Computer Vision: Fall 2022 — Lecture 13

Dr. Karthik Mohan

Univ. of Washington, Seattle

November 11, 2022

References

Generic ML/DL

1. Good Book for Machine Learning Concepts
2. Deep Learning Reference

CNN

1. Convolutional Neural Networks for Visual Recognition
2. Convolutional Neural Net Tutorial
3. CNN Transfer Learning
4. PyTorch Transfer Learning Tutorial

CNN Publication References

CNN surveys

1. Convolutional Neural Networks: A comprehensive survey, 2019
2. A survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, 2021

CNN Archs

1. GoogLeNet
2. Top models on ImageNet
3. ResNet ILSVRC paper

Last couple lectures

1. **Intro to CNN arch** and comparison with NN

Last couple lectures

1. **Intro to CNN arch** and comparison with NN
2. **Evolution** of CNN archs

(ImageNet) ≈ MM data/pts.
↳ (largest ImageData sets 1000 classes

Last couple lectures

1. **Intro to CNN arch** and comparison with NN
2. **Evolution** of CNN archs
3. **Deep dive** into performance, trade-offs, quirks in training CNNs (Hacks)

Last couple lectures

1. **Intro to CNN arch** and comparison with NN
2. **Evolution** of CNN archs
3. **Deep dive** into performance, trade-offs, quirks in training CNNs
4. **Specific archs we looked at**: LeNet, ZFNet, AlexNet, VGGNet, Inception and ResNet

Today

1. Transfer learning in CNN (a.k.a how to not reinvent the wheel with CNN training!)
2. PyTorch Tutorial on Transfer Learning

Transfer Learning

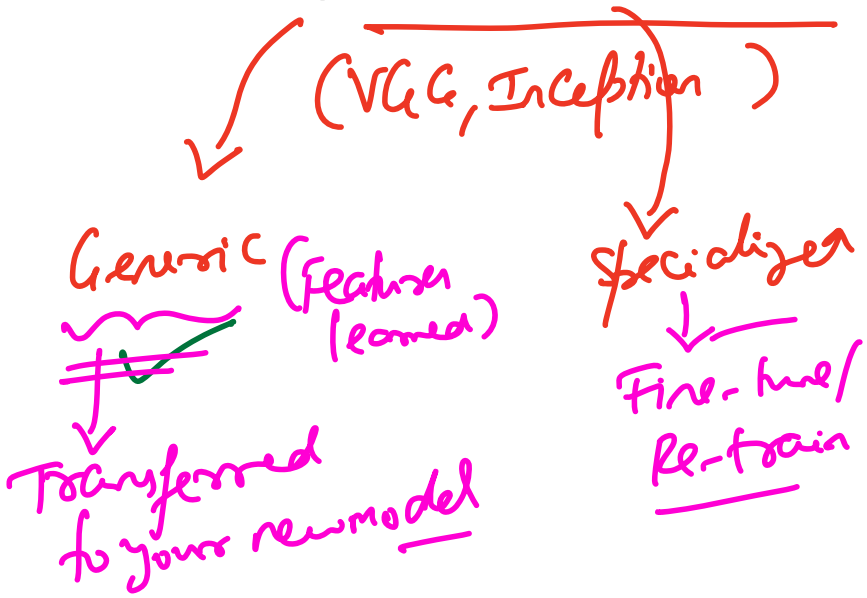
1. **Key Idea:** A model trained on one domain or data set can perhaps transfer to another domain/data set

Transfer Learning

1. **Key Idea:** A model trained on one domain or data set can perhaps transfer to another domain/data set
2. **Key Idea:** With big CNN archs like Inception or ResNet - Training from scratch is cumbersome and wasted resource

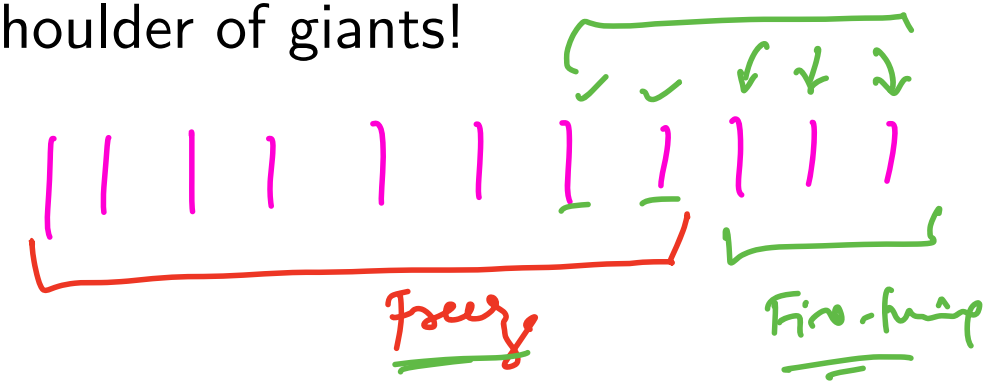
Transfer Learning

1. **Key Idea:** A model trained on one domain or data set can perhaps transfer to another domain/data set
2. **Key Idea:** With big CNN archs like Inception or ResNet - Training from scratch is cumbersome and wasted resource
3. **Transfer Learning:** Instead use **pre-trained models** and **fine-tune** as needed!

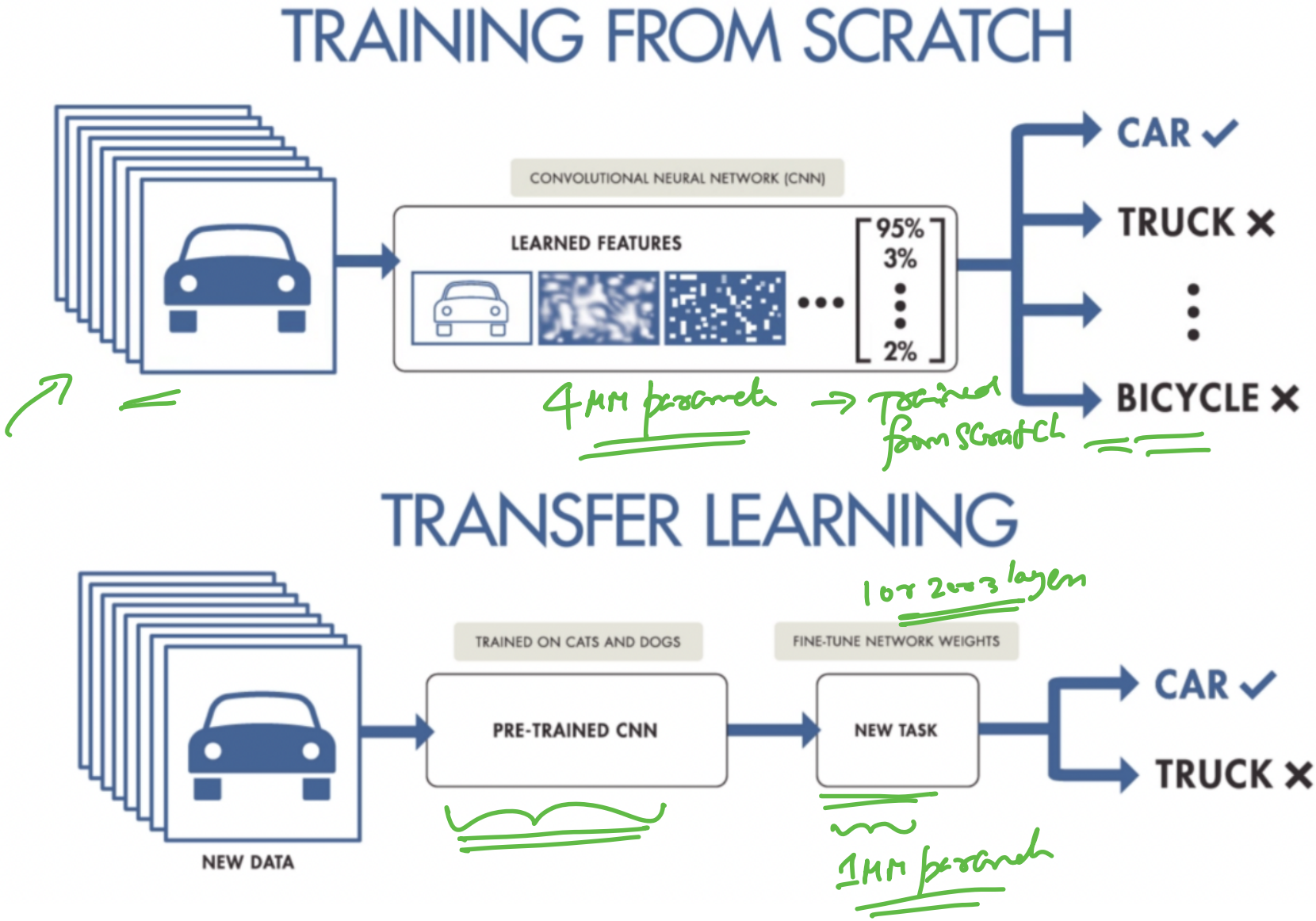


Transfer Learning

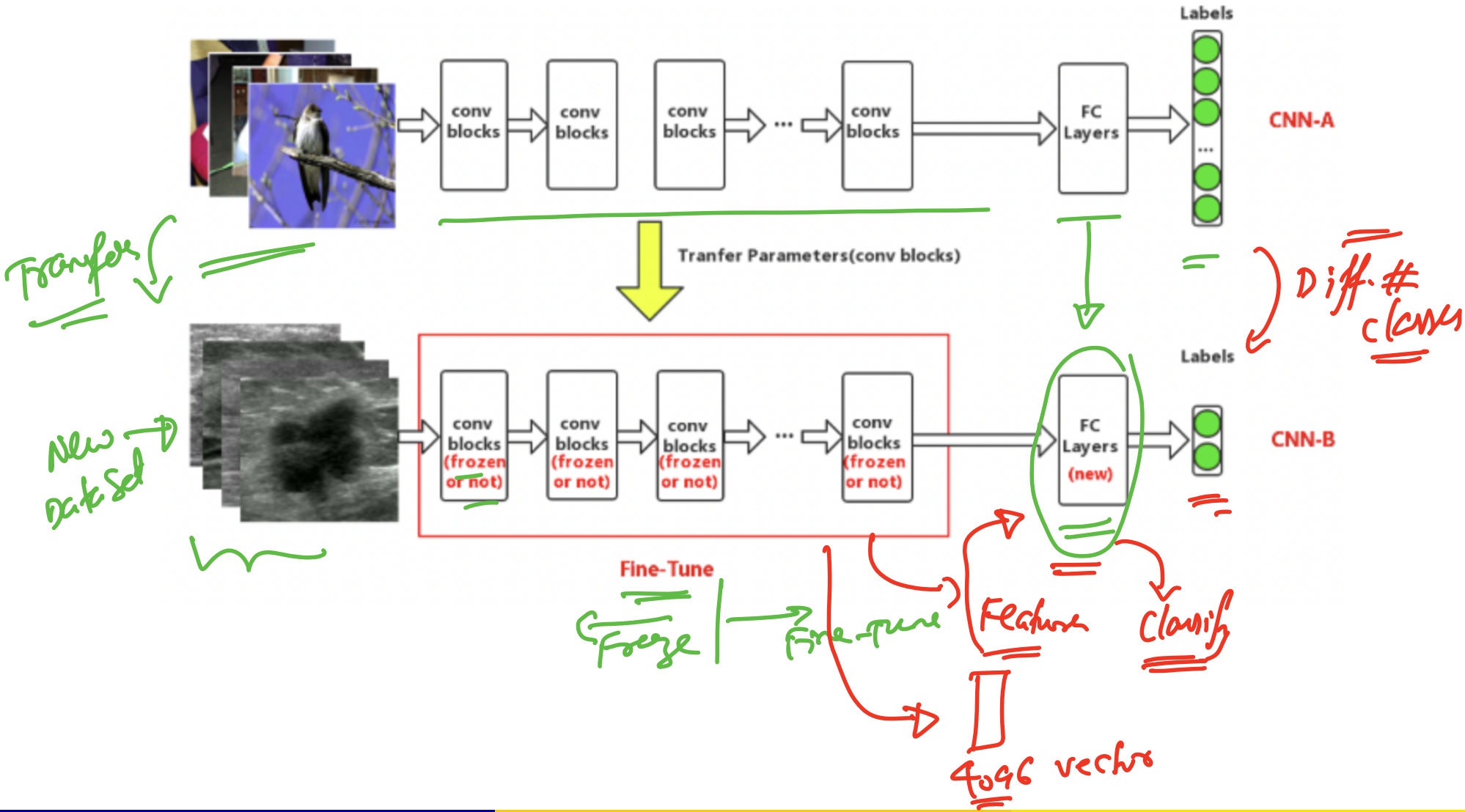
1. **Key Idea:** A model trained on one domain or data set can perhaps transfer to another domain/data set
2. **Key Idea:** With big CNN archs like Inception or ResNet - Training from scratch is cumbersome and wasted resource
3. **Transfer Learning:** Instead use **pre-trained models** and **fine-tune** as needed!
4. **Transfer Learning:** Get a better starting point, saves train time and stand on the shoulder of giants!



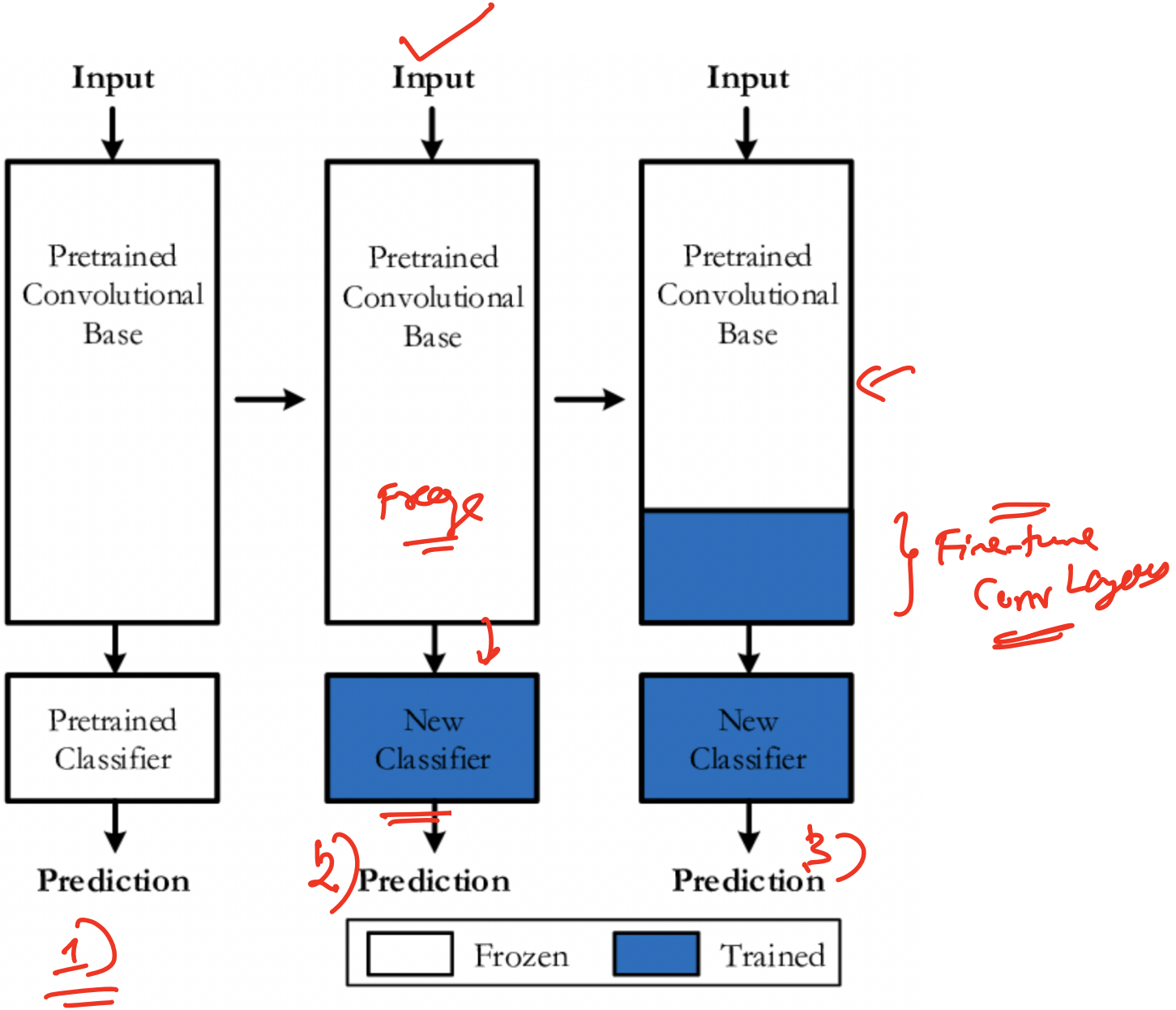
Transfer Learning



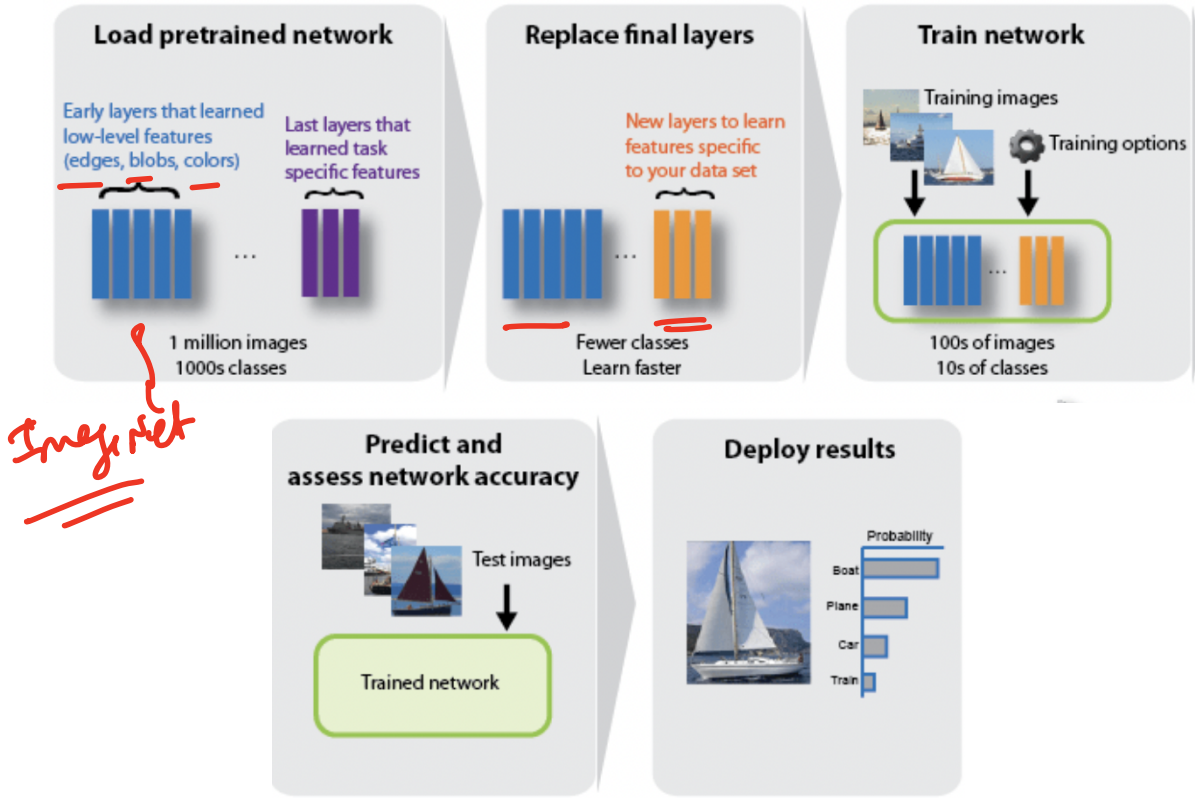
Transfer Learning



Transfer Learning



Transfer Learning



Why Transfer Learning?

1. **Don't Reinvent the wheel!** Broader image features such as edges are useful for most tasks - So why train from scratch? Earlier layers of a trained CNN Net can be a starting point.

Why Transfer Learning?

1. **Don't Reinvent the wheel!** Broader image features such as edges are useful for most tasks - So why train from scratch? Earlier layers of a trained CNN Net can be a starting point.
2. **Time and effort to train:** If a large CNN model took a lot of effort and time to train (e.g. a week to train) - No need to duplicate effort. Reduce time by using CNN codes or features from a pre-trained model.

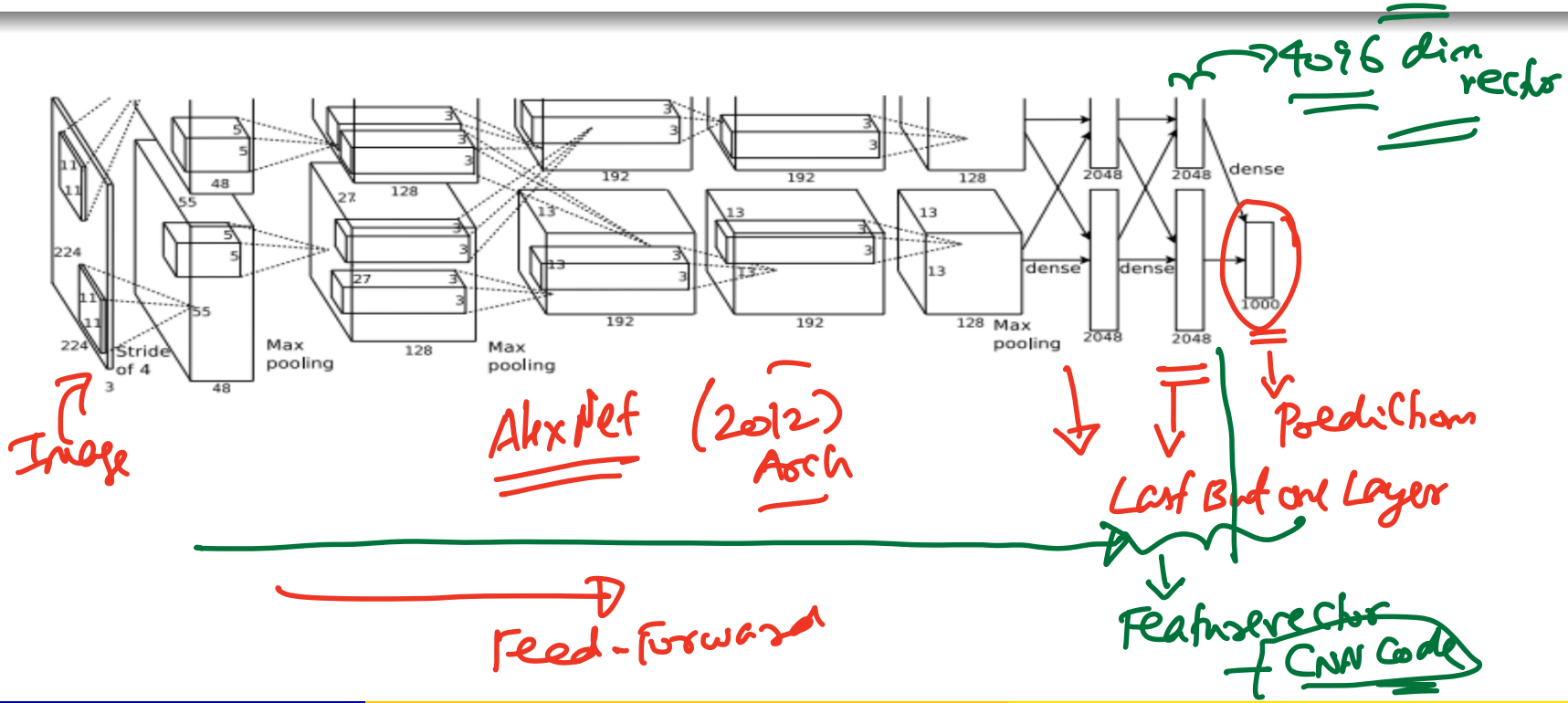
Why Transfer Learning?

1. **Don't Reinvent the wheel!** Broader image features such as edges are useful for most tasks - So why train from scratch? Earlier layers of a trained CNN Net can be a starting point.
2. **Time and effort to train:** If a large CNN model took a lot of effort and time to train (e.g. a week to train) - No need to duplicate effort. Reduce time by using CNN codes or features from a pre-trained model.
3. **Context:** Your data set may not permit training of a full CNN model. Why?

CNN Code

CNN Code

For any given image, and a given arch (net), the CNN code for that image is the activations at the last but one layer of the image obtained from the arch!

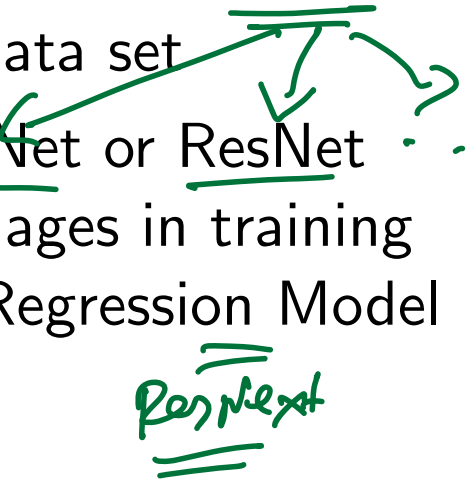


Transfer Learning Strategies for CNNs

1. **CNN Code + Linear Classifier** Use last but one layer of a CNN Net as a starting point for a linear classifier on the new data set

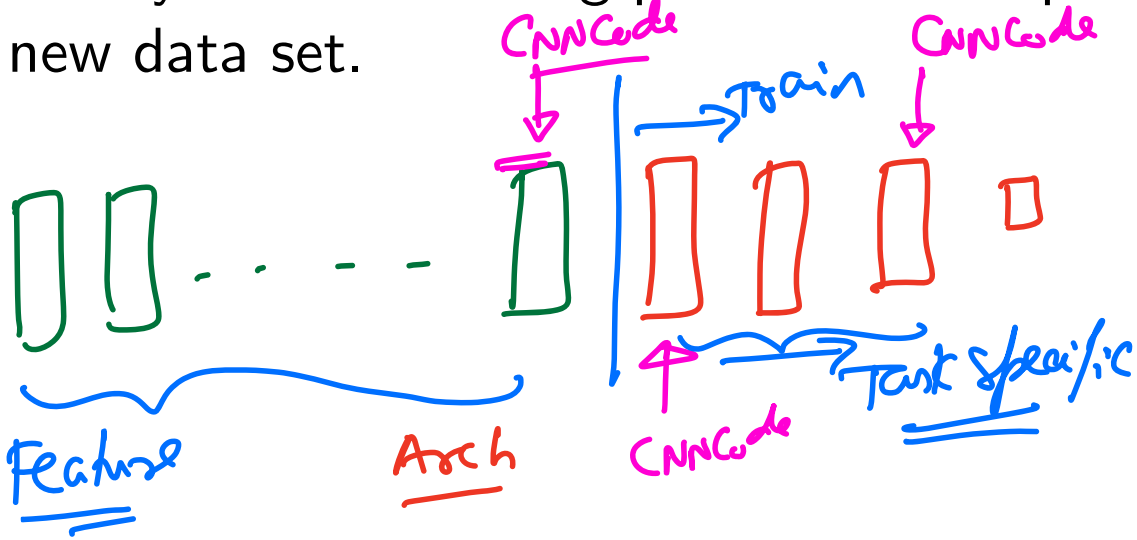
Transfer Learning Strategies for CNNs

1. **CNN Code + Linear Classifier** Use last but one layer of a CNN Net as a starting point for a linear classifier on the new data set
2. **CNN Code + Linear Classifier** Example: Let AlexNet or ResNet give you CNN codes (last but one activations) for images in training data. Use the CNN codes as *features* for a Logistic Regression Model



Transfer Learning Strategies for CNNs

1. **CNN Code + Linear Classifier** Use last but one layer of a CNN Net as a starting point for a linear classifier on the new data set
2. **CNN Code + Linear Classifier** Example: Let AlexNet or ResNet give you CNN codes (last but one activations) for images in training data. Use the CNN codes as *features* for a Logistic Regression Model
3. **Freeze + Fine-Tune** Example: Freeze all but the FC layers of a net. Fine-tune the FC layers with starting point from the pre-trained model on the new data set.



Transfer Learning Strategies for CNNs

1. **CNN Code + Linear Classifier** Use last but one layer of a CNN Net as a starting point for a linear classifier on the new data set
2. **CNN Code + Linear Classifier** Example: Let AlexNet or ResNet give you CNN codes (last but one activations) for images in training data. Use the CNN codes as *features* for a Logistic Regression Model
3. **Freeze + Fine-Tune** Example: Freeze all but the FC layers of a net. Fine-tune the FC layers with starting point from the pre-trained model on the new data set.
4. **Freeze + Fine-Tune + New Layers** Freeze some of the CNN Net layers, fine-tune the rest of the layers and add new layers as well to classify.

When to pick what Strategy?

High-level

How does the new data set compare to the pre-trained model data set (e.g. image net) in terms of size and similarity?

1. **Smaller data set and similar to pre-train data set:** Use CNN Codes + Linear Classifier. Why?

When to pick what Strategy?

High-level

How does the new data set compare to the pre-trained model data set (e.g. image net) in terms of size and similarity?

1. **Smaller data set and similar to pre-train data set:** Use CNN Codes + Linear Classifier. Why?
2. **Smaller data set but different from pre-train data set:** Use CNN Codes from a couple of layers before. Why?

When to pick what Strategy?

High-level

How does the new data set compare to the pre-trained model data set (e.g. image net) in terms of size and similarity?

1. **Smaller data set and similar to pre-train data set:** Use CNN Codes + Linear Classifier. Why?
2. **Smaller data set but different from pre-train data set:** Use CNN Codes from a couple of layers before. Why?
3. **Larger data set:** Freeze + Fine-Tune

ICE #1

Let's say you want to use ResNet model arch for the Mini-Project. For the Fashion MNIST data set, which strategy would probably work well if you wanted to do transfer learning?

1. Use CNN Codes from ResNet + Linear Classifier
2. Same as a) but use CNN Codes from a few layers before the last layer
3. Freeze all CNN layers but fine tune/train the FC layers on the Fashion MNIST data set
4. Use CNN Codes from ResNet + a 3 layer feed-forward NN

Mini-Project pointers

1. If using CNN codes + Linear Classifier: Can train within pytorch - Freeze + retrain last layer
2. If using CNN codes + Linear Classifier: Can also use a different library e.g. Logistic Regression in Sci-Kit to take the CNN codes as input and learn weights on new data set!
3. Any other models?

Transfer Learning PyTorch Tutorial

PyTorch Transfer Learning
Tutorial on Transfer Learning

ICE #2

Transfer Learning

When would Transfer Learning not see significant gains over regular learning/training on an image data set?

1. Never
2. When data set size is small but distribution is different from any of the data sets the pre-trained models were trained on
3. When data set size is small but distribution is similar to any of the data sets the pre-trained models were trained on
4. When data set size is large but distribution is different from any of the data sets the pre-trained models were trained on

Next Lecture

1. Object Detection
2. Methods for Object Detection
3. Image and instance segmentation