Data-efficient Deep Learning for Earth Observation

Joëlle Hanna, Linus Scheibenreif
University of St. Gallen

Michael Mommert
Stuttgart University of Applied Sciences

What this tutorial is about

As part of this tutorial, we will introduce and discuss different techniques to make more efficient use of data in Deep Learning for Earth observation.

In detail, we will focus on two aspects:

- **Data efficiency**: “How can we use available data most efficiently?”
- **Label efficiency**: “How can we use available labels (or even unlabeled data) most efficiently?”

We will address these questions in lecture-style presentations of the fundamentals, hands-on coding labs and discussions.

Who we are

Joëlle Hanna

PhD student
"Multi-modal Representation Learning for Remote Sensing"

Linus Scheibenreif

PhD student
"Self-supervised Deep Learning for Earth Observation"

University of St.Gallen

Michael Mommert

Prof of AI in Remote Sensing
Stuttgart University of Applied Sciences

Hochschule für Technik Stuttgart

Today's syllabus

Time	Content
9:00 – 9:15	**Introductions** (Michael)
9:15 – 10:00	**Deep Learning Recap and Data Fusion** (Michael)
10:00 – 10:15	**Multitask Learning** (Joëlle)
10:15 – 10:45	*Coffee Break*
10:45 – 11:15	**Multitask Learning** (cont'd) (Joëlle)
11:15 – 12:00	**Self-supervised Learning** (Linus)

Resources for this tutorial

- All coding will be done in Jupyter Notebooks. You can access these Notebooks through github: https://github.com/mommermi/IGARSS2024_DataEfficientDeepLearningEO
- We will run the Jupyter Notebooks in the cloud. If possible, we prefer to use Google Colab for this purpose. If you do not have a Google account, please let us know.
- The dataset that we will be using is the ben-ge dataset (see https://github.com/HSG-AIML/ben-ge for more information). In this tutorial, we will use a tiny version of ben-ge, which will be made accessible for the time of the tutorial. If you are following this tutorial at some other time, feel free to use the ben-ge-8k dataset (see ben-ge website).

https://github.com/mommermi/IGARSS2024_DataEfficientDeepLearningEO

Data-efficient Deep Learning for Earth Observation

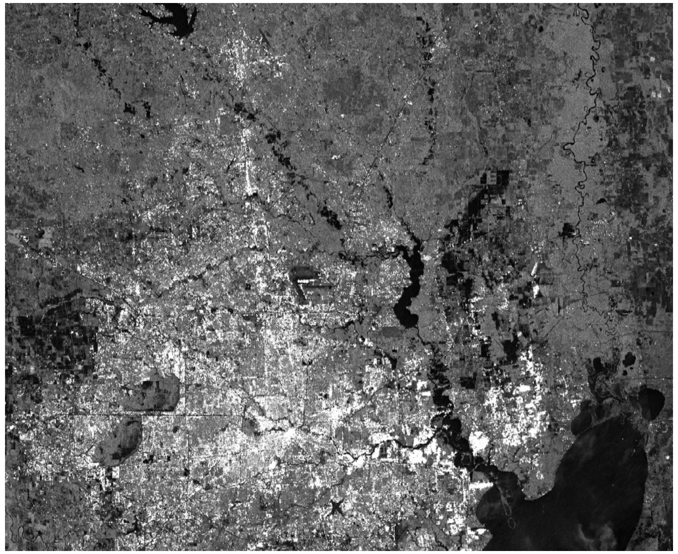
Deep Learning | Data Fusion | Multi-task Learning | Self-Supervised Learning

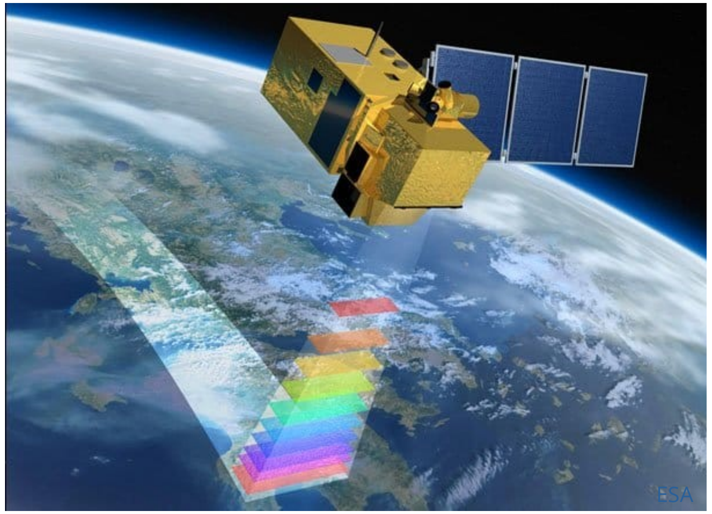
Deep Learning for Earth observation

Earth observation data are highly complex (unstructured, multi-modal).

How can we analyze these vast amounts of data?

Deep Learning offers the **scalability** to analyze large amounts of data.





Deep Learning for Earth observation

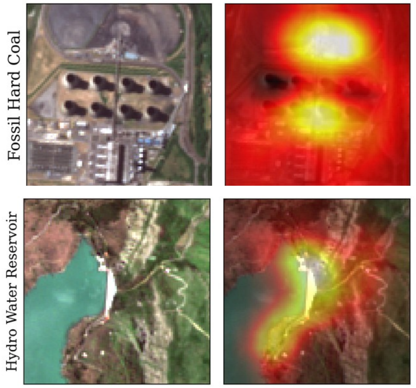
Earth observation data are highly complex (unstructured, multi-modal).

How can we analyze these vast amounts of data?

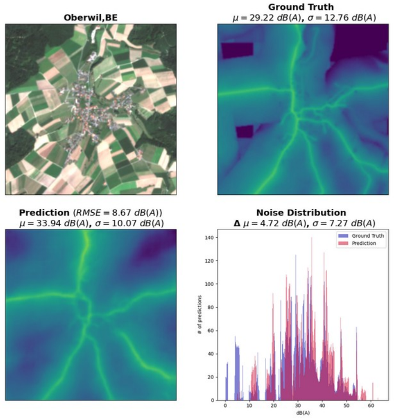
Deep Learning offers the **scalability** to analyze large amounts of data.

Deep Learning also offers the **flexibility** to deal with a range of different tasks.

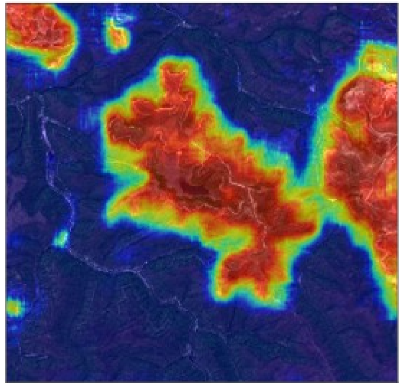
How does it work?



Classification



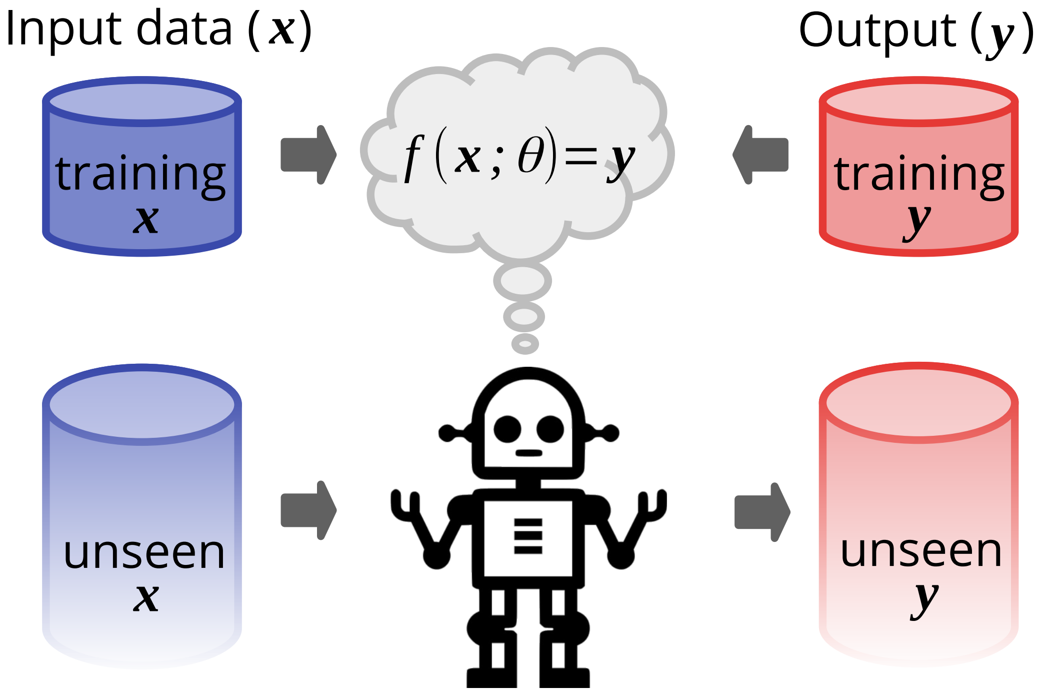
Regression



Segmentation

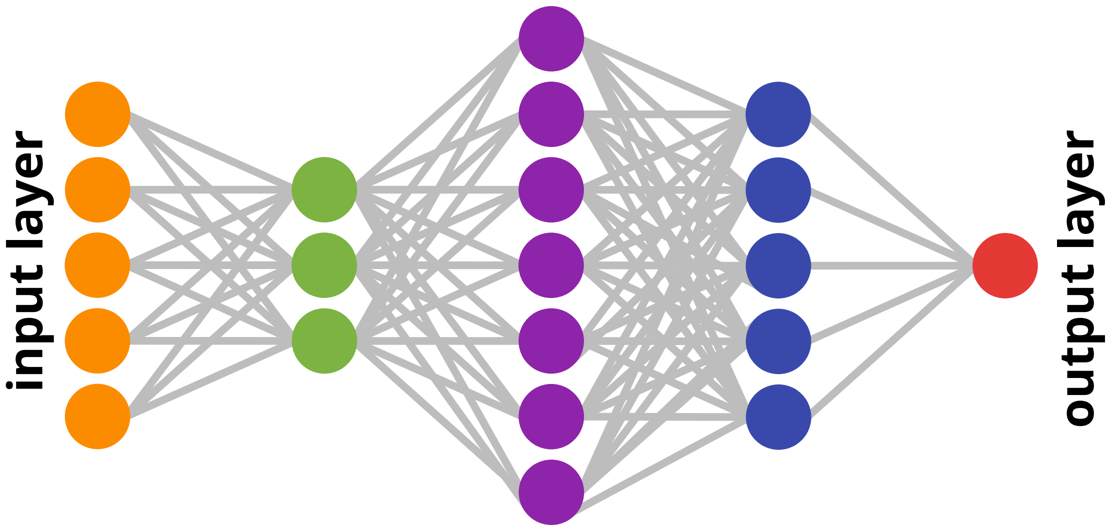
Object Detection

Supervised learning with Neural Networks



A machine learns a task from **annotated examples**.

Mathematically, it learns a function, f, that maps input data, x, to the output, y.



A Neural Network is a cascade of mathematical functions; each neuron contains learnable weights that represent the learned knowledge.

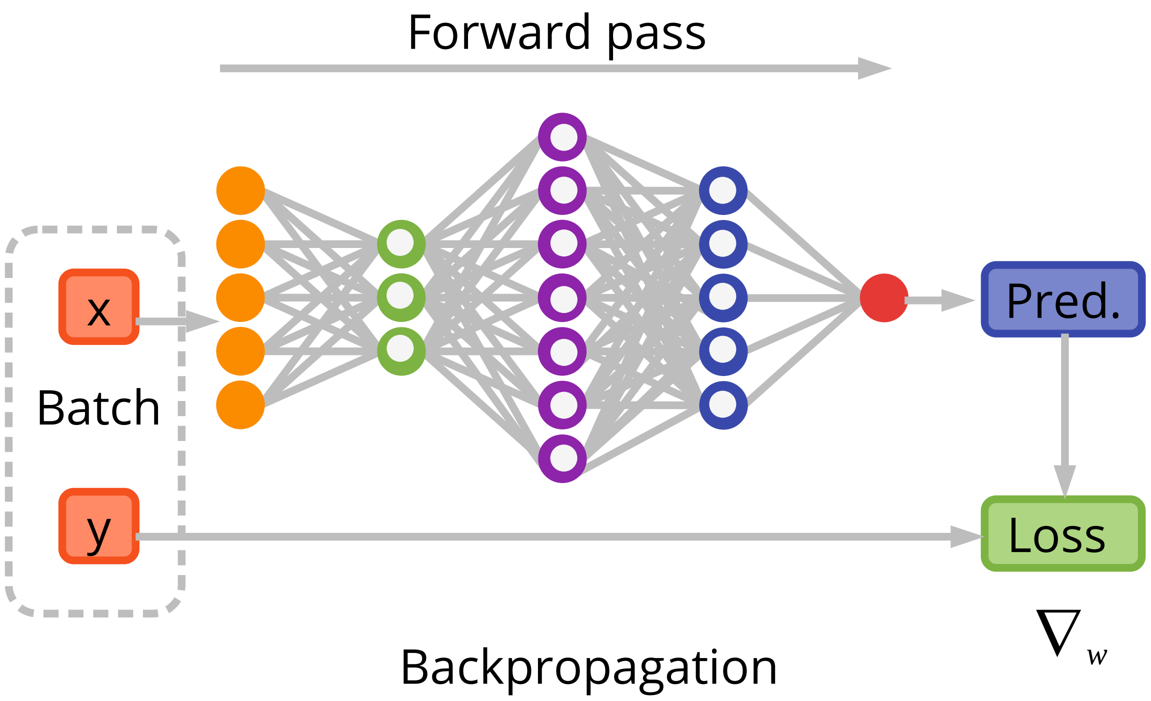
How does the model learn?

Neural network training pipeline

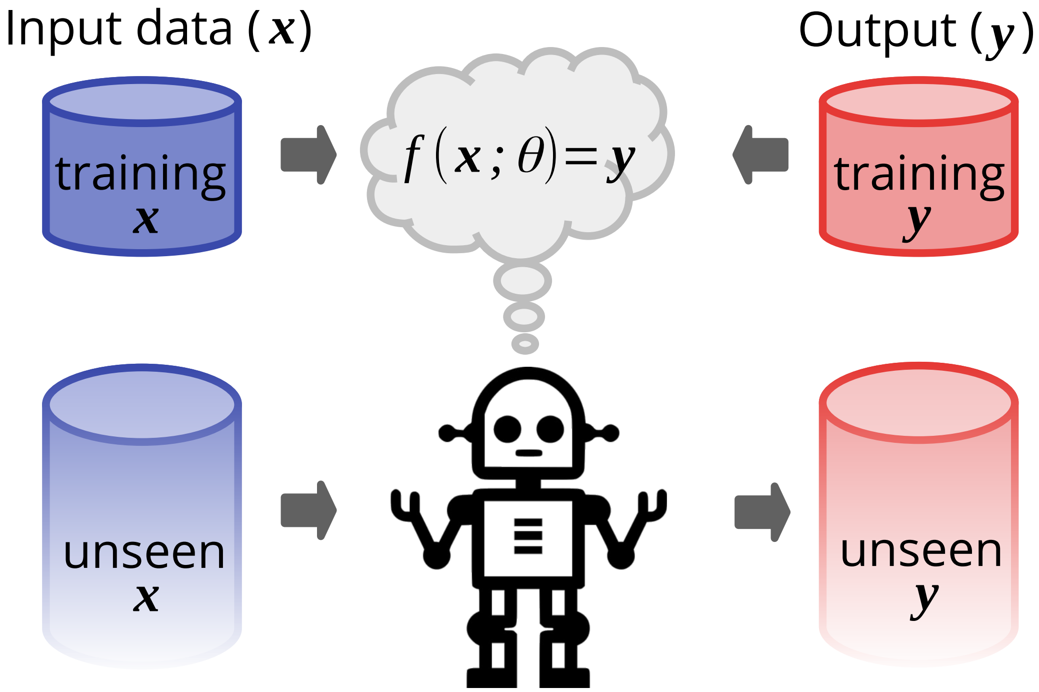
- Sample batch (input data x and target data y) from training dataset:

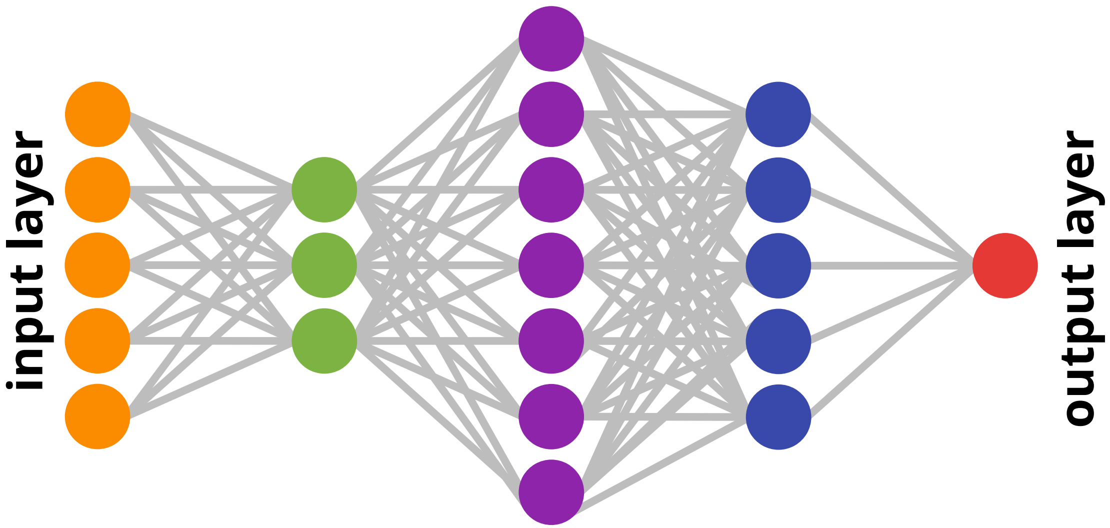
  1 epoch

  - Evaluate model on batch input data (=prediction) in forward pass
  - Compute loss on prediction and target y
  - Compute weight gradients with backprop.
  - Modify weights based on gradients and learning rate
  - Repeat for all batches
- Repeat for a number of epochs, monitor training and validation loss + metrics
- Stop before overfitting sets in



Supervised learning with Neural Networks



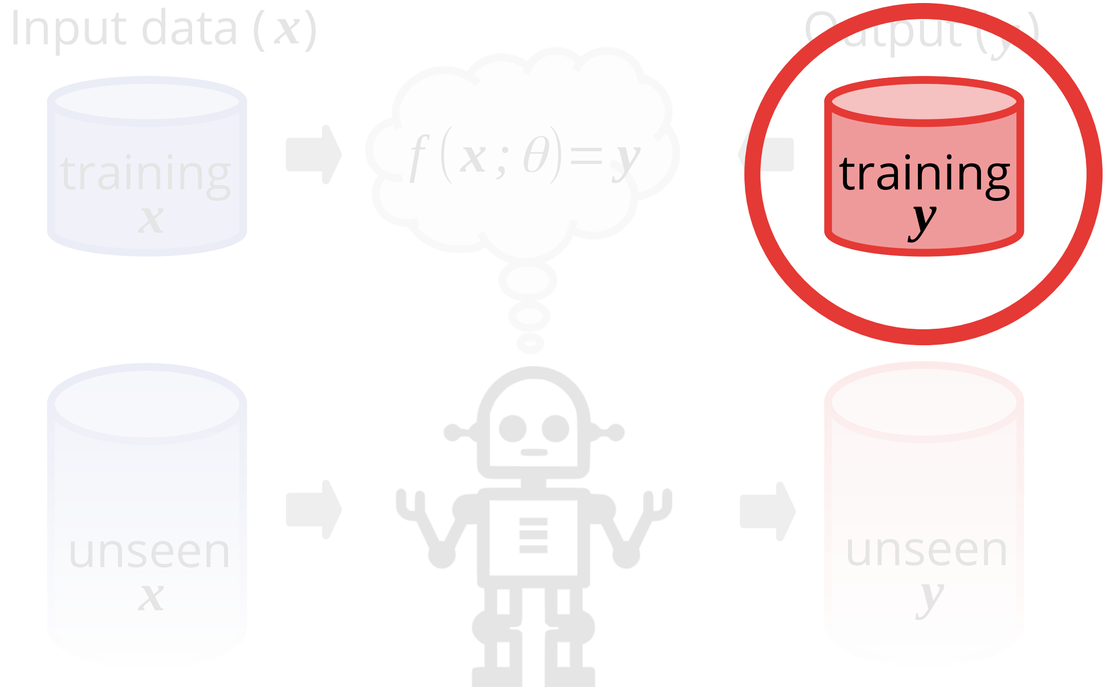


A machine learns a task from **annotated examples**.

Mathematically, it learns a function, f, that maps input data, $\boldsymbol{x}$, to the output, $\boldsymbol{y}$.

A Neural Network is a cascade of mathematical functions; each neuron contains learnable weights that represent the learned knowledge.

Supervised learning with Neural Networks



The availability of annotations typically represents the most important **bottleneck** in supervised learning.

Can we force the model to use the available annotations more **efficiently**?

Can we take advantage of the vast amounts of **unannotated data**?

A machine learns a task from **annotated examples**.

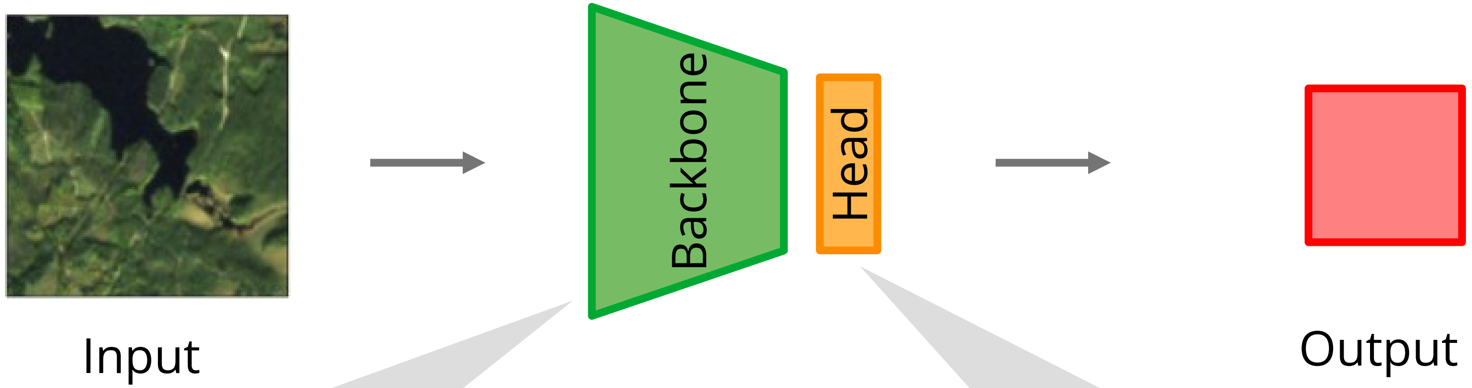
Mathematically, it learns a function, f, that maps input data, x, to the output, y.

A Neural Network is a cascade of mathematical functions; each neuron contains learnable weights that represent the learned knowledge.

How can we use annotated data more efficiently?

Before we answer this question, let's have a look at how to implement the

Supervised Learning Setup



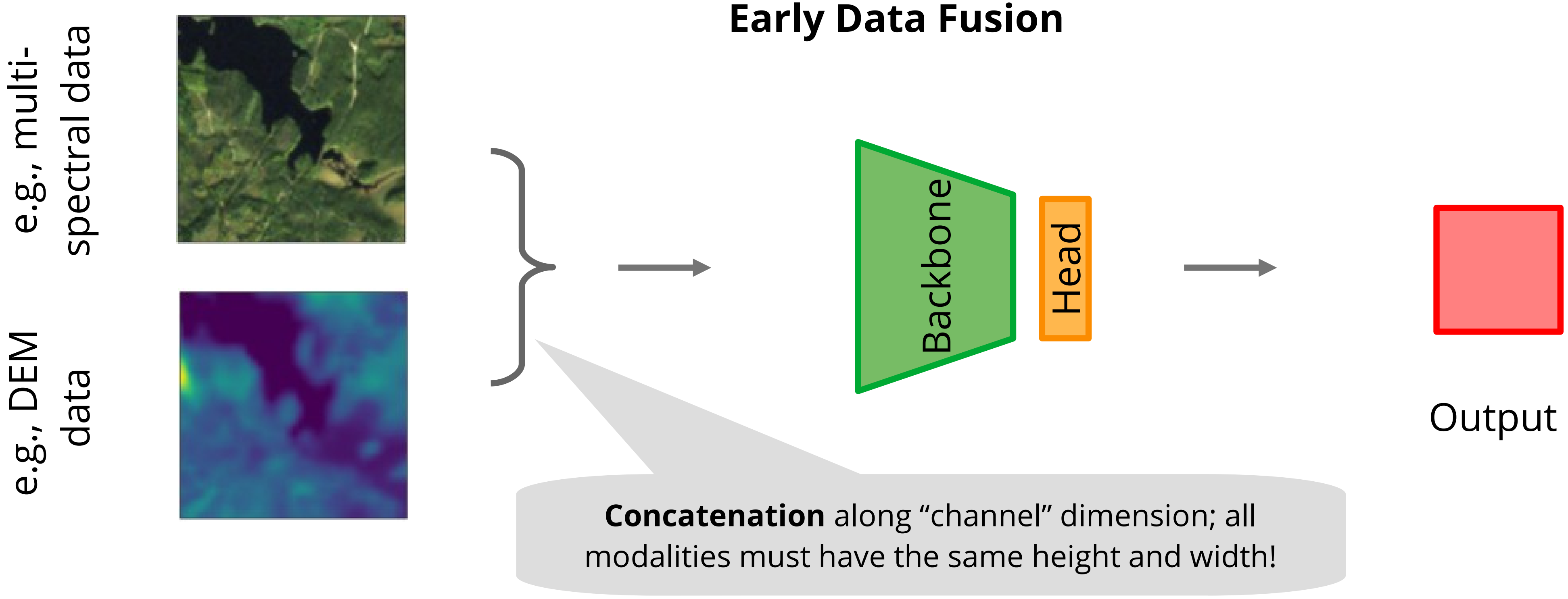
The **backbone** extracts rich representations from the data (e.g., in a CNN, this would be the conv layers)

The **head** uses these representations to learn a specific task (e.g., in a CNN, this would be the linear layers)

How can we use annotated data more efficiently?

We can improve efficiency by leveraging different data modalities in one of two data fusion approaches.

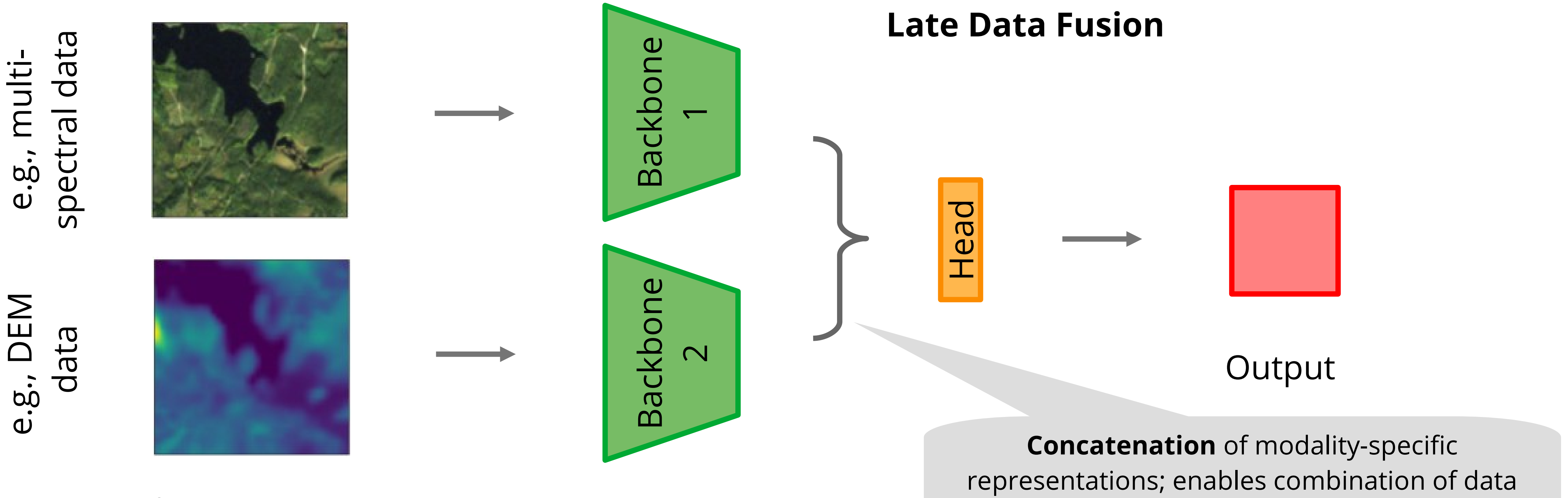
Early Data Fusion



By combining different data modalities early on, we provide additional information to our model.

How can we use annotated data more efficiently?

We can improve efficiency by leveraging different data modalities in one of two data fusion approaches.

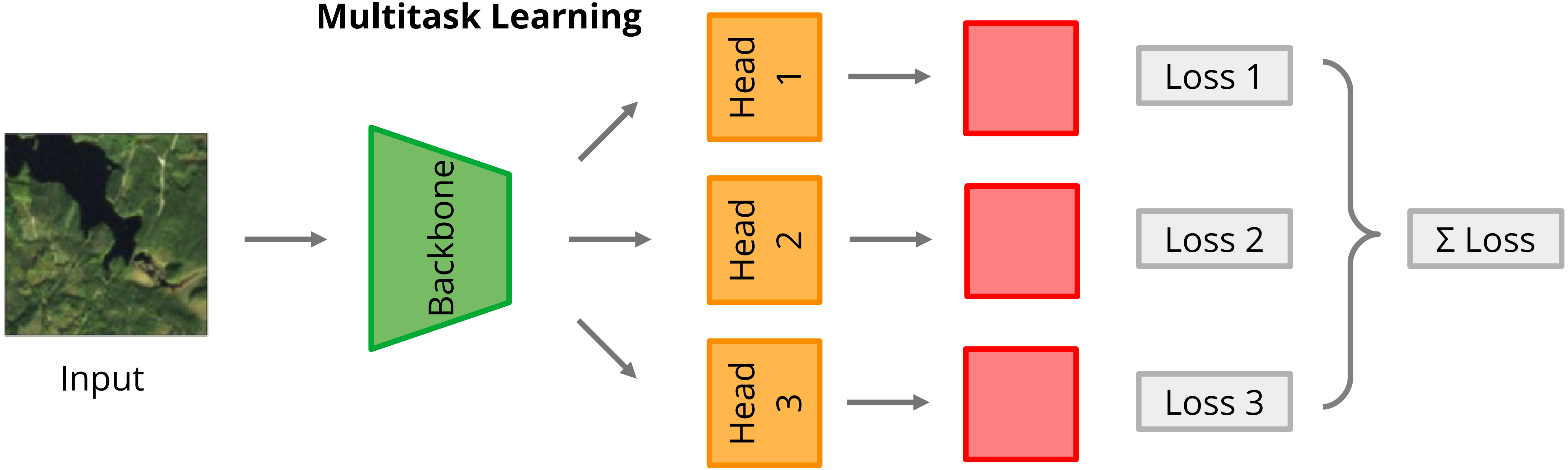


Late data fusion combines representations from different data modalities, representing a more flexible approach for data fusion.

How can we use annotated data more efficiently?

By forcing the network to learn several tasks at the same time, it will focus on extracting only the most relevant information.

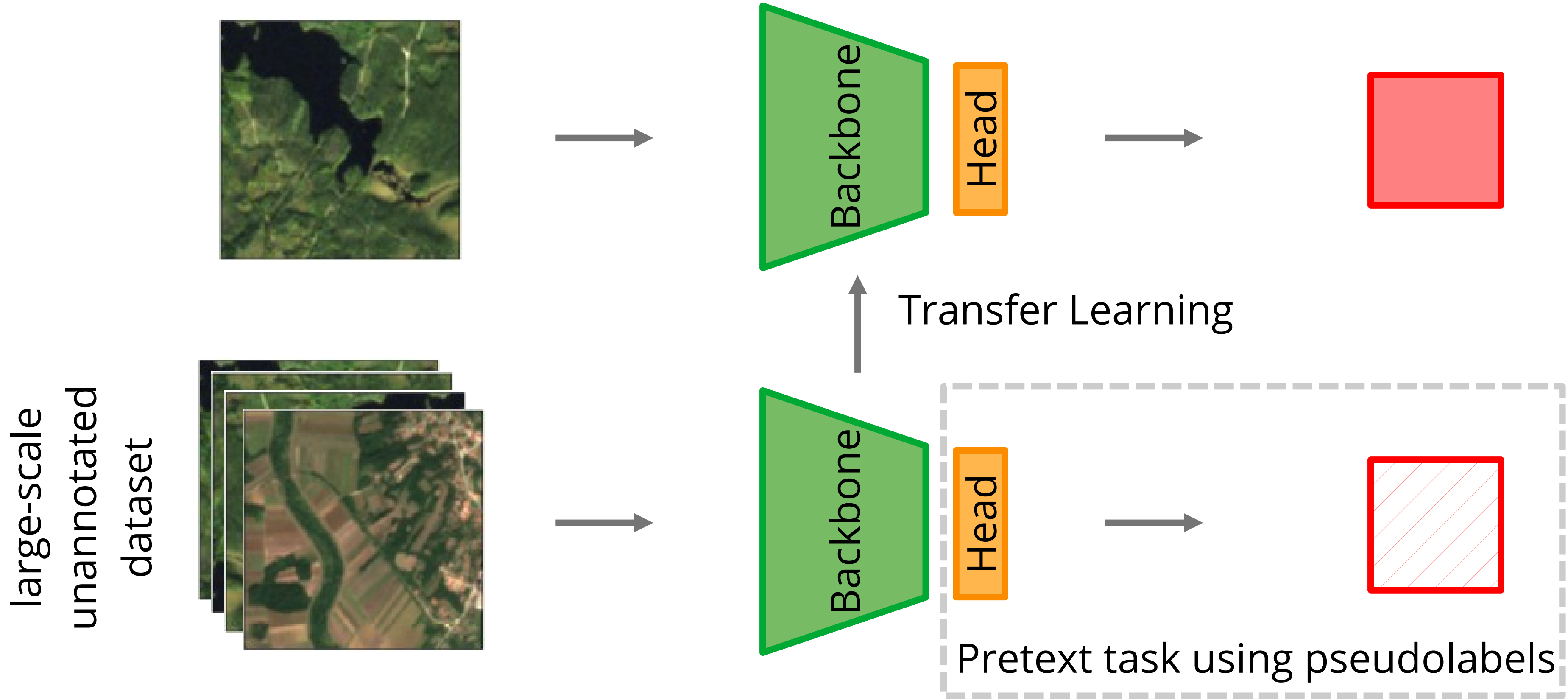
Multitask Learning



The different heads are trained simultaneously by optimizing the (weighted) sum of the individual losses. As a result, the performance on each task is (typically) better than if trained individually.

How can we use unannotated data?

We can leverage (large amounts of) unannotated data for pretraining our model on a suitable pretext task.



Self-supervised Learning

Pretrainig with self-supervised learning drastically reduces the amount of labeled data necessary and generally improves the overall performance of the trained model. Different pretext tasks are available.

Let’s get our hands dirty

In the remainder of this tutorial, we will implement and experiment with some of the aforementioned methods and techniques:

- Data Fusion (Michael)
- Multitask Learning (Joëlle)
- Self-supervised Learning (Linus)

We will discuss these techniques in separate Jupyter Notebooks. The overall structure and methods for data handling and other aspects should be consistent between these Notebooks for easier understanding.

Keep in mind that the results provided here are not really representative, as the dataset that we use is tiny. Feel free to use the code from these Notebooks for your own research!