Few-Shot Learning

Toward Deep Learning and Convolutional Neural Network

Bachelor Thesis

by

Milad Navidizadeh
2953248

in fulfillment of requirements for degree
Bachelor of Science (B.Sc.)

submitted to
Rheinische Friedrich-Wilhelms-Universität Bonn
Institute of Computer Science III
Bachlor Thesis for Information Systems and Artificial Intelligence

in degree course
Informatik (B.Sc.)

First Supervisor: Prof. Dr. Stefan Wrobel
University of Bonn

Second Supervisor: Prof. Dr. Christian Bauckhage
University of Bonn

Bonn, December 27, 2019

Acknowledgement

A special thanks goes out to Dr. rer. nat. Wolfgang Hansmann, for maintaining the guideline up to version 0.4 and Prof. Dr.-Ing. André Miede for his work on an extended template for `classicthesis` that I learned (and borrowed) a lot from.

» *It has been a long journey to this moment.* « Sidney Poitier, 1964

Contents

1 Motivation

Nowadays machine learning and deep learning have become a distinguished approach for visual recognition tasks and has achieved great success in this process. However, they seek a large amount of labeled data to learn. Providing this amount of labeled data not only will bring much effort along but also will occupy a huge size of storage and seek large storage. In contrast, humans are very good in visual recognition so that, they can learn with one [1] or few [2] examples. Imagine one kid who can recognize a lion in a picture after looking a few pictures of lions as an example. We want to simulate and apply this human's ability to the deep learning and make them learn with few examples, with desirable accuracy. In this bachelor thesis, we concern ourselves with few-shot learning in deep learning. We aim to learn and train a model when there are few labeled examples obtainable. We approach to generate artificial examples from a few available labeled examples and enalrge our dataset artificially. This technique known as data augmentation. These artificial labeled examples aid us to learn better with more accuracy and prevent overfitting. Data augmentation is our focus in this work to achieve few-shot learning and prevent overfitting. In this thesis, we will introduce different well-known methods of data augmentation. The first purpose will be to discover if and how far data augmentation can improve the learning process and accuracy. The second step will be to compare their accuracy. In the end, we aim to discover the potential possibility of combining the different methods of data augmentation to increase accuracy and reduce error-rate and improve the learning process. We will focus on visual recognition tasks and their classification. Additionally, we will concentrate on the implementation of various methods of data augmentation for convolutional neural networks

[1] This is known as one-shot learning in deep learning and represents the scenario when there is one instance of each class in training-set to learn.
[2] This is known as few-shot learning in deep learning and represents the scenario when there are just few instances of each class in training-set to learn.

2 Introduction

Neural networks can possibly contain multiple non-linear layers and this makes them very expressive models that can learn very complicated relationships between their inputs and outputs. With even limited input data, neural networks can discover and learn many relations from the data, however, sometimes the discovered and learned relations do not exist or just consist of redundant information and relations. Redudant relation can potentially arise of data-noise or lack of data-generalization. Non exist relations can potentially emerge from lack of enough data. These phenomena known as *overffiting* in deep learning. In other words, learning with few labeled examples or noisy data causes overfitting. Overffiting cause low accuracy and high error-rate. Hence, we approach to propagation of artificial labeled examples from a few given examples to prevent overfitting and reduce the error rate and increase the accuracy.

As we mentioned aquiring a huge labeled dataset is expensive and seeks much effort and time. Therefore we aim to generate artificial example from few obtainable labeled examples. In other words, we augment our data and this strategy is known as data augmentation. There are a few well-known methods for data augmentation. We aim to introduce them in this thesis. Besides we will implement these methods to compare their efficiency and capability. These methods are as follow:

- **Label Preserving Transformation** 5.1
- **Elastic Distrotion** 5.2
- **Storke Wraping** 5.3
- **Bayesian Model** 5.4
- **Manifold Approach** 5.5

3 Data Representation

Here should be an Introdution of chapter and introduce the data

3.1 MNIST

The MNIST dataset (Modified National Institute of Standards and Technology) is a large handwritten digits dataset, provided by Yann Le Cun, derived from NIST Special Database 19 [STA].

The MSNIT dataset consists of 60,000 train- and 10,000 test-images and each image is grayscale with 28×28 pixels. It has 10 classes that represent $0-9$ digits and data is fairly splitted between classes [LeC]. MNIST is one of the most popular datasets for deep learning because of the not too high complexity and compatibility with almost all deep learning models. Hence many papers attempted to reach a low error-rate on this dataset. One of them manages to reduce the error-rate on the MNIST by up to 0.23% [Dan12]. You can find the information about the dataset at table 1 and figure 1 shows an example of the dataset.

Figure 1: *7 examples per class of MNIST dataset, merged in one image [Ath]*

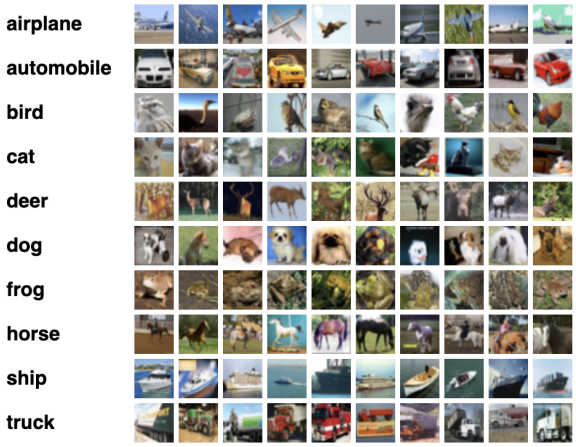

Figure 2: *10 examples per class of CIFAR-10 dataset, merged in one image [Kri]*

Table 1: *Structure and organization of the datasets.*

Dataset	NO. Classes	NO. Train	NO. Test	Size (pixel)	NO. Channel
MNIST	10	60,000	10,000	28×28	1 (Grayscale)
EMNIST	26				
CIFAR-10	10	50,000	10,000	28×28	3 (RGB)
CIFAR100	100				

3.2 EMNIST

blablabla

3.3 CIFAR-10

The CIFAR-10 (Canadian Institute for Advanced Research) , collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton is a subset from 80 million tiny images dataset [Uni].

The dataset consists of 60,000 RGB with 32×32 pixels images, which are divided to the 50,000 train and 10,000 test datasets. As the name makes it clear the CIFAR-10 contains 10 classes ([plane, car, bird, cat, deer, dog, frog, horse, ship, truck]) [Kri]. One of the lowest reported error-rate with a convolutional neural network managed to achieve 2.56% [Ter17]. You can find the information about the dataset at table 1 and figure 2 shows an example of the dataset.

3.4 CIFAR100

blablabla

4 Neural Network

4.1 Introduction

4.2 Convolutional Neural Network

4.3 Overffiting

5 Data Augmentaion

Here should be Introdution for data augmenation

5.1 Label Preserving Transformation

One of the most in common method to enlarge the dataset artificially is the label preserving transformation. This method provides the possibility to generate artificial data with non-heavy computation. The advantage of a very little computation aids us to save storage. In other words, it wouldn't be required to save the generated data on a storage and the data can always be enlarged artificially in a short time and with a little computation. As the method's name makes it clear, this method aims to generate new data from a single data with the same label. We explain the method and its approach for image datasets because as we mentioned our focus is image datasets.

This method consists of generating image translations and horizontal reflectins. Image translations mean, random patches from the original images. The size of the patches are smaller than the size of the original images. We will extract all pssobile translations (all possible patches which fit in the image) from our image. These extracted translations (extracted random patches) and their horizontal reflection will be used for training our network. Given a single channel (gray) n×n image and the translation pathce with size of m×m and $n > m$ then the training dataset will increase by a factor:

$$2 \times (n - m + 1) \times (n - m + 1)$$

In the figur 3 is the process of the translations and reflections of a small image presented.

At test time, the method extracts the patches with the same size. This time the patches will be extracted from 4 corners and center of the original image.

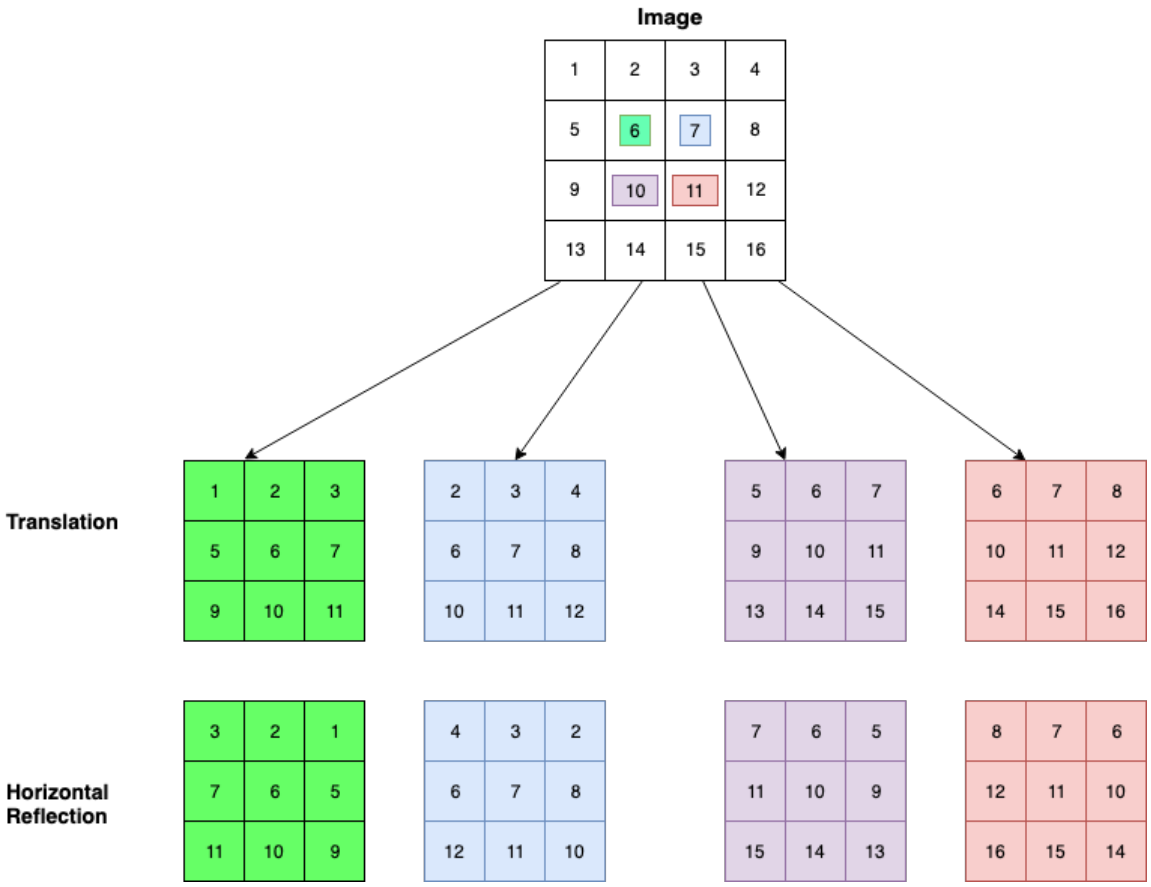

Figure 3: *An example of single channel image with size of* 4×4 *with its translations with size of* 3×3 *patches and their horizontal reflections. The numbers determinate the pixels intensity*

The network predicts on these five patches and their horizontal reflections (10 patches altogether). In the end, the average of the predictions will be our network's prediction.

5.2 Elastic Distrotion

Another well-known method for data augmentation is Elastic distortion. This method as the same as label preserving transformation generates artificial data (images) from a single data (image) but with this difference that this time instead of resizing the image (translation) the pixels inside of the image will be moved and displaced. The intensity of pixels also will be changed with respect to their new places and their neighbors' intensity and their old places and their origin intensity.

As it cleared elastic distortions concerns itself with a displacement of the pixels and an adjustment of the pixels intensity. This approach is known as interpolation.

There are a few schemes like (nearest neighbor-, bicubic-, spline- and bilinear interpolation) to reach the goal. Bilinear interpolation is one of the simplest with a desirable result. Hence we use bilinear interpolation. We will explain the process below.

We will start with pixels displacement. For instance if $\Delta x(x,y) = \alpha x$ and $\Delta y(x,y) = \alpha y$, this means that each pixels will be moved αx in the direction of x-axis and αy in the direction of y-axis. α would be our scale parameter and since α can be a non-integer value, interpolation is necessary and as we mentioned we use the bilinear interpolation. In the next step after pixels displacement, the intensity of the pixels in the new locations should be adjusted concerning the intensity of the neighbors' pixels in the original image (origin square). Hence bilinear interpolation aids us to reach this target. The bilinear interpolation interpolates the moved pixel horizontally. Then it interpolates the pixel vertically with respect to yielded values from horizontal interpolations. We will show and summarize the process formally in below.

Definition 1: Given p' the pixel which we want to dispalacment it with $\Delta x(x,y) = \alpha x$ and $\Delta y(x,y) = \alpha y$ and $p_{(x,y)}, p_{(x+1,y)}, p_{(x,y-1)}, p_{(x+1,y-1)}$ are the neighbors (on origin square) of p' in the new location after displacment and $I(p)$ shows the intensity of pixel p. Then the vertical interploation yields:

$$V_1 = I(p_{(x,y)}) + \Big(\Delta x(p', p_{(x,y)}) \times I(p_{(x+1,y)})\Big)$$
$$V_2 = I(p_{(x,y-1)}) + \Big(\Delta x(p', p_{(x,y-1)}) \times I(p_{(x+1,y-1)})\Big)$$

And then the horizontal interploation yields a new intensity for pixel p' after displacment:

$$I(p') = V_1 + \Big(\Delta y(p', p_{(x,y-1)}) \times V_2\Big)$$

To reach elastic deformation or elastic distrotion we approach to generate $\Delta x(x,y) = \alpha x$ and $\Delta y(x,y) = \alpha y$ with $\alpha \times rand(-1,+1)$ since α is image-scale parameter and $rand(-1,+1)$ generate a number between -1 and $+1$ with uniform distribution. After all, the fields Δx and Δy are convolved with a Gaussian filter with standard deviation of σ. The values of σ and α depends on the image size and the image entropy. This process will generate elastic deformed image from orginal image which called elastic distrotion.

5.3 Storke Wraping

5.4 Bayesian Model

5.5 Manifold Approach

6 Bibliography

[Ath] P. Athul. *Medium Handwritten digit recognition using PyTorch*. https://medium.com/@athul929/hand-written-digit-classifier-in-pytorch-42a53e92b63e. Accessed: 2019-12-16.

[Dan12] Juergen Schmidhuber Dan Cireşan Ueli Meier. *Multi-column Deep Neural Networks for Image Classification*. 2012. arXiv: 1202.2745 [cs.CV].

[Kri] Alex Krizhevsky. *The CIFAR-10 dataset (Canadian Institute for Advanced Research)*. http://www.cs.toronto.edu/~kriz/cifar.html. Accessed: 2019-12-16.

[LeC] Yann LeCun. *exdb THE MNIST DATABASE of handwritten digits*. http://yann.lecun.com/exdb/mnist/. Accessed: 2019-12-16.

[STA] STANDFORD. *NIST National Institute of Standards and Technology*. https://www.nist.gov/data. Accessed: 2019-12-16.

[Ter17] Graham W. Taylor Terrance DeVries. *Improved Regularization of Convolutional Neural Networks with Cutout*. 2017. arXiv: 1708.04552 [cs.CV].

[Uni] New York University. *Visual Dictionary Teaching computers to recognize objects*. http://groups.csail.mit.edu/vision/TinyImages/. Accessed: 2019-12-16.

*

List of Figures

List of Tables

Statement of Authorship

I hereby confirm that the work presented in this bachelor thesis has been performed and interpreted solely by myself except where explicitly identified to the contrary. I declare that I have used no other sources and aids other than those indicated. This work has not been submitted elsewhere in any other form for the fulfilment of any other degree or qualification.

Bonn, December 27, 2019

Milad Navidizadeh