

Ash

March 9, 2018

Contents

1	Introduction	3
2	Background	4
2.1	Hand Engineered vs Learnt Features	4
2.2	Supervised Learning	4
2.3	Optimisation	4
2.3.1	Loss	4
2.3.2	SGD	5
2.3.3	Backprop	5
2.3.4	Auto optimizers like ADAM and RMSProp	5
2.4	Dealing with Small Training Data Sets	5
2.4.1	Overfitting	5
2.4.2	Transfer Learning	5
2.4.3	Few-Shot Learning	5
2.4.4	Meta Learning	5
2.5	Continuous Learning	5
2.6	Modern Deep Learning Architectures	6
2.6.1	Convolutional Neural Networks	6
2.6.2	Recurrent Neural Networks	6
3	Related Works	7
3.1	Meta Learning	7
3.1.1	Model Based	7
3.1.2	Metric Based	7
3.1.3	Optimization Based	7
3.2	Continuous Learning	7
4	Proposal	8

List of Figures

Chapter 1

Introduction

Very brief background Deep learning good for large data set. Motivation is to work with less training examples

Existing work on meta learning and continuous learning

Gap the literature

Describe the Problem A system that can do continual learning. Refer to this doc: <https://paper.dropbox.com/doc/A-classes-an-existing-classifier-RdKxXHh7M9OWbHvEvCCsV> We want it to be scalable with respect to the number of classes So it should work faster than nearest neighbour based approaches for large number of classes

High level how you will solve it and why it is different from existing work

Brief description of experimental setup

Chapter 2

Background

2.1 Hand Engineered vs Learnt Features

Features are a general term for characteristic attributes which exist across all samples in a data-set or domain. These features were traditionally hand-engineered by machine learning experts, carefully selecting the base-components of which the data-set in question appears to comprise.

A fundamental problem with hand-engineered features is that it imposes human knowledge onto a problem to be solved by a computer. Furthermore, key features for complex data such as images and video are incredibly difficult to ascertain – especially if desiring generic, transferable features. With the rise of neural networks – specifically CNNs, which will be discussed in detail later – feature-learning has become the norm. This essentially takes the task of feature engineering and solves it in a data-driven manner.

2.2 Supervised Learning

Supervised learning is a machine learning strategy whereby the target solution is presented after each training iteration. This differs from unsupervised learning in that unsupervised learning has no direct target to learn from and is used to find underlying commonalities or patterns in data.

Supervised learning is the most commonly used method for image and video tasks, as typically the objective is to perform tasks where the target is well-defined. Common supervised learning tasks are *image classification* - where the objective is to assign an input image a label from a fixed set of categories; *localisation* - where the objective is to produce the coordinates of an object of interest from the input image; and *detection* - which combines the previous two tasks.

There are a multitude of supervised and unsupervised learning problems. As with almost all meta-learning strategies, I will focus primarily on the supervised task of image classification.

2.3 Optimisation

2.3.1 Loss

The process of optimising a machine learning system is to present it with a target of some sort – either in the supervised or unsupervised setting – and compute a numerical quantity called either *loss* or *cost*. It is the system’s objective to minimise this value through some optimisation algorithm.

The particular loss function is specifically chosen for the task at hand, cross-entropy being a common choice for image classification tasks.

Before discussing the cross-entropy loss function, it's important to understand the outputs of a machine learning system when performing image classification.

For a system that makes predictions between N classes, its output is a vector of length N , where each of the output values N_i is a score for that class i being the correct answer.

As the outputs aren't normalised and thus cannot be interpreted as a true confidence measure, the outputs then normally go through a softmax function σ .

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}} \quad (2.1)$$

The softmax function (eq 2.1) squashes the arbitrary scores into a vector of values such that their values are in the range $[0, 1]$ and that all components sum to 1. The resultant values can be interpreted as the probability measures for the input image falling into each of the classes.

- What loss is
- Loss curve
- Gradient descent
- Why
- How

2.3.2 SGD

- Why use Sgd

2.3.3 Backprop

2.3.4 Auto optimizers like ADAM and RMSProp

2.4 Dealing with Small Training Data Sets

2.4.1 Overfitting

2.4.2 Transfer Learning

2.4.3 Few-Shot Learning

2.4.4 Meta Learning

Break into meta training, testing, episodes, etc.

2.5 Continuous Learning

Catastrophic forgetting

2.6 Modern Deep Learning Architectures

2.6.1 Convolutional Neural Networks

2.6.2 Recurrent Neural Networks

Chapter 3

Related Works

3.1 Meta Learning

Approaches in meta learning with neural networks are generally groupd into three categories.

3.1.1 Model Based

3.1.2 Metric Based

3.1.3 Optimization Based

3.2 Continuous Learning

Chapter 4

Proposal