AMRITA SCHOOL OF ENGINEERING

COIMBATORE - 641112

COMPUTATIONAL ENGINEERING AND NETWORKING

MACHINE LEARNING TERM PROJECT REPORT

Convex Optimization of Image Denoising using CVXPY layers.

P H V PAVAN KUMAR	-	CB.EN.P2CEN20020
RAHESH R	-	CB.EN.P2CEN20021
SACHIN ADITIYA B	-	CB.EN.P2CEN20030

Project Guide:

Dr. Sowmya V

Associate Professor

Computational Engineering and Networking(CEN)

Coimbatore ,TN

June 29, 2021

ACKNOWLEDGEMENTS

We would like to extend our gratitude to our project guide, **Dr. Sowmya V**, for her guidance, invaluable support and dedication that she shared with us to help the conduct of this study. we have immense pleasure in expressing our heartfelt thanks to our honorable Dean **Dr. Sanangan Ramanathan** for providing us an opportunity to work on this project. It's our honor to express our reverence to **Dr.K.P.Soman**, Chairperson, Department of Computational Engineering and Networking for this opportunity, and providing with all the facility that was required. we also extend our earnest appreciations to all our associates who have patiently extended all kind of help for completing this undertaking.

ABSTRACT

As we know image classification plays a very important role in either machine learning or deep learning task. The main problem for mis-classification of data is due to the presence of either noise in the data or either due to the class imbalance, here in this paper we are dealing with the noise that present in the high dimensional data i.e., images and we are using the convex optimization(CVXPY) techniques which are implemented as CVXPY layers in python for denoising images. We are classifying the original MNIST data set (before adding noise) with VGG16 architecture and after adding various levels of Gaussian noise to the data we are denoising the data using CVXPY layers and again classifying our data back with the VGG16 architecture and comparing the results what we got with original noise free images.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Digitized images have become an integral part of our lives in day-to-day applications, which is an enthralling part of Computer Vision and Robotics. But due to the influence of environmental constraint, transmission issues, and other factors, images are unavoidably corrupted by noise due to acquisition, compression, and transmission. This noise level will vary from low to high range, which leads to image distortion. In order to restore the originality of the image, the Image denoising technique is being used.

The image denoising technique helps in reducing noise. But the common problem faced by many researchers is to find an effective convex optimization solver for the denoising technique

In this project,our primary objective is to analyze the performance of Convex Optimization CVXPY layers solver in modified VGG16 Net for Image Classification of Noisy Image data set. The rest of the report is arranged as follows:2. Literature Survey 3.Objectives 4.Theoretical Background 5.Experimental Results 6.Conclusion

Chapter 2

Literature Review

In [6] Srikanth, M et al, they have presented a novel methodology called least square approach for image in-painting. Also the missing sample estimation methodology was extended from 1D to 2D image for in-painting.

A novel methodology [3] was introduced by called the least square weighted regularization which is first applied column-wise and then applied row-wise to denoise the image.

In Convex Optimization written by Stephen Boyd, a great exposure was given to different types of Convex optimization problems.

In Differentiable Convex Optimization Layers [1] paper provides a basic understanding of convex optimization. Also a subset of disciplined convex programming was introduced which helped in reducing new solver solution to a existing problems.

Chapter 3

Objectives

In this project, our primary objective is to analyze the effect of the Convex Optimization python library CVXPY layers, in VGG16 architecture for Image Classification of noisy Image data set. We have added different level of Gaussian noise in the MNIST dataset and reduced the noise using CVXPY layers with different parameter. The analysis of these experiments are recorded and documented in this paper.

Chapter 4

Theoretical Background

4.1 Noise Reduction

Noise reduction or noise removal generally means removing unwanted data from the medium which delivers some useful data. This medium can be audio, text, video or image data. The noise disturbs the quality of the data already available in the medium resulting in corruption of data. The nature of the noise differs with the medium on which it inherits. Traditional filters are used for the removal of image in 1D and 2D spacial data.

In an image, the noise is the presence of random pixel values throughout the 2D space. This is due to many external reasons like malfunction of camera, disk etc. In image processing there are many types of noise which are being added to the images manually for various reasons. These noises include Gaussian, salt, salt and pepper, white, brown etc. Each one has its own properties.

In our approach we are adding the Gaussian noise to the datasets in different levels and analysing the performance of the CVXPY layer for each levels.

4.2 Least Square Image Denoising

The noise in the 1D signal can be removed though the least square weighted regularization algorithm which was proposed by Ivan. W. Selesnick [7] in his paper.

$$min_x ||y - x||_2^2 + ||\mathbf{D}x||_2^2 \qquad (1)$$

In equation(1), the L2-norm of the second order derivative of output denoised signal. The L2-

norm squared energy of the derivative captures the degree of smoothness. This reduces the level of noise present in the input signal. The minimization of first term in objective function in eqn. (1) forces the output y to be similar to input signal. The minimization of the second term in eqn. (1) leads to smoothing of the noisy input signal.$\lambda > 0$ is the control parameter used for the least square weighted regularization algorithm. The minimization of the sum in eqn. (1) results in achieving x to be smooth and similar to y.[7].

This approach is applied to the image considering it as a combination of signals of same length. For example, an 512x512 image is considered as 512 signals with length 512. M. Srikanth et al [9]and Merlin et al [6]in their paper has approached in the denoising for large sized images.

4.3 MNIST dataset

The MNIST dataset is an well known toy dataset used for image classification problem. The data set has the images of handwritten digits of range 0-9. There are total of 70000 images of dimension 28x28 available in this dataset. The train and test set have 60000 and 10000 images respectively.[5]

4.4 VGG16 Architecture

VGG16 is one of the state of the art image classification architectures used widely. In this paper VGG16 architecture is used throughout all model for the image classification. The architecture is proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes[8].

The images are passed through set of many convolutional layers of kernel size 3x3 in each layer. The feature maps produced from these convolutional layers are connected with the fully connected layers which in turn connect the output layer. This layer classifies the image according to the class. The entire architecture of the VGG16 is illustrated below.

For this project, the above architecture is constructed using the Keras framework available with the Tensorflow library.

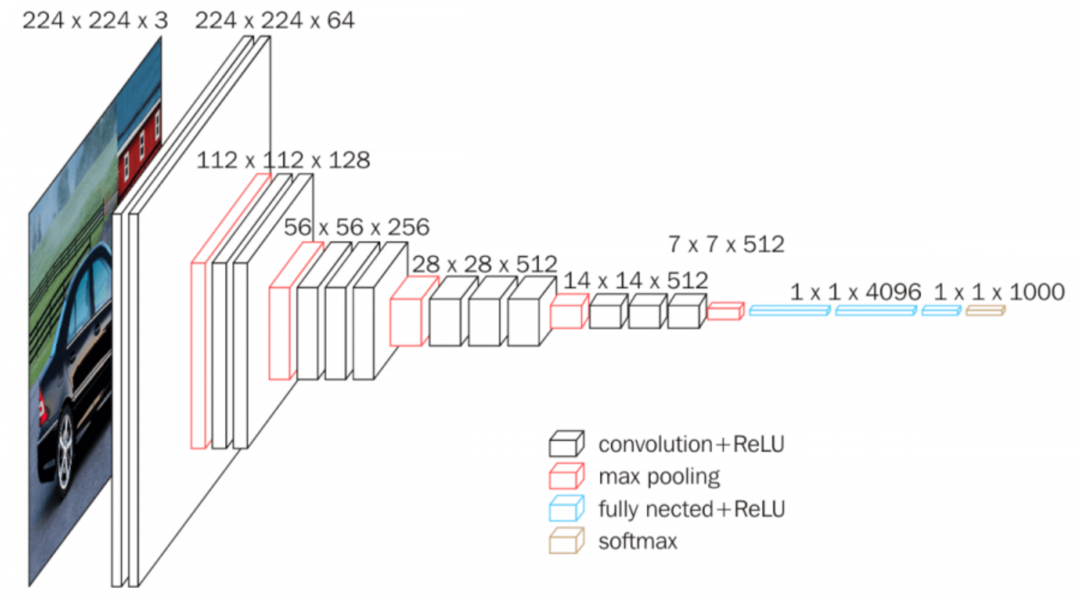

Figure 4.1: VGG16 architecture

4.5 Convex Optimization Models

A convex optimization model predicts an output from an input by solving a convex optimization problem. The class of convex optimization models is large, and includes as special cases many well-known models like linear and logistic regression. We propose a heuristic for learning the parameters in a convex optimization model given a dataset of input-output pairs, using recently developed methods for differentiating the solution of a convex optimization problem with respect to its parameters. We describe three general classes of convex optimization models, maximum a posteriori (MAP) models, utility maximization models, and agent models, and present a numerical experiment for each.[2]

4.6 CVXPY layer

CVXPY is a Python-embedded modeling language for convex optimization problems. It allows us to express your problem in a natural way that follows the math, rather than in the restrictive standard form required by solvers[4]. CVXPY began as a Stanford University research project. It is now developed by many people in different versions, across many institutions and countries. CVXPY layers is a Python library for constructing differentiable convex optimization layers in PyTorch, JAX, and TensorFlow using CVXPY. This library is used for finding the optimum denoised data for the input noisy data. This is attained through the constrains used for the objective given for the problem. Splitting Conic Solver(SCS) is the default solver used by the CVXPY for finding the optimum results.

There are many other solvers available in this library[1].

4.7 Keras custom layer for Convex Optimization

Keras is a python framework used to construct the Deep Learning models. This framework has the class called layers, through which one can create custom layer for custom purpose. The custom layer inherits the modules available in the layers class of Keras. The default constructor receives the arguments needed for the layers and the call method does the execution in forward pass.

A custom layer class in the name of Denoise is built for denoising using CVXPY layers. This class is made inside a python file which inturn imported to the main files we worked. The variable, parameters and constants required for the CVXPY layers including the problem and objective are declared in the default constructor of the custom class. The noisy data from the main program is given as a parameter in this python file and the denoised output returns to the main file.

4.8 Gausssian Noise

Gaussian noise is statistical noise having a probability density function (PDF) equal to that of the normal distribution. This distribution is also called as Gaussian distribution. Skimage python library is used for the addition of the gaussian noise. The MNIST dataset is subjected to this noise as given below for the analysis.

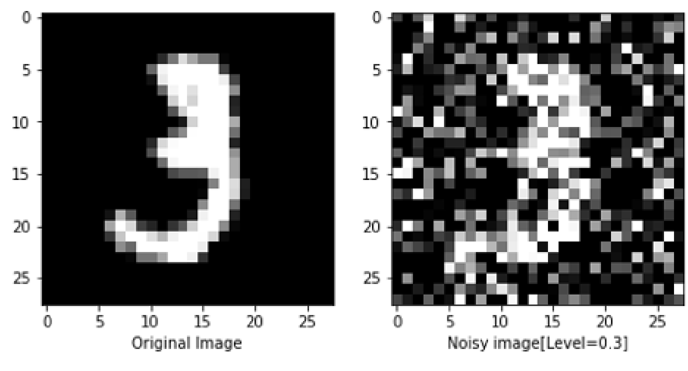

Figure 4.2: Original image(left) and Noisy image(right)

4.9 Metrics: PSNR

PSNR stands for Peak Signal to Noise Ratio. It is used for measuring the quality between the original image and the modified version of the same image. The higher the PSNR, the better the quality of the compressed, or reconstructed image. The mean-square error (MSE) and the peak signal-to-noise ratio (PSNR) are used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

$$PSNR = 20\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right)$$

Chapter 5

Methodology

In order to observe the performance of the CVXPY layer denoising the Gaussian noise present in the image, we have made many number of VGG16 models for classifying the images denoised by the library. The models differs with the type of image used for training and the modification done in the model.

5.1 VGG16 model for MNIST data

This model is created for checking the performance of the original MNIST data as train and test. No modification is done for the VGG16 model here.

5.2 VGG16 model for noisy MNIST data

The MNIST data is added with the Gaussian noise of a particular level. These images are split into train and test sets as per the requirement. A new model is built for training this noisy image set. The model is then tested with the test set. The performance and the metrics are recorded. Few more models are built for different noise levels by repeating the same steps. The performance of those models are recorded.

5.3 Modified VGG16 model for noisy MNIST data

The VGG16 architecture is modified by adding the custom built layer as the first layer. This layer is considered as the first layer for the new model. When the noisy data is given to the model, this custom layer denoise the noise present in the images and return the denoised image to the next layer as tensors.

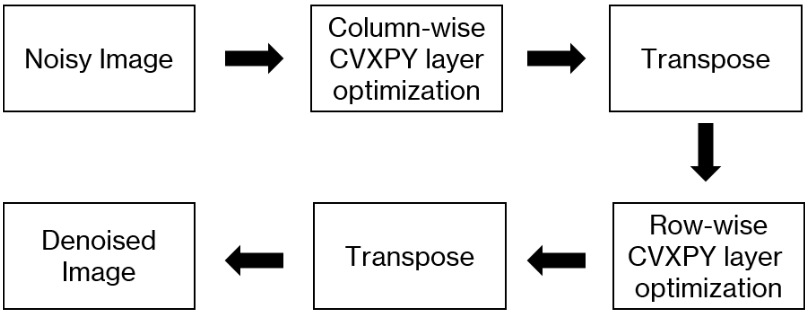

Figure 5.1: Flow of denoising an image in the custom layer

The noise in this output is considered as a optimal denoised images and these images are used for train the a new VGG16 model. Since both process is happening in a same model, this is considered to be a modified VGG model.

Chapter 6

Experimental Results and Discussion

6.1 Results:

As the noise of various level is added to our MNIST data and denoising is done by doing experiments with various values of λ and as a result we are encounter with following results.

NOCVXPY + VGG16			CVXPY + VGG16		
Noise	train accuracy	test accuracy	λ	train accuracy	test accuracy
0	0.9747	0.97	-	-	-
0.3	0.9676	0.95	0.1	0.9236	0.93
			0.2	0.954	0.93
			0.3	0.8425	0.75
			0.5	0.9204	0.88
0.5	0.9642	0.91	0.1	0.8902	0.65
			0.2	0.9148	0.74
			0.3	0.7704	0.79
			0.5	0.7858	0.61
0.7	0.9639	0.91	0.1	0.9568	0.9
			0.2	0.885	0.89
			0.3	0.8808	0.8
			0.5	0.7759	0.63

Table 6.1: Evaluating Noise and Denoise data using VGG16 and modified VGG16 architecture

Here we can clearly notice that when noise level is less then the reduction in noise is achieved pretty decently using lesser value of λ values and which is due the picture size what we used here, which is very small in dimension[28 x28] hence will discuss more about this in the next Chapter 7. To understand more detail behaviour of denoised images we compared the PSNR values of all our experiments for all the 12 conditions mentioned in the Table 6.1, PSNR is calculated with respective to all the noised training set of images along with the denoied training set of images.

NOCVXPY + VGG16(VGG16)		CVXPY + VGG16(MODIFIED VGG16)	
Noise	noised image PSNR	λ	Denoised image PSNR
0	-	-	-
0.3	8.927	0.1	11.919
		0.2	12.633
		0.3	12.865
		0.5	12.943
0.5	7.591	0.1	10.574
		0.2	11.816
		0.3	12.133
		0.5	12.279
0.7	6.864	0.1	9.801
		0.2	10.64
		0.3	11
		0.5	11.276

Table 6.2: Comparison of PSNR Values noised and denoised images

By looking at the PSNR table we can see that the denoising is happening which even we can see in the below mentioned figure.

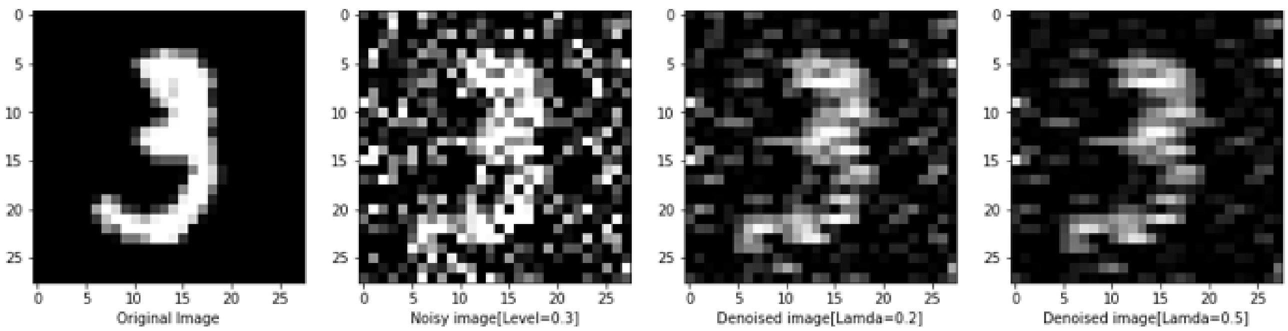

Figure 6.1: Denoising of a 28x28 image with $\lambda = 0.2$ $\lambda = 0.5$

The training accuracy increases in every epoch as the model learns more from the input data. As the training accuracy increases gradually, the training loss decreases epoch by epoch like in the following plots.

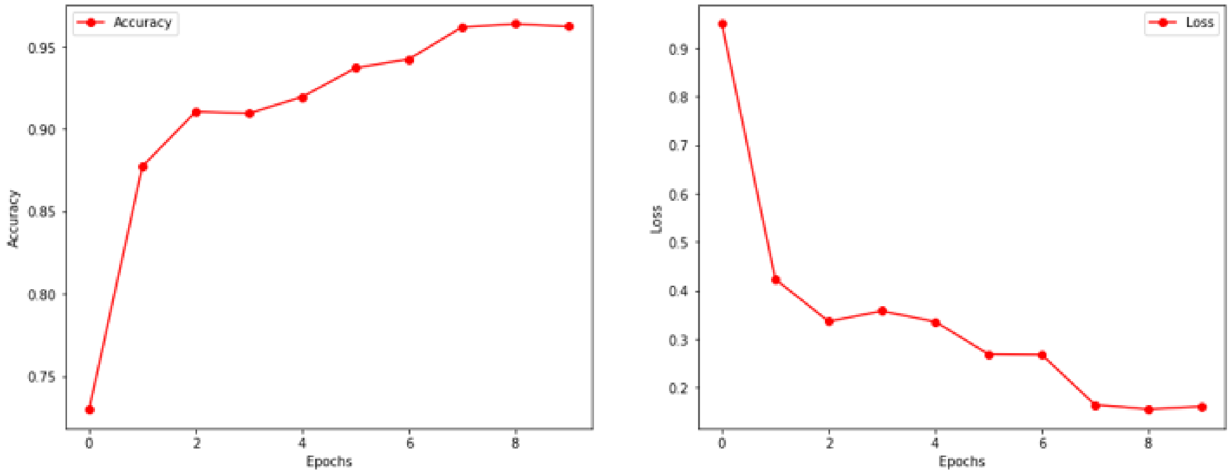

Figure 6.2: Plots for training accuracy and training loss

Chapter 7

Conclusion

Hence we inferred that this approach reduces the noise in the images through convex optimization and accuracy even increases for denoised data in some cases and more over PSNR improves after denoising so for the better result the noise should be less and the control parameter should be optimum and We believe that this approach works well with higher dimension images when inherited with noise in it.

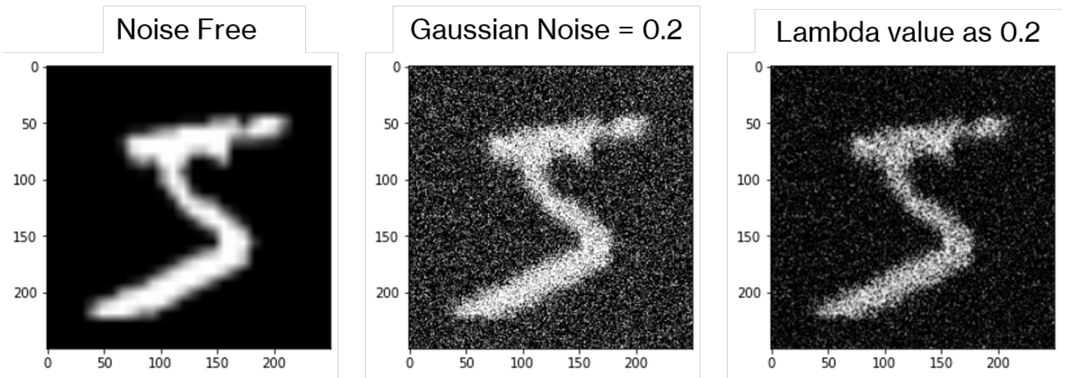

Figure 7.1: Denoising a 500x500 image using CVXPY+VGG16 architecture

Bibliography

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and Zico Kolter. Differentiable convex optimization layers, 2019.

[2] Akshay Agrawal, Shane Barratt, and Stephen Boyd. Learning convex optimization models, 2020.

[3] M. Aiswarya, N. Deepika, Sowmya Vishvanathan, Neethu Mohan, and Soman Kp. Least square based approach for image inpainting. 7:44–59, 04 2016.

[4] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization, 2016.

[5] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[6] Deepa Merlin, Aleena Ajay, Sowmya Vishvanathan, and Soman Kp. Aerial and satellite image denoising using least square weighted regularization method. *Indian Journal of Science and Technology*, 9, 08 2016.

[7] I. Selesnick. Least squares with examples in signal processing 1. *Connexions*, 2013.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[9] M. Srikanth, K. S. Gokul Krishnan, V. Sowmya, and K. P. Soman. Image denoising based on weighted regularized least square method. In *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, pages 1–5, 2017.