Master's Degree in Computer Science, University of Udine

Course of Deep Learning, Academic Year 2021/2022

Assignment - Emotion Recognition with CNNs

Course held by: Prof. Giuseppe Serra, Dr. Beatrice Portelli,
Dr. Giovanni D'Agostino

Dipartimento di Scienze Matematiche, Informatiche e Fisiche

Udine, 02/12/2021

Instructions

You can do this assignment **either alone or in group of 2**. Please submit no more than one submission per group.

Be sure to write your code in **Python**; you can make full use of the libraries seen in class (pytorch, pandas, sklearn, numpy, matplotlib...); please justify the usage of eventual additional libraries you used in your code.

Your source code should be appropriately (but not excessively) commented; your code can be presented as either a **.py** (python source code) or a **.ipynb** (Jupyter Notebook) file. If you have more than one source code file please name them in such a way that it is easy to understand what's inside each file and which file contains the code to run your work.

Alongside your code please submit a **short written description** (preferably in **.pdf**) of what you have done, including used methods, results, discussion, conclusions and eventual references; in case of multiple source code files please summarize the content of each .py/.ipynb file. The report *should not exceed 4 pages of text* (images and tables can be moved at the end of the report and won't be counted in the total amount of pages).

Put both the source code in a **.zip** file named after your student id: "idstudent.zip" for an individual work and "idstudent1_idstudent2.zip" for a work in group of two, e.g. "012345_678901.zip". The .zip file must be uploaded to the appropriate folder of the ELearning page of this course no later than the specified deadline.

DEADLINE: Wednesday December 15th, 2021 (you have time up until midnight).

Assignment

In this assignment you will get familiarized with 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 a testing 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.

First, you need to download the dataset on which you will perform your experiments, which is the FER2013. Such dataset consists in roughly 34k images, which can be easily downloaded from **Kaggle** (kaggle.com/ashishpatel26/facial-expression-recognitionferchallenge), providing enough data for both a training and testing dataset.

Next you have to read, parse and load the dataset in a **PyTorch Dataloader**; additionally (and *optionally*), transformations can be applied on your images for improving their representation (e.g. cropping, contrast, saturation changes).

Once you've done the previous steps you have to design the architecture of your model; building your CNN model is one of the most important issues for this task, 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 we have seen during the practical lessons.

Please try **different architectures and parameters** and then, based on the performed experiments, select a final model. The decisions made 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?); please also add diagrams of the architectures you designed for this task (we suggest using tools like **draw.io** for this). 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?).

Remember that you can optionally can compare your model with a pre-trained one like VGG, but the goal is not to achieve state-of-the-art results but to learn how to configure a good CNN model.

Good luck!