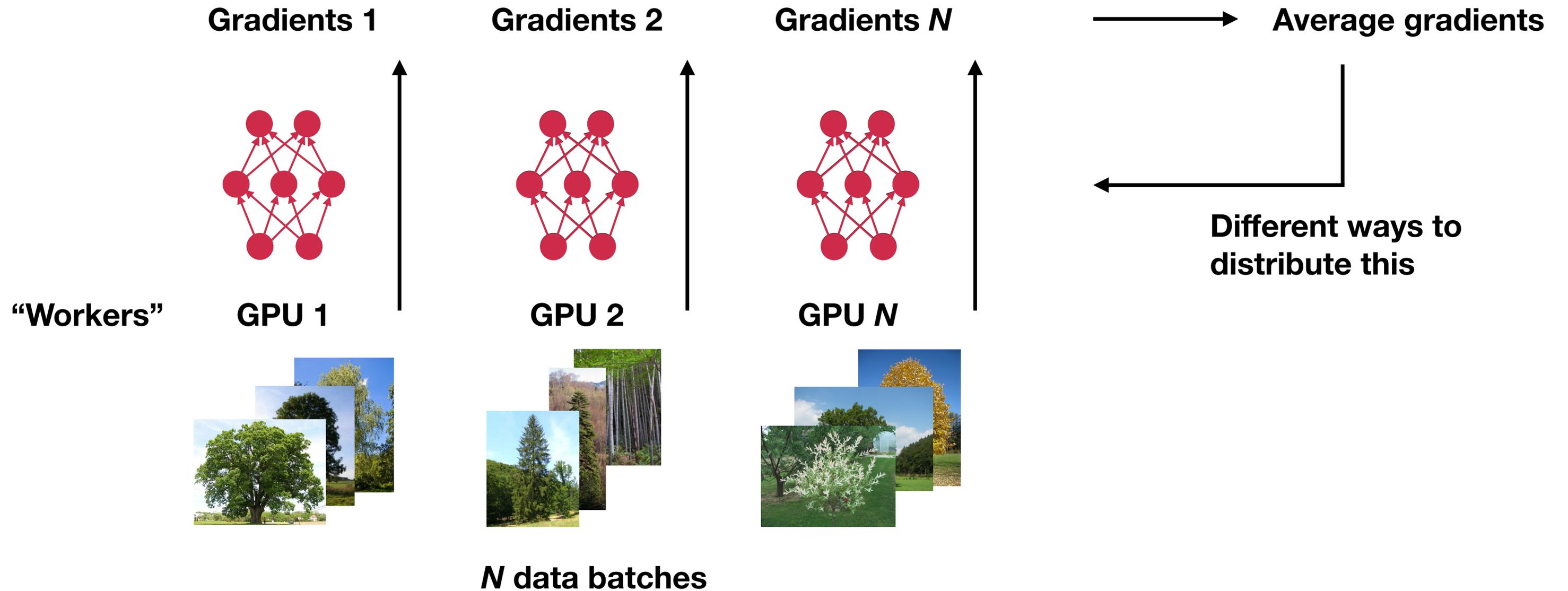
Memory issues: What to do, if ...



Data Parallelism

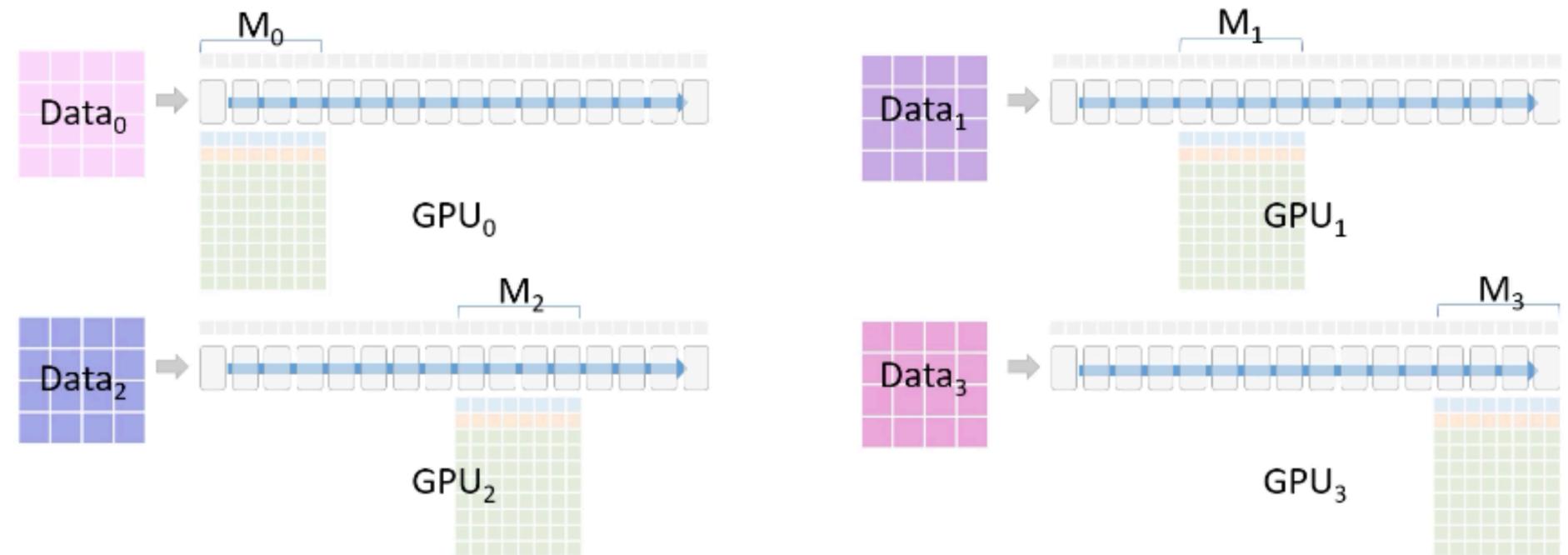


Data Parallelism



Sharded data-parallel

- Idea: optimizer states take most of model GPU memory
- copy model parameters around, only 1 GPU keeps optimiser states for 1 part of the model
- data is also sharded



Each GPU is responsible for 1 piece of the end model
ZeRO P_{os+g+p} and Gradient accumulation are used with the 4-way data parallelism

<https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/>

Pipeline model-parallelism

- Idea: (processed) data is moved around between the GPUs
- Needs fine-tuning, otherwise only 1 GPU active at a time (and others are idle)

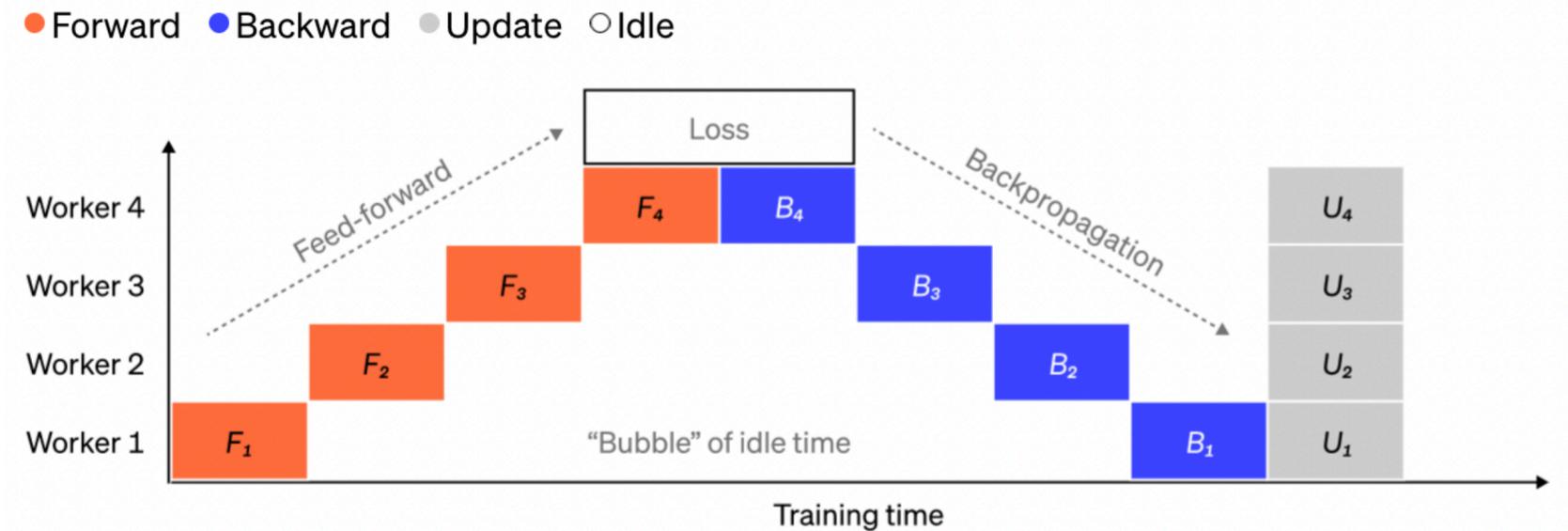
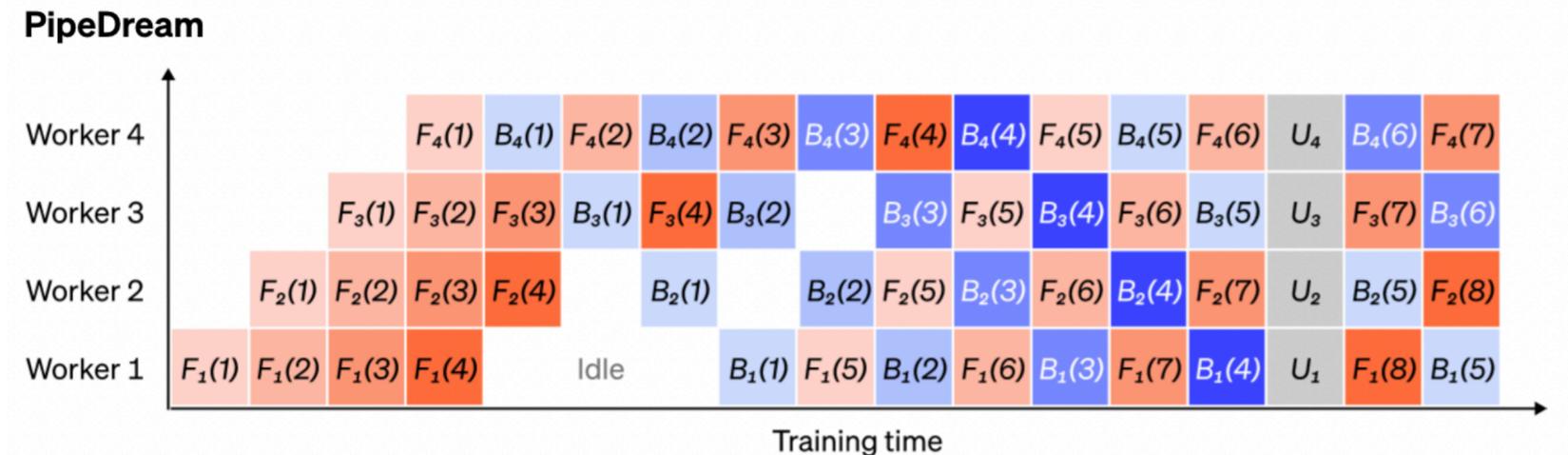


Illustration of a naive pipeline parallelism setup where the model is vertically split into 4 partitions by layer. Worker 1 hosts model parameters of the first layer of the network (closest to the input), while worker 4 hosts layer 4 (which is closest to the output). "F", "B", and "U" represent forward, backward and update operations, respectively. The subscripts indicate on which worker an operation runs. Data is processed by one worker at a time due to the sequential dependency, leading to large "bubbles" of idle time.



Tensor-parallelism

- Idea: think of matrix multiplication as dot-product between pairs of rows and columns, so it can be splitted among GPUs
- Example: Megatron from Nvidia for Transformers

Q&A

- **Course certificates:** If you need a printed certificate for course participation, please write an e-mail to dorothea.sommer@gwdg.de until 5th of April (including the mail where it should be sent to).
- Your course accounts can be used until **11.4.2023!**
Today: Use the reservation with `--cpu-per-gpu=4` and `-p grete:shared`
All the time: You can submit normally to Grete with `-p grete` and adapt `-G 1`
- Who can get **access to Emmy**, specifically also for projects?
<https://pad.gwdg.de/s/cAA-M2vpl#>
- **Other questions:** Is there anything you would
 - like to discuss regarding deep learning/GPU?
 - see covered in another course?