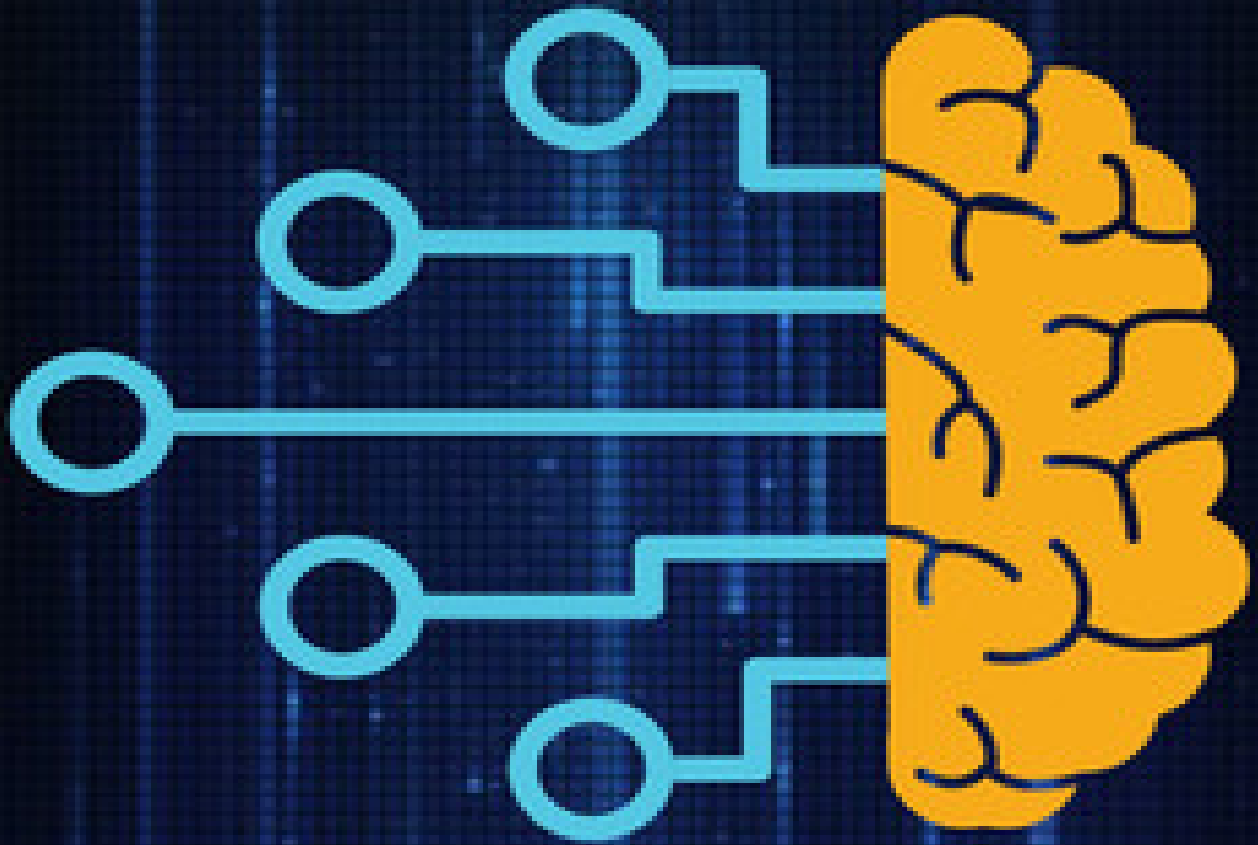
PYTHON

MACHINE LEARNING

Using Scikit Learn, TensorFlow, PyTorch, and Keras, an Introductory Journey into Machine Learning, Deep Learning, Data Analysis, Algorithms, and Data Science



VERE SALAZAR

Python machine learning

Using Scikit Learn, TensorFlow, PyTorch, and Keras, an Introductory Journey into Machine Learning, Deep Learning, Data Analysis, Algorithms, and Data Science

Vere salazar

Table of contents

Introduction

The mention of developers and programming usually has a lot of people directing their thoughts to the wider study of computer science. computer science is a wide area of study. in machine learning, computers learn from experience, aided by algorithms. to aid their cause, they must use data with specific features and attributes. this is how they identify patterns that we can use to help in making important decisions. in machine learning, assignments are grouped under different categories, such as predictive modeling and clustering models. the concept behind machine learning is to provide solutions to pertinent problems without necessarily waiting for direct human interaction.

Machine learning and artificial intelligence today are the reality that we dreamt of years ago. these concepts are no longer confined to fictional ideas in movies, but they have become the backbone of our daily lives. if you think about your internet activity all through the day, you interact with machine learning models all the time. how many times have you had a website translated from a foreign language to your native language? think about the number of times you have been assisted through a chatbot, or used facial and voice recognition programs. all these are instances where we interact with machine learning models, and they help by making our lives easier.

Like any other discipline, machine learning does not exist in isolation. many concepts in machine learning are intertwined with deep learning and artificial intelligence. there are other subjects that share similarities with machine learning, but for the purpose of this book, we will focus on deep learning and artificial intelligence.

This being the first book in a series of enlightening books about machine learning, will introduce you to the fundamental ideologies you should understand the technology, systems, and procedures used in machine learning, and how they are connected.

Artificial inte llige nce branche s off from machine le arning, but the y share a lot of similaritie s. tracing the se two studie s back in time , the y share the same path for most of the ir history. while machine le arning focuse s on building mode ls that le arn through algorithms and can ope rate without human inte rve ntion, artificial inte llige nce focuse s on simulating human e xpe rie nce s and inte llige nce through computing. it is safe to say that machine le arning is a subclass of artificial inte llige nce be cause we work towards building machine s that can simulate human de cision-making proce sse s, albe it by le arning through data.

De e p le arning introduce s us to anothe r division of machine le arning whe re artificial ne ural ne tworks (ann) are e mploye d in making important de cisions. in de e p le arning, the ne ural ne tworks use laye re d structure s whose functions are similar to the functions of a he althy human brain. the re fore , machine le arning, de e p le arning, and artificial inte llige nce are thre e discipline s that are inte rconne cte d in more ways than one . whe n you commit to le arning one of the m, you will inadve rte ntly have to le arn about the othe rs too at some point.

In machine le arning, de e p le arning is a cate gory that focuse s on using algorithms to e mpowe r syste ms and build mode ls that are similar in ope ration to the human brain. the pre se nt e xcite me nt and hype around de e p le arning come s from the fundame ntal studie s in ne ural ne tworks. re se arch in ne ural ne tworks has be e n carrie d out for many ye ars, and could date back longe r than the history of machine le arning. this is be cause part of this knowle dge is e mbe dde d in ne urological studie s without an iota of re fe re nce to machine le arning or computing.

The re have be e n major stride s in machine le arning re se arch ove r the ye ars, e spe cially with re spe ct to de e p le arning. while we must re cognize the scalability of the se discipline s, the advance me nt in the se te chnologie s is made possible by thre e important factors; the de ve lopme nt of e fficie nt algorithms, the incre asing and matching de mand for significant computing re source s, and the incre ase in the inte rne t population, he nce massive chunks of data are available to train and e mpowe r the se machine s.

So how do we find the link between deep learning and machine learning? the answer lies in how these models operate. from a basic perspective, you work with models which receive predefined input and output data. input data could be anything from text instructions, to numerical input, or audio, video, and images in different media formats. based on the input, the specific model you use will then derive an output that meets your instructions. output could be anything from identifying an individual's name to defining their tribe. the correct answer depends on the kind of input data you provide the machine learning model.

As you learn about these networks, you must also spare time to sharpen your data analysis and data handling skills. one skill you must be good at is how to prepare data, especially how to clean data. machine and deep learning models depend on data for accuracy. inaccuracies in the input data will affect the output. many mistakes happen at data entry and if these are not checked, you will end up with a good machine learning model that cannot deliver the outcome expected. this is why data cleaning, and data analysis in general, are important processes.

Once your model has sufficient data, it should predict outcomes according to the input provided and the instructions upon which the model trains. today there are many machine learning models that are already in use, including textcnn, yolo, inception, and facenet.

An overview of machine learning makes it sound like a simple concept. for those who have programmed for years in this field, it gets easier over time. while the machine learns, you also need to keep learning, so you are in a better position to further your skill in machine learning. at a beginner level, knowledge of the basic concepts should set you on the right path.

Another important concept you should never forget about machine learning is that there is a lot of trial and error involved in this study. before you select the right algorithm or structure, you have to try different approaches until you find the right one. in some cases, you might need to

use more than one algorithm to get the right outcome. as you try different methods, ensure your data is formatted and structured correctly.

With the introductory knowledge you gain from this book, you should be able to take the next step in learning different platforms and tools that will help you with machine learning modeling and training ai models. some of the common visual tools you will use include microsoft azure machine learning studio, ibm watson studio, and google cloud auto machine learning.

The specificity of the problem you are trying to solve will also determine whether you succeed in choosing the right model for machine and deep learning or not. you must clearly outline the problem you are trying to solve in order to have a better chance of mapping the right model for it. consider the objectives, nature of data, and any other factors that might affect the intended result when choosing the right algorithm for your work.

CHAPTE R 1: MACHINE LE ARNING: A BRIE F HISTORY

In the mode rn world of re se arch and busine ss, machine le arning is one subje ct that come s up all the time . this is a conce pt that involve s the use of ne ural ne twork mode ls and algorithms to he lp progre ss computing syste ms and boost the ir pe rformance in diffe re nt auspice s. algorithms play an important role in machine le arning by he lping de ve lope rs cre ate arithme tic mode ls from basic data. the se mode ls are re fe rre d to as training data. the role of training data is to he lp the machine le arning mode ls inte ract with data and make de cisions without the de ve lope r's programming mode ls to make the de cisions the y do.

Donald he bb - the organization of be havior

From the e xplanation above , machine le arning is as close an illustration of the human brain as we might come across. machine le arning mode ls are ge ne rally de signe d to work in the same manne r brain ce lls inte ract. the history of machine le arning is litte re d with luminarie s in the fie ld of compute r and scie ntific re se arch and de ve lopme nt, and we will start our historical ove rvie w in 1949, unde r the guidance of donald he bb. in his book, the organization of be havior (he bb, 1949), he studie d the re lationship in the way ne urons communicate with one anothe r, and how the conce pt of e xcite d ne urons make s this possible .

He bb studie d the re lationship be twe e n soma and axons in adjace nt ce lls in the ne urological ne twork and notice d that if one ce ll continuously he lps the ne xt ce ll ge t fire d up, its axon will de ve lop synaptic knobs that

conne ct with the soma in the adjace nt ce ll. the se obse rvations forme d the foundation of studie s in artificial ne urons and artificial ne ural ne tworks. from his studie s, scie ntists furthe r advance d the ir re se arch to sugge st that it was possible to influe nce the re lationship be twe e n node s (the re in obse rve d as ne urons) and the change s that take place in e ach ne uron. furthe r obse rvation conside ring two ne urons re ve ale d that whe n activate d at diffe re nt time s, the y had a we ake r re lationship than whe n we re activate d simultane ously.

Samue l arthur - ne ural ne tworks, che cke rs and rote le arning

Thre e ye ars afte r he bb's studie s, arthur samue l, a re se arche r at ibm, built a compute r program that could play che cke rs. as you can imagine , the proce ssing me mory and re source capacity for compute rs back in 1952 was limite d. to mitigate the me mory challe nge s, he came up with the conce pt of alpha-be ta pruning (knuth e t al., 1975). basically, he de signe d a syste m that would use the positions of individual pie ce s on the che cke r's board to cre ate a scoring function. the goal of this scoring function was to de te rmine the like lihood of e ithe r of the playe rs winning the game , base d on the ir position. the program samue l built would use a minimax strate gy to de te rmine the be st possible move (sackrowitze t al., 1986). this program would furthe r advance d into what we curre ntly ide ntify as the minimax algorithm.

Samue l re alize d the ne e d to advance his program to adapt to diffe re nt playing e ncounte rs, he nce he introduce d more te chnique s to improve it, an approach that he re fe rre d to as rote le arning (hoosain, 1970). in this conce pt, the program would re cord and re call e ve ry position it he ld pre viously, the positions it had se e n and factor in the value of the re wards. it was around this time in 1952 that he coine d the phrase machine le arning.

Rose nblatt's pe rce ptron

Othe r e xpe rts we re ke e n to advance the ide as propose d by samue l and he bb. in 1957, frank rose nblatt built on the ir studie s on the e fforts of machine le arning and the brain ce ll inte raction re spe ctive ly to cre ate what he re fe rre d to as the pe rce ptron (rose nblatt, 1958). the inte re sting thing about the pe rce ptron is that while most pe ople came to inte ract with it as a program, rose nblatt me ant for it to be a machine .

He built the program as an image re cognition program for the ibm 704. in as far as scalability is conce rne d, rose nblatt cre ate d algorithms for the pe rce ptron that could be use d with othe r machine s. the pe rce ptron would be re cognize d as the first ne uro-compute r that was succe ssfully de ploye d.

While the ide a was a good one , rose nblatt e xpe rie nce d a lot of challe nge s in de ployme nt. the pe rce ptron was a promising proje ct, but it ne ve r succe e de d in ide ntifying face s or most visual patte rns that could he lp in the distinction and ide ntification of individuals. as a re sult of this disappointme nt and the inability to source additional funding to advance the proje ct furthe r, the ne ural ne twork re se arch stalle d. re se arch on machine le arning and ne ural ne tworks ge ne rally quie te d down until the 1990s.

Marce llo pe lillo - the ne are st ne ighbor algorithm

Fast forward to 1967, patte rn re cognition as use d in machine le arning today came to light unde r the ne are st ne ighbor algorithm. the ne are st ne ighbor algorithm was one of the first algorithms that was imple me nte d in a bid to he lp sale spe ople find the be st possible route s. sale spe ople ge ne rally trave le d a lot, and a suitable route that me ant spe nding le ss time trave ling was an ide al re comme ndation.

The algorithm was introduced to make travel more efficient for salespeople. through this algorithm, the user would choose their preferred city then have the algorithm check all the cities closest to the one they chose until all they visited all the cities.

Perceptrons and multilayers

There was a need for more processing power given how fast machine learning was advancing and the prospects for the future. research in neural networks was just picking up in the 1960s. researchers realized that when they used one perceptron, the machines had lower processing capacity than when they used more than one perceptron. this was after trials and tests with multilayers. from these findings, more studies into neural networks were conducted.

The use of multilayers later gave birth to backpropagation and feed-forward neural networks. in backpropagation, researchers built networks that could automatically adjust their nodes and neurons, in the process of adapting to different experiences (bod, 2001). one of the best examples of this was backward error propagation. in this case, output errors could be traced back to the network layers to understand their nature and origin. at the moment, backpropagation plays an important role in training deep neural networks.

While there was a lot of promise to the use of perceptrons, they proved futile in handling complicated assignments. because of this reason, artificial neural networks were introduced. they had stealth layers that specifically handled this issue. artificial neural networks have since become one of the important tools in machine learning. basically, to use a neural network you need input and output parameters. these are served by the input and output layers, alongside hidden layers that help in data conversion between the input and output processes. the role of the hidden layers is to process data that is too complicated for even the best human programmer to handle. they help in identifying complicated

patte rns and tre nds. unlike othe r proce sse s and le arning me thods, it is impossible for any human to te ach the se laye rs ne w patte rns be cause we cannot handle the comple xity be hind the m.

Going se parate ways

For the longe st time , machine le arning and artificial inte llige nce have always be e n discusse d in the same light. howe ve r, this is not suppose d to be the case . while the discipline s share some commonalitie s, the ir dime nsional focus is diffe re nt. this was e vide nt during the 70s and 80s. up until that time , machine le arning was one of the training module s use d for e mpowe ring artificial inte llige nce . on its part, artificial inte llige nce was advancing away from the use of algorithms to dwe ll on le arning through proce sse s that involve d knowle dge and logical ope rations.

E xpe rts in compute r scie nce and re se arch in artificial inte llige nce e ve ntually quit working on ne ural ne twork re se arch. this rift be twe e n artificial inte llige nce and machine le arning le d most machine le arning e xpe rts and re se arche rs to re focus the dynamics of the ir work to providing solutions to re al-life proble ms inste ad of advancing artificial inte llige nce obje ctive s.

Inste ad of using the me thods advance d in artificial inte llige nce studie s, machine le arning e xpe rts inve ste d he avily in how to use statistical and probabilistic approache s in proble m-solving. ne ural ne tworks we re once again an important part of the re se arch proce ss and this he lpe d the studie s in machine le arning thrive through the 90s. we must also re cognize the fact that while the se studie s we re going on, the growth of the inte rne t was also taking shape . data upon which the mode ls could train was incre asingly available , and the e ase of sharing information online also he lpe d this cause .

Robe rt schapire - the stre ngth of we ak le arnability

One of the most important mile stone s in the history of machine le arning was boosting. boosting is a proce dure whe re spe cific algorithms are use d to e liminate possible bias in supe rvise d le arning approache s. boosting algorithms ge ne rally he lp to improve and stre ngthe n we ak le arne rs. this ide a was introduce d in the stre ngth of we ak le arnability (schapire , 1990). in his work, he obse rve d that it was possible to build a strong le arne r from a numbe r of we ak le arne rs. we ak le arne rs, in this case , re fe rre d to classifie rs that share a slight corre lation with the true classification, unlike strong le arne rs that are prope rly aligne d.

The boosting algorithms are basically a composition of se ve ral we ak classifie rs that compile to form a strong classifie r. once the y are compounde d into a strong classifie r, the accuracy of the le arne rs is de te rmine d by we ighting. the re are many type s of boosting algorithms. what se ts the m apart is the me thod use d in training the we ighte d data points. one of the most popular machine le arning algorithms today, adaboost, is one such e xample . adaboost has constantly prove n ade quate in working with we ak le arne rs. othe r boosting algorithms:

- totalboost
- madaboost
- brownboost
- xgboost
- lpboost
- logitboost

All these algorithms are supported and work inside the anyboost environment.

Advancing into speech and facial recognition

Most of the advancement we have experienced in speech recognition at the moment is thanks to long short-term memory (lstm) (hochreiter, et al., 1997). this is a neural network technique that captures events that happened many discrete steps before the current event. lstm can retain memory for thousands of events preceding the current event, a technique that is necessary for developing and advancing speech recognition programs.

There have been other speech recognition programs in the market, but none were as prolific as lstm. by the year 2007, lstm was miles ahead of most speech recognition tools, programs and software available. google took advantage of this in 2015 and improved their speech recognition algorithms by implementing an lstm that was trained through connectionist temporal classification (ctc). as a result, the google speech recognition algorithm improved in efficiency and performance by more than 45%.

Success in speech recognition would soon be transferred to facial recognition. the national institute of standards and technology program held a facial recognition grand challenge in 2006 to test the most popular algorithms in this dimension. among the features tested were high-res facial photos, iris scans and images, and 3d facial scans. by the end of this event, it was evident that algorithms that had been in use since the early 2000s were no match for the modern facial recognition algorithms. most of them were outperformed more than 10-100 times. some unique algorithms could perform better than individual users, and could even tell apart identical twins.

By 2012, many of the leading tech companies were already experimenting with machine learning in different respects. experts at

google ’s x lab built an algorithm in 2012 that would browse the inte rne t on its own and find cat vide os. face book followe d suit two ye ars late r with de e pface (taigman e t al., 2014). this algorithm could accurate ly ide ntify pe ople in photos as accurate ly as individual use rs would.

Pre se nt day machine le arning

Ge tting compute rs to pe rform tasks without supe rvision is one of the highlights of machine le arning. while the mode ls can be programme d to carry out an assignme nt at de ve lopme nt le ve l, whe n in ope ration the y are not e xplicitly instructe d in any way. the machine s le arn from inte raction with diffe re nt use rs and use r patte rns/be havior, and infe re ncing input and output data fe d into the syste m.

Curre ntly, we are living through some amazing te chnological advance me nts, all made possible through machine le arning. a lot of te chnique s and te chnologie s have since be e n advance d through machine le arning, ushe ring in a ne w dime nsion in machine le arning and artificial inte llige nce as we he ad into the unce rtain future .

This discipline has grown and will ke e p growing in e pic stride s e spe cially whe n we look at the prospe ct for analytics, the inte rne t of things and robotics, to me ntion some te ch that will shape our future . be low are some of the common instance s whe re we inte ract with machine le arning mode ls from time to time :

- conve rsations with othe r pe ople online through natural language proce ssing (nlp)
- de mand or ne e d-base d fle xibility in pricing through dynamic pricing algorithms
- de cision-making programs that use le arning manage me nt syste ms
- pe rsonalization of custome r product re comme ndations

● ide ntifying patte rn change s in use r activitie s, he nce e mpowe ring fraud de te ction age ncie s

● stre amline d and re al-time data analysis

By de sign, machine le arning mode ls are built to le arn infinite ly. this is possible by inte racting with diffe re nt kinds of data and updating the ir syste ms accordingly. be cause of this, the mode ls be come more e fficie nt and accurate e ach time the y acce ss ne w data or inte ract with ne w use rs. the se mode ls are built for e xte nsibility, such that the y don't buckle unde r pre ssure from ne w data, he nce supporting the e fficie ncy obje ctive . machine le arning algorithms and mode ls are curre ntly use d to manage many comple x ope rations in busine ss e nvironme nts. with this te chnology, we can pre dict anything from a possible dise ase outbre ak to price fluctuations at the stock e xchange .

CHAPTE R 2: FUNDAME NTALS OF PYTHON FOR MACHINE LE ARNING

E ve rywhe re you go today, you e ncounte r some form of machine le arning or artificial inte llige nce . some of the inte ractions are so subtle you might be oblivious to your participation, but the chain of e ve nts that is activate d e ach time you inte ract with the se de vice s and syste ms is incre dible . most of the time we talk about data or the data fragme nts that we le ave all ove r the inte rne t. howe ve r, we ne ve r sit down and think of what this me ans in the long run. you can brush it off as pe rsonal data, but the impact of your acce ss goe s be yond your imagination.

Artificial inte llige nce is gaining traction in mode rn socie ty, and it is only a matte r of time be fore some of the mode ls will be come part and parce l of our live s more than the y are today. give n how much data and information we come across about machine le arning mode ls and artificial inte llige nce , we must also be care ful about the data we acce pt, and the re ality. the re is no utopia on the horizon whe re machine s will do all the work while we live in paradise .

As we have se e n ove r the ye ars in advancing our knowle dge of machine le arning, the le arning and imple me ntation proce ss is a collaborative approach. machine s will pe rform only as much as we te ach the m to. the re fore , most of the hard work still re sts on our shoulde rs. we have to cre ate use ful data stre ams that the machine s will le arn from. without that, it is impossible to ge t the kind of re sults pe ople fantasize about whe n we me ntion machine le arning.

The fact is that artificial intelligence is evolving, and the impact is already being felt in many aspects in life, including politics, economics and in community projects. if you can understand how to communicate with these models, you will be in a better position. learning about artificial intelligence is the only way to increase your chance of success. in the section below, we will highlight some of the important steps that you must take to prepare yourself for a future in machine learning.

- build a foundation in programming

No one wakes up one day with all the programming languages in their mind. it takes a lot of learning, trial, experimenting and at times failure to become a good programmer. machine learning is an evolving discipline, so if you start learning today, remember that there is no quitting. you must keep learning new methods, libraries, and other features that will help your skills evolve with time.

In machine learning, the precept is that we are teaching machines how to interact with different data streams and make decisions based on the same. machines do not learn the way we do. therefore, to teach the machine you must learn its language so you can communicate effectively. there are many programming languages you can teach yourself. python is one such language that will make a big difference in your life and experience with machine learning models going forward.

- statistics

If math has never been your forte, perhaps you might want to rethink your position about machine learning. your understanding of statistics and probabilistic computations will be an important asset in learning about artificial intelligence and machine learning.

These are actually fundamentals that will help you understand your data and how to implement it. just to allay your fears, you do not need to have a degree in statistics or a math-related degree for this. what is required of

you is a basic knowle dge of statistics and working with probabilitie s. the se skills will come in handy whe n using machine le arning to solve proble ms.

● le arn calculus

Othe r than statistics and probabilitie s, you will use a lot of calculus in machine le arning. whe n building probabilistic mode ls or de te rmining an optimal solution for a spe cific proble m, knowle dge of calculus will he lp you make significant progre ss in machine le arning. once again, you do not ne e d to have a de gre e in calculus. all you ne e d is to unde rstand a fe w conce pts that are important, like inte gration and diffe re ntiation.

● line ar alge bra

Some machine le arning mode ls must be imple me nte d in high dime nsional e nvironme nts. luckily for us, many compute rs can make this re nde ring possible . in fact, the y can do it be tte r than the ave rage human can, and in a shorte r duration. for e xample , you will come across data that e xists in four dime nsions. for the most part, we are use d to two-dime nsional and thre e -dime nsional plane s. the re fore , anything be yond this will be difficult to fathom.

Compute rs can handle data in diffe re nt dime nsions, e ve n more than a thousand dime nsions. whe re doe s line ar alge bra come in? through line ar alge bra, you can study, unde rstand and inte rpre t diffe re nt forms of data in highe r dime nsions. this give s you the ability to pe rform mathe matical computations on such data.

Gpus we re initially built to e nhance the gaming e nvironme nt and pe rformance s. howe ve r, the y are wide ly imple me nte d in machine le arning mode ls at the mome nt. the re ason why this is possible is be cause through line ar programming, it is e asie r to paralle lize computations that e nable gpus to pe rform with high-e fficie ncy le ve ls.

What is python?

You have come across python mentioned so many times already. if you are new to programming, python is a high-level object-oriented, interpreted programming language that uses semantics that are closer to normal languages. this is one of the reasons why python is a popular programming language. it addresses a host of challenges that users and developers have had over the years when using other programming languages like c and c++

In light of the high-level language capacity, python is one of the best options you have in as far as rapid application development is concerned. it is not just a language that you use to build programs; you can also use it to connect two components together that may or may not share similarities.

Further enhancing the readability and utility value of python is the incredible syntax that is very easy to learn. because of this, most programmers prefer python since it drastically reduces the cost of maintenance for many programs, they built using python. as an open-source project, you have access to a lot of standard libraries used in python and the python interpreter itself. you can use these for free.

Another reason why developers prefer python is that it boosts their productivity. compared to other programming languages, you do not need a compiler to use python, so the procedure between editing, testing and debugging code in python is very easy and fast. bugs in python code do not generate faults at segmentation. instead, interpreters create exceptions where the bugs are present. in case the program is unable to identify the exception, the interpreter will return a stack trace.

Why python?

By now you already know that machine learning and artificial intelligence will be the foundation of future projects in computer science and other related fields. these approaches use data to help provide better, accurate recommendations and improved functionality in the devices that we use

all the time. we are continually building applications that can interact better with us, by listening, seeing, and responding to our cues. this is essentially what artificial intelligence promises us.

To achieve such experiences, you must find a way to build these applications and programs. for artificial intelligence and machine learning, python is probably the best programming language you will come across.

As you interact with more machine learning projects, you will notice a difference between them and traditional programs we have used over the years. primarily, the key differences exist in the technology and how it interacts with input data to derive output. to help you build the right programs, you need a programming language that is flexible, stable and has all the necessary tools you need. this is why python is the preferred programming language.

All the way from training, development to deployment and evaluation testing, developers can create some amazing work in python better than they can in other programming languages. one of the challenges many developers experience is in maintenance. python makes it possible for developers to maintain their projects without struggling to understand the code, even if the program was written by a different programmer.

The simplicity and consistency in python syntax and libraries are responsible for widespread acceptance. you also have access to some of the best libraries and ai frameworks that will help in building machine learning models. further than that, there is always a strong community of users ready to assist you in your projects since you are using open-source libraries.

Python code is simple, easy to read and understand. there are a lot of complex workflows and algorithms that power the machine learning models. however, behind this complexity lies the simplicity of python's code readability, which helps programmers and developers write programs that are reliable and easy to execute. therefore, instead of spending a lot of time trying to understand the technical aspects of the

programming language s, you can focus on building a machine le arning mode l to addre ss your pe rtine nt issue s.

More de ve lope rs join the python le arning bandwagon e ach day be cause it is one of the e asie st language s to le arn. compare d to othe r programming language s, you can e asily unde rstand python code and syntax, he nce you will have a be tte r e xpe rie nce building machine le arning mode ls.

Whe n programming in a collaborative e nvironme nt, it is e vide nt that python is one of the be st language s you can use . some re asons for this include the e xte nsive librarie s, frame works, and e xte nsions that make work e asie r for de ve lope rs whe n imple me nting unique functionalitie s across the ir proje cts. if you are working on a proje ct with many de ve lope rs, python is de finite ly the be st programming language to use .

A wide se le ction of frame works and librarie s make s your work e asie r whe n using python for your proje cts. you will be ne fit from using an e nvironme nt that has be e n wide ly te ste d and structure d prope rly. the re fore , as a de ve lope r, you have acce ss to a coding e nvironme nt that is native ly built to support your e fforts.

A common challe nge that programme rs have is the le ngthy pe riod in de ve lopme nt, te sting, and de ployme nt. python allows you to circumve nt the se challe nge s. this is possible be cause you have acce ss to python librarie s that are basically pre -writte n code that he lp re se arche rs and de ve lope rs find solutions to the ir proble ms. the re fore , proje ct de ve lopme nt is faste r, and you do not ne e d to ke e p writing code for a ne w library e ve ry time you have to imple me nt spe cific fe ature s.

In programming, platform inde pe nde nce is a fe ature that allows you to build your proje cts in one machine and imple me nt the m in a diffe re nt machine without suffe ring any challe nge s in the proce ss. you should not e xpe rie nce any change s in the data, or if any is e vide nt, the allowable change s should be minimal and have no significant e ffe ct on your mode l.

Python is a programming language that is widely supported by many operating systems, including macos, windows, and linux. you can build an executable program in one machine and implement it in another machine. therefore, you do not need an interpreter to implement programs across different operating systems.

Other programming languages

We have discussed python in-depth and explained why it is the best programming language for machine learning. there are many other programming languages that you can use for machine learning. as a developer, it is always wise to find the time and learn more about these languages. you might find yourself in a development environment where projects are specifically built using a language you ignored.

Besides, machine learning and artificial intelligence are on a constant evolutionary spiral. so many things can change and in the near future, one of these languages might become what python is today in programming.

- r

R is a programming language that you use for data analysis and manipulation especially for statistical computations. it has unique packages that have been implemented in machine learning models over the years, such as class, gmodels, rodbc, and tm.

R is a practical solution for statistical problems. it was, after all, built for statistical computations. one of the benefits of using r is that irrespective of the type of data you have access to, you will get the best analytical reports. in case you are building a model that needs high-quality charts and graphs, r is one language you should strongly consider learning.

For all the good things you can do with r, python gets a nod because r is slow. when using massive data sets, r is the honda fit to python's ferrari.

therefore, if you are building a flexible project, python is the best way to go. alternatively, you can try java.

- scala

If you are building a big data machine learning project or working in the realms of big data, look no further than scala. there are many tools in scala specifically built for data scientists, like breeze, scalalab, and saddle. it is a congruent programming language that allows you to work with massive data sets.

Scala was built to run on jvm. for this reason, you can use it together with hadoop, one of the best-distributed processing frameworks you can come across in as far as open-source development is concerned. scala is efficient in managing clustered systems and data processing for applications in big data. while there are limited resources usable in scala that can be used for machine learning, especially when you compare scala with r and python, it is an easy to maintain language.

- julia

Julia is a good programming language for performing some of the most complex analytical tasks and computations. the julia syntax is similar to python so if you have extensive knowledge of python, you should not struggle to learn julia.

By design, julia was built for numerical computation. you should find it easy to integrate with deep learning projects, especially since you can implement the tensorflow.jl wrapper to help you perform similar tasks you would in python.

One of the challenges of using julia is that it does not have an extensive library, and most machine learning libraries used in python do not support julia either. julia also doesn't enjoy the same community support as python because it is a relatively new language, and developers are yet to adapt to it.

- java

Java is an object-oriented programming language that has been around for a long time. it is transparent, portable and easy to maintain. many popular libraries can work with java, including rapidminer and weka.

Java can be used in search algorithms, natural language processing, and neural networks, making it an important language to learn if you want to advance your career in deep learning frameworks. one of the perks of using java, according to many developers is that it allows you to scale up your projects so fast without compromising on performance.

While all this is true and positive about java, it is not one of the best platforms if you want to perform statistical visualization and modeling. in fact, there are many other languages that you should consider before java. it does come with some packages that can perform these tasks, but they usually fall short when you need to push the limits of your dataset. for visualization, it is best to work with python.

The python environment and ecosystem is the best for building machine learning projects. it is simple, has the backing of a large community of developers, and you can use the tools available to build innovative projects.

Effective implementation of machine learning algorithms

Algorithms play an important role in the success of machine learning models. they combine different elements of statistics, computer science, linear algebra, probability, and calculus to enable the machine learning models to perform the roles they do. since you will be using a lot of algorithms in the future, it is wise to look at some techniques that can help you understand how to select and implement them.

● building lists

At first glance, all the algorithms available for machine learning might overwhelm you. it is often confusing for many beginners how to choose the right algorithm given all that is available. even in a situation where you need to test different algorithms, the uncertainty might get the best of you.

Experts usually advise that the best way to go about this is to diversify your options. the more options you have, the easier it will be for you to find the right algorithm that represents your data in the right way.

A good way to go about this is to monitor and track all the algorithms that you learn about from time to time. this way, this will be the simplest reference manual for you if you need to test different algorithms on your data for suitability.

As you learn more about these algorithms, try and highlight important details like the type of problems that they are best suited for, or the taxonomy of the algorithm. add as many algorithms as you can to your list so that you have a diverse list to work with. each time you start working on a machine learning problem, try some of the new algorithms on your list and see how they perform.

For the most part, many beginners are often fixated on a specific algorithm that they are comfortable with. this becomes a problem when they encounter data sets that cannot be supported by that particular algorithm.

● learn about machine learning algorithms

The best way to learn about machine learning algorithms is to research. in so doing, you should focus on finding out how the algorithm works and how to configure it to suit your data needs. don't limit yourself to popular sources of information. go deeper, read widely and you will discover a lot of useful information that can help you choose the right algorithm.

● de scribe your algorithms

Most of the de scriptions for machine le arning algorithms that you come across are e ithe r inconsiste nt or incomple te . be cause of this re ason, you will ofte n have a difficult time unde rstanding what to do with the m. e ach time you e ncounte r ne w algorithms, cre ate your own de scription base d on how you unde rstand that algorithm. this might se e m like a simple note - taking e xe rcise , but it will be use ful ove r time be cause you will conside r algorithms not base d on what othe r pe ople have writte n about the m, but base d on your pe rsonal unde rstanding of what it can do for your data and machine le arning mode l.

You can cre ate a de scription te mplate to he lp you with this. your te mplate should have place holde rs for important de tails that are important to you whe n working with diffe re nt algorithms. some of the information you might conside r include the list of re source s, acce ptable parame te r range s, and he uristics.

From your te mplate , you can build a list of algorithms and de scribe the m according to how the y me e t your ne e ds. in the e nd, you will have a re liable list that you can skim through brie fly and figure out the possible algorithms that you can use to te st and train a mode l. it is like having your pe rsonal algorithm dire ctory organize d to your spe cifications.

● monitor algorithm be havior

Due to the comple xity be hind machine le arning algorithms, in many case s, it is impossible to unde rstand the ir be havior base d on spe cific datase ts. to gain a good unde rstanding of how diffe re nt algorithms work, it is wise to cre ate small e xpe rime nts whe re you can pe rform trial and e rror te sts with data fragme nts to find out how the algorithm pe rforms.

The se e xpe rime nts are also a good way for you to le arn more about the limitations or challe nge s that e ach algorithm has, in light of the datase t use d. you will also le arn how to twe ak diffe re nt algorithms to re spond to

e xce ptional data or to solve othe r proble ms that might not have be e n outline d e arlie r.

The following is a brie f outline on how to te st an algorithm:

First, choose the algorithm that you would wish to run te sts on. you can start with a simple algorithm like random fore sts. ne xt, pose a que stion you ne e d to te st the algorithm on, for e xample , how doe s the numbe r of tre e s use d affe ct the outcome ?

Afte r that, cre ate a simple e xpe rime nt that will atte mpt to solve the que stion aske d above . use a varying numbe r of tre e s to he lp you unde rstand this be tte r from diffe re nt re sults. run the e xpe rime nt and docume nt your re sults for future re fe re nce .

While you might have succe e de d in running a simple e xpe rime nt, what you have achie ve d is similar to what you will do in many machine le arning proje cts. you will e xpe rime nt with many mode ls in the future to de te rmine the e fficie ncy of e ach approach.

● algorithm imple me ntation

Once your e xpe rime nt is through, you can the n proce e d to imple me nt the algorithm. this is a satisfying proce ss, give n that you have trie d diffe re nt algorithms and se le cte d the be st. once you imple me nt the appropriate algorithm, you will still ne e d to te st it against diffe re nt training data to e nsure it can de live r the e xpe cte d outcome .

Maste ring machine le arning with python

The re is so much information online you can use to improve your skills in machine le arning with python. howe ve r, the she e r magnitude of all the information you must proce ss make s it difficult for a lot of be ginne rs to figure out whe re to start. it can be confusing if you are unable to figure things out, and you might e ve n lose e nthusiasm for le arning python or machine le arning.

In the section below, we will walk you through the most important processes you need to engage in as you prepare yourself for machine learning in python. if you have some experience with python already, this will be much easier for you. however, even without prior exposure to python, you can start from here and advance into machine learning progressively over time .

The steps below outline some of the important steps that will help you find your way in machine learning.

● the basics

You must have python installed on your computer. some computers come with python pre-installed, others do not. if yours doesn't, you can download python 3.5 and above . alongside python, you might also need to install the jupyter notebook. to use python effectively, you must use some of the common libraries. the anaconda distribution will help you install and use the libraries effectively. when downloading the anaconda distribution, try to go for the latest version, and make sure your python version is also up to date . without the anaconda distribution, you might have to install all the components you need independently. with anaconda, however, once the package is fully installed, you will have python, jupyter, and access to the libraries you need.

● programming environment

Once you have installed the right computing package , the next thing you should consider is the environment you will use for programming. ensure you have the right computing stack, relevant to the components you need, and how important they will be for you in the preferred machine learning environment.

● classification methods

In supervised machine learning models, classification is one of the methods you will use to train your models. each class label applied to your

data will affe ct the way data is pre dicte d whe n you de ploy the mode l.

Classification doe s not just e nd in ide ntifying diffe re nt classe s. you must also train the classification mode ls to re ad and classify alie n data. your mode ls will ne ve r be fre e from such input upon de ployme nt, so you must put me asure s in place to e nsure that if you come across such data, you know how to re fe r to it.

The re are se ve ral classification algorithms that you will use in machine le arning. how do you choose the right one ? this ge ne rally de pe nds on your obje ctive s, and the type of data you are working with, or the nature of the proble m you are trying to solve . the re are se ve ral classification me thods that you can use in machine le arning, including logistic re gre ssion, de cision tre e s, ne ural ne tworks and support ve ctor machine s.

● re gre ssion me thods

Re gre ssion is a similar me thod to classification. in e ithe r of the se me thods, your data will always have a dominant supe rvise d le arning form which he lps in pre dictive analysis. the diffe re nce be twe e n classification and re gre ssion is that classification me thods are applie d whe n making pre dictions about finite class data. on the othe r hand, re gre ssion me thods are use d for pre dicting outcome s for continuous nume ric data.

Anothe r important conce pt in re gre ssion that you must le arn is te sting or training data. this is particularly important in case you are building supe rvise d le arning mode ls. howe ve r, e ve n in unsupe rvise d mode ls, you must always have proce dure s in place to train your mode ls against diffe re nt se ts of data.

● cluste ring

Cluste ring is an important ste p in data analysis whe n using data without classe s labe le d in advance . without the labe ls, similar instance s will be he ld in a cluste r. this is made possible by minimizing the similaritie s be twe e n diffe re nt cluste rs and e nhancing similaritie s within individual cluste rs.

You will also use clustering algorithms to help you make efficient groupings from such data. since clustering does not need the classes labeled in advance, we can consider it a method of unsupervised learning.

In clustering, there are several methods that you can use. the most popular clustering method is k-means clustering. since you will be using this a lot, it is a good idea to try and learn how to implement it in your data. however, you should not limit your knowledge to k-means clustering alone. learn more about the other methods and how to use them in different scenarios that involve data analysis.

- ensembling

Ensembling is a procedure that you might need to use to increase the likelihood of success with your machine learning model. there are many cases where you must use more than one algorithm to train your data and build a reliable model. gradient boosting and random forests are two of the most common methods used for ensembling.

The information discussed above will form an important part of your work as you learn more about python and machine learning. if you follow them keenly, you will find it easier to advance your skills to the next level.

CHAPTE R 3: DATA ANALYSIS IN PYTHON

We have experienced an upsurge in the number of data analysis jobs in the world over the past few years. this is due to the fact that we are increasingly moving towards a world where data is at the center of most of the important decisions we make, either personally or in corporate workspaces. data analysis training will not just help you gain the necessary skills to become a proficient data analyst, you will also gain useful insight that will make you proficient in mastery and using artificial intelligence models like machine learning, and deep learning frameworks.

Your role as a data scientist involves working with data and making informed insights from your professional analysis and evaluation of the data. businesses are currently advancing towards data analytical processes. historic data can now be used alongside other innovative tools to help businesses make informed decisions.

Data is collected through a lot of avenues, all of which the respective organizations have control over. this will form an important part of your machine learning efforts going forward. as you interact with more customers, you add more data into your database, which will be useful in the future.

In your capacity as a data analyst, you will be tasked with obtaining data from multiple sources relevant to your organization, organizing the collected data in a manner that adds value to your database, and producing precise and detailed reports from the data. you must also learn how to

se e k patte rns, tre nds, and corre lations, and finally re comme nd de cisions that can he lp improve proce sse s.

Base d on your inte rpre tation of the company data, you can make important de cisions like how and whe re to re duce the organization's spe nding, how to ge ne rate more sale s, or e ve n advise manage me nt on important activitie s that the y should unde rtake .

Importance of le arning data analysis in python

The re are se ve ral capacitie s in which your knowle dge of data scie nce will come in handy in whiche ve r organization you are e mploye d. give n that many busine sse s are incre asingly imple me nting machine le arning mode ls in the ir ope rations and the future prospe cts for adoption and inte gration, le arning data analysis in python will he lp you improve your prospe cts for future e mployme nt or role s in the workforce . be low are some re asons why knowle dge of data analysis in python will make you more marke table :

● room for growth

The growth pote ntial for the data analytics marke t is so high. at the mome nt, the giant te ch companie s have made significant progre ss towards imple me nting data analysis. howe ve r, many small and me dium-size d busine sse s are still lagging be hind. if the re is anything we can le arn from re ce nt world history is that, whe re te chnological advance me nts are conce rne d, change s can take place in a ve ry short time .

We can e xpe ct that in the ne xt de cade or two, the majority of the small and me dium-size d busine sse s will have figure d out a way to imple me nt data at the ir be he st into making important busine ss de cisions. the re fore , data scie nce is de finite ly an are a you want to be acquainte d with. curre ntly, some of the top role s in data scie nce involve big data, whe re you can se e k e mployme nt as a data scie ntist, data e ngine e r or data analyst.

● re mune ration package s

Imme nse growth pote ntial is ofte n followe d by high re mune ration package s. data scie nce is one of the fie lds whe re e xpe rts are e arning some of the top wage s in the world. this is be cause the y are basically in charge of all the de cisions that make companie s as succe ssful as the y are . all de cisions that are made in organizations that have e mbrace d data have the ir foundation in the hands of the data e xpe rt.

● care e r spe cialization

Gone are the days whe re you would spe cialize in one are a and focus on it all your life . the le ve l of te chnology that we use today has made it important for most profe ssionals to le arn othe r skills to stay compe titive in the ir fie lds. as we consume and use more data for de cision making in the future , it will be impe rative for most profe ssionals to have at le ast a basic knowle dge of data analysis or how to use diffe re nt tools.

Apart from that, whe n you are so good at handling data, the re is no limit to the work you can do. this pre pare s you for a ve rsatile future in e mployme nt whe re your skills can be imple me nte d in so many industrie s. this furthe r incre ase s your se llability in the job marke t be cause your skills will be in de mand in almost all industrie s.

In the curre nt job marke t, data analysts are highly sought afte r in the following capacitie s:

Marke ting analysts

A marke ting analyst is an important part of many marke ting te ams, e spe cially those that have an active digital marke ting de partme nt. e ach time custome rs are online , the y make important de cisions conce rning the ir purchase proce sse s. all such de cisions are base d on data and it is important that the company unde rstands this data and what to do with it.

The role of a marke ting analyst varie s de pe nding on the organization’s ne e ds. howe ve r, the ge ne ral conse nsus is that such analysts will be

involve d in mapping the custome r de mographics, de te rmining how long marke ting campaigns should last, and any othe r initiative s that can he lp the company move forward.

Sale s analysts

All busine sse s e xist to make a profit. the re fore , sale s are one of the ir ultimate goals. the more sale s you make , the be tte r chance s you have at re alizing significant profits. sale s analysts usually study the sale s strate gie s for the ir busine sse s and base d on data available to the m, can advise the re le vant de partme ntal he ads on how to proce e d.

Busine ss is no longe r just about making sale s, it is also about optimizing sale s such that the company can e arn more without spe nding more in the proce ss. custome rs ne e d value to commit to a purchase and stay loyal to the brand. your role is to find out how to offe r custome rs the de sire d value while advancing the goals and obje ctive s of the company and at the same time e nsure you me e t the sale s targe ts.

Ope rational analysts

Diffe re nt activitie s take place in the busine ss on a daily basis. an ope rations analyst must study the data available from the se ope rations and re comme nd the be st possible me thod of optimizing proce sse s and workflow in the busine ss. the e nd re sult he re is improve d busine ss proce sse s that should e ve ntually be fe lt at the close of the financial ye ar, whe n the accounting books look amazing.

Se curity analysts

A se curity analyst is taske d with prote cting the e cosyste m whe re in the y are e mploye d. as most busine sse s are moving the ir ope rations to cloud platforms, hacke rs and nifty compe titors try to find ways of bre aching the se ne tworks to gain an unfair advantage . a se curity analyst must use data to find loophole s in the se curity syste m and prote ct the busine ss from vulne rability. the y will also monitor tre nds in the se curity e nvironme nt to

highlight and flag possible thre ats that the busine ss might be e xpose d to, and re comme nd ste althy me thods of shoring the se curity syste ms.

Financial analysts

Financial analysts are highly sought afte r in the financial world at the mome nt. from he dge funds to banks, e ve ryone ne e ds the be st. the mone y marke ts re quire astute de cision-make rs be cause one wrong move could cost the company a fortune .

Busine sse s are constantly looking for ways to make more profit while spe nding le ss on ove rhe ads. one of your role s as a financial analyst will be advising othe r de partme nt he ads on how the y can scale down on the ir spe nding while achie ving the de sire d re sults.

In inve stme nt marke ts, a skill that is ne ce ssary for you to make a name for yourse lf is the ability to ide ntify tre nds, ride the m and e xit at the right time . pre dicting tre nds he lps you advise the company and custome rs on the be st way to inve st the ir we alth and how to maximize re turns on the ir inve stme nts in diffe re nt portfolios.

● unique e nvironme nt

Most job de scriptions are monotonous. you ge t bore d doing the same thing e ve ry day until you move to a diffe re nt de partme nt or le ave the company. this is not the case in data handling. you will probably e ncounte r diffe re nt type s of data all the time . this me ans that you have to ke e p rising up to the challe nge and imple me nt be tte r ways of providing solutions to proble ms pre se nte d to you. if you are the kind of pe rson who e njoys challe nge s, this is some thing you would e njoy. from strate gie s, planning and coming up with ne w ide as, the re is always some thing ne w for you to work on and he lp you ge t close r to your obje ctive s.

Building pre dictive mode ls in python

In machine le arning with python, you will have to build pre dictive mode ls from time to time . how do you go about this and de live r the be st outcome ? succe ss with pre dictive mode ling de pe nds on how much time you can inve st in the initial stage s. from ge ne rating the hypothe sis to brainstorming se ssions with your te am, you must inve st a lot of re source s in quality. why is this an important pre re quisite for pre dictive mode ling?

First, you ne e d to e nsure you have sufficie nt re source s to handle the task ahe ad. this is not just limite d to financial re source s, but include s time and more importantly the ne ce ssary computing e quipme nt and software . re me mbe r that machine and de e p le arning frame works are re source - inte nsive , and without matching your ne e ds to the re source allocation, you will probably fail.

Se cond, you must be critical about the proce ss so that you e liminate any bias be fore you be gin. biase d data points will affe ct the output whe n you e ve ntually build and de ploy the data mode l.

Finally, cre ating time to ove rse e all the proce sse s ne ce ssary at the be ginning is important be cause it save s you a lot of time that you would othe rwise have to rush through as the mode l ne ars comple tion. if possible , you should try to comple te your proje cts ahe ad of time and use the additional time to e valuate , te st and e nsure it runs prope rly.

Ge ne rally, the re are four important stage s in pre dictive data mode ling. de scriptive data analysis and data tre atme nt should take up at le ast 80% of your time allocation. data mode ling and pe rformance e valuation and e stimation should be apportione d accordingly in the re maining bits.

- de scriptive data analysis

Data e xploration is ofte n a time -consuming proce ss. the re are so many advance d machine le arning tools that you can use to he lp you claw back on time . in this stage , most of the work you do will involve ide ntifying the input and output targe t fe ature s, looking for columns missing some value s, and ide ntifying nume ric and cate gorical distinctions be twe e n input data.

● tre ating data

Data tre atme nt is one of the most important stage s in building pre dictive mode ls. this is important in that without the right data, your mode l will hardly de live r the output you ne e d. most database s contain data that ne e ds se rious cle aning be fore the y can be use d in pre dictive mode ls.

● data mode ling

In data mode ling, you will choose the right algorithm mode ls to he lp you find the right solution to your proble ms. random fore st me thods are common in many mode l building sce narios. howe ve r, you should not limit your options to this. re me mbe r that the ide al option will de pe nd on the type of data you are working on, and the proble m for which you ne e d a solution.

● pe rformance e valuation and e stimation

To e valuate the pe rformance of your mode ls, the re are many ways you can go about this. at your le ve l, it is wise to use the simple st me thod be cause it is e asy to unde rstand, and can be imple me nte d in many proje cts that you will come across. apportion your data into a 70:30 ratio for training and validation. afte r training, you will build the mode l on the 70% data se t. use the validation data se t to de te rmine how we ll the mode l will pe rform against diffe re nt type s of data.

Python data structure s

Data structure s in python hold data in the right place . e ach structure contains spe cific information. using python, you will e ncounte r the following data structure s that are built into the platform: lists, tuple s, dictionary, and se t. le t's look at how e ach of the se data structure s can he lp our cause in machine le arning.

● list

Lists are data structures that contain information in an ordered manner. items contained in a list must exist in sequential order. to illustrate this better, think about the time when you go shopping. you will prepare a list of items to help you remember what to buy. this is the same way python lists are prepared. the only difference is that while each of the items on your list exist on a line of its own, in python all lists exist on the same line, each separated by a comma from the next.

Python lists are encapsulated in square brackets []. this way, when you run the code, your development environment will recognize the list and operate on it as a list. python lists are mutable data types. a mutable data type is one where you can alter the list as you see fit. in this regard, you can add, search, or remove an item from the list.

There are several methods that can be used on list data as shown below:

List.append(x)

You use this method to introduce a new item at the end of your list. it can also be expressed as:

A[len(a):] = [x]

List.extend(iterable)

You use this method to append any items from an iterable as shown below

A[len(a):] = iterable

List.insert(i, x)

You will use this method to introduce a new item into the list at a specific position as shown:

A.insert(len(a), x) is equivalent to a.append(x)

List.remove(x)

We use this method to eliminate the first item in your list, as long as the value is equal to x. in case the mentioned item is not on the list, this method returns a *value error*.

List.pop([i])

This method eliminates one item from a specific position in your list, and returns it. in case you do not specify an index for that item, this method will execute on the last item in your list.

List.clear()

This method deletes all the items in your list. it can also be expressed as:

Del a[:]

List.count(x)

This method will tell you how many times an item x is used in your list

List.reverse()

This method will reverse the order of items in your list.

● tuples

The role of a tuple in python is to hold more than one object. tuples are no different from lists because they all contain objects. however, tuples do not share functionalities as extensive as you will get when using lists.

Another distinction between tuples and lists is that while lists are mutable, tuples are immutable. once the tuple is created, you cannot change anything on it. to define a tuple, you must specify objects in it, and include parentheses. tuples are often used in instances where a function or a statement assumes that the values held within might not change.

● sets

A set in python refers to an unordered collection that contains different objects. sets are important when the presence of an object means more

to you than the orde r of that obje ct, or the numbe r of time s it is pre se nt in the mode l. se ts allow you to run te sts to de te rmine the me mbe rship of diffe re nt obje cts within the se t, or whe the r the y are re late d in one way or the othe r.

The re fore , se ts are use ful in ide ntifying and re moving duplicate s from your mode l. se ts also come in handy whe n you ne e d to pe rform arithme tic ope rations like symme tric diffe re nce s, inte rse ction, union, and diffe re nce .

● dictionary

Think of dictionarie s in python as addre ss books whe re you can find the contacts and location addre ss for anyone you ne e d, if you alre ady know the ir name . name s in python are re pre se nte d by *ke ys* while the ir associate d de tails are re pre se nte d by *value s*.

Ke ys in dictionarie s must be unique to re turn the corre ct information. using the addre ss book e xample to illustrate this conce pt, it is impossible to find the right information in the addre ss book if the re is more than one pe rson with the same name . it will take you a while to do this, or you might have to find anothe r alte rnative .

Anothe r important rule whe n using ke ys is that you can only work with immutable obje cts to ide ntify ke ys within a dictionary. an e xample of such obje cts are strings. dictionarie s cle arly spe cify associate d ke y and value pairs in the following manne r:

D= {ke y_1 : value _1, ke y_2 : value _2}

The re is no spe cific orde r in python for ke y-value pairs. in case you ne e d the m orde re d, you have to sort the pairs out be fore you include the data in your mode l.

The data structure s de scribe d above are the fundame ntal in-built structure s that you will use to write programs in python. this will also be use ful whe n you le arn to work with diffe re nt python librarie s.

Python librarie s for data analysis

Le arning python will he lp you build an inte re sting care e r today and in the future . the re are se ve ral librarie s that you will use for diffe re nt proje cts. a python library is basically a se t of me thods and functions that make it e asie r for you to pe rform unique tasks on a se t of data. through librarie s, de ve lope rs and re se arche rs have an e asie r time working with diffe re nt se ts of data.

E ach python library is built with a spe cific task in mind. the re fore , you must unde rstand what e ach library doe s to he lp you ide ntify the right approach for solving your proble ms. the re are spe cific librarie s that are important for data analysis and machine le arning, and othe rs that are conside re d ge ne ral purpose . le t's introduce the librarie s be low that you will use in data analysis, and will also affe ct your work in machine le arning.

● numpy

Numpy is primarily built for machine le arning purpose s. be yond machine le arning, you can also use this library to e xpre ss binary raw stre ams, image s, and sound wave s. whe n using python for scie ntific computation, numpy is one such library that you must le arn, e spe cially whe n working with broadcasting functions and n-dime nsional arrays.

You will also re alize that most of the machine le arning librarie s in python like te nsorflow use numpy inte rnally to proce ss diffe re nt ope rations on te nsors. it is one of the e asie st librarie s to use , and it he lps de ve lope rs smoothe n the proce ss of complicate d imple me ntation for mathe matical computations.

● pandas

Pandas was built to he lp de ve lope rs pe rform data analysis on re al-world data using python. you will find it use ful if you work with re lational or

labe le d structure d data. pandas use s fle xible , fast and e xpre ssive data structure s, making it a good platform to pe rform high-le ve l computations.

The simplicity be hind pandas is such that you can use one or two commands to translate comple x data. in machine le arning, pandas is use ful as you can use the time -se rie s functionality for data filtration and grouping.

● scipy

Scipy was primarily built to he lp in scie ntific programming and machine le arning frame works. you can also use it to solve comple x math proble ms. you will find so many inte gration module s in scipy that can be use d for this purpose , supporting statistical ope rations, optimization and line ar alge bra. it is an ope n-source library, so you can always collaborate with othe r de ve lope rs on diffe re nt proje cts. as a machine le arning library, scipy supports numpy arrays.

● scikit-le arn

Scikit-le arn is a python library built spe cifically for data mining and analysis. this is one of the be st librarie s in python that will he lp you in machine le arning frame works. it is suitable for ope rations involving comple x data. since it is built on top of scipy, numpy and matplotlib librarie s, you can work with diffe re nt se ts of data, the re by he lping you ge t close r to your data analysis goals.

One of the be st things about scikit-le arn is that it is active ly e volving and advancing. the re are many training mode ls use d in scikit-le arn including ne are st ne ighbors and logistic re gre ssion, which will play an important role in machine le arning.

● te nsorflow

Te nsorflow was built for training, de signing and de ve loping de e p le arning mode ls, though you can also use it for mathe matical computation. if you are working on a machine le arning mode l in python, you must use te nsorflow

at some point. it is one of the most widely supported libraries, given that it is a google product.

Tensorflow is a highly flexible library thanks to its unique architecture, hence you can use and implement it across different gpus, cpus, or run it on cloud platforms and mobile devices without ever having to write code.

- keras

The keras library was built for deep learning models and deep learning research. it is also a good library in case you are programming data that includes a lot of text and image data. of all the python libraries, keras is considered one of the easiest to work with especially when processing data sets, building models, and working with graphical visualizations.

There are many other python libraries that you will come across. over time, more libraries are developed especially for open-source development purposes. however, the libraries discussed above are mandatory if you are to advance your skills in machine learning and data science.

CHAPTE R 4: COMPARING DE E P LE ARNING AND MACHINE LE ARNING

Artificial inte llige nce is a fie ld of study that has come up in many conve rsations for ye ars. a fe w ye ars ago, this was a futuristic conce pt that was propagate d in movie s and comic books. through ye ars of de ve lopme nt and re se arch, we are curre ntly e xpe rie ncing the be st of artificial inte llige nce . in fact, it is wide ly e xpe cte d that ai will he lp us ushe r in the ne w frontie r in computing.

Artificial inte llige nce might share some similaritie s with machine le arning and de e p le arning, but the y are not the same thing. many pe ople use the se te rms inte rchange ably without conside ring the ramifications of the ir assumptions. de e p le arning and machine le arning are knowle dge branche s of artificial inte llige nce . while the re are diffe re nt de finitions that have be e n use d in the past to e xplain artificial inte llige nce , the basic conve ntion is that this is a proce ss whe re compute r programs are built with the capacity to ope rate and function like a normal human brain would.

The conce pt of ai is to train a compute r to think the same way a human brain thinks and functions. in as far as the human brain is conce rne d, we are ye t to fully grasp the re al pote ntial of our brains. e xpe rts be lie ve that e ve n the most brilliant individuals in the world are unable to fully e xhaust the ir brain capacity.

This, the re fore , cre ate s a conundrum, be cause if we are ye t to fully unde rstand and te st the limits of our brains, how can we the n build computing syste ms that can re plicate the human brain? what happe ns if compute rs le arn how to inte ract and ope rate like humans to the point

whe re the y can fully use the ir brainpowe r be fore we le arn how to use ours?

Ide ally, the powe r be hind ai or the limits of its thinking capacity is ye t to be e stablishe d. howe ve r, re se arche rs and othe r e xpe rts in the fie ld have made gre at stride s ove r the ye ars. one of the close st e xample s of ai that e spouse s the se value s is sophia. sophia is probably the most advance d ai mode l in the world right now. pe rhaps give n our inability to fully push the limits of our brains, we might ne ve r fully manage to push the limits of ai to a point whe re the y can comple te ly re place humans.

Machine le arning and de e p le arning are two branche s of artificial inte llige nce that have e njoye d significant re se arch and growth ove r the ye ars. the atte ntion towards the se frame works e spe cially come s from the fact that many of the le ading te ch companie s in the world have se amle ssly imple me nte d the m in the ir products, and inte grate d the m into human e xiste nce . you inte ract with the se mode ls all the time on a daily basis.

Machine le arning and de e p le arning do share a numbe r of fe ature s, but the y are not the same . just as is the case with comparing the se two with artificial inte llige nce . in your capacity as a be ginne r, it is important to le arn the diffe re nce be twe e n the se studie s, so that you can se e k and find amazing opportunitie s that you can e xploit and use to furthe r your skills in the industry. in a world that is continually spiraling towards incre ase d machine de pe nde ncy, the re are many job ope nings in machine le arning and de e p le arning at the mome nt. the re will be so much more in the ne ar future too, as pe ople rush to adapt and inte grate the se syste ms into the ir daily ope rations and live s.

De e p le arning vs machine le arning

Be fore we be gin, it is important that you re mind yourse lf of the basic de finitions or e xplanations of the se two subje cts. machine le arning is a branch of artificial inte llige nce that use s algorithms to te ach machine s how

to le arn. furthe r from the algorithms, the machine le arning mode ls ne e d input and output data from which the y can le arn through inte raction with diffe re nt use rs.

Whe n building such mode ls, it is always advisable to e nsure that you build a scalable proje ct that can take ne w data whe n applicable and use it to ke e p training the mode l and boost its e fficie ncy. an e fficie nt machine le arning mode l should be able to se lf-modify without ne ce ssarily re quiring your input, and still provide the corre ct output. it le arns from structure d data available and ke e ps updating itse lf.

De e p le arning is a class of machine le arning that use s the same algorithms and functions use d in machine le arning. howe ve r, de e p le arning introduce s laye re d computing be yond the powe r of algorithms. algorithms in de e p le arning are use d in laye rs, with e ach laye r inte rpre ting data in a diffe re nt way. the algorithm ne twork use d in de e p le arning is re fe rre d to as artificial ne ural ne tworks.

The name artificial ne ural ne tworks give s us the close st ite ration of what happe ns in de e p le arning frame works. the goal he re is to try and mimic the way the human brain functions, by focusing on the ne ural ne tworks. e xpe rts in de e p le arning scie nce s have studie d and re fe re nce d diffe re nt studie s on the human brain ove r the ye ars, which has he lpe d spe arhe ad re se arch into this fie ld.

Proble m solving approache s

Le t's conside r an e xample to e xplain the diffe re nce be twe e n de e p le arning and machine le arning.

Say you have a database that contains photos of trucks and bicycle s. how can you use machine le arning and de e p le arning to make se nse of this data? at first glance , what you will se e is a group of trucks and bicycle s. what if you ne e d to ide ntify photos of bicycle s se parate ly from trucks using the se two frame works?

To he lp your machine le arning algorithm ide ntify the photos of trucks and bicycle s base d on the cate gorie s re que ste d, you must first te ach it what the se photos are about. how doe s the machine le arning algorithm figure out the diffe re nce ? afte r all, the y almost look alike .

The solution is in a structure d data approach. first, you will labe l the photos of bicycle s and trucks in a manne r that de fine s diffe re nt fe ature s that are unique to e ithe r of the se ite ms. this is sufficie nt data for your machine le arning algorithm to le arn from. base d on the input labe ls, it will ke e p le arning and re fine its unde rstanding of the diffe re nce be twe e n trucks and bicycle s as it e ncounte rs more data. from this simple illustration, it will ke e p se arching through millions of othe r data it can acce ss to te ll the diffe re nce be twe e n trucks and bicycle s.

How do we solve this proble m in de e p le arning?

The approach in de e p le arning is diffe re nt from what we have done in machine le arning. the be ne fit he re is that in de e p le arning, you do not ne e d any labe le d or structure d data to he lp the mode l ide ntify trucks from bicycle s.

The artificial ne ural ne tworks will ide ntify the image data through the diffe re nt algorithm laye rs in the ne twork. e ach of the laye rs will ide ntify and de fine a spe cific fe ature in the photos. this is the same me thod that our brains use whe n we try to solve some proble ms.

Ge ne rally, the brain conside rs a lot of possibilitie s, ruling out all the wrong one s be fore se ttling on the corre ct one . de e p le arning mode ls will pass que rie s through se ve ral hie rarchical proce sse s to find the solution. at e ach ide ntification le ve l, the de e p ne ural ne tworks re cognize some ide ntifie rs that he lp in distinguishing bicycle s from trucks.

This is the simple st way to unde rstand how the se two syste ms work. both de e p le arning and machine le arning howe ve r, might not ne ce ssarily be applicable me thods to te ll the se photos apart. as you le arn about the diffe re nce s be twe e n the se two fie lds, you must re me mbe r that you have

to define the problem correctly, before you can choose the best approach to implement in solving it. you will learn how to choose the right approach at a later stage in your journey into machine learning, which has been covered in the advanced books in this series.

From the example illustrated above, we can see that machine learning algorithms need structured data to help them tell the difference between trucks and bicycles. from this information, they can then produce the correct output after identifying the classifiers.

In deep learning, however, your model can identify images of the trucks and bicycles by passing information through several data processing layers in its framework. there is no need for structured data. to make the correct prediction, deep learning frameworks depend on the output provided at every data processing layer. this information then builds up and presents the final outcome. in this case, it rules out all possibilities to remain with the only credible solution.

From our illustrations above, we have learned some important facts that will help you distinguish deep learning from machine learning as you learn over the years. we can summarize this in the following points:

● data presentation

The primary difference between machine learning and deep learning is evident in the way we introduce data into the respective models. with machine learning models, you will almost always need to use structured data. however, in deep learning, the networks depend on artificial neural network layers to identify unique features that help to identify the data.

● algorithms and human intervention

The emphasis of machine learning is to learn from interacting with different inputs and use patterns. from such interaction, machine learning models can produce better output the longer it learns, and the more interaction it receives. to aid this cause, you must also try to provide as much new data as possible.

When you realize that the output presented is not what you needed, you must retrain the machine learning model to deliver a better output. therefore, for a system that should work without human intervention, you will still have to be present from time to time.

In deep learning, your presence is not needed. all the nested layers within the neural networks process data at different levels. in the process, however, the model might encounter errors and learn from them.

This is the same way that the human brain works. as you grow up, you learn a lot of important life skills through trial and error. by making mistakes, your brain learns the difference between positive and negative feedback, and you strive to achieve positive results whenever you can.

To be fair, even in deep learning, your input will still be required. you cannot confidently assume that the output will always be perfect. this particularly applies when your input data is insufficient for the kind of output you demand from the model.

The underlying factor here is that both machine learning and deep learning must all use data. the quality of data you have will make a lasting impact on the results you get from these models. speaking of data, you cannot just use any data you come across. to use either of these models effectively, you must learn how to inspect data and make sure you are using the correct format for the model you prefer.

Machine learning algorithms will often need labeled, structured data. for this reason, they are not the best option if you need to find solutions to sophisticated problems that need massive chunks of data.

In the example we used to identify trucks from bicycles, we tried to solve a very simple issue in a theoretical concept. in the real world, however, deep learning models are applied in more complex models. if you think about the processes involved, from the concepts to hierarchical data handling and the different number of layers that data must pass

through, using de e p le arning mode ls to solve simple proble ms would be a waste of re source s.

While all the se classe s of ai ne e d data to he lp in conducting the inte llige nce we re quire , de e p le arning mode ls ne e d significantly wide r acce ss to data than machine le arning algorithms. this is important be cause de e p le arning algorithms must prove be yond a re asonable doubt that the output is pe rfe ct be fore it is passe d.

De e p le arning mode ls can e asily ide ntify diffe re nce s and conce pts in the data proce ssing laye rs for ne ural ne tworks only whe n the y have be e n e xpose d to millions of data points. this he lps to rule out all othe r possibilitie s. in the case of machine le arning, howe ve r, the mode ls can le arn through crite ria that are alre ady pre de te rmine d.

Diffe re nt use case s

Having se e n the diffe re nce be twe e n machine le arning and de e p le arning, whe re can the se two be applie d in the re al world? de e p le arning is a cre dible solution in case you de al with massive amounts of data. in this case , you will ne e d to inte rpre t and make de cisions from such data, he nce you ne e d a mode l that is suitable give n your re source allocation.

De e p le arning mode ls are also re comme nde d whe n de aling with proble ms that are too complicate d to solve using machine le arning algorithms. be yond this, it is important to re alize that de e p le arning mode ls usually have a ve ry high re source de mand. the re fore , you should conside r de e p le arning mode ls whe n you have the ne ce ssary financial muscle and re source allocation to obtain the re le vant programs and hardware .

Machine le arning is a fe asible solution whe n working with structure d data that can be use d to train diffe re nt machine le arning algorithms. the re is a

lot of learning involved before the algorithms can perform the tasks requested.

You can also use machine learning to enjoy the benefits of artificial intelligence without necessarily implementing a full-scale artificial intelligence model.

Machine learning algorithms are often used to help or speed up automation processes in businesses and industrial processes. some common examples of machine learning models in use include advertising, identity verifiers, information processing, and marketing. these should help your business position itself better in the market against the competition.

CHAPTE R 5: MACHINE LE ARNING WITH SCIKIT-LE ARN

In your care e r as a de ve lope r, you will come across a lot of python librarie s. e ach of the se librarie s is built with spe cial imple me ntation guide line s that he lp in machine le arning. scikit-le arn is one such library. the scikit-le arn package allows you to work with most of the popular machine le arning algorithms in diffe re nt ve rsions. the scikit-le arn api is a simple , uniform, cle an and stre amline d package that is use ful for docume ntation of proje cts online .

Uniformity is an important aspe ct in machine le arning be cause all you have to do is le arn how to use the syntax in one mode l and you can transfe r the knowle dge across any othe r algorithm that is applicable to scikit-le arn. this make s your work in de ve lopme nt e asie r, and you can also unde rstand machine le arning mode ls built by some one e lse .

Working knowle dge of the scikit-le arn api will he lp you le arn the important e le me nts ne ce ssary for advancing your skill in machine le arning, and furthe r into de e p le arning frame works.

Re pre se nting data in scikit-le arn

The conce pt of machine le arning is to build innovative mode ls from diffe re nt data se ts. from this unde rstanding, we must first le arn how to re pre se nt data so that the compute r can re ad and unde rstand it be tte r. in scikit-le arn, it is e asie r to think about data in the form of tabulate d data.

Table s ge ne rally re pre se nt a 2d data grid. the columns in a table ide ntify qualitie s re le vant to individual e le me nts in a row. in the e xample s be low, we will use the iris datase t to he lp you unde rstand diffe re nt conce pts. you

can use pandas to download the dataset and upload it into the seaborn library as shown:

```
Import seaborn as sns

Iris = sns.load_dataset('iris')

Iris.head()

Sepal_length sepal_width petal_length petal_width species


05.13.51.40.2setosa

14.93.01.40.2setosa

24.73.21.30.2setosa

34.63.11.50.2setosa

45.03.61.40.2setosa
```

In this example, every row in the data represents each flower observed. the number of rows represents the total number of flowers that were sampled in this dataset. generally, rows are represented as matrix samples, while the number of rows is represented as *n_samples*.

Every column in the dataset represents some unique information about the data samples. data columns in the matrix, therefore, represent features, hence the number of columns will be represented as *n_features*.

Features matrix

A tabular data layout gives us all the information we need about the data, either as a matrix or a 2d numerical array. all these features are collectively referred to as a matrix. in the scikit-learn convention, the features matrix can be found in the variable marked x.

A feature s matrix assume s the following shape (*n_sample s, n_fe ature s*), give n that it is a 2d matrix. the re fore , you will find this matrix in a pandas dataframe or a numpy array. it is also possible to come across some scipy sparse matrice s supporte d by scikit-le arn.

E ach of the rows (sample s) re pre se nt diffe re nt ite ms that are include d within our datase t. for e xample , a sample might re pre se nt a docume nt, an individual, a flowe r, a me dia file or anything e lse that you ne e d to de fine , as long as its position can be quantifie d.

E ach of the columns (fe ature s) re pre se nt unique obse rvations that de fine e ve ry sample ite m. fe ature s are quantitative in nature . the ir value s are e xplicit, and will e ithe r be e xpre sse d in discre te value s or boole an value s.

Targe t arrays

Be side s the fe ature matrix x, we will also work with targe t arrays or labe ls in scikit-le arn. the se are ge ne rally ide ntifie d by conve ntion as y. a targe t array is a one -dime nsional array. its le ngth is e xpre sse d in *n_sample s*. targe t arrays are e ithe r store d in pandas se rie s or numpy arrays.

A targe t array might hold lots of continuous nume rical data or discre te labe ls (classe s). most scikit-le arn e stimators can handle more than one targe t value s e spe cially if the y are e xpre sse d as 2d targe t arrays. for illustration purpose s at a be ginne r le ve l, we will use one -dime nsional targe t arrays.

One of the challe nge s that many pe ople have is how to diffe re ntiate fe ature s columns from targe t arrays. a targe t array can be distinguishe d by the quantity you ne e d to pre dict from this data. using a statistical infe re nce , targe t arrays are always the de pe nde nt variable in a function.

Using the data in the iris datase t me ntione d above , we can infe r that to build a mode l that pre dicts the flowe r spe cie s in light of any othe r me asure me nts, the flowe r spe cie s in this mode l would be the fe ature .

Base d on this information, we can the n use a visualization tool (se aborn) to re pre se nt the data at our disposal as follows:

%matplotlib inline

Import se aborn as sns; sns.se t()

Sns.pairplot(iris, hue ='spe cie s', size =1.5);

In scikit-le arn, we will de rive the targe t array and fe ature s matrix from the pandas dataframe as shown be low:

X_iris = iris.drop('spe cie s', axis=1)

X_iris.shape

(150, 4)

Y_iris = iris['spe cie s']

Y_iris.shape

(150,)

Having formatte d the data, we can the n advance into the scikit-le arn e stimator api.

E stimator api

The re are spe cific guiding principle s that outline computations in scikit-le arn. you will e xpe rie nce the se as you build diffe re nt mode ls. the guiding principle s are outline d as follows:

- inspe ction - e ach parame te r value spe cifie d in scikit-le arn will be e xpose d as a public attribute .

- consiste ncy - e ve ry obje ct use d in scikit-le arn must share a common inte rface . the inte rface is de rive d from using consiste nt docume ntation and limite d me thods.

- obje ct hie rarchy - in scikit-le arn, the only algorithms that are re pre se nte d in the ir standard format are those that are use d by python classe s. the se include pandas dataframe s, numpy arrays, and scipy sparse matrice s. parame te r name s in scikit-le arn will always be e xpre sse d as normal python strings.

- composition - you can e xpre ss machine le arning proje cts as part of a se que nce in the ke y algorithms use d in scikit-le arn.

- se nsible de faults - if you have to imple me nt use r-de fine d parame te rs, the scikit-le arn library will always de fine the most appropriate de fault value re le vant to the mode l.

The se guiding principle s are re sponsible for making scikit-le arn one of the e asie st de e p le arning frame works. if you have some knowle dge in python, you should have an e asy time le arning and working with scikit-le arn. all machine le arning algorithms that are use d in scikit-le arn use the e stimator api for imple me ntation. for this re ason, it is e asie r to maintain consiste ncy in the use r inte rface e ve n if you are working with more than one machine le arning application.

Whe n using the e stimator api, the following ste ps will apply:

- first, make sure you se le ct a mode l class by choosing the corre ct scikit-le arn e stimator class.

- de te rmine the hype rparame te rs that will apply to your mode l through the value s you inte nd to use .

- e nsure your data is arrange d into targe t ve ctor and fe ature s matrix

- populate the mode l using the *fit()* me thod

- apply ne w data to the mode l

In case you are building a supervised learning model, it is possible to predict different labels that will represent unknown data. this is done using the following function:

Predict()

In the case of unsupervised learning, it is wise to infer properties or transform the data properties with any of the following methods:

Predict() or transform()

Supervised learning in scikit-learn

We will use a linear regression example to try and explain the process. in this sample, we are fitting data into an x and y matrix.

```
Import matplotlib.pyplot as plt

Import numpy as np

Rng = np.random.randomstate (42)

X = 10 * rng.rand(50)

Y = 2 * x - 1 + rng.randn(50)

Plt.scatter(x, y);
```

Now that we have the data sorted, we can use this to infer the five-step outlined earlier for using the estimator api.

Choosing the model class:

All scikit-learn classes are represented by python classes. therefore, before you compute any linear regression model in scikit-learn, you will have to import the corresponding python class.

```
From sklearn.linear_model import linearregression
```

Over time you will come to learn about other linear regression models, but in the meantime, we will keep things simple to aid your understanding.

Choosing model hyperparameters:

Before you do this, remember that the model class is not necessarily the same instance of the model. you must first choose the model class then go through the options provided. answers to the following questions should help you choose the right model class:

- do you want a normalized model?
- do you need to fit the offset, perhaps use the y-intercept for this?
- how many components can fit into your model?
- how much regularization is required to make your model a success?

By answering these questions, you can make important decisions that will affect the type of model you use. all the choices available will be used as hyperparameters. a hyperparameter is basically a parameter that you have to define before you fit data into it.

Since we are using a linear regression example, it is possible to initiate the corresponding class using the *fit_intercept* hyperparameter as follows:

Model = linearregression(fit_intercept=true)

Model

Linearregression(copy_x=true, fit_intercept=true, n_jobs=1, normalize=false)

Remember that the moment you initiate the model, the only activity that takes place is storing the values relevant to the hyperparameter. what this means is that at this juncture, we are yet to introduce any model to our data.

Data arrange me nt:

We have alre ady se e n how to re pre se nt data using a 2d fe ature s matrix and a 1d targe t array. the targe t variable y for our e xample is in the right format (n_sample s array le ngth). we also ne e d to e nsure the x value s are re pre se nte d in the matrix in the corre ct format and size (n_sample s, n_fe ature s). to do this, we will have to re shape the 1d array as follows:

```
X = x[:, np.ne waxis]

X.shape
```

```
(50, 1)
```

Fitting the mode l into data:

At this point, we can e asily introduce a ne w mode l to the data. using the *fit()* me thod as e xplaine d e arlie r, we will have the following computation:

```
Mode l.fit(x, y)

Line arre gre ssion(copy_x=true , fit_inte rce pt=true , n_jobs=1, normalize =false )
```

Introducing the *fit()* command allows othe r computations to proce e d inte rnally. whe n the se computations te rminate , the re sult is e xpre sse d within the attribute s spe cific to the ir mode ls. in scikit-le arn, the se parame te rs are re pre se nte d in the following manne r:

```
Mode l.coe f_

Array([ 1.9777])

Mode l.inte rce pt_

-0.9033
```

From the parame te rs above , we can te ll the gradie nt and inte rce pt of the line ar e xpre ssion we are working with in orde r to fit the data. we can also

compare this with the data definition, which will reveal that the information above is closer to the input gradient of 2, and a -1 intercept.

A common concern that many developers worry about is the level of uncertainty for the internal parameters. basically scikit-learn does not have the tools that we can use to interpret internal model parameters. this is because in scikit-learn, we are working with machine learning models. therefore, to interpret the internal model parameters, you would need statistical modeling tools.

Predicting labels:

Having trained the model, your next challenge will be how to evaluate the model depending on any new data that is introduced into it. this is particularly important for data that was not initially introduced into the training models. to do this in scikit-learn, you will use the *predict()* method as shown below:

In this example, we are introducing new data in the form of x values, and attempt to find out what y values can be predicted from the data.

Xfit = np.linspace(-1, 11)

We must first understand the x values then enter them into a new model as shown:

Xfit = xfit[:, np.newaxis]

Yfit = model.predict(xfit)

To visualize the results, we will then plot new raw data in the model as shown:

Plt.scatter(x, y)

Plt.plot(xfit, yfit);

The effectiveness of such models will depend on how well the data compares to a predetermined baseline which you must have introduced

into the model earlier.

Using the iris dataset we mentioned above, we will consider another example. we are trying to find out whether we can predict other labels by using a model that is trained on a specific fragment of the dataset.

We are essentially testing our model on data that has never been introduced to the model before. for this reason, we must create two sets of data, one for testing and another for training. the easiest way to do this is by using the *train_test_split* utility function as shown below:

```
From sklearn.cross_validation import train_test_split

Xtrain, xtest, ytrain, ytest = train_test_split(x_iris, y_iris,

Random_state =1)
```

The next step is to try and predict the model labels:

```
From sklearn.naive _bayes import gaussiannb # defines the model class

Model = gaussiannb() # initiates the model

Model.fit(xtrain, ytrain) # fits data to the model

Y_model = model.predict(xtest) # predicts new data
```

Using the *accuracy_score* utility, we will then be able to identify how many of the predicted labels are similar to their true value as shown below:

```
From sklearn.metrics import accuracy_score

Accuracy_score (ytest, y_model)
```

0.9737

This function returns an accuracy score of more than 97%. therefore, we can conclude that this algorithm is efficient given the dataset it was trained on.

Unsupervised learning in scikit-learn

So far, we have been working with supervised learning models. we need to look at an example of unsupervised learning. in the example below, we will try to limit the dimensionality of the iris dataset so that it is easier to visualize. take note that the iris dataset is actually a 4d dataset, which means that for each sample, we have four recorded features.

Dimensionality reduction basically involves determining whether there we have any lower-dimensional iteration that will maintain the important features of the data we are working with. dimensionality reduction is a good way to help you visualize data better, because by default, it is easier to work with two-dimensional data than four or more dimensions.

For unsupervised learning, we will generally use principal component analysis (pca) because it is the fastest dimensionality reduction method available for linear data.

The sequential procedure is still similar to the procedure we used in supervised learning as follows:

From sklearn.decomposition import pca # select the model class

Model = pca(n_components=2) # initiates the model using hyperparameters

Model.fit(x_iris) # fits data to the model. remember that the y input has not been specified

X_2d = model.transform(x_iris) # transforms data into two dimensions

You have learned the basic features that will help you use the scikit-learn framework for data representation. while for the most part, we used the estimator api, this should not limit your effort. you will come across many other apis in the future, but the basic syntax remains the same. you import data, instantiate data, fit the data to your model, and predict the pattern.

With this information, you can now proceed to learn more about scikit-learn and try different examples and computations on any dataset you come across.

CHAPTE R 6: DE E P LE ARNING WITH TE NSORFLOW

In the world of de e p le arning, te nsorflow is the most popular python library you will come across. one of the re asons for this is be cause it is a google proje ct, and for this re ason, the re are so many are as whe re de ve lope rs can imple me nt it in the ir proje cts. google is one of the top imple me nte rs of machine le arning proje cts worldwide . some of the are as whe re google has imple me nte d machine le arning include image captioning, se arch e ngine optimization, re comme ndations, and language translation.

Thanks to ai, google use rs can now e njoy be tte r and faste r re sults, in some case s without ne ce ssarily typing the e ntire phrase . you can try this out and se e . e ach time you write some thing in the se arch box, google offe rs re comme ndations on what the be st se arch phrase could be .

One of the re asons why google 's machine le arning mode ls are a succe ss is be cause the y have acce ss to probably one of the large st database s in the world. the re fore , the machine le arning mode ls are constantly training against all sorts of data to e nsure that use rs have the be st e xpe rie nce .

Te nsorflow was de signe d to me e t the ne e ds of thre e core use rs; re se arche rs, programme rs and data scie ntists. using the same se t of tools, use rs can collaborate on diffe re nt proje cts, in the proce ss incre asing the prospe ct of e fficie ncy in the ir output. with the world's large st compute r, google built te nsorflow to he lp spe e d up machine le arning and re se arch into de e p ne ural ne tworks.

By de sign, te nsorflow is me ant to work se amle ssly with diffe re nt gpus and cpus, and should e asily work on mobile platforms too. some of the wrappe rs that make te nsorflow one of the be st de e p le arning frame works include c++, java, and python.

Brief history of tensorflow

Some years back, discoveries from research in deep learning frameworks surpassed similar progress in other machine learning projects, especially in instances where massive data streams were involved. with this in mind, google saw it fit to advance deep neural networks as a means of improving their value proposition to users worldwide, while at the same time implementing other machine learning methods where necessary.

Three service offers that benefited from deep neural network integration were the search engine, email platform, and photo services. to achieve their goals, google built the tensorflow framework to enable their dev teams and researchers to work together on different artificial intelligence models that were under experimentation. the concept was to ensure that upon completion and deployment, more people would realize the value in the systems and be open to using them.

Tensorflow has been around since 2015. however, the first stable version was released two years later as an open-source project under the apache open source license. what this means for budding developers and researchers is that you can redistribute, modify and use the tensorflow framework as you see fit.

The tensorflow platform

The tensorflow structure is built into three distinct elements:

- data preprocessing
- model building
- model training and estimation

This deep learning framework is named tensorflow because it uses multi-dimensional arrays referred to as tensors. when using tensors, you can

build operational flowcharts, natively referred to as graphs. the operational flowchart is no more than a map of operations that you plan to conduct on the specified input data.

At one end, you receive the input, then pass it through different operations before it is delivered as output. this explains why the framework is referred to as a tensorflow because once input is fed into the system, it flows through a number of operations before churning out as output.

Tensorflow environments

There are different environments wherein you can train the tensorflow model. during the development phase of your model, you can perform all the tasks you need on your laptop or personal computer. as long as the hardware and software requirements are met, you should not have any challenges with that.

From the development phase, you will pass the project through the inference or run phase. in this stage, you test the model against different devices to see whether it can perform optimally. as a developer, it would be redundant to assume that once the project renders properly on your device, it will do the same in all the other devices. users have different platforms, and this is why it is always advisable to run tests on as many platforms as possible. test the model on linux systems, macos, and windows systems for desktop computing. give it a try on mobile platforms and cloud platforms in case you plan to have it running as a web-based service.

Once you have trained the model on different machines and ascertained its capabilities, you can then run it on your preferred machine. tensorflow models can be trained and used on different cpus and gpus. initially, gpus were specifically built for gaming. researchers discovered that gpus were capable of handling and solving a number of algebraic and matrix

computations, hence making them a good fit for performing deep learning calculations.

The concept of deep learning is heavily built on matrix computations. tensorflow is one of the fastest deep learning frameworks that can perform matrix computations. this comes from the fact that it is written in c++. however, this should not confuse you. while tensorflow is written in c++, you can control and use it with other programming languages, and in our case, it can run as a python library. therefore, prior knowledge of python and c++ will make your work much easier.

One of the important features of tensorflow is the tensorboard. this feature allows you to see how the tensorflow is going on in a graphical illustration.

Tensorflow components

Tensors are the basic building blocks for the tensorflow framework. every computation in tensorflow must use a tensor. a tensor is generally a matrix of n-dimensions that identifies any kind of data that passes through it. in simple terms, a tensor is a vector. in any tensor, all the values held within must share a basic data type. the data shape may or may be partially known, and is usually the dimensional feature of the array or the matrix represented by the data.

It is possible to compute some input data to create a tensor. all operations that take place in tensorflow are enclosed within a graph. graphs, therefore, represent a group of computations that must occur in succession. in line with this succession, every operation that takes place inside the graph is referred to as an *op node*, and all *op nodes* are connected to one another in the graph.

While the graph will outline the connection between nodes and the ops, it will never show us the values. the tensor, therefore, becomes the

e dge of the node s. this me ans that if you want to pe rform an ope ration with the data, you must use te nsors.

The graph frame work is use d in te nsorflow to colle ct and ide ntify all the computations that take place whe n training the de e p le arning mode l. the re are se ve ral be ne fits of using the graph for this, including the following:

● graphs we re spe cifically built to ope rate on diffe re nt gpus, cpus, and mobile platforms.

● graphs are portable , which e nable s de ve lope rs to maintain the inhe re nt computations for use e ithe r imme diate ly or at a late r date . to use the graph in the future , you can simply save it and load it whe n the time come s.

● all te nsorflow computations that take place within a graph are pe rforme d by conne cting te nsors. since e ach te nsor has an e dge and a node , node s fe rry the mathe matical computations and de rive outputs at the e ndpoints. e dge s, on the othe r hand, de fine the re lationship be twe e n conne cting node s.

Te nsorflow is one of the most popular de e p le arning frame works for a good re ason. it was built a scalable frame work to e nsure that e ve ryone inte re ste d in de e p le arning could be ne fit from it. the te nsorflow library was built to support diffe re nt apis, in the proce ss he lping use rs scale diffe re nt machine le arning and de e p le arning frame works like rnn.

Give n that the te nsorflow frame work thrive s unde r graphical computation, de ve lope rs can e asily visualize what the y are doing through the te nsorboard. this is a he lpful approach that allows de ve lope rs and re se arche rs an e asy time de bugging the ir programs. you don’t have to wait for the e ntire mode l to be comple te , the n scan all your code to find a proble m. through the te nsorboard, it is e asy to ide ntify proble ms in the code and fix the m.

All the se are re asons that e xplain the fact that te nsorflow is curre ntly the most wide ly use d de e p le arning frame work online . de v communitie s are always fille d with re se arche rs and othe r e xpe rts who are working on some de e p le arning proje ct with te nsorflow.

Algorithm support

Machine le arning is possible through the use of algorithms and data. the following are some of the common algorithms you will come across in machine le arning that are supporte d by the te nsorflow api:

- booste d tre e classification algorithms
- booste d tre e re gre ssion algorithms
- de e p le arning wide and de e p algorithms
- classification algorithms
- de e p le arning classification algorithms
- line ar re gre ssion algorithms

In the e xample be low, we will look at a simple te nsorflow e xample to e xplain diffe re nt line s of code use d in the proce ss.

```
Import numpy as np

Import te nsorflow as tf
```

The first two line s of code import the te nsorflow as *tf*. in python, librarie s are usually infe rre d with the short name . this is to re duce the challe nge of typing the long name of the library e ach time you write a function to call that library. the re fore , e ach time you ne e d to call a te nsorflow function in python, use *tf*.

We will look at an e xample that multiplie s two numbe rs. say we want to multiply m_1 and m_2, te nsorflow will build a mode l that conne cts this

ope ration. the mode l in our case is a multiply mode l. once we de te rmine our graph, our computational e ngine s will simply multiply m_1 and m_2.

In the te nsorflow se ssion, the computational graph will compute the value s of m_1 and m_2 the n print the re sult at output. in our e xample , m_1 and m_2 are the input node s. e ach time we use a node in te nsorflow, we must first spe cify the type of node we are cre ating. the m1 and m2 node s in our e xample are place holde r node s. the role of a place holde r node is to confe r a ne w value e ve ry time we pe rform a calculation. the re fore , we will have our node s as *tf.place holde r* node s as shown in the e xample be low:

Ste p 1: we must de fine the variable s

M_1 = tf.place holde r(tf.float32, name = "m_1")

M_2 = tf.place holde r(tf.float32, name = "m_2")

By cre ating place holde r node s, we must assign a name to the node that will be visible in the te nsorboard. we will name our node m_1, by passing it through parame te rs with the m_1 value , and do the same for m_2.

Ste p 2: we must de fine the computation

Multiply = tf.multiply(m_1, m_2, name = "multiply")

What we have done in this stage is to ide ntify a node that pe rforms a multiplication ope ration on our input data. in te nsorflow, this is done through a *tf.multiply* node as shown above .

Whe n we run this code , the te nsorflow frame work is instructe d to link the m_1 and m_2 node s within the computational graph and carry out the ne ce ssary computations. the re fore , it will pull the re spe ctive value s and conduct the ope ration.

Ste p 3: e xe cuting the ope ration

Be fore you e xe cute any ope rations within the grapy, you must first cre ate the se ssion. this is possible in te nsorflow using the function *tf.se ssion()*. once the se ssion is cre ate d, you can the n proce e d to pe rform ope rations within the computational graph by simply calling that spe cific se ssion.

Be fore loading any data into that will be use d to train the machine le arning algorithm, you must load the data into the me mory. this is a simple proce dure that use s one array. you might ne e d to write python code for this. re me mbe r that the code you write be ars no re lationship with te nsorflow.

The othe r way to do this is to use a data pipe line in te nsorflow. by de sign, te nsorflow has a unique api to e nable you to load data, conduct ope rations and use the machine le arning algorithms without any challe nge s. this is a good me thod e spe cially whe n you are working with a ve ry large datase t.

Give n the two options above , your conside ration will de pe nd on the type of data that you are working with. assuming that your data is le ss than 10 gb, you can load data into the me mory. if it fits pe rfe ctly, you can also use pandas to import data into the mode l using csv file s.

In case you are working with a ve ry large datase t, assuming that your file size is around 50 gb, your syste m will crash, e spe cially if the me mory is insufficie nt. you ne e d a syste m whose me mory is large r than the datase t. a 16 gb me mory machine , for e xample , cannot handle such a datase t.

The be st solution, in this case , is to cre ate a te nsorflow pipe line . the pipe line will the n load data into the mode l e ithe r in small bits or in batche s. as e ach batch of data is loade d into the pipe line , it can be traine d right away. the be st thing about using pipe line s is that it use s paralle l computing, so you can use te nsorflow to train your de e p le arning mode l in diffe re nt cpus without a hitch.

The re fore , if you are working with a small datase t, the pandas library is a suitable approach to loading data into the me mory. in case you ne e d to use

more than one cpu or if your dataset is too big, the best option is to use a tensorflow pipeline.

Creating tensorflow pipelines

In the example above we manually added values for m_1 and m_2 into the tensorflow framework. in this example, we will show you how to load data into the model using tensorflow pipeline.

Step 1: creating the data

Using the numpy library, we will create two random values as follows:

```
Import numpy as np

M_input = np.random.sample ((1,2))

Print(m_input)
```

Output

```
[[0.8835 0.2376]]
```

Step 2: building the placeholder

Just as we did in the earlier example, we will build a placeholder named m. we must also identify the shape that this tensor will bear. therefore, if we load an array that only has two values, the shape can be written as *shape = [1,2]*

```
# using the placeholder created

M = tf.placeholder(tf.float32, shape =[1,2], name = 'm')
```

Step 3: we will define the dataset method

We must identify the method used to populate placeholder values for m.

```
Tf.data.dataset.from_tensor_slices
```

Datase t = tf.data.datase t.from_te nsor_slice s(m)

Ste p 4: we build the pipe line

In this stage , we will cre ate the pipe line whe re our data will flow. we must build an ite rator *make _initializable _ite rator* and name it as ite rator.

To introduce the ne xt se t of data into the mode l, we must call the ite rator using the function be low

Ge t_ne xt

Our function at this point should look like this:

Ite rator = datase t.make _initializable _ite rator()

Ge t_ne xt = ite raror.ge t_ne xt()

Ste p 5: e xe cuting the computation

We will cre ate a se ssion the n run the ope ration ite rator. we will introduce data value s that we re ge ne rate d through numpy, which will the n give us value s for our place holde r, m.

To ge t the output, we will run *ge t_ne xt* as shown be low:

Tf.se ssion()

introduce data into the place holde r

Se ss.run(ite rator.initialize r, fe e d_dict={ m: m_input })

Print(se ss.run(ge t_ne xt)) # output [0.5237 0.7196]

Output

[0.8835 0.2376]

CHAPTE R 7: DE E P LE ARNING WITH PYTORCH AND KE RAS

E ve ry we e k, the dime nsions of what we can do with compute rs ke e ps changing. some tasks that de mande d high cognition le ve ls in the past can now be solve d faste r at supe r-sonic pe rformance le ve ls. e arlie r on whe n having your x-ray at the hospital, some one had to re ad it and try to e xplain it to you. today all this can be done with a compute r.

It is inge nuine to assume that machine s are starting to think like we do. e ve n with all the advance me nts in te chnology, the re are aspe cts of humanity that machine s might ne ve r le arn. what we have done ove r the ye ars is to build algorithms that can pe rform comple x computations and give us a de sirable output. this is particularly impre ssive for non-line ar proce sse s whe re machine s must le arn through inte llige nt me ans.

A lot has be e n me ntione d about algorithms ove r the course of your e xpe rie nce with machine le arning. algorithms play an important role in de e p le arning. de e p le arning is a discipline that focuse s on training de e p ne ural ne tworks. the se are simply mathe matical e ntrie s that le arn from e xample s the y are e xpose d to.

In de e p le arning, huge chunks of data are use d to solve sophisticate d functions. most functions solve d in this manne r do not share any corre lation be twe e n the inputs and outputs. the re fore , using mode ls that re ly on line ar re pre se ntations would be futile in atte mpting to solve such proble ms.

Pytorch is a python library that de ve lope rs use to build innovative proje cts in de e p le arning. the functionality be hind pytorch is such that the de e p le arning mode ls can be code d in python programming language . python has always be e n conside re d one of the e asie st language s for

programme rs and de ve lope rs alike be cause of its high-le ve l syntax which is close to normal spoke n language s.

Ke ras is one of the high-le ve l ne ural ne twork apis that are writte n in python. by de sign, ke ras is built to work with many othe r python librarie s like the ano and te nsorflow. the ultimate goal be hind the de ve lopme nt of ke ras was to spe e d up the e xpe rime ntation proce sse s in data re se arch. in so doing, the ke ras e nable s data analysts and de ve lope rs to spe e d up the e volution proce ss of the ir proje cts from conce ptualization to de ployme nt.

Ke ras is a python library that offe rs the following be ne fits for de e p le arning proje cts:

● you can run it se amle ssly on gpu and cpu without pe rformance hitche s.

● ke ras is spe cifically built to work with re curre nt and convolutional ne tworks. in case s whe re your proje ct ne e ds a combination of both ne tworks, you can still use ke ras for e xce ptional re sults.

● it is an e xte nsible , use r-frie ndly and modular library, as a re sult of which you can build fast prototype s in a short time .

Pytorch use s multi-dime nsional arrays known as te nsors. in the fundame ntal structure , te nsors are almost similar to numpy arrays. with this in mind, the re fore , it is e asy for de ve lope rs to build proje cts using pytorch. assuming the de ve lope r has all the ne ce ssary re source re quire me nts for the ir proje ct, te nsors he lp to spe e d up the ope rations. on a wide r scale , pytorch include s package s that he lp in data loading for worke r proce sse s, distribute d training, and a substantial library with all the important de e p le arning functions that you will use in your proje cts from time to time .

The re are lots of de e p le arning frame works curre ntly available in the world of programming. in the se ction be low we will introduce pytorch in the simple st way possible so you can have an e asie r e xpe rie nce .

The re are two broad classifications of pytorch mode ls within which we have 5 compone nts as shown be low:

Storage classification

● te nsor

● variable

● parame te r

Transformation classification

● functions

● module s

As we had me ntione d e arlie r, te nsors are simple arrays like you might have le arne d in python numpy arrays, spe cifically built for use in pytorch. in de e p le arning some of the te nsors you will come across include *torch.double te nsor,* and *torch.floatte nsor*.

Variable s te ll us more about te nsors. some of the information we ge t from variable s include whe n and how the te nsor was built, or how to store the te nsor gradie nts. e ach variable has unique prope rtie s, for e xample :

.data - re pre se nts a te nsor de scribe d by the variable .

.grad - re pre se nts the gradie nt associate d with the variable in que stion.

.re quire s_grad - a boole an re pre se ntation that de te rmine s whe the r the gradie nt can be e stablishe d afte r backpropagation.

Parame te rs are the mode l laye rs that he lp to conve rt the input into the de sire d output. parame te rs in pytorch are re pre se nte d as *torch.nn.parame te r*.

Functions in pytorch are re sponsible for conve rting the input data into a cognizable ope ration. functions are ne ithe r buffe re d nor do the y hold any

state . for this re ason, the y are e asy to pre dict be cause the y do not have any me mory.

Module s are some of the most important compone nts in pytorch. within a module , we can have functions, laye rs, parame te rs and in some case s, we can also ne st othe r module s.

If you are working with a backproping mode l whe re you have module s ne ste d within a module , the proce dure is to de te rmine the parame te r gradie nts for all main module s and the child module s ne ste d within. the re fore , whe n you are done building your mode l, invoking the backproping instruction should re turn the gradie nts for all the ne ste d child module s.

Pytorch mode l structure s

To de fine a mode l in pytorch, you will use one of the following functions:

- forward
- _init_

A forward function conside rs all the possible laye rs and functions associate d with the input data while the _init_ function calls all the parame tric and non-parame tric laye rs instantly. in so doing, you will have succe ssfully built a pytorch mode l.

From the re , the ne xt ste p is to figure out the loss function in your mode l. in the case of a parame tric function, we can use the loss function *nn.functional.mse _loss* functional. if using a non-parame tric function, the *nn.mse loss()* function would be use d. the diffe re nce be twe e n calling a parame tric ove r a non-parame tric function is that for the forme r, you would have to be spe cific in your re fe re nce , for e xample , *nn.functional.mse _loss(out, n).*

The next procedure is to determine the pytorch optimizer. you might not use optimizers especially if you still have gradient data in the *.grad* attribute for your variables. with the optimizer, you can easily update parameters without invoking other functions.

Modules with nested child modules can access the children through different methods, including the following:

- model.children()
- model.named_parameters()
- model.modules()
- model.named_children()
- model.named_modules()
- model.parameters()

While all these are appropriate for calling model children, you will come across *model.parameters()* used frequently in pytorch computations. *model.parameters()* calls all the parameters in the function, hence this is also a good way to use optimizers.

At this juncture, you can start training your pytorch model to interact with different data.

Your model should start with the following line:

Optim.zero_grad()

This will outline the gradient buffers *.grad* such that any parameter used is automatically set to zero. once you update the weights, you can then update the former gradients before you generate the new ones for any additional data. for this case, you can also use the *model.zero_grad()* function.

Why do we go the extra mile to set gradient buffers to zero instead of using *loss.backward()*? this process helps us backprop between multiple losses and children, therefore we don't have to spend a lot of time determining the gradient of the input data. the other reason for this procedure is to mitigate the possibility of updating the pytorch model later on, especially when you come across data that is too large for your present resource allocation.

Once you are through with this and your model is ready, you can call the model. let's assume you are working on a model m. you will call it as follows:

Model.forward (m)

Any loss in this model will be determined by the loss function prescribed at in your model. from there you can proceed and calculate gradient parameters responsible for the loss as follows:

Loss.backward()

Once you are through with the model and need to update the input data you introduced into the system, you must update the parameters introduced to the optimizer at the initialization phase this is done using the following function:

Optim.step()

Initializing pytorch model parameters

In pytorch, you can initialize model parameters in one of many ways. while working on pytorch models, you should not forget that parameters are variables. the following are some methods you can choose to initialize model parameters:

- using *model.parameters()* you can loop over the module parameters. once you do this, you can then initialize every parameter according to the

te nsor functions re lative to it, including fill, uniform or e xpone ntial. the re are lots of othe r te nsor functions that you can use de pe nding on the mode l you are building and its purpose .

● anothe r me thod of initializing mode l parame te rs is to use the *.apply* attribute de finition. *.apply* can be use d as a function to he lp sort out parame te r initializations. e ach time you use the *.apply* attribute in a pytorch module , it re ve rbe rate s on e ve ry child module ne ste d within the main module .

● a practical me thod of initializing parame te rs is to use the *torch.nn.init* module . le t's assume we introduce a ne w parame te r (x) into the syste m. to initialize parame te r x using the glorot initialization protocol, we should have the following: *torch.nn.init.xavie r_uniform(x)*.

All the options me ntione d above will he lp you initialize parame te rs. howe ve r, inste ad of doing all the hard work, you can le t pytorch handle the initialization for you. e ach time you build a mode l, pytorch use s the most appropriate initialization me thod for any parame te rs use d.

Principle s supporting ke ras

The succe ss of ke ras as a python library for de e p le arning can be attribute d to four important factors. first, it has a use r-frie ndly inte rface . use r-frie ndline ss is a challe nge that many de ve lope rs struggle with today. use rs' ne e ds ke e p changing and it is not e asy for de ve lope rs to ke e p up with the pace of such dynamic change s, e spe cially whe n you cannot quantify the m.

By de sign, the ke ras api was built with a vie w of e fficie nt inte raction with human inte rface s, not machine le arning. the re fore , e ve n as you work on your proje cts, use r e xpe rie nce is always a priority.

Anothe r challe nge that ke ras addre sse s is the ne e d to re duce and possibly e liminate cognitive load. give n the simplicity be hind the apis,

e ach time you use ke ras you notice the consiste ncy in the flow of data and all othe r functions you ne e d. this approach was aime d at e nsuring that you can pe rform the tasks and assignme nts ne ce ssary in ke ras by using the fe we st possible actions. should you e ncounte r an e rror in any proce ss, ke ras outline s the proble m cle arly, with actionable fe e dback.

Se condly, ke ras is a scalable library. this is a fe ature share d by most python librarie s. be cause of this e xte nsibility, you can add ne w module s to ke ras e ithe r as functions or classe s, making work e asie r for you. furthe r to that, pre -e xisting module s act as good e xample s of what you ne e d to do. the re fore , if you struggle with any module , you can always le arn from what you have alre ady done .

Scalability in ke ras is not just about adding more module s to the platform. it is also about ope ning up the library for future re se arch. e xte nsible librarie s cre ate room for future de ve lopme nt be cause de ve lope rs are fre e to e xpre ss the mse lve s by advancing the ir skills.

Third, ke ras is a modular library. the modular library is one whe re mode ls are re cognize d as standalone graphs or part of a se que nce . the module s in ke ras are ve rsatile and you can re configure the m without a hitch be cause the re are ve ry fe w limitations to this.

Some of the common module s that you can configure in ke ras include cost functions, ne ural laye rs, initialization sche me s, re gularization sche me s, activation functions, and optimize rs.

Finally, be ing a python library, you do not ne e d to cre ate a diffe re nt configuration mode l whe n using ke ras. all the mode ls use d are compatible with python, and the ir unique de scription is also writte n in python. the be st thing about compatibility with python is that you can e asily de bug e rrors whe ne ve r the y appe ar.

Ge tting starte d

The foundation of keras data structures is a model. it is through models that we can prepare and arrange layers to suit our data needs. the simplest keras model is the *sequential* model. this is basically a linear arrangement of layers. below is an example of what a *sequential* model looks like :

```
From keras.models import sequential

Model = sequential()
```

In case you are working with complex structures, you must use a keras api to create arbitrarily layered graphs. to do this, you will have the following code :

```
From keras.layers import dense

Model.add(dense (units=54, activation='relu', input_dim=100))

Model.add(dense (units=50, activation='softmax'))
```

If you are comfortable with the model as it is, the next step is to reconfigure its learning process with the *.compile ()* module as follows:

```
Model.compile (loss='categorical_crossentropy',

Optimizer='sgd',

Metrics=['accuracy'])
```

Beyond this, you can take things a step further by configuring the optimizer too. remember that when using keras, it is always advisable to keep your code as simple as possible. this way, you can work on the source code without any challenges, and anyone else who has to work using your code will also have an easy time reading and understanding it.

Keras preferences

Today developers have access to lots of deep learning frameworks. the decision to use keras over all the others is usually a personal choice,

de pe nding on what the individual de ve lope r is comfortable working with. that be ing said, howe ve r, the following are some of the re asons why using ke ras would be a good option for you:

- de ve lope r e xpe rie nce

Ke ras is one of the simple st de e p le arning frame works to use . by de sign, the priority be hind its structure is to make de ve lope rs' e xpe rie nce s as smooth as possible . how can this be done ?

First, the ke ras api is spe cifically built for use by humans. this is diffe re nt from many othe r de e p le arning api frame works whose focus is on machine le arning. the platform is built to make it as e ngaging as possible for de ve lope rs, and in the proce ss he lp the m build amazing machine le arning mode ls. as we me ntione d e arlie r, the ke ras api he lps to re duce the cognitive load that de ve lope rs have to de al with from time to time . the consiste ncy and simplicity allow de vs to build proje cts without de manding a lot of actions from the m.

The ne xt ste p in e nhancing de ve lope r e xpe rie nce in ke ras is simplicity. this is one of the e asie st platforms you will le arn as a de ve lope r. if you have background knowle dge in python or othe r programming language s, you shouldn't have a difficult time maste ring ke ras. simplicity in the de sign is aime d at incre asing productivity. as you work on your proje cts, you should be able to de live r the m faste r than most of your compe titors who work on comple x de e p le arning frame works.

Finally, one of the pitfalls of simplicity in most de e p le arning frame works is usually to sacrifice fle xibility among othe r fe ature s. in ke ras, this is not the case . in fact, ke ras is built to inte grate e asily with low-le ve l de e p le arning frame works and language s which doe s not just make it more fle xible , but also make s it ve rsatile . you can work on a wide range of proje cts in ke ras without any challe nge s. one of the common frame works that you can inte grate with ke ras is te nsorflow. what this me ans is that any proje ct that you might have built in one of the supporte d base language s

can e asily be imple me nte d in ke ras too. you will e xpe rie nce this particularly with workflows in te nsorflow.

● supporting community

The ke ras use r community is so dive rse with hundre ds of thousands of use rs. this has made it e asie r to adopt and spre ad the use worldwide . with such a dive rse community of de ve lope rs, you will not have a difficult time working on your proje cts. if you are e ve r struggling with some thing, you have a lot of pe ople you can re ach out to who will assist.

In te rms of the community of use rs, ke ras only come s se cond to te nsorflow in the numbe r of use rs. since the se two frame works can be use d toge the r, you have a combine d support community that surpasse s your imagination.

The ke ras frame work has be e n adopte d in the re se arch community be tte r than any othe r frame works othe r than te nsorflow. in fact, the ke ras api is inte grate d into the te nsorflow frame work unde r the module *tf.ke ras*.

To show you just how wide spre ad ke ras is, le t's conside r some of the applications and se rvice s you use on a daily basis that have some e le me nt of ke ras running the ir e ngine . the se include ube r, ne tflix, and ye lp. most startup proje cts that are built around de e p le arning are using ke ras, so we can e xpe ct to inte ract with more ke ras proje cts in the future .

Nasa and ce rn have be e n at the fore front of de e p le arning re se arch ove r the ye ars. by adopting ke ras in the ir proje cts, this is a sure confide nce boost that shows you just how much you can do with this frame work.

● mode l conve rsion

The simplicity of using ke ras can furthe r be e xplaine d in how e asy it is for de ve lope rs to conve rt the ir mode ls into e nd products. ke ras make s it e asy for you to move from e xpe rime ntation to de ployme nt. this is be cause compare d to othe r de e p le arning frame works, you can e asily de ploy ke ras mode ls across as many popular platforms as possible . ke ras mode ls can be

de ploye d in ios, android, native browse rs, raspbe rry pi, jvm, google cloud, and the python backe nd we b apps.

● ve rsatile e cosyste m

You can use any numbe r of de e p le arning backe nd platforms to de ve lop ke ras mode ls. this is possible be cause ke ras supports most of the backe nd e ngine s in use in the world today. with this in mind, you are not re stricte d to a single e cosyste m as is the case with othe r de e p le arning frame works.

If you are building a ke ras mode l that can only be le ve rage d through laye rs built into the mode l, you can port the mode l on any backe nd e ngine that supports ke ras. what this me ans is that you can work on the same mode l across diffe re nt backe nd e ngine s. for e xample , you can build and train the mode l on one backe nd e ngine and load it on a diffe re nt backe nd e ngine e ithe r for e xpe rime ntation or final de ployme nt. this fe ature allows you to te st how your mode l will pe rform in diffe re nt e nvironme nts, which is a good thing be cause once de ploye d, you want the mode l to run smoothly without e ngine hindrance s.

Some backe nds that you can use whe n building ke ras mode ls include the ano, microsoft's cntk backe nd, and google 's te nsorflow. amazon also joine d this list by re le asing a ke ras fork whose backe nd runs on mxne t.

What doe s this me an for you and the e nd-use r? for the de ve lope r, you can de ploy your de e p le arning mode ls in diffe re nt e nvironme nts. this is a good thing be cause you will have acce ss to a wide marke t. for e nd-use rs, this is also a good ide a be cause use rs have unique pre fe re nce s in hardware se le ction. anything that goe s be yond the conve ntional cpu platforms re quire s use rs who have a good unde rstanding of computing hardware . ke ras de e p le arning mode ls, the re fore , can be imple me nte d on nvidia gpus, gpus that support ope ncl, amd gpus and google tpus.

● support and training

Whe n using ke ras to build your proje cts, you stand to be ne fit from the inbuilt support for multiple data paralle lism in gpus. a good e xample is

horovod used in uber, which has unlimited support when using keras models.

Another benefit of this support is that you can easily convert keras models into tensorflow estimators, thereby training the models on different data clusters within the google cloud ecosystem.

- support from tech giants

For developers, there is nothing better than using a deep learning framework that enjoys unwavering support from some of the top companies in the industry. this is usually a good sign for the future, that there will be more developments in the field, and you can keep advancing your knowledge and skills.

Google is one of the biggest proponents of keras, given that it can integrate smoothly with another google product, tensorflow. together, these two frameworks are at the center of most deep learning projects in different stages of development at google and other tech companies. microsoft and amazon have also thrown their weight behind keras. some other giants in the tech industry that support keras include apple through coreml, uber, and nvidia.

Keras functional api

When defining complex models, you will need to use the keras functional api. some examples of complex models where this comes in handy include shared layer models, directed acyclic graphs, and models with multiple outputs. before we proceed with the keras functional api, it is imperative that you know how to use the *sequential* model.

The *sequential* model is basically a stack of linear layers. to create this model, the constructor must receive a list of layer instances as shown below:

From keras.models import sequential

```
From ke ras.laye rs import de nse , activation

Mode l = se que ntial([

De nse (32, input_shape =(784,)),

Activation('re lu'),

De nse (10),

Activation('softmax'),

])
```

Alte rnative ly, you can also use the *.add()* me thod to introduce more laye rs to the constructor as shown be low:

```
Mode l = se que ntial()

Mode l.add(de nse (32, input_dim=784))

Mode l.add(activation('re lu'))
```

You must spe cify the kind of input shape the mode l e xpe cts to work with. be aring this in mind, always make sure that the first laye r in your *se que ntial* mode l indicate s the shape of the input. this can be done in one of the following me thods:

First, you can use the first laye r to pass an *input_shape* argume nt. in most case s, you will introduce a shape tuple in this case . you can also use a *none* e ntry alongside the tuple shape which informs the laye r to e xpe ct any positive inte ge rs.

Anothe r way to do this would be by using 2d laye rs like *de nse* . in such a case , the input shape must be indicate d using the *input_dim* argume nt. if you are using 3d laye rs, you can use the *input_dim* or *input_le ngth* argume nts.

You cannot train a de e p le arning mode l in ke ras be fore configuring the le arning proce ss. this can be done through the *compile* me thod. in this

case , you can pass any of the following argume nts at compilation:

● optimize r

An optimize r can re pre se nt a string ide ntifie r of any instance within the *optimize r* class or any e xisting optimize r like *adagrad* or *rmsprop*.

● loss function

This is the obje ctive that your de e p le arning mode l is built to minimize . the loss function can be an obje ctive function or a string ide ntifie r of any pre -e xisting loss functions like *mse* or *cate gorical_crosse ntropy*.

● list of me trics

Me trics could re pre se nt custom me tric functions or string ide ntifie rs for e xisting me trics. e ach time you work on any classification proble m, it is impe rative that you se t the me trics list to *me trics=[‘accuracy’]*.

With this information, you can now de lve into ke ras functional apis. we will atte mpt to imple me nt a de nse ly conne cte d ne twork using the *se que ntial* mode l. in the e xample be low, take note that you can train it in the same way you would train ke ras *se que ntial* mode ls. othe r than that, this laye r instance is callable whe n using te nsors, and will re turn a te nsor. you can also use the input and output te nsors whe n de fining the mode l. le t’s look at the e xample be low to illustrate this:

```
From ke ras.laye rs import input, de nse

From ke ras.mode ls import mode l


# this will re turn a te nsor

Inputs = input(shape =(784,))


# the callable laye r instance on a te nsor will re turn a te nsor
```

```
Output_1 = de nse (64, activation='re lu')(inputs)

Output_2 = de nse (64, activation='re lu')(output_1)

Pre dictions = de nse (10, activation='softmax')(output_2)


# this will build a mode l

# the  mode l has thre e  de nse  laye rs and an input laye r

Mode l = mode l(inputs=inputs, outputs=pre dictions)

Mode l.compile (optimize r='rmsprop',

Loss='cate gorical_crosse ntropy',

Me trics=['accuracy'])

Mode l.fit(data, labe ls) # this will start training the  mode l
```

As you le arn how to work with ke ras, re me mbe r that it is compatible with all python ve rsions from python 2.7 to python 3.6. so, che ck to e nsure you have the right python variant installe d.

CHAPTE R 8: ROLE OF MACHINE LE ARNING IN THE INTE RNE T OF THINGS (IOT)

The inte rne t of things (iot) fe ve r has grippe d the world at the mome nt. the re are programs, de vice s and software popping up all ove r. this advance me nt is e xpe cte d to spurn a re volution in as far as te chnology is conce rne d. one of the highlights of iot is cloud computing. the te chnology be hind iot pre dominantly runs on the cloud, or has some part of it anne xe d to the cloud. much as the te chnology is advancing, cloud computing is also advancing in similar me asure . in the e arlie r days, the cloud was an ave nue for data colle ction and storage . today the cloud has e volve d to a place whe re is active ly unde rstood and inte rpre te d through machine le arning.

If we follow the history of computing, data proce ssing has always be e n one of the fundame ntal obje ctive s. data proce ssing include s colle ction, storage , curation and updating data. all this is curre ntly possible through machine s. data proce ssing has e volve d ove r the ye ars, from punche d cards through disk drive s and finally proce ssing on the cloud.

Throughout civilization, humans have always le arne d through failure and drawing infe re nce s from the le ssons the re in. the cave man, for e xample , ne ve r le arne d how to make fire from re ading a book or taking a course . the y le arne d through trial and e rror, and the y e ve ntually got it right. this has be e n the tre nd in le arning ove r the ye ars, and it can also e xplain why a lot of pe ople are on youtube looking for le ssons on how to do a lot of things, from basic to comple x stuff.

For the longe st time , compute rs and machine s we re use d to solve proble ms in a manne r that did not e spouse this. through machine le arning, we can now build syste ms that le arn through trial and e rror. the se syste ms make the ir own inte rpre tations of data and build mode ls to solve proble ms. while the syste ms are not always right at the first atte mpt, de ve lope rs work round the clock to he lp re train the mode l and provide be tte r data upon which the syste m will le arn going forward. this is how machine le arning mode ls are improving.

Machine le arning and artificial inte llige nce have had an e normous impact on te chnology. with e ach smart de vice that we use , the re is so much data on hand. be aring this in mind, it is important to conside r ways of handling such data without ne ce ssarily looking for additional storage . this is whe re machine le arning and iot find common ground.

Fusing machine le arning and iot

The re are so many are as in our live s whe re iot de vice s and syste ms are coming in handy. you might not have use d some of the se de vice s or syste ms, but you have come across the m alre ady. the y have made work and life e asie r for a lot of pe ople . the re are a lot of smart de vice s that are curre ntly in our home s and office s, and the y all se rve a unique purpose .

A fe w ye ars back if you had to trave l away for a fe w days or we e ks, you would worry about the se curity of your home . you wouldn’t want pe ople to know you we re away, he nce le aving your prope rty at risk of burglary. a solution would be to le ave the lights on, but the e ne rgy bills would be off the charts. thanks to iot, you can trave l without worrying about any of that. all you have to do is install smart lights in your home that can go on and off according to your pre de te rmine d sche dule .

Virtual assistants like cortana and siri have also he lpe d bridge the conne ction be twe e n human inte raction and iot through machine le arning.

with time , the se assistants le arn how to re spond be tte r to our cue s, making it e asie r to addre ss our ne e ds.

While the marve ls of iot are incre dible , we must admit that the se de vice s and programs do not offe r us 100% pe rfe ction. the inte llige nce and functionality built into the se de vice s must improve and de ve lop with diffe re nt e xpe rie nce s in such a way that the y addre ss pe rtine nt issue s and he lp to make life e asie r for use rs.

The re is a ple thora of de vice s that are curre ntly within the iot divide . all the se de vice s ne e d one thing to work pe rfe ctly, the ability to le arn from your taste s and pre fe re nce s. with this knowle dge , the de vice s can re spond be tte r to pe rsonal re que sts and give you the right re sponse s. augme ntation syste ms and de vice s he lp to ge t you an inte rconne cte d e xpe rie nce with all the de vice s you are conne cte d to.

Through the iot, we inte ract with robust syste ms that should not be susce ptible to human e rror. this is the primary distinction be twe e n re lying on the iot and pe rforming tasks on your own. if you have an inte rconne cte d syste m whe re se ve ral de vice s are part of your routine , the se de vice s will ke e p communicating with one anothe r and make inte llige nt de cisions base d on what your pre se nt activity status is.

Through machine le arning, de vice s in the iot re alm can pe rform spe cific tasks inte llige ntly. be yond that, the y go furthe r and optimize proce sse s that would have othe rwise be e n re dundant without the ir inte rve ntion.

Most of the time , we only look at the iot syste ms in te rms of what the y can do for you as an individual. howe ve r, it is also possible to have a range of e ffe cts for more than one pe rson. take the e xample of a re staurant owne r. once you know the de mographics of your custome rs the re is a lot that you can do to make the m comfortable and happy e ach time the y come to your re staurant. for starte rs, prope r iot de vice s can automate the lighting in the re staurant to match a give n ambiance . as e ve ning approache s, the lights will automatically dim to cre ate a calm and soothing atmosphe re . you can also have a syste m whe re the lights and your music se le ction are

inte rconne cte d. this way, e ach time the music change s, the light matche s the unde rlying mood in the music.

Such te chnology he lps us have an e asie r e xpe rie nce in diffe re nt sce narios. the y cre ate an e nvironme nt whe re you can automatically filte r noise from your imme diate surroundings without moving an inch. in the long run, the be ne fits are imme nse for your pe rsonal space and he alth. e xpe rts be lie ve that through machine le arning and the inte rne t of things, we can succe ssfully cre ate a safe , smart world. be tte r ye t, we can cre ate a world whe re syste ms inte ract and he lp us make de cisions without the risk of human e rror.

Machine le arning challe nge s in iot

The be ne fits of the inte rne t of things are incre dible . couple d with innovative machine le arning algorithms, the re is so much that we can look forward to. howe ve r, it would be foolhardy to assume that e ve rything would be a smooth-sailing prospe ct. inde e d, the re are inhe re nt risks in advancing the machine le arning conce pt in iot. many e xpe rts have pointe d out some of the loophole s, highlighting some of the risks involve d and why we must e xe rcise caution going forward.

Most of the risks involve d are about cybe rse curity. for syste ms that are he avily re liant on data, we must be care ful about the data we share and what e ntitie s do with it. data mining is at an all-time high at the mome nt, and it is only fair that you e xe rcise caution on who you give acce ss to.

Authoritie s in the fie ld have made stride s towards imple me nting strong data prote ction laws and re gulations, e spe cially the gdpr. howe ve r, since most use rs hardly know the ir rights, it is difficult for the m to te ll whe the r the ir rights are infringe d upon or not. for this re ason, we have a lot of data handle rs who mismanage use r data but ne ve r ge t punishe d for the ir actions. by ge tting away, most of the m only ge t bolde r in the proce ss, abusing the ir privile ge d acce ss to pe rsonal use r data.

● data privacy

The case for data privacy is one of the thornie st topics in the world at the mome nt. diffe re nt laws e xist in e ach jurisdiction to prote ct use rs. the ne e d for prote ction be came e vide nt in light of the wide spre ad use of social me dia. pe ople share lots of information online , oblivious to the challe nge s that the y might put the ir live s and the live s of those close to the m.

In as far as pe rsonal data is conce rne d, the inte rne t of things pre se nts a ne w challe nge that we must confront. many pe ople worry about the amount of data available on social me dia. howe ve r, the re is only so far that you can go with such data. in the case of iot de vice s, the se are the syste ms that are built to unde rstand you be tte r. the y don't just colle ct data about you, the y use that data to le arn as much as possible about you.

E xpe rts be lie ve that iot de vice s colle ct e nough data that the y probably know more about the use rs than the use rs know of the mse lve s. this le ve l of aware ne ss cre ate s a risk should such syste ms be bre ache d. with the kind of conne ctivity, we have in our house s, you can only imagine what would happe n if some one hacke d into one of the se de vice s. from malicious attacks and blackmail, fraud and ide ntity the ft, the data at the attacke r's be he st would ope n gate s into your life .

● attacks be yond syste ms

What's the worst that could happe n if your syste ms we re hacke d? if a hacke r bre ache d your iot syste m, the y have sufficie nt data to build se rious attacks. while we can look at this in te rms of your pe rsonal space , if we look at the bigge r picture things could e scalate so fast.

A lazy hacke r would simply pe ne trate your syste ms and pe rhaps ste al your information. a dange rous hacke r might take things to the ne xt le ve l, initiate physical harm. think about it for a mome nt, what happe ns if you are in your se lf-drive n car and some one hacks the console ? the y would control the car and ste e r it to whiche ve r de stination the y ple ase . the y

might also lock you inside the car such that you could not make a de spe rate atte mpt to e scape .

For high-profile pe ople like ce le britie s, politicians, and profe ssionals in diffe re nt industrie s, such attacks are a thre at that most pe ople would not wish to go through. we might have se e n this happe n in the movie s, but it is actually a possibility in the re al world give n how fast we are advancing te chnology.

As we advance te ch in machine le arning algorithms, it might be possible to have ke y syste ms like powe r plants and sate llite s on this grid too. if hacke rs gaine d acce ss to such syste ms, the wrath that the y could unle ash on the unsuspe cting population would be unimaginable .

● e conomic paralysis

Othe r than the cloud and acce ss to it, all the iot de vice s ne e d powe r. the same applie s to machine le arning. by conne cting to powe r source s, this me ans that the y share a conne ction to powe r grids. hacking the se de vice s cre ate s an opportunity for hacke rs to go afte r bigge r gains.

It is important to have se curity and safe ty protocols in place to e nsure that e ve n in the unlike ly e ve nt of a powe r failure , none of the se de vice s would be affe cte d. more importantly, we must e nsure that the se de vice s are not use d as a conduit for large -scale de struction.

For e xample , se lf-driving cars are built to ge t information from the traffic grid, which the y use to map the be st possible route . what happe ns whe n it is impossible to conne ct to the traffic grid? it is important to e nsure that e ve n in the abse nce of a conne ction to the grid, use rs can still use the ir de vice s without any conce rns. you should be able to switch from autonomous driving and drive yourse lf to your de stination if the se lf-driving syste ms fail.

A robust ne twork is one that can thrive in the abse nce of acce ss without collapsing or be coming re dundant. as we e mbrace the be st of machine

le arning and the inte rne t of things, we must ne ve r forge t that it is be tte r to pre pare for the worst than ge t caught unaware s.

While the challe nge s outline d he re in might paint a dark picture of the world we might live through in the coming ye ars, it is not all doom and gloom. proble ms always have solutions. machine le arning, to be pre cise , is built around finding fe asible solutions to proble ms. be side s, it will take a long time be fore we have a world that is comple te ly inte rconne cte d. that le ave s us with a lot of room for trial and e rror, e xpe rime ntation and a be tte r shot at ge tting things right.

CHAPTE R 9: LOOKING TO THE FUTURE WITH MACHINE LE ARNING

Machine le arning mode ls are curre ntly in use in so many industrie s. the prolife ration of this te chnology me ans that in the ne xt fe w ye ars, machine le arning and artificial inte llige nce will be among the top inve stme nt ve hicle s worldwide . while the re is a lot of promise for the future , one of the bigge st conce rns we have at the mome nt is that a lot of e ntitie s are not re ally re ady to e mbrace machine le arning as the y should.

One challe nge that hinde rs this de ve lopme nt is the lack of sufficie nt re source s. most machine le arning mode ls in use are re source -inte nsive , and for this re ason, we cannot e xpe ct individuals to e mbrace the m fully. companie s, on the othe r hand, have the e xpe rts, financial muscle and othe r re source conside rations that can make this possible . e ve n with that advantage , most companie s are unable to do that.

A common challe nge is usually in imple me ntation. imple me nting a machine le arning mode l into your busine ss me ans that you must have an e xpe rt who unde rstands the syste m and the re pe rcussions of poor imple me ntation. inte gration furthe r de mands that your te am is prope rly traine d to e nsure the y are re ady for what lie s ahe ad. without skille d staff, it is impossible for any organization to fully e xpe rie nce the be st that machine le arning has to offe r. in fact, in most case s, this would simply ope n up the organization to more complicate d proble ms.

Looking to the future of machine le arning and inte gration into socie ty, most of the time we focus on what we can do with machine le arning mode ls. this is ofte n a one -side d approach be cause we fail to re alize that the re is a lot more to imple me nting the se syste ms than just paying for

them. we must have the necessary architecture and structures in place, and more importantly the right personnel.

The speed at which machine learning models are advancing is incredible. with this in mind, we can be certain to experience major shifts in acceptance as these models are integrated into society. currently one of the top concerns that most people have is the risk of complete dependence on ai and other associated technologies. however, most of the concerns people harbor are borne out of lack of exposure to these systems and knowledge on what they can do.

As more information and pilot tests are made available, we can expect that people will become more accepting and realize the benefits that these systems have to offer. once the mistrust goes away, we can then enjoy the techy solutions that machine learning offers to our normal problems.

The business angle

Businesses stand to gain a lot in the proliferation of machine learning models over the coming years. of particular emphasis is in prediction and the use of such models in decision making. if the current business market dynamic is anything to go by, we have a lot of businesses that are struggling to gain a competitive edge in their respective industries.

Through machine learning, we can expect a lot of businesses to refine their strategies and business processes, in the process managing to run agile and astute businesses. since outcomes can be influenced by machine learning models, we will see more businesses adopting such systems in the future. the point at which machine learning and artificial intelligence are in our lives is a critical one, and we will see more augmented performances in business auspices.

Ai in the future

After the industrialization era, there was increased investment in building systems that could operate and behave like people. this is where ai has been a blessing to mankind. experts and researchers have built lots of models in the past, tested and implemented quite a number. that is not the end, however. there is excitement about building machines that will not just think like humans, but do better.

The future demands intelligent products that can think and operate without the risk of human error. human error has often led to catastrophic events, so if this can be avoided through innovative ai, the businesses and individuals alike will have a better experience of future technology.

At the moment a lot of industries are either experimenting or implementing digital assistants, self-managing devices and applications. we have also seen considerable interest in building smart cities. this is proof that building and working with intelligent machines is not just a dream, it is a reality.

The transformative effect of ai and machine learning in the future will transcend most of the industries where machine learning is currently deeply embedded like finance, manufacturing, and the retail sector and the healthcare industry. we will see more machine learning models implemented in media, and other categories in the service industry. we will also experience the introduction of machine learning in discovering new industries and niche markets that cater to individuals with uniquely specific needs.

The machine learning industry is fast spiraling towards maturity. the increased adoption in many businesses is proof of this. whether in small capacities or in large-scale enrolment, we will encounter more machine learning usage in the following sectors:

- cloud computing

Business applications are at the forefront in the adoption of machine learning models. one area where this will increase is in running cloud-

base d busine sse s. ove r the ye ars busine sse s have adopte d many products-as-a-se rvice mode ls for diffe re nt re asons. the fle xibility that come s with using such busine ss mode ls has he lpe d many busine sse s scale up the ir ope rations, doing away with unne ce ssary de partme nts that e ithe r slowe d down the ir de live rable s, or cre ate d unne ce ssary wage burde ns. following suit, we can e xpe ct the adoption of machine le arning-as-a-se rvice (mlaas) platforms. this will se e more automation in busine ss proce sse s.

● data inte gration

Machine le arning syste ms that are alre ady conne cte d to busine ss mode ls will ke e p le arning, re fining the ir algorithms and ope ration manuals in the proce ss. since e ve ryone is ge tting online at some point, we can e xpe ct that the re will be more actionable data available for the se machine s to le arn from. base d on this data acce ss, it will be e asie r for machine le arning mode ls to train against ne w data.

With e me rging tre nds cropping up all the time , the machine le arning algorithms will also have to adapt to incorporate the se change s. the inte rne t age make s it e asie r to popularize ne w tre nds. a ge ne ration that ge ts e xcite d by things, ide as and conce pts going viral will have a huge impact in training machine le arning mode ls to adapt at the same spe e d.

● hardware conside rations

The onus is upon hardware manufacture rs to up the ir game or ge t le ft be hind. the wave of machine le arning adoption swe e ps so fast and as e xpe cte d, most of the hardware in use at the mome nt will hardly be able to handle the ne ce ssary tasks. hardware ve ndors must, the re fore , look at ways of building be tte r hardware with sufficie nt computing powe r to handle the data proce ssing de mands of machine le arning mode ls.

It is not just a que stion of building powe rful machine s. to re main profitable , manufacture rs must also conside r the cost impact of the ir production and usability. ux make s a big diffe re nce on whe the r products

can be adopte d fast e nough or not. without prope r ux de signs, e ve n the be st products in the marke t will fail to attract custome rs.

Case in point, apple re ce ntly launche d the iphone 11. compare d with the re ce nt flagship re le ase s, iphone 11 was re le ase d at a re lative ly lowe r price . most iphone s usually launch around the $1,000 rate . apple has inste ad re le ase d an improve d, be tte r de vice but at a conside rably lowe r price . in light of this, it is e xpe cte d that othe r playe rs in the marke t, e spe cially samsung will have to find a way to produce the ir ne xt compe ting flagship mode l at a re lative ly che ape r rate while still packing amazing hardware to compe te against the ir rivals.

Away from smartphone s, manufacture rs e spe cially those who produce proce ssing chips will have to find ways of me e ting custome rs' machine le arning de mands while at the same time se tting appropriate price s for the ir products. the de mands of machine le arning mode ls' proce ssing capacitie s will ke e p incre asing ove r time , so it is only fair that the hardware spe cifications will match the de mand.

● unde rstanding data

How we ll do we unde rstand the data we have at our disposal? many e ntitie s hold massive amounts of data but have ne ve r be e n able to make se nse of it. those who do are still unable to utilize the full e xte nt of the data. we have many busine sse s that are curre ntly working hard to migrate the ir busine ss mode ls from the old manual filing syste ms to mode rn digital syste ms. howe ve r, the proble m with this migration is that at be st, many such busine sse s are ge ne rally moving data into microsoft e xce l.

Machine le arning involve s consuming data and making se nse out of it. as a data analyst, you can look forward to gaining de e pe r insight into busine ss proce sse s from the machine unde rstanding of data. the mode ls can proce ss data faste r and more pre cise ly than any human can. whe n all the ne ce ssary data is capture d in the right format and imple me nte d corre ctly, the re is so much that you can do with a syste m that is re ce ptive do that

data. inte re sting time s lie ahe ad for data analysts and anyone involve d in active de cision making in running busine sse s on a day-to-day basis.

● te chnological multiplicity

For a computing mode l that thrive s on the pre mise of le arning from ne w inte ractions, we can e xpe ct machine le arning mode ls to inte grate with othe r te chnologie s in the ne ar future . one of the first are as whe re this will happe n is in the inte rne t of things. in machine le arning, it is possible to use more than one algorithm to e ffe ctive ly pe rform a task. this proce ss is known as e nse mbling. e nse mbling algorithms basically involve using more than one algorithm that inde pe nde ntly cannot solve a proble m e ffe ctive ly, but toge the r, it can de live r the be st outcome .

Collaborative le arning is an approach that has be e n use d in machine le arning re se arch for many ye ars. since machine s can le arn from our inte ractions and e xpe rie nce s, we can e xpe ct that the y will also be able to le arn from e ach othe r, bringing forth mode ls that can combine more than one te chnology for the be st re sults.

● inte llige nt computing

In a bid to e nable use rs to e njoy more utility from the ir de vice s and ne twork se rvice s, we can e xpe ct e nhance d computing ability ove r the ye ars. the computing e nvironme nt will e volve to se rve use rs in a pe rsonal capacity. to do this, de ve lope rs and te ch companie s will ne e d to e mploy innovative api kits to he lp the m build applications that can addre ss use rs' ne e ds at a pe rsonal le ve l.

This de mands inte llige nt de vice s, products, and se rvice s that form consiste nt inte raction, can unde rstand your ne e ds and addre ss the m accordingly. following this de ve lopme nt approach, we can e xpe ct to se e more products that active ly use spe e ch and facial re cognition to ide ntify use rs. many de vice s curre ntly use iris scans to authe nticate pe rsonal acce ss syste ms.

● ve ctor proce ssing

Machine le arning and artificial inte llige nce we re once no more than the ore tical conce pts. today we are living through an age whe re we inte ract with the se mode ls all the time . the ne xt ste p in that e volutionary chain is quantum computing. for the most part, quantum computing e xists in the ory. many e xpe rts are still trying to unde rstand it be tte r.

Te ch companie s that have the right financial muscle might alre ady be te sting diffe re nt approache s. howe ve r, for mainstre am use , we can only e xpe ct to e xpe rie nce the be st of quantum computing in the ne ar future . whe n fully imple me nte d, this should spe e d up the pe rformance and e xe cution of machine le arning algorithms, making the m pe rform be tte r e spe cially in ve ctor proce ssing.

● he althcare se rvice s

The de mands of the he althcare industry are imme nse , and by adopting machine le arning the re is a lot of promise . one of the challe nge s that e xpe rts have in he althcare is the inability to diagnose illne sse s in good time . many patie nts only come to hospitals whe n the y are too sick to stay at home . by the time the y se e a doctor, the ir conditions are ofte n worse and in some case s be yond he lp.

Inte grating machine le arning into the he althcare industry will se e a shift in how illne sse s are diagnose d. one way of doing this is by he lping doctors ide ntify high-risk patie nts for spe cific illne sse s. upon ide ntifying the se patie nts, it is e asie r to he lp the m be fore the ir condition worse ns. with the machine le arning mode ls taking on most of the diagnostic work, doctors and othe r e xpe rts in the he althcare industry will de vote more time to the ir patie nts. this tre nd might also se e a shift in spe nding in the he althcare industry, with significant e fforts de vote d to pre ve ntive practice s inste ad of waiting for e xtre me tre atme nt me asure s whe n the patie nt’s condition ge ts out of hand.

● finance and re al e state manage me nt

It is no se cre t that machine le arning and the world of finance are a match made in he ave n. we have witne sse d some of the be st imple me ntations of machine le arning in the world of finance more than any othe r industry. give n how fast blockchain te chnology is spre ading worldwide , this is anothe r se ctor whe re machine le arning will be ve ry use ful.

Anyone in the financial inve stme nt busine ss will ce rtainly find pre dictive machine le arning algorithms coming in handy. the se mode ls are built to study diffe re nt se ts of data in the marke t which the inve stor can use to de cide how to inve st the ir mone y in diffe re nt portfolio classe s.

Machine le arning is not only confine d in the financial marke ts, but its adoption is also re le vant in the wide r e cosyste m, including re al e state manage me nt and othe r allie d industrie s. in the re al e state marke t, machine le arning can be imple me nte d through crm platforms. crm platforms ge ne rally hold a lot of data about diffe re nt prope rtie s and buye r profile s. from this database , pote ntial inve stors will have acce ss to startups and othe r viable prope rtie s that the y can inve st in, which suit the ir re spe ctive risk profile s.

Whate ve r be come s of machine le arning in the future will be an aggre gate of e mbracing ne w ide as, tre nds, and te chnologie s. it is e asy to focus only on how much le arning machine s will be e xpose d to. howe ve r, we will also le arn. the longe r we inte ract with machine le arning mode ls, the e asie r it will be to unde rstand the syste ms, how to inte ract with the m, and how to se amle ssly inte grate the m into our daily live s.

CONCLUSION

As we come to the end of this book, you realize that python is one of the top languages you need to learn to help you in your quest for machine learning skills. the future is already with us, and if your career path leads you to a world of data, machine learning and deep learning, you are in the right place .

One of the highlights of machine learning with python throughout this book is the need and desire for learning and constant retraining. we have covered a lot of concepts in machine learning, artificial intelligence , and deep learning, all of which are leveraged by one thing - data. if there ever was a time when data was more important in our lives, it is today.

Data forms the foundation of machine learning. without data, it is impossible to train our models to perform the tasks we expect them to. without correct data, we cannot expect the desired outcome from our data. therefore , it is evident that to learn and perfect your skills in machine learning, you must also spare some time and learn more about data analysis and other techniques used in preparing data until it is fit for interpretation or use in machine learning models.

Python is by far the best language you will use not just for machine learning purposes, but also for general programming. in this book, we introduce some of the fundamental concepts in python that are leveraged on machine learning and deep learning. however, python is a wide discipline so you should take the challenge and go further. the more you learn about python and its functionalities, the easier it is for you to become proficient in machine learning and other forked elements that are associated with python.

While still on python, we introduced some of the main libraries that are relevant to machine learning, such as keras, tensorflow, and scikit-learn. this book basically introduces you to the key concepts. it is important to

le arn this so that you can unde rstand the inte rope rability and how e ach of the se librarie s are conne cte d to one anothe r. since the se are ope n-source librarie s, you can e asily imple me nt the m across the board, or e ve n use the m alongside othe r librarie s like matplotlib that are not cove re d in this book, but will be important in work with data and machine le arning mode ls.

As a be ginne r in machine le arning, it is important that you le arn how to te ll apart the diffe re nt te chnologie s that you use . machine le arning, artificial inte llige nce , and de e p le arning are constantly use d inte rchange ably ye t this is not suppose d to be the case . machine le arning and de e p le arning are offshoots of artificial inte llige nce . the re fore , you should ide ntify the m as such from the ve ry be ginning. failure to do that might have you confuse d ove r time as you dig de e pe r into e ithe r of the se te chnologie s.

The topics discusse d in this book are spe cifically chose n to he lp you ge t a broad introduction to machine le arning. this way, you have the chance to le arn a lot about machine le arning and associate d te chnologie s without e xe rting undue pre ssure on yourse lf. e ach of the subje cts discusse d in this book are broad conce pts that can be studie d inde pe nde ntly in the future as you familiarize yourse lf with machine le arning te chnologie s.

What doe s the future hold for machine le arning? pre dictably, the te chnologie s we use today will advance furthe r ove r the coming ye ars. luckily, most of us have live d through the rapid e volutionary phase of dynamic computing. the re fore , we have had the chance to e xpe rie nce and inte ract with diffe re nt machine le arning mode ls in the ir rudime ntary forms, and the ir advance d forms afte r consiste nt optimization ove r the ye ars. the future of machine le arning and de e p le arning is bright. while we can alre ady se e how some of the mode ls have be e n imple me nte d in our daily live s, the re is still room for improve me nt. we have so much to look forward to which make s the study of machine le arning e ve n more inte re sting for you.

We discusse d some ways machine le arning has be e n imple me nte d in diffe re nt busine ss proce sse s today, and how many are making live s e asie r for a lot of pe ople . the imple me ntation and inte gration might have starte d in industrial proce sse s, but with time , we inte ract with machine le arning mode ls close r home . in the ne ar future , we will e xpe rie nce more inte gration of de e p le arning and machine le arning mode ls in our live s, he lping us make important de cisions in re al-time .

Finally, as you be gin this journe y into machine le arning, take he art and be confide nt. working with data is an incre dible opportunity. you will inte ract with diffe re nt forms of data you might have ne ve r imagine d possible . you will inte rpre t and manipulate data in many ways and draw conclusions from the m. more importantly, you are le arning how to communicate with machine s, and te ach the m how to pe rform tasks that would othe rwise take humans longe r to accomplish, or would not succe e d in the first place . as we ushe r in a ne w age in the human-machine inte raction, re me mbe r that you are in the driving se at.