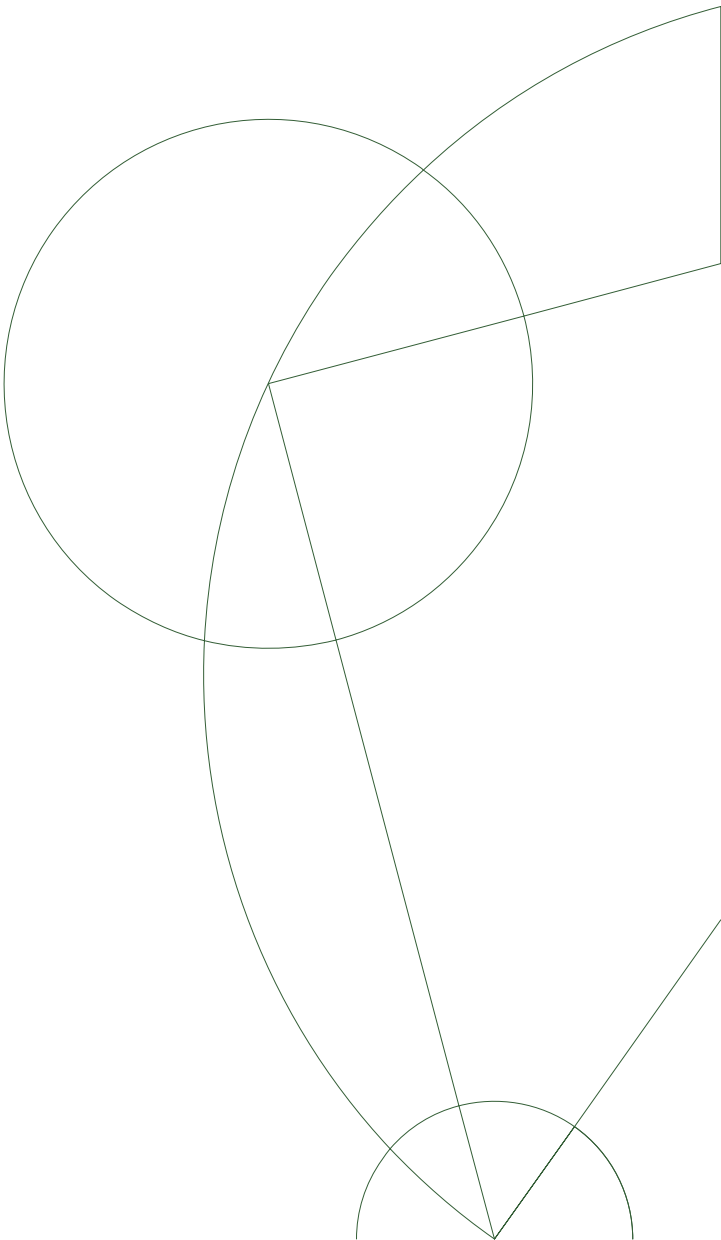
FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

Master Thesis

Sune Andreas Dybro Debel

Deep Multi-Task Learning

For Relation Extraction



Dirk Hovy

February 24, 2017

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

Part 1

Introduction

Part 2

Background

In this part, we exemplify the information extraction problem and formally describe the supervised machine learning and multi-task learning setting. In addition, we describe neural networks, the concept of representation learning and its relevance to multi-task learning.

2.1 Information Extraction

In natural language processing, information extraction is the problem of extracting structured information from unstructured text. Many practical information extraction problems fall in one of two categories: **named entity recognition**, or **relation extraction** (Jurafsky and Martin, 2009).

2.1.1 Named Entity Recognition

A named entity is roughly anything that has a proper name. The task in named entity recognition is to label mentions of entities such as people, organisations or places occurring in natural language. As an example, consider the sentence:

Jim bought 300 shares of Acme Corp. in 2006.

A named entity recognition system designed to extract the entities *person* and *organisation* should ideally assign the labels:

$[\text{Jim}]_{person}$ bought 300 shares of $[\text{Acme Corp.}]_{organisation}$ in 2006.

This is a difficult problem because of two types of ambiguity. Firstly, two distinct entities may share the same name and category, such as *Francis Bacon* the painter and *Francis Bacon* the philosopher. Secondly, two distinct entities can have the same name, but belong to different categories such as *JFK* the former American president and *JFK* the airport near New York.

Named entity recognition can be framed as a sequence labelling problem. A common approach is to apply so called tokenisation to the text, i.e finding boundaries between words and punctuation, and associate each token with a label indicating which entity it belongs to. BIO (figure 2.1) is a widely used labelling scheme in which token labels indicate whether the token is at the **B**eginning, **I**nside, or **O**utside an entity mention.

Jim	bought	300	shares	of	Acme	Corp	.	in	2006	.
B-PER	O	O	O	O	B-ORG	I-ORG	I-ORG	O	O	O

Figure 2.1: A sentence labelled with the BIO labels for named entity recognition.

2.1.2 Relation Extraction

Relation extraction refers to the problem of identifying relationships between entities. As an example, consider the sentence

> Yesterday, New York based Foo Inc. announced their acquisition of Bar Corp.

Imagine we have designed a relation extraction system that recognises the relation *MergerBetween(organisation, organisation)* between two mentions of organisations. Ideally, we would like that system to extract the relation *MergerBetween(Foo Inc., Bar Corp.)* from the above sentence.

2.2 Supervised Machine Learning

Most solutions to the information extraction problems in 2.1 are based on supervised machine learning techniques. In this setting, a system learns to predict the named entities or relations between them given a unit of natural language by inspecting examples provided by a human annotator.

2.2.1 The Supervised Learning Problem

A set $\mathcal{D}_{train}$ of N training examples $(\mathbf{x}_i, \mathbf{y}_i)$ of inputs $\mathbf{x}_i$ and corresponding labels $\mathbf{y}_i$ is created by a human annotator. Each $\mathbf{x}_i$ belongs to a space $\mathcal{X}$ of input dimensions and each $\mathbf{y}_i$ belongs to a space $\mathcal{Y}$ of labels. We want to use this data to automatically assign labels to a new set of un-labeled inputs $\mathcal{D}_{test} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{X}\}$ at some point in the future. The task is to learn a function $h : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}_{train}$ that performs well on $\mathcal{D}_{test}$. In particular, we are not explicitly interested in the performance of h on $\mathcal{D}_{train}$ (Abu-Mostafa et al., 2012).

We can formalise the preference for functions h that perform well on examples outside of the training set with a quantity known as **generalisation error**.

Definition 2.2.1 (generalisation error)**.** Let $P(\mathbf{x}, \mathbf{y})$ be a joint probability distribution over inputs $\mathbf{x} \in \mathcal{X}$ and labels $\mathbf{y} \in \mathcal{Y}$. Let $e(\mathbf{y}_1, \mathbf{y}_2)$ be an error measure that measures agreement between labels $\mathbf{y}_1$ and $\mathbf{y}_2$. Then the generalisation error E of a function $h : \mathcal{X} \mapsto \mathcal{Y}$ is defined as:

$$E(h) = \mathbb{E}_{\mathbf{x},\mathbf{y} \sim P(\mathbf{y},\mathbf{x})}[e(h(\mathbf{x}), \mathbf{y})]$$

Now, formally, the objective of supervised machine learning is to find a function h^* in a space of functions $\mathcal{H}$ that minimises $E(h)$. Unfortunately, $P(\mathbf{x}, \mathbf{y})$ is unknown (if it was not, no learning would be required!). However, we can use $\mathcal{D}_{train}$ to estimate $E(h)$ with a quantity known as **training error**:

Definition 2.2.2 (training error)**.** Let $\mathcal{D}$ be a set of N training examples $\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i, \mathbf{y}_i \sim P(\mathbf{x}, \mathbf{y})\}$. Then the training error $\hat{E}$ is defined as:

$$\hat{E}(h, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} e(h(\mathbf{x}_i), \mathbf{y}_i)$$

Using $\hat{E}$ to estimate E is dangerous for two reasons. Firstly, $\mathcal{D}$ is a random quantity. It may happen that the samples available to us are not representative of $P(\mathbf{x}, \mathbf{y})$, leading us to choose h that does not perform well in terms of e on examples outside the training set.

Secondly, even if $\mathcal{D}$ is a good sample, it's possible to choose h such that $\hat{E}$ is small, but E is large. In particular, if $\mathcal{H}$ is very large, there likely exists a h such that $\hat{E}(h, \mathcal{D}) = 0$. This is problematic for two reasons. Firstly the relationship between $\mathbf{x}$ and $\mathbf{y}$ may be noisy, that is $P(\mathbf{y} \mid \mathbf{x})$ is non-deterministic. In this case, by selecting h such that $\hat{E}(h, \mathcal{D}) = 0$, we have fitted not only the general relationship between $\mathbf{x}$ and $\mathbf{y}$, but also the noise which is particular to $\mathcal{D}$.

Secondly, even if $P(\mathbf{y} \mid \mathbf{x})$ is deterministic, fitting it exactly will likely lead to poor generalisation since $\mathcal{D}$ is a finite sample.

Understanding the relationship between E, $\hat{E}$, $\mathcal{H}$ and $\mathcal{D}$ is the objective of a field of research known as **statistical learning theory**.

2.2.2 Statistical Learning Theory

In 2.2.1 we argued informally that the relationship between E and $\hat{E}$ depends on $\mathcal{D}$ and $\mathcal{H}$. The objective of statistical learning theory is to bound E in terms of $\hat{E}$, $\mathcal{H}$ and $\mathcal{D}$. A central aspect of this problem is the question of what features of $\mathcal{H}$ determine how well a function h in $\mathcal{H}$ is able to fit $\mathcal{D}$, since the examples therein are finite and may be noisy.

A **dichotomy** is an important concept in measuring how well $\mathcal{H}$ can fit an arbitrary $\mathcal{D}$.

2.3 Neural Networks

2.3.1 Feed-Forward Neural Networks

2.3.2 Learning Algorithm

2.3.3 Convolutional Neural Networks

2.3.4 Recurrent Neural Networks

2.4 Multi-Task Learning

2.5 Representation Learning

2.6 Deep Multi-Task Learning

2.7 Summary

Part 3

Experiment

3.1 Network Architecture

3.2 Auxiliary Tasks

Part 4

Results

Part 5

Discussion

Part 6

Conclusion

Bibliography

Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*. AMLbook.com, 2012.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson Education, international edition, 2009.

Appendix A

Appendix

A.1 First