Telaprolu P Balamanikantashanmukha Sree Charan

CS07556 - Machine Learning

March 24, 2021

**FACIAL EXPRESSIONS IDENTIFICATION BY APPLYING KERAS**

**Convolutional neural networks:**

Convolution neural networks is a type of neural networks which is also known as ConvNet. They are mainly used for image analysis. They can also be used for data analysis and classification problems as well. They identify a pattern between the data. This pattern detection makes them useful in the image analysis. Even for the facial expression recognition, it is so necessary to detect the pattern as this will be useful for differentiating the various facial expressions. So, this convolution neural networks will be useful in my project in analysing my data set ( which consists of 35887 grayscale, 48x48 sized face images with various emotions -7 emotions) i.e., Image analysis and differentiating the various facial expressions. In a CNN, the hidden layers consist of convolutional layers. There can also be non-convolutional hidden layers, but the basis of CNN are the convolutional layers. Convolutional layers are the ones which decide the pattern in the CNN. We have to decide the number of filters a convolutional layer will have. Patterns the filter can detect which may be the edges or images. Filters may be edge detectors, or they may detect corners.

The main categories where CNN is used are,

1.Image Recognition

2.Speech Recognition

3.Natural language processing

4.Video analysis

Pooling is used in the convolutional neural networks. There are 3 types of pooling,

(a) Maximum pooling

(b) Average pooling

(c) Global pooling

In CNN, Maximum pooling or the max pooling is used. Maximum pooling is a sample-based discretization process. It is a pooling algorithm which calculates the largest value in the patch of each feature map. In CNN the pooling layers are used for reducing the parameters to know the computation performed in the network.

In CNN, there are 4 steps which are,

1. Convolution operation- ReLU layer

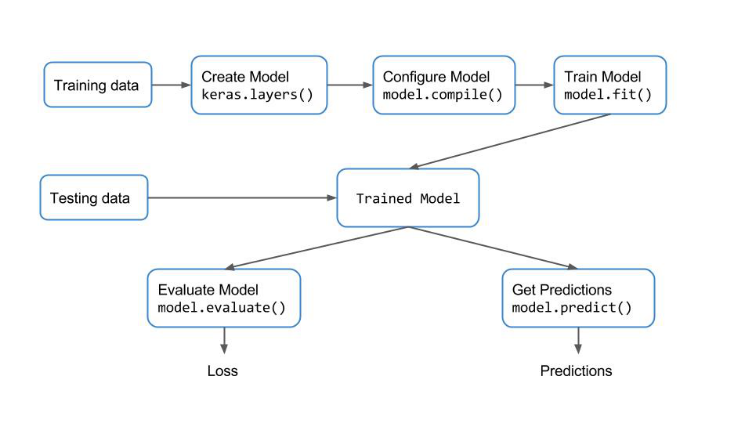
2. Pooling

3. Flattening

4. Full connection

**KERAS**

Keras is a deep learning **Application Programming Interface** (API) written in Python, running on top of the machine learning platform Tensor flow. Keras which is a high-level neural network API, leads the way to the commoditization of deep learning and also artificial intelligence. Its minimalistic, modular approach makes it a breeze to urge deep neural networks up and running. It runs on the top of lower-level libraries, which are used as back end including Tensor Flow, Theano, CNTK. The Keras code is portable which implements a neural network in Keras using Theano as a backened and then specify the backend to subsequently run on the Tensor Flow, and no further changes are required to the code. Keras is easy to use neural network library which promotes a simple and intuitive syntax than any other. It has been developed with a focus on enabling faster experimentation.



These are the steps involved in deep learning neural network model in Python using Keras.

1. Load Data.
2. Define Keras Model.
3. Compile Keras Model.
4. Fit Keras Model.
5. Evaluate Keras Model.
6. Tie It All Together.
7. Make Predictions.

# Methodology

In this paper, deep learning with the convolutional neural network approach is used. The Keras Application Programming Interface and OpenCV framework were used. OpenCV is used for the automatic detection of faces and drawing bounding boxes around them. OpenCV consists of many pretrained classifiers for face, eyes, smile etc. The model here is trained with the dataset used is FER2013 (which is also used in International Conference on Machine Learning). This dataset can be taken from the Kaggle website.

All the images of the dataset are of size 48\*48. To select some images from every category, utility function can be used.