A

PROJECT REPORT ON

GALAXY MORPHOLOGY CLASSIFICATION USING MACHINE LEARNING

Submitted by

YASH ANIL TAMBE

Roll No - 201503115

Under the guidance of

Ms. T. N. PHALKE

In the partial fulfillment of B. Tech. in Electronics and Telecommunication Engineering course of Dr. Babasaheb Ambedkar Technological University, Lonere (Dist. Raigad) in the academic year 2018-2019.

Department of Electronics and Telecommunication Engineering

Dr. Babasaheb Ambedkar Technological University

Lonere-402103

2018-2019

A

PROJECT REPORT ON

GALAXY MORPHOLOGY CLASSIFICATION USING MACHINE LEARNING

Submitted by

YASH ANIL TAMBE

Roll No - 201503115

Under the guidance of

Ms. T. N. PHALKE

Department of Electronics and Telecommunication Engineering

Dr. Babasaheb Ambedkar Technological University

Lonere-402103

2018-2019

Dr. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY

"VIDYAVIHAR", LONERE- 402103. Tal. Mangaon, Dist. Raigad. (Maharashtra State) INDIA

CERTIFICATE

This is to certify that the Project entitled **Galaxy Morphology Classification Using Machine Learning** submitted by **Yash A. Tambe** is record of bonafide work carried out by him under my guidance in the partial fulfillment of the requirement for the award of Degree of B. Tech. in Electronics and Telecommunication Engineering course of Dr. Babasaheb Ambedkar Technological University, Lonere (Dist. Raigad) in the academic year 2018-2019.

Ms. Tejashree N. Phalke
Project Guide
Department of Electronics and
Telecommunication Engineering
Dr. Babasaheb Ambedkar Technological
University, Lonere-402103

Dr. S. L. Nalbalwar
Professor and Head
Department of Electronics and
Telecommunication Engineering
Dr. Babasaheb Ambedkar Technological
University, Lonere-402103

External Examinar: .....

Date:
Place: Lonere, Dist. Raigad

Acknowledgements

It gives me immense pleasure to present my report for project on **Galaxy Morphology Classification Using Machine Learning**. The able guidance of all teaching staff of this department made the study possible. They have been a constant source of encouragement throughout the completion of this project. I would like to express my grateful thanks to **Dr. S. L. Nalbalwar** who has motivated me and to **Ms. T. N. Phalke** who guided properly for this Project. I would also like to express my sincere thanks to Electronics and Telecommunication Department for giving me an opportunity to explore the subject by conducting this Project.

Yash Anil Tambe (201503115)
Department of Electronics and Telecommunication
Dr. Babasaheb Ambedkar Technological University,
Lonere-Raigad

Project Overview

In this project I aim to apply machine learning techniques to the problem of Galaxy Morphology classification. My aim is to train deep learning models for the above problem using different number of layers and varying parameters of optimization and loss and find out the best performing model.

Contents

List of Figures

Chapter 1

INTRODUCTION

Machine learning is a part of artificial intelligence that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed. There are hundreds of billions of galaxies scattered throughout the cosmos which define the structure of the universe on the largest scales. The distribution of the physical properties of these galaxies hold the clues necessary for our understanding of the past, present, and future of the universe. This project will be based on Image Processing and Machine Learning to classify the galaxies according to their morphology using dataset from Sloan Digital Sky Survey (SDSS) and the Galaxy Zoo Citizen Science Astronomy Project.

Chapter 2

LITERATURE SURVEY

1. In paper [1] Karen Simonyan, Andrew Zisserman, In this work they investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Their main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 1619 weight layers. These findings were the basis of their ImageNet Challenge 2014 submission, where their team secured the first and the second places in the localisation and classification tracks respectively. They also showed that their representations generalise well to other datasets, where they achieve state-of-the-art results.

2. In paper [2] Sander Dieleman, Kyle W. Willett and Joni Dambre , They present a deep neural network model for galaxy morphology classification which exploits translational and rotational symmetry. It was developed in the context of the Galaxy Challenge, an international competition to build the best model for morphology classification based on annotated images from the Galaxy Zoo project. For images with high agreement among the Galaxy Zoo participants, their model was able to reproduce their consensus with near-perfect accuracy (99%) for most questions. Confident model predictions are highly accurate, which makes the model suitable for filtering large collections of images and forwarding challenging images to experts for manual annotation. This approach greatly reduces the experts' workload without affecting accuracy.

3. In paper [3] Nour Eldeen Khalifa , Mohamed Hamed Taha , Aboul Ella Hassanien , IbrahimSelim In this paper, a robust deep convolutional neural network architecture for galaxy morphology classification is presented. A galaxy can be classified based on its features into one of three categories (Elliptical, Spiral, or Irregular) according to the Hubble galaxy morphology classification from 1926. The proposed convolutional neural network architecture consists of 8 layers, including one main convolutional layer for feature extraction with 96 filters and two principle fully connected layers for classification. The architecture is trained over 4238 images and achieved a 97.772% testing accuracy. In this version, Deep Galaxy V2, an augmentation process is applied to the training data to overcome the overfitting problem and make the proposed architecture more robust and immune to memorizing the training data. A comparative result is present, and the testing accuracy was compared with those of other related works. The proposed architecture outperformed the other related works in terms of its testing accuracy.

Chapter 3

MACHINE LEARNING

3.1 What is Machine Learning?

Machine learning (ML) is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) from data, without being explicitly programmed.

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders, and computer vision.

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

3.2 Machine Learning Tasks

Machine learning tasks are typically classified into several broad categories.

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.
- Semi-supervised learning: The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.
- Active learning: The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- Reinforcement learning: Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

3.3 Examples and Applications

Machine learning is being used in a wide range of applications today. One of the most well-known examples is Facebook's News Feed. The News Feed uses machine learning to personalize each member's feed. If a member frequently stops scrolling to read or like a particular friend's posts, the News Feed will start to show more of that friend's activity earlier in the feed. Behind the scenes, the software is simply using statistical analysis and predictive analytics to identify patterns in the user's data and use those patterns to populate the News Feed. Should the member no longer stop to read, like or comment on the friend's posts, that new data will be included in the data set and the News Feed will adjust accordingly.

Machine learning is also entering an array of enterprise applications. Customer relationship management (CRM) systems use learning models to analyse email and prompt sales team members to respond to the most important messages first. More advanced systems can even recommend potentially effective responses. Business intelligence (BI) and analytics vendors use machine learning in their software to help users automatically identify potentially important data points. Human resource (HR) systems use learning models to identify characteristics of effective employees and rely on this knowledge to find the best applicants for open positions.

Machine learning also plays an important role in self-driving cars. Deep learning neural networks are used to identify objects and determine optimal actions for safely steering a vehicle down the road.

A few applications of machine learning include:

- Agriculture
- Automated theorem proving
- Adaptive websites
- Affective computing

- Bioinformatics
- Brainmachine interfaces
- Cheminformatics
- Classifying DNA sequences
- Computational anatomy
- Computer Networks
- Telecommunication
- Computer vision, including object recognition
- Detecting credit-card fraud
- General game playing
- Information retrieval
- Internet fraud detection
- Computational linguistics
- Marketing
- Machine learning control
- Machine perception
- Automated medical diagnosis
- Computational economics
- Natural language processing
- Natural language understanding

- Online advertising
- Recommender systems
- Robot locomotion
- Search engines
- Sentiment analysis (or opinion mining)
- Sequence mining
- Software engineering
- Speech and handwriting recognition
- Financial market analysis
- Structural health monitoring
- Syntactic pattern recognition
- Time series forecasting
- User behavior analytics
- Machine translation

3.4 Challenges and limitations

The two biggest, historical (and ongoing) problems in machine learning have involved overfitting (in which the model exhibits bias towards the training data and does not generalize to new data, and/or variance i.e. learns random things when trained on new data) and dimensionality (algorithms with more features work in higher/multiple dimensions, making understanding the data more difficult). Having access to a large enough data set has in some cases also been a primary problem.

One of the most common mistakes among machine learning beginners is testing training data successfully and having the illusion of success; Domingo (and others) emphasize the importance of keeping some of the data set separate when testing models, and only using that reserved data to test a chosen model, followed by learning learning on the whole data set.

When a learning algorithm (i.e. learner) is not working, often the quicker path to success is to feed the machine more data, the availability of which is by now well-known as a primary driver of progress in machine and deep learning algorithms in recent years; however, this can lead to issues with scalability, in which we have more data but time to learn that data remains an issue.

3.5 Artificial Neural Networks and Deep Learning

3.5.1 Artificial Neural Networks

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An

artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers.

Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology.

3.5.2 Deep Learning

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output.

For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence

the name "deep" networks. The goal is that eventually, the network will be trained to decompose an image into features, identify trends that exist across all samples and classify new images by their similarities without requiring human input.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didnt accurately recognize a particular pattern, an algorithm would adjust the weights. That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modeling. Long short-term memory is particularly effective for this use.

Convolutional deep neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).

Chapter 4

GALAXY CLASSIFICATION

4.1 How are galaxies classified?

Galaxy morphological classification is a system used by astronomers to divide galaxies into groups based on their visual appearance. There are several schemes in use by which galaxies can be classified according to their morphologies, the most famous being the Hubble sequence, devised by Edwin Hubble and later expanded by Grard de Vaucouleurs and Allan Sandage.

4.1.1 Hubble Sequence

The Hubble sequence is a morphological classification scheme for galaxies invented by Edwin Hubble in 1926. It is often known colloquially as the Hubble tuning fork diagram because of the shape in which it is traditionally represented.

To this day, the Hubble sequence is the most commonly used system for classifying galaxies, both in professional astronomical research and in amateur astronomy.

Hubbles scheme divides galaxies into three broad classes based on their visual appearance (originally on photographic plates):

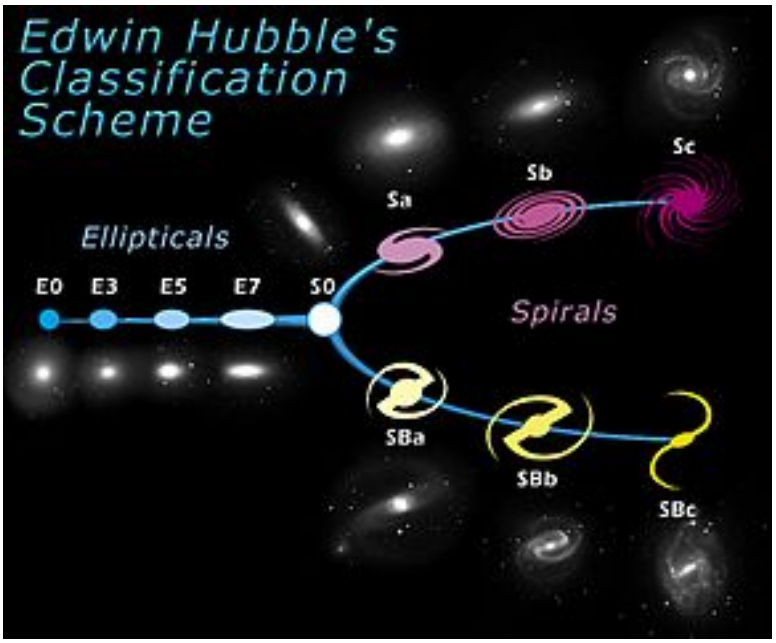

Figure 4.1: Hubble Sequence

- Elliptical galaxies have smooth, featureless light distributions and appear as ellipses in images. They are denoted by the letter "E", followed by an integer n representing their degree of ellipticity on the sky.

- Spiral galaxies consist of a flattened disk, with stars forming a (usually two-armed) spiral structure, and a central concentration of stars known as the bulge, which is similar in appearance to an elliptical galaxy. They are given the symbol "S". Roughly half of all spirals are also observed to have a bar-like structure, extending from the central bulge. These barred spirals are given the symbol "SB".

- Lenticular galaxies (designated S0) also consist of a bright central bulge surrounded by an extended, disk-like structure but, unlike spiral galaxies, the disks of lenticular galaxies have no visible spiral structure and are not actively forming stars in any significant quantity.

These broad classes can be extended to enable finer distinctions of appearance and to encompass other types of galaxies, such as irregular galaxies, which have no obvious regular structure (either disk-like or ellipsoidal).

The Hubble sequence is often represented in the form of a two-pronged fork, with the ellipticals on the left (with the degree of ellipticity increasing from left to right) and the barred and unbarred spirals forming the two parallel prongs of the fork. Lenticular galaxies are placed between the ellipticals and the spirals, at the point where the two prongs meet the handle.

4.2 Importance of Galaxy Classification

Understanding how and why we are here is one of the fundamental questions for the human race. Part of the answer to this question lies in the origins of galaxies, such as our own Milky Way. Yet questions remain about how the Milky Way (or any of the other 100 billion galaxies in our Universe) was formed and has evolved. Galaxies come in all shapes, sizes and colors: from beautiful spirals to huge ellipticals. Understanding the distribution, location and types of galaxies as a function of shape, size, and color are critical pieces for solving this puzzle.

With each passing day telescopes around and above the Earth capture more and more images of distant galaxies. As better and bigger telescopes continue to collect these images, the datasets begin to explode in size. In order to better understand how the different shapes (or morphologies) of galaxies relate to the physics that create them, such images need to be sorted and classified.

4.3 Use of Machine Learning

The use of machine learning to classify galaxies has been an active area of research over the past two decades. Some of the early works attempted to use neural networks, decision trees, and Naive-Bayes classifiers on relatively small datasets of hundreds of

objects At best, these attempts achieved classification errors of 20%. Recently, more advanced techniques have been successfully applied to larger datasets, including the GZ dataset. A neural network was trained on 900,000 objects from GZ using a novel set of features and achieved over 90% classification accuracy . Another group used convolutional neural nets to classify a subset of the 50,000 brightest objects in the GZ dataset with over 99% accuracy. While these results are impressive, the vast majority of classifiers in the literature classified objects into roughly three bins: elliptical galaxies, spiral galaxies, and other. In order to take full advantage of the science contained in the SDSS, LSST, and other surveys, more detailed object classifications need to be made. Prior attempts at extending analyses in this way have performed extremely poorly. For instance, Calleja and Olac Fuentes (2004) found that just going from 3 to 5 categories dropped the accuracy of their neural network from 90% to 50% .

Chapter 5

PROPOSED SYSTEM

5.1 Project Overview

Application of machine learning techniques to the problem of galaxy morphology classification. Use artifical neural networks, especially convolution neural network methods to study the Galaxy Zoo dataset of pre-classified galaxies will be done. After image pre-processing, we will perform multi-class classification. Galaxies will be classified as either spiral, elliptical, round, disk, or other and find the best performing algorithm. We will predict the probabilities of galaxies being associated with each class.

5.2 Block Diagram Of Proposed System

The proposed method for classifying galaxy images will consist of three parts:

- Image Analysis Module (IAM)
- Data Compression Module (DCM)
- Machine Learning Module (MLM)

The method will work as follows:It takes as input the galaxy images, which are then randomly rotated and centred and cropped in the IAM. Next, in the DCM, the images

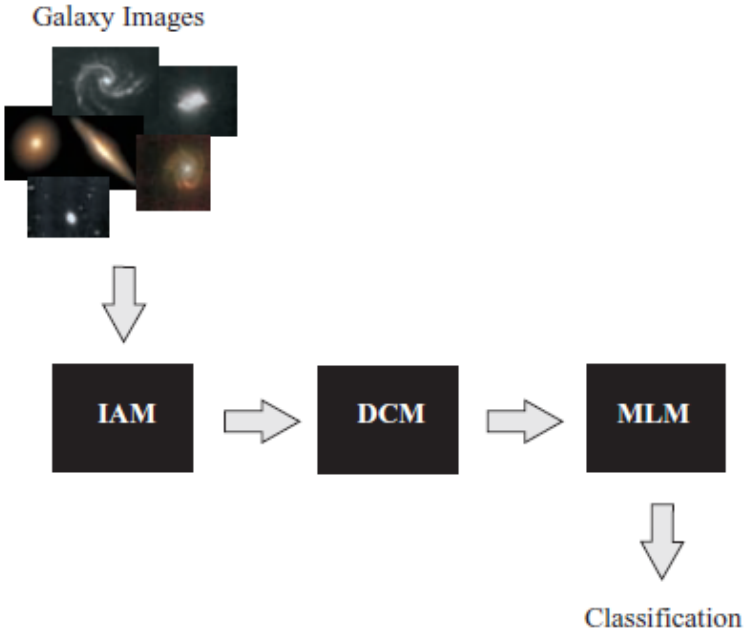

Figure 5.1: Block Diagam of Proposed System

ae cropped to reduce the number of parameters and then in the MLM, machine learning algorithms will be applied.

5.3 Technologies and Software used

- Python
- Jupyter Notebooks

Chapter 6

LIBRARIES AND SOFTWARE USED

6.1 GOOGLE COLAB NOTEBOOK

Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

Whether a student interested in exploring Machine Learning but struggling to conduct simulations on enormous datasets, or an expert playing with ML desperate for extra computational power, Google Colab is the perfect solution. Google Colab or the Colaboratory is a free cloud service hosted by Google to encourage Machine Learning and Artificial Intelligence research, where often the barrier to learning and success is the requirement of tremendous computational power.

6.1.1 Benefits of Colab

Besides being easy to use, the Colab is fairly flexible in its configuration and does much of the heavy lifting.

- Python 2.7 and Python 3.6 support.
- Free GPU acceleration.
- Pre-installed libraries: All major Python libraries like TensorFlow, Scikit-learn, Matplotlib among many others are pre-installed and ready to be imported.
- Google Colab notebooks are stored on the drive.

6.2 Numpy

Numpy is a math library for python. It enables us to do computation efficiently and effectively. It is better than regular python because of its amazing capabilities.

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

6.2.1 Benefits of Numpy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions.
- Tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities.

Installation:Mac and Linux users can install NumPy via pip command:–¿ pip install numpy .

6.3 Pandas

Pandas stands for Python Data Analysis Library

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Whats cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example.

This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

6.3.1 Benefits of Pandas

- Loading and Saving Data with Pandas.
- Viewing and Inspecting Data.
- Selection of Data.
- Filter, Sort and Groupby.
- Data Cleaning.

6.4 Scikit-learn

It is a free software machine learning library for the Python programming language.

Scikit-learn is one the most popular ML libraries. It supports many supervised and unsupervised learning algorithms. Examples include linear and logistic regressions, decision trees, clustering, k-means and so on.

It builds on two basic libraries of Python, NumPy and SciPy. It adds a set of algorithms for common machine learning and data mining tasks, including clustering, regression and classification. Even tasks like transforming data, feature selection and ensemble methods can be implemented in a few lines.

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

6.4.1 Scikit-learn features

- Classification.
- Regression.
- Clustering.
- Dimensionality reduction.
- Preprocessing.

6.5 Matplotlib

It takes you through the basics Python data visualization: the anatomy of a plot, pyplot and pylab, and much more.

This library is very flexible and has a lot of handy, built-in defaults that will help you out tremendously.

you need to make the necessary imports, prepare some data, and you can start plotting with the help of the plot() function! When youre ready, dont forget to show your plot using the show() function.

6.5.1 Matplotlib features:

- Plot creation.which could raise questions about what module you exactly need to import (pylab or pyplot?), how you exactly should go about initializing the figure and the Axes of your plot, how to use matplotlib in Jupyter notebooks, etc.
- Plotting routines,from simple ways to plot your data to more advanced ways of visualizing your data.
- Basic plot customizations, with a focus on plot legends and text, titles, axes labels and plot layout.
- Saving, showing, clearing, your plots: show the plot, save one or more figures to, for example, pdf files, clear the axes, clear the figure or close the plot, etc.
- you can customize Matplotlib: with style sheets and the rc settings.

6.6 Tensorflow

It is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning

applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015

The interesting thing about Tensorflow is that when you write a program in Python, you can compile and run on either your CPU or GPU. So you dont have to write at the C++ or CUDA level to run on GPUs.

It uses a system of multi-layered nodes that allows you to quickly set up, train, and deploy artificial neural networks with large datasets. This is what allows Google to identify objects in photos or understand spoken words in its voice-recognition app.

6.7 Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is Franois Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library. Chollet explained that Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used. Microsoft added a CNTK backend to Keras as well, available as of CNTK v2.0

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier.

In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling

Chapter 7

TRAINING THE MODEL

The model was trained on 6655 images using Google Colab, tensrflow-gpu and keras.

7.1 Dataset

The model was developed in the context of the Galaxy Challenge, an online international competition organized by Galaxy Zoo, sponsored by Winton Capital, and hosted on the Kaggle platform for data prediction contests. It was held from December 20th, 2013 to April 4th, 2014. The goal of the competition was to build a model that could predict galaxy morphology from images like the ones that were used in the Galaxy Zoo 2 project.

Figure 7.1: A Sample Galaxy Image

Images of galaxies and morphological data for the competition were taken from the Galaxy Zoo 2 main spectroscopic sample. Galaxies were selected to cover the full observed range of morphology, colour, and size, since the goal was to develop a general algorithm that could be applied to many types of images in future surveys. The total number of images provided is limited both by the imaging depth of SDSS and the elimination of both uncertain and over represented morphological categories as a function of colour (primarily red elliptical and blue spiral galaxies). This helped to ensure that colour is not used as a proxy for morphology, and that a high-performing model would be based purely on the images' structural parameters.

The final training set of data consisted of 61,578 JPEG colour images of galaxies, along with probabilities6 for each of the 37 answers in the decision tree. An evaluation set of 79,975 images was also provided, but with no morphological data the goal of the competition was to predict these values. Each image is 424 by 424 pixels in size. The morphological data provided was a modified version of the weighted vote fractions in the GZ2 catalog; these were transformed into cumulative probabilities that gave higher weights to more fundamental morphological categories higher in the decision tree. Images were anonymized from their SDSS IDs, with any use of metadata (such as colour, size, position, or redshift) to train the algorithm explicitly forbidden by the competition guidelines.

Galaxies in this set were already classified once through the help of hundreds of thousands of volunteers, who collectively classified the shapes of these images by eye in a successful citizen science crowd-sourcing project. However, this approach becomes less feasible as data sets grow to contain of hundreds of millions (or even billions) of galaxies. That's where machine learning comes in.

This competition was to analyze the JPG images of galaxies to find automated metrics that reproduce the probability distributions derived from human classifications. For each galaxy, determine the probability that it belongs in a particular class.

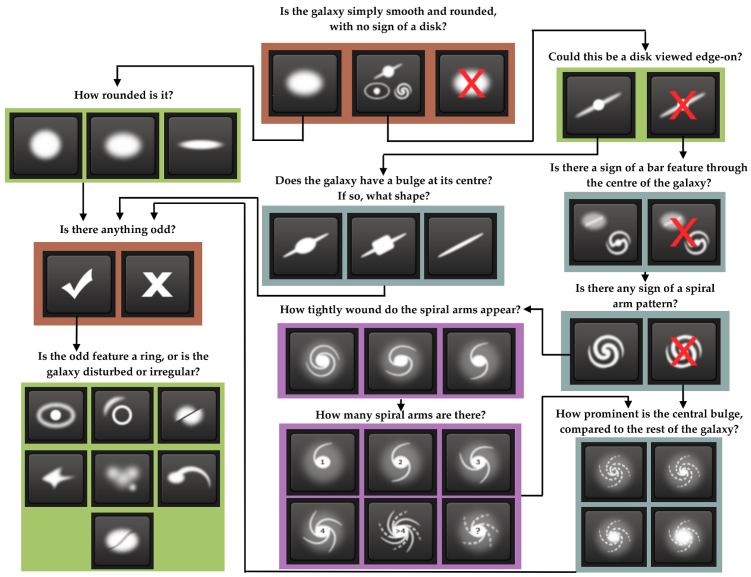

Figure 7.2: Galaxy Zoo Decision Tree

	question		answers	next
Q1	Is the galaxy simply smooth and rounded, with no sign of a disk?	A1.1	smooth	Q7
		A1.2	features or disk	Q2
		A1.3	star or artifact	end
Q2	Could this be a disk viewed edge-on?	A2.1	yes	Q9
		A2.2	no	Q3
Q3	Is there a sign of a bar feature through the centre of the galaxy?	A3.1	yes	Q4
		A3.2	no	Q4
Q4	Is there any sign of a spiral arm pattern?	A4.1	yes	Q10
		A4.2	no	Q5
Q5	How prominent is the central bulge, compared with the rest of the galaxy?	A5.1	no bulge	Q6
		A5.2	just noticeable	Q6
		A5.3	obvious	Q6
		A5.4	dominant	Q6
Q6	Is there anything odd?	A6.1	yes	Q8
		A6.2	no	end
Q7	How rounded is it?	A7.1	completely round	Q6
		A7.2	in between	Q6
		A7.3	cigar-shaped	Q6
Q8	Is the odd feature a ring, or is the galaxy disturbed or irregular?	A8.1	ring	end
		A8.2	lens or arc	end
		A8.3	disturbed	end
		A8.4	irregular	end
		A8.5	other	end
		A8.6	merger	end
		A8.7	dust lane	end
Q9	Does the galaxy have a bulge at its centre? If so, what shape?	A9.1	rounded	Q6
		A9.2	boxy	Q6
		A9.3	no bulge	Q6
Q10	How tightly wound do the spiral arms appear?	A10.1	tight	Q11
		A10.2	medium	Q11
		A10.3	loose	Q11
Q11	How many spiral arms are there?	A11.1	1	Q5
		A11.2	2	Q5
		A11.3	3	Q5
		A11.4	4	Q5
		A11.5	more than four	Q5
		A11.6	can't tell	Q5

Figure 7.3: All questions that can be asked about an image

7.2 Image Pre-processing

Before the images were used to train the model, they were cropped to 212x212 around the centre parameters. They were then randomly rotated to avoid overfitting.

7.3 Best performing model

The best performing model was found to be VGG16 like architecture of 16 layers with Mean Squared Error loss function and RMSprop optimizer. It got test set accuracy of about 60.76%.

7.3.1 Typical CNN strructure

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.

- Convolutional: Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters.

- Pooling: Convolutional networks may include local or global pooling layers. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional

layer. In addition, pooling may compute a max or an average. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Average pooling uses the average value from each of a cluster of neurons at the prior layer.

- Fully Connected: Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

7.3.2 VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 33 kernel-sized filters one after another.

My model, like the VGG16 model consisted of 16 layers - 13 convolutional layers and 3 fully connected layers. Input to first convolution layer is of size 212 x 212 x 3 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3 x 3 (which is the smallest size to capture the notion of left/right, up/down, center). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3 x 3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2 x 2 pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 37-way classification and thus contains 37 channels (one for each class).

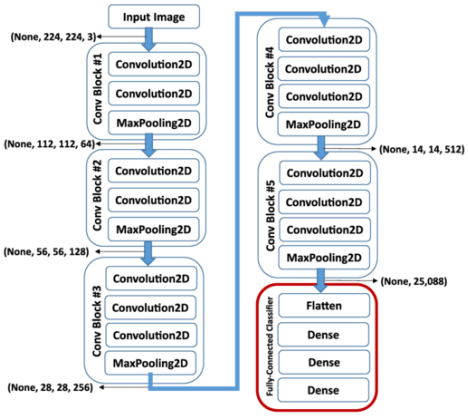

Figure 7.4: Schematic Diagram of VGG16

The final layer is the sigmoid layer.All hidden layers are equipped with the rectification (ReLU) non-linearity.

Layer (type)	Output Shape
conv2d_1 (Conv2D)	(None, 212, 212, 64)
conv2d_2 (Conv2D)	(None, 212, 212, 64)
max_pooling2d_1 (MaxPooling2	(None, 106, 106, 64)
conv2d_3 (Conv2D)	(None, 106, 106, 128)
conv2d_4 (Conv2D)	(None, 106, 106, 128)
max_pooling2d_2 (MaxPooling2	(None, 53, 53, 128)
conv2d_5 (Conv2D)	(None, 53, 53, 256)
conv2d_6 (Conv2D)	(None, 53, 53, 256)
conv2d_7 (Conv2D)	(None, 53, 53, 256)
max_pooling2d_3 (MaxPooling2	(None, 26, 26, 256)
conv2d_8 (Conv2D)	(None, 26, 26, 512)
conv2d_9 (Conv2D)	(None, 26, 26, 512)
conv2d_10 (Conv2D)	(None, 26, 26, 512)
max_pooling2d_4 (MaxPooling2	(None, 13, 13, 512)
conv2d_11 (Conv2D)	(None, 13, 13, 512)
conv2d_12 (Conv2D)	(None, 13, 13, 512)
conv2d_13 (Conv2D)	(None, 13, 13, 512)
max_pooling2d_5 (MaxPooling2	(None, 6, 6, 512)
flatten_1 (Flatten)	(None, 18432)
dense_1 (Dense)	(None, 4096)
dense_2 (Dense)	(None, 4096)
dense_3 (Dense)	(None, 37)

Figure 7.5: Model Summary

Chapter 8

CONCLUSION

In this project I have successfully created and trained deep learning models for galaxy morphology classification.The best performing model was found to be VGG16 like architecture of 16 layers(13 convolutional layers and 3 fully connected layers) with Mean Squared Error loss function and RMSprop optimizer. It achieved test set accuracy of about 60.76% after 20 epochs.

Bibliography

[1] Karen Simonyan, Andrew Zisserman, **Very deep convolutional networks for large scale image recognition** , *arXiv:1409.1556*

[2] Sander Dieleman, Kyle W. Willett and Joni Dambre, **Rotation-invariant convolutional neural networks for galaxy morphology prediction** , *Mon. Not. R. Astron. Soc. 000, 1-20 (2014)*

[3] Nour Eldeen Khalifa , Mohamed Hamed Taha , Aboul Ella Hassanien , Ibrahim Selim, **Deep Galaxy V2: Robust Deep Convolutional Neural Networks for Galaxy Morphology Classifications**, *2018 International Conference on Computational and Information Sciences*