

UNIVERSITY OF BERGEN
DEPARTMENT OF INFORMATICS

Multitask variational autoencoders

Author: Edvards Zakovskis

Supervisors: Nello Blaser and Kristian Gundersen



UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

September, 2023

Abstract

Lorem ipsum dolor sit amet, his veri singulis necessitatibus ad. Nec insolens periculis ex. Te pro purto eros error, nec alia graeci placerat cu. Hinc volutpat similique no qui, ad labitur mentitum democritum sea. Sale inimicus te eum.

No eros nemore impedit his, per at salutandi eloquentiam, ea semper euismod meliore sea. Mutat scaevola cotidieque cu mel. Eum an convenire tractatos, ei duo nulla molestie, quis hendrerit et vix. In aliquam intellegam philosophia sea. At quo bonorum adipisci. Eros labitur deleniti ius in, sonet congrue ius at, pro suas meis habeo no.

Acknowledgements

Est suavitate gubergren referrentur an, ex mea dolor eloquentiam, novum ludus suscipit in nec. Ea mea essent prompta constituam, has ut novum prodesset vulputate. Ad noster electram pri, nec sint accusamus dissentias at. Est ad laoreet fierent invidunt, ut per assueverit conclusionemque. An electram efficiendi mea.

Your name

Monday 11th September, 2023

Contents

1	Introduction	1
1.1	Research Question	2
2	Background	3
	Bibliography	4
A	Generated code from Protocol buffers	5

List of Figures

List of Tables

Listings

A.1 Source code of something	5
--	---

Chapter 1

Introduction

In recent years, Variational Autoencoders (VAEs) have emerged as a powerful tool in the field of deep learning and generative modeling. VAEs offer a principled framework for capturing complex data distributions and have found applications in diverse domains, including image generation, anomaly detection, and natural language processing. However, like any other machine learning model, VAEs have their own set of limitations and challenges.

One of the key challenges in training VAEs is achieving a balance between the reconstruction accuracy and the effectiveness of the learned latent representations. Traditional VAEs are typically trained to optimize a single objective, which often results in suboptimal performance on complex datasets with multiple sources of variation. In real-world scenarios, data often exhibits multiple underlying structures and tasks, and capturing these structures with a single VAE can be a challenging endeavor.

Multitask learning, on the other hand, is a paradigm that aims to improve model generalization and performance by simultaneously learning multiple related tasks. The idea of combining VAEs with multitask learning has recently gained attention in the machine learning community. Multitask Variational Autoencoders (MT-VAEs) extend the VAE framework to enable the joint learning of multiple latent variable spaces, each dedicated to a specific task or source of variation. This approach holds the promise of not only improving the reconstruction accuracy of VAEs but also enhancing their interpretability and generalization capabilities.

This research endeavors to investigate the effectiveness of MT-VAEs and their potential to outperform traditional VAEs in terms of data generation, representation learning,

and task-specific performance. By conducting a comprehensive analysis and experimentation, we aim to shed light on the advantages and limitations of MT-VAEs and provide insights into the conditions under which they excel.

1.1 Research Question

The central research question addressed in this thesis is as follows:

Can Multitask Variational Autoencoders (MT-VAEs) enhance the performance and versatility of Variational Autoencoders (VAEs) by jointly optimizing multiple tasks, and under what conditions do MT-VAEs demonstrate superior performance in comparison to traditional VAEs?

Chapter 2

Background

Variational Autoencoders (VAEs) have emerged as a prominent class of generative models in the field of machine learning, where they first emerged in

Bibliography

Appendix A

Generated code from Protocol buffers

Listing A.1: Source code of something

```
1 System.out.println("Hello Mars");
```