Second Assignment (Deep Learning based on CNNs)

TUTORS: MIRELA POPA

DEADLINE: 25/05/2021 AT 11.59 PM

DESCRIPTION

In this assignment you will get familiarized with deep learning models, namely CNNs and you will develop a solution for a classification problem, namely the emotion recognition task. Given a dataset of images with different facial expressions labeled as one of the basic emotions (e.g. anger, happiness, surprise, sadness, fear, disgust or neutral), you will need to implement a model based on CNNs, which gets trained on a subset of the dataset and is tested on another set. You will analyze the role and effect of different architectures and parameters of your CNN model (e.g. number of layers, filters, pooling, normalization, etc.), and report your accuracy and observations in a report. Optionally, you can compare your results to the ones achieved using a pre-trained model (e.g. VGG), but the scope of this assignment is to understand the effect of the different CNN operations on your results.

This assignment is a group assignment and will be done in pairs.

STEPS

- First, you need to select a deep learning framework on which to perform your experiments. I recommend PyTorch (https://pytorch.org/), you will also find on Canvas a short tutorial about it, but you can use a detailed tutorial from the ones available online (e.g. https://www.tutorialspoint.com/pytorch/index.htm).
- Then, you need to select the emotion dataset on which to perform your experiments (e.g. you can use the FER2013 dataset consisting of 34k images, which can be easily downloaded from Kaggle, providing a training and a testing set). You can also use another emotion dataset, but the one I mentioned is small enough to be trained on a CPU (big

datasets might require more powerful resources, such as GPU processing power, which goes beyond the scope of this assignment).

- Next, you need to read, parse and load your dataset and for this task you can use PyTorch dataset functionalities such as torch.utils.data.DataLoader (https://www.tutorialspoint.com/pytorch/pytorch_loading_data.htm), but also available parsers for csv files (format in which the FER dataset is saved) to h5py. Additionally, transformations can be applied on your images for improving their representation (e.g. cropping, contrast, saturation changes).

- Building your CNN model is one the most important issues and you should try different architectures and configurations, including different number of layers (e.g. experiment with max. 10 layers), filters (e.g. check the Conv2D function from torch.nn, which provides many options for configuring CNN parameters). You can use a Sequential model, for which you can specify the types of desired operations after each convolutional layer, including batch normalization and the activation function (e.g. ReLU). After building your model, you need to select a loss function (e.g. cross entropy), and an optimization module, along with a learning rate. Train and test your CNN model using the usual pipeline (check the PyTorch tutorial available on Canvas for a simple example, but also other resources online).

- Please try different architectures and parameters and based on the performed experiments select a final model. The made decisions and the rationale behind them will be explained in the report (e.g. why did you use specific filter sizes and how did it help at improving the accuracy?).

- Check the model accuracy, precision, recall on the test data and report it using several measures (quantitative vs. qualitative), also make sure you use a good testing methodology (e.g. n-fold cross validation or at least split the data into training/testing sets).

- Visualize the learnt filters on different layers to check which is the internal representation of your CNN model (e.g. you can check the model parameters for each convolutional layer and transform the filters into an image). Display the relevant activations for different emotions.

- Formulate your critical reflections about the performed experiment and the lessons learnt (e.g. what is important when constructing a CNN model and how to optimize it?).

- Optionally you can compare your model with a pre-trained one (e.g. VGG), but the goal is not to achieve state-of-the-art results but to learn how to configure a good CNN model.

Deliverables

A) The code and a well-documented readme file with specific details about installation/running procedure, employed parameters.

B) A report of maximum 1200-1500 words consisting of your results, plots and findings and an explanation for your final CNN architecture.

C) Upload everything on Canvas as a zip file, having the following format: Surname1_Surname2_Second_Assignment_CV2021.zip. As this is a group assignment, only one of you needs to upload it.

Grading

A) Code – 1.5 Points

B) Report– 1 Point