Transfer learning

19/03/2025

Vito Paolo Pastore

Deep learning a.y. 2024/2025

Credits

These slides have been built upon the following tutorials or lecture:

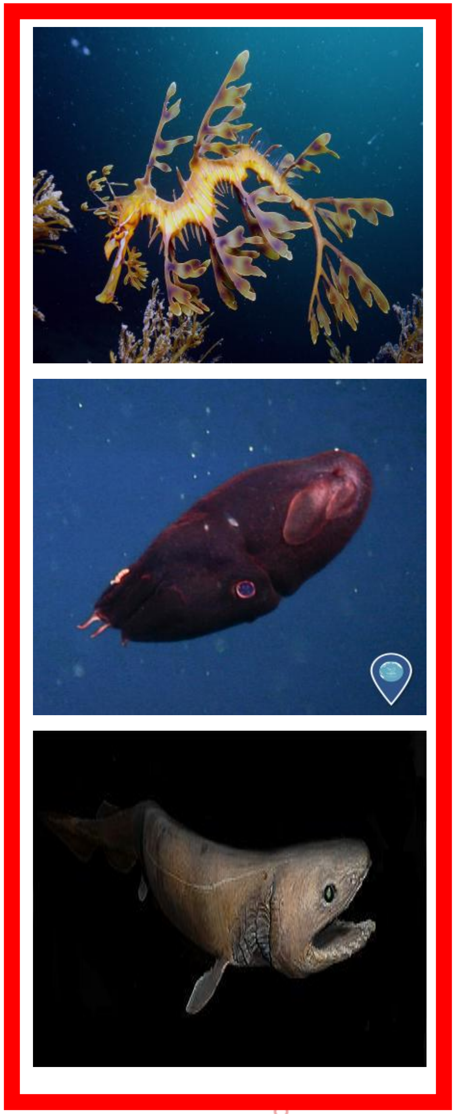
- https://harvard-iacs.github.io/2020F-AC295/lectures/lecture5/
- https://www.cse.cuhk.edu.hk/~byu/CMSC5743/2021Fall/slides/Lec10-KD.pdf

Some slides from:

- Vittorio Murino

An introduction to transfer learning

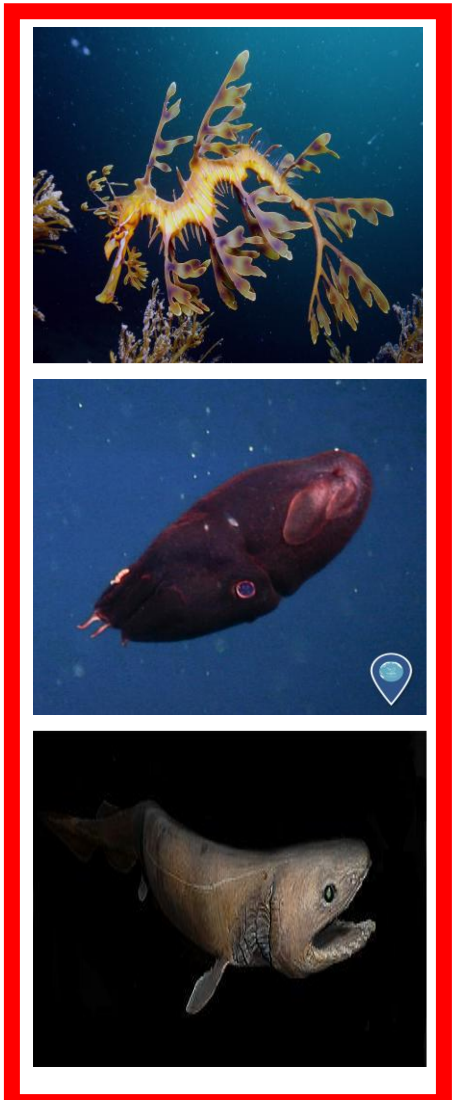
Toy Problem: Classify rare water animals



- Images are difficult to acquire;
- Few hundreds images in total;

UniGe | MaLGa

Naive solution: Training a CNN on available images



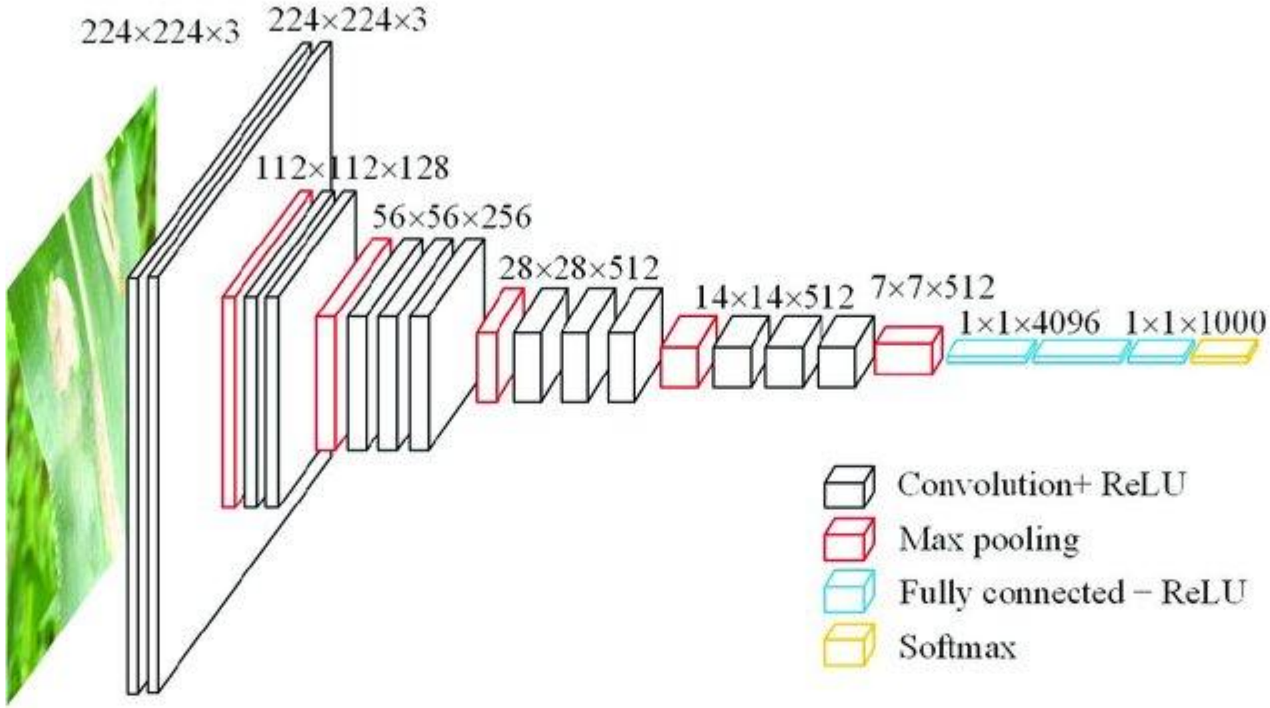
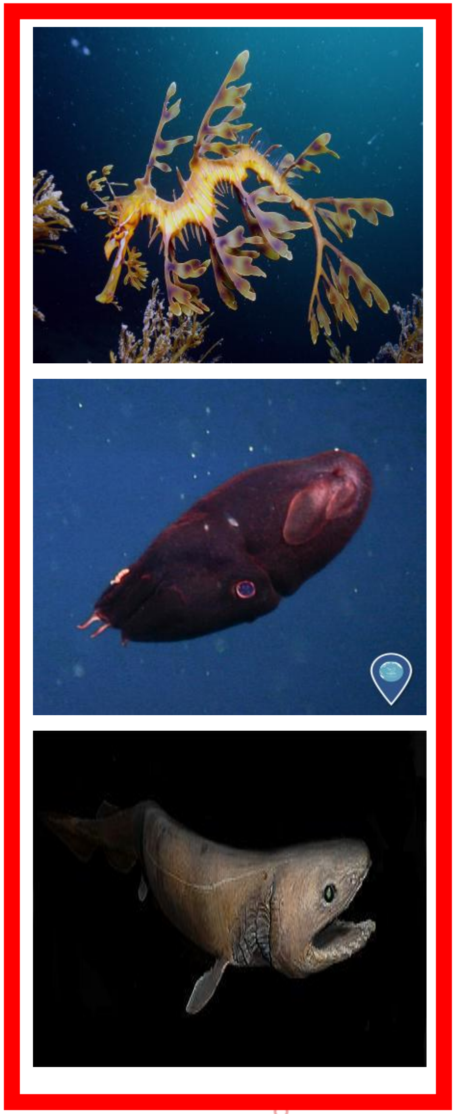
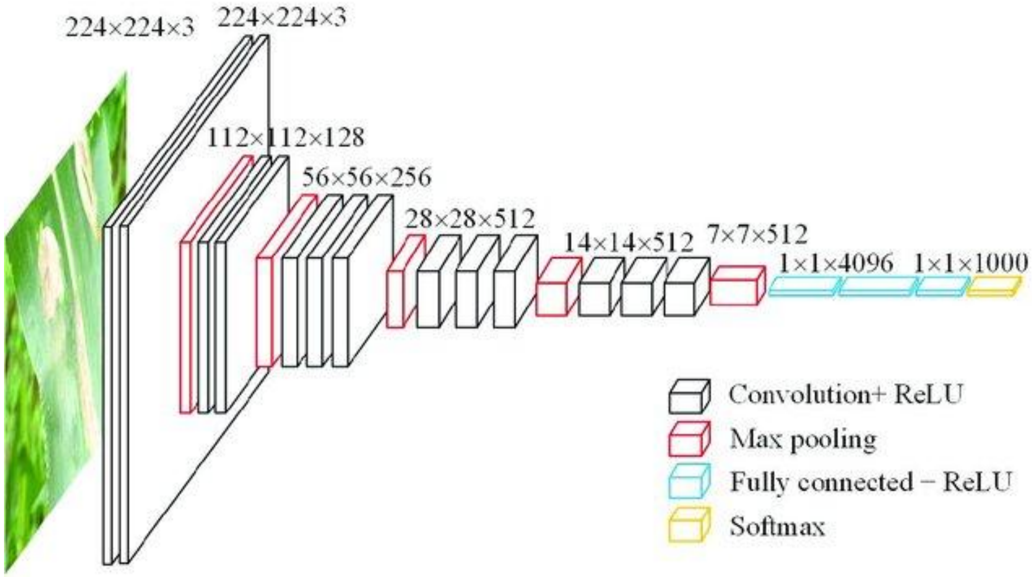


Image from: Fan, Xiangpeng, and Zhibin Guan. "Vgnet: A lightweight intelligent learning method for corn diseases recognition." *Agriculture* 13.8 (2023): 1606.

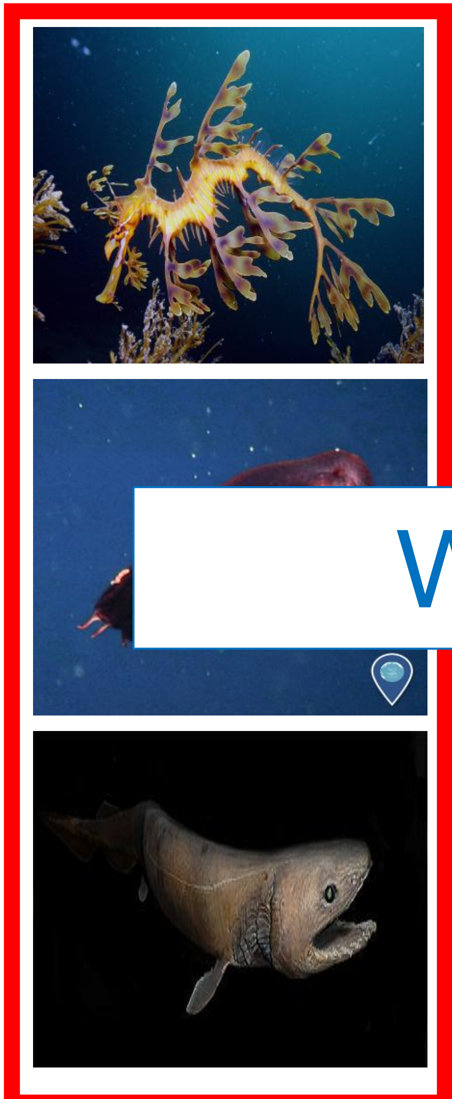
Potential outcome

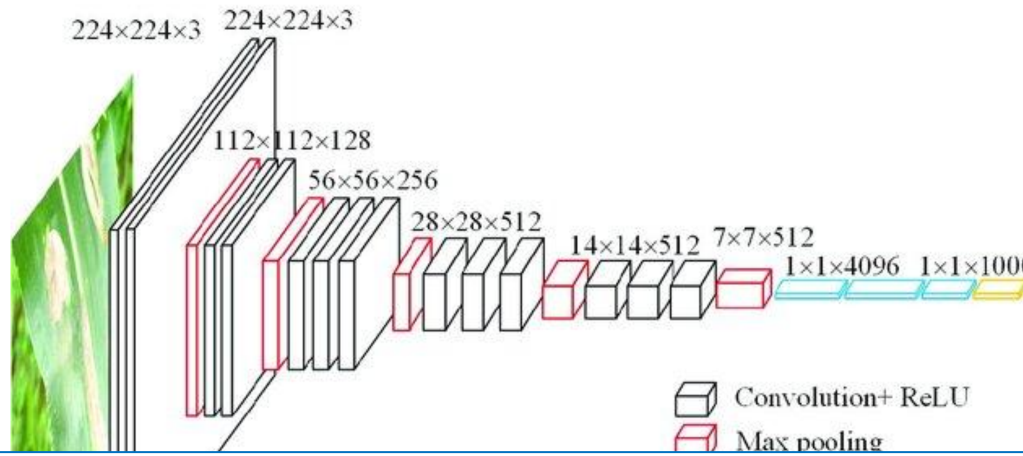




- VGG16 number of parameters (~ 134M);
- Not enough training data to learn the parameters;
- **The model fails to generalize**

Potential outcome

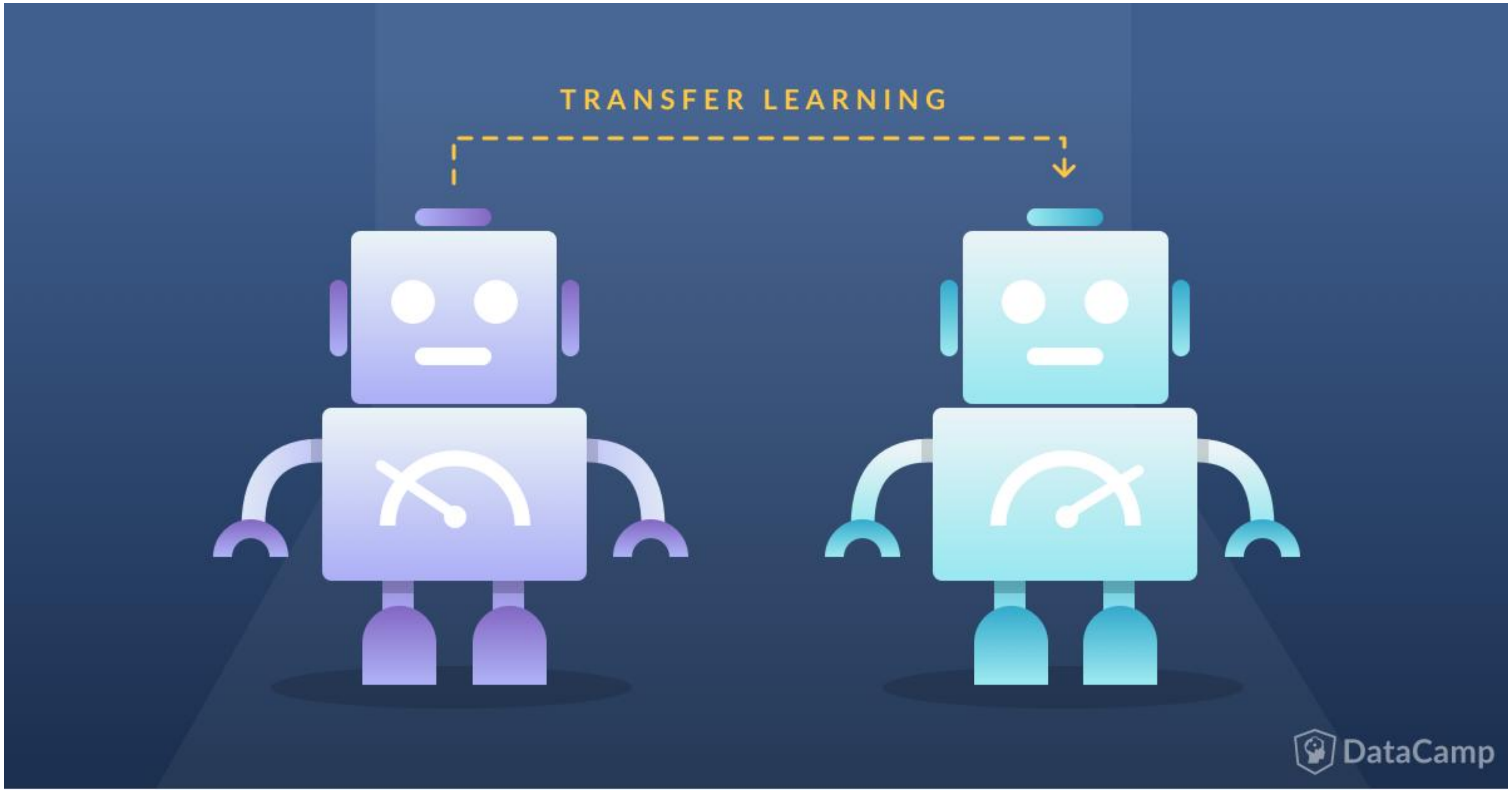




We need more data

- VGG16 number of parameters (~ 134M);
- Not enough training data to learn the parameters;
- **The model fails to generalize**

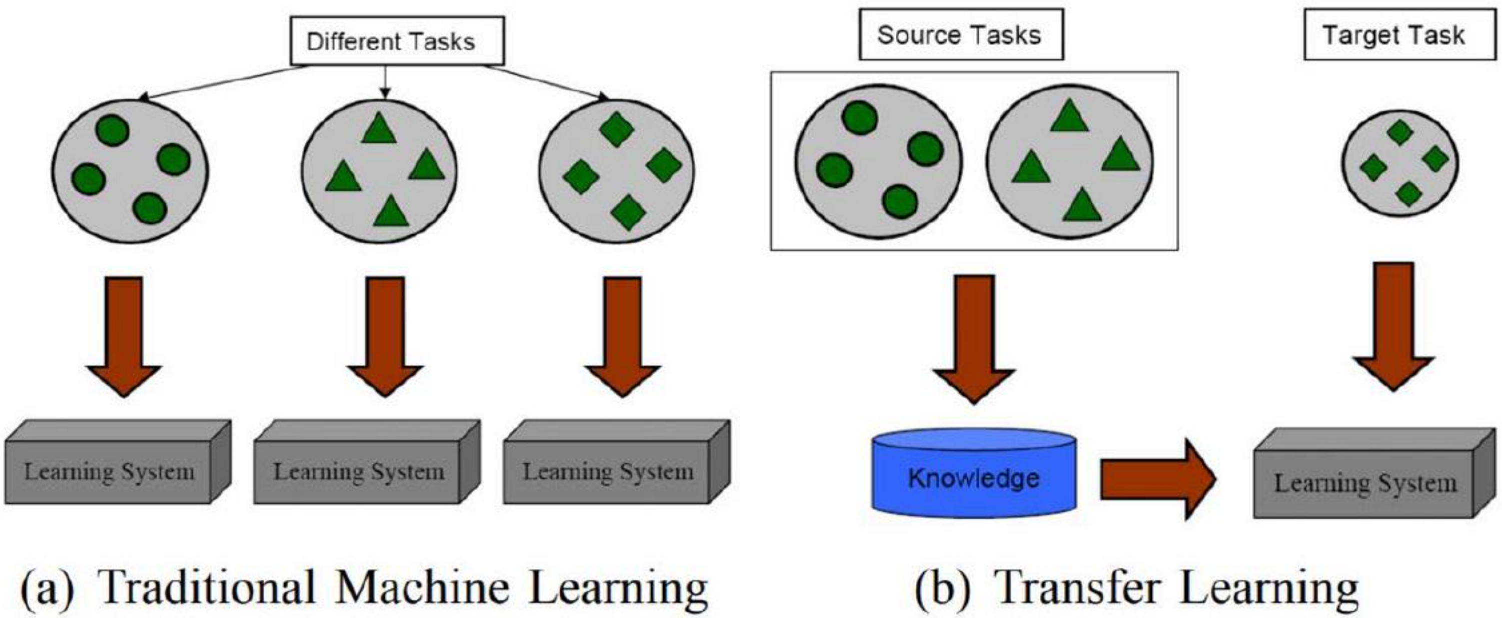
Possible solution



Source image: https://www.datacamp.com/community/tutorials/transfer-learning

Transfer learning

Different Learning Processes between Traditional Machine Learning and Transfer Learning



Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

Transfer learning

*A **domain** $\mathcal{D}$ is defined as a two-element tuple consisting of:*

- *Image/Feature space $\mathcal{X}$*
- *Marginal probability $P(X)$, where X is a sample data point.*

*Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a **task** $\mathcal{T}$ consists of two components:*

- *a label space $\mathcal{Y}$*
- *an objective predictive function $f(\cdot)$, which is not observed but can be learned from the training data, which consist of pairs $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.*

Example:

- *MNIST ->* $\mathcal{X}$ (pixel values of 28x28 gray scale images); P(X) (specific distribution for hand-written digits images)

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

Domain and tasks (examples)

*Given a source domain $\mathcal{D}_S$, a corresponding source task $\mathcal{T}_S$, as well as a target domain $\mathcal{D}_T$ and a target task $\mathcal{T}_T$, **the objective of transfer learning** is to learn the target conditional probability distribution $P(Y_T|X_T)$ in $\mathcal{D}_T$ with the information gained from $\mathcal{D}_S$ and $\mathcal{T}_S$.*

- There are different **possible scenarios** of transfer learning, based on the relationship between $\mathcal{D}_S$, $\mathcal{D}_T$, $\mathcal{T}_S$, $\mathcal{T}_T$

- A limited number of labeled target examples, which is much smaller than the number of labeled source examples, or just unlabeled samples are assumed to be available.

- We will investigate four common scenarios

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering, 22*(10), 1345-1359.

Scenario 1 : Different feature space

$$\boxed{X_S \neq X_T}$$

- The feature spaces of the source and target domain are different..

Example of scenario: document A – the source - is written in one language while document B – the target - is written in a different language

Example of task: cross lingual adaptation. Can we use the weights learned training a model that distinguish phonemes on the source to distinguish those in the target written in another language?

Scenario 2: Different feature space

$$\boxed{P(X_S) \neq P(X_T)}$$

- The marginal probability distributions of source and target domain are different

Example of scenario: MRI and TC images of human lungs

Example of task: Can I use the model trained on MRI to classify a certain disease to do the same on human lungs?

MNIST

SVHN

Second Example

Scenario 2: Different feature space

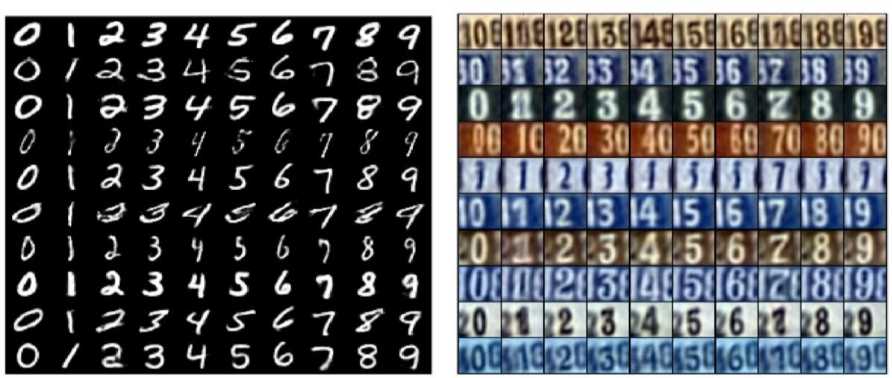
$$\boxed{P(X_S) \neq P(X_T)}$$

Domain adaptation

- The marginal probability distributions of source and target domain are different

Example of scenario: MRI and TC images of human lungs

Example of task: Can I use the model trained on MRI to classify a certain disease to do the same on human lungs?



MNIST SVHN

Second Example

Scenario 3: Different feature space

$$\boxed{Y_S \neq Y_T}$$

- The label spaces between the two tasks are different.

Example of Scenario: ImageNet natural image classes as source, and farm animals as classes in the target.

Scenario 4: Different conditional distributions

$$\boxed{P(Y_S|X_S) \neq P(Y_T|X_T)}$$

- The conditional probability distributions of the source and target tasks are different, e.g. source and target domains are unbalanced with regard to their classes.

Key aspects

- **What to transfer:** Identify which part of the knowledge can be transferred from the source to the target in order to improve the performance of the target task. Understand what is domain/task-specific and what is common between the source and the target.

- **When to transfer:** We aim to improve performance on target task and not degrade them. Need to avoid *negative transfer*.
  Indeed, performance on *source tasks/domain* should be maintained

- **How to transfer:** Identify algorithmic solutions for transferring the knowledge across domains/tasks.

Feature extraction and fine-tuning

Transfer learning and its strategies – CNN codes

- Very few people train from scratch (with random initialization) a CNN (no data, time - weeks!)
- Instead, it is common to (*let others*) pretrain a ConvNet on a very large dataset (e.g., ImageNet, which contains 1.2 million images with 1000 categories), and then:

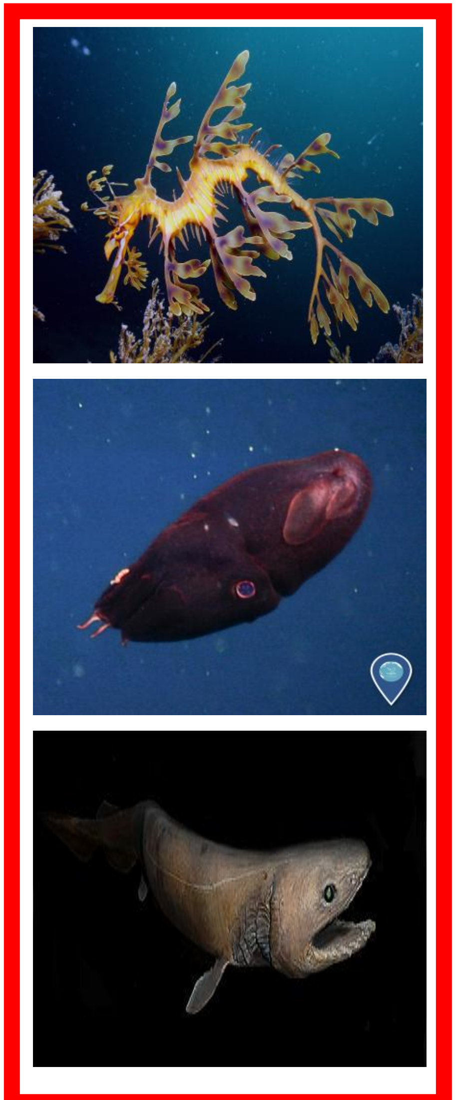
1. **ConvNet as fixed feature extractor**:

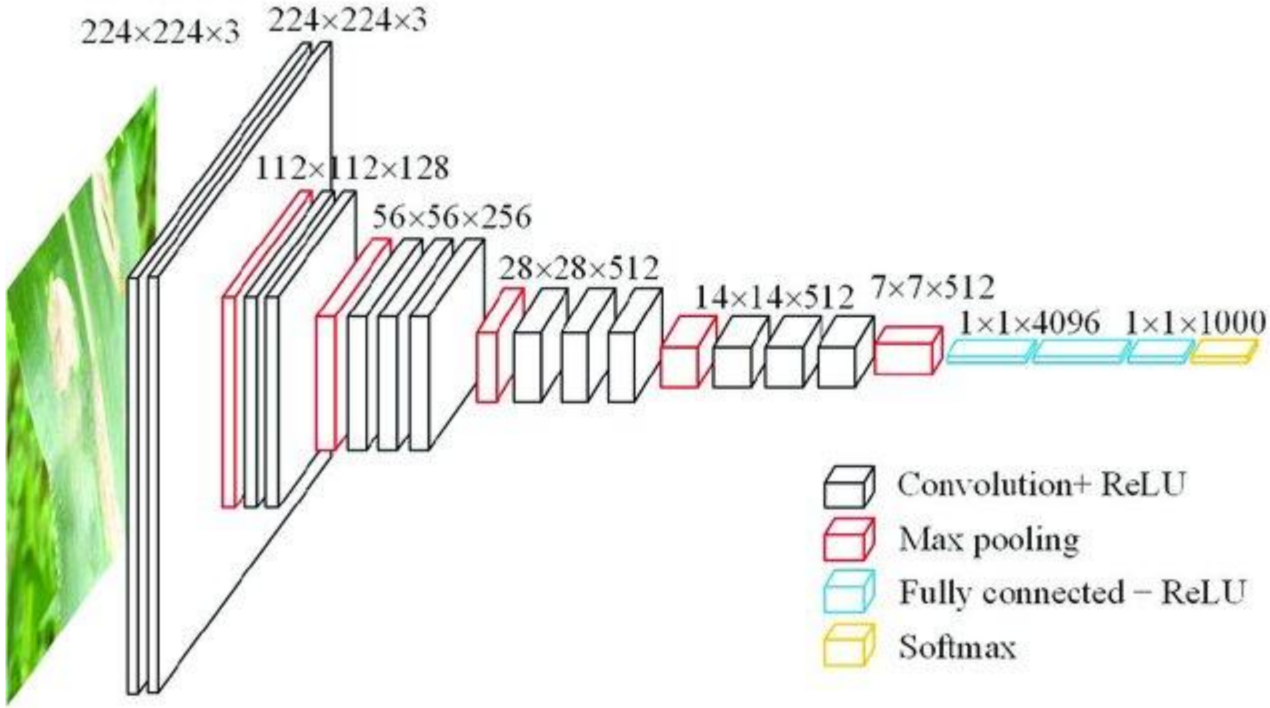
    a. Take a ConvNet pretrained on ImageNet

    b. Remove the last fully-connected layer (this layer's outputs are the 1000 class scores for a different task like ImageNet)

    c. Treat the rest of the ConvNet as a fixed feature extractor for the new dataset.
        - In an AlexNet, this would compute a 4096-D vector for every image that contains the activations of the hidden layer immediately before the classifier.

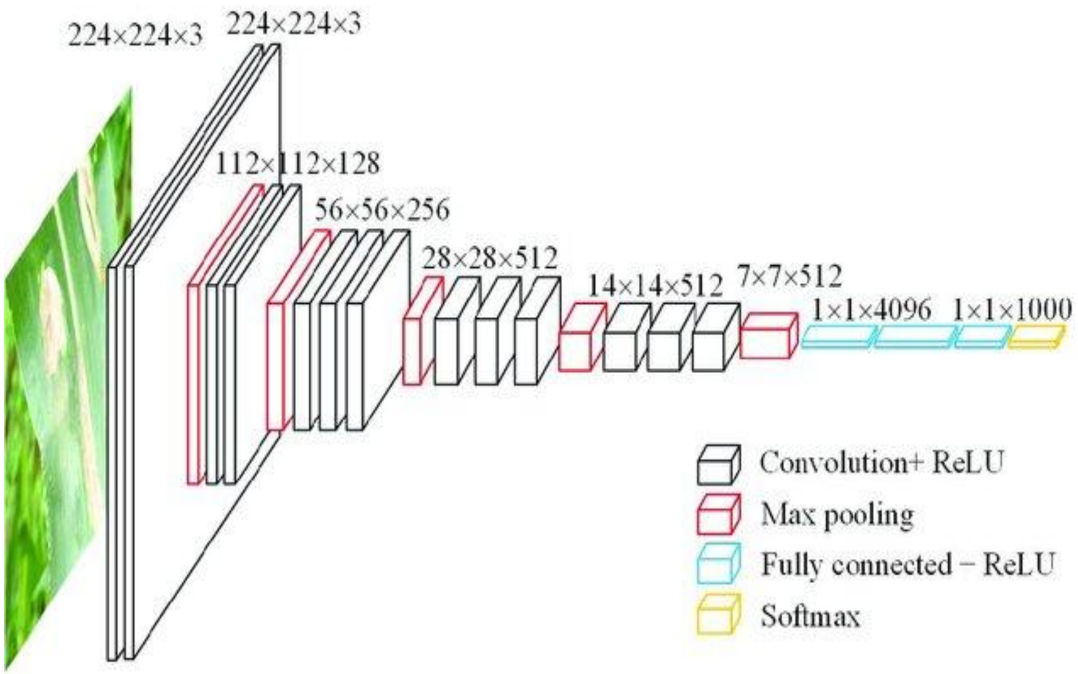
Back on the original toy problem





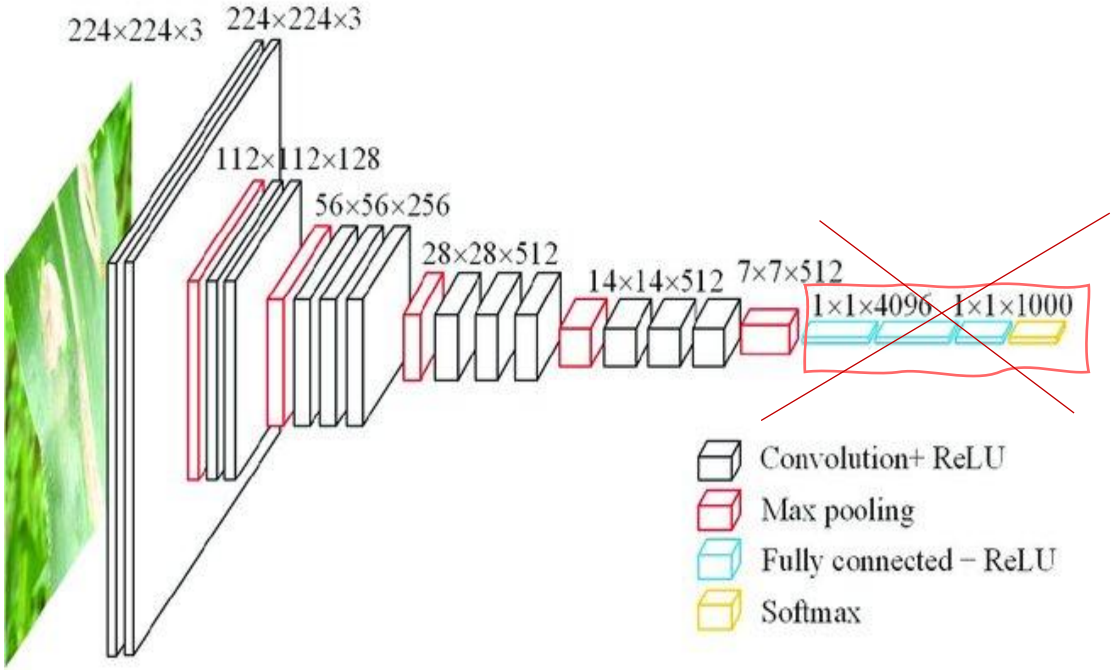
We can train our CNN on a large-scale dataset

https://cs.stanford.edu/people/karpathy/cnnembed/



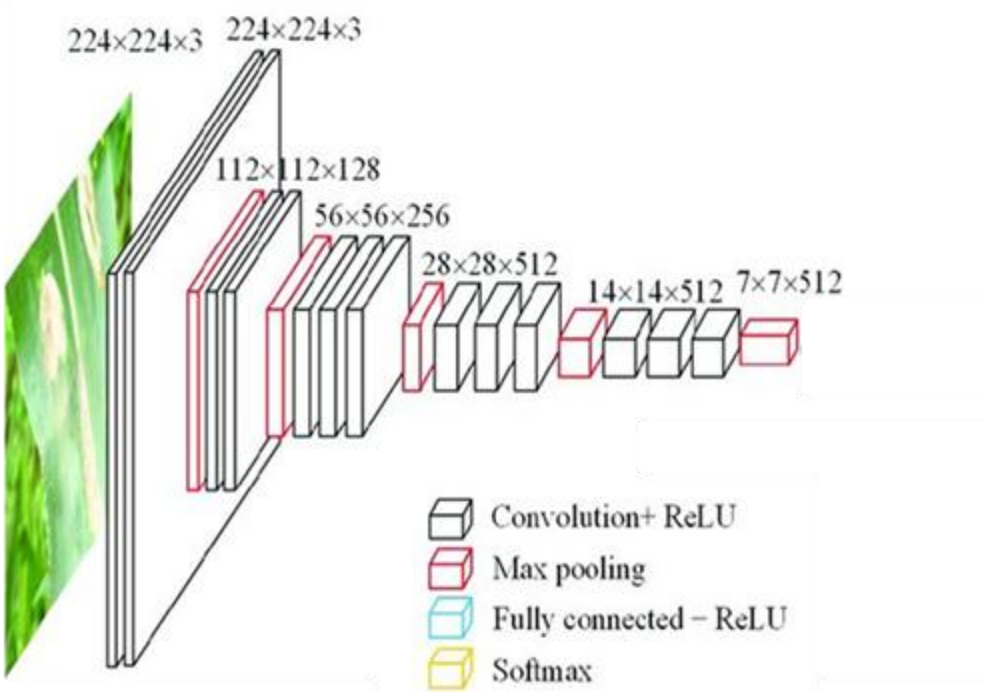
Now re remove the fully connected layers

https://cs.stanford.edu/people/karpathy/cnnembed/

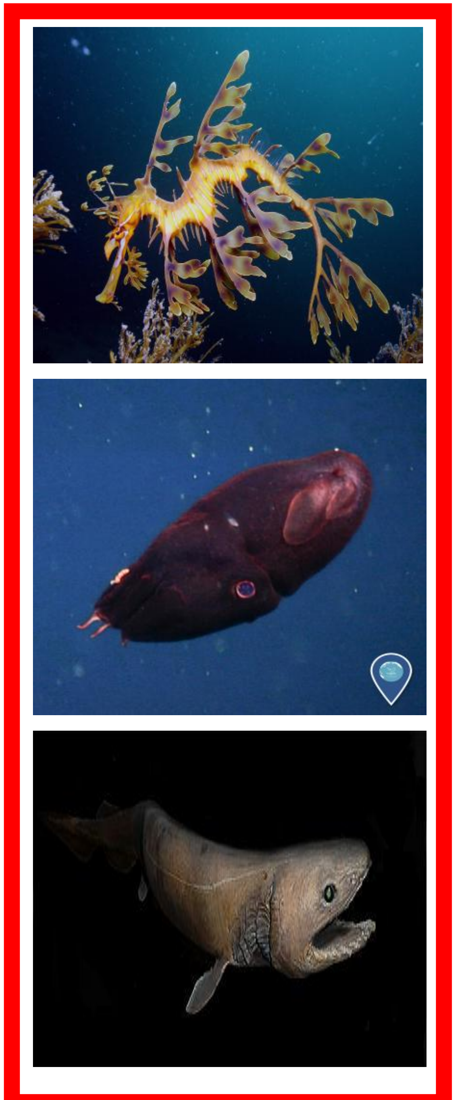


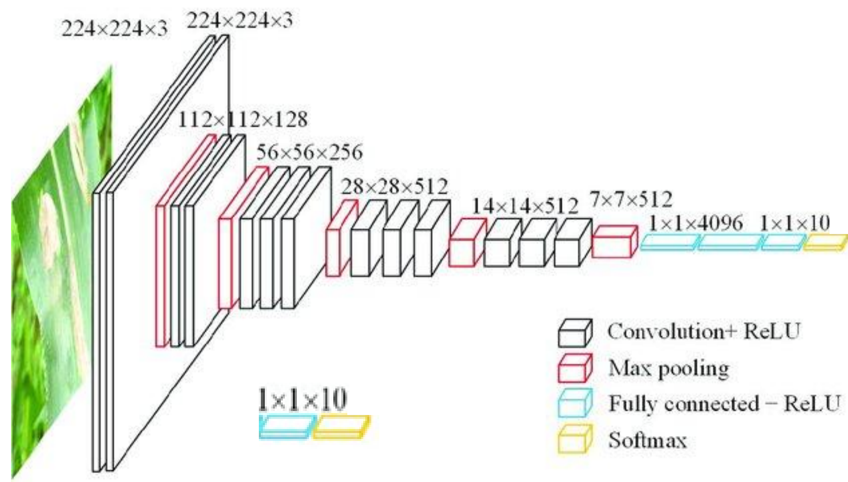
Now remove the fully connected layers

https://cs.stanford.edu/people/karpathy/cnnembed/

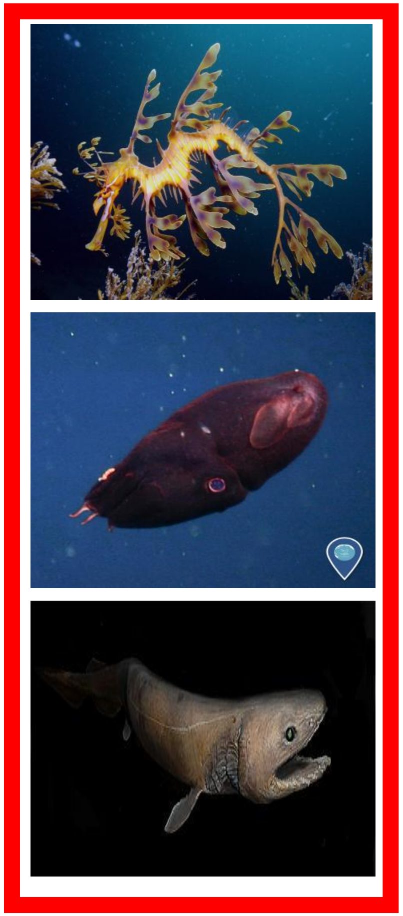


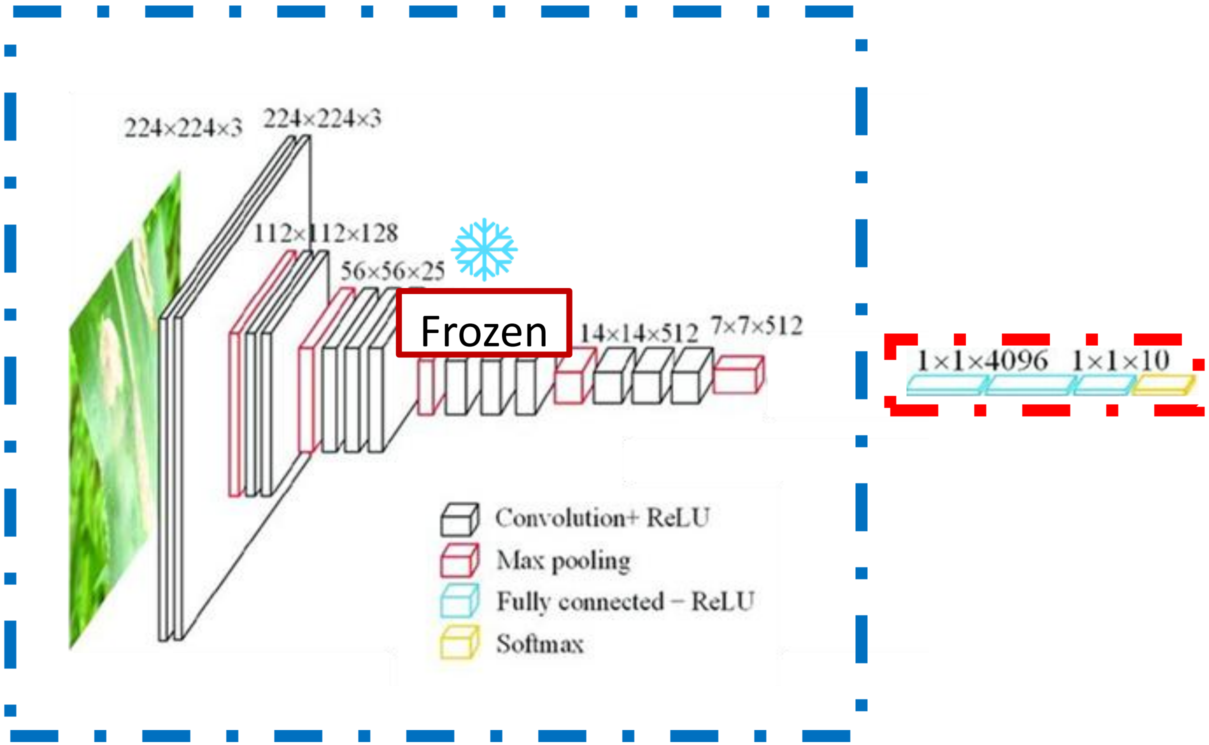
Option 1: Feature extraction and classifier





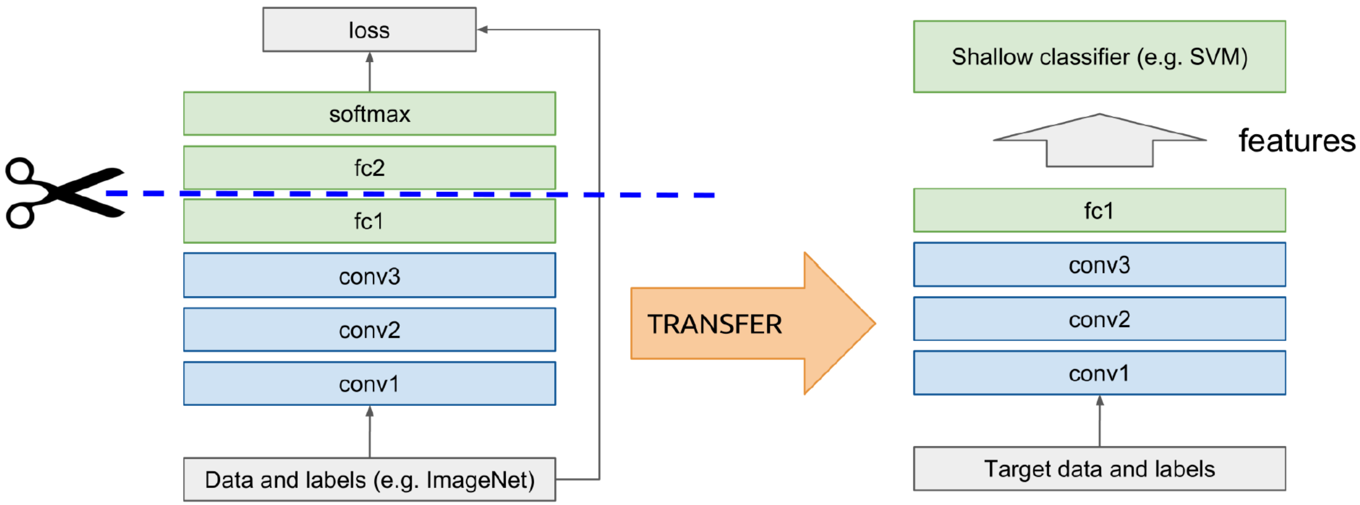
Option 1: Feature extraction and classifier





- Exploit the pre-trained convolutional part as feature extractor (1x1x4096);
- Train a classifier on top of these features

Transfer learning – Fine tuning



- Fine-tune a pre-trained model
- Effective in many applications: computer vision, audio, speech, natural language processing

Transfer learning – Fine tuning

1. **Fine tuning**:
    a. Start with an initialization already computed by backpropagation
    b. Do backpropagation on the layers you want
        - Usually, only the last layers are trained, the earlier are more generic and are preferred to be left unchanged

In particular, four scenarios are available:

- *New dataset is <u>small</u> and <u>similar</u> to original dataset (**NO FINE TUNING**)*. Since the data is small, it is not a good idea to fine-tune the ConvNet due to overfitting concerns. Since the data is similar to the original data, we expect higher-level features in the ConvNet to be relevant to this dataset as well. Hence, the best idea might be to train a linear classifier on the CNN codes.

- *New dataset is <u>large</u> and <u>similar</u> to the original dataset*. Since we have more data, we can have more confidence that we won't overfit if we were to try to fine-tune through the full network.

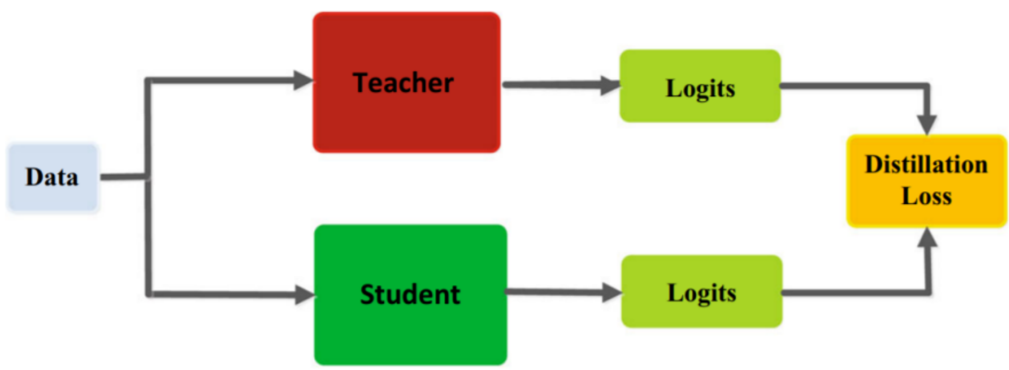
Transfer learning – Fine tuning

- *New dataset is <u>small</u> but <u>very different</u> from the original dataset*. Since the data is small, it is likely best to only train a linear classifier. Since the dataset is very different, it might not be best to train the classifier from the top of the network, which contains more dataset-specific features. Instead, it might work better to train the classifier from activations somewhere earlier in the network, but also fine-tuning only few layers may work

- *New dataset is <u>large</u> and <u>very different</u> from the original dataset*. Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch. However, in practice it is very often still beneficial to initialize with weights from a pre-trained model. In this case, we would have enough data and confidence to fine-tune through the entire network.

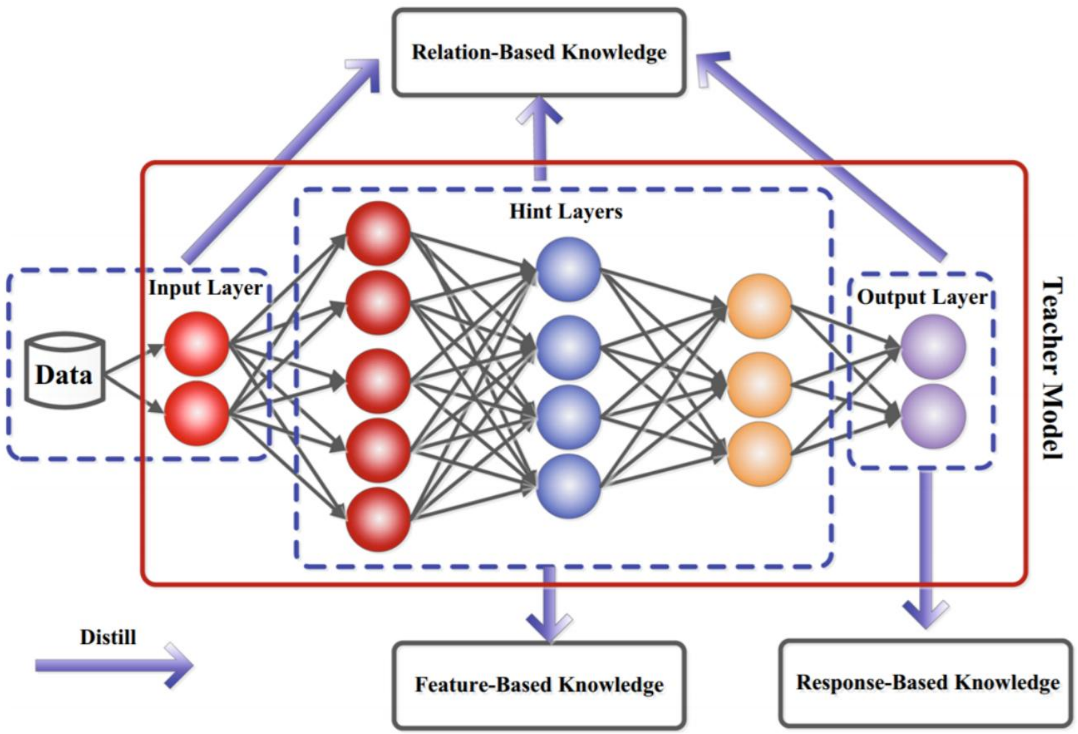
Transfer learning – Knowledge distillation (1)

- Knowledge distillation (KD) is a model compression method in which a small model is trained to mimic a pre-trained, larger model (or ensemble of models).
- This training setting is sometimes referred to as "teacher-student", where the large model is the teacher and the small model is the student.
- In distillation, knowledge is transferred from the teacher model to the student. To simplify, we can say by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model.
- Specifically, KD is accomplished by minimizing the KL divergence between the predictions of teacher and student



Geoffrey Hinton, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network"

Transfer learning – Knowledge distillation (2)



- **Response-based knowledge** usually refers to the neural response of the last output layer of the teacher model.
- **Relation-based knowledge**. Both response-based and feature-based knowledge use the outputs of specific layers in the teacher model;
- **Feature-based knowledge**. The output of intermediate layers, i.e., feature maps, can also be used as the knowledge to supervise the training of the student model, which forged feature-based knowledge distillation.

Take home messages

Transfer learning

- Allows to transfer knowledge from one task to another;
- Knowledge typically means pre-trained weights;
- Fine-tuning and feature extraction are only possible implementation of transfer learning, which includes also other frameworks:
    - Knowledge distillation;
    - Zero-shot and few-shot learning;
    - Self-supervised learning (when the target dataset is different from the source one);
    - Multi-task learning;
    - Continual learning.

Transfer Learning: strategies

Example: ImageNet pre-trained model Fine-tuned on medical images

Self-supervised learning with different datasets

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	*Inductive Transfer Learning / Unsupervised Transfer Learning*	the same	different but related
		different but related	different but related
	Transductive Transfer Learning	different but related	the same

Example: ImageNet pre-trained model Fine-tuned on medical images

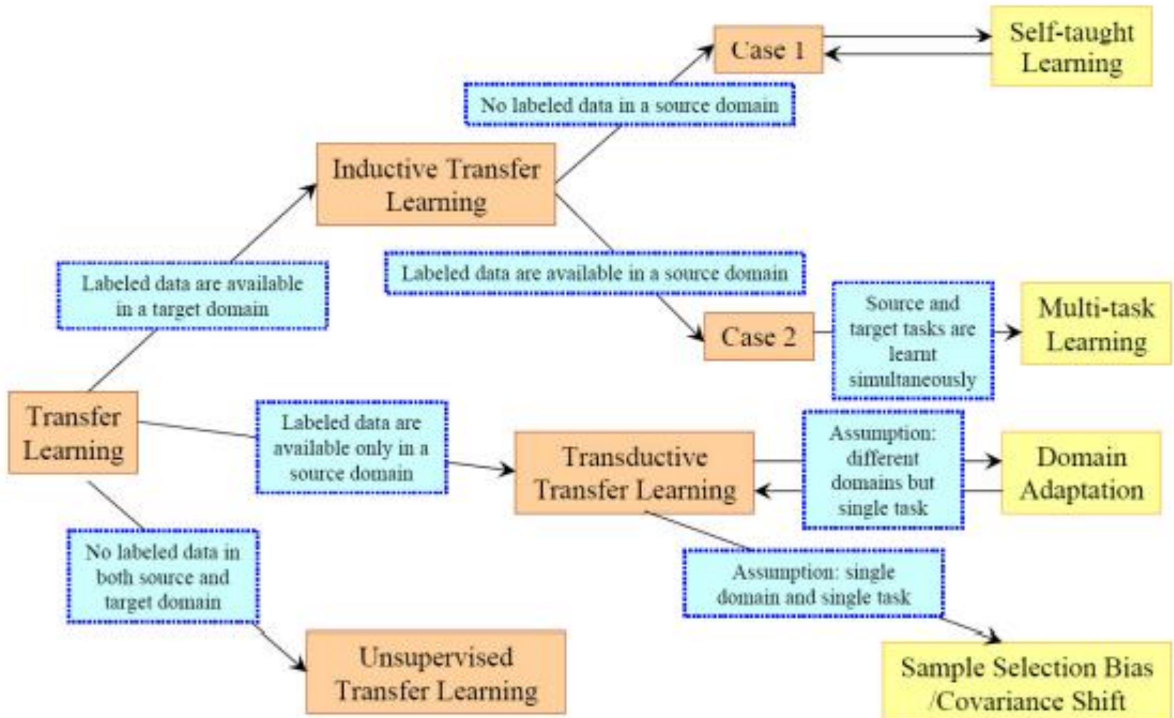
Domain adaptation

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

Taxonomy

UniGe | MaLGa

Transfer Learning: strategies



Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

Taxonomy