Book of Abstracts

Deep Learning Course 2016
Chalmers University of Technology

Olof Mogren, Mikael Kågebäck, Fredrik Johansson
[mogren,kageback,frejohk]@chalmers.se

May 27, 2016

1 Introduction

This is the book of abstracts for the student projects in the Deep Learning course at Chalmers University of Technology spring 2016.

Contents

2 Tuesday, May 31st

2.1 Classification of Human Facial Expression

Authors: Arvid Nilsson and Jonas Karlsson
Keywords: CNNs, ResNets, Tensorflow, Image Processing
Abstract:
As interaction with artificial intelligence based systems becomes more common, the need for the system to understand the context in which instructions or queries are given increases. In this project we aim to design a deep neural net that can classify the facial expression of a person into one out of seven expressions using supervised learning. To achieve this, a set of convolutional neural nets (CNNs) are implemented in Python using the deep learning framework Tensorflow. Two plain CNNs and three residual nets are evaluated and the effects of regularization as well as several methods of data augmentation are explored using grid search. The training is done using the Adam optimizer in combination with early stopping based on the validation error.

2.2 Sentiment Classification

Authors: Raphael Konstantinou and Sebastian Eriksson
Keywords: LSTMs, Attention Mechanisms, Encoder-Decoder, Sentiment Analysis
Abstract:
Information can be presented in many different ways through text. The writing style e.g. formal or informal, tell us something more than the information it contains. Text often contain emotions, ideas and feelings. As humans, we are very good at distinguishing the different nuances in text. For machines however, this is difficult as the way something is said or the emotive state of the writer may convey a completely different message. For instance, sarcasm in text completely changes its conceptual polarity.

The idea is to improve an existing model for the sentiment classification task (binary classification) that uses LSTMs, testing different hyper parameters and implementing an attention mechanism. The model was bench marked on the IMDB and UMICH data sets. The attention mechanism, assumes that each word in the input text has different importance. In the approach we used was to change the logistic regression network to a sequence to sequence network. The input language to the encoder is text which is encoded and then decoded to a binary language. The idea is that the binary language represents the sentiment class of the text. This was done by using

a LSTM cell as encoder and then decoding the output by using an attention decoder which looks at all of the encoder states.

2.3 Using deep convolutional networks to make dead time more alive

Authors: Josefin Ondrus and Gabriel Andersson
Keywords: CNNs, Theano, Lasagne, Image Classification, Localization, Image Processing
Abstract:
We present a proof of concept model intended to be able to classify and localize objects in images residing in an online social networking service. This by constructing a Convolutional Neural Network inspired by the YOLO network, giving the network a head that handles both multi-regression and classification in a fast manner. Training the network with the Pascal VOC 2007 dataset and then testing it on images from the Facebook Graph api was planned to produce a suitable result. Constructing the network using the frameworks Lasagne and Theano in Python gave some unforeseen complications which resulted in limitations of reaching the intended goal. Therefore the project was narrowed down to not include the Graph api and instead focus on solving the problem with images as is.

2.4 Multi-Word Expression Detection

Authors: David Alfter and Luis Nieto Piña
Keywords: CNNs, RNNs, NLP
Abstract:
The task which we are trying to solve is the classification of multi-word expressions (MWE) into literal and non-literal meanings. I.e., given an MWE like *work out*, can we train a model to discriminate instances with compositional meaning (*I work out of the office today.*) from non-compositional instances (*I work out at the gym to improve my strength.*) based only on the context in which such MWE appear? To do this, we propose to use two different deep neural models separately: a convolutional neural network and a recurrent neural network. The motivation for doing this is to be able to compare their results and try to understand any differences in performance in relation to each architecture's own characteristics. Are we right to think that recurrent networks are better suited to work with language data?

3 Thursday, June 2nd

3.1 Playing Solitaire using CNN and Deep Reinforcement Learning

Authors: Robert Nyquist and Daniel Pettersson
Keywords: Deep Reinforcement Learning, CNNs, Q-Learning
Abstract:
Our project has been to train a network to play the solitaire game klondike (3 turn). By using a combination of Deep Neural Networks and Reinforcement Learning we were hoping to get a network that could solve a game of klondike. The model is a convolutional neural network trained with Q-learning. The input is the image of the board with cards and output is a move that (with highest probability) will move the game towards the highest reward sum state.

We will present what results we got and what methods might improve the performance of the network.

3.2 Spotting distracted drivers

Authors: Emilio Jorge and Benjamin Lindberg
Keywords: CNNs, Image Processing, Limiting Overfitting, Transfer Learning, Data Augmentation
Abstract:
Distracted drivers has long been an hazard when it comes to traffic safety. One solution to this problem could be to use computer vision to detect when drivers are not paying sufficient attention. In this project we attempt to classify drivers by type of distraction. An issue with this task has been that the available dataset is quite small leading to severe overfitting. The project has thus consisted of methods to combat overfitting. One approach has been to use a small convolutional network. Another approach has been to use a larger network that is combined with pretraining on another dataset. Yet another important aspect of the project has been different kinds of data-augmentation to try to reduce the overfitting.

3.3 Image captioning

Authors: Luca Caltagirone and Sina Torabi
Keywords: CNNs, ResNets, LSTMs, GRUs, Torch, Image Processing, NLP, Multi-Modal Models
Abstract:

One of the most remarkable and fascinating aspects of human visual systems is its ability to understand and interpret a complex environment in a very short time. Such an ability is absolutely essential to, almost, all intelligent machines, since a true machine intelligence should exhibits behaviors, at least, as skillful and flexible as humans do. Consequently, developing visual recognition systems is one of the main steps towards true AI. Therefore, in this project, a multi-modal model that can caption images will be investigated.

Image captioning systems are end-to-end systems that can take an image as input and generate a caption for that image as output. In order to build such a system, one can combine different networks, e.g. a convolutional neural network for visual system and a recurrent neural network for natural language, to approach image captioning. Such systems have become more popular recently due to several reasons such as powerful convolutional networks for feature extraction and efficiency and performance of recurrent neural networks in language modelling.

As for the starting point, an implementation of image captioning system by Andrej Karpathy has been used for this course. This system has been implemented in Lua programming language using Torch deep learning library. In this work, Karpathy combined a VGGnet and LSTM units, and trained it on MSCOCO dataset.

In this project, some modifications have been made in the current implementation. In order to improve the performance of the system, instead of a VGGnet, a ResNet convolutional neural network (pre-trained on ImageNet classification task) has been used. It turned out that by using a better convolutional neural network, one can improve the performance significantly. The implemented network achieved a CIDEr score of 0.970 which places it in the second place of the Microsoft COCO leaderboard. Moreover, a system using GRU (Gated Recurrent Unit) instead of LSTM is currently training so, at the moment, there is no information available on its performance.

3.4 Recognition of Handwritten Mathematical Expressions

Authors: Christian Ågren and Alexander Ågren
Keywords: CNNs, LSTMs, Multi-Modal Models
Abstract:
Abstract—In our work we have studied the problem of recognizing handwritten mathematical expressions from digital ink data using neural networks. The problem of recognizing mathematical expressions introduces extra com-

plexity compared to recognizing regular handwritten text. The structure of an expression as well as 2-dimensional relations between symbols need to be learned, together with correct segmentation and classification of separate symbols.

This project takes a few different approaches to this problem, using both Convolutional and Recurrent Neural Networks, and tries to evaluate the different hardships and benefits of using these. As a result this paper suggests some important aspects to consider when approaching these kinds of problems, as well as some performance results implementing this.

3.5 Deep reinforcement learning for autonomous highway driving

Authors: Carl-Johan Hoel
Keywords: Deep Reinforcement Learning, Q-Learning, Autonomous Driving
Abstract:
Deep reinforced learning has been used to create a decision making algorithm for a simple case of autonomous driving. Q-learning was used, where the maximum future discounted reward given the current state (Q-function), was represented by a neural network, which outputs the Q-value for each possible action. This network was trained in a simulation environment.

The studied scenario was autonomous highway driving with some surrounding traffic. The input to the algorithm was features expected to be delivered by a sensor fusion function, such as ego lane, ego speed, relative position and velocity of surrounding vehicles etc. The action space of the decision algorithm was the choice of changing left, right or staying in the current lane, and increasing, decreasing or keeping the current speed.

3.6 Distracted Driver Detection

Authors: Florian Schäfer and Mats Uddgård
Keywords: CNNs, Image Classification, Transfer Learning
Abstract:
Of all traffic accidents there are approximately one out of five which is caused by distracted drivers. If these situations could be avoided it would result in fewer deaths in traffic. The idea was to reliably identify if a driver is currently paying attention or is inattentive to the current driving situation based on a set of 2D-images taken of the driver in ten different attention classes based on their current pose and activity.

Since we have very limited computational resources available, we had to focus on building a network that produces the best possible results without requiring much computational effort from our side.

Our solution to this was to use the following two-part architecture: The first part consists of a VGG-16-based CNN similar to the one proposed in Google's DeepPose paper. For this we rely on a VGG-16 pretrained on the ImageNet dataset since training it ourselves would be computationally impossible. With the poses predicted from the first part, the second part which is a basic fully connected NN will be able to predict one of the ten possible attention classes. Since this network only uses the poses which are of much lower dimension than the images, we are able to train this network using common backpropagation.