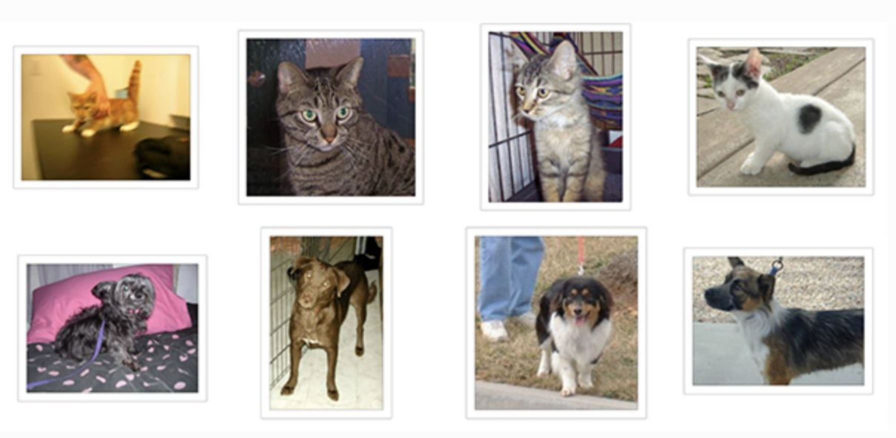
Intro to Pre-trained Models

Charlie Snell, Arnav Gudibande, Divi Schmidt

Motivating Pre-trained Models

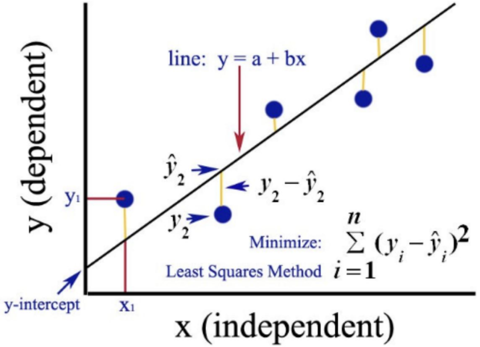
- Binary Image Classification
- How are images represented?
  - Pixel values in the range of [0, 255]
  - 3 channels (RGB)
- What are some different techniques to classify these images?
  - OLS/Ridge
  - Logistic Regression
  - SVMs/ConvNets/Decision Trees



Source: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

A First Approach

- Using Ordinary Least Squares (OLS)
  - Classes can have labels +1 / -1
  - Solve for a closed-form solution
  - Set the classification threshold at 0
- How do create the matrix X?
  - Map every pixel to an element in a vector
  - Stack these vectors to create the design matrix
- Drawbacks
  - Computational Complexity
  - The use of MSE for classification



1D OLS visualization

$$w = (X^T X)^{-1} X^T Y$$

OLS Closed form solution

Source: https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

A Second Approach

- Using Logistic Regression
    - Logistic Loss is better-suited for classification
    - SGD to minimize logistic loss
- How do create the matrix X?
    - Create our own edge detectors to extract features
    - Stack the output of edge detectors together
- Drawbacks
    - No closed form solution
    - How do we determine good features?

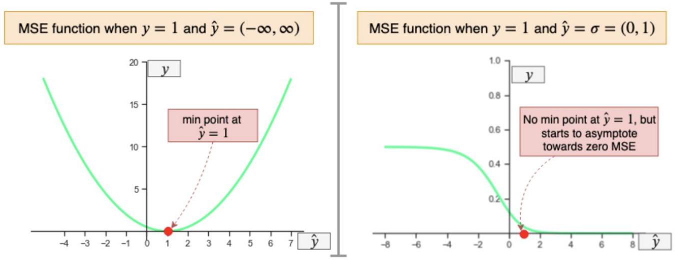


Fig 3. Non-convexity of MSE when output is from a Sigmoid/Logistic function

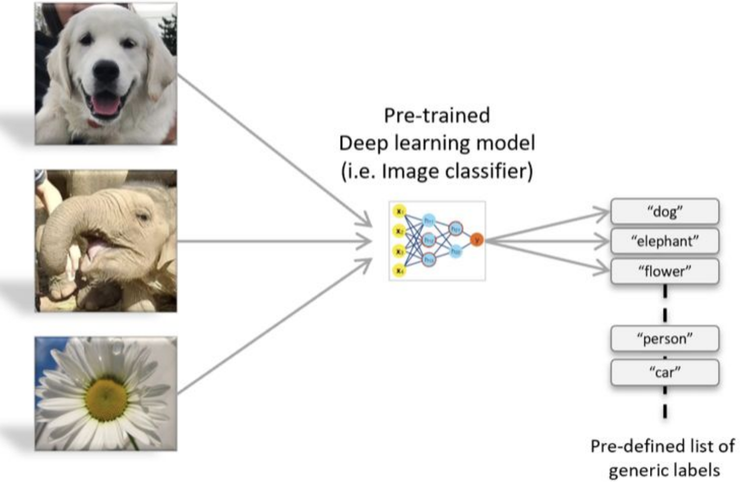
$$\underbrace{\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix}}_{6\times 6} * \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}}_{3\times 3} = \underbrace{\begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}}_{4\times 4}$$

Figure 1: A vertical edge detector convolving an image. Source: Andrew Ng Coursera

Source: https://towardsdatascience.com/why-using-mean-squared-error-mse-cost-function-for-binary-classification-is-a-bad-idea-933089e90df7

A Third Approach

- Using Pre-trained Models
  - Pre-trained models are trained on a large dataset that solve a task similar to the one you are trying to solve
  - In doing so, these models learn expressive and comprehensive features
  - These features can be effectively transferred to our task of binary classification
- Limitations
  - Choosing the pre-trained model
  - Not applicable to every task



Source: https://devblogs.microsoft.com/cesardelatorre/training-image-classification-recognition-models-based-on-deep-learning-transfer-learning-with-ml-net/

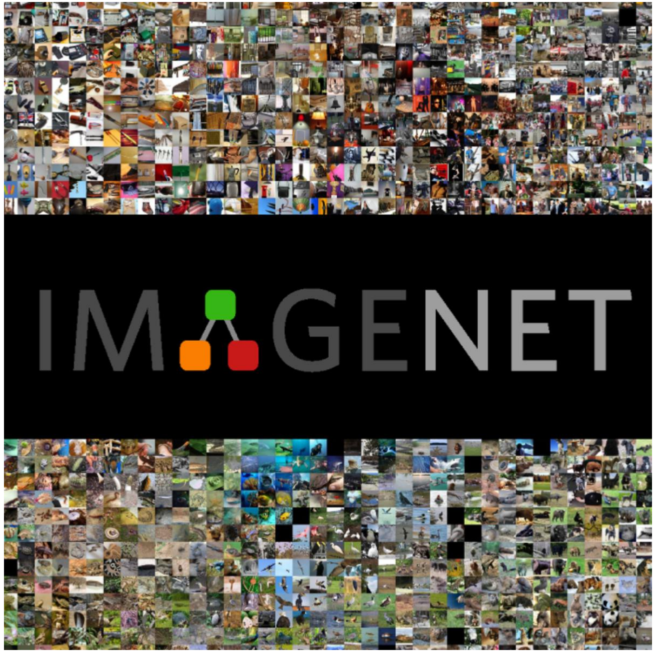
Overview of Common Pretrained models

A Brief History

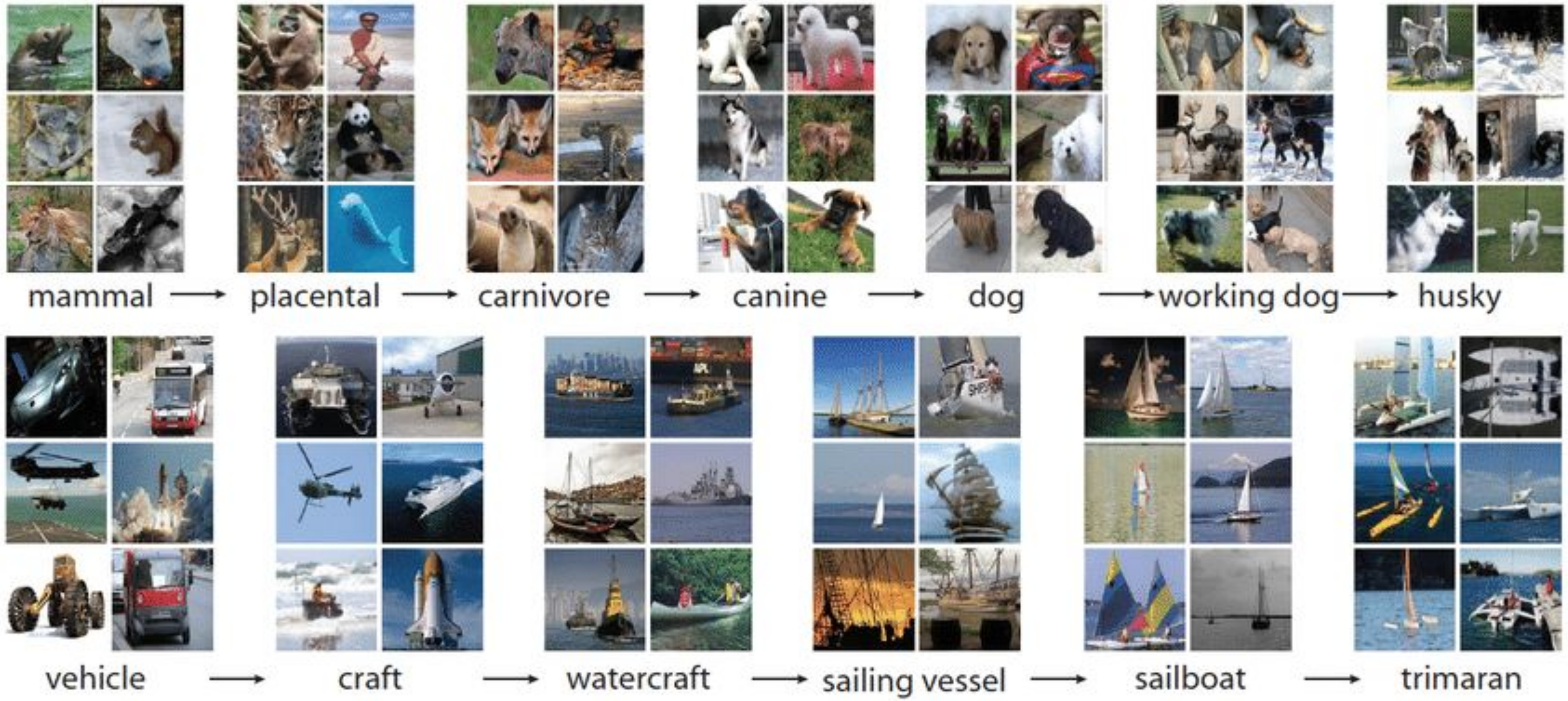
- Pre-deep learning, SOTA CV was done with handcrafted filters
  - These filters were designed to extract features like edges from images
  - Then train a classifier with these extracted features
- Modern deep learning algorithms learn these kinds of filters on their own
  - Extract edges, textures, frequency changes
  - Multiple layers allows the model to build on these features and learn higher level features from the images

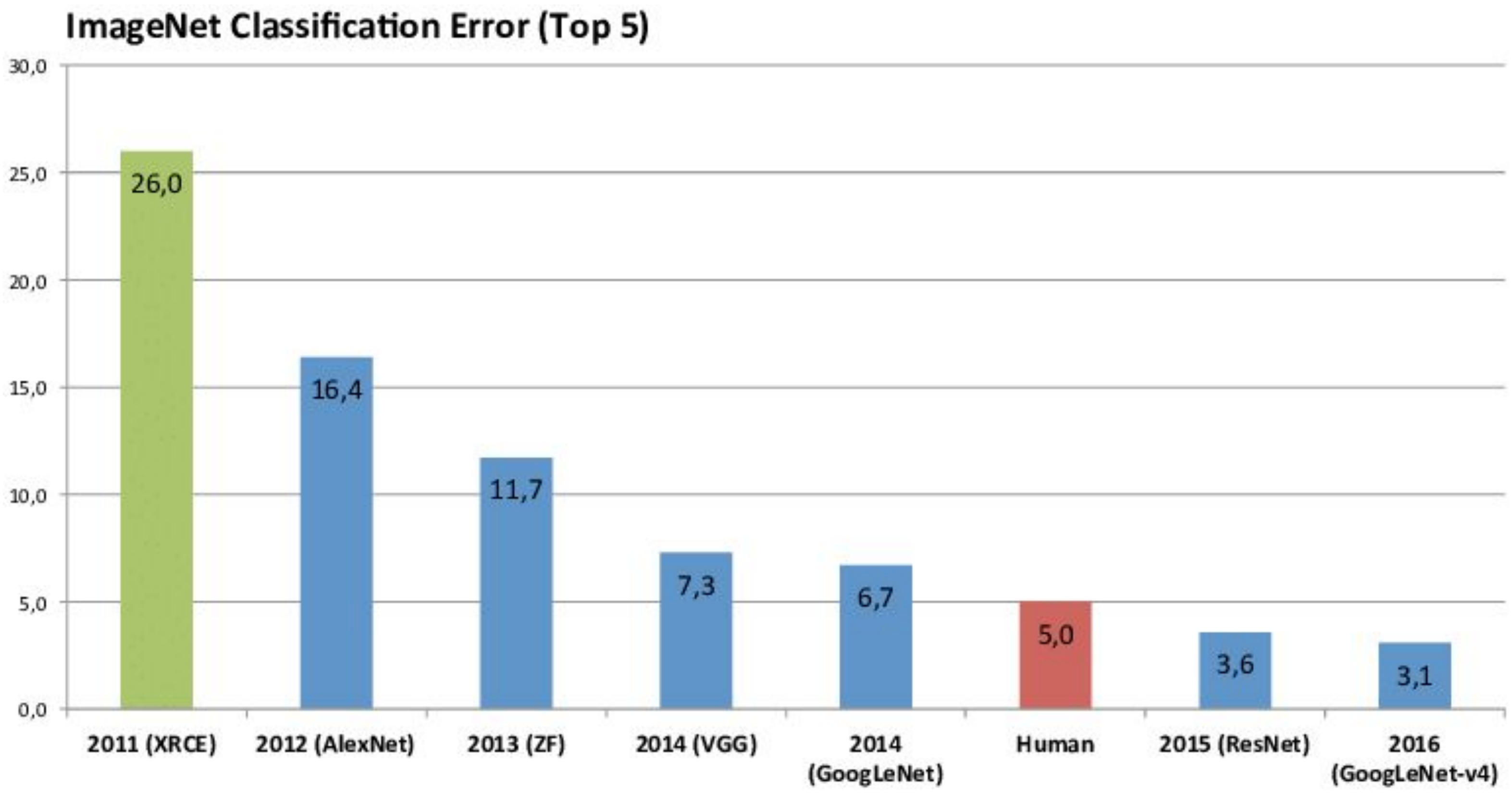
The Imagenet Dataset

- ImageNet classification dataset
  - A diverse dataset of over 14 million color images
  - 12,841 categories
- ImageNet classification challenge
  - Subset of ImageNet
  - 1000 classes
  - 1.2 million training images
  - Annual contest to get the highest top-1 classification accuracy on this dataset
  - Most SOTA models are pre-trained on some version of this dataset



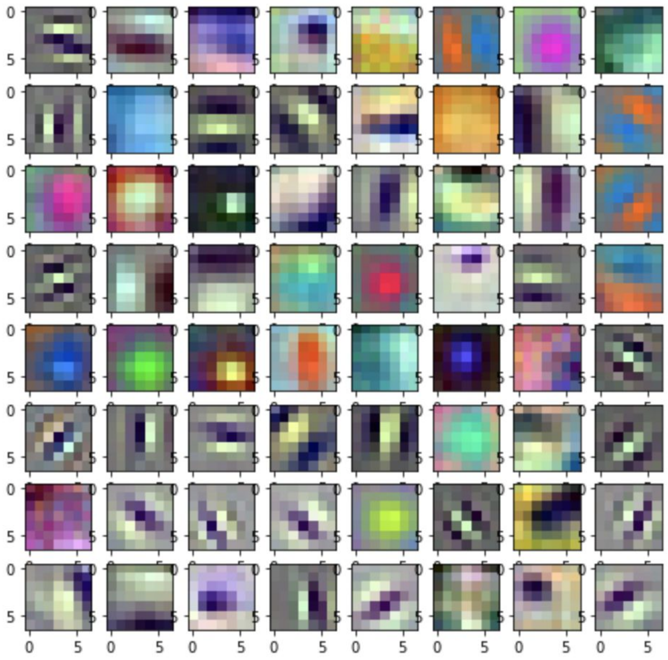
Some examples from ImageNet:




ImageNet Classification Error (Top 5)
30,0
25,0
20,0
15,0
10,0
5,0
0,0
26,0
16,4
11,7
7,3
6,7
5,0
3,6
3,1
2011 (XRCE)
2012 (AlexNet)
2013 (ZF)
2014 (VGG)
2014 (GoogLeNet)
Human
2015 (ResNet)
2016 (GoogLeNet-v4)

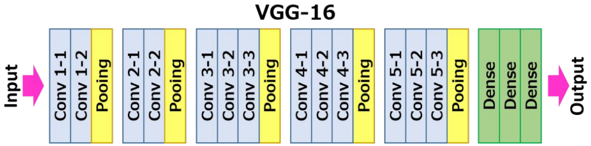
Learned Features

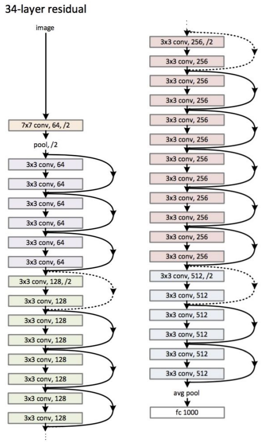
- Models trained on ImageNet learn fairly general image features
  - Can use these features to train models on other CV tasks like object detection
- The first layer learned filters demonstrate the generality of the learned features
  - Learn things like edge and texture detectors on their own
  - Later layer features and only build on these

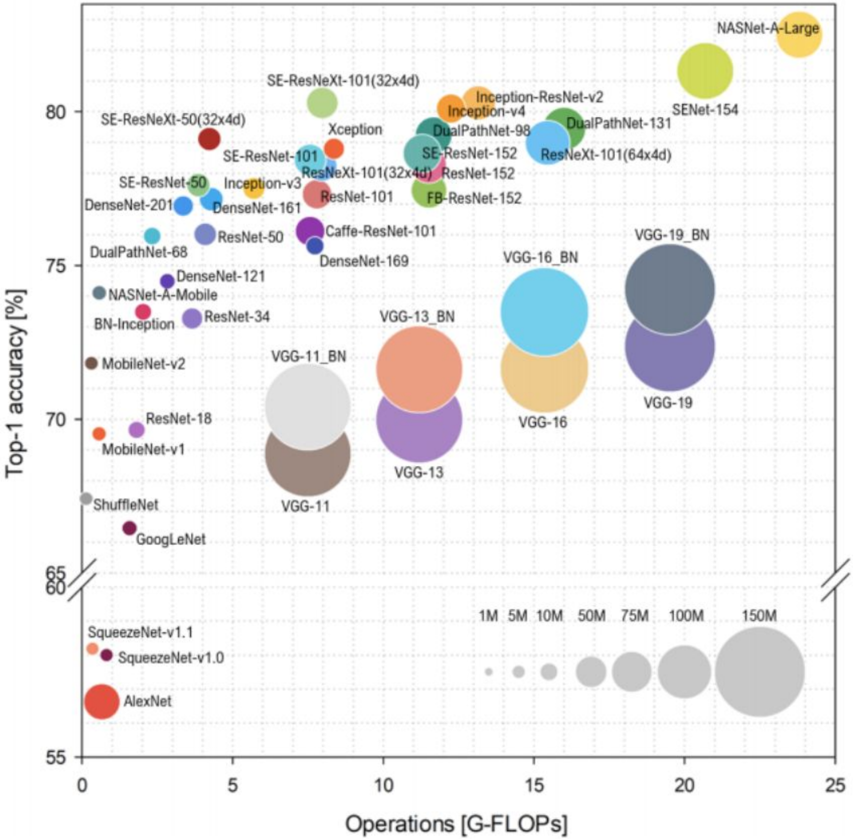


SOTA Pretrained Models

- Many different pretrained models
  - Some are more accurate (ResNetXt)
  - Others run faster or used fewer parameters (MobileNet)







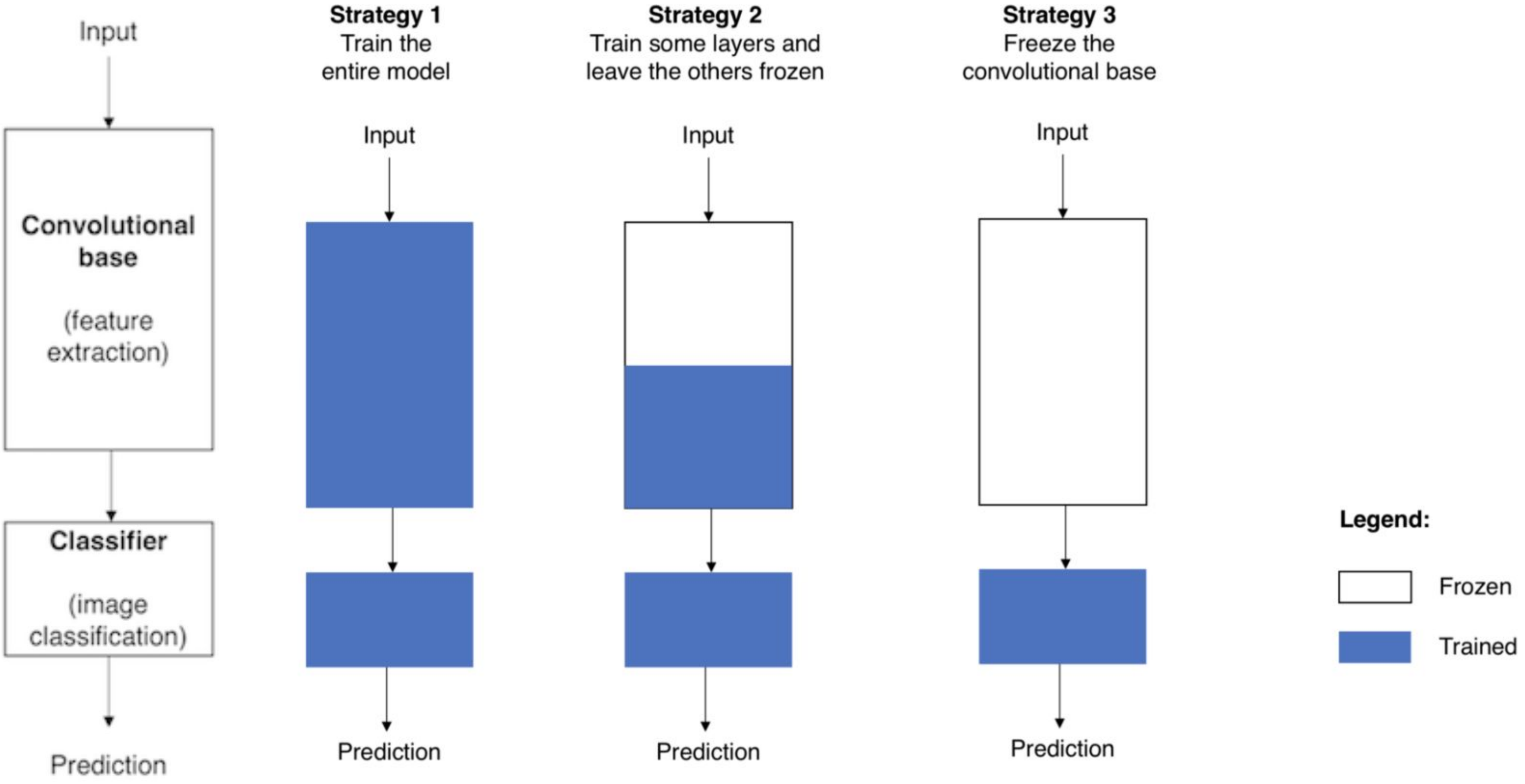
Using pretrained models

Transfer Learning

Transfer learning is a technique where a model trained on one task is "repurposed" for a second task.

Pre-trained models are used often to perform transfer learning, as these are both easy to obtain and have high quality features

Transfer Learning Strategies



https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

Train the Entire Model

This is analogously called fine-tuning all layers of the network.

We can treat our pre-trained model as a sort of “intelligent weight initialization” since we start with pre-trained weights instead of random weights, and then train our network.

Train Some Layers

This strategy is also referred to as fine-tuning, but differs from the previous approach in that we fix some layers in our network.

This means that when training our network, we use the pre-trained model initialization, and do not update the weights of our fixed layers.

- We can fix almost all layers to almost none of the layers
- How many layers we fix is problem dependent

Freeze the Convolutional Base

This is using a pre-trained network as a **fixed feature extractor**

This differs from the previous two strategies in that we do not update any of the layers in our network.

- First, we take the output from a specific layer in our network (this could be the last layer or one of the first)
- Then, we use these outputs (also known as features) to train a linear classifier
    - This could be a fully connected layer, SVM, etc.

When do we use each of these strategies?

Gaining Intuition on Pre-trained Models

Small vs Large Datasets

Small Datasets

- May cause overfitting due to lack of data
- Freeze more layers to prevent overfitting

Large Datasets

- Not worried about overfitting
- Can afford to finetune more layers

Gaining Intuition on Pre-trained Models

Similar vs Dissimilar Datasets

- When data is similar to data used for original pre-training
- Example of dissimilar data would be ImageNet vs microscope images

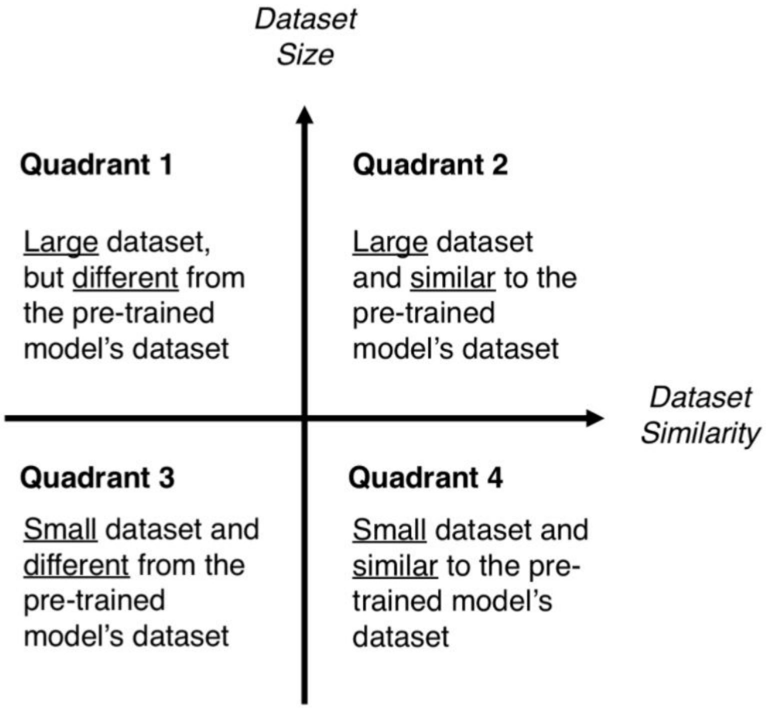
Similar Datasets

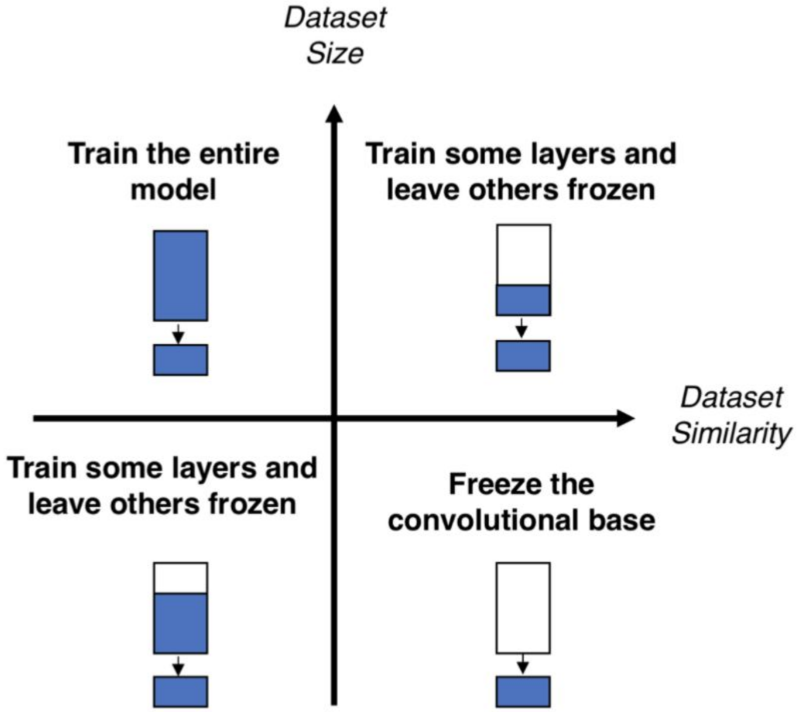
- Higher layers contain relevant information to the new task
- Want to maintain useful features from pre-trained model

Dissimilar Datasets

- Higher layers do not contain relevant information
- Want to re-train or discard higher layers of network

Summary





https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751