FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Effect of adversarial training on the decision boundary of convolutional neural networks

Optionaler Untertitel der Arbeit

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Software und Information Engineering

eingereicht von

Matthias Plasser
Matrikelnummer 0123456

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.univ.Prof. Dr. Andreas Rauber
Mitwirkung: Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle

Wien, 1. Jänner 2001

Matthias Plasser	Andreas Rauber

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Title of the Thesis

Optional Subtitle of the Thesis

BACHELOR’S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software and Information Engineering

by

Matthias Plasser
Registration Number 0123456

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.univ.Prof. Dr. Andreas Rauber
Assistance: Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle

Vienna, 1st January, 2001

Matthias Plasser   Andreas Rauber

Erklärung zur Verfassung der Arbeit

Matthias Plasser
Address

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Jänner 2001

____
Matthias Plasser

Danksagung

TODO: Ihr Text hier.

Acknowledgements

TODO: Enter your text here.

Kurzfassung

TODO: Ihr Text hier.

Abstract

TODO: Enter your text here.

Contents

CHAPTER 1

Introduction

Neural Networks are more reliable than ever in 2019, and outperform human experts in a broad range of tasks, from playing Go to recognizing fractures in MRIs. The existence of adversary examples - examples that are carefully, (almost) imperceptibly manipulated in order to get misclassified by neural networks challenges that reliability. Adversary inputs can be constructed by starting from and input that is initially correctly labelled by a classifier. By carefully manipulating features by a tiny amount in the correct direction - towards the decision boundary, it's possible to craft data that to a human still appears like the original input, yet gets labeled as something else by the given classifier. Adversarial inputs can even be crafted in hardware, like street signs. By adding special stickers, that do not influence a human's recognition, neural networks can be tricked to misclassify a picture of the manipulated sign. As long as neural networks can be fooled so easily, autonomous driving is very vulnerable. A remedy against adversarial inputs is adversarial training - using those manipulated inputs for training. In this paper, the effectiveness of different kinds of adversarial training, and especially how it influences the decision parameters of CNNs will be discussed and visualized. As decision parameters are hard to interpret, a technique called class activation maps is used for visualizing the decision parameters. The form of adversary inputs also exposes features of the decision boundary, which lays between the original and the adversary input.

CHAPTER 2

Fundamentals

CHAPTER 3

Related Work

This chapter will briefly summarize publications that are similar to this work, or contain fundamentals for it, and state, how this publications are relevant.

Intentionally crafted adversarial examples for deep neural networks were first described by Christian Szegedy et al. in 2013 [?]. They conclude, that although (deep) neural networks are thought to achieve high generalization, they show discontinuities in their input-output mappings, which enable the existence of adversarial inputs. The authors also propose the L-BFGS method for finding adversarial examples, and suggest using this method for hard negative mining and generating valuable training data.

Warren He et al. examined decision boundaries around adversarial images. They introduced an adversarial attack method able to evade common adversarial defense techniques, like adversarial training. Furthermore, He et al. found examining the distance of input samples from decision boundaries could reveal differences between adversarial and benign inputs [?]. However, they did not visualize the decision boundary of the networks they trained in form of class-activation maps nor heat-maps.

Chattopadhyay, Sarkar et al. proposed the method of visualizing, what areas of an image appear especially important for image classification to convolutional neural networks used in this work. Due to the fact, that spatial information is preserved in the convolutional layers of CNNs, and the ability to calculate the gradient between outputs of the last convolutional layer and the desired output, weights can be assigned to all parts of the input image. Weighing the localizable activations in the last convolutional layer of a CNN results in a map, that highlights the areas of greatest importance for the made prediction [?].

CHAPTER 4

Methodology

This chapter will in describe how the experiment performed in this work was performed in detail.
The effects of adversarial training on the decision boundary of CNNs is investigated in an experimental-interpretative way. After an initial training with the original data, several iterations of the following procedure are performed. Adversary data are created for the trained network and combined with the original training data, for another run of training of the classifier. During the iterations of this procedure, relevant data like accuracy on both the test set and the adversarial test set before and after adversary training are collected. Furthermore, heatmaps that highlight areas of highest importance for the network's decision are created. For a distinction of the effects of adversarial training from the effects of a greater amount of regular training, a control-experiment using normal-distributed gaussian noise instead of adversarial data is performed. Finally, that data is interpreted.

4.1 Training Data

The German Traffic Sign Recognition Benchmark is used as original training data. The set consists of 39209 labeled 64x64 images for training, and 12630 for validation. The images are distributed unevenly among the classes, and contain little distortions.

4.1.1 Neural Network Architectures

Several architectures of CNNs are considered, from the shallow letnet-5 up to a version of the Imagenet, the ResNet50 having 50 layers. The varying depths may have effects on the vulnerability for adversarial examples, and hence also on the influence adversarial training has on the decision boundary. The networks are modeled and trained using tensorflow and keras in python.

4.1.2 Adversarial Input Generation

From the discovery of adversarial examples for deep neural networks in 2014, dozens of methods for finding adversarial inputs for neural networks were developed.

Methods

For computational cost reasons, so far only the Fast Gradient Sign Method was examined. Images generated with this method sometimes have perceptible artifacts, thus this method may not be the best for examining the effects of adversary training.

Libraries

So far, the most convincing adversarial images could be generated with IBM's adversarial-robustness-toolkit. For different attacks, Google's cleverhans and foolbox will be considered as well.

4.2 Experiment

For the examination of the effects of adversarial training on the decision boundary, a CNN is first trained with original input data. For later comparison, gradient based heatmaps, that highlight areas in the original inputs that were contributing to the resulting prediction the most are created. Given a trained network, adversarial inputs are generated, and used for another run of training the network. This process is repeated, until adversary inputs can be recognized by the CNN with a certain confidence before the adversarial training. A comparison of the heatmaps, and thus the features that were most important for the decision of the neural network should allow some conclusions about the effects of adversarial training on the decision boundary of neural networks. Likewise, the appearance of adversary images, that could successfully fool the classifier can expose characteristics of the decision boundary, as the decision boundary lies between the original input and the tempered one.

4.3 Frameworks

The current implementation of the experiment is done in python 3.5 and uses several Frameworks, of which some are listed below:

Name	Version
Numpy	1.17.3
Tensorflow	1.14.0
Tensorflow-GPU	1.14.0
Keras	2.3.1
Adversarial-Robustness-Toolkit	1.0.1
Cleverhans	3.0.1
Foolbox	2.3.0

CHAPTER 5

Results

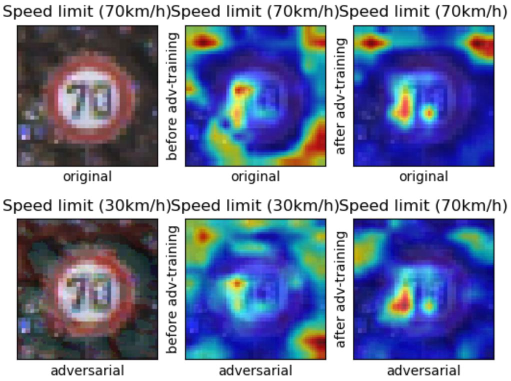


Figure 5.1: First result, first training run of alexnet using FGSM-adversarials

Figure **??** depicts an example of results produced by the current implementation of the experiment. In this example, an alexnet was trained and fooled using FGSM. The original input (top left) is correctly classified (70km/h), the adversary input below gets misclassified (30km/h). The heatmaps in the middle show, which areas of the image were most important for the classifier's decision before adversary training. The right column shows the areas of most importance for the decision after adversarial training. After adversarial training, the adversary input gets classified correctly, and the heated area at the bottom right corner of the image disappeared. This could be interpreted as shift of "attention" to areas that seem more important and reliable for classification.

CHAPTER 6

Outline of planned calculations

Currently, the experiment could only be done for the lenet-5 and the alexnet, using FGSM, as generating adversarial inputs for deeper architectures requires more computational power. Also, the results presented before only used one run of training and adversarial training, I expect clearer results after 10-100 iterations. The planned calculations include training lenet-5, alexnet, vgg19 and resnet50, and generating adversarial training data for them using at least FGSM and one stronger method like deepfool. For a better resolution of the heatmaps, an upscale of the input images to at least 224x224 might be necessary, resulting in an increase of the computational costs for the whole experiment.

CHAPTER 7

Introduction to LaTeX

Since LaTeX is widely used in academia and industry, there exists a plethora of freely accessible introductions to the language. Reading through the guide at `https://en.wikibooks.org/wiki/LaTeX` serves as a comprehensive overview for most of the functionality and is highly recommended before starting with a thesis in LaTeX.

7.1 Installation

A full LaTeX distribution consists of not only of the binaries that convert the source files to the typeset documents, but also of a wide range of packages and their documentation. Depending on the operating system, different implementations are available as shown in Table **??**. **Due to the large amount of packages that are in everyday use and due to their high interdependence, it is paramount to keep the installed distribution up to date.** Otherwise, obscure errors and tedious debugging ensue.

7.2 Editors

A multitude of TeX editors are available differing in their editing models, their supported operating systems and their feature sets. A comprehensive overview of editors can be

Distribution	Unix	Windows	MacOS
TeX Live	**yes**	yes	(yes)
MacTeX	no	no	**yes**
MikTeX	no	**yes**	no

Table 7.1: TeX/LaTeX distributions for different operating systems. Recomended choice in **bold**.

	Description
1	Scan for refs, toc/lof/lot/loa items and cites
2	Build the bibliography
3	Link refs and build the toc/lof/lot/loa
4	Link the bibliography
5	Build the glossary
6	Build the acronyms
7	Build the index
8	Link the glossary, acronyms, and the index
9	Link the bookmarks
	Command
1	`pdflatex.exe example`
2	`bibtex.exe example`
3	`pdflatex.exe example`
4	`pdflatex.exe example`

5	`makeindex.exe -t example.glg -s example.ist`
	`-o example.gls example.glo`

6	`makeindex.exe -t example.alg -s example.ist`
	`-o example.acr example.acn`

7	`makeindex.exe -t example.ilg -o example.ind example.idx`
8	`pdflatex.exe example`
9	`pdflatex.exe example`

Table 7.2: Compilation steps for this document. The following abbreviations were used: table of contents (toc), list of figures (lof), list of tables (lot), list of algorithms (loa).

found at the Wikipedia page `https://en.wikipedia.org/wiki/Comparison_of_TeX_editors`. TeXstudio (`http://texstudio.sourceforge.net/`) is recommended.

7.3 Compilation

Modern editors usually provide the compilation programs to generate Portable Document Format (PDF) documents and for most LaTeX source files, this is sufficient. More advanced LaTeX functionality, such as glossaries and bibliographies, needs additional compilation steps, however. It is also possible that errors in the compilation process invalidate intermediate files and force subsequent compilation runs to fail. It is advisable to delete intermediate files (`.aux`, `.bbl`, etc.), if errors occur and persist. All files that are not generated by the user are automatically regenerated. To compile the current document, the steps as shown in Table **??** have to be taken.

7.4 Basic Functionality

In this section, various examples are given of the fundamental building blocks used in a thesis. Many LaTeX commands have a rich set of options that can be supplied as optional arguments. The documentation of each command should be consulted to get an impression of the full spectrum of its functionality.

7.4.1 Floats

Two main categories of page elements can be differentiated in the usual LaTeX workflow: *(i)* the main stream of text and *(ii)* floating containers that are positioned at convenient positions throughout the document. In most cases, tables, plots, and images are put into such containers since they are usually positioned at the top or bottom of pages. These are realized by the two environments `figure` and `table`, which also provide functionality for cross-referencing (see Table **??** and Figure **??**) and the generation of corresponding entries in the list of figures and the list of tables. Note that these environments solely act as containers and can be assigned arbitrary content.

7.4.2 Tables

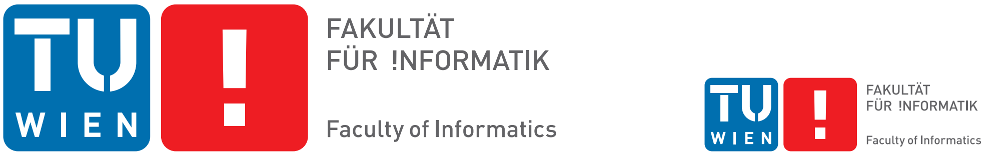
A table in LaTeX is created by using a `tabular` environment or any of its extensions, e.g., `tabularx`. The commands `\multirow` and `\multicolumn` allow table elements to span multiple rows and columns.

Position		
Group	Abbrev	Name
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucus Radebe
	DC	Michael Duburry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Table 7.3: Adapted example from `https://en.wikibooks.org/wiki/LaTeX/Tables`. This example uses rules specific to the `booktabs` package and employs the multi-row functionality of the `multirow` package.

7.4.3 Images

An image is added to a document via the `\includegraphics` command as shown in Figure **??**. The `\subcaption` command can be used to reference subfigures, such as Figure **??** and **??**.



(a) The header logo at text width.

(b) The header logo at half the text width.

Figure 7.1: The header logo at different sizes.

7.4.4 Mathematical Expressions

One of the original motivation to create the TeX system was the need for mathematical typesetting. To this day, LaTeX is the preferred system to write math-heavy documents and a wide variety of functions aids the author in this task. A mathematical expression can be inserted inline as $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ outside of the text stream as

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

or as numbered equation with

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}. \tag{7.1}$$

7.4.5 Pseudo Code

The presentation of algorithms can be achieved with various packages, such as `algorithmic`, `algorithm2e`, `algorithmicx`, or `algpseudocode`. See `https://tex.stackexchange.com/questions/229355` for an overview. An example of the use of the `alogrithm2e` package is given with Algorithm **??**.

7.5 Bibliography

The referencing of prior work is a fundamental requirement of academic writing and well supported by LaTeX. The BibTeX reference management software is the most commonly used system for this purpose. Using the `\cite` command, it is possible to reference entries in a `.bib` file out of the text stream, e.g., as [**?**]. The generation of the formatted bibliography needs a separate execution of `bibtex.exe` (see Table **??**).

Algorithm 7.1: Gauss-Seidel

```
Input: A scalar $\epsilon$, a matrix $\mathbf{A} = (a_{ij})$, a vector $\vec{b}$, and an initial vector $\vec{x}^{(0)}$
Output: $\vec{x}^{(n)}$ with $\mathbf{A}\vec{x}^{(n)} \approx \vec{b}$
for $k \leftarrow 1$ to maximum iterations do
    for $i \leftarrow 1$ to $n$ do
        $x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)} \right)$;
    end
    if $|\vec{x}^{(k)} - \vec{x}^{(k-1)}| < \epsilon$ then
        break for;
    end
end
return $\vec{x}^{(k)}$;
```

7.6 Table of Contents

The table of contents is automatically built by successive runs of the compilation, e.g., of `pdflatex.exe`. The command `\setsecnumdepth` allows the specification of the depth of the table of contents and additional entries can be added via `\addcontentsline`. The starred versions of the sectioning commands, i.e., `\chapter*`, `\section*`, etc., remove the corresponding entry from the table of contents.

7.7 Acronyms / Glossary / Index

The list of acronyms, the glossary, and the index need to be built with a separate execution of `makeindex` (see Table **??**). Acronyms have to be specified with `\newacronym` while glossary entries use `\newglossaryentry`. Both are then used in the document content with one of the variants of `\gls`, such as `\Gls`, `\glspl`, or `\Glspl`. Index items are simply generated by placing `\index{`⟨*entry*⟩`}` next to all the words that correspond to the index entry ⟨*entry*⟩. Note that many enhancements exist for these functionalities and the documentation of the `makeindex` and the `glossaries` packages should be consulted.

7.8 Tips

Since TEX and its successors do not employ a What You See Is What You Get (WYSIWYG) editing scheme, several guidelines improve the readability of the source content:

- Each sentence in the source text should start with a new line. This helps not only the user navigation through the text, but also enables revision control systems (e.g. Subversion (SVN), Git) to show the exact changes authored by different users. Paragraphs are separated by one (or more) empty lines.

- Environments, which are defined by a matching pair of `\begin{name}` and `\end{name}`, can be indented by whitespace to show their hierarchical structure.

- In most cases, the explicit use of whitespace (e.g. `\hspace{4em}` or `\vspace{1.5cm}`) violates typographic guidelines and rules. Explicit formatting should only be employed as a last resort and, most likely, better ways to achieve the desired layout can be found by a quick web search.

- The use of bold or italic text is generally not supported by typographic considerations and the semantically meaningful `\emph{...}` should be used.

The predominant application of the LaTeX system is the generation of PDF files via the PdfLaTeX binaries. In the current version of PdfLaTeX, it is possible that absolute file paths and user account names are embedded in the final PDF document. While this poses only a minor security issue for all documents, it is highly problematic for double blind reviews. The process shown in Table **??** can be employed to strip all private information from the final PDF document.

	Command
1	Rename the PDF document `final.pdf` to `final.ps`.

2	Execute the following command:
	`ps2pdf -dPDFSETTINGS#/prepress ^`
	` -dCompatibilityLevel#1.4 ^`
	` -dAutoFilterColorImages#false ^`
	` -dAutoFilterGrayImages#false ^`
	` -dColorImageFilter#/FlateEncode ^`
	` -dGrayImageFilter#/FlateEncode ^`
	` -dMonoImageFilter#/FlateEncode ^`
	` -dDownsampleColorImages#false ^`
	` -dDownsampleGrayImages#false ^`
	` final.ps final.pdf`

On Unix-based systems, replace # with = and ^ with \.

Table 7.4: Anonymization of PDF documents.

7.9 Resources

7.9.1 Useful Links

In the following, a listing of useful web resources is given.

`https://en.wikibooks.org/wiki/LaTeX` An extensive wiki-based guide to LaTeX.

`http://www.tex.ac.uk/faq` A (huge) set of Frequently Asked Questions (FAQ) about TEX and LATEX.

`https://tex.stackexchange.com/` The definitive user forum for non-trivial LATEX-related questions and answers.

7.9.2 Comprehensive TeX Archive Network (CTAN)

The CTAN is the official repository for all TEX related material. It can be accessed via `https://www.ctan.org/` and hosts (among other things) a huge variety of packages that provide extended functionality for TEX and its successors. Note that most packages contain PDF documentation that can be directly accessed via CTAN.

In the following, a short, non-exhaustive list of relevant CTAN-hosted packages is given together with their relative path.

algorithm2e Functionality for writing pseudo code.

amsmath Enhanced functionality for typesetting mathematical expressions.

amssymb Provides a multitude of mathematical symbols.

booktabs Improved typesetting of tables.

enumitem User control over the layout of lists (`itemize`, `enumerate`, `description`).

fontenc Determines font encoding of the output.

glossaries Create glossaries and list of acronyms.

graphicx Insert images into the document.

inputenc Determines encoding of the input.

l2tabu A description of bad practices when using LATEX.

mathtools Further extension of mathematical typesetting.

memoir The document class on upon which the `vutinfth` document class is based.

multirow Allows table elements to span several rows.

pgfplots Function plot drawings.

pgf/TikZ Creating graphics inside LATEX documents.

subcaption Allows the use of subfigures and enables their referencing.

symbols/comprehensive A listing of around 5000 symbols that can be used with LATEX.

voss-mathmode A comprehensive overview of typesetting mathematics in LATEX.

xcolor Allows the definition and use of colors.

List of Figures

List of Tables

List of Algorithms