University of Bergen
Department of Informatics

Machine Teaching using minimzation over complexity of instances

Author: Brigt Arve Toppe Håvardstun

Supervisors: Jan Arne Telle

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

March, 2022

Abstract

In todays society AI and machine learning is becoming more and more relevant. (find source) For instance, some might say the age old problem of the protein folding has finally been solved thanks to Deep mind and their Alphafold 2 [1] . Their deep learning approach achieved close to 90% accuray, matching experimental approaches. However, even if we now can get greatly accurat approximation solutions to the question of protein folding, the question of how each protein fold into their 3D structure has not been solved yet. Motivated by this fact; that knowing solutions to instances not nessecery gives insight to the question at large, as well as the expanding use of AI in our everday life [image recognition, recommendation systems, personal medecin and law]. This places Explaineable AI as a topic of of extrem relevancy in the comming years. Our work will regard the topic of model-agnoitic, global, example baed Explaineable Machine Learning. We will focus on the complexity of each instance to be used for teaching, and not on how many examples we are going to use.

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.

Brigt Arve Toppe Håvardtu
Thursday 3rd March, 2022

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

In this master thesis we implement and discuss the result of the propsed model for machine teaching.

1.1 Background

This master thesis is connected to the reasearch of {citation or refrence on research proposal}. In their research proposal they put forward a formula for machine teaching 1.1.

$$T(\Theta_{AI}) = \underset{S:\Theta_{AI}\models S}{\operatorname{argmin}}\{\delta(S) + \lambda(\Theta_{AI}, \Theta_M) : L_M(S) = \Theta_M\} \tag{1.1}$$

$$L_M(S) = \underset{\Theta_M:\Theta_M\models S}{\operatorname{argmin}}\{\beta(\Theta_M)\} \tag{1.2}$$

In this thesis we discuss different implementations of 1.1. In particular we discuss different definitions of δ,λ, L_M, the selection of subsets S and selection of Θ_M.

Chapter 2

Models

In this thesis there are multiple ”models” and we will therefor establish cleare notations and descriptions of each.

2.1 Θ_{AI} - the AI model to be learned

With Θ_{ai} we refere to the AI model we are trying to teach to a human. In this thesis we will experimented with two different implementations of θai. The first being Convolutional Nural Network (CNN) [2], and the other being Fully Connected Nural Network (FCNN). Θ_{ai} will be trained on imags, and will try to preditc a ground truth boolean function. This will be discussed in the chapter Dataset. Both CNN and FCNN will be discussed in further implementations detail in [chapter FCNN] and [chapter CNN].

2.2 L_M - modeling human learner

L_M will be an algorithm trying to mimic human behavior. Given a set of instances and corresponding predicted values, L_M outputs a guessed Θ_M. {Should i here discuss different implementations, difficulty of mimicing human behavior etc?}

2.3 Θ_M - L_Ms model of Θ_{AI}

Θ_M is the output of L_M, and a model of Θ_{AI}. Model is understood as giving the same output on, essentially[note to explaination], the same input. Our Θ_M will be a boolean function, and hence not take in a bitmap. Rather it takes as input a truth assigment to four literals, corresponding to the four litterals "A", "B", "C", and "D".

Chapter 3

Dataset

In this chapter we discuss how the dataset was choosen, how we create it, and the different parameters acceseable when creating it.

3.1 Background

Our first decision was in desciding the task our AI was going to solve. In selecting the task our AI model was going to solve we had a few attributes in mind. We did not want something to complex, it had to be simple to implement and change, as we were to make a Proof Of Consept. At the same time the problem should not be too simple, we want to check if our method will convey the AI models errors with respect to the ground truth. We also wanted to able to test our system with different AIs. To achive this we wanted our task to achive this we wanted the ground truth to be easily changeable, so we can compare AIs trained on different ground truths.

In the end we chose the task of predicting an underlaying boolean function given bitmaps contaning a subset of the literals A,B,C,D. Present literals in the bitmap is set to True, and all others are set to False. The label for a given instance is then found by calulcateing the underlaying boolean function with literal assigned as descrebied above, giving a label of 1 – being True – or 0 – being False.

This choice fits our demands as we can easily create a new ground truth by swapping the ground truth boolean function. This means our model is not over complex. At the same time the bitmaps representation of literals gives us the possibility of huge traing datasets, which has the possibility of creating more or less complex datasets – different orientations, sizes and letter styles.

3.2 Description of an instance

In more detail, the input to our AI will be a 64x64 bitmap. The cells in the bitmap will either have the value 0 or 255. The cells having the value 255 will together make up one or more letter from a predetermined letter set $L = A, B, C, D$. Three examples of the instances are shown below.

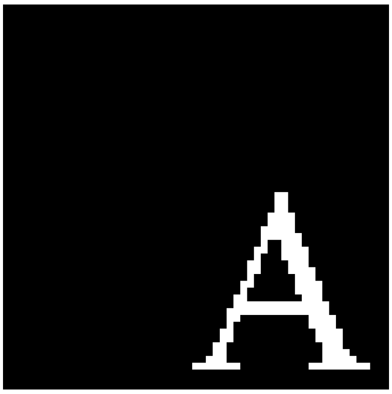

Figure 3.1: Bitmap A

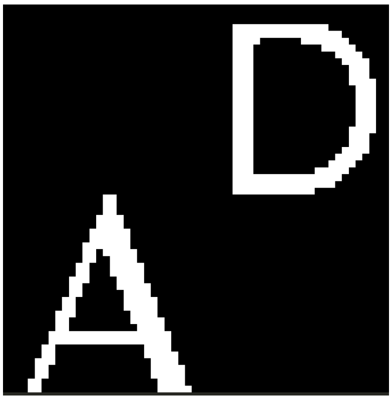

Figure 3.2: Bitmap AD

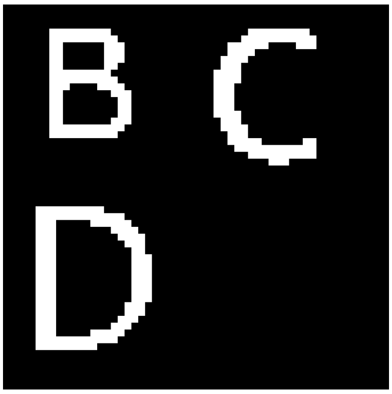

Figure 3.3: Bitmap BCD

We also have different parameters to active. These will be discussed in 3.4. The images above are of the smallest varience possible. Below are examples showing images from our dataset at highest possible varience.

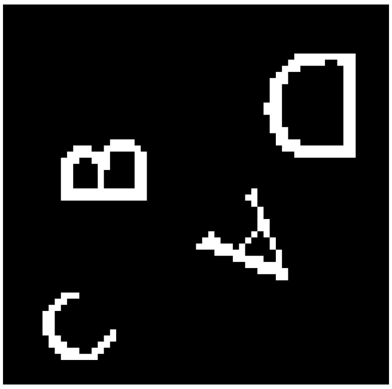

Figure 3.4: Bitmap ABCD

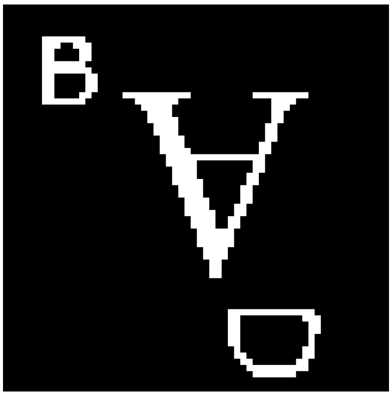

Figure 3.5: Bitmap ABD

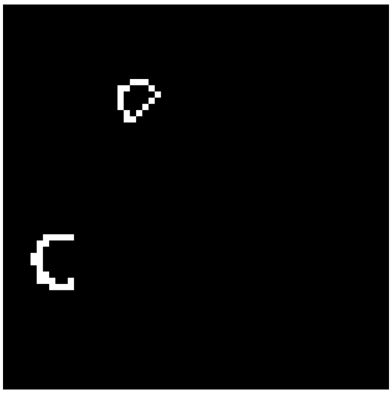

Figure 3.6: Bitmap CD

3.3 Properties of instance

We want to have som clear properties for each data instance. These are mostly sanity checks to make sure our AI will be given decent data. Altough intuitive, its important to make sure our data is sane. Bad data in, means bad perdictions comming out.

Our rules for the data goes as follows:

- For each letter, the entirety of the letter is to be contained within the picture.
- No letter is to overlap with another letter.
- The size of the letter should not be to small. The letter shall contain enough pixles, such that the characteristics of the letter is mainteind. E.g. clearly recognizeable for a human.

3.4 Parameters generating instances

When generating data we have quite a few differtent parameters, all contributing to more or less complexity in our data. The goal of having different parameters for dataset generating is to be able to have a smooth increasing in difficulty for our AI. Using this we can find a suitable dataset for our AI models, where they preforme resonabley well. We can also try to detect different overfittings based on these parameters.

3.4.1 $FixedSquares$ – placement of letter

For the parameter $FixedSquares$ we divide the image into four squares –top left, top right, bottom left, bottom right. Each of these square might then contain a letter. Crucially no letter is overlapping these squares. All of figures 3.1, 3.2 and 3.3 have $FixedSquares$. Having this parameter turned on will make the dataset less complex, as its counter part is to just place freely. Without $FixedSquares$ the algorithm might find all letters at the top of the image or at the bottom, increasing the varience of the dataset. This corresponds to a more complex dataset, less prone to be easily overfitted.

3.4.2 *Rotation* – orientation of the letter

With the parameter *Rotation* we alow the letters to have different orientations. We have slected a subset of orientations $O = \{0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°\}$. Letting each letter take on one of these orientations increase varience and complexity in our dataset. Below is an example of an image from a dataset with *Rotation* allowed.

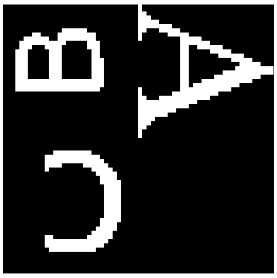

Figure 3.7: Bitmap w/rotation ABC

3.4.3 *Scale* - size of the letter

With *Scale* we allow letters to have different sizes. Importantly no letter can be made to big 3.8, as it has to fit inside the image. Nor can it be made to small 3.9, as it needs to be recognizeable.

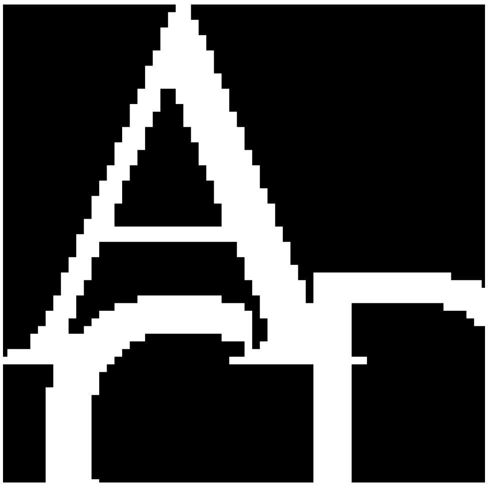

Figure 3.8: Bitmap ACD, too big. Overlapping, and out of image.

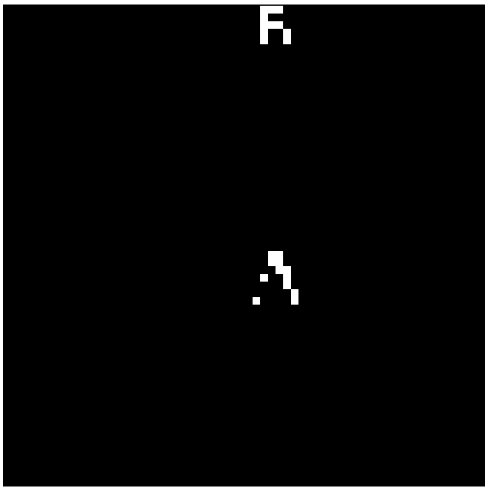

Figure 3.9: Bitmap AB, too small. A is broken up, B looks like R.

Having a broader range of sizes on the images will increase varience and complexity of our dataset.

3.5 Labels

After having described our data instance, we now describe the corresponding labels in the dataset. Our labeles will hold either the value 0 or the value 1. This corresponds to our ground truth boolean function returing True (1) or False (0). When evalueting a instance we retrive letters used to genereated this instace, and set the corresponding letters to true in our ground truth boolean function, and letters not present will be set to false. For instance given ground truth boolean function (A and not B) Bitmap A would be true, and hence get label 1, while Bitmap ABD would be false and have label 0.

Glossary

Θ_{AI} AI-model is the CNN or FCNN we want to teach to a human..

List of Acronyms and Abbreviations

CNN Convolutional Nural Network.

FCNN Fully Connected Nural Network.

Bibliography

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.
URL: `https://www.nature.com/articles/s41586-021-03819-2`. Number: 7873 Publisher: Nature Publishing Group.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015. ISSN 1476-4687. doi: 10.1038/nature14539.
URL: `https://doi.org/10.1038/nature14539`.

Appendix A

Generated code from Protocol buffers

Listing A.1: Source code of something

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