Project Deep Learning

Author:

Lila De Oliveira

Responsable:

LE Van Linh

February 15th 2019

Contents

1 Introduction

State of art

Supervised learning techniques can provide very good results, and their performance strongly depends on the quality of features previously found. There are several methods for extracting and describing features.

In practice, the classification error is never zero. The results can then be improved by creating new feature extraction methods that are more adapted to the studied images, or by using a "better" classifier.

But in 2012, a revolution occurs: at the annual ILSVRC computer vision competition, a new Deep Learning algorithm explodes the records. This is a convolutional neural network called AlexNet.

AlexNet was developed by researchers in Toronto in 2012, but is largely inspired by the LeNet convolutional neural network, developed in 1998 by French researcher Yann LeCun. LeNet is the pioneer, and AlexNet the one who created the enthusiasm around Deep Learning.[1]

Convolutional neural networks have a methodology similar to traditional methods of supervised learning: they receive input images, detect the features of each of them, and then drag a classifier over them.

The pytorch library is often used in searches on CNN applied to image recognition.[2] [3]

2 Material and Methods

Material

For this project, the python language is used. In addition, it is the Pytorch library that will help implement the CNN on the given database.

Python is an interpreted programming language, multi-paradigm and multiplatform. It promotes structured, functional and object oriented imperative programming.

Python is a programming language known to be completely free. It is a complete and powerful language in many fields. It is object oriented but does not impose this type of programming.

Moreover, its syntax remains very simple and the code can be very readable. His performances seem to be very honorable for a so-called scripting language. In addition, it shortens the development cycle compared to compiled languages and allows rapid prototyping of projects.

As for PyTorch, it is a Python open source machine learning software library that relies on Torch developed by Facebook. PyTorch makes it possible to perform the necessary tensor calculations, especially for deep learning.

A database containes the images of 95 types of fruits distributed in 65429 separate fruit images on file. A folder represents a fruit. It is therefore on this database that a CNN must be applied in order to recognize the fruits presented to the algorithm. There is a file of training set with 48905 images and there is a file of test set with 16421 images.

Methods

There are four types of layers for a convolutional neural network: the convolutional layer, the pooling layer, the ReLU correction layer, and the fully-connected layer.

The convolutional layer is the key component of convolutional neural networks, and is still at least their first layer.

The pooling layer reduces the number of parameters and calculations in the network. This improves the efficiency of the network and avoids over-learning.

ReLU (Rectified Linear Units) denotes the nonlinear real function defined by ReLU (x) = max (0, x). The ReLU correction layer therefore replaces all negative values received as inputs with zeros. It plays the role of activation function.

A CNN is a stack of these layers, layers can be repeated several (or many) times. You can apply a convolutional layer then a ReLU layer, then convolutional layer, then another layer ReLU, then a pooling layer etc...

Moreover, the convolutional layer and pooling layers have modifiable hyperparameters. The convolutional layer has 4 hyperparameters while the pooling layer has 2.

So, in this project, the chosen architecture was a convolutional layer then a ReLU correction and finally a max pooling layer, and these three layers repeated again in the same order. Then the algorithm goes to a fully connected layer to

replace all the negative values with 0 and then a ReLU correction layer again. This is repeated 2 times before making a final layer "Fully connected" which will be the soft max. This algorithm is summarized in Figure 1

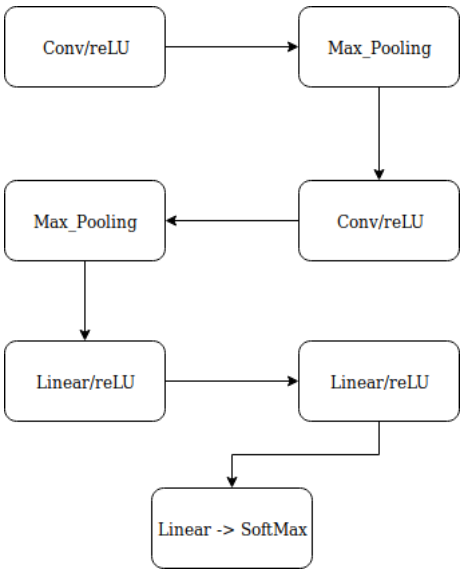

Figure 1: Architecture of CNN in this project

3 Results

With pytorch, the different layers are written this way :

- *torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride, padding)* – **applies convolution**
- *torch.nn.relu(x)* – **applies ReLU**
- *torch.nn.MaxPool2d(kernel_size, stride, padding)* – **applies max pooling**
- *torch.nn.Linear(in_features, out_features)* – **fully connected layer (multiply inputs by learned weights)**

In training neural network, one epoch means one pass of the full training set and lr is for *learning rate*. Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient. We must therefore try to modify these parameters to obtain the best possible accuracy as shown in Figure 2.

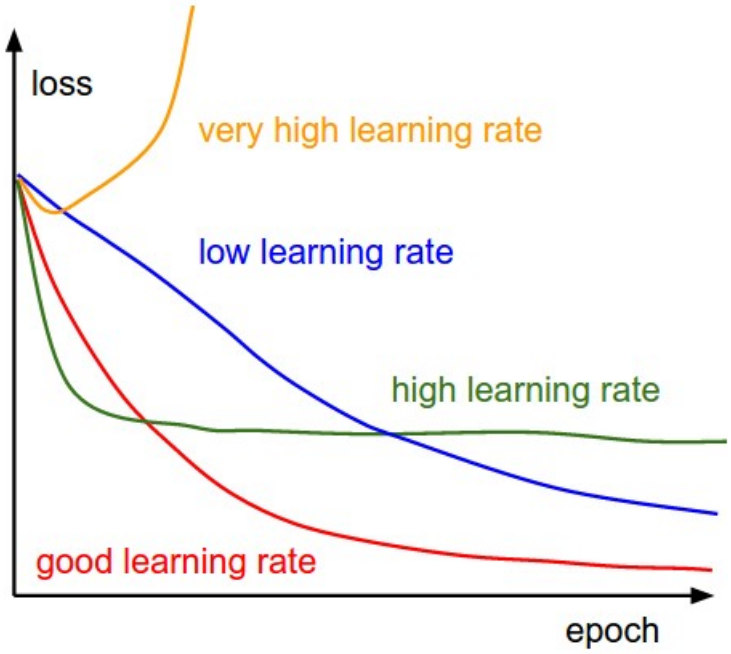

Figure 2: Effect of various learning rates on convergence (Img Credit: cs231n)

Once this algorithm is tested with an epoch in parameter of 1 and a lr of 0,1, rather bad results appear. Indeed, the accuracy of the algorithm does not exceed 15 %.

In constrast, by modifying this parameters, and putting an epoch at 5 and a lr at 0.0001, the algorithm reaches more than 80% accuracy. In addition, as the iteration of the algorithm on the images during the training, the running loss goes from about 4.4 to less than 0.07, which is very good. This means that the algorithm is training properly.

However, it is notable that increasing the epoch by 5, and therefore iterating the algorithm 5 times considerably increases the execution time.

4 Conclusion

To conclude, this project made it possible to understand how the CNN method works in the field of deep learning and image recognition. By looking at the different layers and their roles and changing the parameters, it is quite easy to obtain results with good accuracy.

Deep learning has the great quality to look for features to find to try to get the best accuracy possible. This is not possible with machine learning techniques such as SVM where you have to search and extract even the features.

In addition, nowadays, using libraries such as Pytorch, it is possible to create algorithms to answer our problems. Communities that follow bookstores like Pytorch are, in addition, very reactive. The problems encountered are therefore easily solved.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012. 3

[2] N. Inoue, E. Simo-Serra, T. Yamasaki, and H. Ishikawa, "Multi-label fashion image classification with minimal human supervision," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 3

[3] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, "Iris recognition with off-the-shelf cnn features: A deep learning perspective," *IEEE Access*, vol. 6, pp. 18848–18855, 2018. 3