Demystifying Artificial Intelligence

Simplified AI and Machine Learning Concepts for Everyone

PRASHANT KIKANI

bpb

Demystifying Artificial Intelligence

Simplified AI and Machine Learning Concepts for Everyone

Prashant Kikani

www.bpbonline.com

FIRST EDITION 2021

ISBN: 978-93-89898-705

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

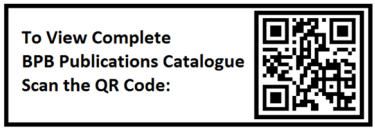
Distributors:

BPB PUBLICATIONS
20, Ansari Road, Darya Ganj
New Delhi-110002
Ph: 23254990/23254991

DECCAN AGENCIES
4-3-329, Bank Street,
Hyderabad-500195
Ph: 24756967/24756400

MICRO MEDIA
Shop No. 5, Mahendra Chambers,
150 DN Rd. Next to Capital Cinema,
V.T. (C.S.T.) Station, MUMBAI-400 001
Ph: 22078296/22078297

BPB BOOK CENTRE
376 Old Lajpat Rai Market,
Delhi-110006
Ph: 23861747



Published by Manish Jain for BPB Publications, 20 Ansari Road, Darya Ganj, New Delhi-110002 and Printed by him at Repro India Ltd, Mumbai

Dedicated to

My family and friends, who have provided support throughout my life

About the Author

Prashant Kikani is an experienced data scientist, who has ranks as *top 1% worldwide* in competitions and kernels on Kaggle, which is the world's largest community and platform for data science and machine learning.

As part of his day-to-day work, he is trying to solve some of the hardest problems for the human kind, like language translation using state-of-the-art deep learning based NLP models and infusion of knowledge graphs in NLP models. He is one of the youngest students to achieve the Master title on the Kaggle kernel platform. Also, he has worked on other deep learning sub-fields, like computer vision via Kaggle competitions.

His interest lie in AI/ML and deep learning, and teaching others what he has learned in a simple and intuitive manner. This book is part of his interest to share his knowledge in the simplest possible manner with everyone so that they can learn about this fascinating technology called AI!

Prashant's ultimate goal is to make a positive impact on millions of people's daily lives and to always be optimistic about the future. To achieve these goals, he has started multiple initiatives. You can know more about him at **https://prashantkikani.com**.

About the Reviewers

- **Shikhar Kwatra** works as a Data and AI Architect in IBM with primary focus on Operationalizing AI models as core ML-Ops Leader. He pursued his Master's in Electrical Engineering from Columbia University. He is recognized as the **Youngest Indian Master inventor** with over 250 Filed Patents and Inventions in the areas of AI/ML, IoT, Blockchain, Drones etc. He also holds the title of one of the youngest *Academy of Technology* members in IBM. He focuses his spare time on exploring AI-IoT technologies and leverages his technical background to work on new ideas with the aim to inspire every engineer to think from an inventive mindset.

  Know more about Shikhar here: **https://www.linkedin.com/in/shikharkwatra/**

- **Darshan Patel** has 1.5 years of experience in the field of data science, such as building ETL data pipelines, and developing machine learning algorithms to enable the businesses to take data-driven decisions. Darshan pursued B.Tech CSE with Specialization in Big Data Analytics from Ganpat University. He is working as a Data Science Engineer in Embibe.

Acknowledgement

I would like to thank my parents, Mr. Vinodrai and Mrs. Gulab, who have always supported me for whatever dream I wanted to pursue.

Also, I would like to thank some of my friends, who have always helped me and provided support whenever I needed it. My gratitude goes to them for being with me in my journey and allowing me to be a part of their journey as well.

This book wouldn't have come into existence if I never started exploring the machine learning and AI field in my first year of college. So, I thank all the seniors who have guided me in this field.

Finally, thanks to the publishers of this book for giving me the opportunity to write this and share my thoughts to millions of people.

Preface

In this book, we will to cover a number of topics which fall under artificial intelligence (AI).

Without requiring any kind of prerequisites, you will learn many exciting topics and concepts in the AI field in plain, simple and easy-to-understand English language.

I have made it very easy for you to understand all these mathematical based concepts by conveying them in a simple and effective manner while keeping the original concept intact.

Recently, the biggest breakthroughs in the field of AI research are being made in machine learning, especially in deep learning fields.

Why? Mainly because of the following two factors:

1. Availability of huge amounts of data
2. Comparatively cheaper powerful computation resources like **GPUs (Graphics Processing Unit)**

It has become easier to train machine learning systems to do certain tasks like never before.

My main focus will be on machine learning and deep learning subfields as that's what drives the artificial intelligence field nowadays.

Artificial Intelligence (AI) is a fascinating field as we are trying to copy a thing which is right there in our skull; the human brain.

It's so complex, no one to this date completely knows how it operates and takes decisions.

By the way, **Mckinsy** says that AI is to create $13 Trillion Value by 2030, mostly to be used in retail, followed by the travel and automotive sector.

This may be a good reason to gather some knowledge about AI!

Though the AI field is several decades old, it came to people's attention after big tech companies started using AI and especially machine learning in their products in the last decade (i.e. 2010-2020).

Before that, AI was kind of limited to research labs and universities.

Now, companies, especially the newly emerging startups, are using AI and ML technologies in their main flagship products. Some startups are completely based on AI and machine learning.

One of the most exciting things is that AI and machine learning are not just limited to computer related products and companies.

Today, we are already leveraging benefits of AI and ML in fields like,

1. Medical and health care
2. Defense and security
3. Climate change
4. Entertainment
5. Infrastructure
6. Education
7. Economics and finance
8. Sports
9. Arts, like paintings, singing etc.

   And in many other fields as well.

With this scale of usage, basic knowledge about AI and machine learning is good to have for almost everyone in the 21st century.

That's what the goal for this book is. *Without any requirements of prerequisites*, this book covers many diverse topics of AI, especially machine learning, in very simple non-technical and easy-to-understand language.

The basic goal of AI and machine learning is **automation**. Companies try to automate everything they can to increase customer's engagement and profits.

Let's see some of the following examples:

- Google uses it to automate its best personalized search results.
- Netflix and YouTube use it to recommend the best movies / TV shows / videos for each customer to create more engagement.
- Uber uses it to select the best ride for you.

- Facebook uses it to automatically tag all the persons in a photo post.
- Amazon uses it to lower the return rates of shipped products.
- Google's Gmail filters out spam emails from our inboxes.
- Finance companies and banks detect fraudulent credit/debit card financial transactions.

We will see why automation is crucial for businesses and how AI helps them in the upcoming chapters of this book.

It is not only limited to automation anymore; companies are now building **whole new markets and products** around AI.

- Tesla, Uber, and Lyft created self-driving vehicles.
- Google is using speech recognition in their home related products.
- Tinder is using AI to figure out who you're likely to "Super Like"!
- **Google, Amazon** etc., are making the **Internet of Things (IoT)** based smart home products like **Google home** and **Amazon Alexa**. AI based IoT is a whole new market.
- Google is using their language translation AI, Vision AI, and NLP to generate revenues by selling them to other businesses, schools, and universities.
- Companies make chatbots to lower the traffic to customer service call centers. Chatbots are a whole new market today.

And make no mistake, all these things were done in the last decade alone. Rate of progress in the AI field is very fast. Big companies and universities have dedicated teams for research and development (R&D) in AI and machine learning.

AI automates or will automate what a human can do. For example:

- It can recommend you movies like a pro movie reporter/critic.
- It can predict cancer in human breasts like an experienced good doctor.
- It can help radiologists to pick out tumors in x-rays images.
- It can drive cars like an expert driver.
- It can suggest your future life partner on dating sites like a pro matchmaker.

- It can detect fraudulent transactions like a master CIA agent.

Additionally, AI can do these jobs 24x7, 365 days in a year with any cost!

How cool is that?

Field of Artificial Intelligence is filled with its own technical words and definitions. **I will try to avoid those words as much as possible while keeping the original concept intact.**

We will explore those mathematical and technical concepts in easy and non-technical terms.

Even though AI technology is in its early stage, AI backed businesses and companies are already launching AI based products/features and making tons of money!

Along with knowledge of AI and machine learning concepts, we will also see the business side of them. Like, how one can monetize AI technology. **In other words, how to earn money from AI and machine learning.**

Our goal for this book is to learn what artificial intelligence is and how things happen in it. We will first understand the basic meanings of terms and then we will discuss how and why things happen.

Chapter 1 is all about the introduction of the field. Meanings of various terms related to the AI field and relations between them.

Chapter 2 covers an in-depth explanation of all the subfields of AI and machine learning and gives you an idea about how everything actually works at its core.

Chapter 3 is about the business perspective of AI. How small, medium, and large companies are leveraging AI to maximize their profits and earning billions using AI. We'll also see how you can start leveraging AI in your company/business.

Chapter 4 is all about how to get started in the AI field and what are some of the pitfalls to avoid while leveraging AI and machine learning.

So, let's start!

Downloading the code bundle and coloured images:

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/e9b75

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

BPB is searching for authors like you

If you're interested in becoming an author for BPB, please visit **www.bpbonline.com** and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Demystifying-Artificial-Intelligence**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at **https://github.com/bpbpublications**. Check them out!

PIRACY

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at :

business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**.

REVIEWS

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Table of Contents

Chapter 1
Introduction

In this introductory chapter, we will explore some of the basic terms related to **artificial intelligence (AI)** such as machine learning, deep learning, data science, neural networks, etc.

After reading this, you will know what "the field" really is, what each term means, and how each part connects to the other. We will take a look at all this by keeping it short and simple.

Structure

In this chapter, we will cover the following topics:

- What is data?
    - How to convert images, text, and audio data into numbers?
- What is AI?
    - Subfields of AI
    - Types of AI
- What is ML?
- What is DL?

- What are artificial neural networks and how are they built?
- What is "*deep*" in deep learning?
- Advantages/limitations of DL
  - What is data science?
  - Why do we need AI?
- Why does automation matter?
  - Machine learning and maths
- Why is ML so dependent on maths?

Objectives

After studying this chapter, you should be able to:

- Learn about various terms related to AI and their correlation
- Have a general idea about what an AI system might look like
- Understand how machines can do something better than humans and what makes them so much better at some tasks

What is data?

Before we start discussing AI and machine learning, let's first discuss data.

Data is the main reason why AI and ML algorithms/models are able to learn anything meaningful. It plays a crucial role in the performance of any ML model. In general, data can mean anything like tables in spreadsheets, images, videos, audios, text, track records, etc. We use data to teach machines/computers, but, guess what, machines can only understand numbers. Other than numbers, it can't understand anything. Data is the crux of every ML model, and data is the only thing that the ML model gets from us. Quality and quantity of data are very important as that's what the model uses to learn something. As long as we can somehow convert something into numbers, it "becomes" data for the machine learning models.

(By the way, a model is just an algorithm that learns to capture patterns in data. Basically, we teach models to make predictions in the future.)

It is okay for spreadsheet tables as they already have numbers, but what about images, audios, videos, or text? They are not numbers. Then, how can a machine learn from them?

Computers only understand numbers; nothing more or less. They operate in 0 and 1. So, we need to communicate with computers in numbers because that's what they understand.

Well, if the data is not in numbers, we convert it into numbers.

How? Let's see.

1. Text

Text is made from different words. We give each word a unique number. Let's say, there are a total of 80,000 unique English words in our test dataset. We give each of them a unique number. So, the sentence "*machine learning is very easy*" becomes something like "*19768 45734 3 2349 459*". Now, machines/computers can read this sentence. This process is called encoding in technical terms. Did you get the idea? We replaced each word with its corresponding number to convert text into numbers. While doing this, we maintain the order of the words.

2. Images

Actually, an image is already in the number format. When we capture a photo or shoot a video from a mobile phone or camera, it's stored in numbers in memory. So, unlike text data, there is no need to exclusively convert images into numbers.

An image is nothing but a group of pixels in a certain order.

A pixel is the smallest unit of an image. Every pixel has a number associated with it. So, in a nutshell, an image is nothing but thousands of numbers in a certain order. For example, let's take a look at the following image:

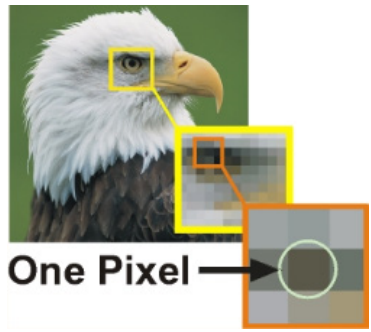


Figure 1.1: *How a pixel looks like if we zoom in a photo*

If we zoom the image to some extent, we can see each pixel. You may have seen these types of pixels in your mobile or camera. Each number in a pixel represents how much of these fundamental colors (i.e. Red, Green, and Blue) contribute to a particular pixel.

3. Audio

The audio or sound is transmitted by waves in the air. We convert these waves into numbers. How? It's a bit technical, but in simple terms, we store something called the *amplitude* of the waves at every small interval of time.

A sound wave becomes a series of numbers. We just need to represent every voice signal with a unique chain of numbers. So, we picked the amplitude values for the wave at every small time interval

The essence of voice recognition is that no matter who speaks the word, the series of numbers almost always remains more or less the same for that particular word.

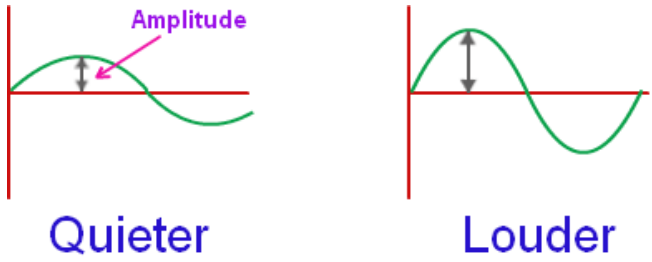


Figure 1.2: *Quiet and loud sound's amplitudes*

For example, let's take the word *hello*.

No matter who speaks the word *hello*, whether it's a male from Asia, or a female from Australia, or a kid from America, amplitudes for the word *hello* for each of them will remain more or less similar. That's what a machine *learns* in speech recognition. The machine captures the same pattern of numbers to detect the spoken word. Converting all these types of data into numbers comes under something called the data preprocessing part.

We play around a lot with data as data is the thing that makes or breaks the performance of our AI/ML model.

In machine learning, it's all about data. In real-world projects, most of the work goes into making the data right. In preprocessing, along

with converting data into the number format, we also do multiple things like:

- Filling missing values
- Feature engineering
- Scaling/normalizing
- Augmentation

We will read about this in the upcoming chapters. Some well-known open-source datasets for images are ImageNet and Google's open images. **ImageNet** has more than 14M images that were created by more than 50k people around the world.

Because of these huge datasets, the progress of AI and machine learning fields is skyrocketing. If you want to get hands-on experience on Python coding for machine learning, take a look into the following Python notebook: *Introduction to Basic Python Libraries for ML* (**https://github.com/bpbpublications/Demystifying-Artificial-Intelligence/blob/main/Chapter01/introduction-to-basic-python-libraries-for-ml.ipynb**).

In this Python notebook, I have covered basic Python libraries which every professional data scientist and ML engineer uses. So, in machine learning, data is made from numbers. No matter if it's text, audio, or image, we can still convert them into numbers.

What is artificial intelligence (AI)?

AI is a very broad field in science whose ultimate goal is to make computers/machines to make decisions on their own in tasks which require intelligence and reasoning.

In other terms, as you might have guessed, to create intelligence artificially! We know one form of intelligence that is our very own biological brain; though, we don't quite understand how it operates, takes decisions, and stores information.

Figure 1.3: *Human brain, a mystery yet to be solved*

The earliest definition of AI from 1950 is *any task performed by a computer program or a machine that if a human carried out the same activity, we would say the human had to apply intelligence to accomplish the task.* Another way to describe intelligence according to *Francois Chollet is the ability to understand something from a partial description.*

How minimalist this description can be is a measure of intelligence. As of now, AI is made of multiple fields like computer science, neuroscience, mathematics, statistics, etc. That doesn't mean you need to understand all these fields before entering into AI. Almost anyone can learn about AI with almost no prerequisites.

In fact, that's what this place is about; to make AI simple enough so that everyone can understand it at its core. Some popular fields that fully/partially fall under AI are as follows:

- Machine learning (ML)
- Deep learning (DL)
- Computer vision
- Natural language processing (NLP)
- Robotics
- Knowledge representation
- Logic
- Reasoning
- Problem solving
- Creativity

There are some controversies/confusion about which field falls under which. The only thing is the one that I have mentioned in the diagram. All we know is that they all fall under the AI field:

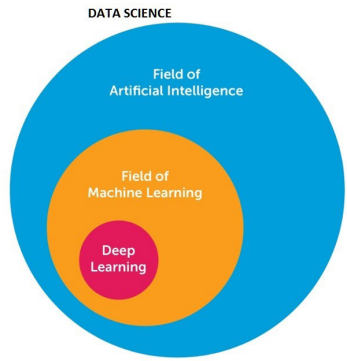


Figure 1.4: *Hierarchy of the artificial intelligence field*

Keep in mind that AI is a very broad term.

As it's a comparatively new field, many people confuse AI with terms like machine learning, deep learning, robotics, **Internet of Things (IoT),** etc. Actually, all of them are subfields of AI. Like all other subfields, **machine learning (ML)** and **deep learning (DL)** both come within AI. When we hear 'AI', it just means that it's something related to *making a machine make intelligent decisions on its own,* almost nothing more or less than that.

The goal of AI is to enable machines to make intelligent decisions on their own.

To achieve this, we need to teach machines how to learn from data as machines are not living objects. They cannot learn on their own. We have already seen what data is. We use this type of data to teach machine-specific tasks like seeing images, reading and understanding text data, making a conversation, etc.

Let's make it easier.

We can imagine AI to be like a little child. Children learn some things in school. The child's teacher teaches him/her to read, write, etc. Teachers also take tests of children to check his/her understanding. We use similar approaches to teach machines just like how a human child learns.

Types of artificial intelligence

The two types of AI are as follows:

- Narrow AI
- General AI

Let us understand them one by one.

1. Narrow AI

Today, as of Feb. 2020, we live in a narrow AI world. Almost everything about AI we see all around us falls under narrow AI.

Narrow AI, as the term says, is only focused on specific tasks.

For example, spam email filtering. As of now, an AI system that filters out spam emails cannot drive a vehicle or detect fraudulent financial transactions. It can just filter out spam emails. That's all.

Unlike narrow AI systems, we humans have one brain that can do multiple tasks. We can drive cars, talk to people, and identify what's there in a photo with a single brain!

Narrow AI only does specific tasks. Narrow AI focuses on mastering specific tasks like image classification, detecting faces in the photo, etc. They can't do multiple tasks together. Applications of narrow AI are endless. From helping doctors to detecting tumors in X-ray images to visual inspection of infrastructure such as ongoing construction in real-estate using drones.

Here's what an imaginary conversation between a human and narrow AI will look like:

Conversation time

Human: Hey Machine, I need you to learn to recognize faces in images. Here is the data. Learn from that data.

Narrow AI: Okay. I will learn to recognize faces in images from the data you gave me.

[*Machine is capturing patterns in data after some time.*]

Narrow AI: Okay. I have learned to recognize faces in images.

Human: Cool. Now, I want you to also learn to predict whether a financial transaction is fraudulent or not.

Narrow AI: What? No. I can't do that. I am a Narrow AI. I can only learn one task at a time. I have to forget to recognize faces in images to learn to predict whether a financial transaction is fraudulent.

Human: Oh, you can't do that! Maybe general AI can do this.

Narrow AI: Yes. General AI can do this.

2. General AI

We haven't reached this stage as of now. It's hypothetical.

General AI, just like our human brain, is one single adaptable and flexible form of intelligence that can learn diverse tasks.

It is also known as **Artificial General Intelligence (AGI).** If predictions about AGI become reality, AGI will be able to do almost anything that a human does plus some other tasks that humans can't

do. Scientists are predicting that there are high chances that AGI will become a reality between the years 2050 to 2075.

So, artificial intelligence (AI) is a broad field with the ultimate goal to make machines to make decisions on their own.

Let's explore machine learning now.

What is machine learning (ML)?

Machine learning is a subfield of AI in which a computer system "*learns*" to perform specific tasks by capturing and remembering relevant patterns in data that we have given to it. So, in a nutshell, this is what machine learning is:

Capturing relevant patterns/characteristics in the data and remembering them for future use.

That's it. Is it that simple? Yes. It's all about capturing relevant patterns and remembering them.

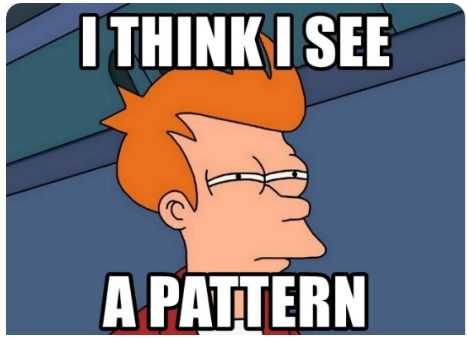


Figure 1.5: *Machine learning models are like.*

Capturing relevant patterns in the data is the core thing in machine learning. This is how machines "learn" to do different tasks.

The machine can capture patterns in some tasks such as:

- Predicting whether a fruit in the photo is an apple or a banana or an orange.
- Checking whether the use of the word "bank" in a sentence relates to a river bank or a financial bank.
- Recognizing speech accurately enough to generate captions for a YouTube video.

The best part about machine learning is that we don't need to give the machine any instructions on how to do the task or something like that. The machine will figure it out itself. It will "*learn*" to do tasks on its own. That's what makes machine learning different from traditional computer software.

The way we capture patterns depends on the data; whether it's an image or text or audio, etc. But, capturing patterns is not easy.

It's simple but not easy.

It's the same way how we humans operate. We also capture patterns to perform specific tasks, but unconsciously most of the time. Let's take an example. What do you see in the following image?

Figure 1.6: *Banana*

Banana right?

Yes. How do you know? Think!

You have captured some patterns which are unique to bananas such as it is yellowish in color, it has a **unique shape;** its specific textures on its surface, etc. We have already remembered those captured patterns in our minds. So, the next time we see something like this, we call it "banana".

That's how our brain captures patterns in things and whenever those similar patterns emerge, it comes to know about things.

When you were a baby and your mom was holding a banana in her hand and telling you "this is a banana", your mind was capturing those patterns and remembering it. Here, "banana" is a kind of a label for those captured patterns. So, whenever you capture the same patterns again, you call it a banana.

Here's what a conversation between a human and a machine will look like in machine learning.

Conversation time

Human: Hey Machine, I need you to learn to translate English sentences into the Spanish language. Here are 1 million sentence pairs; English and their respective Spanish translations.

Machine: Okay sure.

….[Machine is capturing linguistic patterns like grammar and syntax.]....

Machine: Okay. I have learned to translate English sentences into Spanish!

Human: Cool. Then, tell me what's the Spanish translation of the English sentence, "Hello, how are you"?

…. [Machine is remembering all the linguistic patterns that it has captured while training and using them to translate English sentences.]....

Machine: I think, the Spanish translation is, "¿Hola como estas?"

Human: Cool, you are correct. Good boy!

Let's take another example. If you hear the sound of dogs barking, you can predict or estimate the presence of some dogs nearby, right? How? Think again. It's the same concept as we just saw in bananas.

You have captured those similar sound waves, and you know from previous experience that it was the sound of a dog. So, this time too, it must be the dogs barking. You captured the pattern and you remembered it.

Here, "dogs barking" to that sound kind of acts as a label. We labeled that sound with "dogs barking". In a nutshell, that's how a computer or a machine "learns" from data.

So, now you know why we feel amazed and confused in magic shows.

Because whatever patterns we have captured till date, all of them break when we see magic shows. It shocks us. You know why? Because it's against what we see in the real world.

It goes against the patterns we have captured till now. That's why it amazes us. Now, you kind of know why people love magic shows. Let's explore machine learning a bit more. There are mainly

two types of machine learning. They call it supervised and unsupervised.

So, what are they?

1. Supervised machine learning

"Supervised" is that kind of learning type in which we provide some labels (like banana or dogs barking, etc.) along with each data point. We will explore this in the latter part of this book.

2. Unsupervised machine learning

As you might have guessed, "unsupervised" is that kind of learning type, where we only provide data points but don't provide labels for them. We will explore this as well in the latter part of this book.

There is also a third type called semi-supervised learning which is kind of in-between supervised and unsupervised. We will read about the semi-supervised method in detail in the next chapter.

The basic idea of semi-supervised is to use less labeled data to gain better performance in different tasks.

Creating labeled data is costly in terms of money and time. That's why the usage of semi-supervised is increasing. In the news or social media, whatever we see about artificial intelligence (AI), most of the time, it is actually machine learning (ML).

Machine learning is enabling computers to tackle tasks that have, until now, only been carried out by people.

The ML field also has subfields. One of them is deep learning. We will see that in the next chapter. Here's the link to a Python notebook to solve your first problem: *1st step in Machine Learning - The Titanic problem* (**https://github.com/bpbpublications/Demystifying-Artificial-Intelligence/blob/main/Chapter01/1st-step-in-machine-learning-titanic-problem.ipynb**). In this notebook, I have explained 6 steps that we can follow to solve almost any ML problem. Check this out if you have an interest in how ML appears in code.

So, remember one thing. In a nutshell, machine learning is capturing patterns and remembering it for future use.

Let's now see what deep learning is.

What is deep learning?

Deep learning is a subfield of machine learning which only uses one specific type of model called artificial neural networks to capture patterns and remember them.

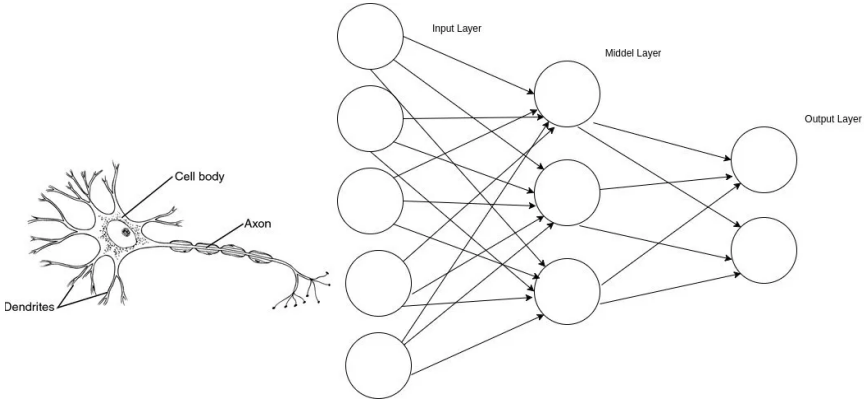


Figure 1.7: *A biological neuron on the left and a simple artificial neural network on the right*

In the preceding figure, the *left* part contains our biological brain's neurons. Our brain is made from billions of those. The *right* part is a sample artificial neural network. Each circle is a neuron. In artificial neural networks, each neuron is fully or partially connected with other neurons.

Going from left to right in the artificial neural network from the preceding figure, the first layer is called the "input" layer and the last layer is called the "output" layer.

Easy enough!

And middle layers are called "hidden" layers. Why hidden? Because they are not visible from outside. Only the input and output layers are visible. They are in between the input and output layers.

So, what are artificial neural networks then? They are kind of a copy of the human brain itself, but on a very tiny scale.

Yes. You read it right.

The core idea of neural networks is inspired by our own biological brain.

Neural networks are a network of neurons! They are called "**artificial**" as they are created by humans, not biology. They are not natural.

Each neuron is connected with some other neurons and this network formed from neurons is called a neural network.

That was easy, right?

What's a neuron then?

In a nutshell, we can imagine it as an electric wire which sometimes passes incoming signals from previous neurons to the next connected neuron, and sometimes it doesn't pass the signal to the next neuron. That was also easy, right?

Then, how does the neuron decide when to pass the incoming signal and when not to? Well, that's what it needs to "learn" from the data we give.

We have some data at hand and we give it to a neural network (which can contain tens of thousands of neurons connected to each other); it learns to capture patterns from them. We will see how a neuron "learns" when to pass the signal and when not to in upcoming phases.

Just remember one thing.

There's something called "weights" which is associated with "when to pass the incoming signal and when not to."

We will explore this in the upcoming chapters.

Ladies and gentlemen, that's it! That was the neural network, a marvelous innovation of the 20th century, in a nutshell.

The whole world is using the preceding model — Facebook, Google, Amazon, Microsoft, Apple, Alibaba, Baidu, Tencent — everyone is using the model we just discussed earlier and making billions of dollars!

It's these deep neural networks that have fueled the current leap forward in the ability of computers to carry out tasks such as speech recognition and computer vision. Some examples where deep learning is useful are predicting whether a given photo has a male's face or a female's face, translating sentences from one language to another language, listening and understanding the human voice and replying to them accordingly, etc.

All of the preceding tasks are almost very easy to do for humans. But it was never possible before to teach these tasks to machines.

All neural networks have an input layer through which data is fed in and an output layer through which we get a final prediction from the neural network. (Refer to the *figure 1.8* of leaf classification.)

Now, in deep neural networks, there will be multiple hidden layers. Hidden layers are those layers that are in between input and output layers.

What is "deep" in deep learning?

We use the term "deep" as there can be multiple middle/hidden layers in a neural network. So, because of those multiple layers, our network becomes deep. That's why people call those "deep" neural networks.

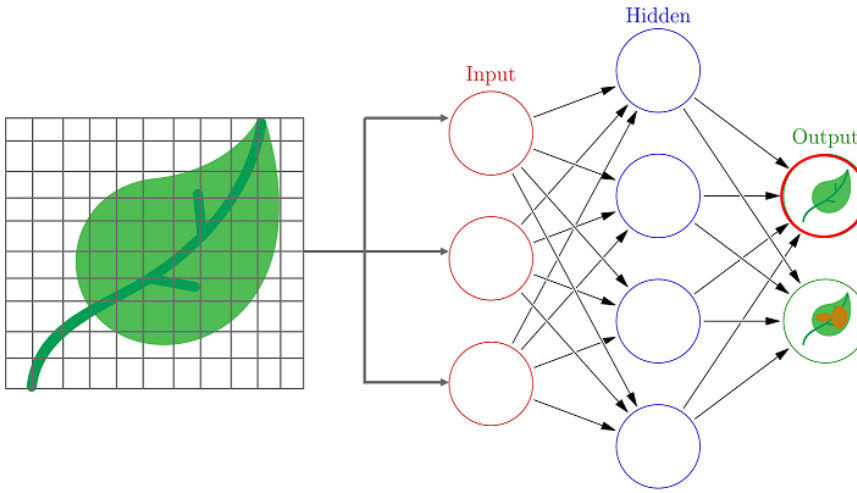


Figure 1.8: *Input, hidden, and output layers in a neural network predicting a leaf having some disease or not*

Each hidden layer feeds data into subsequent layers.

In the preceding diagram, each circle represents a **neuron** in the network with the neurons organized into vertical layers. As you can see, each neuron is linked to every neuron in the following layer, representing the fact that each neuron passes a value into every neuron in the subsequent layer.

The output of one layer is the input of the next layer in the network with data flowing through the network from the input to the output layer. Deep learning is responsible for amazing technologies like image recognition, natural language processing, speech recognition, etc.

Roots of the deep learning field are highly dependent on a branch of mathematics called "calculus".

You might have heard terms like "differentiation" or "integration", etc.

Those are some things that come under the calculus field. In deep learning, we use differentiation to teach our neural network. Things get a bit mathematical from this point. It's not that hard but we don't need to read that in details for now. There are various types of neural networks, with different strengths and weaknesses; for example, **convolutional neural networks (CNNs)** are generally used for images, and **recurrent neural networks (RNNs)** are generally used for sequential data like text.

Let's see the pros and cons of deep learning.

Advantages of deep learning

Listed here are the advantages of deep learning:

- They are really good on unstructured data like images, text, and audio. Other ML algorithms are not that good for this type of data.
- Feature engineering is not required. (By the way, feature *engineering* is giving our model more relevant data using our domain knowledge. We can create more relevant data out of the existing data.)
- As feature engineering is not needed, domain knowledge is also not needed.
    - Anyone with literally no knowledge about lung cancer can build a neural network that can predict lung cancer better than an experienced doctor.
- They can capture really complex patterns in the data. Normal ML algorithms sometimes can't capture very complex patterns.

Limitations of deep learning

The following are the limitations of deep learning:

- We need a huge amount of data to train them compared to other machine learning algorithms.
- They are comparatively hard to train in terms of the amount of computation power needed. Training often requires access to high-powered and expensive computer hardware, typically high-end GPUs, and they are costly.
- They are not much transparent compared to other ML models. In other words, we can't really know why the neural network is predicting this answer.
- They can be easily fooled by adversarial examples. We will see this in detail in the section of limitations and advantages of machine learning.

However, despite the preceding limitations, universities and companies are heavily investing their resources in the fascinating research and development for deep learning models. The majority of the internet's data is in unstructured format, i.e., images, videos, text, audios, etc.

That's a big win for neural networks as they are inherently good on that type of data. Because of that the number of fields where we can use neural networks is very huge.

And that's what makes neural networks so popular.

Truth be told, nowadays, researchers are more interested in deep learning compared to other machine learning algorithms.

It will be very interesting to see how new research will make a positive impact on normal people's lives. To get an introduction on deep learning via hands-on coding, check out the Python notebook: *Solving the Titanic problem - Deep Learning way* (**https://github.com/bpbpublications/Demystifying-Artificial-Intelligence/blob/main/Chapter01/solving-the-titanic-problem-deep-learning-way.ipynb**).

Thus, deep learning is machine learning with artificial neural networks. And artificial neural networks are specific types of mathematical models which are inspired from the biological brain.

So, that was deep learning. Let's now read about data science.

What is data science?

The data science field is a mixture of maths + machine learning + business knowledge + computer science + some data related software/technologies. Refer to the following figure for the data science Venn diagram:

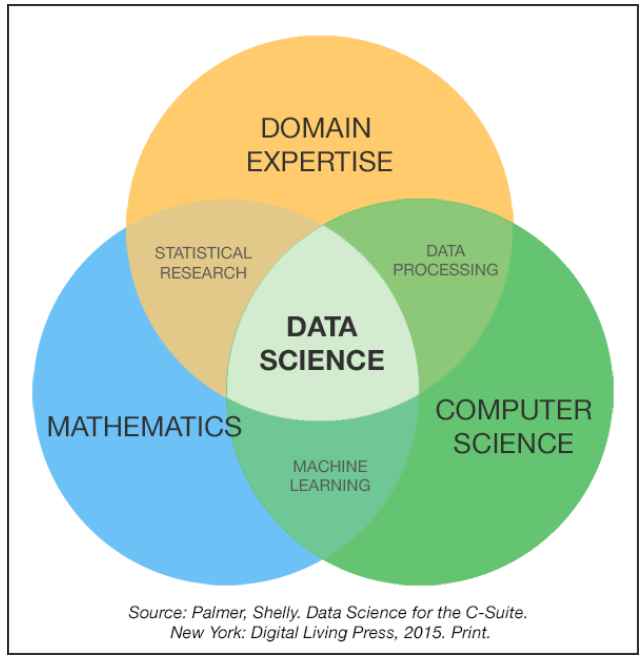


Figure 1.9: *Data Science Venn diagram*

In other words, it is science that is related to data as you might have guessed. Data scientists gather data, process them, give it to computers to learn from them, make sure they learn well, and ultimately serve it to satisfy some business needs using data related software.

As seen in the preceding figure, in the real world, data science is a little bit of machine learning, maths and statistics, business knowledge, and computer science.

It's a combo package. No wonder why most of the companies are building teams of data scientists. It's like "one key for all locks!"

So, what's the size of this data? Well, sometimes it's in gigabytes, sometimes even in terabytes or petabytes! They use tools that can handle this much amount of data.

Let me tell you what reality looks like for data science. A typical data science job in the real world can mean a candidate is training ML models or working with data cleaning and preprocessing, making graphs and plots or pie charts, etc.

The following diagram shows the relation between all the fields:

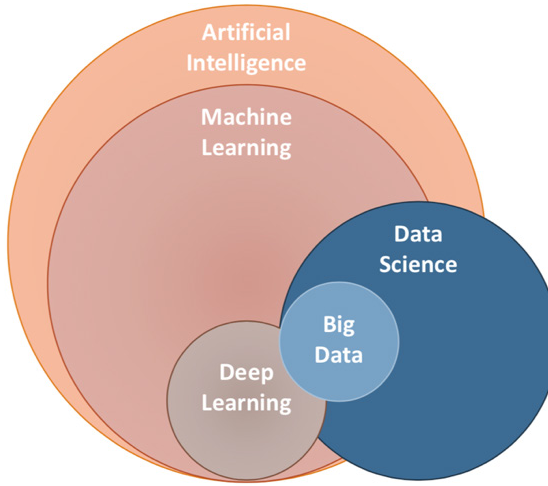


Figure 1.10: *Relations between all the fields*

So, in the real world, data science is a mix of multiple fields — machine learning + little bit of maths and statistics + little bit of business understanding + some computer science knowledge.

Why do we need artificial intelligence (AI)?

This is a very exciting question. After understanding what is AI, machine learning, deep learning, etc., one question may come to your mind — Why do we need it? What's the point of all these? What will we get if we learn AI and machine learning?

Those are very fair questions. One should ask these questions to oneself before learning about AI.

Let's see. We all want to grow in our career and business, right? No question or doubt about that. To grow more, we need to automate things in our business so that we can invest time in some other more important things.

What if someone comes to you and says, "I can do this for you. And you don't need to pay me!" That "someone" is machine learning.

You can teach **machine learning (ML)** models to do some things for you. You can automate it. Your ML models will do things for you, while you are busy with some other more important things, including sleeping!

How cool is that? That's the whole point of machine learning.

Automation

Companies do it to reduce their cost and increase their revenue. AI will help you to automate things in your business, which were only possible by humans before, which ultimately increases profits. We can calculate the profit by using the following formula:

"Revenue - Costs = Profit"

People like you and me use it to improve their productivity and lifestyle. Government uses it for security and surveillance purposes. One question may arise in your mind:

"Why does automation matter?"

The reason is that it will help you to save the cost of human employees while not compromising on the quality of work.

It is not hard to understand that automation is crucial in businesses. Companies want to increase revenues and decrease costs (and eventually, increase profits). AI can do just that.

Simple automation is also possible with simple computer algorithms.

However, machine learning enables us to automate things that only humans are able to do. We want to automate almost all the tasks and things so that we don't need to do them ourselves.

We have already seen this in the beginning but mentioning it again as it's related to the topic. Giant companies use machine learning to automate almost everything. Some examples are as follows:

- Google uses it to automate the best-personalized search results.
- Netflix uses it to recommend the best movies and TV shows for each user to create more engagement.

- Uber uses it to select the best rides for you.
- Facebook uses it to automatically tag all the people in a photo post.
- Amazon uses it to lower the return rates of shipped products.
- And many more like these.

Startups can also scale fast leveraging AI not just big companies. We are going to read about the business perspective of AI in the latter part of this book; in other words, how to monetize AI or how to earn profits from AI.

In short, to automate things on a great scale, we need machine learning. Why do we need automation? To reduce costs and improve performance.

Conclusion

So, in this chapter, we learned all the terms related to AI and saw they correlate with each other. Now, you have a general idea about what an AI system might look like, and how exactly a machine can do something better than humans.

In the next chapter, we will dive deep into these fields.

Questions

1. How artificial intelligence, machine learning, deep learning, and data science fields are related to each other?
2. What's the biggest reason that companies are adapting AI and machine learning?
3. Why does automation matter for businesses?
4. What's the core thing in machine learning?
5. Which 3 domains together make the data science field?

Chapter 2
Going Deeper into ML Concepts

In this chapter, we will go deeper into **machine learning (ML)** concepts such as supervised learning, unsupervised learning, reinforcement learning, transfer learning, computer vision, NLP, etc. We will also take a look at how machines actually "*learn*" things, what neural networks are, and how they learn.

We will read about all these in simple sentences, which everyone can understand. The goal of this chapter is to have a general idea and clear understanding of every concept in machine learning.

Structure

- Machine learning and Maths
    - Why is ML dependent on Maths?
- Types of machine learning
    - Supervised learning
        - Classification
            - What are decision trees, and how do they work?

        - Regression
    - Unsupervised learning
        - Clustering
        - Dimensionality reduction
    - Semi-supervised learning
        - Pseudo labeling
    - Self-supervised learning
        - Language models in NLP (Natural Language Processing)
    - What is metric? Why do we need it?
- How do machines "learn" to do things?
    - How does a machine see images?
    - How does a machine read and "understand" texts?
    - How does a machine listen to audio?
    - What are overfitting and under-fitting?
    - What is loss function? Why do we need it?
    - What is gradient descent?
- How does a neural network "learn" to do things?
    - What is back-propagation?
    - What are "weights" in a neural network? What's their role?
    - Why do we need a "deep" network or multiple layers in neural network?
- What is reinforcement learning?
    - What is the *reward* in reinforcement learning?
    - Real-world examples of reinforcement learning
- What is transfer learning?
    - Why do we need it?
    - The intuition behind transfer learning
    - How do we use transfer learning in deep learning?
- What is computer vision?
    - Why do we need computer vision?

    - How does a machine capture patterns in images?
    - Types of problems in computer vision
        - Image classification
        - Image detection
        - Image segmentation
    - Downsides/flaws in computer vision
        - Adversarial attack
            - Why do neural networks make these mistakes?

- What is natural language processing (NLP)?
    - How does a machine read and find patterns in text data?
    - What are word embeddings? What's their role?
        - How do we get word embeddings?
    - "Attention" mechanism in NLP
    - Text cleaning and preprocessing
    - Ensemble in NLP
        - Why do we need ensembles?
        - How do we do the ensemble?
- Genetic algorithms in ML
    - 6 steps of genetic algorithms
    - Real-world applications of genetic algorithms
- Generative Adversarial Networks (GANs)
    - What makes GANs so interesting and popular?
    - Preview of things that GANs have done till now
    - How GANs work?
- Recommendation
    - How do the recommendation systems work?
        - User-based recommendation
        - Item-based recommendation
    - Association rule mining

Objectives

After studying this chapter, you should be able to:

- Learn all the important and crucial sub-fields of machine learning
- Understand that all these topics are important subfields of machine learning, and that together they make the machine learning field

Machine learning and Maths

Are ML and Maths related to each other? If so, how much?

Yes, ML and Maths are very much related to each other. Now, let's see why that's the case, and what's the reason behind it.

Why is ML dependent on Maths?

All the things that happen in ML are because of Maths. You know why? You already know the answer. We read that in "*what is data*?" section for machines in the previous chapter, where data is nothing but numbers; it's all numbers at the end of the day.

The fact *machines can only understand numbers* makes machine learning closely related to Maths. And to deal with numbers, we need to deal with Maths frequently. Don't worry if you don't have a background in Maths.

I'll make it simpler for you to understand what artificial intelligence (AI) and machine learning are and how things work around it, and how you can even make money from it. Maths is the only language which machines understand. So we talk with machines using numbers.

However, this book's aim is to make it simple for everyone to understand this fascinating technology without requiring any prerequisites. Machine learning mainly revolves around linear algebra and statistics fields.

Deep learning depends on calculus; especially differentiation. Back-propagation, which is the core algorithm behind neural networks, is from differentiation. This is what makes the ML field look difficult. It's the high dependence on Maths; not many people have a background or interest in Maths. Many people find it difficult. My job is to make the machine learning and AI fields as easy as possible for them.

We will discuss them in simple English using non-technical language. Knowledge in Maths will not be required to understand them intuitively. To conclude what we have read so far, the answer is "yes", machine learning and Maths are closely related as computers can only understand numbers.

That's why we need the "Maths" language to speak with them. But don't worry, we will discuss them in purely non-technical and simple English intuitive words.

Types of machine learning

Basically, machine learning is of two types, which are as follows:

- Supervised learning
- Unsupervised learning

There are other types of ML called semi-supervised learning and self-supervised learning. They are almost derived versions of supervised and unsupervised. We have read a little about these two previously in the "*What is machine learning?*" chapter.

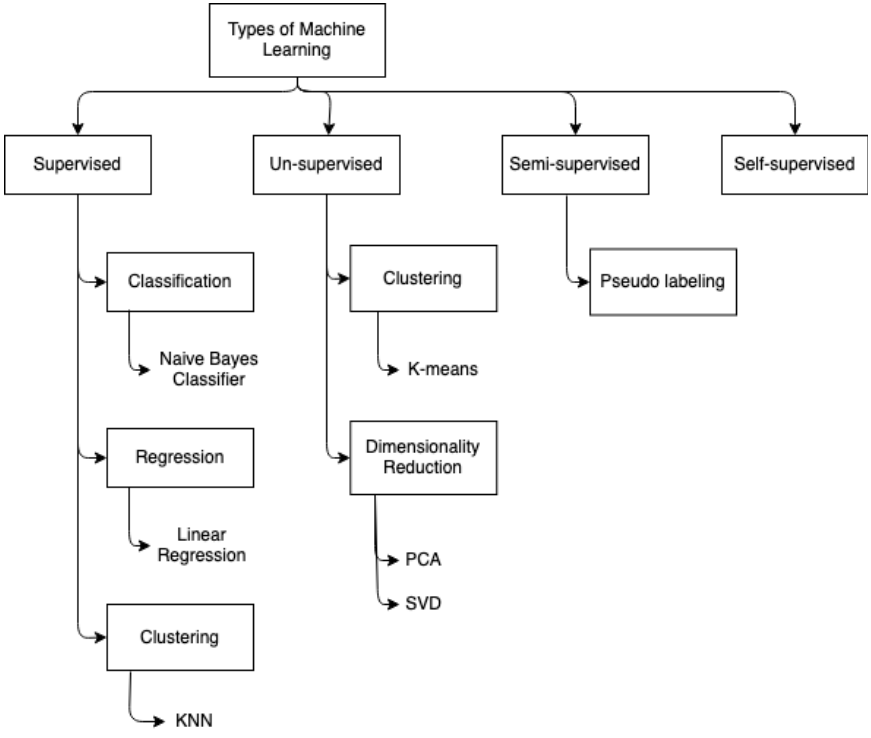


Figure 2.1: *Types of machine learning*

Most of the applications that we see in the real-world products are of a supervised type. Unsupervised learning is not that much grown as compared to supervised as of now. In a nutshell, "*supervised*" is a type of learning in which we provide supervision (that's why the term "supervised") by providing correct answers/labels.

Contrary to that, in unsupervised learning, we don't provide any label for data. Then how does the machine learn without correct answers/labels? We will see that in just a few moments. Let's discuss them in details now.

1. Supervised machine learning

In supervised learning, we provide the computers/machines with both data and their correct answers/labels. Basically, we teach machines by example. For example, while giving the following image to the computer, we also tell the computer that this image is of an "*apple*":

Figure 2.2: *Apple*

Here, "*apple*" is the label for this image. Here's how the conversation will look like between a human and a machine about supervised learning:

Conversation time

Human: Machine, look at this photo. We call this item an "*apple*". Learn this.

Machine: Okay. I will capture features/characteristics from this image; like its unique shape, its reddish color, patterns on its surface, etc., and will remember that these characteristics together are called an "*apple*".

Human: Cool. Good boy. Capture patterns/characteristics and learn their labels.

In unsupervised learning, we just give the image. We don't give the correct answer, which is "*apple*". So, the question arises that in unsupervised learning, if we don't provide labels, then how does the machine learn to predict/estimate the label in the future?

Let's see.

2. Unsupervised machine learning

In unsupervised learning, we only give machines/computer data and not their correct answers/labels. So, the question arises that in unsupervised learning, if we don't provide labels, then how does the machine learn to predict/estimate the label in the future? Well, it doesn't predict the label. Simple.

In unsupervised learning type, the machine just tells, this fruit and that fruit are different, or this sound and that sound are not the same. For example, let's take a look at the following image:

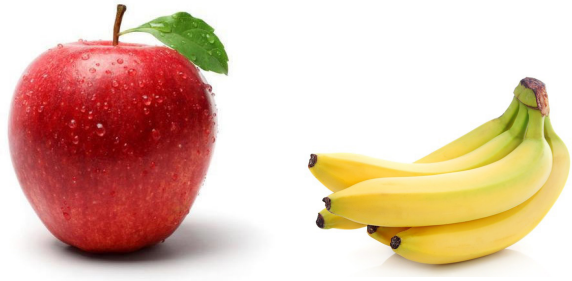


Figure 2.3: *Apple and Banana*

We just give the machines a lot of different pictures like in the preceding image, and tell them to group similar items together and dissimilar items far from each other. It's called clustering. Here's how the conversation will look like between a human and a machine about unsupervised learning:

Conversation time

Human: Machine, look at these photos. I won't tell you what they are called. You just need to group similar items. Tell me if they are similar or not.

Machine: Okay. I will capture features/characteristics from these images and I will put items with similar features/characteristics together.

…. [Machine is capturing patterns]....

Machine: Hmm.. After capturing the patterns, I think they are not similar. I will put them in different groups.

Human: Cool. Good boy.

In short, based on captured features, the machine groups similar items together and different items far from each other. This process is called "**clustering**". It is a type of unsupervised learning. It just makes different clusters with items with "*similar*" features next to each other.

For example, we want to make graphs/plots of different types for similar customers. Now, using clustering (the type of unsupervised ML), we group similar users next to each other (different users far from each other). For example, take a look at the following figure:

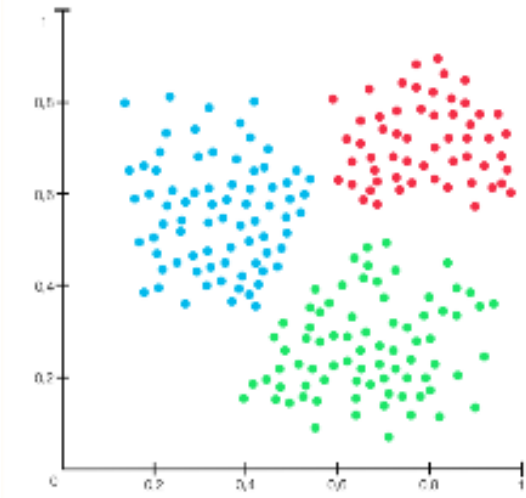


Figure 2.4: *Clustering*

We see three clusters emerging. Each dot in the preceding graph is a unique customer. One possibility is that maybe **red** represents hot customers, **green** represents warm, and **blue** represents cold customers. Hot are those who have at least bought one of your products. Warm are those who have visited your site or liked your social media pages. Cold are those who don't even know about your company.

We haven't given labels (i.e., hot, warm, cold) to them. We just know that there are three types of customers. It's our assumption that those colors stand for hot, warm, and cold. So, how does a machine group similar items together? It just captures patterns/characteristics, and then if we plot them in a graph, similar items will automatically be together.

The crux is in capturing meaningful patterns. Following which, it uses the distance between each point as a measure to define groups, like these items are together in one group and those items are together but in another group, etc. The lesser the distance, the more likely they will fall under the same group. "K-means" is one of the most famous algorithms for clustering.

Problem types in supervised learning

There are several types of problems that fall under supervised learning. However, the two main types of problems are classification and regression.

1. Classification type problems

In classification type problems, machines predict/choose one of the many classes for an item. In short, possible predictions are finite. For example, take a look at the following figure:

Figure 2.5: *Apple*

The above figure is an apple or a banana? There are two possible answers in this case, i.e., two classes; "Apple" and "Banana". However, there can be multiple classes for multiple kinds of questions. For example, let's say, in our training data, there are a total of 60,000 images. Each image contains a handwritten number between 0-9. Here the number of classes is 10. 10 unique possible answers are possible from the machine. In short, in classification, machines answer the following type of question:

Is this item A or B or C or … ?

That *A, B, C …* can be anything. Here's what a conversation will look like between a human and a machine about classification:

Conversation time

Human: Machine, look at this photo. Tell me if this photo is of an Apple, a Banana, and an Orange?

Machine: Okay. Let me first capture patterns/characteristics in the image.

…. [Machine is finding patterns] ….

Machine: Ohh, after capturing patterns, I remember these types of characteristics occur together in "apple". So, my prediction is an "apple".

Human: Cool. You are right!

We use "metrics" like accuracy to measure the performance of our machine.

What is metric?

Metrics are useful to measure and compare the performance of AI and machine learning models. Metrics tell us how much our model is correct or wrong. "Accuracy" is one popular metric for classification.

Let's say we have a total of 5000 different images and our model predicted 4500 correct outputs. In the remaining 500, it made mistakes. It predicted the wrong labels.

So, accuracy will be 4500 / 5000 = 0.9

To convert that into percentage, 0.9 * 100 = 90%

So, our model is correct 90% of the time and wrong 10% of the time. That's our model's performance. Algorithms like logistic regression and SVM also fall under classification type. Other common classification metrics are LogLoss, AUC, and F1 score.

2. Regression type problems

In regression type problems, possible answers from machines are not finite. Generally, it predicts a number that can be anything. In short, in regression, machines answer the following type of question.

How much or how many of these?

For example, the following is a face of a human, for which the machine has to predict his/her precise age:

Figure 2.6: *Person's face*

Now, a person's age can be anything: 10.23 years, 25.74 years, 50.00 years, or 120.12 years.

There are lots of possible answers. When possible answers for machines are very high or not finite, we treat that type of problem as a regression problem. MAE (mean absolute error) is one of the famous metrics for regression type problems.

What is MAE?

As discussed before, the mean absolute error metric tells the model how much it is wrong. MAE tells us how much the model's prediction was far from the actual correct answer/label.

For example, let's say the correct age of the person in the preceding photo is 19.20 years, but our model predicted 18.60 years.

So, MAE will be 19.20 - 18.60 = 0.60 years.

Our model was 0.6 years wrong from the actual label. We sum these errors across the whole data and average them out to measure the performance of our model. We also use **MAE (mean absolute error)** as a loss function in regression type problems. We will see what loss functions are, in a bit.

Here's what a conversation will look like between a human and a machine about regression:

Conversation time

Human: Machine, look at this person's photo. Tell me this person's age.

Machine: Okay. Let me first capture patterns/characteristics in the image.

…. [Machine is finding patterns] ….

Machine: Ohh, after capturing patterns, I think this person is 18.60 years old.

Human: Cool. You are somewhat closer. Actually, that person is 19.20 years old. You were 0.60 years far from the actual answer.

Machine: Ohh, I will learn this and try to minimize my errors in the future.

…. [Machine sees patterns like facial expressions, eyebrows, etc. and minimize errors.] ....

Human: Cool, good boy!

Generally (there are exceptions), in regression type problems, machines don't predict the exact answer but it will be very close to the actual answer after we have trained it. Algorithms like linear regression fall under this.

So, we have seen problem types in supervised learning. Let's now explore unsupervised learning. One well-known classification and regression algorithm is Decision Tree.

Let's explore that a bit.

Decision trees are comparatively easier to understand as it kind of relates to how we make decisions in real life. They are a kind of "hierarchy of conditions". Machines make predictions based on those conditions.

Let's understand the decision tree with an example. You want to predict that a person will buy a 1 BHK home or 2 BHK or 3 BHK or 4 BHK.

You know the following three things (or features in technical terms) about the person:

1. Number of family members

2. His/her annual salary
3. His/her marital status

From these three things, you need to predict which type of home he/she is more likely to buy. So, your brain will think of something, just like the following figure:

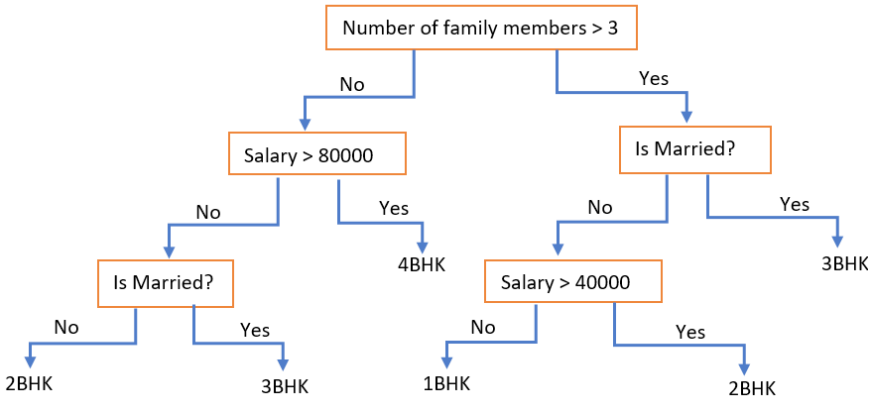


Figure 2.7: *A simple decision tree to predict what type of home a person will buy*

After going through these conditions, you will predict something, like the person is more likely to buy this type of home. That's exactly what a decision tree does.

Let's take one more small example. You want to decide if the weather is good to play Golf or not. You are given the following four things (or features in technical terms) about the weather:

1. Humidity
2. Wind
3. Temperature
4. Outlook

Based on these four things (or features), you will decide whether to play outside (target feature) or not. That's what decision trees do —

they find patterns in data. So, that they can make accurate predictions and perform better. Take a look at the following figure:

Figure 2.8: *A simple decision tree to decide whether the weather is good to play Golf*

Now, one question remains. How come a decision tree knows which conditions to check to perform better? It decides that based on something called **information gain**. Algorithms need to decide which feature should be split based on that information gain score. Constructing a decision tree is all about finding features that return the highest information gain.

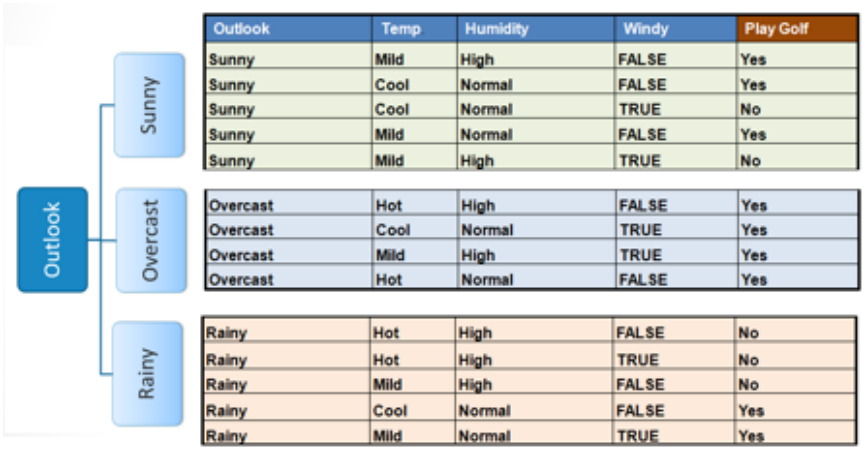
Information gain is based on the decrease in entropy after a dataset is split on a feature. Entropy is nothing but a degree of randomness. Higher the entropy, higher the chaos!

What does the splitting of data mean?

It just means dividing data into multiple parts based on one of the features.

Like, below we split our whole data by "*outlook*" feature. The same outlook values (i.e., "Sunny", "Overcast" and "Rainy") will fall in the same group. Our goal in the decision tree is to reduce the entropy of our target feature per group. In a nutshell, that's what a decision

tree is — a simple algorithm to make predictions. Take a look at the following figure as an example:



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Figure 2.9: *Data split by the "outlook" feature*

Problem types in unsupervised learning

1. Clustering

As we have seen earlier in this chapter, in clustering, we group similar items/objects together and dis-similar items far from each other.

Why do we do clustering?

We need it to get to know more information about our data. Grouping similar items together will help us understand our data better.

2. Dimensionality reduction

This is a bit technical, but, let's see. The essence of dimensionality reduction is, *to reduce the size of data while maintaining most of the "information" that the data contains.*

In other words, it is the distilled or condensed representation of our original data. It compresses data without losing important information. We will lose some information but we will maintain the more important ones.

How do we do it?

In a nutshell, we only keep those dimensions in which data capture has maximum variance. Higher the variance, more the amount of information that dimension contains.

The preceding lines may be confusing to you as it is highly dependent on mathematics, but don't worry. Our goal here is just to know that we can reduce the data size by keeping data that only captures maximum information.

In other words, we ignore redundant information in the data, as shown in the following figure:

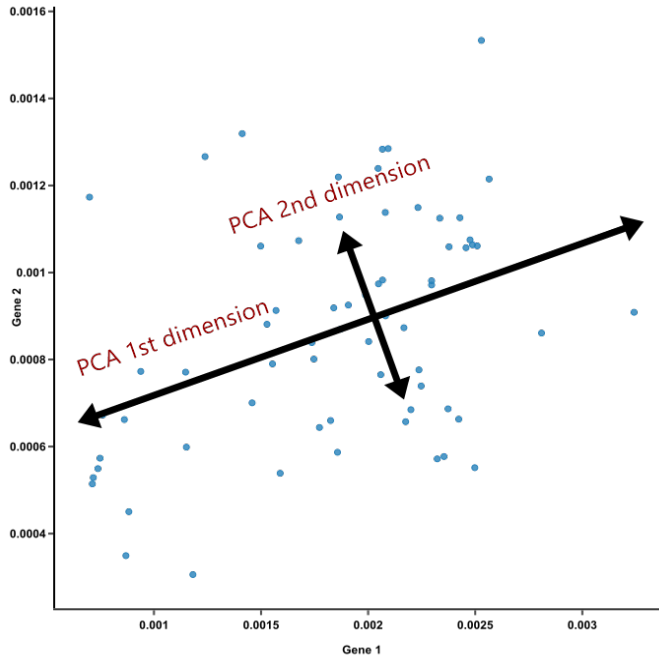


Figure 2.10: *PCA algorithm finding dimensions that capture maximum variance.*

In the preceding figure, the longer diagonal line is capturing maximum variance in the data. So, in PCA, we will choose that as our 1st dimension. And the shorter one is the 2nd dimension.

Algorithms like PCA, ICA, NMF, etc., are famous dimensionality reduction algorithms. Again, all of them are derived from mathematics.

So, that was dimensionality reduction.

Semi-supervised learning

There is a 3rd type of machine learning, which is called "**semi-supervised**" learning. As the name suggests, the approach mixes supervised and unsupervised learning. The technique relies on using a small amount of labeled data and a large amount of unlabeled data to train the systems.

Why do we need semi-supervised learning?

We need semi-supervised learning because creating labeled data is costly in terms of time and money. Supervised learning algorithms need tens of thousands or sometimes millions of data items. Creating that is not easy. There's one interesting technique in semi-supervised learning called **pseudo labeling**.

Let's say we have 10k labeled data samples and 100k non-labeled data samples. In other words, we have 10k samples of data with the correct answer/label for each of them and 100k samples for which we don't have the correct answer for any of them.

So, what we do is that we build a supervised model on those 10k examples and using that trained model, predict labels for the other 100k samples, for which we didn't have labels.

So, now we have labels for 110k samples (previously we only had for 10k samples). Next, we will train another model on the resulting mix of the labeled and pseudo-labeled data. So, generally, the performance of that model which is trained for 110k samples will be better than the previous model, which was only trained for 10k samples.

Semi-supervised algorithms also include generative adversarial networks or GANs, which we will read in detail in upcoming chapters in this book.

Self-supervised learning

This is comparatively a new type of machine learning. The essence of self-supervised learning is that labels or correct answers are somehow data itself. In other words, the data we use to train models is useful as correct answers/labels as well.

One famous example from natural language processing is language models. In language models, we try to predict the next word in a sentence by giving words up to a point. Let's say we have the

following sentence with us, “Ben is from Japan and Ben's mother tongue is Japanese". We want to use this sentence as the training data for our language model.

For example, what will be the next word for the following sentence…?

"Ben is from Japan and Ben's mother tongue is…”

Japanese right?

That's what the language models try to learn.

We can do this for any word in the sentence. So, the answer lies in the sentence itself.

We can follow this process for millions of sentences from newspapers, books, magazines, social media comments, etc. Text data is everywhere. We can use plain text data in the language model to capture patterns. We will explore this language model later.

The essence in self-supervised is that we somehow use data itself as a label/correct answer. In short, supervised and unsupervised are the two major types of machine learning.

Classification and regression are the two problem types of supervised learning, while clustering and dimensionality reduction are the two problem types of unsupervised learning.

How does a machine "learn" to perform a task?

How do computers/machines learn to do things? Is the machine even learning or is it just faking intelligence? These are the main questions. Interesting too! However, before doing that, let's understand the two important terms in machine learning, which are as follows:

1. Overfitting
2. Under-fitting

They are related to how a machine learns, and are very intuitive and interesting concepts.

1. Overfitting

Overfitting in simple terms means that machines start to remember irrelevant information instead of general patterns/characteristics which it was supposed to capture across the whole training data.

Let's take an example. Let's say we teach a machine that the following images are of the letter "A" along with other letters i.e., A-Z:

Figure 2.11: *Different images of the same letter "A".*

Now we ask the machine, what is the letter given in the following figure:

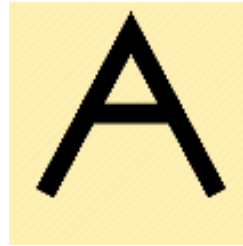


Figure 2.12: *The same letter "A" in a different font style than the previous figure.*

The model may fail to predict correctly because it hasn't seen anything like this. But if the machine would have correctly captured the underlying pattern/characteristic in the letter "A", it should have correctly predicted the preceding letter as "A".

So, in overfitting, the machine fails to capture the core underlying patterns/characteristics and remembers something else which is

irrelevant. Here's what a conversation will look like between a human and a machine about overfitting:

Conversation time

(Refer to the preceding figures of letter "A" for a better understanding of this)

Human: Machine, look at these photos. They are called the letter "A". Find the common patterns/characteristics for the letter "A" from them.

Machine: Okay. Let me try to find some common patterns in all these images of the letter "A".

….[Machine is capturing patterns like the structure of the latter and the shape which makes the character unique from others.]….

Machine: Okay. I have captured patterns. I think now I can predict the correct answer on any letter "A" images.

Human: Cool. Tell me do you recognize this photo as the letter "A"?

….[Machine is capturing patterns of the given letter "A".]….

Machine: No. I don't think this is the letter "A". I haven't seen images of this letter before.

Human: What? This is the letter "A". Your prediction is wrong. You have not captured patterns properly. You need to capture them again.

Machine: Ohh, I will try better next time.

Human: Cool. Good boy!

Overfitted ML models seem to perform very well in training data, but in validation data (data which we didn't use in training), it will not perform that well.

That's what the preceding figure tells; looking at the good training of an ML model, we can't tell whether it's a good model or it's just overfitting (or memorizing) it. So, it's not that easy to say whether our model is actually good or it's just overfitting.

For example, let's say, a student can correctly answer all the questions which he/she has seen before. But, if one asks a similar question which the student hasn't seen, he/she will fail to answer it. This may

mean, the student is just remembering (or cramming) the answers, and not actually "learning".

In technical terms, overfitting is *high variance and low bias*. Refer to the upper-right archery board in the following figure:

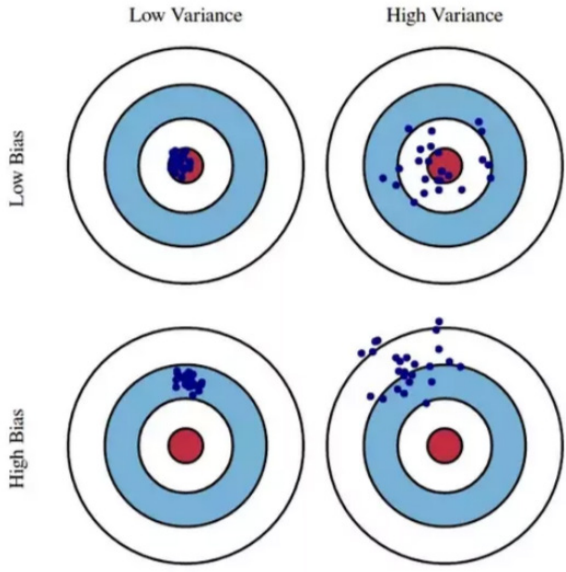


Figure 2.13: *Bias and variance by archery board example*

Bias and variance are two technical terms. The preceding figure illustrates it well. It is an archery board. The closer all of the blue points are to the red center, the better. As we can see, in low bias, low variance, it's perfect.

This phenomenon is called bias-variance tradeoff.

A very good situation for our model is low bias and low variance.

2. Under-fitting

Under-fitting is simpler to understand if compared to overfitting. In a nutshell, underfitting means that the machine does not have enough power to capture patterns/characteristics in the given data.

What is the "power" of a model? It's the model's complexity. The higher the complexity of a model, the more it is capable of capturing complicated patterns. In technical terms it's called **parameters**. Higher the number of parameters, the more complex the model. Handling

underfitting is comparatively easier, compared to overfitting. We just need to make our model more powerful to make it capable of capturing patterns.

In technical terms, underfitting is low variance and high bias. Refer to the lower-left archery board in the preceding figure for better understanding.

We just need to increase the capacity of our ML model/machine to make it capable of capturing patterns. In technical terms, people call it, increasing the model's "*complexity*" or an increasing number of "*parameters*" in the model as shown in the following figure:

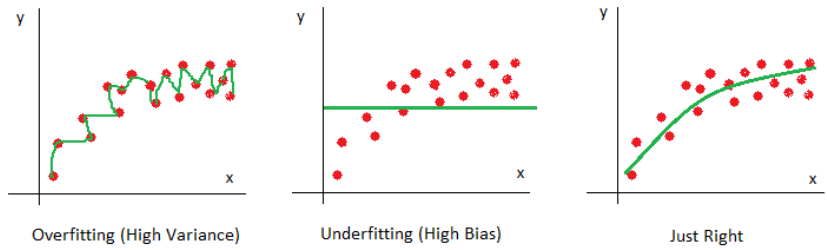


Figure 2.14: *Overfitting, underfitting. and ideal desired "just right" curves.*

In the preceding figure, let's say, we want to capture the pattern of all the red points by a curvature line. Let's see the first one. We have one green curve which is covering every red point. So, is that good? We want to cover all the red points so, it should be good right? No.

Our goal here is to capture the pattern, not to capture all the red points. And, as we can see, that green curve is not capturing the pattern; it's just too curvy!

Now, see the middle one. We have one green straight line. We will never be able to capture all the points with a straight line. The line is not powerful enough to capture this data. So, it's underfitted here.

Now, let's see the third one. It's perfect. The green curve is capturing the core pattern of all the red points. That's what we wanted. So, it's "just right". That was overfitting and underfitting. The two important terms related to how a machine is capturing patterns.

How does a machine "learn" to do things?

We have seen what machine learning and deep learning are. Now let's see how a machine actually learns from things? What makes it learn? Again, the answer is very simple. We humans learn by making mistakes!

Machines make mistakes in capturing meaningful patterns. And they learn from them.

Figure 2.15: *Machine learning in a nutshell.*

Eventually, it learns to capture meaningful patterns and remember them. While we are training our model, it will repeatedly modify itself by tweaking how it functions so that it will make lesser mistakes in the future. To make lesser mistakes, or to perform better, it needs to capture meaningful patterns in data.

How does it capture meaningful patterns? For most of the algorithms, the answer is something called the "gradient descent" algorithm. What is gradient descent? It's a mathematical optimization algorithm that tells the model how to tweak itself.

Gradient descent internally uses differentiation rules from a mathematical branch called calculus. We don't need to go much deeper into gradient descent, as long as we understand that "it tells the model how to tweak itself" to make lesser errors.

During our childhood, we have failed multiple times while learning to walk, riding a bicycle, etc. In school, we have made many mistakes while writing, reading, speaking properly, etc. We learn from those failures.

We apply the same concept to teach computers to do specific tasks. Ultimately, our goal is to replicate the human brain, right?

Let's take one example. Let's say we have 5000 images. 2500 of them are dogs; the other 2500 of them are cats. We want to teach machines to learn a simple task. "*In the following image, identify whether it's a dog's image or a cat's image*":

Figure 2.16: *Cat or Dog?*

Remember, what is machine learning? It is *capturing patterns/characteristics and remembering them for future use.* So, using those 5000 images, we train the computer to capture some patterns and learn to identify whether it's a dog or a cat. In the initial face of that training, computers will make lots of mistakes. But it learns from them.

It tries to capture some patterns that are different in dogs and cats, such as the shape of eyes, nose, mouth, legs, etc. What makes a dog different from a cat? The machine tries to capture those patterns. In other words, it finds unique characteristics to each label, i.e., cat and dog.

Once the training is over, the computer would have captured some pattern which differentiates dogs from cats and would have remembered it. Following this, we give the machine an image to identify whether it's a cat or a dog. The computer will try to match

patterns that it remembers of dogs and cats, and predict either “this a dog” or "this is a cat”.

What did I tell you?

It's all about "*capturing patterns/characteristics and remembering it for future use*."

But how does the machine capture patterns? That depends on the type of data — whether we are using image data, text data, audio data, or spreadsheets like tabular data, etc. We will read about all that in a moment.

So, how many times does it have to make mistakes to learn meaningful patterns? Tens of thousands of times or in deep learning, millions of times.

From every mistake, we give a penalty to the ML model. It is called a "loss". Bigger the mistake, higher is the loss.

What is loss and loss function?

Loss functions are mathematical formulas using which we calculate the loss. Loss tells us how far are a model's predictions from actual correct answers/labels. Here's what a conversation will look like between a human and a machine about loss function:

Conversation time

Human: Machine, look at this person's photo. Tell me about this person's age.

Machine: Okay. Let me first capture patterns/characteristics in the image.

…. [Machine finds patterns like facial expressions, eyebrows, etc. and minimizes errors.] ….

Machine: Ohh, after capturing patterns, I think this person is 18.60 years old.

…. [Human (thinking): Hmm.. the actual answer is 19.2 years, so the model is 0.6 years wrong.]….

Human: Okay. But your prediction/estimate is 0.6 years far from the actual answer.

Machine: Okay. I made a mistake. I will learn from this and will try to avoid this type of a mistake in the future!

Human: Cool, good boy.

Higher the loss, the further our model's predictions are from actual labels. So, the higher the loss, the more we will penalize our model. We need to decide on how to give this loss to the model so that it can learn faster and better. We make loss functions for it.

It tells the model it's mistaken. Okay, that was a bit technical.

The essence is, using loss, we tell AI models how much it was wrong. That's it!

In a nutshell, that's how machines learn from data. By making mistakes and knowing how much it was wrong. MAE and Cross-entropy are famous loss functions.

As the type of data changes, the model/algorithm we use to capture patterns also changes. Computers/machines only understand numbers. We have already seen how different types of data can be converted into numbers in the previous chapter. The way we capture patterns differs, as the type of data changes.

But "capturing patterns" remains essential in machine learning. Let's explore how computers/machines capture patterns in different formats.

We will now discuss three different types of data, namely, images, text, and audio.

1. Images

This is what images look like to computers:

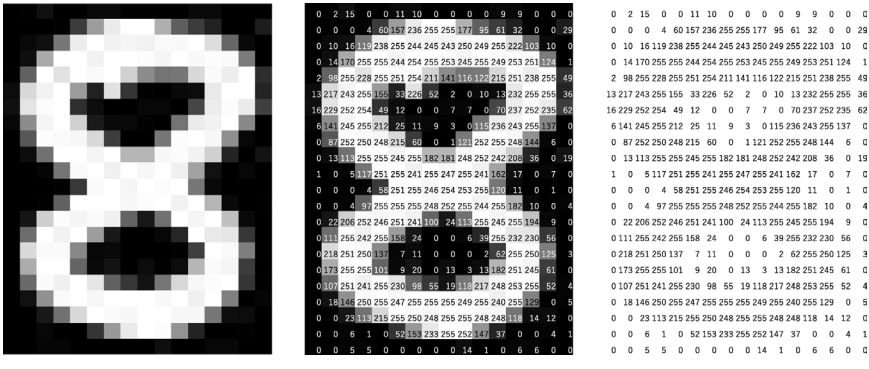


Figure 2.17: *Numbers; numbers everywhere!*

Images are made from pixels. To a computer, an image is really just a series of numbers that represent how dark each pixel is. (See the preceding image). The machine tries to find different edges, lines, and textures in images. Then it combines them to know about the full image.

Figure 2.18: *An elephant*

For example, in the preceding image of an elephant,

- First, the machine captures the edges of the elephant.
- Then, it combines the nearby edges and lines to make out the body parts of the elephant — legs, ears, trunk, etc.
- Then it combines all the body parts to form a full elephant.

After combining the body parts, the machine remembers from its training that these are the body parts that only one animal can have together — elephant.

Seeing and fetching information from images are done by a specific type of neural networks called **convolution neural networks (CNNs).** CNNs first learn to capture edges and textures in the images. Then it combines nearby edges to make small parts of images.

After combining all the small parts of images, it analyses that these types of small parts only occur in one thing together. And then, it predicts that one thing. In short, machines go from tiny details of images to large parts of images by combining them and eventually identifying what image it is. In a nutshell, that's how computers find unique patterns in images.

We will read about this in detail in the "computer vision" chapter in this book.

2. Text

We will read in detail about how machines capture patterns in text data in the **Natural Language Processing (NLP)** chapter in this book. Basically, we first give each word a unique number, and then the machine finds some linguistic features inside the text data. We will discuss more in the NLP chapter, later in this book.

3. Audio

This is a bit technical but let's see. Audio is nothing but sound waves traveling in the air. We capture something called the amplitude of sound waves every few milliseconds. So, we get a stream of amplitudes from the sound wave.

We capture amplitudes of the sound wave at every few milliseconds. In technical terms, it's called **sampling**, as seen in the following image:

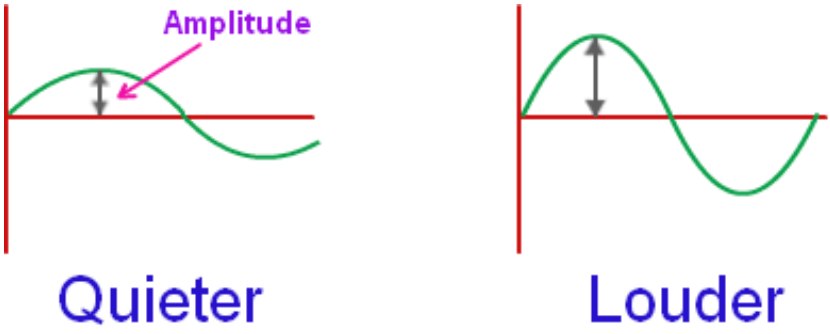


Figure 2.19: *Quiet and loud sound's amplitudes*

Every word we speak, no matter which language, has a unique stream or series of these amplitudes. Refer to the following image for a better understanding:

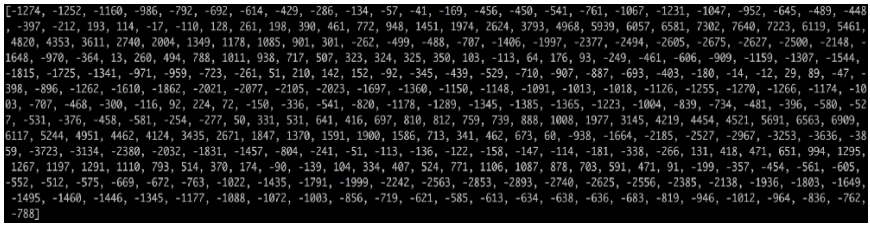


Figure 2.20: *This is what a sound wave looks like to a computer. Each number is the amplitude of the sound wave at 1/16000th of a second interval.*

The essence of voice recognition is that no matter who speaks the word, the series of numbers almost always remains more or less the same for that particular word.

For example, let's take the word "*hello*". No matter who speaks the word "hello", whether it's a male from Asia or a female from Australia, or a kid from America, amplitudes for the word "hello" for each of them will remain more or less similar.

That's the essence of voice recognition. A computer just needs to map those unique stream patterns with its corresponding word as can be seen in the following image:

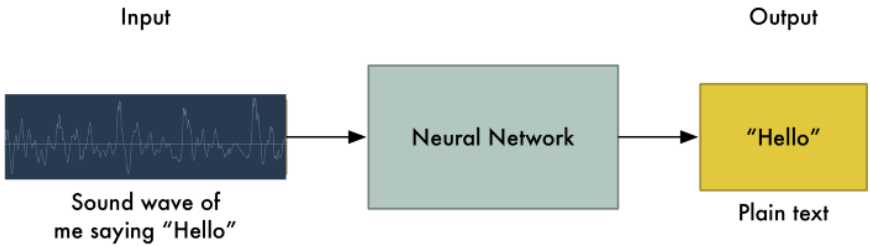


Figure 2.21: *This is how speech recognition basically works.*

To reiterate what we have learnt; how does a computer learn? By making mistakes. It learns from the mistakes it makes and tries not to repeat them in the future. It does that by capturing patterns and remembering them.

How does the neural network "learn" to perform a task?

As we have seen, in deep learning only one type of model, called **artificial neural networks**, is used.

Neural networks learn by trying to minimize loss/error. It does that by something called the back-propagation algorithm.

In other words, it learns how to minimize loss / error. So, automatically it learns to predict correct answers / labels. By loss, we tell the model that *your prediction was this much wrong compared to the actual answer.*

At first, it tries different ways to capture patterns / characteristics. It fails tens of thousands of times. Then, we give "loss" to it, which tells them that you are this much wrong compared to the actual answer.

It learns from those losses. In short, it learns by committing tens of thousands of mistakes. The network learns how to recognize each component of the numbers during the training process by gradually tweaking the importance of data, as it flows between the layers of the network.

This "*tweaking*" is done with the help of back-propagation (mathematical algorithm which runs behind training neural networks). This is possible because each link between layers have an attached weight to it, whose value can be increased or decreased to alter that link's significance. This "**back-propagation**" algorithm is based on gradient descent.

What is gradient descent?

As we have seen, it's a mathematical optimization algorithm that tells the model how to tweak itself. The goal of this algorithm is to find some combination of "weights" for which the total loss across the whole training dataset is minimum.

What are "weights" in neural networks?

Remember we read in *"what is deep learning?"*, the part that each neuron is a wire which sometimes passes incoming signals from the previous neuron to the next neuron and sometimes it doesn't? Well, "weights" are the deciding factor of whether to pass an incoming signal to the next neuron or not.

Each connection/bondage between two neurons has some weightage. Higher the weightage, more the possibility that the signal will pass through the next neuron from the current neuron. In other words, weights are nothing but the strength of the connection between any two neurons, as illustrated in the following figure:

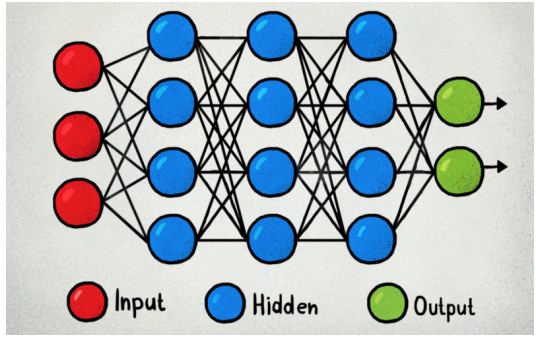


Figure 2.22: *A typical deep neural network*

Neural networks store their captured patterns/characteristics in the form of "weights". As we see in the preceding example of neural network, each neuron is connected with another neuron via a bondage/connection (by the way, each circle in the above image is a neuron). Each of those connections has some weightage. That's what a neural network learns while training; a combination of weights for which the loss will be minimum across the whole training dataset.

The "learning process" is made possible by how the network is able to alter the importance of the links/connections between the neurons in each layer. Each link has an attached value called **weight**, which will modify the value spat out by a neuron as it passes from one layer to the next. By altering the value of these weights, it is possible to emphasize or diminish the importance of links between neurons in the network.

The model learns which links between neurons are important in making successful predictions during training. *Gradient descent* is one of the most famous algorithms in finding these weights for which the loss is minimum. Again, this algorithm is highly dependent on mathematics.

This gradient descent algorithm tells the model how it should update the value of the weights attached to each link, with the ultimate aim of improving the performance of the neural network.

We don't need to explore it, as long as we understand that gradient descent is the reason why neural networks can learn to perform tasks by making a lot of mistakes in capturing patterns/characteristics. At the end of each training cycle, the system will examine whether the neural network's final output is getting closer or moving further away from what is desired.

If it's moving further away, we need to tell the model that you are moving away from what is desired. To close the gap between the actual output and desired output, the system will then work backward through the neural network, altering the weights attached to all of these links between layers.

This process is called back-propagation.

Why do we need a "deep" network or multiple layers in a network?

The short answer is, to see/capture the higher-level picture. In other words, to make information more abstract so that the model can easily capture it.

Let's take an example. We have seen that machines capture patterns in images by combining small parts to make big ones. Well, multiple layers do this "combining" part.

The first layer of a network like *Figure 2.22* just captures small things, like edges. Then the next layer will train itself with a combination of edges, like some shapes, textures, etc. The next layer will train itself to capture combinations of shapes like eyes, legs, nose, etc. The next layer will combine all the small parts and capture things like the face. Then after combining those parts, we capture the whole body.

The same is true for any type of data.

To conclude what we have seen, neural networks learn by failing to capture meaningful patterns, tens of thousands or sometimes even millions of them. Neural networks store patterns that they found in the form of "*weights*". Weights are nothing but the strength of the connection between any two neurons.

What is reinforcement learning (RL)?

Reinforcement learning is a subfield of machine learning, *where a software robot/agent takes actions according to its surrounding situation/ environment so that it can get maximum reward.*

In other words, our machine "learns" to take the right actions in certain situations.

What is the "reward" in RL?

The reward is a result that the robot software/agent gets after it took some action in a certain environment. Rewards can be positive (+ve) or negative (-ve). Positive reward means good results and negative means a bad result. A reward can be anything. It can be some physical thing or a virtual thing.

Reinforcement learning is a very general framework for learning sequential decision-making tasks. I know it's a little hard to understand without some examples. So, let's take one example.

Imagine in cold winter, a baby is standing near a bucket filled with very hot water. Baby (agent) is curious about what this thing is. So, the baby will take some actions and get some results.

Our curious baby tried to touch the water but as the water was very hot, the baby didn't like the experience of touching it (receiving a negative reward). Now, there's a cold water tap nearby and he/she adds some cold water into the hot water (baby took action).

Now the water becomes warm and good for bathing in winter. The baby touches it again and he/she feels good (receiving a positive reward). So, the baby will remember this experience next time and will not touch hot water without adding cold water to it. In other words, our curious baby will try not to get negative rewards again and will try to get a positive reward in the future. Baby remembers adding cold water into very hot water, and turning it into warm water, which feels good when we touch it/use it for the bath.

Actually, much of a human's learning is from reinforcement learning. We learn from good/bad experiences (i.e. +ve / -ve rewards). We try to avoid mistakes that have given us bad experiences or -ve rewards in the past and we try to take similar actions which have given us good results. In other words, we take actions that decrease -ve rewards and increases +ve rewards. We apply this same analogy to train software robots.

Here's how the conversation will look like between a human and a machine about reinforcement learning:

Conversation time

Human: You need to learn to walk with your two feet in this environment and I will give you some reward based on your activity.

Machine: Okay. I will try to walk according to my understanding.

…. [Machine tries to take some steps with its robotic feet and it falls.]….

Human: No. That's not how you walk. You just fell on the ground. I will give you a negative reward for this.

Machine: Ohh… I don't like negative rewards. I will try to avoid walking like this in the future so that I can avoid getting negative rewards.

…. [after some time, the machine tries to walk again. This time it takes one step successfully]....

Human: Yes. That's how you take a step with your feet. I will give you a positive reward for this action in this state.

Machine: Yeah, I like positive rewards. I will try to get more of these by taking similar actions which I took to increase positive rewards.

Human: Cool. Good boy.

The following image shows how reinforcement learning works:

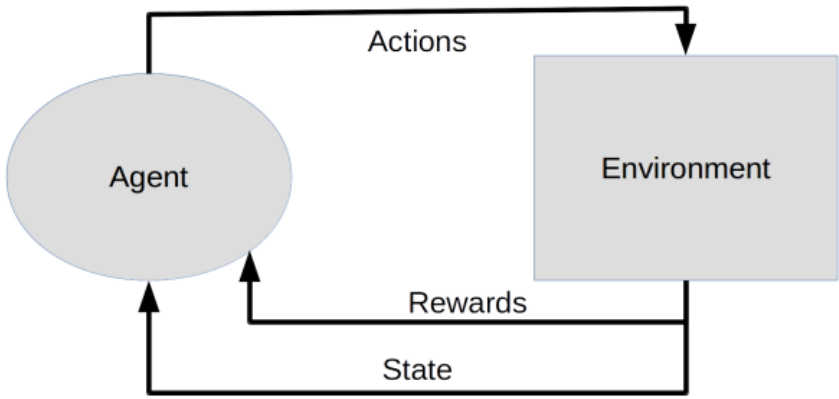


Figure 2.23: *Reinforcement learning, in a nutshell*

Agent (i.e., software robot) takes some actions on the environment surrounding it and gets rewards along with the next state. According to that reward, it learns whether the actions it took were good or not. In other words, the machine learns, *what should be the next step to maximize reward given the current state?*

Let's say, we want our robot model to learn how to walk. Now, our robot will start with no knowledge of how to move its legs so that it can walk. So, it will take some random actions with its feet. But they will not work so it will fall on the ground.

At that moment we give it a negative reward. So, the robot learns that it took this action and got a negative reward. So, it should not take this same action again in the future so that it will not receive a

negative reward. It tries again and again with different actions (trial and error). After some time, it learned how to take the first step.

At that point, we give it a positive reward. So, our robot likes it. In the future, it will take similar actions which will lead it to get more positive rewards. Ultimately it will learn how to walk. We can use this same strategy to train our robot/machine almost anything. It can learn from how to playing chess to learning how to drive using this same technique!

Real-world examples

1. Gaming

One of the real-world examples of reinforcement learning techniques is gaming bots. People train robots/machines to play certain games like DOTA or warcraft. These bots become such an expert at these games that they beat the best humans.

Recently an AI bot defeated human gamers in warcraft 99.8% of the time!

Google DeepMind's Deep Q-network has beaten humans in many vintage video games. Deep Q-network used reinforcement learning to choose actions to maximize reward. They learn from their rewards and try to take similar actions and build a long-term strategy to get a big positive reward, i.e., winning the game!

2. Chemistry reactions

Chemical reactions take time. It's almost a trial and error process for scientists to produce successful results giving reactions. Scientists applied the reinforcement learning to optimizing chemical reactions and it reduced the total time consumed and the trial-and-error work, which ultimately led to fast experiments and receiving results in lesser time.

Other than these, people have also tried to teach robots how to drive cars using reinforcement learning. Maybe it's too much to say that **reinforcement learning (RL)** can one day evolve into a state where it can surpass everything a human can do. However, RL surely has the potential to help humans in certain ways and improve a typical human's productivity.

What is transfer learning?

It is not a sub-field of machine learning but it's a technique we use to improve the performance of our model. Transfer learning is the transfer of patterns/characteristics captured from one model to another model.

Why do we do it?

It is done so that other models don't need to spend time in learning those patterns/characteristics from the start. It's actually kind of what we humans do all the time. Our minds keep learning new stuff unconsciously from our parents, teachers, mentors, friends, and other real-world experiences all the time.

The biggest benefit of transfer learning is saving time and resources. I remember one famous proverb while writing this:

Don't reinvent the wheel!

That really conveys what transfer learning in machine learning is doing. In other words, it just means that don't spend your precious time, effort, and money on something which is already being solved or done.

That's why many successful people read books. Books are a massive example of transfer learning from the author to readers. The author shares his lifelong precious lessons in one book and readers get them in days.

What they learned after committing many many mistakes throughout their life, they are giving it to others so that others don't need to spend time making those mistakes again, and learn those lessons. If we see, transfer learning is not just a machine learning concept. It's a technique that we can apply in many different and diverse fields.

So, now let's see how we use transfer learning in machine learning.

Transfer learning in deep learning

Transfer learning is mostly used in deep learning. So, let's explore that in terms of neural networks. Remember the "weights" of neural networks? That's what makes neural networks "learn" the data.

In other words, neural networks store their captured patterns/characteristics in the form of "weights".

So, everything our neural network has learned, it is stored in those weights. Weights are the crux of neural networks. Transfer learning in neural networks is transferring those weights from one model to another model. It's that simple. Let's take the following example:

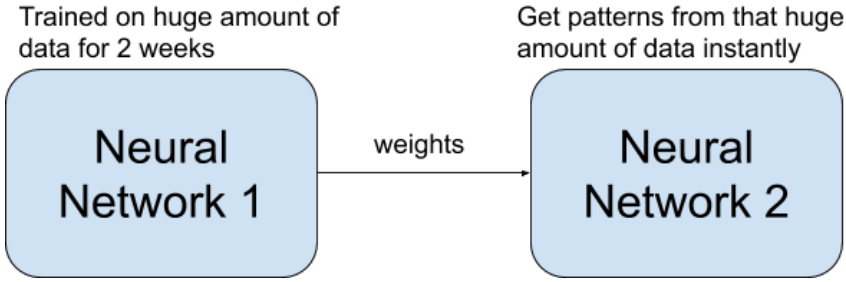


Figure 2.24: *Transfer learning in a nutshell.*

The first neural network is trained on a huge amount of data for 2 weeks 24x7. That model has now captured patterns/characteristics from the data. So, we copy-paste the 1st neural network's "*weights*" into the 2nd neural network. Now, the 2nd neural network doesn't need to have a huge amount of data and don't need to spend 2 weeks learning those patterns/characteristics.

How cool is that!

Big companies like Google, Microsoft, Facebook, etc., have huge amounts of data and huge amounts of resources to train neural networks. Almost no individual has that amount of data or resources. (By resources I mean huge capacity computers; thousands of GBs of RAM, etc.)

Now, after training these huge neural networks, they make those "weights" public. Anyone can download those weights for free!

Why do we need to download those weights?

We just saw the answer. We use "*transfer learning*" here using those weights so that our model doesn't need to have a huge amount of data, and we don't need to spend a long time training on that data.

This technique alone is the reason behind how everyone can train machines to do complex tasks at home, having a moderate level

computer with not that much data. Transfer learning increased the popularity of deep learning models (i.e., artificial neural networks).

Students and professionals are not limited to having huge amounts of data (by "huge" I mean tens or hundreds of GBs; sometimes even thousands of GBs). Transfer learning created a huge impact on machine learning by making neural networks easy enough to train, so that almost anyone can train it for their own uses.

Previously, only huge companies or organizations were able to do so, as only they had access to a huge amount of data and resources. Here's how the conversation will look like between a human and a machine about transfer learning:

Conversation time

Human: I want you to learn what the object is in an image. The object can be a plane, human face, monkey, ship, tree, chair, bag, laptop, sofa, etc. There are a total of 100 unique objects from which you need to choose one.

Machine 1: Okay. But I will need a huge amount of data to capture patterns in those diverse sets of objects.

Human: I don't have that much data. I just have ~10k images. What can I do to solve this problem?

Machine 2: I can help you.

Human: How?

Machine 2: I have captured patterns/characteristics from 1 million images of 1000 unique objects.

Human: Ohh that's a huge amount of data! So, can you please give me your weights in which you have stored whatever patterns you have learned?

Machine 2: Sure, I will.

Human: Thank you. Now I can give those weights to Machine 1, so that it will not need much data.

Machine 1: Yes. I can directly learn those patterns from Machine 2's weights. I will not need much data in that case.

…. [Human copies weights of Machine 2 to Machine 1 and gives ~10k images he/she have]....

Machine 1: I can see Machine 2 has indeed captured good patterns in the data. I can use that to improve my performance with less data!

Human: Yes. Good boy!

In short, transfer learning helps us get good performance from our ML model despite us not having much data and resources to have that level of good performance.

To conclude what we have seen, transfer learning is the transfer of patterns/characteristics from one neural network to another one, so that another one doesn't need to have huge amounts of data and doesn't need to spend time on capturing those patterns again. In short, we don't need to reinvent the wheel.

What is computer vision?

In this chapter, we will explore all the image- and video-related tasks that machines do. By the way, video is nothing but a sequence of images; so, if we understand images, we can kind of understand videos too. Computer vision is a subfield of artificial intelligence that focuses on capturing patterns/characteristics in digital images or videos to automate things that our human visual system (i.e., eyes) does.

We can apply the techniques and algorithms of this field, as long as the problem is somehow related to images or videos. So, it's all about images and videos. Computer vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images.

The goal of computer vision is "*automatic visual understanding*". Nowadays, almost all computer vision tasks are done by deep learning, i.e., artificial neural networks, particularly, a special type of neural network called **convolutional neural networks (CNN).** CNN is the reason why computers can capture patterns in images.

Some tasks/products that are powered by computer vision are:

- Facial recognition (e.g., detecting a face in mobile phone's camera)
- Cancer detection from scanning images of body parts (lung, breast, etc.)
- Self-driving vehicles

- Video surveillance (e.g., traffic departments use CCTV to monitor and control traffic and accidents)
- Visual inspection (manually checking if the quality of the product is good in factories and manufacturing plants)
- Malls use people counting to calculate how many people visited the mall on a particular day, how many people are currently inside the mall, etc.
- Monitor ships, water animals, and oil factories in the sea from satellite images

Not only that, computers can even recognize facial expressions like smiling, sadness, anger, stressed, etc. Applications of computer vision are endless.

Things that computer vision can do are as follows:

- Identifying whether the given photo is of a male or a female
- Given a photo with a human face, predicting his/her age
- Counting how many ships are there in the sea in a satellite image
- Detecting the border of each road to create a map in a satellite image
- Detecting in which part of the image, a dog is there, by making a rectangle around it

We will explore some of them by taking examples in this chapter. Why do we need computer vision? This is a very fundamental question to ask. One should know the answer to this question before learning about this topic.

Due to social media and the growing usage of smartphones, images and videos are probably the most consumed content type (in terms of the number of views and downloads) in today's world. Because of that fact, there's so much data in the form of images and videos; therefore, processing and extracting information from them becomes crucial.

There are a lot of use-cases and products which are backed by computer vision. Many tech companies are using computer vision as their product or as a feature in their product.

For example,

- Facebook is using facial recognition to automatically tag people in a post

- Tesla, Uber, and Lyft are using computer vision in their self-driving vehicle's software
- Apple is using computer vision in their face unlock feature on the iPhone
- NASA and other space organizations/companies are using this to capture information from satellite images

Companies need computer vision to provide cool features in their products. People like us need it to improve our productivity and lifestyle. Governments need it to make sure people are obeying laws by CCTV cameras across roads (especially in China).

It seems, almost everyone needs computer vision!

How does a computer capture patterns in images? Let's take a look at the following image:

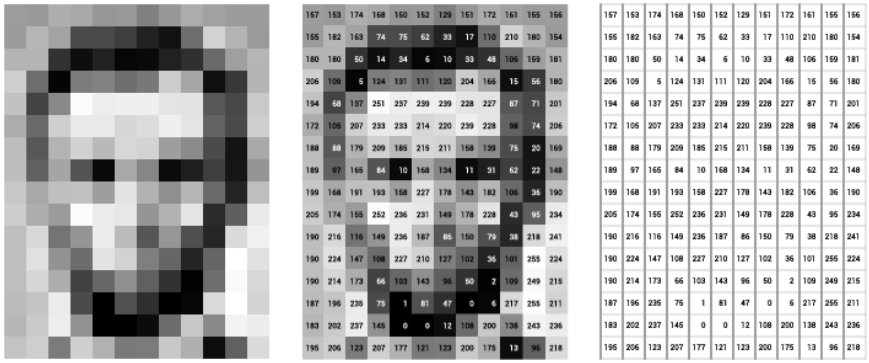


Figure 2.25: *An image to a computer*

We have seen this in brief previously using the following image of an elephant:

Figure 2.26: *An elephant*

Another example can be, handwritten digit recognition. We want to teach machines to recognize which number from 0-9 is written by a person on some paper. First, it will detect edges or boundaries. Then it will look at higher-level details, like upper rounded loop in digit "9". This carries on all the way through to the final layer, and finally it will output the probability that a given handwritten figure is a number between 0 and 9.

Let's see in detail. CNN has multiple layers. Each layer captures some patterns. 1st layer only captures edges (vertical, horizontal, diagonal, etc.), and borders as shown in the following figure:

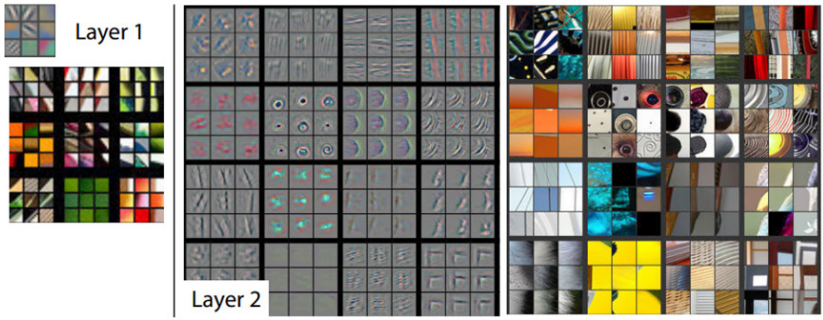


Figure 2.27: *This is what layer 1 and 2 of CNN look like.*

The 2nd layer combines patterns captured by the first layer and makes some textures. It will further pass these textures to the 3rd layer, as shown in the following figure:

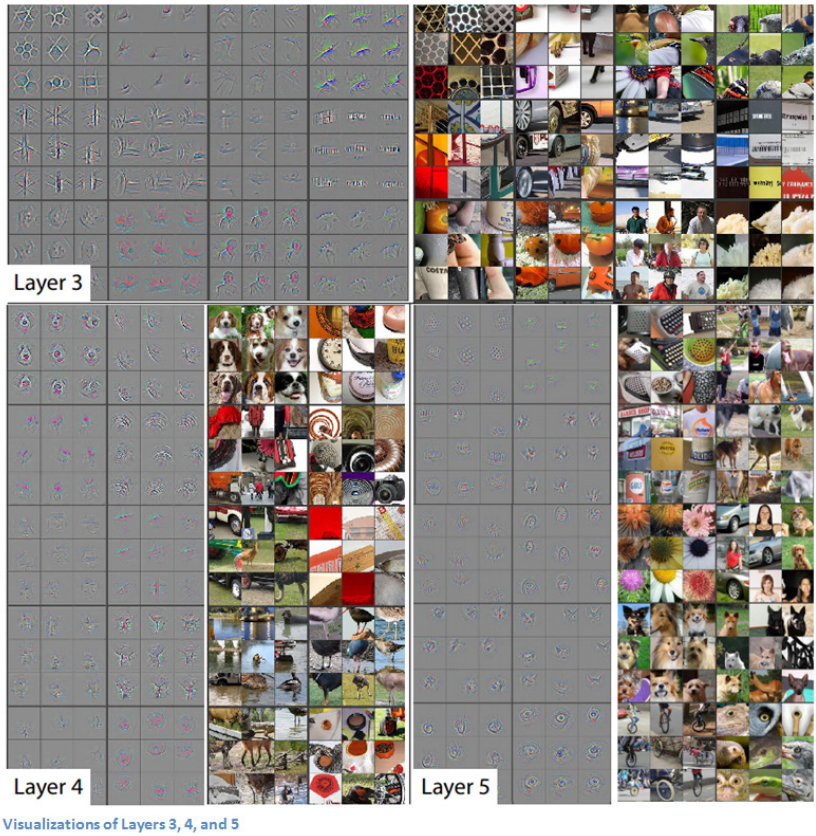


Figure 2.28: *This is what layer 3, 4, and 5 of CNN look like.*

The 3rd layer again combines little details and passes them to the 4th layer. In the 4th layer, small parts, like the nose, face, etc. start emerging. It then passes this information to the 5th layer. The 5th layer combines them and makes higher-level details, like face, flower, cycle, etc.

The preceding images are taken from this awesome blog: : **https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html**. Do check it out if you want to dive deep into mathematical details and different architectures of CNN. The following is what each layer looks like in images:

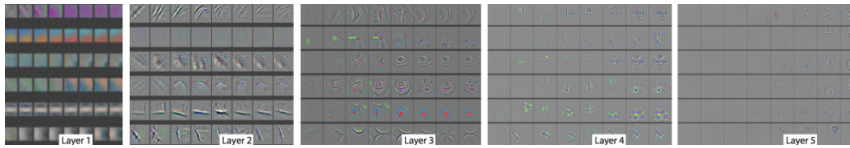


Figure 2.29: *Patterns each layer captures in CNN*

The higher we go from the first layer to the 5th or 6th layer, the more abstract and detailed patterns are captured. Reiterating how machines capture patterns in images — by combining small details of the image into big ones. It's kind of "*zoom-out*" in a digital camera:

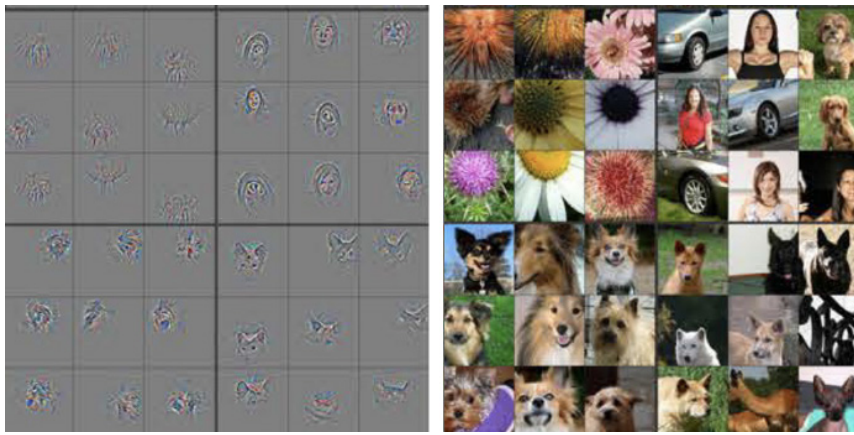


Figure 2.30: *This is what the machine sees in the final layers*

As we saw in the preceding image, this layer has learned to capture the faces of humans and animals. Now you know how a machine sees images.

There are several kinds of problem types in computer vision, which are as follows:

1. Image classification (or object classification)
2. Image detection (or object detection)
3. Image segmentation

Let's read about the different types of problems in computer vision.

1. Image classification

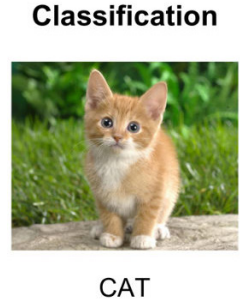


Figure 2.31: *Image classification*

In one statement if we want to define image classification, then it will be -- in the given image, tell me what kind of object do you see? Is it a dog or a cat; is it a male's face or a female's face, etc. Following is the link to the Python notebook, in which I have shown how we can build a CNN model that can classify whether an image contains a dog or a cat:

Is it a cat or dog in the image - CNN - Computer Vision

(https://github.com/bpbpublications/Demystifying-Artificial-Intelligence/blob/main/Chapter02/is-it-a-cat-or-dog-convolutional-nn.ipynb).

2. Image detection

Image detection is nothing but detecting the different kinds of objects that are there in an image and where they are located. Like we tell machines to "*draw rectangles around all the objects you see in the following image*":

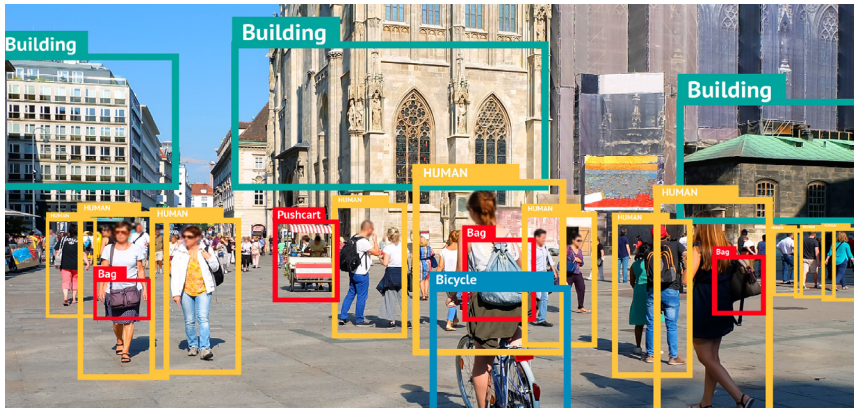


Figure 2.32: *A typical object detection output*

Self-driving vehicles will see images like the following on the road:

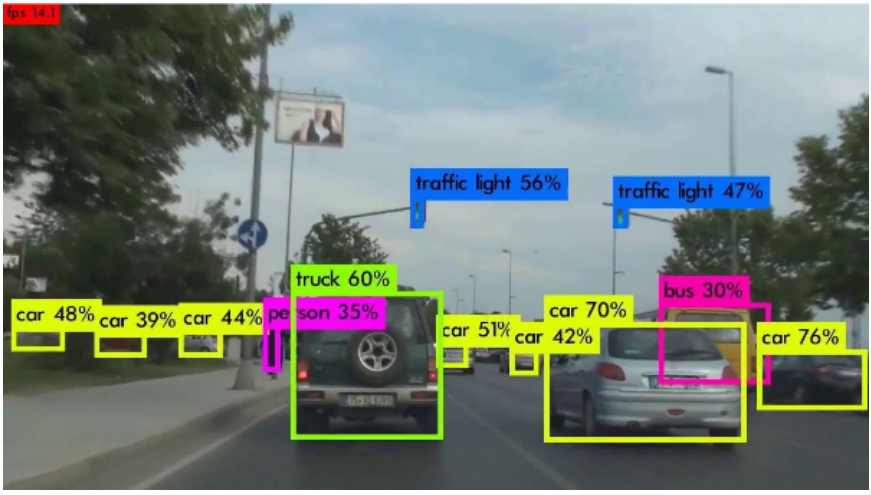


Figure 2.33: *View on a road*

Image classification is a subpart of image detection.

3. Image segmentation

In image segmentation, we classify every pixel of a given image to a certain class. For example, in the given image, some pixels belongs to the road, some belong to the car, and some belong to a person, etc. Every pixel in the given photos belong to some object. The machine has to identify what those objects are. Refer to the following images for a better understanding:

Figure 2.34: *This is something like what a self-driving vehicle will see while driving on the road.*

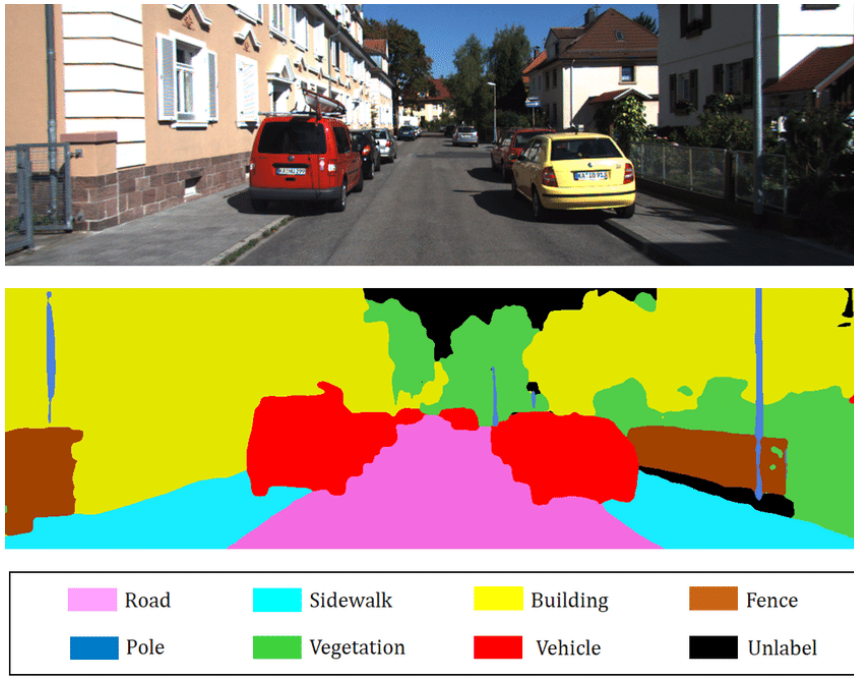


Figure 2.35: *Assigning every pixel of the image to a class.*

Basically, computer vision automates human eyes in a very cost-efficient way. It's all about automation at the end of the day. We will see what the business opportunities are for computer vision in the latter part of this book. We have seen the upsides of computer vision.

Now let's see some downsides.

Downsides/flaws of computer vision

1. Adversarial attack

This one is very crucial. Let's discuss what that thing is.

The adversarial attack is basically a way to fool neural networks to make wrong predictions. Neural networks "*learn*" to capture

patterns/characteristics in weird ways. Have a look at some of the following examples:

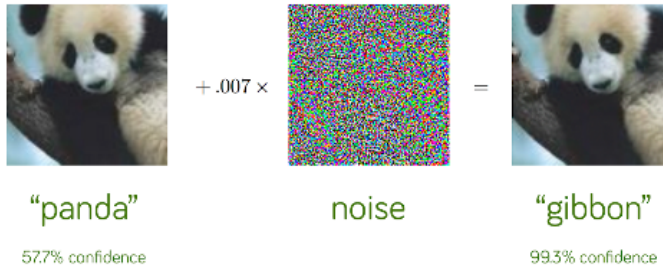


Figure 2.36: *Adding random noise to every pixel of the image fooled the model to predict "gibbon" with 99% confidence*

There are 3 images in the preceding figure. The machine predicted the 1st image as "panda" with 57.7% confidence, which is correct. Now, when we added some noise pixels (2nd image) to the 1st image, the machine predicted "gibbon" with 99.3% confidence, which is completely false.

Both 1st and 3rd images are almost the same to the human eyes. But for machines, a lot has changed. That's why they are making such wrong predictions.

Let's look at one more example; refer to the following image:

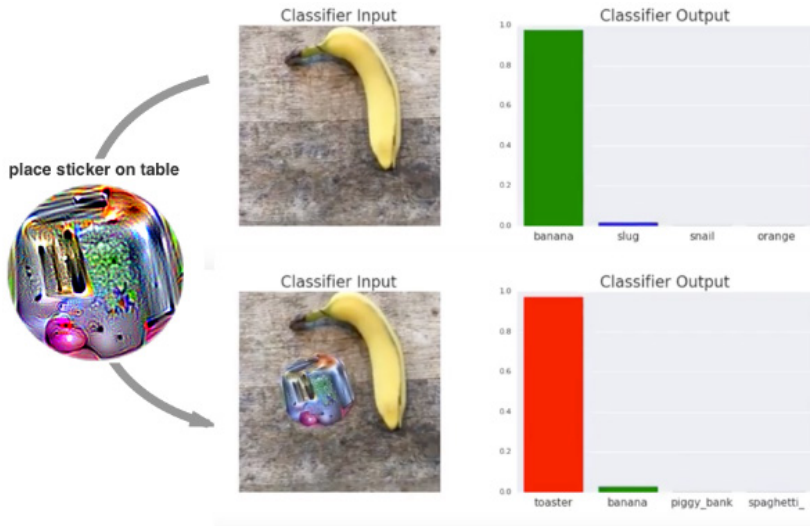


Figure 2.37: *How a plastic sticker can fool deep learning model to predict "toaster" instead of "banana"*

There's a banana on the table. The machine predicts that it's a banana with very high confidence (look at the first bar graph. Green is banana). So far so good. Now, if we put a carefully created plastic sticker beside the banana, the machine now predicts that this image contains a toaster! (See 2nd bar graph. Red is toaster.)

I mean what? How's that even possible? That plastic sticker is not even touching the banana. Neural networks indeed "learn" to capture patterns/characteristics in weird ways. This may create serious chances of causing road accidents.

Imagine a self-driving vehicle is going and there's a STOP sign (with some modifications) and the machine predicts it's a "Speed Limit: 45 mph" sign. Accidents can happen because of this.

Many companies and universities are working on these serious problems. However, we will not experience the full usage of these technologies without having solved these problems. Don't need to worry. Hundreds of brilliant researchers are working on these problems. They will figure out the solutions soon.

Figure 2.38: *Regular STOP sign and STOP sign with physical perturbation*

We can see how dumb a machine can be.

For us humans it's very easy to figure out the meaning behind images. But for machines, it's really hard.

Machines look at this world with very different eyes than humans. They will take some time to get mature like us.

Why do neural networks make these mistakes?

Neural networks are kind of a blackbox to us, as of now. In other words, if we give thousands of images to it, it will "*learn*" from them and make correct predictions most of the time. But we are not completely sure as to how it is doing it. We don't really know how a neural network makes decisions. As we don't really understand neural networks, it's easy to fool neural networks.

As of February 2020, we don't have much idea about how we can solve these problems. Few solutions are there but this isn't a fully solved problem yet. It's an open research problem for all of us.

Other than adversarial attacks, some other limitations are the requirements of data and resources. The purpose of including this adversarial attack topic was to make you aware of the current state of AI and machine learning in general.

After reading this chapter, believe me, you will have a good amount of knowledge about computer vision.

What is natural language processing (NLP)?

Natural language processing (NLP) is a subfield of machine learning which deals with capturing patterns/characteristics in text data. Again, like computer vision, deep learning algorithms (i.e., neural networks) dominate in NLP.

Whenever text data is there and we want to automate something, most probably we are looking for NLP. Companies are using deep learning algorithms of NLP in their products to increase customer engagement and to improve product quality.

Some examples of NLP in the real world are as follows:

- Virtual assistance like Apple's Siri, Amazon's Alexa, Google's Ok Google, etc., use NLP algorithms to make meaningful conversations and make sense of what we are saying.
- Chatbots also use NLP to somewhat reduce the burden on call centers and customer support teams.
- We can translate sentences from one language to another to leverage the content of one language in another language.

- Google has an “auto-complete" feature in Gmail, which predicts what we are going to type next, based on the subject of the mail and other things.
- We can somewhat guess the stock market trend of a company by analyzing tweets and people's comments on the internet about that company.
- Google Search also uses NLP to make more sense of our queries. They try to predict what exactly we want to know by a particular search.
- Companies can determine the reviews of customers about their product by extracting information from social media.
- Companies like Google and Yahoo try to detect spam emails from the text to improve our experience.

Basically, NLP tries to find patterns in text data. So, how does the machine read and find patterns in text data? Computers/machines can't understand text data. It only understands numbers.

So, first, we convert words into numbers by giving each word a unique number. Let's say we have 80,000 unique English words. We give each of them a unique number from 1 to 80,000. So, the sentence "*Welcome to Distilled AI”* becomes *”125 34 67463 34698*" or something like that. We replace each word from the sentence with its corresponding number. This process is called **encoding** in deep learning's technical language. The computer/model will see each word as its corresponding number.

Then we feed those numbers into the ML model, which observes some "linguistic features". For example,

- "am" mostly occurs after "I"
- "was" is used for events of the pasts
- "will" is used for the events in the future
- "if" and "then" mostly occur together
- "Italy" and "China" have something in common (they both are names of countries)
- Usage of the character "s" to signify the plurality of items

Machines learn this without knowing any rules of grammar or syntax. They store all these findings in something called **word embeddings**.

What are word embeddings?

It's basically a numeric form of representation of a text word. In word embeddings, each word has ~400-600 floating-point numbers that "*represent*" them. It kind of stores the "meaning" of words or "insights" of the words inside them. One famous example of word embeddings is as follows:

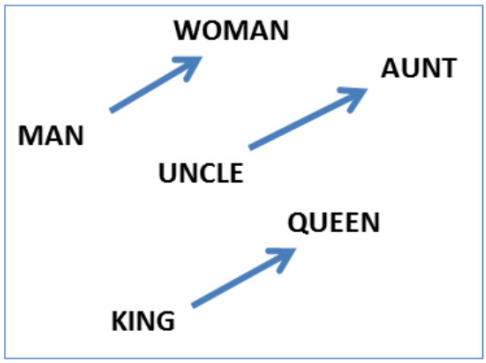


Figure 2.39: *Word embeddings store "meaning" of each word inside them*

Machine understands the way the words "man" and "woman" are correlated, words "uncle" and "aunt" are related in the same way. That's what word embeddings stores inside them.

We even do mathematical operations like add, subtract, etc., with them! Refer to the following image for a better understanding:

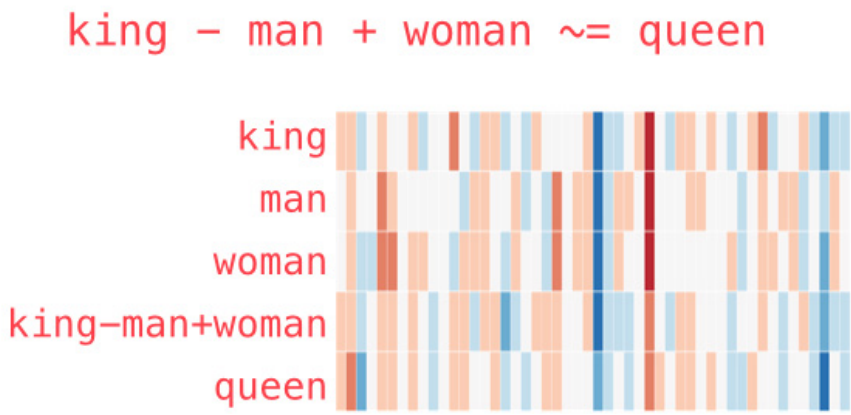


Figure 2.40: *We can even do mathematical operations with word embeddings and get expected results*

If we subtract the "man" word embedding from the “king” word embedding and add the "woman" word embedding, guess what, the resulting word embedding will be very similar to the word "queen"!

They even capture that the relationship between the words "Italy" and "Rome" is the same as "China" and "Beijing".

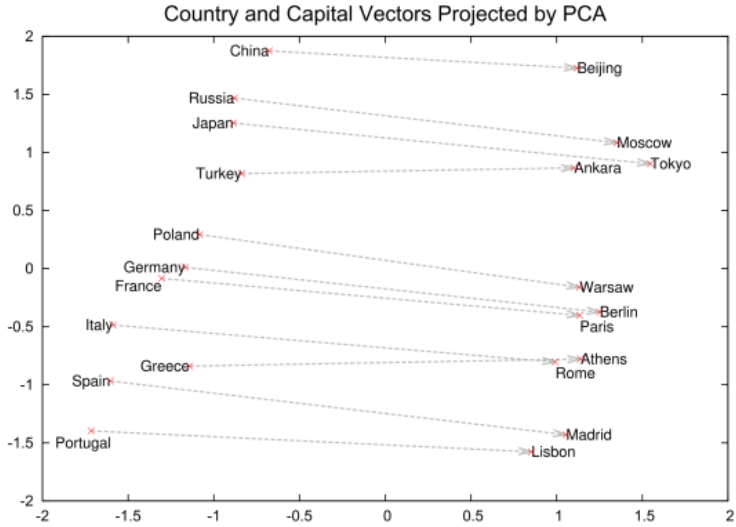


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

Figure 2.41: *Distance between each country and their respective capitals is similar in word embeddings*

This word embedding is one of the major breakthroughs in NLP's research history. Because of this, the tasks we can do in NLP has increased drastically. How did we get these word embeddings? By training something called language models. Language models basically try to predict the next word of the sentence, if provided with a partial sentence. Language models fall under a special type of machine learning called **self-supervised learning.**

Now the question arises, how come we get word embeddings by just predicting the next word correctly? To correctly predict the next word of the sentence, the model, first of all, needs to understand the meaning of the sentence, like on which words it should pay more "attention" to be less wrong or what should they remember in the text (LSTM) to be less wrong.

That's what we wanted our model to learn — abstract meaning of the sentence. So, we get these word embeddings by helping machines learn to predict the next word of millions of sentences. Models that learn to predict the next word of sentences are called language models in technical terms.

In NLP, there are several types of language models, such as RNN (LSTM and GRU), Attention, etc. We will discuss Attention here as it performs better and is more intuitive to learn.

Let's discuss one of the most exciting ideas in NLP, known as "*Attention*".

Attention mechanism in NLP

Attention basically answers the following question:

On which word should our ML model pay maximum attention/focus to be less wrong at the task it is doing?

In other words, to understand a statement, what are those words, to which the model should pay more attention? You may wonder how this "*attention*" is helping machines to understand the meaning of a sentence.

Basically, a sentence conveys some information. When humans read a statement, to get the information from the sentence, we also need to figure out important words and meanings from them. For example,

- **Question:** "Ben is from France. Abby is from the USA. Then what's the mother tongue of Ben?"
- To answer the above question, we need to pay more attention to the word "France". Then we'll be able to correctly predict the answer as "French".
- The sentence "Abby is from the USA." is irrelevant here. It does not require our attention to answer this question.

We are learning all this subconsciously. We just learn this naturally with no extra effort. Machines gradually learn things, just like we have seen previously via this "attention" mechanism.

Let's take one more example.

Consider the sentence, "The animal didn't cross the street because it was too tired". Now, what does the word "it" refer to in the previous statement? "The animal", right?

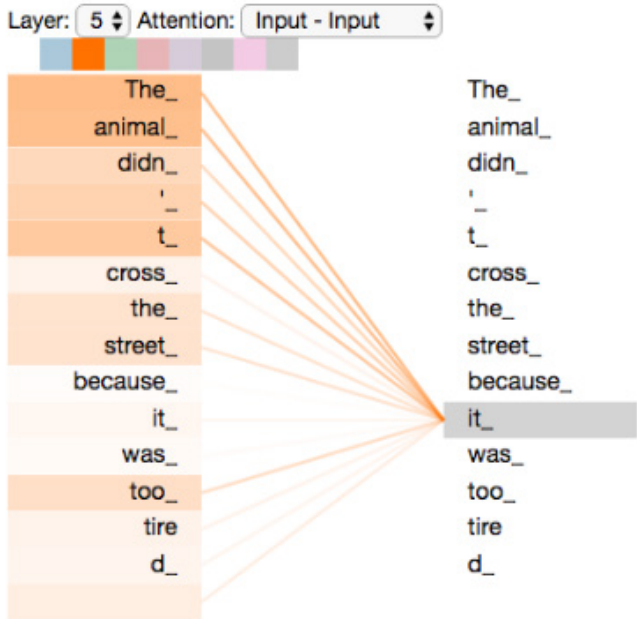


Figure 2.42: *Attention of each word while the machine is looking at the word "it"*

Easy enough. So, when the word "it" comes, the model learns to pay more attention to "The animal" words. (See the orange background of words in the preceding image. The darker the orange background, the higher the attention).

Our Attention-based ML model has learned this by seeing millions of English sentences. By the way, the preceding image was taken from this awesome blog: **http://jalammar.github.io/illustrated-transformer/**. Have a look at it if you want to know more about the mathematical side of Attention.

In a nutshell, that's how a machine captures patterns in text data. Let's go through some common things we generally do in NLP projects and problems.

1. Text cleaning and preprocessing

The essence of this part is to make our text data good enough to use it as training data, so that our ML model can capture them inside it.

In the text cleaning part, we clean all the unnecessary data in the text, like HTML tags, extra spaces, etc.

In the preprocessing part, we do things like correcting spelling mistakes, removing extra emojis, correcting word abbreviations. For example, take a look at the following sentence:

"can u plea.se tel me your name?  "

Becomes,

"Can you please tell me your name?"

Preprocessing also includes things like:

- Tokenization
  - Separating each word from a sentence.
  - Sentence "hello, how are you?" becomes "hello", ",", "how", "are", "you", "?"
- Contraction mapping
  - Replacing short forms of words with full long terms.
  - Like, "can't" becomes "cannot", "ain't" becomes "am not", "aren't" becomes "are not" etc.
- Stemming
  - Reducing words to root form to make it easier for ML models to find patterns.
  - Like, words "laughing", "laughs", and "laughed" becomes "laugh".
- Removing common words (they call it "stopwords")
  - Words like "a", "are", "the", "is" etc., don't add much information to the sentence compared to other words.
  - So, sometimes, we remove these words to allow our ML model to only focus on more important words.

The purpose of doing all the above things is to give more relevant information to our ML model to allow it to capture patterns/characteristics better and faster. The higher the quality and quantity of our data, the better the model will perform in finding patterns.

After the preprocessing data step, we train ML models to find patterns in it.

2. Building ML models

There are several models to choose from, but the performance of deep learning models is generally better than traditional machine learning models like logistic regression, SVM, etc. As of now (February 2020), deep learning models like BERT and GPT-2 have dominated NLP. They are huge models (hundreds of millions of parameters), trained on tens of gigabytes of text data by big companies like Google and OpenAI.

They have made those publicly available for everyone to use in transfer learning. We have already seen what transfer learning is. It's basically transferring the captured patterns from one model to another model so that the other model doesn't need to spend time in capturing those patterns again.

We have already seen the "Attention mechanism" in NLP section. That's what these models (BERT, OpenAI) rely on. Sometimes we can even combine two or more ML models to make better and more robust predictions. It's called "**Ensemble**".

What is ensemble?

It's a technique to improve the predictive performance by combining multiple ML model's predictions and achieve better performance than what could be obtained from any ML model alone. The ensemble is not limited to the NLP field. It is a generic concept used for any two or more machine learning or deep learning models.

Generally, we build one model to capture patterns/characteristics, but to improve performance, we train multiple models to do the same task. All of them will capture patterns in slightly different ways, but their end goal is the same. Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model.

So, ensembles may be thought of as a way to compensate for poor learning algorithms by performing a lot of extra computation.

Why do we need ensemble?

We need ensemble to reduce overfitting of single models. When we combine many of them to make predictions, as they have captured patterns in slightly different ways, the combined output of the majority will be right.

So, if one model has overfitted on the data, it will have very little effect on our final predictions. Nowadays, most of the deep learning models (both in NLP and computer vision) are using transfer learning. Without using that technique, it's very hard to achieve good performance. So, that was NLP — a wide field with tons of applications. Its importance in the future is only going to increase. And make no mistake, there's a lot more in NLP than we have seen.

We can write a whole new book, especially on NLP. The purpose of this chapter was to give you a basic idea about the NLP field in a very simple way. Things we have seen in this chapter cover almost all the main things on NLP. Again, if you have read and understood this chapter carefully, believe me, you know quite a lot about NLP.

Congratulations! Here's the link to the Python's notebook, in which I have shown how we can build an NLP model from scratch to detect Sarcasm.

Are you being sarcastic? - Sarcasm detection - NLP

(**https://github.com/bpbpublications/Demystifying-Artificial-Intelligence/blob/main/Chapter02/are-you-being-sarcastic-sarcasm-detection-nlp.ipynb**)

Check it out if you are curious about the coding part of NLP.

Now, let's see what are genetic algorithms.

Genetic algorithms in ML

The core idea of genetic algorithms lies in the roots of biology and humans' natural evolution. They are also sometimes referred to as evolutionary algorithms. If we consider humans' evolution from Darwin's perspective of natural evolution, it can be summarized as,

Those who are the fittest will survive.

In other words, those who are not capable of facing and solving challenges will not survive in the long term.

Figure 2.43: *Human DNAs or genes*

The process of natural selection starts with the selection of the fittest individuals from a population. They produce offspring (i.e., children,) who inherit the characteristics of the parents and will be added to the next generation.

If the parents are fit, the offspring (we can call it a child) has a higher chance to be better than its parents in terms of fitness. So, it will have a higher chance of survival in the long term. This process keeps on iterating and in the end, a generation with the fittest individuals will be found.

There are 6 steps that the genetic algorithms follow:

1. Population
2. Fitness calculation
3. Parent selection
4. Crossover
5. Mutation
6. Offspring

Let's look at them in a summarized way.

1. Population

The process begins with a group of individuals, which is called population. Each individual is a solution to the problem you want to solve. In other words, they are a random set of candidates, some of whom will survive the difficulties, while others will fail. We are more interested in the ones who will survive.

We do the crossover process from their genes.

2. Fitness calculation

For each individual in the population, we calculate their fitness. That fitness score represents how fit an individual is. Higher the fitness score, higher the chances for them to get selected for reproduction.

3. Parent selection

The idea of the selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two parents are selected based on their fitness scores.

4. Crossover

This is the most critical part of the process. For each parent (A1 and A2) to be mated, crossover points are randomly chosen.

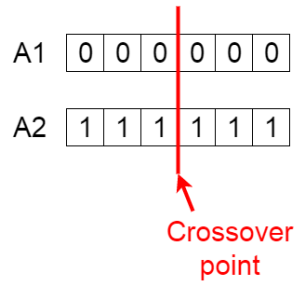


Figure 2.44: *Crossover*

Off springs are created by exchanging (or swapping) the genes of parents.

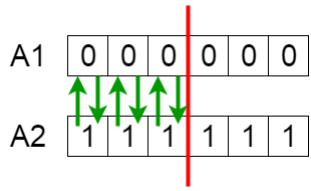


Figure 2.45: *Exchanging genes*

Resulting offsprings (A5 and A6) are added to the population.

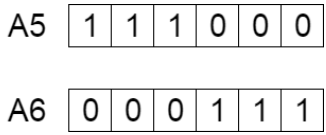


Figure 2.46: *Resulting offsprings*

5. Mutation

For some off springs or children, we change their genes randomly.

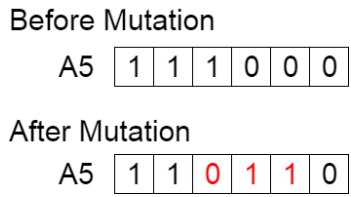


Figure 2.47: *Random mutation*

We carry out mutation to maintain diversity within the population and prevent premature convergence.

6. Offspring

Finally, after crossover and mutation, we get an offspring or child. We add this child to the population and sometimes remove the least fit individual to control the size of the population.

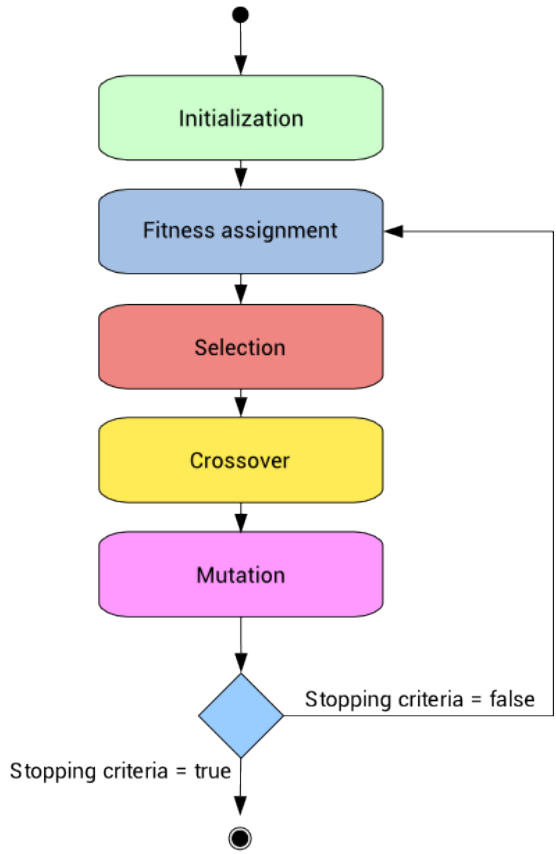


Figure 2.48: *A typical genetic algorithm*

We keep repeating steps 3 to 6 (i.e., parent selection to offspring) over and over again until we get individuals with higher fitness. This is called a **genetic algorithm.**

Real-world applications

1. Medical and health care

Human DNAs are very complex. Genetic algorithms have been used to determine the structure of DNA using spectrometric data about the sample. Other than that, it can also be used in inventing new medicines and drugs. Discovering new drugs is a time-consuming process. It's almost a trial-and-error process.

Genetic algorithms can speed up the whole process and help doctors and researchers to invent medicines faster!

2. Vehicle routing problems

Finding the best optimal path to travel from one place to another is a critical problem. Because of free services like Google Maps, our task becomes easier. Genetic algorithms can also solve this. The problem consists in designing the optimal set of routes for a fleet of vehicles in order to serve a given set of customers.

Other than the aforementioned two usage cases, genetic algorithms can help us in finance, image processing, etc. Almost any problem which includes some kind of trial-and-error process can be at least partially solved by genetic algorithms in some way.

To conclude what we have read, genetic algorithms are inspired by biological/natural evolution. The core idea behind them is, "*survival of the fittest*". We choose two fittest parents and do the crossover of their genes. The resulting generation will have better fitness than their parents.

Generative adversarial networks (GANs)

This is one of the most interesting types of networks according to a recent research done in deep learning. I know, the name is a bit complicated. But they are very easy to understand. One of the godfathers of deep learning, *Yann LeCun* (father of computer vision), described GANs as,

The most interesting idea in the last 10 years in ***machine learning****.*

So what makes them so interesting — their ability to generate/create new stuff from scratch. They kind of behave like an artist.

Normal neural networks can predict what's there in an image, can detect where they are located in an image, etc. But GANs can generate new things; not only that, GANs can generate almost anything.

They can even generate videos of famous celebrities that are 100% not real but they look very real. One famous example is *deepfake*. This is what makes GANs look like MAGIC! (Obviously, it's not magic. It's all Maths after all.)

First of all, let's see what they can do. Generate images of humans that never existed. Take a look at the following images for a better understanding:

Figure 2.49: *Fake generated images of Human faces which looks very real*

The preceding images of human faces are not real. They are generated by GANs. Have a look at this website: **https://openai.com/blog/glow/** to play with merging of two faces!

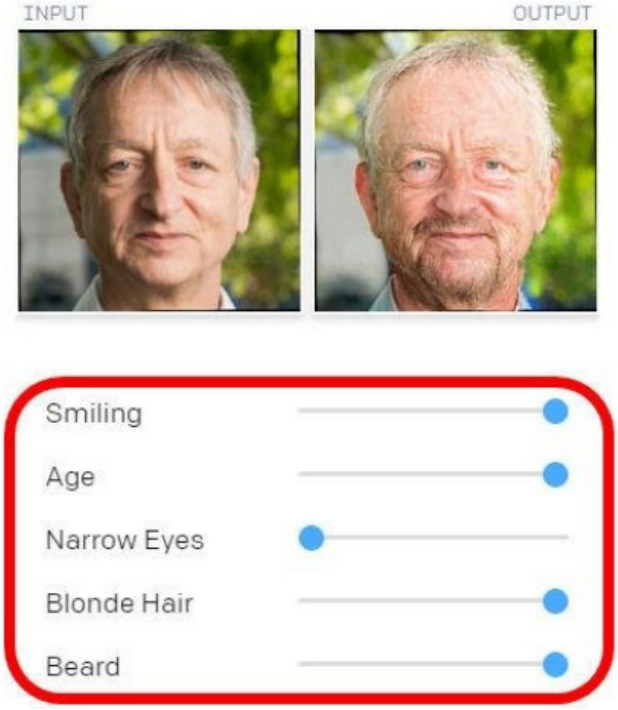


Figure 2.50: *We can increase/decrease smiling, age, beard, etc. features into the image using GANs*

We can even increase/decrease smiling, age, beard, etc! By the way, the preceding photo is of the godfather of deep learning.

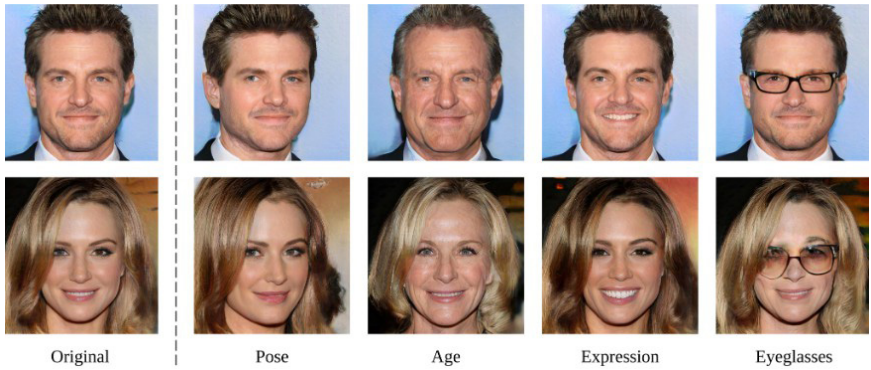


Figure 2.51: *Given original faces, GANs generated different style faces*

And make no mistake, the resulting images were not seen by our neural networks before they began working on them. They just captured patterns/characteristics of how a person looks in old age, or in eyeglasses, etc. Then you give it your face's image, and the machine predicts how you will look like in old age!

How cool is that?

They can even combine two or more faces!

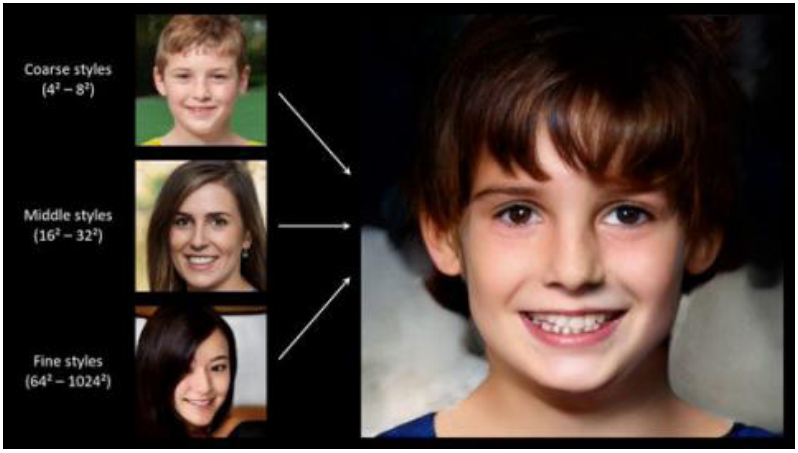


Figure 2.52: *Combining multiple human faces*

In the preceding image, the right-hand side image of the face is created by combining parts from multiple left-hand side faces! Okay, what about creating paintings? Using these images, painters and artists can get new insights or possible ways to create new paintings that they may never have thought of. This is all because of GANs. Take a look at the following images for a better understanding:

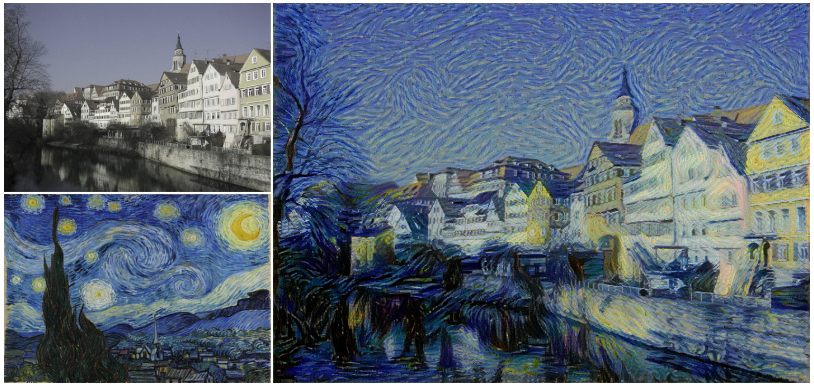


Figure 2.53: *The preceding right-hand, side image is made from a creative combination of two images on the left*

Now you have an idea of what GANs can do.

Now let's see how they work.

How do GANs work?

GANs basically play a game between two neural networks:

- Generators
- Discriminators

The goal of the generator is to fool the discriminator by generating fake content. The goal of the discriminator is to detect fake generated data from the real one. Both try to compete with each other and eventually both get better at their tasks.

The generator will get better at generating fake but realistic images. The discriminator will get better at detecting fake images so that, the generator can get better. By pitting the two networks against each other during training, both can achieve better performance.

After some time, the generator will be so powerful, that it starts to create/generate fake data which looks very real. Then we use the generator to create fake images. In this architecture, two neural networks battle; the generator network tries to create convincing "fake" data and the discriminator attempts to tell the difference between fake and real data.

Let's understand this by taking examples of images. Generators generate new images (or any content) and discriminators try to guess whether this image from the generator is real or fake. Basically, generators try to fool discriminators by showing fake generated images that look very real. Discriminators also try to catch generators whenever they show fake generated images. Both get better at their job over time.

Discriminators get better at predicting whether it is a real image or a fake image and generators get better at generating realistic images.

The following diagram illustrates what we just discussed:

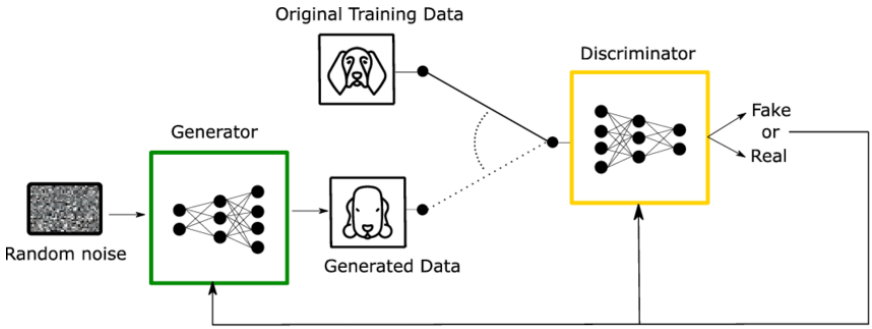


Figure 2.54: *Generator trying to fool Discriminator by showing it generated fake data*

The generator produces images (which are 100% fake). We also have real data. So, sometimes we feed fake generated data to discriminators or sometimes we feed original data.

If discriminators correctly identify an image as fake, we tell the generator that *"discriminator is catching your fake images. Generate images as real as possible so that you can fool discriminators"*.

Here's how a conversation will look like between a human and generator-discriminator about GANs:

Conversation time

Human: Hey generator, generate images that look as real as possible.

Generator: Okay. I will see real images and capture image patterns and will create fake images that look real.

Human: Hey discriminator, I will sometimes give you real images and sometimes fake images generated from the generator. Your job is to predict whether the image I give you is real or fake.

Discriminator: Okay. I will capture patterns in images and will try to predict real or fake images as accurately as possible.

….[Generator generates 100 fake images.]….

….[The human takes those images and mixes them with 100 real images and gives those 200 images to the discriminator to predict which is real and which is not.]….

….[Discriminator makes predictions and it's 70% accurate.]….

Human: Generator, you need to generate more realistic images. The discriminator is able to predict your images as fake. You need to generate images that look real so that, the discriminator can't tell which is real and which is fake.

Generator: Okay. I will try better next time while creating new fake images that look as real as possible.

Human: Okay. Good boy.

So, that was GANs, a marvelous innovation in the ML field.

Let's now read about recommendation. How recommendation algorithms work and how they provide better recommendations to us.

Recommendation

Recommender systems are an important class of machine learning algorithms that offer "relevant" suggestions of items to users. Look at the following image for a better understanding:

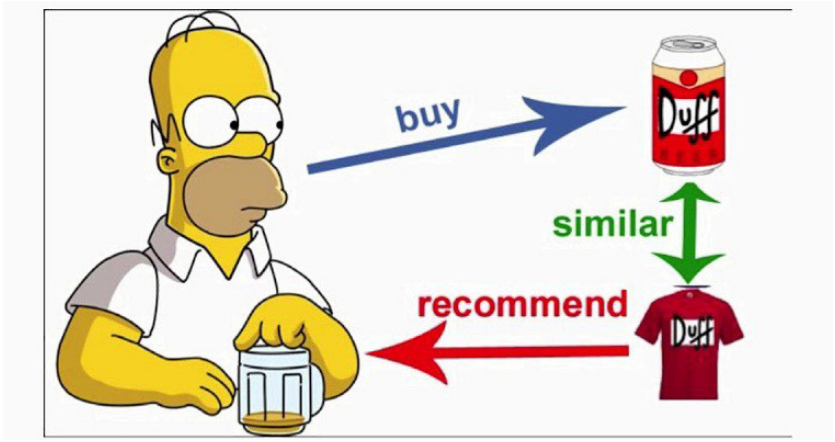


Figure 2.55: *Recommendation in a nutshell*

It is one of the most used ML fields in the day-to-day products that we consume:

- Netflix uses it to recommend TV shows and movies.
- Spotify gives personalized recommendations on songs and playlists.

- Amazon recommends items to buy.
- Facebook recommends people to send friend requests in their "people you may know" feature.
- Food chains, like McDonalds and KFC, recommend other items when we buy some food.

Companies increase their profits by a huge amount, just by using good recommendation algorithms. If we see, recommendation algorithms essentially capture human behavior. They find patterns in people's buying habits.

E-commerce companies cross-sell and up-sell items using these algorithms. We can also use deep learning to capture patterns in people's buying habits. Deep learning is mostly used for more personalized recommendations using the user's history of interactions with the product.

How do recommendations work?

Recommendation algorithms are comparatively less complex than deep learning and computer vision algorithms. In recommender systems, we use simple heuristics to suggest items. For example,

- **User-based:**
    - Let's say we want to show recommendations to user A.
    - In this method, we try to find a similar user B who also tends to like items that user A likes.
    - So, we recommend user B's other liked items to user A.
    - The logic behind this is that similar people may like similar items.
    - Here "similar people" means, people who tend to like similar items like songs, movies, etc.

- **Item-based:**
    - Let's say one user buys item P.
    - Now, from all the user's data, there's one item S which users bought almost all the time whenever item P gets bought.

- So, we recommend item S to users whenever they buy item P.
- The logic behind this is, similar items may be sold together.

The preceding 2 recommendation methods are very basic and simple ones. They fall under collaborative filtering algorithms in the recommendation field. There are other deep learning-based methods too. But they show how simple recommendation algorithms work.

More advanced (deep learning-based) algorithms also take the history of interactions of users with items. For example, on YouTube, liked/disliked videos, saved to watch later videos, etc. Those personalized actions of each user help the algorithm to more accurately recommend items to each user.

Items that we see next to each other in shopping malls are also carefully chosen based on items that are frequently bought together. Some sample real-life patterns which we can capture are as follows:

- People who visit this website are more likely to visit this other website as well.
- People who belong to the age-group [30, 40] and income [>$100k] are more likely to own a home.

Amazon uses it in their famous "*frequently bought together*" feature. This is called "association rule mining" in technical language.

Here's the link to the Python notebook in which I have shown how we can build a simple recommender system to recommend movies based on the user's past data:

Hey Machine, can you recommend a movie? - Recommendation

(**https://github.com/bpbpublications/Demystifying-Artificial-Intelligence/blob/main/Chapter02/can-you-recommend-me-a-movie-recommendation.ipynb**)

To conclude what we have seen, recommendation is used to suggest "relevant" items to people to enrich customer experience.

Conclusion

So, in this chapter, we understood the basics of computer vision, NLP, reinforcement learning, transfer learning, genetic algorithms, generative adversarial networks, recommendation, etc. All of these topics are important subfields of machine learning; all of these together make the machine learning field. Now you have a good idea about how things work in the machine learning field. Congratulations!

In the next chapter, you will be learning about the business perspective to AI.

Questions

1. How do machines recommend items to users?
2. What is 'reward' in reinforcement learning?
3. What is the major difference between supervised and unsupervised learning?
4. Why do we need transfer learning?
5. What is a pixel in a photo?

Chapter 3
Business Perspective of AI

In this chapter, we will read about the business side of artificial intelligence (AI) — how companies/organizations are monetizing AI/ML to earn profits, and how you can use AI to increase your revenues and cut costs.

After reading this, you will have a decent idea about how AI can generate cash flows and how you can leverage AI in your existing products/services.

Structure

In this chapter, we will cover the following topics:

- How do AI/ML (machine learning) projects work in the real world?
    - Best practices for predictive models
    - What does a typical ML project pipeline look like?
    - What is feature engineering?

- How to monetize AI?
  - Traditional non-technical businesses
    - Limitations of traditional non-technical businesses
    - How computers and AI can remove those limitations
  - Benefits of automation
  - 3 main things AI can do to generate profits
    - Create new products (to increase revenues)
      - ⌛ Real-world examples
    - Improve existing ones (to increase revenues)
      - ⌛ Real-world examples
    - Automate processes (to decrease costs)
      - ⌛ Real-world examples
  - After growing, market strategy to maintain growth.
    - Real-world examples
- How can you leverage AI in your business?
  - What are the typical things you will need if you want to leverage AI technology in your business?
    - High quantity and quality data
    - Best talents who know AI
    - Domain expertise
- How to use AI in your existing products/services?
  - Things you can do in your existing products that can increase the profits.
    - Automating the most time and cost consuming parts of the process.
    - Increasing sales or user engagement with the product using the data we already have.
    - Adding new AI-backed features in our existing products.
- Real-world cases of AI
  - Netflix recommendation
  - Self-driving vehicles

        - Advantages and limitations of machine learning
    - Advantages
        - Automation
        - Speed
        - Performance
    - Limitations
        - One machine can't do it all
        - Lake of explainability
        - It's not that hard to fool the machine

Objectives

After studying this chapter, you shall be able to:

- Understand how AI projects work in companies and how they increase their profits using AI.
- Leverage AI in your business and use it in your existing products/services.
- Understand how non-technical businesses use AI.
- Understand how traditional non-technical businesses can be improved with AI technology.

How do AI/ML projects work in the real world?

We have seen what AI/ML means in "how machines learn" part of the previous chapter.

Now let's see how professionals complete real-world AI and ML projects or how those projects work in the real world. Generally, a set of steps are followed. These steps are tentative. Some people may merge some of those steps together. As AI and machine learning are relatively new fields, there is no given industry standard to complete AI/ML projects. But people generally abide by the following steps as discussed which we can call as the pipeline/workflow of ML projects.

Generally, ML projects look like as follows:

- First, we decide what thing/task we want a computer to learn.

- Then, we gather data. (This step is very important. We need to have data first.)
- Then, we carry out some pre-processing on it so that computers can understand it.
- Then, using that data, we train the computer/machine. It will make mistakes and learn from those mistakes. It will then capture patterns to not make mistakes in the future.
- Then, we check whether the computer has actually learned or not. (By asking it about the data which it hasn't seen in the training phase) If it has learnt, then use the machine to predict future incoming data and probably make some money from it!

Simple enough right? It will make more sense once we discuss them in detail, with the help of the following image:

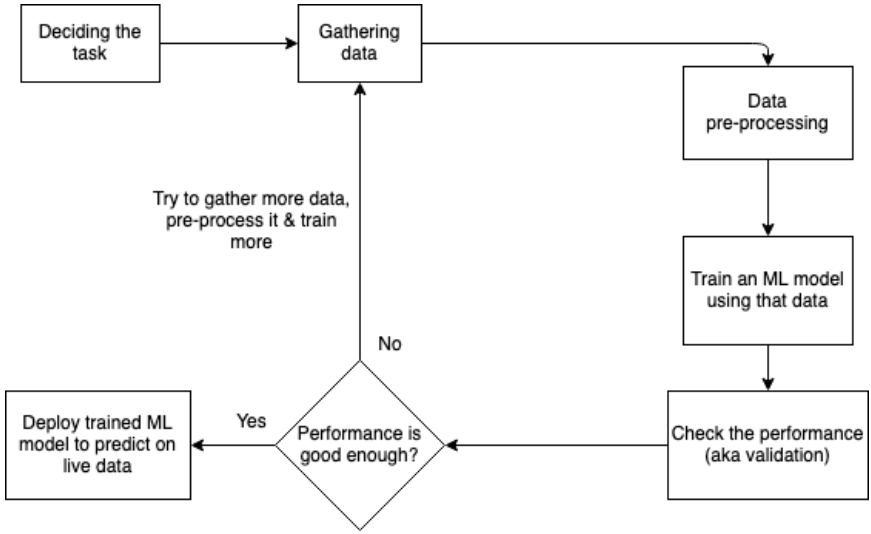


Figure 3.1: *A typical process of an ML project*

Let's break it down.

1. Deciding the task

We first need to decide what we want to teach computers/machines. Once we have a task, we then think,

Is the machine learning the right tool for this problem?

Does it have some value from the business perspective?

By checking all factors, we make sure that investing time and resources in this project makes sense, and whether there are any chances of getting a huge **return on investment (ROI)**.

2. Collecting data

We need to have a good amount of data, without which the process won't go any further.

From where do we can collect/gather data? There are many open-source websites, such as **Google datasets search, Allen, Google Research, and Google Cloud Public datasets**, where people can go and search for whatever data they are looking for.

Sometimes, collecting data becomes a differentiating factor between good products and not so good products. Google created a monopoly in search because they have data that no other companies have.

It's all about data!

Quality and quantity of data is perhaps the most important factor in any machine learning project.

No matter how much we try, if our data is not of high quality, it is very difficult to achieve the best results.

3. Pre-processing data

Now that we have data, we can start exploring and carry out some processing on that data. Exploring the data by plotting some graphs and charts is called **exploratory data analysis (EDA).** We visualize data by plotting them in graphs, like bar charts, pie charts, etc. Let's take a look at the following example of the *Titanic dataset* to understand this clearly:

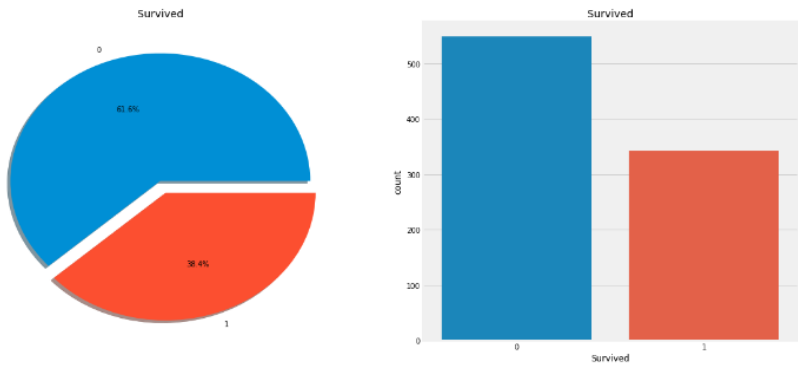


Figure 3.2: *Pie charts and vertical bar charts for Titanic data*

The next step is to carry out pre-processing on the data. For example, converting data into numbers, modifying it, cleaning it, normalizing it, applying some Maths on it, etc.

Let's say there are some empty rows inside our training data spreadsheet, which we need to fill. This is called "handling missing data".

Following are the steps for handling missing data:

- If data is numeric then,
    - We can fill some values in place of missing values.
        - Like, take the average (i.e., mean) of all other non-empty rows and place it there. We can even compute the median or mode of non-empty rows and replace null values with that.
    - We can remove rows with missing data from the table.
- If data is not numeric then,
    - We can select the most common value in the data and replace null values with that.
    - We can also remove the rows with null value here.

People carry out pre-processing on their data for various reasons; for instance, to clean data or remove noise (i.e., unwanted things) to make it easier for models to capture patterns.

There are a lot of things we can do here.

But ultimately, it's all about transforming data in some format so that the computer can learn better.

One of the things people do is *feature engineering*.

What is feature engineering?

In simple terms, feature engineering means using our domain knowledge, creating more relevant data from the data we already have.

In other words, we help machines to find patterns/characteristics by giving more relevant data. Let's consider one example.

Let's say, we want to make an ML model to predict whether a passenger will survive the famous Titanic ship crash or not. We're given all the personal details of the passenger, such as:

- age
- gender
- height
- fare (how much fare a passenger has paid for the journey)
- family_size (how many family members of the passenger are traveling with him/her)
- p_class (Ticket class of the passenger. 1st, 2nd or 3rd)
- survived (whether the person survived the Titanic crash or not)(We use this "survived" data as label / target. Not as input data to model.)

Now, using this information, can we predict whether a passenger will survive or not? All these information about each passenger are called features. Can we make more features out of existing features that may help our model in prediction?

We can make one more column called **alone** which tells whether a passenger has at least one family member on the ship or not? We can make this feature for each passenger using the family_size feature, as shown in the following graphs:

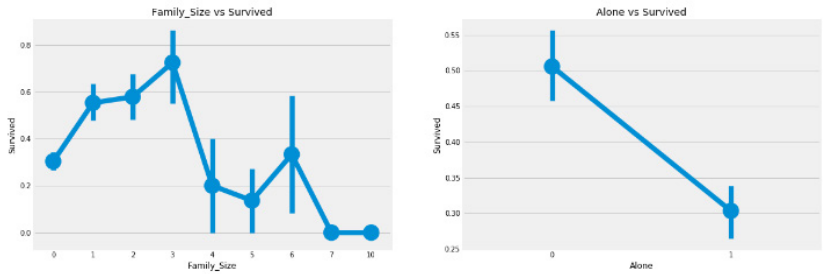


***Figure 3.3**: Graphs of "family_size" and "Alone" vs. "Survived"*

We can see in the RHS graph that passengers with at least one family member have a 50% chance of survival while passengers with no family member have a 30% chance of survival! Our ML model can use this finding to perform better in prediction.

We can also divide the "age" column into an "age_band" column. For instance, if the age is <12 years, call the passenger "child", between 12-18 call them "young", between 18-25 "young_adult", 25-50 "adult",

etc. This feature may help our model to calculate the probability of a child to survive the crash, as shown in the following graphs:

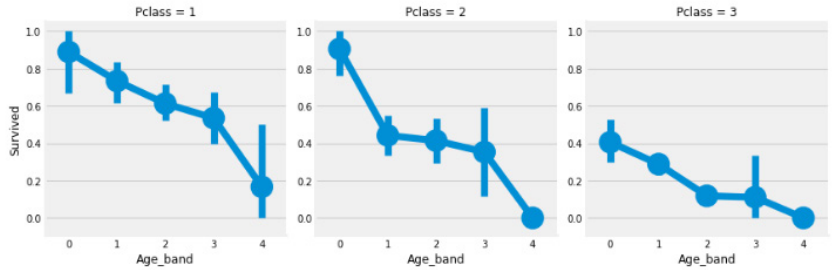


Figure 3.4: *Graphs of "age_band" vs. "Survived"*

We can see that the survival rate decreases as the age increases, irrespective of the "p_class". This type of finding in data may turn out to be very useful in the ML model's prediction. If you notice, what we are doing here is that we are creating more relevant information out of existing information. That's what *feature* engineering is.

Here we have also converted the "age" feature into "age_bucket", by binning them to some buckets. Sometimes, we even transform some features by dividing them by some value to make it more uniform and regular. This type of transformation is called **data transformation**.

People also create more data from existing data using something called **data augmentation**. In data augmentation, we modify the data to create more data.

Why do we need more data?

The more quality data we have, the easier it will be for the ML model to capture features.

Again, preprocessing is all about helping machines to capture patterns/characteristics better and faster.

4. Training the computer/model

This is the main step. Here the model captures patterns in data. Let's call it the training phase. In this step, *using some ML algorithms and techniques*, we train the computer to capture patterns and remember them.

There are many ML algorithms in machine learning to choose from. Each has some strengths and weaknesses depending on the data. Some are good for images and some are good for text while some are only good for spreadsheet type tabular data.

For the introduction, we can call models and algorithms kind of similar. They both learn from data.

These algorithms come from mathematics and computer science fundamentals.

In a nutshell, they all capture patterns but the method of capturing patterns differs from model to model. Sometimes, especially for deep learning, models take days or even weeks to train. The training speed depends on what type of hardware we are using. If we are using a good GPU, the training time will be less.

In other words, models will capture patterns faster.

5. Checking whether the model has learned what we wanted it to or not

This is also called the "validation" phase. Just like how children give exams in their school, we take a test of the computer/model to check what it has learned.

If the results are not good, we train the model again with a different approach. Validation is very critical in real-world projects. We need to check if the model is overfitting or not. (We have already seen what overfitting is in the "*How does a machine learn to do things*" part in the previous chapter)

A trustworthy validation setup is very important.

So, if the model performs well in validation, we can say that this model is good and we can use it for real-time data in the future. In other words, in validation, the model is evaluated using the remaining data that wasn't used during training, helping to gauge its real-world performance.

Here's how a conversation will look like between a human and a machine about validation:

Conversation time

Human: Machine, you have captured relevant patterns from the data I gave you, right?

Machine: Yes. I have captured them.

Human: Okay. So, let me ask you something which is similar to the type of data you have seen in your training but it was not included exactly in your training data.

Machine: Okay. I am ready for the test.

…. [Human asked the machine a question, to which it gave a wrong answer.] ….

Human: No. You are wrong. You haven't captured the pattern I wanted you to capture. I need to train you again.

Machine: Oh, I failed your test. I will try to capture patterns one more time in a different way and will pass your test next time.

Human: Okay. Good boy.

Why do we need validation?

We need validation to check whether our ML model has actually captured relevant patterns/characteristics in data or not. In other words, validation helps us in being sure of whether the model has actually "*learned*" what we wanted it to or not.

How do we work on validation?

Generally, we test our ML model's performance on some other similar data. But this data is not included in part of its training data.

In validation, we use data that was not used in the training phase. In other words, we check whether the model has learned what we wanted using unseen data. There are many methods for validation. **Cross-validation** is one of the most famous validation techniques.

6. Use a trained model for future unseen test data

This is the most awaited phase. The computer has learned what we wanted it to learn. We then use that trained model to perform tasks in the real world.

This is the step where companies make money by providing unique features and experiences. So, now you have an idea, about what a real-world project looks like and what steps people follow in ML projects? "*Now it's time to do nothing and let the machine work for you.*"

We can repeat the above cycle multiple times in the future to keep improving our model's performance. Keep in mind, as AI and machine learning are relatively new fields, there are no industry standards to complete AI/ML projects.

However, people generally follow the steps we have discussed earlier in this chapter. There are many things that we can add to the aforementioned steps, but I haven't added them to keep things simple. Each of those steps are handled by separate teams in a company.

For instance, some people only work on building models; some people gather data, etc. ML projects in the real world follow certain steps.

Create data, preprocess it, train models, validate it, use it in the real world, and make some money!

How to monetize AI?

In other terms, *how to earn profits from AI?*

We have seen the basics of AI and machine learning, how machines learn, limitations, and advantages of ML. Now, let's understand AI from a business perspective.

How can one company/individual monetize AI?

Traditional non-technical businesses don't leverage the powers of computers and AI. Let's first explore them a bit.

Traditional businesses

Traditional businesses are those that use old techniques and methods to reach out to their potential customers and sell their products/services.

Traditional businesses revolve around "people and capital", while businesses backed by AI revolve around "AI and capital".

The fundamental limitation of non-tech, legacy, or old businesses is that they can't scale fast like the IT-based companies.

Traditional businesses reach their potential customers via billboards, print ads in magazines and newspapers, phone calls, etc. while businesses that leverage the benefits of computers and mobiles, reach their customers via platforms like social media, email, and text messages.

The second method is way faster and efficient than the first one.

When WhatsApp or Facebook gets a new user, it costs them almost nothing compared to other non-IT businesses (Customer acquisition cost is non-zero here but it's very less comparatively). They just need to spend on data centers and employees.

So-called "social media influencers" will create content for new users and their friends and relatives will make the platform more engaging for them. That's what makes an IT-based business easily scalable to billions of people. It would take traditional businesses several decades without leveraging technology.

We just need to automate things like customer onboarding, reaching them regularly, accepting payments and selling products, etc. Traditional companies need people to do those tasks.

That's not it.

We haven't added AI to them yet.

AI automates things at a whole new level and scale.

We automate tasks that were unique to humans, such as reading, speaking, listening, translating from one language to another, viewing images, recommending products, etc.

AI and ML scales at the same speed as IT companies. But at broader and new tasks.

Limitations of traditional non-tech businesses

The following section explains the limitations:

- **Can't scale fast:** The number of people that a traditional business can reach in the first year is limited to nearby local areas or cities. One of the main hurdles for the traditional

companies that don't use the powers of computers and AI is distance. Customers need to visit physical stores or offices to use their products or services.

We don't have to visit Facebook or Google headquarters and offices to use their products and services!!

This sounds naive but it's a major factor in their market penetration.

- **Higher dependency on human employees:** As a non-tech business grows, it needs to hire more people who can manage day-to-day tasks and handle their customers. Do you know WhatsApp just needed 50 people to manage their 900 million users? That's huge. Traditional companies need to pay those people regularly.
- **Lake of automation:** This is crucial. Automation of things is a catalyst in businesses. Proper automation cuts the costs of employees and improves the overall performance as well.

The preceding 3 limitations are the major ones that traditional businesses have and they are not present in AI-backed businesses.

Solutions of problems for traditional businesses

Let us understand the solutions now, which are as follows:

1. Websites and mobile applications

These two help IT companies to scale fast.

Almost everyone has a smartphone in their pocket.

Internet connection has reached almost 1 out of 2 people on the whole planet (that's ~4 billion people with an internet connection). Having mobile applications and websites of the company enable you to reach all of them. Then we add AI to those apps.

Post that, customer engagement increases drastically.

This is the reason why IT companies scale so fast.

2. Machines do all the work (almost full automation)

Codes written by software engineers can serve millions of customers/users simultaneously. The whole customer journey through their services and products are automated.

AI and machine learning-based models built by data scientists can improve customer's experience and create more engagement. They chat/talk with customers to handle their queries, recommend customers other items they can use, make sure the customer is happy with the product, etc.

We need to write these codes only once and it will serve us forever!

I keep using the word automation because that's at the very core of AI-backed businesses — automation at a very big scale in diverse tasks.

Two major benefits of automation:

- **Performance:** Humans are error-prone. Machines don't make mistakes. This is a big factor in increasing automation.
- **Cost:** Companies need to pay humans continuously every month/week while they can buy the software once and use it unlimited number of times.

Both "*performance*" and "cost" favor in support of automation. Companies are eager to automate tasks to reduce costs and improve performance.

Full automation will change our daily life by a huge positive impact.

- We don't have to waste hours in traffic.
- We don't need to pay attention to the road while driving. Vehicles will be self-driving. We can do other important tasks than driving.
- We can get personalized education while staying at home. *High quality, low cost.*
- Doctors can become more efficient. Easy tasks can be solved by AI.
- Defense and security will become more efficient and hard to break than ever before. Governments don't have to spend billions of dollars in the defense department.
- We can fight climate change with AI. Even farmers can get help from AI. They can automate their daily tasks.

And many more like that! Automation can increase the productivity of every person, increase revenues, and cut costs for companies and eventually boost the economy. Gradually, things we need to care about in day-to-day life will decrease and it will allow us to focus

only on highly important things. Now let's see what things can AI do for your business.

Three main things AI can do to generate profits, are as follows:

- Create new products *(to increase revenues)*
- Improve existing ones *(to increase revenues)*
- Automate process *(to decrease costs)*

This idea was given by one of the heroes of AI, *Andrew Ng*. Let's discuss each of them in detail. Refer to the following image for a better understanding:

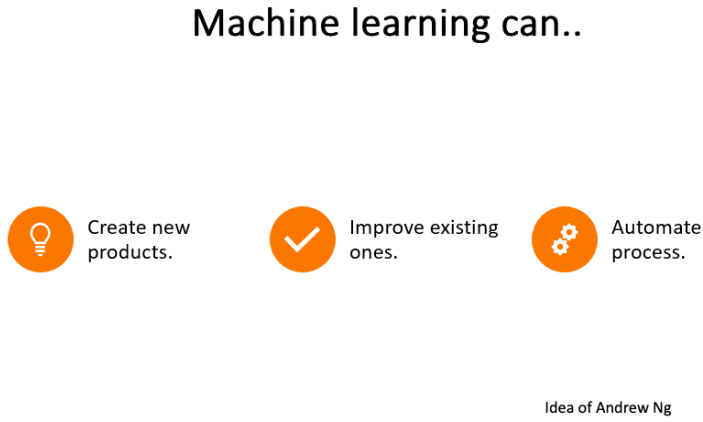


Figure 3.5: *Three fundamental ways ML can help in businesses*

From *Figure 3.5* we learn the following:

1. **How to create new products (to increase revenues)**

   AI and machine learning can create new products and sometimes entirely new markets. Earlier, without AI these products and markets were not possible to create. Let's see some examples in brief.

   "Revenue - costs = Profit"

   So, as revenue increases and costs decrease, profits increase.

   - **Self-driving vehicles:** This is one of the best examples. Vehicles were there for decades but it has changed the way we interact with them. Drivers don't need to pay attention to the road while driving anymore and can do all other important work, like attending a phone call or preparing a presentation.

Not only that, it has also created a whole new market for self-driving taxis and renting our cars for others for transportation and make money for us. We can make money by owning a self-driving vehicle and we don't need to do anything! Customers will use our car for transportation, AI will drive the car and you can sit at home!

How cool is that!

- **Chatbots:** This is also a whole new market to serve your customers automatically. As companies' total number of customers grow, we need to handle their queries and doubts. When you have millions of customers, managing call centers is very costly. Instead, what you can do is hire some bots (or virtual robots) and they will make conversations for you and will handle customer's doubts and concerns.

  Chatbots are machines that can handle thousands of customers at a time. And make no mistake, the quality of serving customers is not compromised. They have conversations just like humans. Conversation bots like Apple Siri, Amazon Alexa, etc. use tons of AI. This technology is not at its full potential as of now (year 2020).

  But it can for sure reduce traffic for companies' call centers.

  Other products and services are as follows:

  - **Translation bots**
    - Language is one of the things that are core to humans. AI can translate whole books and blogs for you!
  - **Stock market analysis**
    - 80% of the stock market is on auto-pilot, handled by machines!
  - **Dating sites**
    - Machines will suggest the customers their possible matches for life partners like an expert matchmaker!
  - **Recommendation engines**
    - AI can recommend movies, TV shows, songs, etc. The idea is to increase customer engagement by making smart suggestions.

In short, AI has the capability to create whole new products and markets.

2. **Improve existing products and services (to increase revenues)**

   Legacy companies can use AI to improve their existing products as well. Refer to the following image for a better understanding:

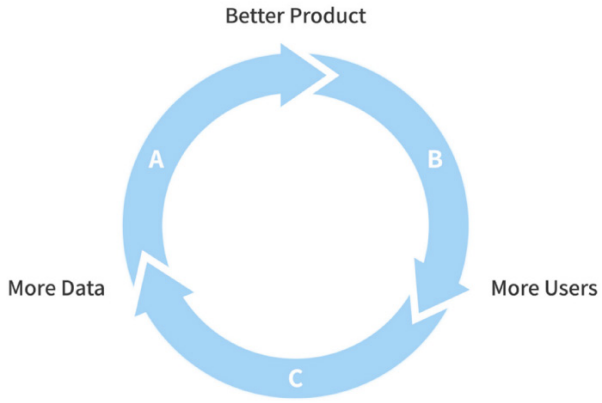


Figure 3.6: *Cycle of AI products*

The preceding figure shows that more the data we gather, better the product becomes as the AI model becomes better. As the product gets better, more users will come. More the users come, more data the come, and this cycle continues.

Traditional and legacy companies/businesses can do a lot of things leveraging AI and machine learning. User attachment will increase with products because of smart services and products.

Overall, this will help AI-backed businesses to kill the competition. Let's take a look at the following examples:

- **Recommendation:** Movies and TV shows were there for decades. Adding recommendations increased user engagement to a whole new level. Recommender systems essentially capture human behavior. Netflix, Spotify, Amazon, etc. are using recommendations every day.
- **Art creation:** AI (deep learning models to be specific) can help artists to imagine new possibilities. The quality

of paintings and music from artists and music creators can increase by a huge amount. This also differentiates artists' work and kill the competition.

- **Google maps:** Maps were there for decades. But Google maps does something different. It optimizes total travel time for every user by providing the best routes. Millions of people use it every day. Google is also earning millions by selling it to companies like Uber, Zomato, Swiggy, Ola, etc.

In short, AI can help improve existing products and services by adding new features to kill the competition.

3. **Automate process (to decrease costs)**

The essence of automation is to train machines to do repetitive tasks. We can save time and resources as things can be handled by machines now. Ultimately, it cuts the costs for the company. Let's take a look at the following examples here as well:

- **Language translation:** This is one of the best examples of automation. There's a huge market for translation. News agencies, academic organizations, social media companies, etc., need to translate their text content into other languages to increase their potential audience size. To correctly translate text from one language to other languages, one needs to have experience in those languages and should also have the knowledge about the linguistic features of those languages.

  It's a hard thing to automate. But thanks to artificial neural network models from deep learning, this has become possible with the expectation of great performance as well.

- **Visual inspection:** Inspecting machines and processes in huge mechanical factories is a very critical and demanding task. Until now, this process was handled by human inspectors. Humans can make mistakes. Using AI (computer vision algorithms to be specific) we can make this faster and more accurate.

  Companies can not only use these tools in their factories but can also sell them to other peer companies. This saves the cost of human visual inspectors.

We have seen how AI can increase revenues and decrease costs, ultimately increasing profits. Now let's see how we can maintain all that we have gained.

After achieving the growth path, how to maintain it?

Growth alone isn't sufficient. For a better long-term goal, companies need to maintain their market share to survive. Can AI help them do that? YES!

Management gurus say *increasing revenue is more important than cutting costs* as cutting costs can damage product quality. There is only one thing that summarizes it — "maintain" our growth.

To maintain market share,

Create a competitive moat: Okay, what is a moat then? In business and economic terms, it simply means creating a competitive advantage over your competitors. It's simple to understand; in the long term, if one company needs to generate huge profits, it must have something that its competitors don't have.

In other terms, the product you are building must not be easily replicable.

That's what competitive moat means. If you can create something which is very hard to possess or copy for your competitors, congrats, you have a competitive moat.

Now, let's see how to do this with some of the following examples:

- **Google search:** Google is almost a monopoly in web search with ~94% of the market share. How did they do it? By providing best-personalized search results to users. Web search is perhaps Google's most used product.

That's what Google is known for. Google earns through selling ads through its web search in the browser and YouTube. There are many products by Google but most of the revenue comes from advertisements itself. Let's take a look at the following figure for a better understanding"

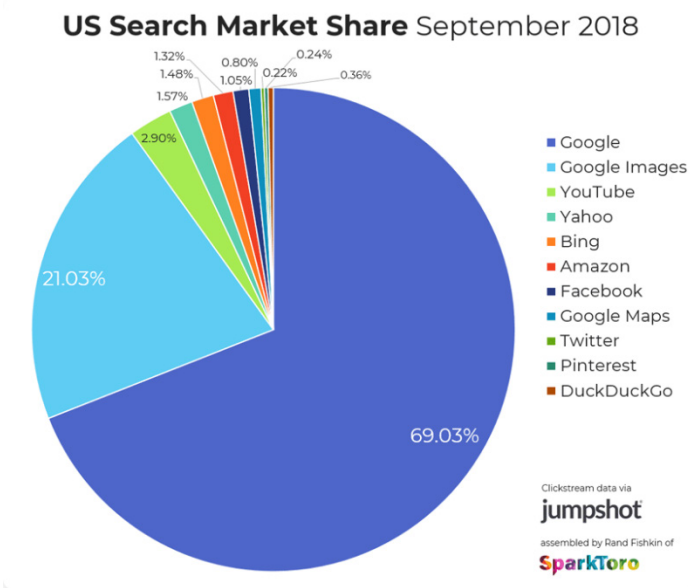


Figure 3.7: *Google's market share in "Web search" https://sparktoro.com/*

They have created and leveraged a diverse set of machine learning and deep learning models to make their search more efficient and scalable across the globe. Nowadays, *they are also using* the latest deep learning models like **BERT** to improve search quality.

- **Facebook feed:** Facebook Inc. is also almost a monopoly in social media with more than 2 billion monthly active users on Facebook alone (they also own Instagram, WhatsApp,

and Messenger). Refer to the following figure for a better understanding:

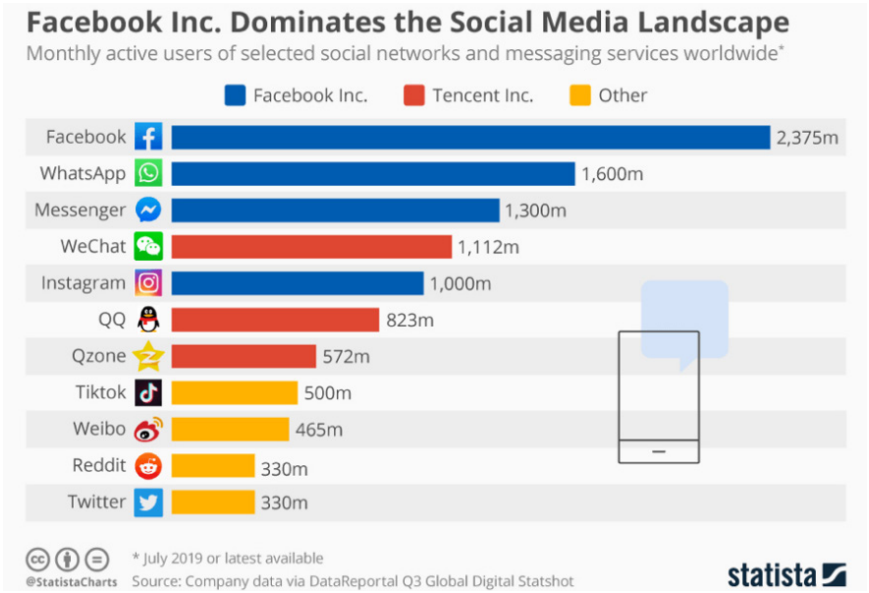


Figure 3.8: *Facebook dominance in social media (source)*

Okay.

As we can see, providing the best products and user experience is the key behind it.

The crux is, we use the powers of AI and machine learning to create the best products. No one can (as of now) provide good personalized search results better than Google. And that is their competitive moat. No one can replicate those search results on their own. That's how AI-backed businesses maintain their market share after the growing phase.

In short, to increase revenues, we can build new products and improve existing ones. To decrease costs, we can automate processes.

How can you leverage AI in your business?

After discussing how AI can play a crucial role in your business, let's discuss how to use it. Basically, we will need 3 things to use AI in your products/services, which are as follows:

- High quantity and quality data
- Best talent who knows AI
- Domain expertise

Let's understand each of them in detail.

High quantity and quality data

Most tasks that require some kind of intelligence or decision making and for which lots of data is available to train the models can be automated to some degree with AI and machine learning (ML).

Data plays a crucial role in the performance of AI models.

Let's take an example. Assume that you are working in the bank transactions sector, and using an AI model, you want to predict whether a financial transaction is fraud or genuine. Using this model, AI will detect fraud or illegal financial transactions 24x7, 365 days for you. Now, you will have all the data about transactions that happened in the past like whether it was fraud or not, transaction amount, date, time, people involved in transactions, etc. Using that data, we can train an AI model, which will learn the following:

"Given all the data about a transaction, predict whether it's genuine or illegal/fraud."

Another example can be,

"A person has taken a loan/mortgage from a bank; AI predicts whether he/she will be able to return it on time or not".

You got the idea. In short, almost any task can be automated to some degree by AI, if we have stored lots of data from the past for that task.

Other than data quantity, data quality also matters a lot. Like, data should not be biased towards any particular thing. By "biased" I mean, it should equally represent every aspect. Like, Google predicted Black people as gorillas! Why? Probably because their data was biased.

Best talent who knows AI

This is kind of obvious. A small team of people who know the ins and outs of algorithms and techniques of AI and machine learning is crucial. A good problem-solving skill is crucial. This is a bit technical

but, one thing that is good to have is intuition. There are tons of algorithms and techniques to use. A good intuition is very helpful to start in selecting the algorithm/model.

Domain expertise

A good AI team may have the knowledge about how to build AI/ML models and all. But it's good to keep business and domain-specific people in the team. By "*domain-specific*" I mean, people related to the project field.

For instance, if our project is related to the biological field, some people who have that kind of knowledge and background should be there.

Some business people inside the AI team are also good to have, in order to check whether solving this problem will help the overall business or not. Generally, this task is handled by the founders of the companies. Ideally, we should select those projects which are feasible and valuable for the business in the long term.

Along with the aforementioned three requirements, a good amount of hardware is also good to have, especially for deep learning algorithms.

Artificial neural networks are slow to train in normal CPU (computer processor unit). People use **GPU (graphics processor unit)** or **TPU (tensor processing unit)** for faster speed.

One should comparatively put more weight on one thing, i.e., quality and quantity of data. Data has enough power to make or break the whole project. Refer to the pitfalls to avoid at the end of this book. One should keep in mind those things before investing in AI projects.

So, to use AI in your business/company, a huge amount of quality data is very crucial.

How to use AI in your existing products?

This is especially important for **small and medium businesses (SMBs).** For SMBs, creating new AI-backed products from scratch may not be a good idea. So, how can they improve their existing products by leveraging AI and machine learning?

Three things we can do to leverage AI in existing products/services as follows:

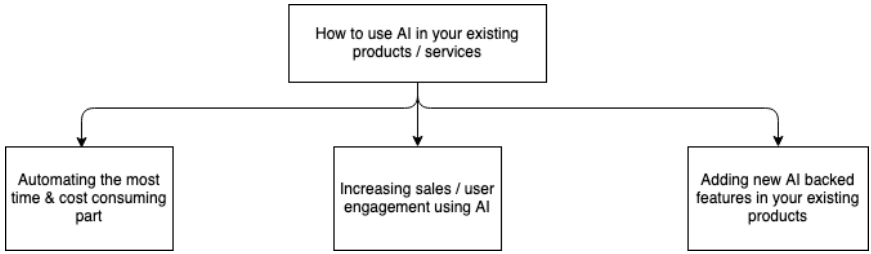


Figure 3.9: *How to use AI in your existing products?*

From the preceding figure we understand how to use AI in the existing products/services, in the following manner:

- Automating the most time and cost consuming part of the process.
- Increasing sales or user engagement with the product using data that we already have.
- Adding new AI-backed features in our existing products.

Let's take a look at each of them in detail

Automating the most time and cost consuming part of the process

Imagine a production pipeline for our existing flagship product. Our product passes through all the stages to be ready to serve the customers.

Now, amongst one of those stages, one stage is the most time/cost consuming. If we can automate it, our production will move faster. The total time taken, starting from raw materials to the final product will reduce.

Let's take one example. We have already discussed this briefly. In our product development pipeline, we have one **visual inspection** task. The goal of the task is to check the quality of the product.

Right now humans are doing it manually. That's why this task is comparatively slower and costlier.

What if we can automate this?

Images will be captured from the camera and that will be sent to our AI model. We have trained that model to classify whether products in the image have good quality or not.

AI models can classify ~5-10 images per second on a decent GPU hardware server setup. After that, some mechanical robot arms will remove bad quality products.

That's so fast! Definitely faster than humans.

Because of this automation, total production time taken, starting from raw material to the final product will reduce.

And our users will get products faster in their hands.

Increasing sales with the product using data we already have

How can AI help in *sales*? Let's take one example.

Demand forecasting

If we can forecast or predict that this product is going to be in high demand in the next month or quarter, we can produce more of it and store them in our warehouses. If we store products that are not going to sell that much, then there is no point in producing or storing them. In other words, it's a kind of "optimized use of warehouses".

Dynamic pricing

We can even set a dynamic price for that product according to demand during the day. E-commerce companies like Amazon use it for their products to increase profits.

Adding new AI-backed features in our existing products

Adding new innovative features will increase customer engagement with our product. It will also help in killing the competition. One good example is **Snapchat's filters.**

Snapchat uses facial recognition and other computer vision techniques to create filters around a user's face that looks real. When it added that feature, users got attached to it.

This type of feature can also become a company's competitive moat. So, for small and medium businesses, AI can improve existing products by automation, increasing sales, and adding new features.

Real-world use cases of AI

We have seen how you can leverage AI in your business and also in your *existing* businesses. Now let's see some real-world examples of companies that use AI in their products. We have already seen some examples in the previous chapters in this book.

- Google uses it to automate the best-personalized search results.
- Netflix uses it to recommend the best movies and TV shows for each customer to create more engagement.
- Uber uses it to select the best ride for you.
- Facebook uses it to automatically tag all the people in a photo post.
- Amazon uses it to lower the return rates of shipped products.
- Tinder is using AI to figure out who you're likely to "Super Like"!
- Tesla, Uber, and Lyft are creating self-driving vehicles.
- Google is using speech recognition in their home-related products.
- Smart personal assistants like Apple's Siri, Amazon's Alexa, Google's Ok Google, Microsoft's Cortana, etc., help us to maintain our tasks and improve our productivity.
- Facebook, Twitter, YouTube, etc. remove the violent and non-appropriate videos, images, and text content from their platform continuously using AI.

As mobile and internet usage is increasing rapidly, the usage of AI has increased quite a lot in the last 5-7 years. Let's discuss some examples in detail.

Netflix recommendation

Netflix is using it's recommendation algorithms every day to create more engagement with its users. Refer to the following image for a better understanding:

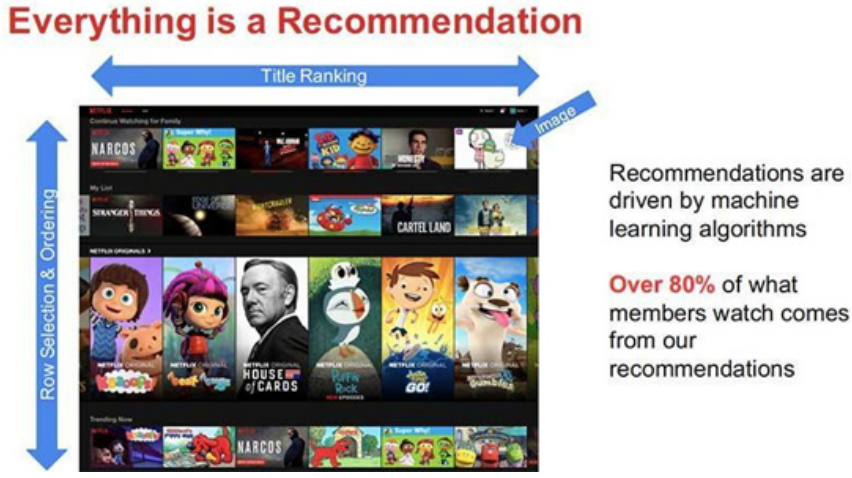


Figure 3.10: *Almost every web series we see on the home screen is a recommendation.*

As of 2014, 75% of their users selected movies based on the company's recommendations. This number was ~6 years ago. Now, this number will be higher. Based on the user's past history, smart suggestions of web series engage users to see more and more content on Netflix.

In 2012, **Netflix hosted a $1 million prize competition** to improve their recommendation algorithm. That is how much the recommendation algorithm is important for the company. Netflix said that its AI-based recommendation algorithms **helped saving them $1 billion per year.** Quoting the exact sentence from their chief product officer,

"The combined effect of personalization and recommendations save us more than $1B per year".

That's what AI can do when we use it on a big scale. We have already seen how recommendations work in one of the previous chapters.

Self-driving vehicles

This is one of the hottest markets in the automobile industry. We have seen this in brief in the "*Create new products*" section in "*How to*

grow a business perspective in AI" part of this chapter. Let's take a look at the following image for a better understanding:

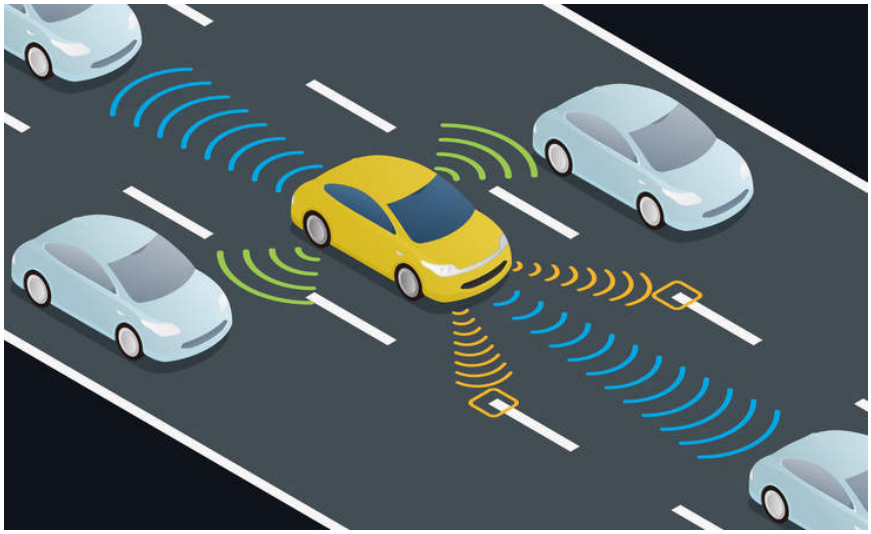


***Figure 3.11:** A self-driving car*

Let's discuss this a bit more.

How self-driving vehicles work?

Self-driving cars fundamentally use **computer vision** at its core. It contains ~6-8 cameras and sensors on different sides of the vehicle, which captures the surrounding and feed it to the machine. Then the machine decides what actions should be taken to move ahead safely.

Models that run behind self-driving vehicles learn to classify objects like cars, pedestrians, traffic signals, etc., from the captured images from cameras. It uses something called **image detection and image segmentation** from the computer vision field. We have read what computer vision is in the previous chapter.

In short, it learns how much the steering wheel should be rotated, when to press breaks, etc., to go ahead safely.

It doesn't rely solely on computer vision. Camera data, sensor data, LiDAR data, etc., all things are considered while making a decision.

One of the most critical factors of self-driving vehicles is safety. Self-driving AI models need to make decisions correctly 99.9999%

or almost 100% of the time, which is very hard to do. But once it is implemented, **a new era of transportation will start.**

Limitations and advantages of machine learning

At this point, you know the basics of AI, machine learning, and deep learning. Let's understand what are some of the things that machines can do and cannot do. We will discuss limitations in detail as they are more interesting than advantages. Refer to the following figure for a better understanding:

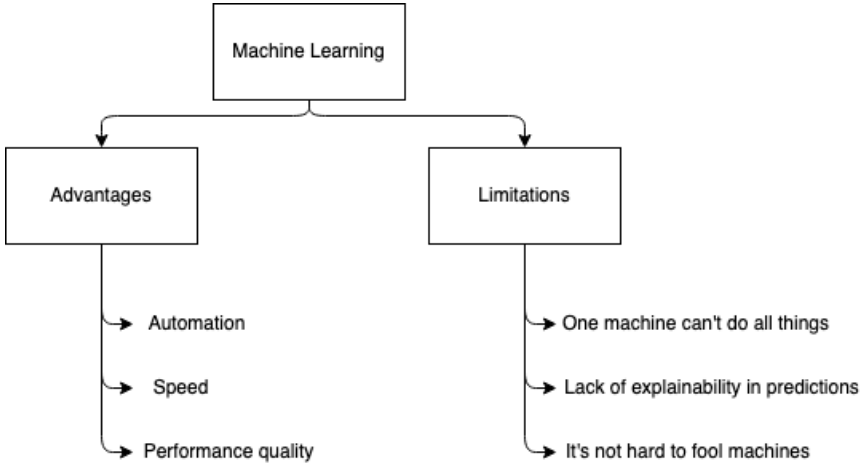


Figure 3.12: *Limitations and Advantages if ML*

Advantages

Let's understand the advantages first. There are three main advantages to machine learning, which are, automation, speed, and performance. Let's discuss them in detail.

Automation

Automation earns profits for companies. We can automate things that were unique to humans, like reading, talking, listening, and speaking. We are essentially trying to copy the human brain itself.

Speed

Once we teach machines to perform certain tasks, like let's say, talking, it can talk to multiple people at a time. That's how chatbots work. Computers are fast. They don't need 5 seconds to reply. They just need a few milliseconds.

Performance

Machines are already better than humans at certain tasks. They can perform better at identifying people from images, translating sentences to other languages, etc., with *lesser errors*. Humans are error prone. We make mistakes. But machines can't make mistakes once we teach them how to do things properly.

Limitations

Let's now see some limitations of ML. It's important to know limitations of ML to understand what it can/can't do.

One machine can't do it all (i.e., one model can't do multiple tasks)

This is a major limitation. Let's say we train a model to translate from English to German. Then that same model can't be taught to identify dogs and cats from images.

We can train our biological brain to learn almost anything, one brain for all the tasks. Contrary to that, machines can't do that. We need to train/build different AI/ML models to learn different tasks.

Currently, scientists are working on this problem; it's not solved yet.

Lake of explainability (also known as trust deficit)

This thing applies more to deep learning models. Neural networks can't explain why it predicted a certain label. Like, let's say we train a model to predict whether a financial transaction was legit or fraud. All the data related to transactions is given. Now, let's say the model predicted that a certain transaction is a fraud. But a banker wants to know why the model predicted this transaction as a fraud. We can't fully do that at present.

Deep learning models (i.e., artificial neural networks) have low explainability. It just predicts labels. This is a major drawback for some industries, like finance and health.

It's not that hard to fool the machine

This is related to the lack of *explainability* of machines which we discussed earlier. Robustness is somewhat missing from deep learning models. Let's say, we train a model to learn whether a photo contains a dog or a cat or a ship or an apple or a human face.

Now, one guy doesn't want you to make money from it. And he attacks your model by just changing a few pixels in the photo. Changes are so small that we humans can't even see what changes were made.

And guess what, the machine failed horribly. This is called an **adversarial attack**. We have read this in the "What is computer vision?" section of the previous chapter. We can't explain why our model failed horribly. This is a major roadblock for many small and big companies.

One more misuse of this technology is called deepfake. Search for "deepfake" on the internet and see what it is. You will be amazed. Other than these, some of the major limitations are also challenges like the following:

- The requirement of large amounts of data
    - Especially for deep learning models, we need millions of samples to train the model for the task.
- Data privacy
    - Can we use millions of users' personal data to train our models?
- Bias in the dataset
    - Whether AI models will perform better or worse, really depends on the quality and quantity of data they are trained on. If there is some bias in the dataset, it'll reflect in the model's predictions.

Machine learning is a fairly new technology. It's in its growing phase. Thousands of researchers from all over the world are working on reducing limitations. In the coming years, let's hope they find some solutions and make machine learning even more helpful for humans!

The major advantages of ML are automation, speed, performance, and minimal errors. The major limitations are that we need more than one model to do different tasks and that they can't explain why they make certain predictions.

Conclusion

So, in this chapter, we understood how AI projects work in companies, how they are using it to increase revenues and decrease the costs, how we can leverage AI in your business and use it in your existing products/services, etc. We also understood how traditional non-technical businesses can be improved with AI technology.

Questions

1. What are the limitations of traditional non-technical businesses?
2. What are the benefits of automation?
3. Which 3 things can AI do to increase revenues and decrease costs?
4. What are the limitations of machine learning?
5. What we can do to leverage AI in your existing products/services?

Chapter 4
How to Get Started, and Pitfalls to Avoid in AI

We have seen what machine learning (ML) can do and what benefits it can bring to individuals and companies. Now let's see how you can start your own journey in the ML field. In this chapter, we will understand how you can get started in this AI/ML field and how jobs are going to be affected by the usage of artificial intelligence (AI). Along with that, we will see at what stage the AI field really is with filtering out the noise made by the social media and news channels.

We will also understand what major pitfalls/mistakes to avoid while using AI.

Structure

- How to get started with AI and machine learning?
    - How to start?
    - How to grow?
    - How to maintain it?

- A realistic view of artificial intelligence
    - Filtering out the noise created by social media and news channels
- Artificial intelligence and employment
    - Fields that can be disrupted by AI
        - Education
        - Customer service/experience
        - Defense and security
- Pitfalls to avoid in artificial intelligence

Objectives

After studying this chapter, you should be able to:

- Start your journey in the AI/ML field.
- Understand the advantages and limitations of the field.
- Realize what common mistakes/pitfalls we can avoid while using AI.

How to get started in AI and machine learning?

After seeing the potential of the AI and machine learning field, you may want to get started with this interesting field. In this chapter, I will show what you can do to get started in the AI/ML field.

I will divide this chapter into 3 parts, as shown in the following image:

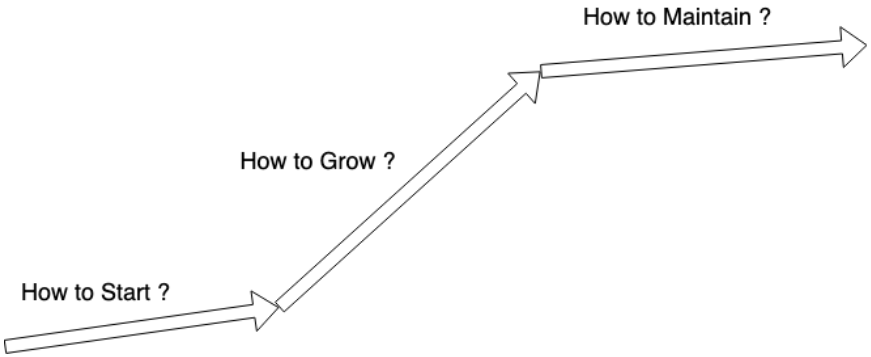


Figure 4.1: *Three stages to start with anything*

- How to start?
- How to grow?
- How to maintain it?

How to start?

This question is very common in almost all of the aspirants.

One basic analogy is that one needs to have at least some basic knowledge of statistics, calculus (especially for deep learning), and linear algebra to understand concepts of machine learning and deep learning.

Remember, I said "basic". You don't need to be an expert in it. And if you have done well in the high school studies, you most probably have the required basic knowledge. So, congrats. You already have a base in ML!!

While starting, remember one thing:

Learn in the BFS manner, not in the DFS manner.

What I want to say is, try to grasp the overall idea of **each and every concept of ML.**

Don't just pick one algorithm and go deep into it. Remember, in ML, there is no one such algorithm that always gives the best result. So, learn the basic concept of every algorithm, and apply them to some real-world data. Now, the sources — Andrew Ng's Coursera course is a perfect place to start. The way he teaches ML concepts is really good for beginners.

Also, his Stanford course is good if you have a background in mathematics, as they are more focused on mathematics. By pursuing these courses, you will at least have some idea on what ML is all about, and how things work. That's what our goal is!!

Congrats, if you have reached this far.

How to grow?

There are two main ways to grow your knowledge in ML, which are as follows:

- Projects
- Competitions

Try to build something which includes some algorithm which you have just learned. It's not compulsory to select a high-level cutting edge project definition. You can even choose to use the simplest linear regression in your project.

The aim is to implement it and build some real-world use cases.

While building them, you might face many issues. But that's perfectly okay!! Find some fix as per your understanding, and move on.

The second way is Competition. **Kaggle** is one of the biggest data science communities. Take part in running **competitions** and learn as much as you can. Platforms like **KDnuggets, Analytics Vidhya** are also good. It's completely okay if you rank low in these competitions.

Your aim is not to win millions of dollars by winning these competitions; your aim is to learn something. RANKS REALLY DON'T MATTER, if you are learning new things and improving yourself.

You know, generally, in these ML competitions, 1st rank will have, let's say, 0.98598 accuracies and the person at 200th rank will have 0.97198 accuracies. The rank difference is very high, but the score is nearly the same. Read and try to understand other people's views on the same problem by **Kaggle kernels.**

Many working professional data scientists share their point of view using kernels. That is the best and fastest way to capture important things. They have experience and knowledge. See how they think, and enhance your thinking process that way. By doing this, you will surely no longer be a beginner in ML. Your level will have improved.

Congrats!!

How to maintain growth?

It's never about making it; it's about maintaining it.

This is very important. ML is a vast ocean. Even some great researchers don't know all the concepts fully. And you actually don't need to digest all concepts. Even if one knows all the concepts fully, his/her knowledge will not be considered "full" after a week or two.

Because in ML, every week something new comes up.

Keep updating yourself by reading the latest **research papers.** Reading research papers is an art. Sometimes, you need to read them at least 2–3 times to grasp the very core of it.

Try to understand what it is saying.

Avoid mathematical formulas at first, if you find it cumbersome. Just grasp the idea. Learn these concepts and apply them to projects and competitions. You can keep yourself up to date with the cutting-edge data science research by referring to the publications from top-tier conferences such as **ICML**, and reading blogs such as *Import AI.*

If you don't know how to know all the things about these top conferences, *here* is the best source to know more about each upcoming event in ML. Pursuing ML alone can restrict your growth. Because you never know how this concept can be viewed in some other way as well. Here come meet-ups and tech-talks.

PyData is one of them. PyData arranges various events and meet-ups around the world. Join them if you can. You will definitely learn something new.

PyCon is also good (especially for those who prefer Python as their working language which many of us do). Watch some success stories like How to Learn Machine Learning in 6 Months (**https://www.youtube.com/watch?v=MOdlp1d0PNA&feature=youtu.be**)and How to become a Data Scientist in 6 months (**https://www.youtube.com/watch?v=rIofV14c0tc&t=100s**) to enhance your self-learning!!

Also, keep visiting websites like *Towards Data Science, Hacker Noon, Becoming Human*, etc., regularly. All these websites have plenty of great blogs written by ML professionals. Also, use *Twitter* wisely. Follow some great professionals that are there and never be late to catch new things. And lastly,

Don't expect results quickly.

You can't expect to understand and implement all the concepts in half a year. Things will be understood as time goes. Don't compare yourself with anyone. I personally don't know many things about the AI field. Knowing what you don't know is also an important thing. Just be the best version of yourself. Everyone is unique.

And yes, you will need support when things are not going well. So, make a few friends with whom you can share everything, including your achievements and sorrows. They will help you.

In short, if you don't give up, ML is your cup of tea!!

A realistic view of artificial intelligence (AI)

We have watched the basics of AI, ML, DL, etc. Now let's have a look at ML by clearing out hype and noise created by (social) media and the news.

What we see on social media and news is just one side of machine learning and AI. Some media and news channels want more people to read their blogs and posts, so some of them try to make them a bit catchy. They only show the bright side of the field.

Have a look at the limitations of machine learning discussed earlier whenever you see such artificial intelligence or machine learning related news. We see news of machines beating humans at a bunch of tasks. But what we don't see is, their limitations, like if we add some noise (irrelevant things) in the data or modify our data a bit, there are chances that their performance will decrease.

The AI field is in its very early stages.

Because of that, a lot of things about the field are not clear in the industry right now. People from many companies are trying to figure out what machine learning actually is and how they can use this technology in their old structured businesses.

With that being said, let me tell you, change is happening faster than we think. Many companies are creating their core business models and flagship products around AI and machine learning and some of them are even earning profits.

Today, basic knowledge of this AI technology is very important to have, which is what this book's aim is — to share knowledge about machine learning and artificial intelligence in a purely non-technical and carefully summarized manner. In short, the ML field is in its very early stage. We need to figure out many things before using it in mainstream day-to-day life.

Artificial intelligence and employment

After understanding the potential of AI and what we can do using AI, one problem seems to emerge.

Loss of jobs

We know that one does not simply get a job! Getting a job is not easy. When we start automation at full scale in the industry, what will those people do who were performing those tasks earlier? This is a very serious problem. We cannot or should not ignore this.

Many scientists and thinkers are trying to figure out this problem. There's no denying that machines are better than us in certain tasks. Then why should employers need to give salaries to humans and not buy a machine to do the same job with better performance?

Let's see some examples.

1. Self-driving vehicles

There are millions of drivers out there, whether it's a bus, truck, car, or bike. Their main source of income is driving. Now, if we teach machines how to drive better than humans in every condition and use them in the mainstream day-to-day life, what will the drivers do?

With self-driving vehicles, one other field is also going to be disrupted, which is transportation.

2. Transportation

Transportation is a multi-billion dollar industry. Trust me if it gets disrupted by self-driving vehicles, a lot of things are going to change. Autonomous driving is considered as one of the most revolutionary uses of AI in the real world.

3. Education

Some companies are making robots and models which can give personalized education to almost every child. The traditional schooling system is not very personalized. There's one teacher for ~50 students. So, the teacher can't pay enough attention to every child regularly. But machines can.

Machines have the potential to give personalized education to thousands of users simultaneously.

4. Customer service/experience

AI has already begun to disrupt customer service. According to a blog, 85% of all customer interactions will be handled without a human agent by 2020. The usage of chatbots is increasing day by day. One of the biggest benefits of chatbots is that they can serve multiple customers at a time as their response time is faster.

Due to their ability to accurately understand what the customer is saying, sufficiently advanced NLP algorithms may replace customer support executives altogether.

5. Defense and security

Advancements in autonomous weapons are also increasing. Apart from autonomous weapons, image recognition and video recognition can be used for the surveillance of the general population.

The government has started using AI in security as well, via CCTV cameras in public places across the country. Such technology has already seen deployment in China, where widespread facial recognition algorithms are being used to *create a social credit system.*

In social credit systems, citizens are measured by some points, based on their actions, which are logged using AI-based cameras. Some people also believe that machine learning may take some jobs but it'll create new jobs as well.

There's a fascinating concept, called **Universal Basic Income (UBI),** which says the government should pay every living person some money regularly for doing nothing so that they can live as their jobs are automated by machines. People are trying to find some ways to solve this problem.

There are chances that this problem will only get bigger over time and at some point, we will need to tackle it.

As AI expert Andrew Ng puts it: *many people are doing routine, repetitive jobs. Unfortunately, technology is especially good at automating routine, repetitive work.*

While AI won't replace all jobs, what seems to be certain is that AI will change the nature of work.

The only question being, how rapidly and how profoundly will automation alter the workplace. In short, there are chances that

employment and jobs will get affected once we start using machine learning systems in the mainstream.

Pitfalls to avoid in artificial intelligence

Let's see some things to keep in mind while making plans to use AI in your business.

1. AI is not magic

This is a very common myth. AI cannot do everything. It's not a magic pill that can solve every problem we want it to solve.

AI is not magic. It's just pure math.

By reading only positive news about AI in the news and social media, *it's easy to assume that AI can do everything*. That's not true. It's not easy to train AI models. So, one common pitfall to avoid is, don't assume AI can do everything. AI/ML models are only as good as the quantity and quality of data provided to train them.

2. Performance of AI models will degrade over time

This is kind of counter-intuitive. Why will the model's performance degrade as time goes by? Let's take for instance, we build a model for fraudulent financial transactions. We achieved 95% AUC (AUC is just a metric to measure the model's performance) on validation data. And we deployed it for real-world live financial transactions.

After a few months, the model performance's will most likely not be 95% AUC. The reason behind this degradation of AI models is,

Real-world is dynamic while AI models are static if we don't update them.

Remember we read that ML is all about capturing relevant patterns in data? Well, those patterns change over time. And to remain relevant, models need to capture new patterns again to maintain the performance. Patterns for fraud transactions that our model with 95% AUC performance has learned, will not remain the same in the

future. People will find new, different, and innovative ways to make fraudulent financial transactions.

The same goes for almost all AI models.

Human behavior is dynamic. Models need to adapt to them to remain useful.

We unconsciously keep changing our daily patterns, just as we see social trends keep changing. There are thousands of factors that apply to our day-to-day life, but one thing's for sure, most of them will not remain the same in the long term.

That's why we need to retrain our AI models to allow them to capture new patterns in several months/years.

3. Biases in our data can sometimes cause serious problems

"*Bias*" in data means that our data does not represent all the real-world possibilities equally. It is biased towards some specific type of data. For example, the police department wants to make an AI model that can predict whether a person has committed some type of crime or not in the past, just by seeing his/her face.

Now, let's say there are more brown toned people in our data who have committed some crime. Then our AI model will be more biased towards those people. Like, if a person's skin color is brown, then there are more chances of him/her having committed a crime. That's obviously not true.

But the model will think that the color of the skin is a more important pattern to capture in predicting whether a person has committed some crime or not.

This can cause serious problems.

4. Human help is still required

As per the current state of the AI field in general, it is not that capable of making decisions by itself. It needs human help very frequently. Like, once the training of AI models is done and we use it for real-time predictions, maintaining those models is not easy.

To maintain performance, we continuously need to train our AI model with new data and make sure that the data quality is high. Otherwise, performance will degrade.

It's unlikely that humans will get cut out of the loop entirely. They will, in one way or the other, depend on humans.

5. AI field is at a very early stage

Despite what we see in the news and social media about how AI and machine learning are performing better than humans, let me tell you that the AI field is at a very early stage. It will take a few years or even decades to make things mature enough to beat humans.

The fundamental fact we don't see here is that we humans just have one single brain to do it all.

Since in AI we need to use multiple models to do different tasks (we have seen this in "One machine can't do it all" in the limitations section of ML), it's good to keep in mind the limitations of the AI field while using it in day-to-day life.

Conclusion

In this chapter, we understood how you can start your journey in the AI/ML field. We also understood the advantages and limitations of the field, and we saw some common mistakes/pitfalls that we can avoid while using AI.

Questions

1. What are the three stages you can follow in your journey of AI/ML?
2. Which field can be disrupted by AI/ML?
3. What are the major pitfalls we can avoid while using AI?
4. What are the two major ways using which we can grow our knowledge in AI?
5. Why does the performance of ML models degrade over time?

And here's another quiz for you to check and refresh your knowledge.

Quiz time!

Here are some questions to check your knowledge about artificial intelligence (AI).

The answers are at the end of the book.

1. How do machines learn to do things?
2. Where do neural networks store their captured patterns/characteristics?
3. How do neural networks capture patterns in images?
4. What is loss function in machine learning? What's the use of "loss" in machine learning?
5. What are overfitting and under-fitting?
6. ML projects in the real world have multiple steps. One of them is validation. Why do we need that?
7. What is reward in reinforcement learning?
8. What is transfer learning? Why is it useful?
9. Suppose you want to predict the price of a share of a company after a week from now; how you will treat this problem — as classification or regression?
10. What do we do in clustering?

Answers for the quiz

1. In a nutshell, machines kind of learn from trial and error. They fail again and again in capturing meaningful patterns in the data. Eventually, they learn to capture meaningful patterns/characteristics. So, we call it "learning to do things".
2. Neural networks store their captured patterns in the form of "weights". These weights are nothing but strengths of connections between adjacent neurons. The higher the weight, the more likely the neuron will pass the incoming signal to the next neuron.
3. In images, neural networks go from tiny details to bigger details. First, it captures edges and boundaries. Then it

combines those small parts to make them into slightly bigger parts, like small textures. Then again it combines those to make larger parts. Eventually, it will have all the major parts of the image, and then it thinks, "In what type of object do all these major parts occur together?" That's how machines learn to capture patterns in images.

4. Loss tells us how far a model's predictions are from actual correct answers/labels.

5. "Overfitting" occurs when our ML model is failing to capture underlying common and generic patterns and captures other irrelevant features instead. "Under-fitting" occurs when our ML model does not have enough power to capture patterns in data.

6. We carry out validation to check whether the model has captured patterns that we wanted it to capture. We do it by testing it on different unseen data.

7. A reward is a result that the robot software/agent gets after it took some action in a certain environment. Positive reward means a good result and negative means a bad result. Rewards can be anything. It can be some physical thing or some virtual thing.

8. Transfer learning is a transfer of captured patterns/characteristics from one neural network to another neural network, so that the other neural network doesn't need to spend time capturing them again. As neural networks store those patterns in the form of weights, we just transfer weights from one neural network to another neural network.

9. Treating this problem as "regression" is a better choice. Because a stock price is a number and it can be anything between $0.00 and some bigger number like $10k-100k. There is no fixed number for a stock price, so it's better to treat it as regression.

10. Clustering is an unsupervised learning type problem. In clustering, we group similar items together and dissimilar items far away. The purpose of clustering is to know more relevant information about our data. We don't give them some labels as we don't have labels in unsupervised, we just group similar items.

References

Page No.	Link
vii	**https://www.zdnet.com/article/mckinsey-ai-will-create-13-trillion-in-value-by-2013/**
ix	**https://www.engadget.com/2017/12/01/tinder-ai-super-likeable/** **https://www.newscientist.com/article/2228752-ai-system-is-better-than-human-doctors-at-predicting-breast-cancer/**
57 (link 1)	**https://www.newscientist.com/article/2221840-deepminds-starcraft-playing-ai-beats-99-8-per-cent-of-human-gamers/**
57 (link 2)	**https://pubs.acs.org/doi/full/10.1021/acscentsci.7b00492**
57 (link 3)	**https://medium.com/@jeremyscohen/deep-reinforcement-learning-for-self-driving-cars-an-intro-4c8c08e6d06b**
83	**https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3**

84	**https://arxiv.org/ftp/arxiv/papers/1205/1205.6412.pdf** **https://aip.scitation.org/doi/abs/10.1063/1.4951885?journalCode=apc**
107	**https://www.wired.com/2015/09/whatsapp-serves-900-million-users-50-engineers/**
110	**https://www.cnbc.com/2019/06/28/80percent-of-the-stock-market-is-now-on-autopilot.html**
113	**https://deloitte.wsj.com/cfo/2013/10/08/why-putting-revenue-before-cost-is-a-competitive-advantage/**
114	**https://www.nytimes.com/2019/05/09/opinion/sunday/chris-hughes-facebook-zuckerberg.html**
116	**https://www.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/**
119	**https://feedvisor.com/resources/industry-news/dynamic-pricing-on-amazon-accelerates-overall-retail-growth/**
121	**https://medium.com/swlh/how-netflix-uses-ai-for-content-creation-and-recommendation-c1919efc0af4**
129	**https://www.coursera.org/learn/machine-learning** **https://www.youtube.com/view_play_list?p=A89DCFA6ADACE599**
134	**https://www.invespcro.com/blog/chatbots-customer-service/**
Figures	**Link**
1.1	**https://www.ripemedia.com/why-i-love-pixels-and-so-can-you/**
1.2	**https://socratic.org/questions/what-does-amplitude-measure**
1.3	**https://pixabay.com/vectors/brain-anatomy-physiology-human-153550/**
1.4	**https://www.mygreatlearning.com/blog/difference-data-science-machine-learning-ai/**
1.7	**https://blog.knoldus.com/first-interaction-artificial-neural-network/**
1.8	**https://www.edureka.co/blog/what-is-a-neural-network/**
1.10	**https://gmggroup.org/how-are-ai-machine-learning-big-data-deep-learning-and-data-science-interconnected/**

2.7	**https://www.dotnetlovers.com/article/214/decision-tree-analysis-with-example**
2.8	**https://medium.com/@rishabhjain_22692/decision-trees-it-begins-here-93ff54ef134**
2.9	**https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96**
2.13	**https://qr.ae/Tz1YDN**
2.17	**https://mozanunal.com/2019/11/img2sh/**
2.21	**https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a**
2.22	**https://www.kdnuggets.com/2017/10/7-types-artificial-neural-networks-natural-language-processing.html**
2.23	**https://towardsdatascience.com/reinforcement-learning-from-grid-world-to-self-driving-cars-52bd3e647bc4**
2.25	**https://algorithmia.com/blog/introduction-to-computer-vision**
2.29	**https://becominghuman.ai/what-exactly-does-cnn-see-4d436d8e6e52**
2.30	**https://becominghuman.ai/what-exactly-does-cnn-see-4d436d8e6e52**
2.32	**https://bitmovin.com/object-detection/**
2.34	**https://neurohive.io/en/popular-networks/u-net/**
2.35	**https://www.researchgate.net/figure/Example-of-2D-semantic-segmentation-Top-input-image-Bottom-prediction_fig3_326875064**
2.36	**https://arxiv.org/abs/1412.6572**
2.37	**https://arxiv.org/pdf/1712.09665.pdf**
2.38	**https://arxiv.org/pdf/1707.08945.pdf**
2.39	**https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/**
2.40	**http://jalammar.github.io/illustrated-word2vec/**
2.41	**https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/**
2.43	**https://pixabay.com/illustrations/dna-string-biology-3d-1811955/**

2.48	**https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection**
2.49	**https://arxiv.org/abs/1812.04948**
2.51	**https://pythonawesome.com/interpreting-the-latent-space-of-gans-for-semantic-face-editing/**
2.52	**http://ricee.or.kr/www/boardview/17/11457**
2.53	**https://heartbeat.fritz.ai/art-soul-part-1-a-style-transfer-website-based-on-tkinter-and-django-7a897741618**
2.54	**https://www.researchgate.net/figure/Generative-Adversarial-Network-Architecture_fig3_334100947**
2.55	**https://towardsdatascience.com/build-your-own-recommender-system-within-5-minutes-30dd40388fbf**
3.6	**https://landing.ai/ai-transformation-playbook/**
3.7	**https://sparktoro.com/**

Index