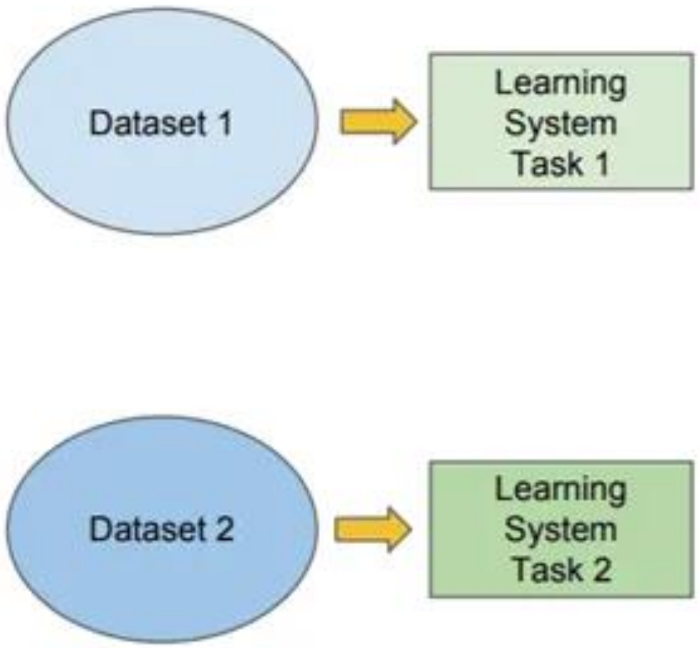
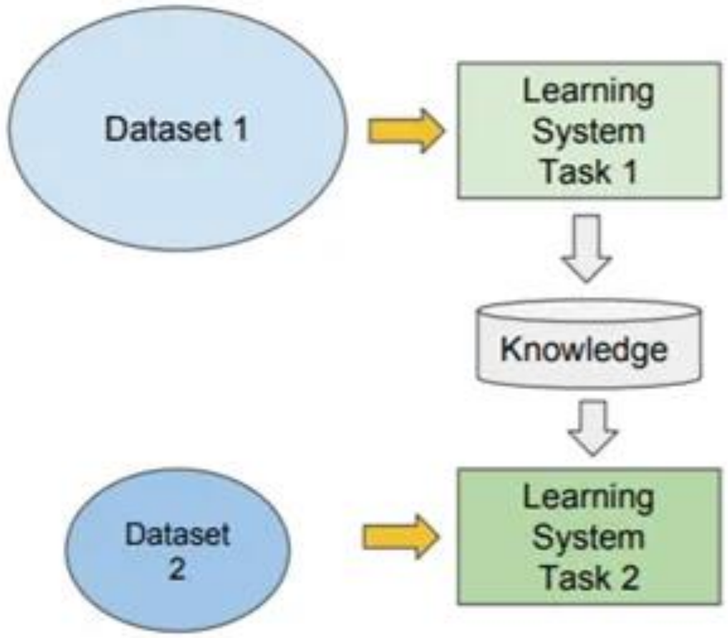
Special learning techniques

Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data



Transfer Learning

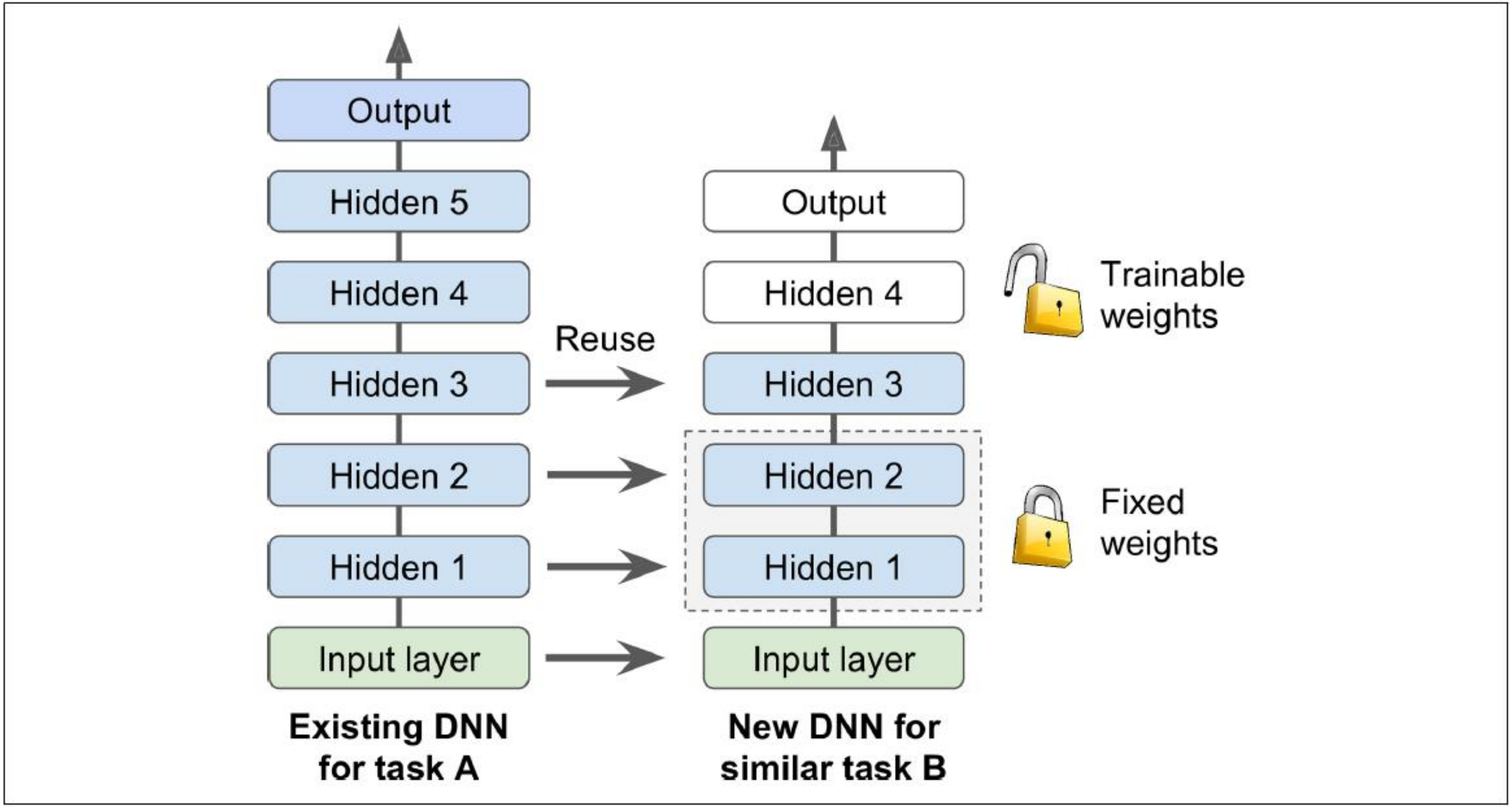
- In transfer learning, you pick an existing model trained on some dataset, and you adapt this model to predict examples from another dataset, different from the one the model was built on. For example, imagine you have trained your model to recognize (and label) wild animals on a big labelled dataset. After some time, you have another problem to solve: you need to build a model that would recognize domestic animals.
- With shallow learning algorithms, you do not have many options: you have to build another big labelled dataset, now for domestic animals.

With neural networks, the situation is much more favourable. Transfer learning in neural networks work like this.

1. You build a deep model on the original big dataset (wild animals).
2. You compile a much smaller labelled dataset for your second model (domestic animals).
3. You remove the last one or several layers from the first model. Usually, these are layers responsible for the classification or regression;
4. You replace the removed layers with new layers adapted for your new problem.
5. You “freeze” the parameters of the layers remaining from the first model.
6. You use your smaller labelled dataset and gradient descent to train the parameters of only the new layers.

Why upper-level layers are removed??

- The output layer of the original model should usually be replaced since it is most likely not useful at all for the new task, and it may not even have the right number of outputs for the new task.
- Similarly, the upper hidden layers of the original model are less likely to be as useful as the lower layers, since the high-level features that are most useful for the new task may differ significantly from the ones that were most useful for the original task.



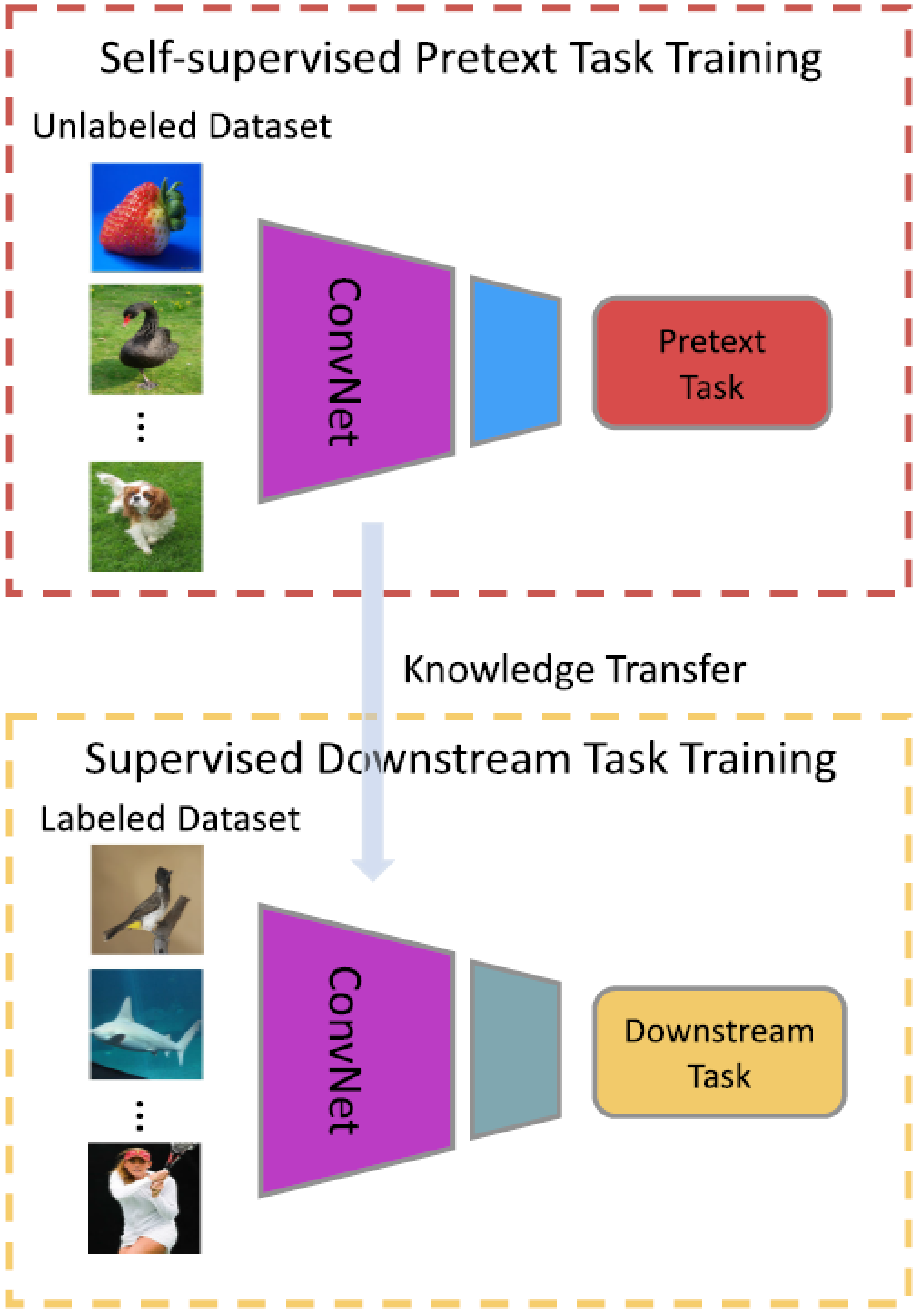
Reusing pretrained layers

Transfer learning

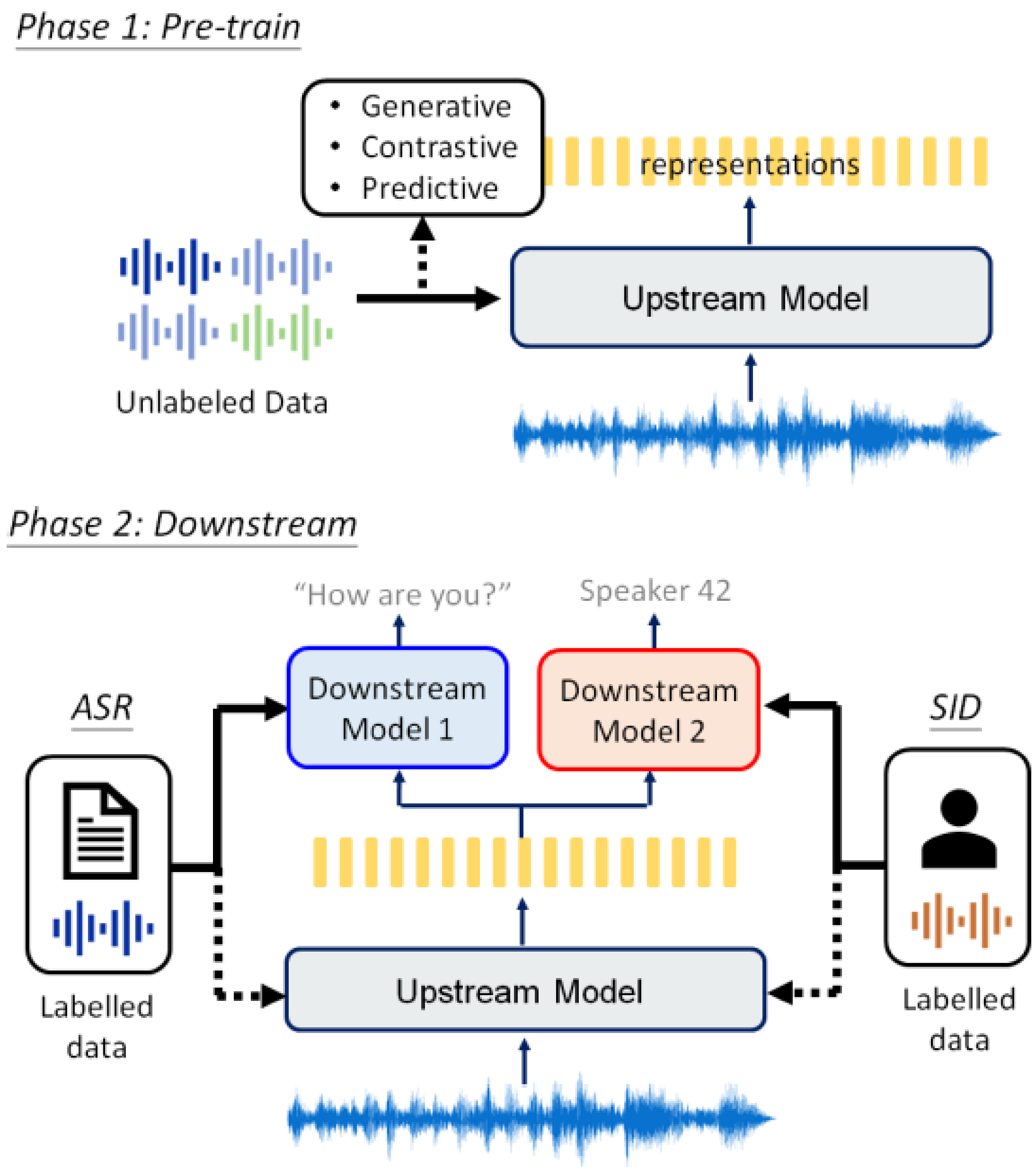
- Apply the knowledge you took in a task A and apply it in another task B.
- For example, you have trained a cat classifier with a lot of data, you can use the part of the trained NN it to solve x-ray classification problem.
- To do transfer learning, delete the last layer of NN and it's weights and:
   i. Option 1: if you have a small data set - keep all the other weights as a fixed weights. Add a new last layer(-s) and initialize the new layer weights and feed the new data to the NN and learn the new weights.
   ii. Option 2: if you have enough data you can retrain all the weights.
- Option 1 and 2 are called **fine-tuning** and training on task A called **pretraining**.

- When transfer learning make sense:
   - Task A and B have the same input X (e.g. image, audio).
   - You have a lot of data for the task A you are transferring from and relatively less data for the task B your transferring to.
   - Low level features from task A could be helpful for learning task B.

- Unlabelled data



Self-Supervised Learning



- Deep supervised learning has achieved great success in the last decade. However, the problem is its heavy dependence on manual labels and vulnerability to attacks have driven people to find other paradigms.
- As an alternative, self-supervised learning (SSL) attracts many researchers for its soaring performance on representation learning in the last several years. Self-supervised representation learning leverages input data itself as supervision and benefits almost all types of downstream tasks

- Self-supervised representation learning methods promise a single universal model that would benefit a wide variety of tasks and domains.

- Such methods have shown success in natural language processing and computer vision domains, achieving new levels of performance while reducing the number of labels required for many downstream scenarios.

- SSL is experiencing progress in three main categories: generative, contrastive, and predictive methods.