Project Deep Learning

Auteurs:

Clément Jouan

Responsable:

Le Van Linh

17 février 2019

Table des matières

1 Introduction

Machine Learning is a learning method which allows a machine to learn some methods. The machine will then be able to recognize some schemas through the use of multiple statistical tests. The latter will also be able to sort the data thanks to an apprenticeship on these data.

The principle behind neural networks is to simulate (like a simplified copy but still faithful) interconnected brain cells. This simulation is done inside a computer so that it learns, recognize patterns and make decisions like a human brain would. Computer simulations are just accumulations of algebraic variables and mathematical equations that connect these simulations among themselves.

A classical neuron network has thousands of artificial neurons arranged in a series of layers. Each of these layers is connected to the layers at its side. Some of these layers are the input units, they receive information that the network will attempt to treat. Some others are on the opposite side of the network (output) and indicate how they react to the information they have learned. Between the input and output units are one or more layers of hidden units, forming the vast majority of this artificial network. Their role is to transform the input units into something that the output units can use. Most networks are entirely interconnected, each hidden unit and each output unit is connected at each unit of the layers at his side. The connections between one unit and another are represented by a number named 'weight'. This weight can be positive if a unit excites an other or negative, if a unit deletes or inhibits an other. Thus, the higher the weight, the greater the influence of a unit on an other is important. This corresponds to the same system occurring in the human brain : one brain cell has an impact on another thanks to synapses.

The information flows through a network of neurons in two ways. When it learns (so when it's trained) or when it's working normally (after being trained). The information is introduced in the network via the input units. This triggers hidden unit layers as well as those arriving at the exit units. However, all units do not send a signal continuously. Indeed, each unit receives entries units to the left, and the entries are multiplied by the weight of connections they go through. Each unit adds up all the entries that she receives in this way. If the sum is greater than a certain threshold, the unit sends a signal and triggers the units to which it is connected (so those on the right).

For a neural network to learn, there must be an element of feedback. In the same way that a child is told if what he does is good or bad. In this way, the more the difference in what was done and the result is large, the more the answer will be radically changed. This feedback process is called retro-propagation. This involves comparing the output that a network produces with the output it was supposed to produce. The difference between the two can then be used to modify the connections between the network units, from the output units to the units of inputs while going through the hidden units and this in 'running back'. Thus, the back propagation will cause the learning of the network, reducing the difference between the actual output and the planned output at the point where both coincide. Therefore, the network will find the 'right' answer as he should. Once the network has been educated with enough learning examples, it arrives at a point where it is possible to present it with a whole new ensemble of entries it has never met before.

In this project, we will focus on designing a Convolutional Neural Network (CNN) to classify a fruit dataset.

2 Material and method

2.1 Material

For this project, we will be using the PyTorch framework which is an open-source library for Python. It is also a deep learning framework with Tensor computation and GPU acceleration. It contains different components such as :

— Torch.autograd which provides all differentiable Tensor operations in Torch
— Torch.nn which is a library for neural network
— Torch.multiprocessing which is used for multiprocessing with Python
— Torch.utils which is used as a DataLoader, Trainer and some other functions
— Torch.legacy which is the legacy code which has inherited from Torch for backward propagation.

We also have a dataset which contains the images of 95 type of fruits (Apple, banana, Apricot...). This dataset has 65 429 images. The training dataset contains 48 905 images with one fruit per image. The test dataset contains 16 421 images with also one fruit per image. Each image has a size of 100x100. This dataset can be upload at : here.

2.2 Method

To propose a CNN to classify the images into different classes, the network has to be tested with different hyper-parameters. These parameters are from the update rule to the number of epochs. The comparaison of the results depending of the different hyper-parameters will allow to adjust the laters. To do so, we have to write a program to test our trained model. It receives an image as test image, the trained model and it is supposed to provide the class for the tested image.

3 Results

I do not have any results to present here. I had trouble with the installation of PyTorch. (I tried on 3 different computers.) But, still, I've made some bibliography in order to understand the CNN.

First, i've taken interest in epochs. I've found out that one epoch is when an entire dataset is passed forward and backward through the neural network once. To optimize the learning, we have to increase the number of epochs. So, the learning goes to underfitting to its optimum. But, if too much epochs are used, we will have an overfitting system which will exclude some error.

I could also have changed the loss function and updated the rule to improve my results.
But, there is no secret recipe to know which value for each parameters is necessary. All of them have to be tested. But I couldn't do it.

4 Conclusion

Despite research on parameters which have shown that they can be changed, no results can be obtained without testing the program. This project, however, allowed me to understand how CNNs work and the role of different parameters.
I understood the complexity behind this type of algorithm. But still, I've already done some machine learning with the library scikit-learn, but yet, I regret i couldn't get to do it again with an other library.

5 Bibliography

https ://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

Yamashita, R., Nishio, M., Do, R.K.G. et al. Insights Imaging (2018) 9 : 611.

https ://www.analyticsvidhya.com/blog/2019/01/guide-pytorch-neural-networks-case-studies/

https ://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html

http ://www.nilsschaetti.ch/2018/03/07/introduction-pytorch-3/