Improving Performance of the Convolutional Neural Network

Pratheerth Padman
Freelance Data Scientist

Module Overview

Why and when do we need better performance?

Procuring additional training data with image augmentation

Hyperparameter tuning

What are overfitting and underfitting?

Demo: Improving model performance through image augmentation and hyperparameter tuning

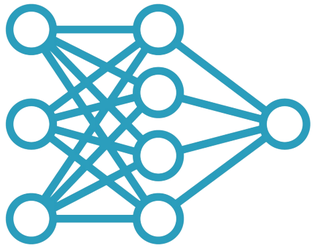
What is transfer learning? When should you apply transfer learning?

Demo: Improving performance metrics through transfer learning

Better Performance – When and How?

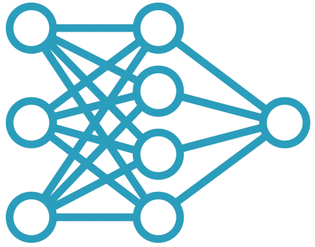
Better Performance – When?

Simple problem



[Simple network]

Complex problem

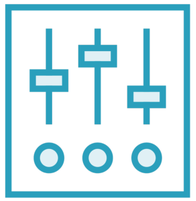


[Simple network]

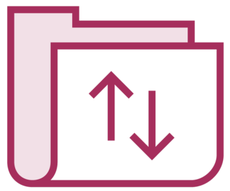
Better Performance – When?

More complex neural network

Procure additional training data



Hyperparameter tuning



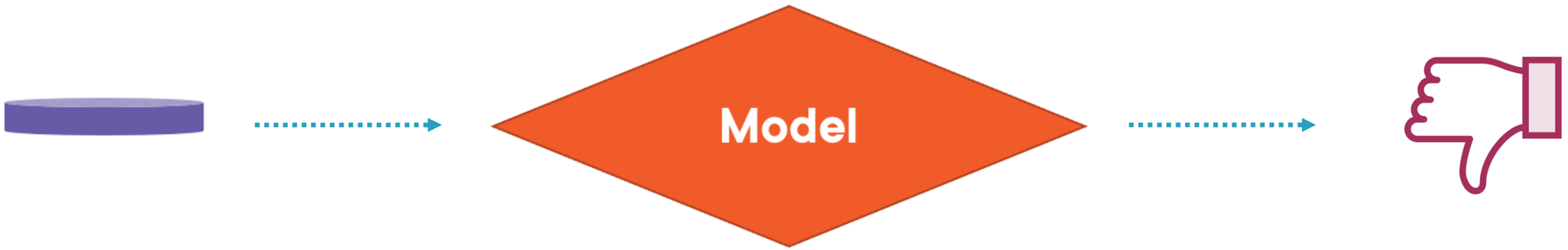
Transfer learning

Procuring Additional Training Data – Image Augmentation

Image Augmentation

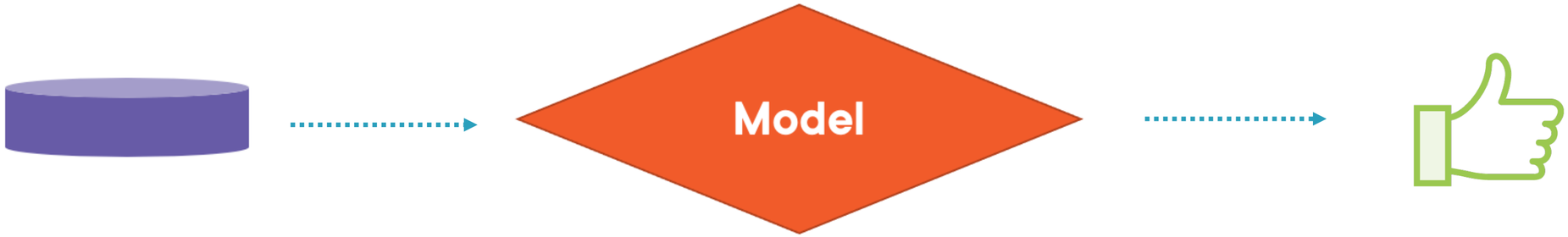
Image Augmentation is a technique of altering existing data to create some more data for the model training process through different processing methods such as rotations, shifts, flips, etc.

Need for Image Augmentation



Our data (small) **Model** **Performance**

Need for Image Augmentation



Our data (augmented) **Model** **Performance**

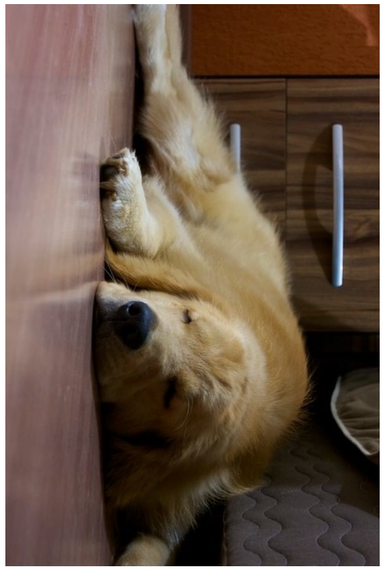
Image Augmentation - Flip

Original image

Flipped image

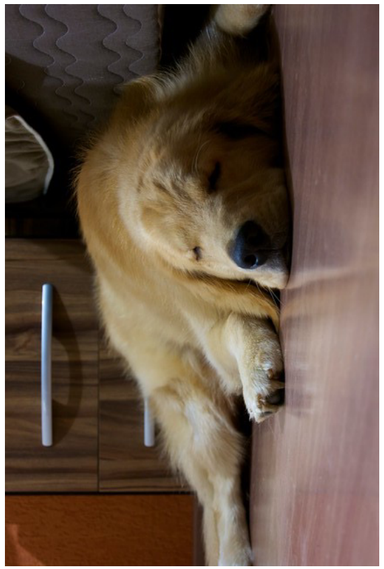
Image Augmentation - Rotation

Original image



Rotated 90 degrees

Rotated 180 degrees



Rotated 270 degrees

Image Augmentation - Scaling

Original image

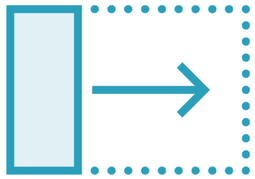
Scaled image

Image Augmentation - Translation

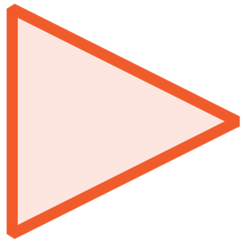
Original image

Translated image

How Does Image Augmentation Help?



Increases data, helping network train properly



Creates invariance which helps models generalize well

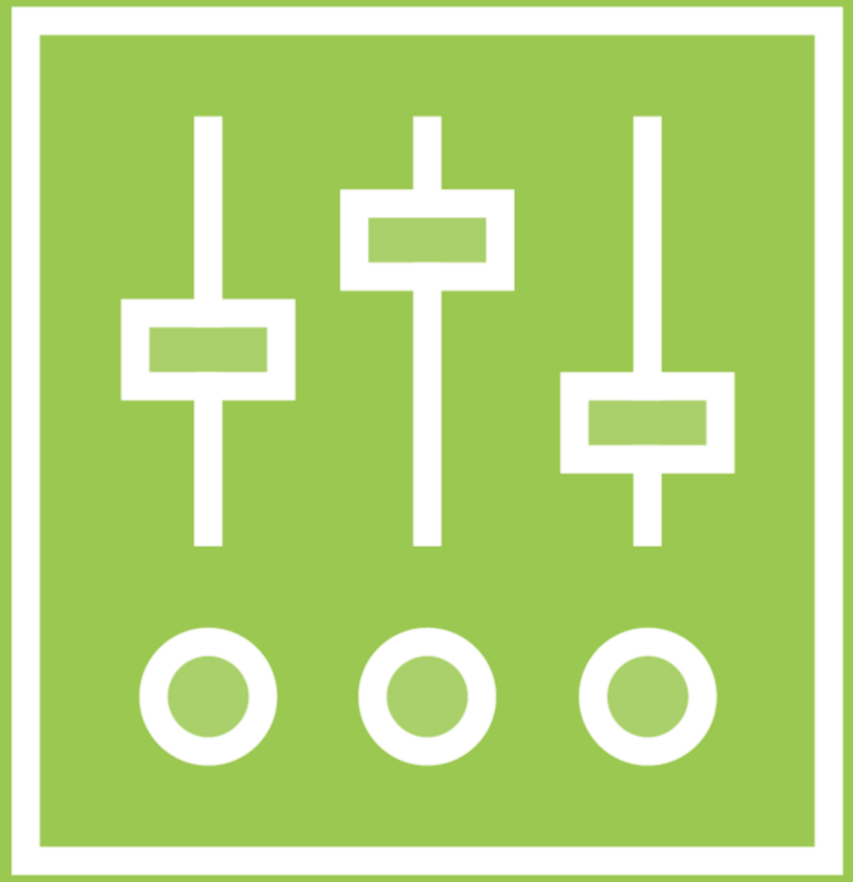
Helps even when presented with lots of data

When Should You Perform Image Augmentation?

Offline augmentation – smaller datasets

Online augmentation – larger datasets

Hyperparameter Tuning



Hyperparameters

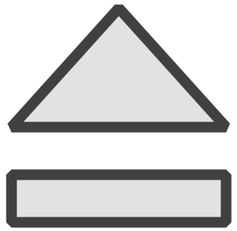
Hyperparameters are settings or “knobs” that can be tuned to control the behavior and performance of a machine learning model.

Types of Hyperparameters

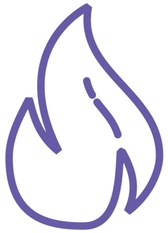
Learning rate

Epochs

Optimizers



Dropout rate



Activation function

Number of layers and units

Methods to Tune Hyperparameters

Trial and error

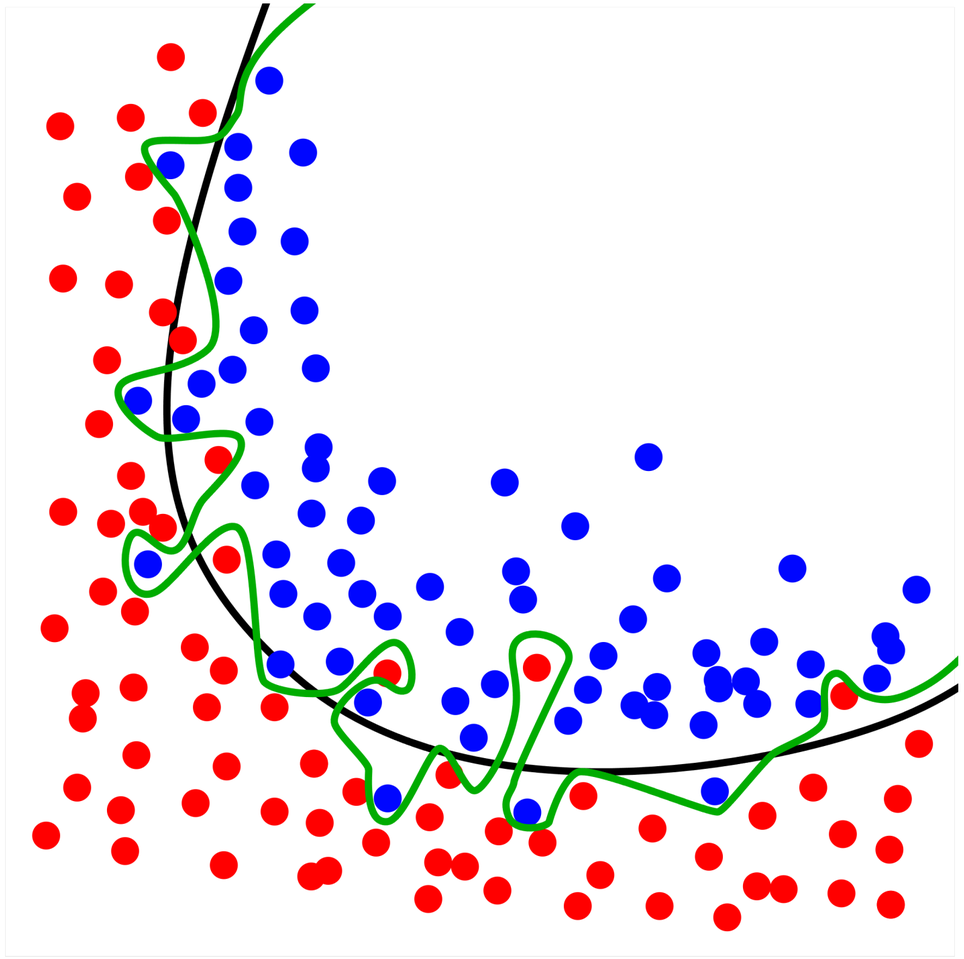
Gridsearch

Randomsearch

3rd party tools

Overfitting and Underfitting

Overfitting



Chabacano, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons

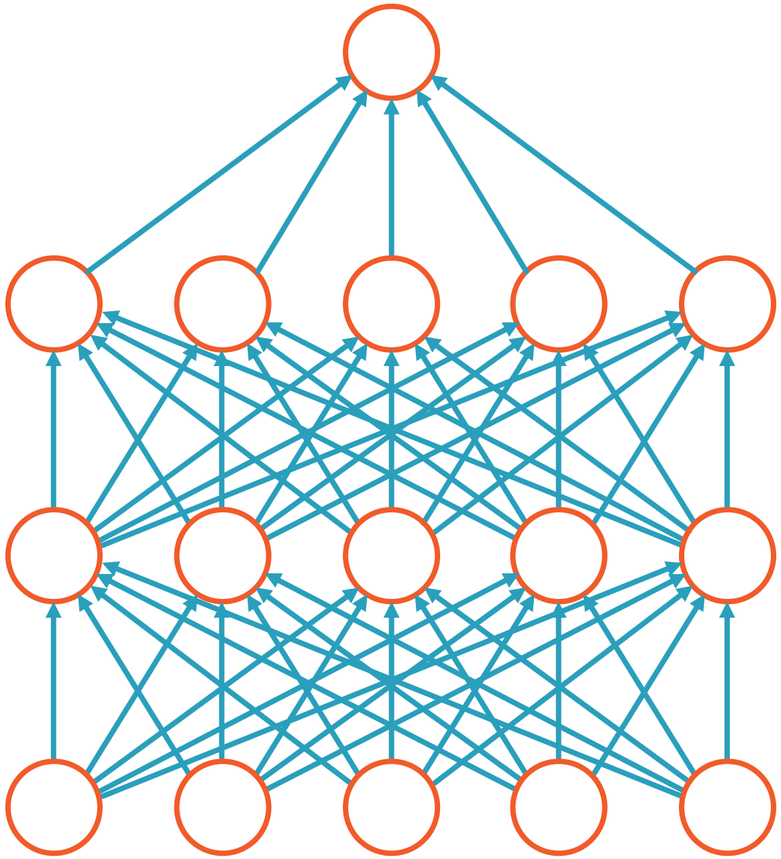
How to Prevent Overfitting

Train with more data

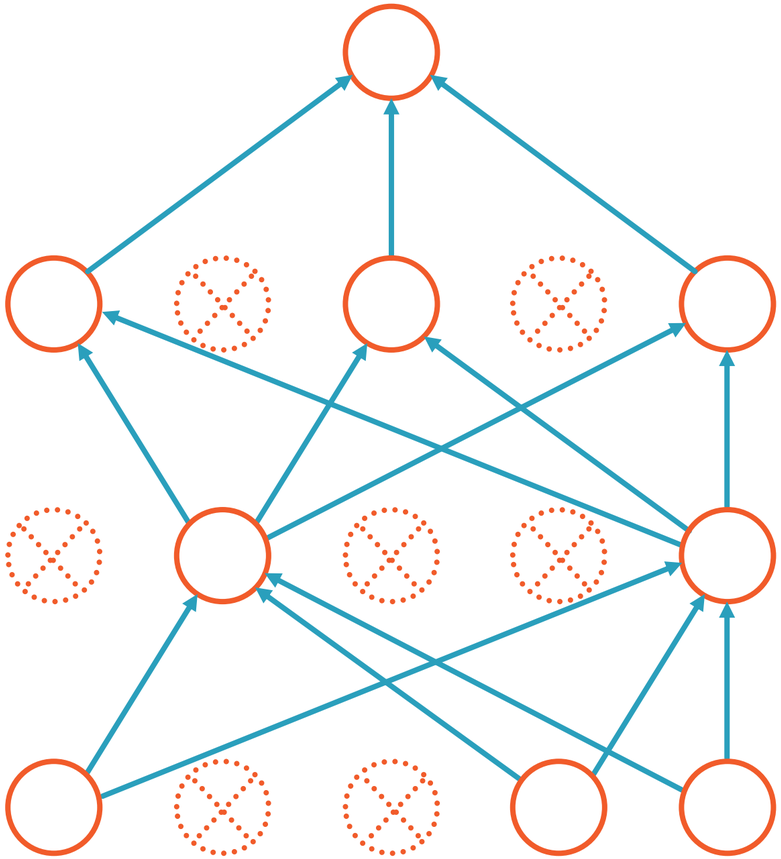
Reduce network complexity

Dropout

Dropout

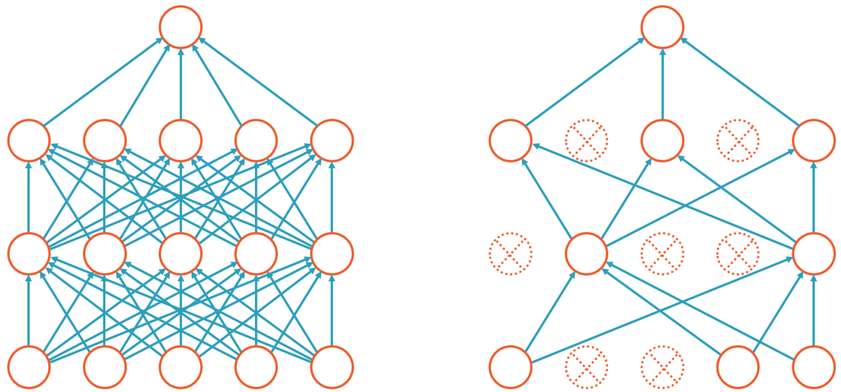


(a) Standard neural net



(b) After applying dropout

How Does Dropout Help?

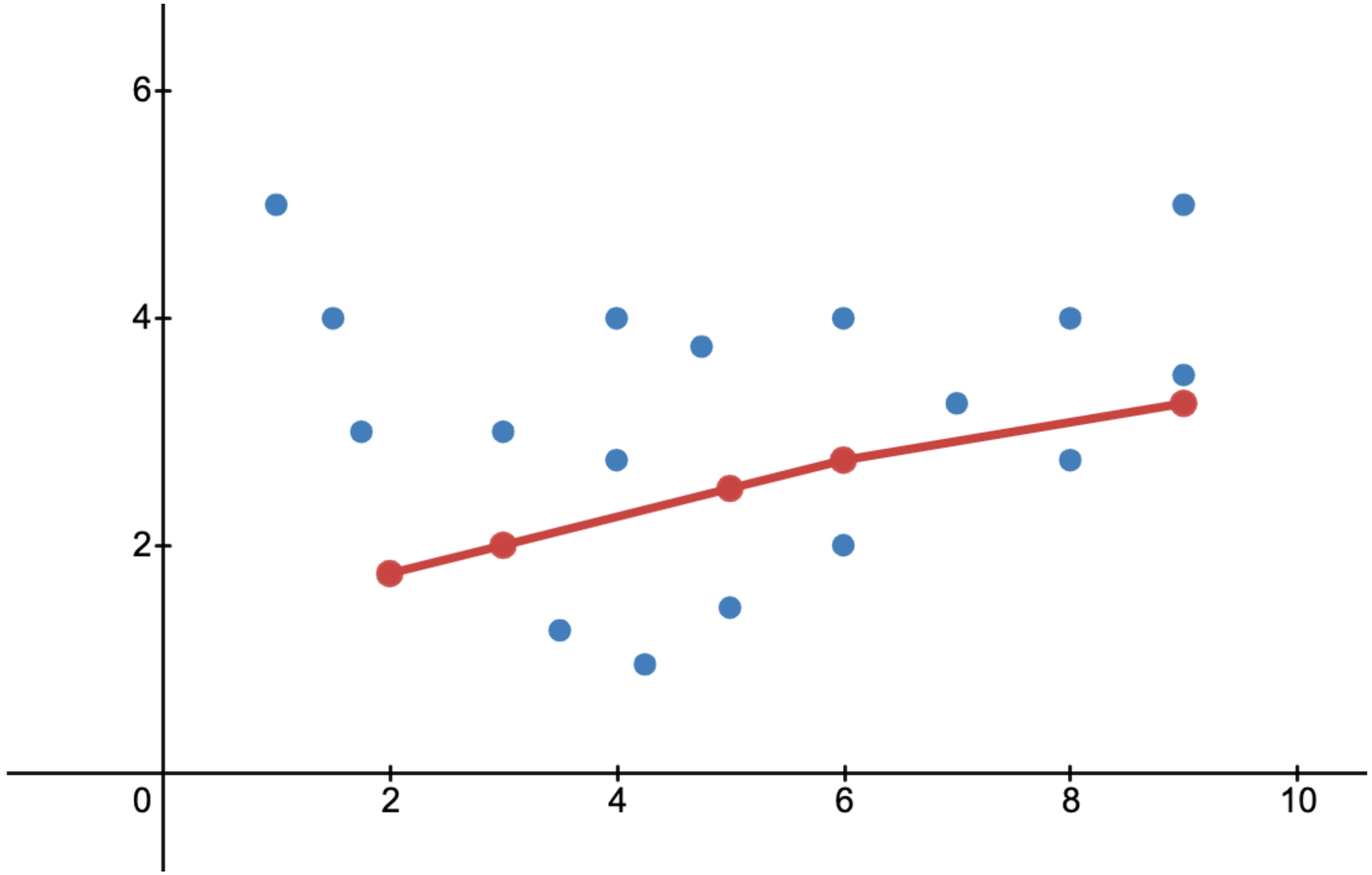


Reduces over reliance on few inputs

Layer learns to effectively use all of its inputs, improving generalization

Simplifies network

Underfitting



AAStein, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons

Demo

Improving model performance with image augmentation and hyperparameter tuning

Transfer Learning

Transfer Learning

Transfer learning is the process of leveraging knowledge gained from building a model for a particular task, and then reusing that knowledge as a starting point for a different task

Transfer Learning Intuition

Transfer Learning Diagram

[Model performs very well]

Transfer Learning

We don’t use the entire pre-trained model

Part of it is used as a starting point

Works because initial layers of pre-trained model have already learned features used commonly for most CV problems

Popular Transfer Learning Models

VGG 16

(Visual Geometry Group, Oxford, 2014)

VGG 19

(Visual Geometry Group, Oxford, 2014)

ResNet

(Microsoft, 2015)

Inception V3

(Google, 2015)

Xception

(Google, 2016)

Transfer Learning – When and How?

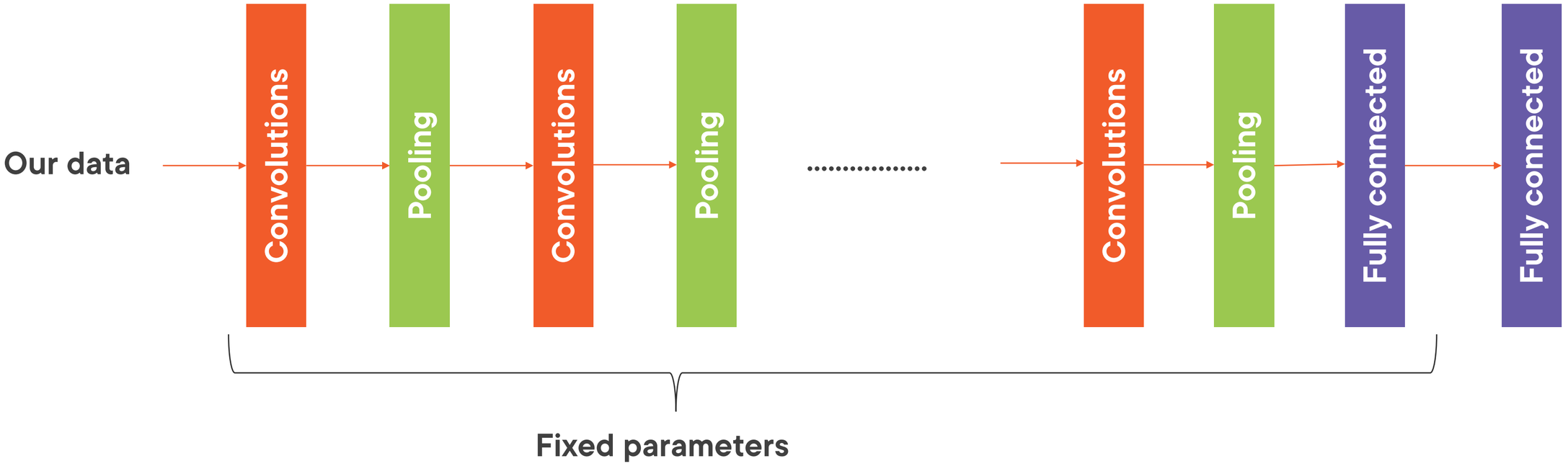
Types of Transfer Learning – Decision Factors

Size of the new dataset

Similarity of the new dataset to the original dataset

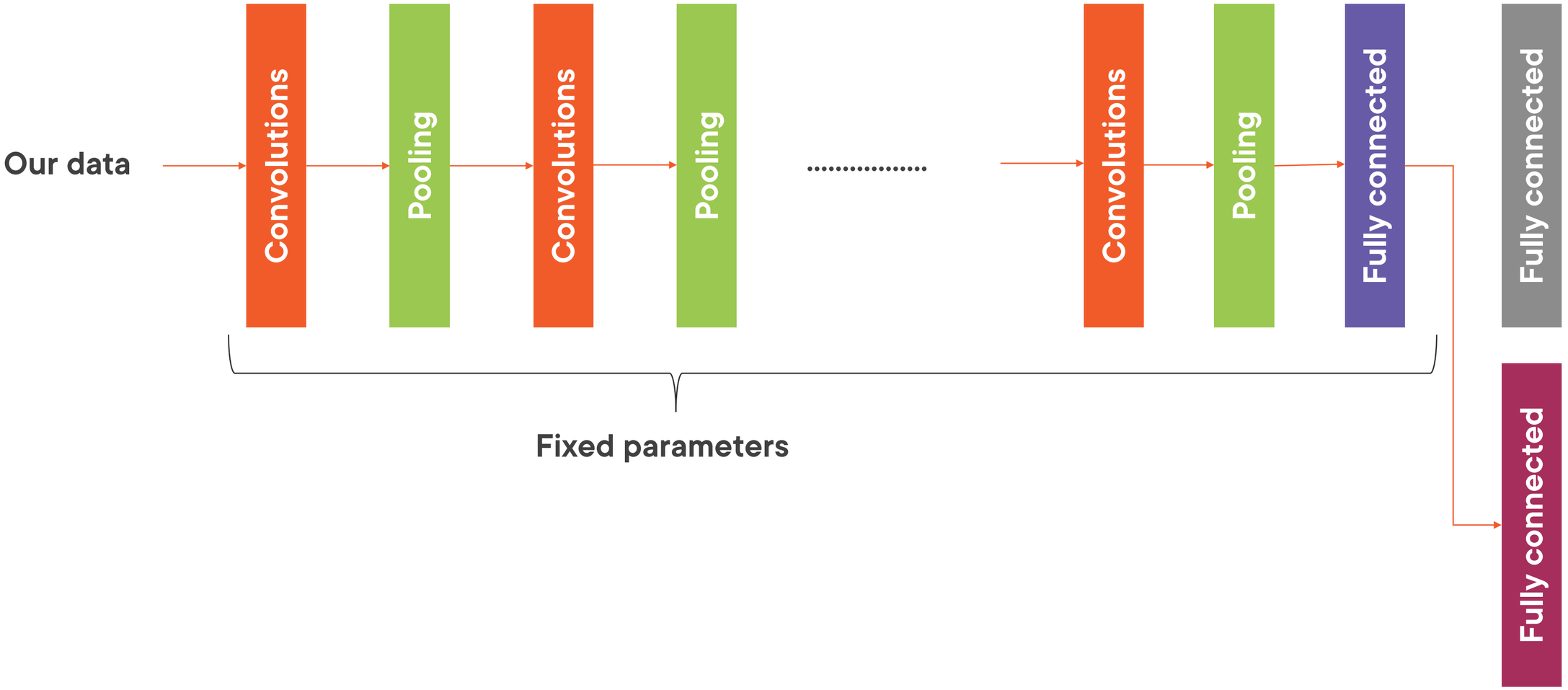
Case 1

Our dataset is small, but very similar to original dataset



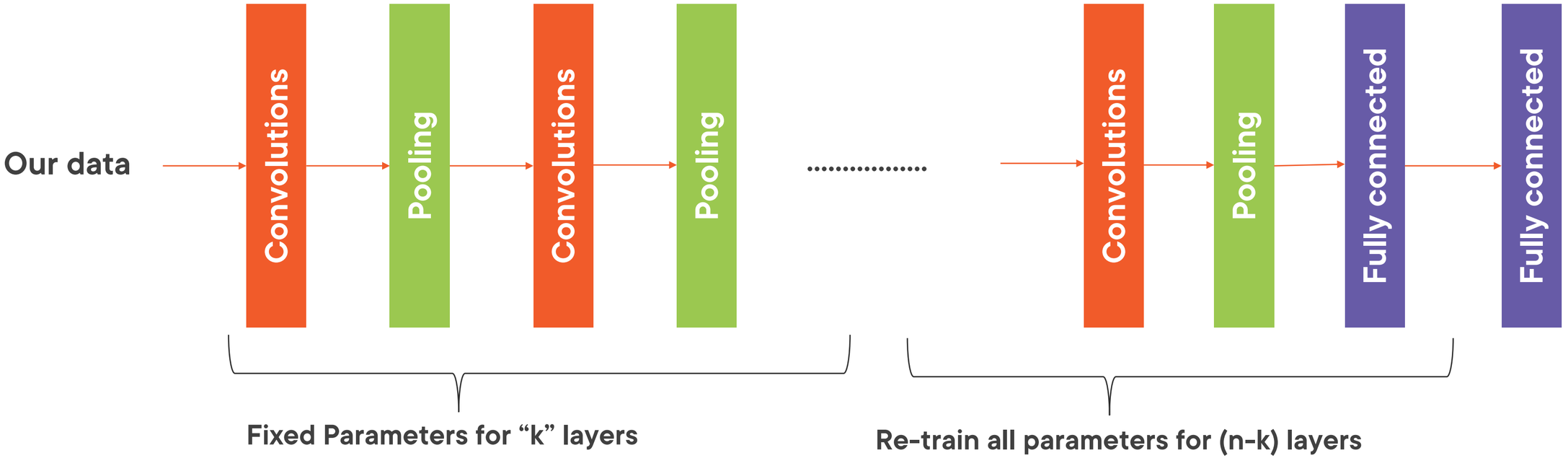
Case 1

Our dataset is small, but very similar to original dataset



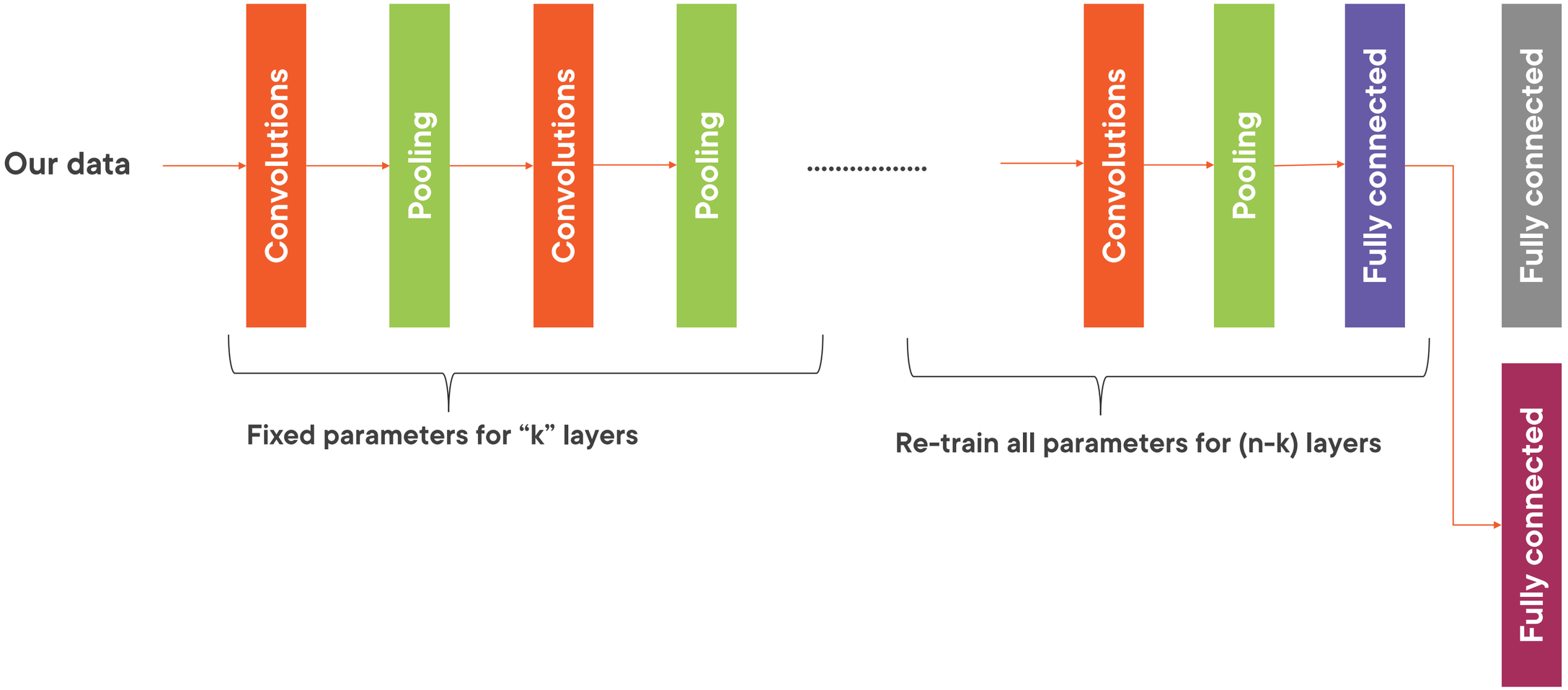
Case 2

Our dataset is small, also not similar to the original dataset



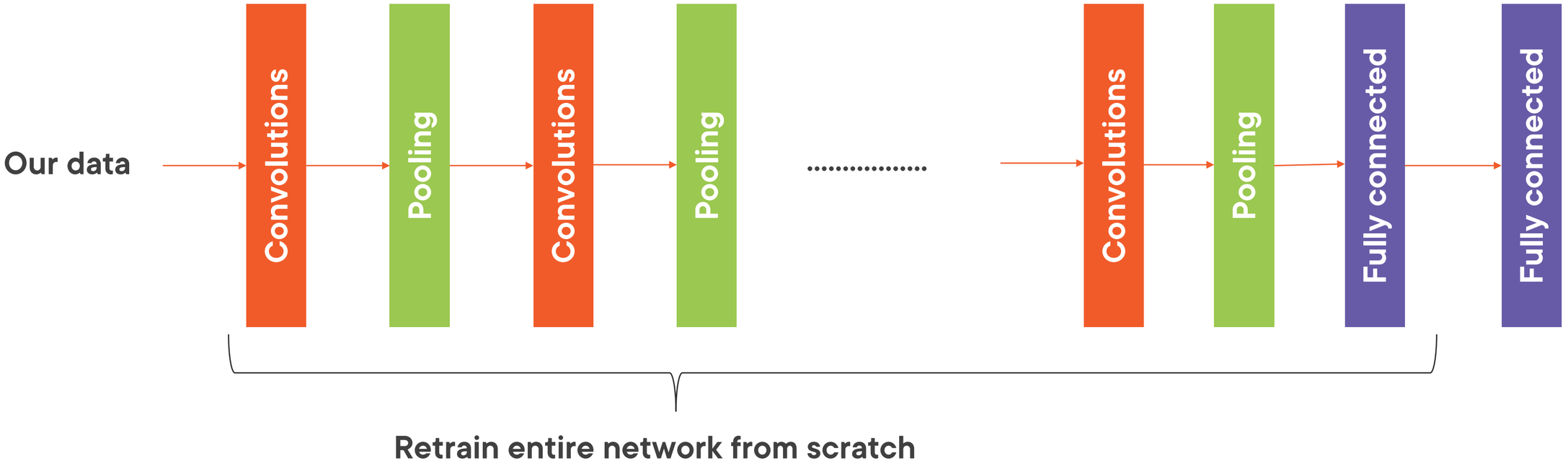
Case 2

Our dataset is small, also not similar to the original dataset



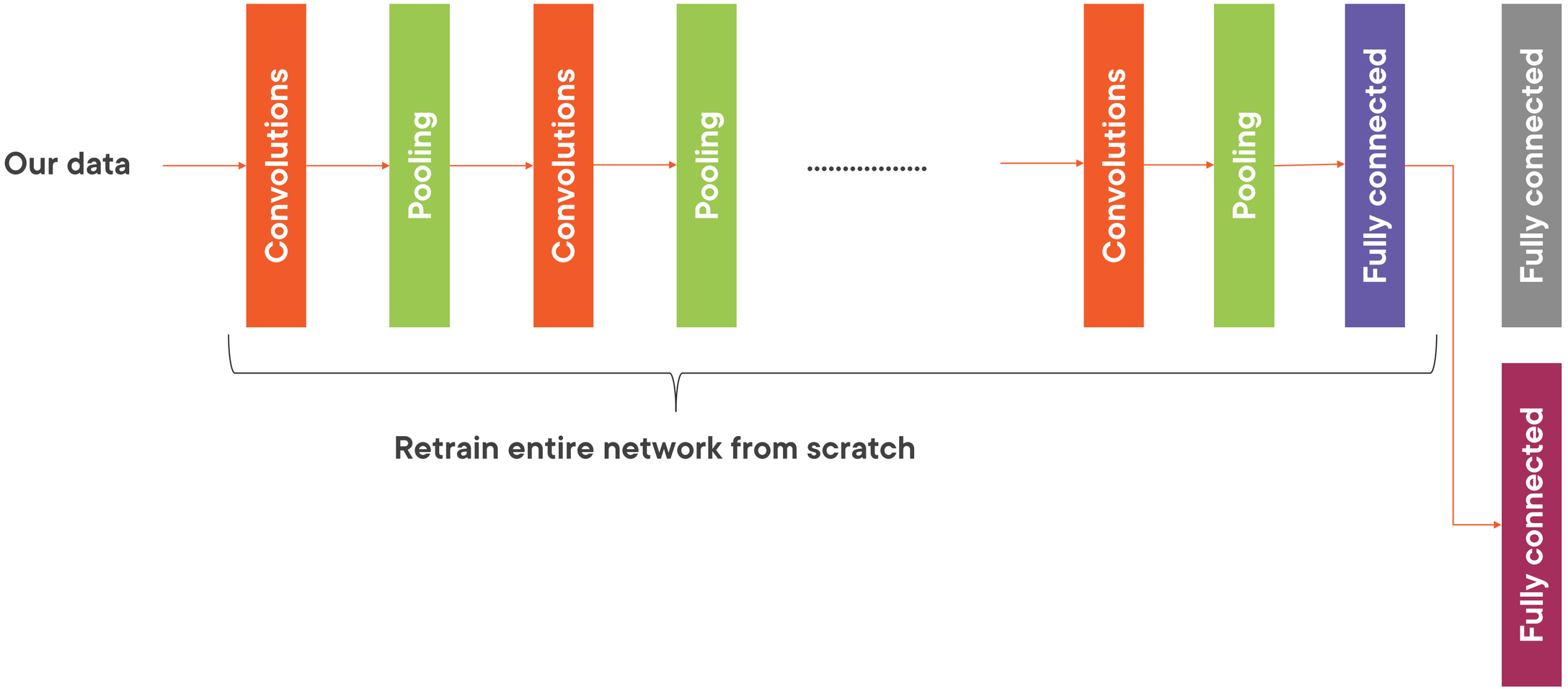
Case 3

Our dataset is large, also not similar to original dataset



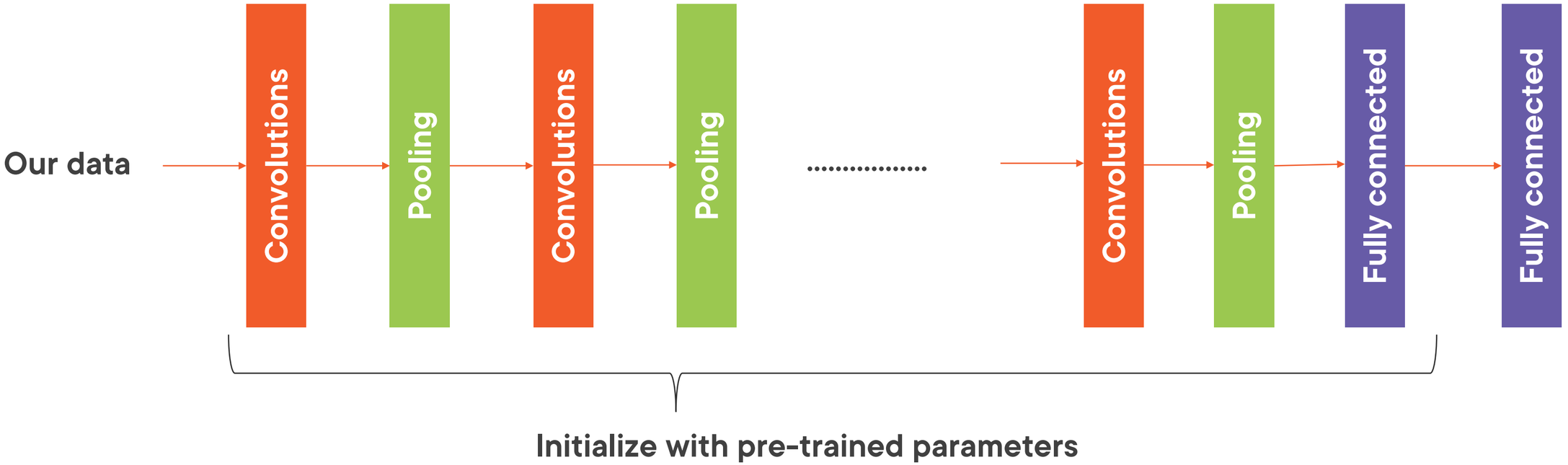
Case 3

Our dataset is large, also not similar to original dataset



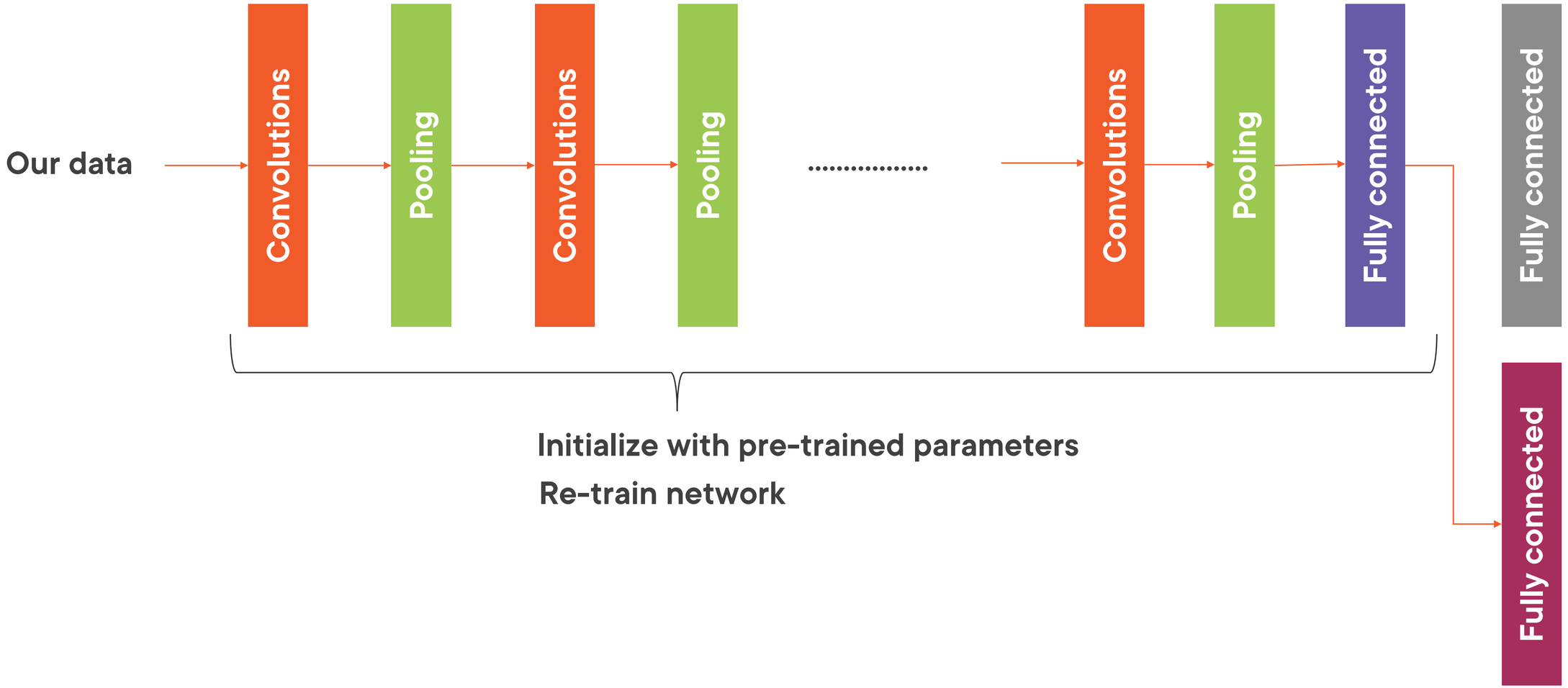
Case 4

Our dataset is large, also very similar to original dataset

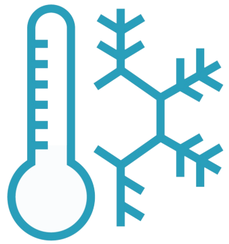


Case 4

Our dataset is large, also very similar to original dataset



Bottleneck Features

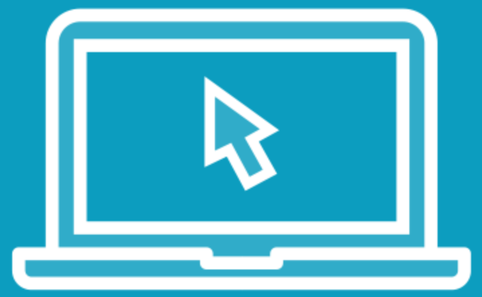


In TL, we fix / free layers of the network according to requirement

These fixed parameters / frozen layers are fed into customized FC layer

The input to the new FC layer is called “Bottleneck Feature”

Demo



Improving performance through transfer learning

Summary

Simple networks will struggle with complex problems

4 methods for better performance

- Complex neural network
- Image augmentation
- Hyperparameter tuning
- Transfer learning

Performance metrics we choose to evaluate on are problem dependent

Feedback

Discussion tab for any feedback or questions

Leave a star rating!

Thank you for watching!