

Directed Study Report
CSE4DIR

Ash Hall
17756156

May 15, 2018

Contents

1.1	Introduction	2
1.2	Transfer Learning	2
1.3	Continuous Learning	3
	1.3.1 Related Works	3
	1.3.2 Environment	3
1.4	Framework	3
1.5	Training Procedure and Metrics	3
1.6	Benchmark Experiments and Results	3
	1.6.1 Experiment 1	3
	1.6.2 Experiment 2	4
	1.6.3 Experiment 3	4
1.7	Summary	4

Directed Study Report

1.1 Introduction

(This will be written properly once the rest is done)

This report is about the effects of catastrophic interference in neural networks. We'll discuss some techniques for mitigating these effects, and will explore the severity of each. The benefit of exploring these things.

1.2 Transfer Learning

The training of a neural network occurs by presenting many examples, and iteratively optimising the weights. A dataset consisting of a small number of examples may be prohibitive to optimisation in this manner, as a model would likely over-fit to the training examples, and not be capable of generalising to novel examples. *Transfer learning* minimising this problem, by allowing a pre-trained model to be re-purposed to a different dataset.

The bulk of an image-classification neural network acts as a *feature-extractor*, with studies[1] showing that a sufficiently-large image dataset leads to a highly-generalisable feature extractor. Transfer learning takes advantage of this property by taking the resultant feature extractor and amending the tail of a neural network to perform classification between different classes. Figure 1.1 shows a model pre-trained on a large image dataset being adapted to another.

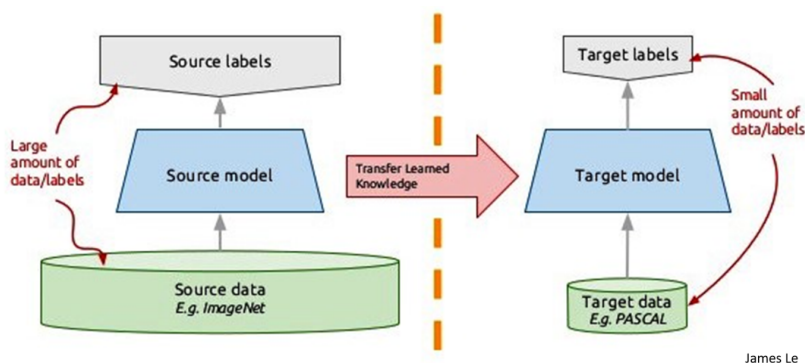


Figure 1.1: Transfer learning overview

There are two fundamental problems with this approach. The first is that although this

lessens the effects of over-fitting, it doesn't entirely avoid them; training on a very small dataset is still prone to this problem. The other issue is that if you wished to not replace the old dataset classes, but add the new dataset classes to the model, you would need to store examples and repeatedly present them to the network during re-training. This introduces a whole suite of problems including training-example balancing, which we won't address here.

Simply put, while providing strong results, transfer learning is limited in its usability, and is inherently inapplicable to the task of continuous learning – which we'll address in the following section.

1.3 Continuous Learning

What continuous learning is.

Why continuous learning is different to transfer learning (don't look at old examples)

Briefly explain some techniques

1.3.1 Related Works

Intro to related works

A few related works that address catastrophic interference with continuous learning, why they're good and bad.

Summary of related works

1.3.2 Environment

State the environment - (Ubuntu, docker, tensorflow etc.)

1.4 Framework

A description of the framework heirarchy. I'll give a simple diagram of the file-structure/python modules in the project, then explain roughly what everything is responsible for.

1.5 Training Procedure and Metrics

Here I'll explain how I trained each of the models (the stuff that's consistent between them), and give an introduction to some terms, and explain the metrics I used.

1.6 Benchmark Experiments and Results

I'll put a short introduction to the results here,

1.6.1 Experiment 1

(These will actually have names, not just experiment 1, 2, ...)

1.6.2 Experiment 2

1.6.3 Experiment 3

1.7 Summary

Summarise the experiment results and what impact they have on my project, summarise the work that I've done and where it leaves my with regards to the second semester and the "actual work".