CV / VLM

Unit 3: Transfer Learning for CV

TIL-AI
TODAY I LEARNED-AI

3.1.1

Transfer Learning for CV

Fine-tuning vs. feature extraction

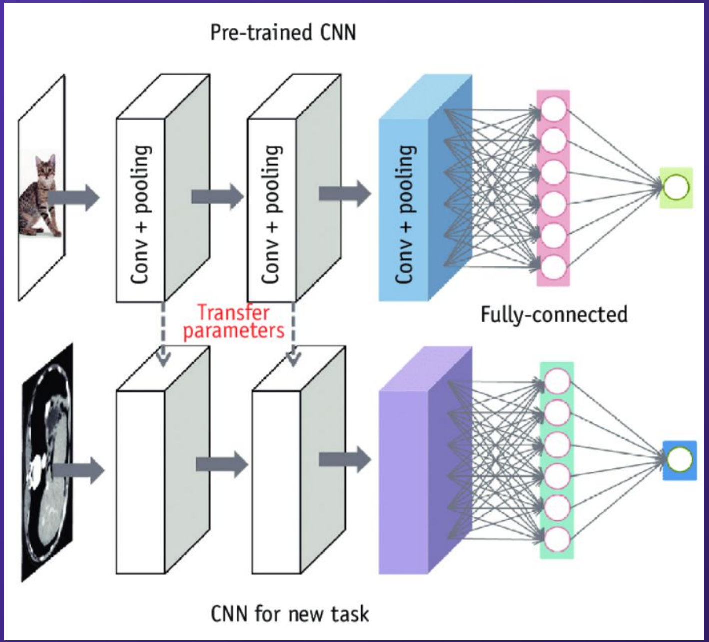
Transfer Learning - Overview

Transfer learning involves adapting a pre-trained model for

I. **Unseen images**
II. **New task**

The two main techniques for adapting the pre-trained model are

1. **Fine-Tuning the pretrain model**
2. **Feature Extraction with transfer learning**



Why Transfer Learning

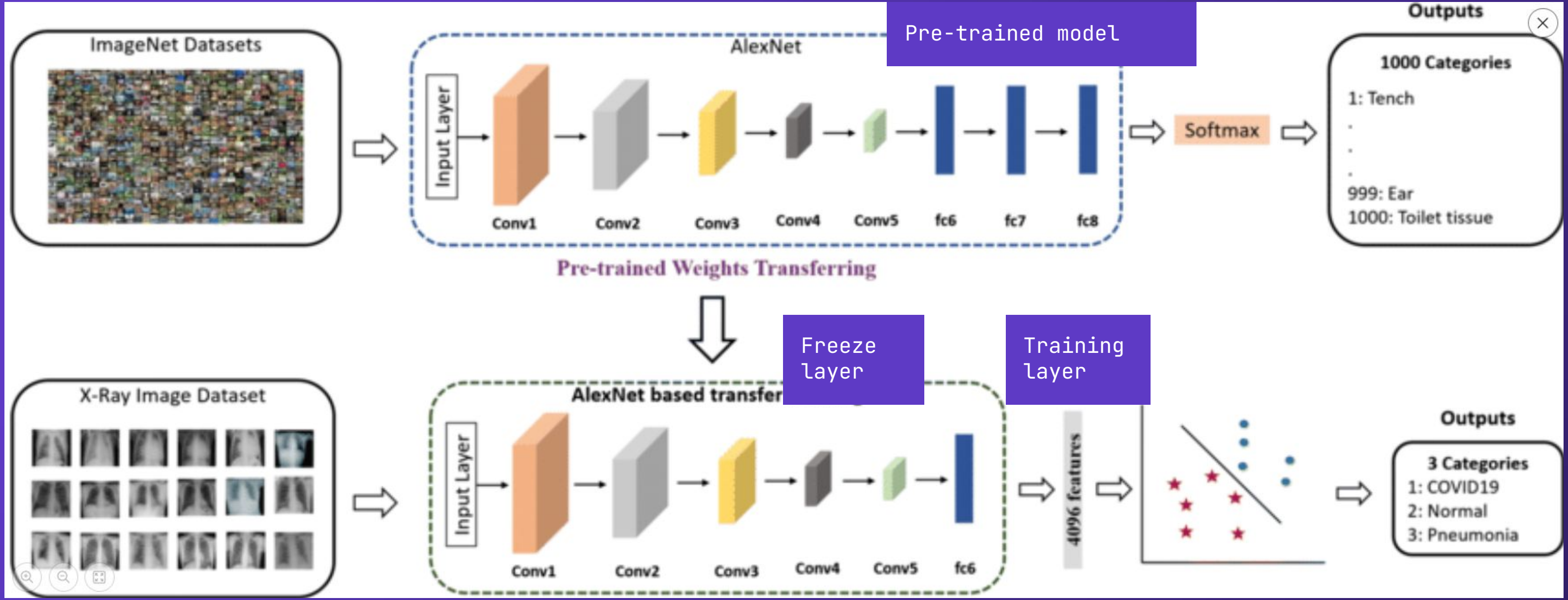
- **Improved Generalization:**
  Transfer learning improves the generalization ability of models. The knowledge gained from a diverse dataset during pre-training allows the model to capture more robust and meaningful features, which can benefit the new task.

- **Knowledge Transfer:**
  When a model is pre-trained on a large dataset (e.g., ImageNet), it learns useful features and representations. These learned features can be transferred to a new task by fine-tuning the model on an unseen dataset specific to it.

- **Time and Resource Savings:**
  Training deep neural networks from scratch is computationally expensive and time-consuming. By using transfer learning, we can save time and computational resources by starting with a pre-trained model and fine-tuning it for the target task.

Transfer Learning

e.g. Adapting AlexNet, trained on ImageNet image categories, to X-ray images for COVID-19 diagnosis



[COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble]

TIL-AI
TODAY I LEARNED-AI

Fine-tuning vs. Feature Extraction

In the context of Computer vision

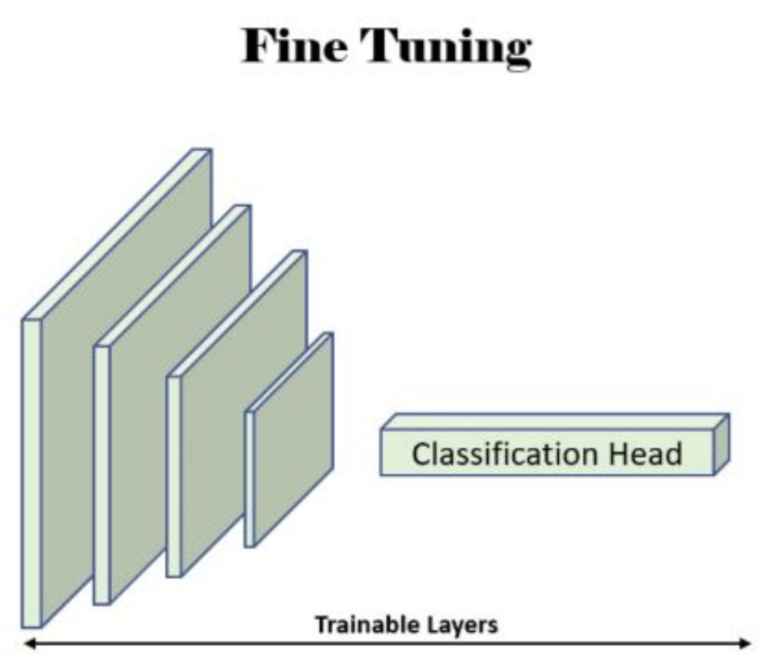
Fine-tuning

- Involves adjusting the pre-trained model's parameters to fit custom task.
- This is done by training the entire model on the new dataset, allowing the model to adapt and learn new patterns.

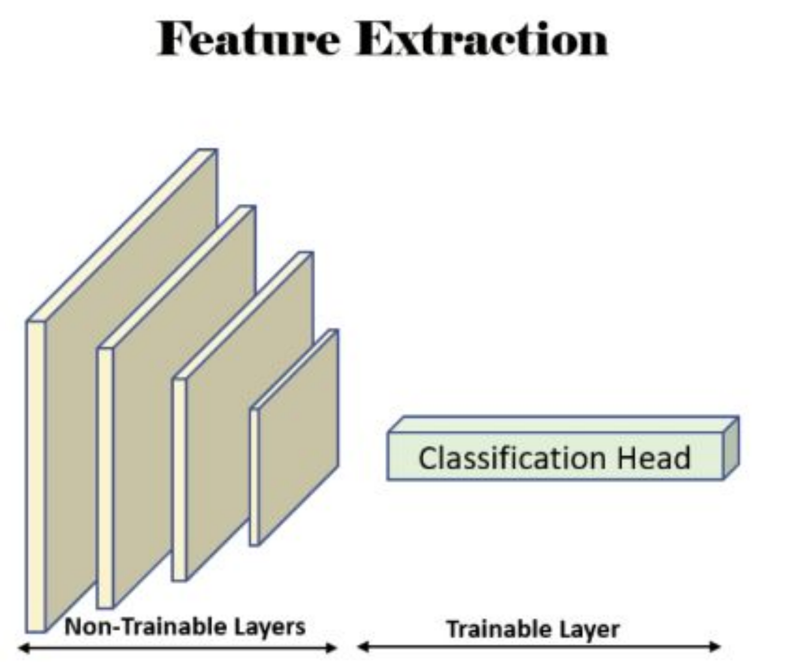
Transfer Learning using Feature extraction

- Involves using the pre-trained model as a feature extractor without modifying its parameters.
- You run your dataset through the pre-trained model and train just a new output layer for the new task.

TIL-AI
TODAY I LEARNED-AI



Fine-tuning



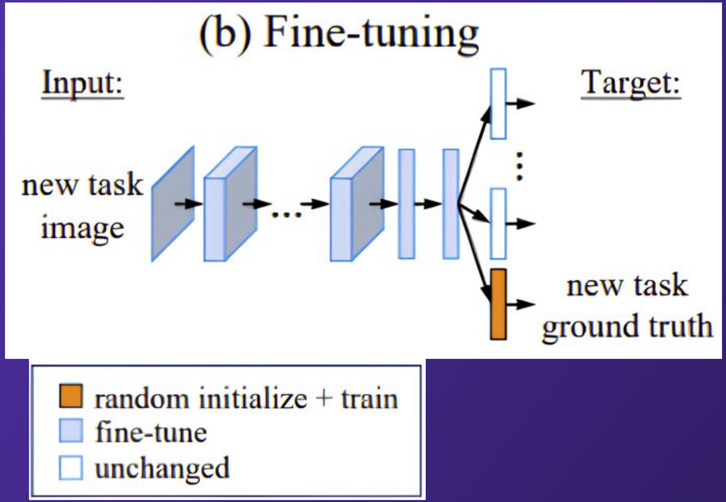
Feature extraction

Fine-Tuning for New Tasks

1. **Instantiating a Pre-Trained Model with Weights**
   Initialize a pre-trained model with its pre-existing weights.

2. **Replacing Classifier Heads**
   Replace the output layer (also known as the classifier head) with a new one that corresponds to the number of categories in our target dataset.

3. **Fine-Tuning the Model**
   Train the new model on our target dataset. During fine-tuning, the parameters of all layers are adjusted to optimize performance on our specific task.



(top) diagram depicting the process of fine tuning a pre-trained model

Note that when fine-tuning, we typically use lower learning rates on pre-trained layers to avoid overwriting the knowledge the model has already gained. By contrast, the output layers (i.e. the new layers) can use more aggressive learning rates.

- [Transfer learning & fine-tuning (keras.io)]

TIL-AI
TODAY I LEARNED AI

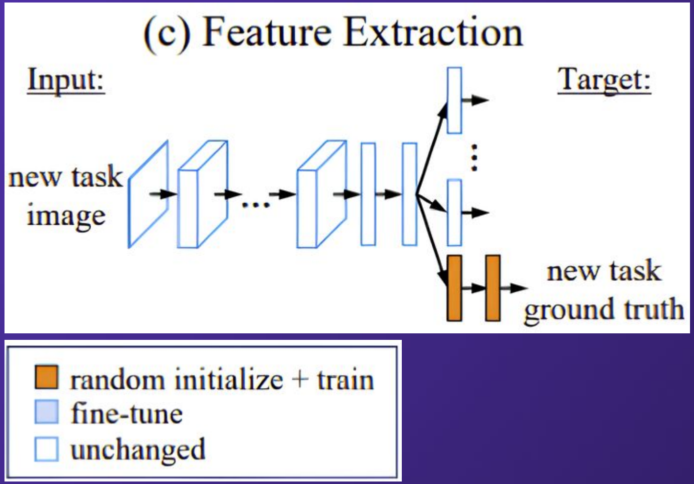
Transfer Learning using Feature Extraction

1. **Instantiating a Pre-Trained Model with Weights**
   Initialize a pre-trained model with its pre-existing weights.

2. **Replacing Classifier Heads**
   Replace the output layer with a new one that corresponds to the number of categories in our target dataset.

3. **Task-Specific Training**
   Freeze all the layers from the pre-trained model, leaving only the outer layer (classifier head) to be trained.

Notes
This technique is known as Transfer Learning using Feature Extraction as it utilizes the original pre-trained model as a feature extractor, with training focusing only on the output layer(s).



(top) diagram depicting the process flow of transfer learning using feature extraction.

- Transfer learning & fine-tuning (keras.io)

Considerations for Fine Tuning vs Feature Extraction

1. **Data Availability & Size:**
   a. **Fine-tuning**: with a large dataset specific to a task, a pre-trained model can adapt its learned features to the data, potentially yielding a better model
   b. **Feature extraction**: with a smaller dataset, fine-tuning might lead to overfitting. In these cases, feature extraction could work better

2. **Task Similarity:**
   a. **Fine-tuning**: if your task is related to the task the pre-trained model was trained on (e.g. classifying dog breeds), fine-tuning is preferred since the model has already learned the relevant features
   b. **Feature Extraction**: if your task is very different from the task the pre-trained model was trained on (e.g. classifying emotions), feature extraction might still provide a valuable starting point for the task

3. **Computational Resources:**
   a. **Fine-tuning**: computationally heavy as more parameters need to be trained
   b. **Feature Extraction**: computationally cheaper

Choosing Transfer Learning Methods

- A good rule of thumb is to start by trying to use feature extraction:
  - This is generally a good default choice, especially for smaller datasets or dissimilar tasks

- Move to fine-tuning if you have large datasets or similar tasks!