



**KTH Information and
Communication Technology**

Data-Efficient Deep Learning for Independent Binary Outputs

Exploration of importance-weighted active learning, ensembling, joint training and class imbalance correction to reduce label complexity and training time in affiliate e-commerce product classification

MATTIAS ARRO

Master's Thesis at KTH Information and Communication Technology
MSc Data Science (EIT Digital track)

Academic Examiner: Magnus Boman
Academic Supervisor: Jim Dowling
Industrial Supervisor: Abubakreledik Karali

2018

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Keywords: Deep learning, machine learning, neural networks, active learning

Referat

Denna fil ger ett avhandlingsskelett. Mer information om
L^AT_EX-mallen finns i dokumentationen till paketet.

Acknowledgment

..... London, UK, March 27, 2018
Mattias Arro

Contents

Introduction	3
Problem	4
Purpose	5
Goal	5
Research Questions	5
Methodology	5
Evaluation	5
Work Environment	5
Deployment Environment	5
Ethics and Sustainability	5
Delimitations	5
Outline	5
Discussion	7
Conclusion	9
References	11
Declaration	13
Appendices	13
RDF	15

Abbreviations

LSTM Long Short Term Memory

NN Neural Network

RNN Recurrent Neural Network

Introduction

Machine learning (ML) has become hugely successful over the past few years, with a lot of this newfound interest, hype, and hysteria directed at neural networks and deep learning. This focus is not unfounded - deep learning approaches continue to break benchmarks in core machine learning research areas such as computer vision [cite], speech recognition [cite], and an increasing number of natural language processing tasks [cite NMT]. Deep learning has also revolutionised reinforcement learning, achieving superhuman performance in complex games and driving vehicles in real-world situations. There are even limited results in beating human at highly uncertain games with various actors such as [Texas Holdem] poker [cite]. Understandably, academics and industry are scrambling to apply panacea to their field.

While their superficial resemblance to natural brains might be a good topic for a sensationalist article, artificial neural networks are simply layers of non-linear transformations capable of learning complex mappings from multidimensional inputs to (usually multidimensional or structured) outputs. The building blocks of neural networks are relatively simple and the algorithms for training them are universal; this makes neural networks applicable to a variety of domains, and opens up fascinating opportunities of multimodal and transfer learning. Being able to arbitrarily increase model complexity by increasing its depth or width allows the same neural network approximate more complex functions. Increased model complexity increases training time and requires more labeled training data, yet deep models are somewhat unique in that their performance continues to increase when the dataset size increases, whereas the benefits of more data taper off for many other kinds of models. This does not automatically mean neural networks can only be used with large datasets - after all a single layer neural network can be equivalent to a logistic regression model - but that model complexity should increase with the amount of data available.

It is not immediately obvious which kind of model should one use for a given task and dataset. A data scientist can consider the following factors: how many labelled and unlabelled data points do we have, how much do we value predictive performance, interpretability, and whether we want to do some transfer learning or joint training with the models. Labelling is often expensive, so in many real world use cases a lower label complexity (number of labels needed to obtain the desired accuracy) is preferred over slightly better performance. Deep learning seems to have

a disadvantage in this aspect, but as we see in section [ref] in cases where unlabelled data is also abundant, semi-supervised and generative models can overcome low label complexity while increasing computation time. In cases where the ability of neural networks to learn features that can be used in downstream models (e.g. features learned for classification could later be used as part of a recommended system) this increased computation and engineering complexity might be justifiable.

In this thesis, we explore three orthogonal ways of efficiently learning on a proprietary dataset for product classification, where initial labels are abundant but noisy. We first evaluate different kinds of models (shallow, deep, tree-structured, convolutional, recurrent) that are trained on different modalities / input dimensions (image, text, categorical, numerical) of the same data. After determining the performance of these baseline models, the strongest models are trained as an ensemble that outperforms each individual baseline model. Finally we fine-tune the ensemble via an active learning strategy described in section [ref], where a combination of uncertainty and disagreement sampling determines a batch of products to be labeled for the next training iteration. This overcomes the noisiness and incompleteness of the initial labels without requiring much manual labeling.

Problem

The client company gathers data from various affiliate networks (that in turn give their data from various retailers) and displays the data on their online store. There are millions of products belonging to roughly 800 categories, and categories follow the usual nested tree structure. The incoming data is extremely noisy and inconsistent: what kind of data is stored in what kind of feature column varies across affiliate networks, across retailers within an affiliate network, and the data within a retailer can have lots of missing values, noisy text, missing images, etc. There is currently a rule-based system for assigning products to categories: all products matching a condition (e.g. title contains the word "trousers") will be assigned to that category, i.e. categories are not mutually exclusive. This way of categorising products works relatively well on some categories, but such a rule-based system has several drawbacks: these rules are cumbersome to define, their evaluation is manual, they failed to match a large fraction of products that in principle should be in a given category, it is hard to trace back the rule that caused a false positive, and such rules are limited to textual data.

The client needs a classification system of binary independent outputs to replace the old way of categorising products. The system should be able to learn from the output of the old system, and if possible produce models that can be used in downstream tasks such as recommend systems and product similarity models (transfer learning). The highest priority is low label complexity, beating requirements for high accuracy and transfer learning capabilities. The system should be robust to noisy inputs, and data preprocessing should not consider the idiosyncrasies of each affiliate network.

PURPOSE

Purpose

the purpose of this work is to

Goal

Research Questions

- How do the models outlined in section [ref] compared to one another in terms of predictive performance (defined in section), converging time, and label complexity?

Methodology

Evaluation

Work Environment

Deployment Environment

Ethics and Sustainability

Delimitations

Outline

Discussion

Conclusion

References

Declaration

I hereby certify that I have written this thesis independently and have only used the specified sources and resources indicated in the bibliography.

London, UK, March 27, 2018

.....
Mattias Arro

RDF

And here is a figure

Figure 1. Several statements describing the same resource.

that we refer to here: 1