College Sciences et Technologies

Deep Learning Project: Fruits 360

Author:

Benjamin BLAIS

Bordeaux
02-15-2019
Report written in LaTeX

Summary

1 Introduction

The current technological advance is pushing people to develop new techniques in order to simplify the use of programs. Artificial intelligence is a booming field that will aim to put in place programs that allow device to imitate real intelligence. When we talk about artificial intelligence we think deep learning. Deep learning is a specific type of machine learning that will allow machine to learn directly by itself in order to improve its performance. Nowadays, deep learning is widely used for automatic translations, financial prediction or diagnosis of some diseases.

During this project, deep learning will be used in order to image recognition. Indeed, we are going from a dataset to create a program allowing the recognition of different types of fruits. From an image our program will have to recognize the fruit on the picture. The advantage of deep learning is the accuracy of results. However, the program can make error recognition, so we want to calculate the percentage of recognition to see if it is successful or not.

2 Material and methods

2.1 Material

The use of a powerful machine was necessary. Deep learning requires a lot of computing power. The computer used is comprised of an E3-1240 CPU with 16GB of RAM with a frequency of 3,70GHz, and a GTX 1060 GPU of Nvidia brand with 6GB dedicated Ram. It's important to specify the GPU because we will use the power of the GPU to run the program. Indeed, the speed of execution being better, we use this technology thanks to CUDA (Compute Unified Device Architecture). The execution time was reduced by 10. Indeed, the execution time of the program was 2400 seconds (with the CPU) against around 200 seconds now (with the GPU). It is therefore not negligible to use CUDA during this project.

2.2 The data

A dataset was provided containing images of different fruits (fruit-360). This dataset is divided into two folders, the "Training" folder and the "Test" folder. Deep learning is composed of two phases: a training phase that will aim to adjust the model and extract the characteristics necessary for image recognition and a test phase that will allow from these characteristics to predict the fruit. Training folder is three times larger than Test folder and contains 83 folders (where each folder corresponds to a fruit). These 83 folders are each composed of a number of images include between 127 and 328. Training folder is composed of 42 798 images. Test folder countain 14 369 images. This dataset comes from the article *Fruit recognition from images using deep learning* from Muresan.H and al.

2.3 Pytorch

The programming language used in this project is python 3. The use of Pytorch was required. Pytorch is an open-source python library of deep learning that relies on Torch. It's used to create neural networks, define different layers that will form these networks, link them together and thus allow training the network. There are different layers in a neural network that are important to describe in order to understand the function of a neural network.

2.4 Convolutional layer

The convolutional layer is the first layer of a neuron network. The convolution layer permit from several input images to achieve a filtering and calculate a convolution product that will be necessary to determine the characteristics of the image. The size of the frame entailing the filtering can be chosen. During our project, we will set the Kernel size value of 5x5. One or more convolutional layers can be used for better results.

2.5 Max pooling layer

For each convolutional layer is added a max pooling layer. Pooling layer are responsible for a reduction of spatial dimensions of the image and will reduce the computing time since the dimensions of the representations are modified. Here, we will define our filter with a 2x2 size. We are, thanks to the max pooling layer limit the over-learning of the convolutional layers.

2.6 Rectified Linear Unit (ReLU)

For each convolutional layer, after application of max pooling layer, a ractified linear unit layer will be applied. This layer will have for only purpose to replace all the negative values by value equal to 0.

2.7 Fully connected layer

Fully connected layer will be used after several convolutional layers. Each neuron of a fully connected layer is connected to all the outputs of the previous layer. These layers allows to classify the images according to the characteristics obtained previously. It's possible to use multiple fully connected layers.

2.8 Loss layer

Loss layer is the final layer of the CNN network (Convolutional neural network). It makes it possible to know during the learning phase (train phase) the difference that exists between the result obtained and the expected result. On Pytorch fonctions exist to calculate the error loss. It's important to minimize this value and so that it tends to 0 to limit errors during the training phase.

3 Results

It was necessary to create a CNN. Indeed, as said before, we want to do image recognition. We had to realize several tests in order to set up layers necessary for the network and to adjust the hyper parameters. At first, it was important to load the images in the train folder used for the training phase and those of the test folder used for the prediction. The number of batch size has been set to 4 (corresponding to 4 images sent over the network). A too low value for this parameter (for example 2) gives too low results. A value too strong when it will have the same consequences by the fact that the size of the sample is too high and will lead to a bad learning.

3.1 Hyper parameters

Then, we decided to adjust the epoch value. Epoch is the number of loops in which the learning phase will take place. So we tested several epoch value. We decided to choose an epoch value equal to 4. Indeed, a value too low of the order of 2 will not be effective enough to train our network. On the other hand, a value that is too high (for example 10) has given us a worse result concerning the prediction. We can therefore think that the network will be over-driven our network and thus provide us with too low prediction results.

An other parameter to take into account is the lr (learning rate). The learning rate corresponds to the network adaptation factor. The higher the value is, the stronger the correction is. However, a too low value is not correct either because it will have a correction that is too low. So we decided to set the value of lr to 0.001. With a lr close to 0.5, our prediction percentage tended to 1% while for an lr of 0.00001 our prediction tended to 6%. So we set our value of lr to 0.001.

3.2 The layers of the network

Now, we will focus on the setting of the network and therefore the layers. All the tests of layers will be carried out with lr = 0.001 and epoch = 4. We thus study 3 cases concerning convolutional layers. In first time, we use 2 convolutional layers, then 3 layers and to finish 4 layers as in the article studied [1]. For two layers, we used different settings for the input channel and the output channel with a kernel size set to 5. For the first layer FC (Fully connected), the input channel is set to 22 * 22 * Output size (conv2). The output of the last layer is set to 83.

Parameters of convolutional layers	Percentage of prediction
3 - 10 - 16	89%
3 - 16 - 30	94%
3 - 16 - 40	95%
3 - 25 - 80	93%

Table 1: Result of the prediction according to the parameters for two convolutional layers

Here, we can see that the best results were obtained for setting 3-16-40. Indeed , an output value that is too high will be responsible for over-driving the network. So, we used 2 convolutional layers like :
self.conv1 = nn.Conv2d(3, 16, 5)
self.conv2 = nn.Conv2d(16,40,5)

The use of three convolution layers has been more complex. In our case we have fixed the input of the first layer at three and the ouput of the last layer at 30 obtained 89%. So, in our example we will privilege two convolutional layers. Regarding the use of four convolutional layers. We tried to create the network present in the publication (Article [1], in commentary on program). So we used 4 layers of convolution whose ouput of the last is equal to 128. We obtained a prediction rate of 90%. It is a good result but less than the result obtained with only two layers. After all these adjustments, we decided to keep two convolutional layers with the first parameters (3,16,5) and the second (16,40,5).

Regarding fully connected layers. We first decided to use 3 layers. They will thus allow to classify the images at the end of the last layer. With three layers we obtained results between 90 and 95% for the previously mentioned parameters.We added a fourth FC layer. We found that the percentages stabilize around 90-92%. So we decided to keep only 3 FC layers since we got very similar results with 3 or 4 layers. We tested our network without an FC layer. Of course our prediction percentage was close to 0 %. This is a value that can be expected since the FC layers allow detection.

So we made a program to know the percentage of fruit prediction. Indeed, in a first phase we created a neural network composed of two convolutional layers and 3 fully connected layers. The learning rate was set at 0.001 and the lap number for the training phase was set at 4. Thanks to these results, we obtained a prediction percentage of 95%. 95% shows that the program will recognise 95 types of fruit out of 100 (5% error). This value is a good value that shows the accuracy of our network for image recognition. It would be interesting to test our results on another dataset.

4 Conclusion

Deep learning is very useful in today's world. Deep learning is composed of two phases, the learning phase to train the network and the prediction phase. In our project, we created a CNN whose goal is the images recognition of fruits. From two folder, one for training and the other for testing, our prediction program reaches 95% with two convolutional layers and three fully connected layers. The learning rate has been fixed at 0.001 and the epoch value at 4. We can therefore say that the accuracy of the program is very high since it goes during the prediction phase to obtain only 5% error.

Pytorch is a very useful framework when one is interested in deep learning as in our case. We can through him create various neural networks and adapt them according to our use. It would be interesting to test another machine learning technique to compare our results with the latter.

References

[1] Mureşan H, Oltean M. Fruit recognition from images using deep learning. Acta Univ. Sapientiae, Informatica, 2018,26-42.

[2] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, Vol 521, 28 May 2015.

[3] Deep learning in https://towardsdatascience.com/