Convolutional Neural Networks for image classification

Julio Nicolás Reyes Torres
Universidad de los Andes
jn.reyes10@uniandes.edu.co

Juan David Triana
Universidad de los Andes
jd.triana@uniandes.edu.co

Abstract

This article presents the concept to built an architecture for a "Convolutional Neural Network (CNN)" to analyze a specific problem of attribute classification. It is presented the main idea of a CNN, its parts, architectures and applications to get one of the most important techniques to attack computer vision problems.

1. Introduction

A recognition problematic for computer vision relies on face detection with multiple features regarding specific people. Among these, certain characteristics are needed to be understood by recognizing different features, such as glasses, hair color, gender, among others. Various algorithms have been implemented to solve this problematic. HOG-SVM, can certainly obtain accurate descriptions of each image and can classify thoroughly each one with a specific discriminate hyper plane. Moreover, the state of the art algorithm relies on the use of Convolutional Neural Networks (CNN), which performs a series of convolutions over images, evaluating its results with the presented groundtruth and optimizing the function through different loss functions [1]. In this work, a face recognition algorithm using CNN is going to be evaluated on the CelebA dataset.

2. Methods

2.1. Data set

The dataset used in this work was the CelebA dataset, which has 200 thousand images of celebrities with 40 different attributes. Some of the classes attained for each image are: arched eyebrows, attractiveness, hair color, among others. [3]

2.2. Alternative Method

A way to address this problematic would be using a HOG-SVM method, where a description with histograms of oriented gradients over each image can be the target for further classification. Certainly, various problems could be seen using this method, since support vector machines are binary discriminative classifiers. This means that for every class or attribute adjacent to an image a support vector machine should be used, meaning a total of 40 SVMs. In general, the computation of this problem is expensive and unaccurate.

3. Convolutional Neural Network

Many applications in machine learning are attacked by the Convolutional Neural Network (CNNs).

A CNN is understood as a combination of multiple algorithms that work well together [4]. The main difference between CNNs and a conventional neural net is the preprocessing. CNNs have two main parts:

1. **Feature engineering / preprocessing** turn our images into something that the algorithm can more efficiently interpret
2. **Classification:** train the algorithm to map our images to the given classes and understand the underlying relationship

At the first part, the process of convolution with a kernel is applied as part of the feature-engineering step. This "filter" gets at the end the output of the convolution process, which is called: "convolved feature" or "feature map". This is important because this process involves a more optimal representation of the input image [2]. In PytTorch this function is given by the sentence: "torch.nn.Conv2d()"

Other important parameters to keep in mind are:

ReLU: It is a nonlinear function to help approximate such a relationship in the underlying data. It is given by: $Max(0, input)$.

Max Pooling: It is the last part of the feature engineering step in CNNs. The idea is to pass over sections of the image and pool them into the highest value for each one. In figure (1) is presented this process.

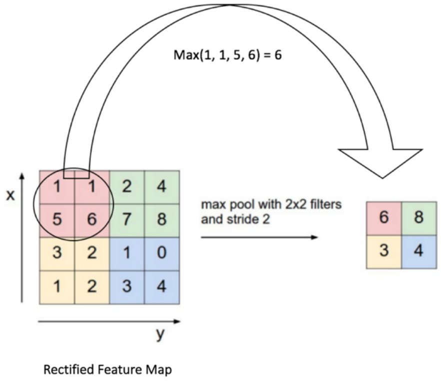

Figure 1: Max pooling to an image.

4. Evaluation

5. Results

6. Discussion

7. Conclusions

References

[1] e. a. Agrawal P. Analyzing the performance of multylayer neural networks for object recognition, 2014.

[2] Algorithmia Blog. Convolutional Neural Nets in PyTorch.

[3] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[4] PyTorch Tutorials. Training a Classifier.