COMPUTER VISION AND PATTERN RECOGNITION PROJECT
CNN CLASSIFIER

Alessandro Cesa

January 2024

Contents

Introduction **5**

Problem statement **7**

Dataset 8

Implementation **11**

Part 1 11

Part 2 11

Part 3 11

Results **13**

Bibliography **13**

Introduction

The final project for the course in Computer Vison and Pattern Recognition consists in a classification task on a provided dataset containing 15 categories of images [2]. The classification has to be carried out by employng a Convolutional Neural Network, built following given specifications and steps described in the project assignement.

Problem statement

The project requires the implementation of an image classifier based on Convolutional Neural Networks, following three major steps.

1. Build from scratch a Convolutional Neural Network following this architecture:

   Convolutional Neural Network Architecture

Layer	**Size**
Image Input	$64 \times 64 \times 1$
Convolution	$8 \times 3 \times 3$
ReLU	-
Max Pooling	2×2
Convolution	$16 \times 3 \times 3$
ReLU	-
Max Pooling	2×2
Convolution	$32 \times 3 \times 3$
ReLU	-
Fully Connected	15
Softmax	-
Classification Output	-

   Following these specifications:

   - Rescale the images to 64x64
   - Split the provided training set in 85% for actual training set and 15% to be used as validation set;
   - Employ the stochastic gradient descent with momentum optimization algorithm, using the default parameters of the library you use, except for those specified in the following;

    - Use minibatches of size 32 and initial weights drawn from a Gaussian distribution with a mean of 0 and a standard deviation of 0.01; set the initial bias values to 0;

   It's required to reach an accuracy of 30%

2. Improve the previous result, according to the following suggestions :

    - **Data Augmentation:** Using left-to-right reflections, reach an accuracy of about 40%.
    - **Batch Normalization [1]:** Add batch normalization layers before the ReLU layers.
    - **Change Convolutional Filters:** Change the size and/or the number of the convolutional filters.
    - **Optimization Parameters:** Play with the optimization parameters (learning rate, weights initialization, weights regularization, minibatch size, etc.), and switch to the Adam optimizer.
    - **Dropout:** Add dropout layers.
    - **Ensemble of Networks:** Employ an ensemble of networks (five to ten), trained independently. Use the arithmetic average of the outputs to assign the class, as in [3].

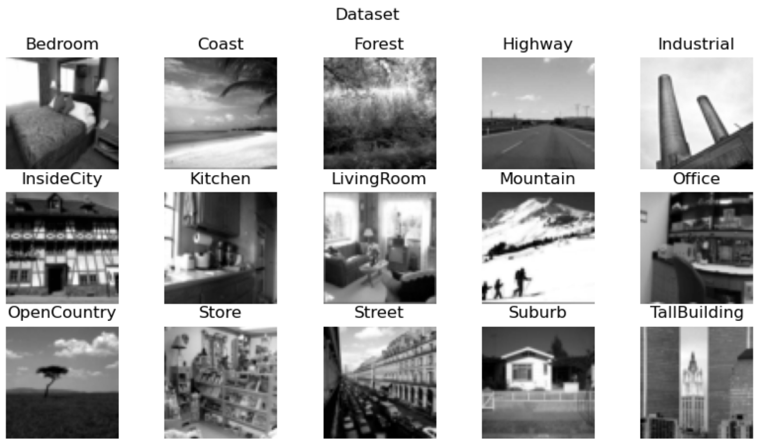
   It's required to reach an accuracy of around 60%

3. Use transfer learning in two manners:

    - Freezing the weights of all the layers but the last fully connected layer and fine-tuning the weights of the last layer based on the same train and validation sets employed before;
    - Employing the pre-trained network as a feature extractor, accessing the activation of an intermediate layer and training a multiclass linear SVM.

Dataset

The Dataset is a collection of gray-scale images of 15 different categories: 'Bedroom', 'Coast', 'Forest', 'Highway', 'Industrial', 'InsideCity', 'Kitchen', 'LivingRoom', 'Mountain', 'Office', 'OpenCountry', 'Store', 'Street', 'Suburb', 'TallBuilding'. It consists of 1500 images for the training set and 2985 for the testing set, making up a total of 4485 images around 300 images for each of the 15 categories. In the training set each category is represented by exactly 100 images, while in the testing set this number varies. The training set was eventually divided into 85% for actual training and 15% for validation, so the actual training was performed on 1275 images.



Implementation

The models were implemented in the Python programming language, mainly using the PyTorch library. They were run on Trieste Area Science Park's cluster ORFEO, using the GPU nodes that mount 2 Intel Xeon Gold 6226 12-Core CPUs (256GB of RAM) and 2 NVIDIA V100 GPUs; the main computations were performed by the GPU.

Part 1

The

Part 2

Part 3

Results

Bibliography

[1] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[2] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.