Project 3: Hyperparameter optimization

Matias Cinera
Computer Science Department, *University of South Florida*
Tampa, Florida, 33612, USA
cinera@usf.edu

***Abstract*— This paper describes my understanding of Convolutional Neural Networks through the manipulation of the hyperparameters.**

This paper describes my implementation and understanding of geometric transformations and pixel operations of images. A geometric transformation is a mathematical relationship between two planar surfaces (images) which hold some type of relationship. Pixel operations modify the value of pixels to alter how the image looks without affecting its shape. This paper will implement (in code) and test the fundamental geometric transformations and pixel operations over a batch of images.
Keywords—Geometric transformations, Pixel operations

I. Introduction

Some introduction introduction introduction introduction introduction introduction introduction introduction introduction introduction introduction introduction

The project required students to implement five geometric transformations (Translation, Rotation, Scaling, Affine and Projection) and three pixel-operations (Color Brightness and Contrast, Gamma Correction, and Histogram equalization). The project also required two normalization operations which are commonly used for deep learning purposes (Mean & Standard Deviation and Batch Normalization). Each image will be processed by all geometric transformation and pixel operation functions (each function outputting an image). Finally, the batches are processed by the two normalization functions and stored in an output folder.

II. Neural Networks.

A. Explain what neural networks are

A geometric transformation is a mathematical relationship between two planar surfaces (images in this project) which hold some type of relationship. A pixel is composed of three integers (red, green, blue) in a range from 0 to 255. These integers represent the intensity of the given values (r, g, b). Pixel operations are essentially just modifying these values.

Since in this project we are dealing with images, we could think of a pixel as a point in a 2-dimensional coordinate plane (our image). For this project, we use homogeneous coordinates to represent 2d points (pixels).

B. Exaplin what conv neural networks are

A 2-dimensional point: $\mathbf{x} = (x, y)$

Can be represented as a 3-dimensional line through the origin.

Where $\tilde{x}$ is the homogeneous representation of the point. By this logic, we could also transform a 3D line into a 2D point. However, this is a theoretical 3D representation, thus, homogeneous coordinates are only used to simplify operations. This is because many transformations can be described as matrix multiplication.

III. Hyperparameters

Explain what hyper parameters are adsfjk asdkf aksdjf sdakfj aksdjf askdfj asdkfj aksdfj

A. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

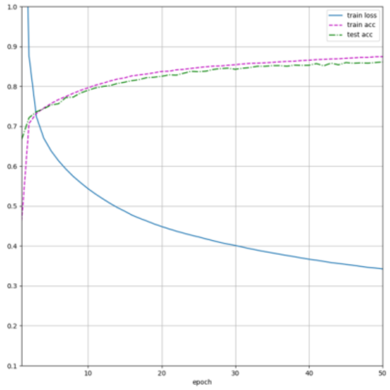

Figure 1. Optimization Function: Adam

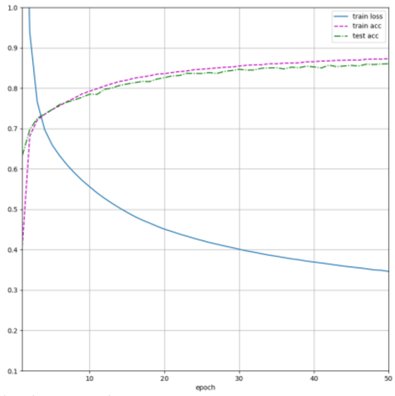

Figure 2. Optimization Function: RMSprop

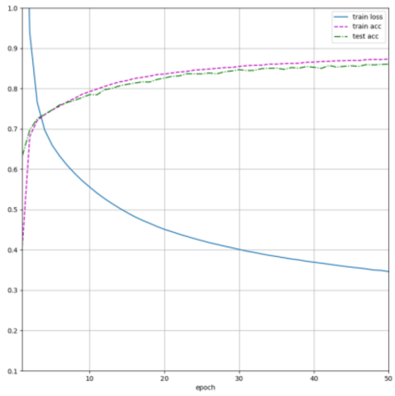

Figure 2. Optimization Function: RMSprop

B. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

C. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

D. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

E. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

F. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

G. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

H. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

I. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

J. Optimization Functions

Optimization functions blacaos[dkncasdkn das;kj va;skdj v;kasj ;kja sdv;kj

Figure 2. Translation, shift x = -0.2, shift y = -0.1

IV. Results

Overall, the results of this project were successful. Almost all of the functions acted as expected. The only function that didn't work as expected was the mean and normal deviation function. The function is supposed to return two vectors of size 3, with the mean and sd of RGB values. Even though my function successfully returned two vectors that met this requirement, the function didn't seem to work as expected.

V. Conclusion

At first, this project seemed to be more complicated than that it was. Most of the functions to be implemented were given by the professor, we just needed to adapt them to the boundaries of the project. Only knowing the basics of python gave me a chance to unravel the power of this programming language. As for the concepts discussed in this paper, most of them are still challenging to me. However, now I have a general understanding of them and a powerful tool to implement them.

References

[1] "Lab color space and values: X-rite color blog," X. [Online]. Available: https://www.xrite.com/blog/lab-color-space. [Accessed: 15-Sep-2021].

[2] T. Mouw, "Lab color space and values: X-rite color blog," X. [Online]. Available: https://www.xrite.com/blog/lab-color-space. [Accessed: 16-Sep-2021].

[3] S.Sarkar, "CAP4410 Lecture 3 Pixel operations color_histogram," Pixel operations color histogram. [Online]. Available: CAP_4410_Lecture_3 [Accessed: 10-Sep-2021].

[4] S.Sarkar, "CAP4410 Lecture 2 2D Geometric Primitives and Transformation [Online]. Available: CAP_4410_Lecture 2 [Accessed: 10-Sep-2021].

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

https://d2l.ai/chapter_convolutional-neural-networks/index.html

good for theory ^