Deep Scratch

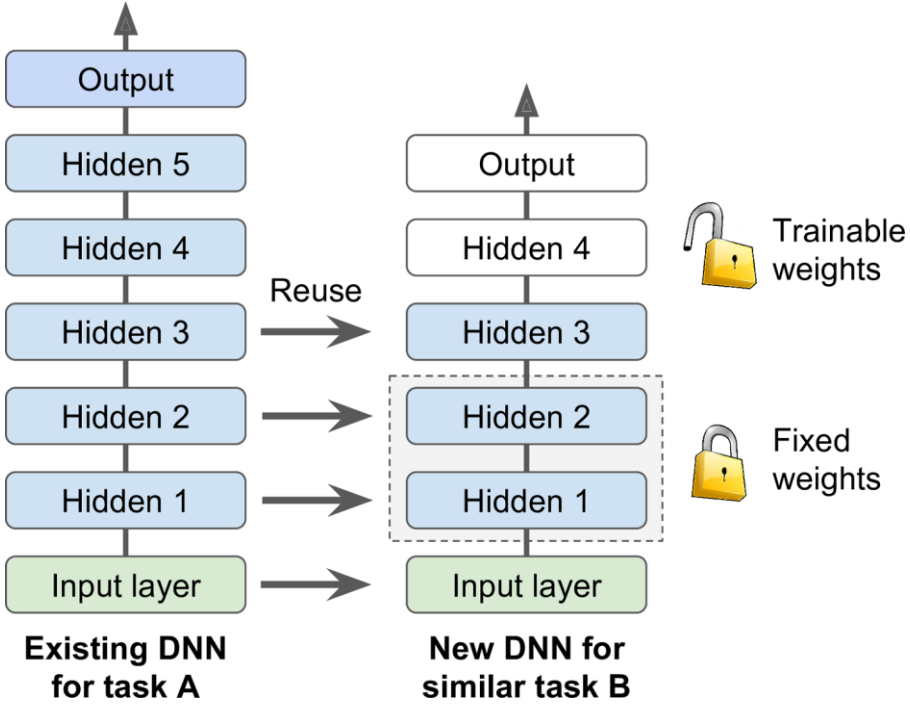
Reusing Pretrained Layers
Transfer Learning

박 희 경

CONTENTS

1. Introduction

Transfer Learning



It is generally not a good idea to train a very large DNN from scratch.

Transfer Learning

: Reusing the lower layers of an existing neural network

If the input picture don't have the same size as the ones used in the original task, you will have to resize them to the size expected by the original model.

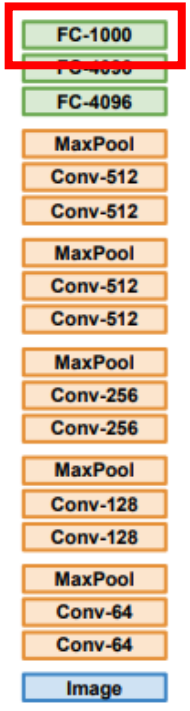
2. Freezing the Lower Layers

Small Dataset

Existing ConvNet

Reuse

Reusing ConvNet



Reinitialize & Train

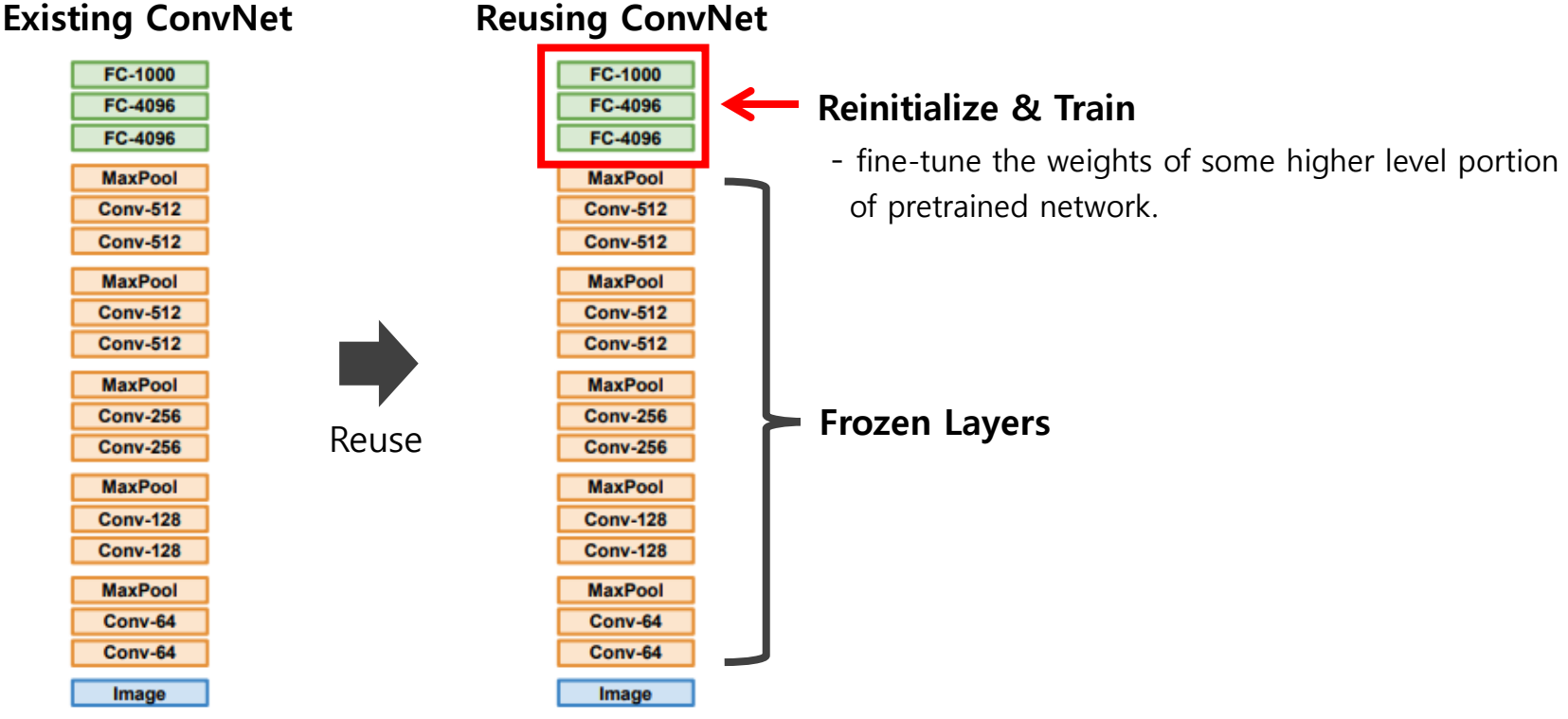
- set for your problem.
  (If you have C classes, then do FC-C.)
- replace and retrain the classifier

Frozen Layers

- freeze these weights when training the network.
- learning rate = 0
- during training, give the optimizer
  the list of variables to train,
  excluding the variables from lower layers.

2. Freezing the Lower Layers

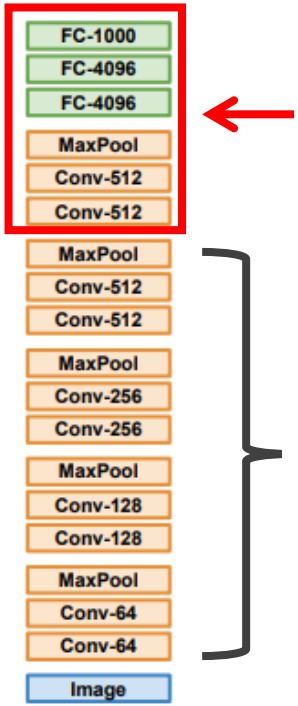
Bigger Dataset



3. Tweaking, Dropping, or Replacing the Upper Layers

Tweaking

Reusing ConvNet



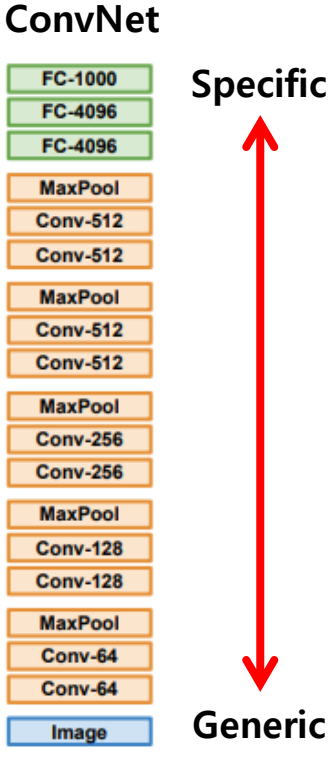
Reinitialize & Train

- the observation that the earlier features of a ConvNet contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks, but later layers of the ConvNet becomes progressively more specific to the details of the classes contained in the original dataset.
- lower learning rate when fine tuning.
  (1/10 of original learning rate is good starting point.
  Especially, On Conv, more lower learning rate)

Frozen Layers

3. Tweaking, Dropping, or Replacing the Upper Layers

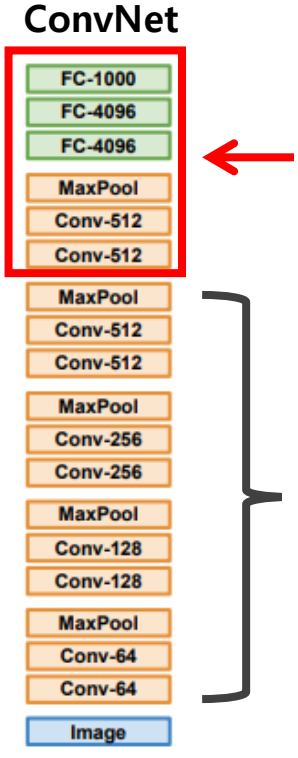
Tweaking



	Very similar dataset	**Very different dataset**
Very little data	Use Linear Classifier on top layer	You're in trouble... Try linear classifier from different stages
Quite a lot of data	Finetune a few layers	Finetune a larger number of layers

3. Tweaking, Dropping, or Replacing the Upper Layers

Dropping & Replacing



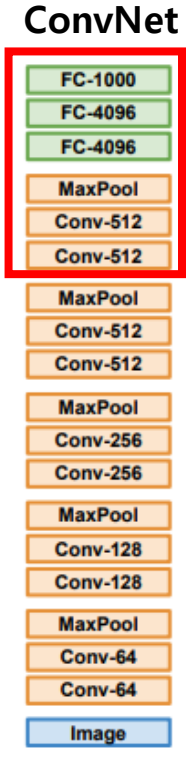
Reinitialize & Train with dropping or replacing

- If you cannot get good performance, and you have little training data, try dropping the top hidden layers.

- If you have plenty of training data, you may try replacing the top hidden layers instead of dropping them, and even add more hidden layers

4. Caching the Frozen Layers

Caching



Reinitialize & Train with Caching

① run the whole training set through the lower layers

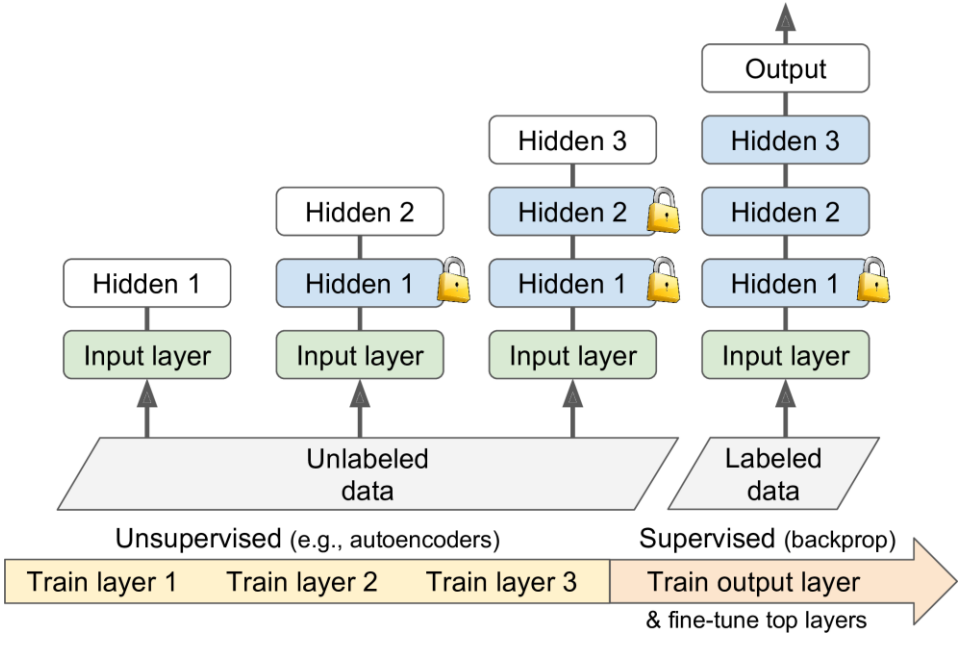
② during training, instead of building batches of training instances, you would build batches of outputs from hidden layers and feed them to the training operation.

→ It will give you a huge speed boost

Frozen Layers

5. Unsupervised Pretraining

Plenty of unlabeled data



If you don't have much labeled training data, you may be able to perform unsupervised pretraining.

① try to train the layers one by one,
starting with the lowest layer
and then going up,
using an unsupervised feature detector
(e.g. RBMs, autoencoders)

② Once all layers have been trained,
you can finetune the network
using supervised learning