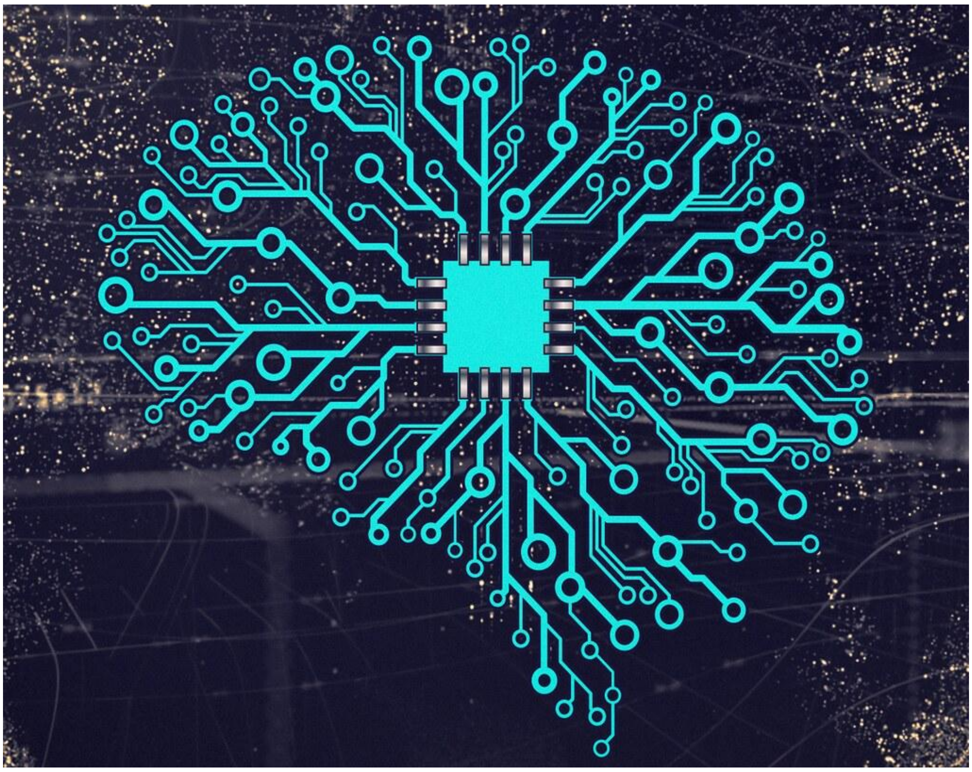


Machine Learning & Content Analytics

- Emotion classification with images -

Instructors:
Papageorgiou Harris
Perakis George

Student:
Kitsou Effranthia Anastasia
p2822118

August 2022
Msc in Business Analytics
AUEB

Contents

PROJECT IDEA

Project Description

This project focuses on emotion detection, building models for robust emotion predictions. The intention of the project is to be the first step of a bigger project to accurately suggest music playlists based on the different emotions.

Data Collection

The Dataset used for this project are a widely used and well known dataset called Fer data. The data consists of several images labeled with the respective emotion and they are pre-split to training validation set and set set. More about the data and the dataset could be found on [“https://www.kaggle.com/datasets/deadskull7/fer2013”].

Dataset Overview

The dataset consist of 34.000 pictures of people while they emote of size 84 x 84 and of a single dimension as they are grayscale. We may see at the dataset overview that in fact most photos are classified as “Happy” this makes also practical sence as we people have a learned habit and tendency to smile when a picture of us are taken no mater the occasion. The dataset has 3 collumns, one for the different emotion code-number, one more collumn for the photo pixels arrays and finally one to classify the observations to the different training and test sets.

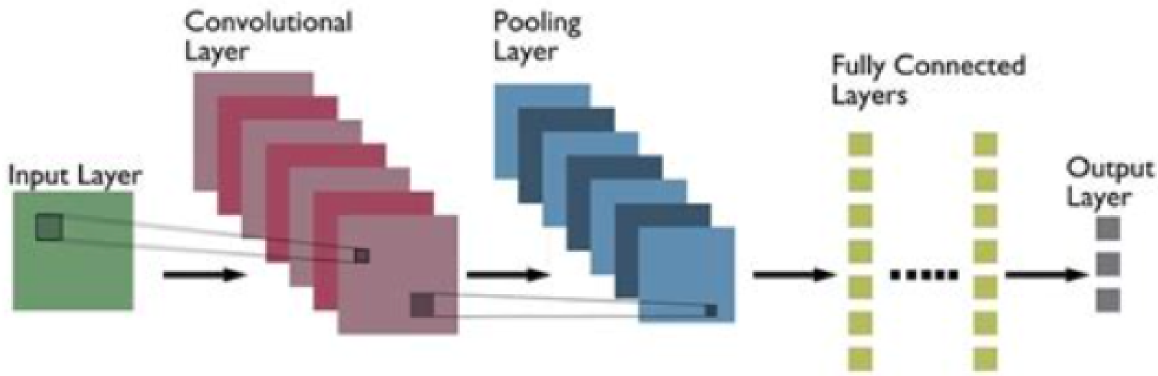
Data Processing

The data for ths project of course needed to be pre-processed as they are actualy consisting of pictures. So first of all after the initial splitting of our data for training, validatation and test purposes, we do proceed to normalize the column containing the pixels that consist the actual photo. The data needed to be normalized and contain only values between zero and one, for this reason we proceed with a division of the whole dataset to 225 as this is the maximum range of pixel values in our dataset.

Then we also need to reshape our dataset in order for the convolutions to procces them correctly. So we have proceeded to create a matrix of shape 48, 48 and depth only 1 as our data are black and white.

Methodology

The models that were used for the Emotion Classification problem are Convolutional Neural Network, as shown below.



The input of the sequential CNN model is a 48-by-48 black and white image. Next, we add a convolution layer to the model. The primary purpose of convolution in case of a CNN is to “learn” from the input pictures and extract features. This learning is conducted in small squares of input data. The size of the squares are determined by the kernel that on the first model is a 3x3 square while on the second it is slightly bigger with a 4x4 square.

As of the activation function, Rectified Linear Unit or ReLU for short is applied here in order for non-linearity to be introduced to our model. This is crucial as in most real world datasets linearity is in fact absent. ReLU is applied per pixel for those that are negative transforming them to zeros for the model to be able to run smoothly.

In both of our models Max pooling is implemented, meaning that only the biggest of numbers within each square is preserved. In more details, we define a spatial neighborhood and take the largest element from the feature map within that chunk of data. Instead of taking the largest element we could also take the average (Average Pooling) or sum of all elements in that square. In practice, Max Pooling has been shown to work better.

Convolution, ReLU and pooling layers are the basic building blocks of any CNN. Together these layers extract the useful features from the images, introduce non-linearity in our network and reduce feature dimension while aiming to make the features somewhat equivariant to scale and translation.

Finally we apply a Flattening Layer. Flattening is the process of converting all the resultant 2 dimensional arrays into a single long continuous linear vector.

Next, we move to the Fully Connected Layer. The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a sigmoid activation function in the output layer. Here we have applied softmax as the emotion classes are multiple. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer. The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

Also, we may see from the summary of the classifier, that the total number of the trainable parameters is 581,657 for the first model and 668,897 for the second. Moreover, we used as loss function the categorical crossentropy, because we have several classes in our classification problem, as optimizer the Adam optimization algorithm is used for both of our models, and as evaluation measure the validation accuracy.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 48)        480

 conv2d_1 (Conv2D)           (None, 48, 48, 48)        20784

 max_pooling2d (MaxPooling2D  (None, 48, 48, 48)       0
 )

 dropout (Dropout)           (None, 48, 48, 48)        0

 conv2d_2 (Conv2D)           (None, 48, 48, 64)        27712

 conv2d_3 (Conv2D)           (None, 48, 48, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 48, 48, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 48, 48, 64)        0

 conv2d_4 (Conv2D)           (None, 48, 48, 124)       71548

 conv2d_5 (Conv2D)           (None, 48, 48, 124)       138508

 max_pooling2d_2 (MaxPooling  (None, 48, 48, 124)      0
 2D)

 flatten (Flatten)           (None, 285696)            0

 dense (Dense)               (None, 1)                 285697

=================================================================
Total params: 581,657
Trainable params: 581,657
Non-trainable params: 0
```

Model 1 Summary

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 48, 48, 48)        816

 conv2d_7 (Conv2D)           (None, 48, 48, 48)        36912

 max_pooling2d_3 (MaxPooling  (None, 48, 48, 48)       0
 2D)

 dropout_2 (Dropout)         (None, 48, 48, 48)        0

 conv2d_8 (Conv2D)           (None, 48, 48, 64)        49216

 conv2d_9 (Conv2D)           (None, 48, 48, 64)        65600

 max_pooling2d_4 (MaxPooling  (None, 48, 48, 64)       0
 2D)

 dropout_3 (Dropout)         (None, 48, 48, 64)        0

 conv2d_10 (Conv2D)          (None, 48, 48, 128)       73856

 conv2d_11 (Conv2D)          (None, 48, 48, 128)       147584

 max_pooling2d_5 (MaxPooling  (None, 48, 48, 128)      0
 2D)

 flatten_1 (Flatten)         (None, 294912)            0

 dense_1 (Dense)             (None, 1)                 294913

=================================================================
Total params: 668,897
Trainable params: 668,897
Non-trainable params: 0
```

Model 2 Summary

Evaluation Protocol
Our models will be accessed in terms of accuracy in emotion prediction for the test set, meaning in new data that the models where not trained on. The percentage of the correct emotion predictions is actualy the model's accuracy and is expected to be within the range of 40% to 70% or potentially even higher in ideal scenarios. Any percentage less that this limit will indicate poor performance and a model unable to be trained.

Tools

Python with Jupiter Notebooks were used to build and train the models. While libraries as Tensorflow and Keras became really handy for the machine and deep learning tasks needed for the project implementation. We should not forget the basic python libraries such as pandas and numpy that are needed and crucial for every project.

Member(s)

Effranthia Anastasia Kitsou was the one responsible for this project and every task was performed by her.

TimePlan

The bigest part of this project was about revising the matterial provided throughout the course and deeply understanding it in order to built new models following the same logic in terms of structuring. Another big part of the project was about selecting the correct dataset as many emotion detections datasets are available, such as the FER dataset or the EMOTIC dataset and also several other widely used datasets.
In fact the whole project took less than a month to be completed as it did not conflict with other assignments both at the start of it and at the last month before the due date.

Resources

R Vemulapalli, A Agarwala, “A Compact Embedding for Facial Expression Similarity”, CoRR, abs/1811.11283, 2018.
https://github.com/dim4o/emotion-recognition