An investigation into feedback and representations in deep vision networks

Martin Dimitrov
2247164D

December 20, 2019

1 Status report

1.1 Proposal

1.1.1 Motivation

Traditional convolutional neural networks use only feedforward connections. This can be extended to use feedback connections so that they can be a better approximation to human vision. Feedback is a potential mechanism to improve obtained results in visual processing tasks using neural networks. An example of such a task is automatic segmentation.

1.1.2 Aims

The aim of this project is to investigate different mechanisms for using cognitive feedback in convolutional neural networks in computer vision. In this way, feedback architectures can be explored with a view to attempt to build a network which automatically segments the classified object by suppressing irrelevant information or enhancing important parts of an image.

1.2 Progress

- Language and framework chosen - the project will be implemented in Python using PyTorch
- Background research on feedback mechanisms for neural networks conducted
- Wrote a draft literature review which summarizes various methods of feedback in neural networks
- Implemented classifier and autoencoder, evaluated both on the FashionMNIST dataset
- Combined the above classifier and autoencoder into a single network by connecting the flattened layer of the autoencoder to the fully connected layers of the classifier
- Formed a joint loss function using the sum of the classifier loss and the autoencoder loss
- Trained the network individually with the autoencoder loss, the classifier loss individually as well as the joint loss and evaluated loss and accuracy of the FashionMNIST

- Scaled the individual losses by subtacting the mean and dividing by the standard deviation
- Evaluated the model using the sum of the scaled losses
- Iteratively subtracted output from original image and used the result as the input for the network for the next iteration

1.3 Problems and risks

1.3.1 Problems

- Limited GPU space on Colab
- Not enough previous practical experience with Deep Learning
- Not having enough theoretical knowledge about complex architectures prior to the start of the project
- Difficulty in translating abstract to code when starting the project

1.3.2 Risks

- Not discovering a suitable architecture for this task
- Proposed architecture not performing as well on some datasets. Mitigation: do a background research on the datasets most commonly used for similar network architectures

1.4 Plan

Semester 2

- Week 1-2 Use pretrained classifier weights
- Week 3-5 Investigate NARX, LSTMs and Deep Predictive Coding,
- Week 6-8 Implement Grad-CAM for these architectures
- Week 7-9 Finish writing dissertation

1.5 Ethics and data

This project does not involve human subjects or data. No approval required.